RK30xx
Technical Reference Manual

Revision 1.0
Feb. 2012

Revision History

Date	Revision	Description
2012-02-29	1.0	Initial Release

Table of Content

Figure Index

Table Index

Acronym Descriptions

Chapter 1 Introduction

RK30xx is a low power, high performance processor for mobile phones, personal mobile internet device and other digital multimedia applications, and integrates dual-coreCortex-A9 with separately NEONand FPU coprocessor.

Many embedded powerful hardware engines provide optimized performance for high-end application. RK30xx supports almost full-format video decoder by 1080p@60fps, also support H.264/MVC/VP8 encoder by 1080p@30fps, high-quality JPEG encoder/decoder, special image preprocessor and postprocessor .

Embedded 3D GPU makes RK30xx completely compatible with OpenGL ES2.0 and 1.1, OpenVG 1.1. Special 2D hardware engine with MMU will maximize display performance and provide very smoothly operation.

RK30xx has high-performance external memory interface(DDR3/LPDDR2/LVDDR3) capable of sustaining demanding memory bandwidths, also provides a complete set of peripheral interface to support very flexible applications as follows :

- 2 banks, 8bits/16bits Nor Flash/SRAM interface
- 8 banks, 8bits/16bits async Nand Flash,LBA Nand Flash and 8bits sync ONFI Nand Flash, allup to 60bits hardware ECC
- Totally 2GB memory space for 2 ranks, 16bits/32bits DDR3-800 , LPDDR2-800, LVDDR3-800
- Totally 3-channels SD/MMC interface to support MMC4.41, SD3.0, SDIO3.0 or eMMC
- 2-channels TFT LCD interface with 5-layers and 3D-display, 1920x1080 maximum display size
- HDMI tx interface(ver1.4) to support 3D-video1080p@30Hz
- 2-channels, 8bits CCIR656 interface and 10bits/12bits raw data interface with image preprocessor
- Lots of audio interface : two 2ch I2S/PCM interface, one 8ch I2S/PCM interface and SPDIF tx interface
- One USB OTG 2.0 and one USB Host2.0 interface
- 10M/100M RMII interface
- High-speed ADC interface and TS stream interface
- Lots of low-speed peripheral interface : 5I2C, 4UART, 2SPI,4PWM

This document will provide guideline on how to use RK30xx correctly and efficiently. The chapter 1 and chapter 2 will introduce the features, block diagram, signal descriptions and system usage of RK30xx, the chapter 3 through chapter 45 will describe the full function of each module in detail.

1.1 Features

1.1.1 MicroProcessor

- Dual-core ARM Cortex-A9 MPCore processor , a high-performance, low-power and cached application processor
- Full implementation of the ARM architecture v7-A instruction set, ARM Neon Advanced SIMD (single instruction, multiple data) support for accelerated media and signal processing computation
- Superscalar, variable length, out-of-order pipeline with dynamic branch prediction, 8-stage pipeline
- Include VFPv3 hardware to support single and double-precision add, subtract, divide, multiply and accumulate, and square root operations
- SCU ensures memory coherency between the two CPUs

- Integrated timer and watchdog timer per CPU
- Integrated 32KB L1 instruction cache , 32KB L1 data cache with 4-way set associative
- 512KB unified L2 Cache
- Trustzone technology support
- Full coresight debug solution
  - Debug and trace visibility of whole systems
  - ETM trace support
  - Invasive and non-invasive debug
- 4 separate power domain to support internal power switch and externally turn on/off based on different application scenario
  - PD_A9_0: 1st Cortex-A9 + Neon + FPU + L1 I/D Cache
  - PD_A9_1: 2nd Cortex-A9 + Neon + FPU + L1 I/D Cache
  - PD_DBG: CoreSight-DK for Cortex-A9
  - PD_SCU: SCU + L2 Cache controller
- One isolated voltage domain to support DVFS
- Maximum frequency can be up to 750MHz@1.0V,worst case or 1.1GHz@1.1V,typical case

1.1.2 Memory Organization

- Internal on-chip memory
  - 10KB BootRom
  - 64KB internal SRAM for security and non-security access, detailed size is programmable
  - 256KB or 512KB internal SRAM shared with L2 Cache Memory
- External off-chip memory①
  - DDR3-800, 16/32bits data widths, 2 ranks, totally 2GB(max) address space, maximum address space for one rank is also 2GB.
  - LPDDR-400, 32bits data width, 2 ranks, totally 2GB(max) address space, maximum address space for one rank is also 2GB.
  - LPDDR2-800, 32bits data width, 2 ranks, totally 2GB(max) address space, maximum address space for one rank is also 2GB.
  - Async SRAM/Nor Flash, 8/16bits data width,2banks,1MB(max) address space per bank
  - Async Nand Flash(include LBA Nand), 8/16bits data width,8 banks,60bits ECC
  - Sync ONFI Nand Flash , 8bits data width, 8 banks, 60bits ECC

1.1.3 Internal Memory

- Internal BootRom
  - Size : 10KB
  - Support system boot from the following device :
    - 8bits/16bits Async Nand Flash
    - 8bits ONFI Nand Flash
    - SPI0 interface
    - eMMC interface
  - Support system code download by the following interface:
    - USB OTG interface
    - UART0 Interface
- Internal SRAM
  - Size : 64KB
  - Support security and non-security access
  - Security or non-security space is software programmable

- Security space can be 0KB,4KB,8KB,12KB,16KB,…,60KB,64KB continuous size

- 256KB or 512KB internal SRAM shared with L2 Cache for Cortex-A9, size is configurable by software.

1.1.4 External Memory or Storage device

- Dynamic Memory Interface (DDR3/LPDDR/LPDDR2)
  - Compatible with JEDEC standard DDR3/LPDDR/LPDDR2 SDRAM
  - Data rates up to 800Mbps(400MHz) for DDR3/LPDDR2 and up to 400Mbps(200MHz) for LPDDR
  - Support up to 2 ranks (chip selects), totally 2GB(max) address space, maximum address space for one rank is also 2GB, which is software-configurable.
  - 16bits/32bits data width is software programmable
  - 5 host ports with 64bits AXI bus interface for system access, AXI bus clock is asynchronous with DDR clock
  - Programmable timing parameters to support DDR3/LPDDR/LPDDR2 SDRAM from various vendor
  - Advanced command reordering and scheduling to maximize bus utilization
  - Low power modes, such as power-down and self-refresh for DDR3/LPDDR/LPDDR2 SDRAM; clock stop and deep power-down for LPDDR/LPDDR2 SDRAM
  - Compensation for board delays and variable latencies through programmable pipelines
  - Embedded dynamic drift detection in the PHY to get dynamic drift compensation with the controller
  - Programmable output and ODT impedance with dynamic PVT compensation
  - Support one low-power work mode: power down DDR PHY and most of DDR IO except two cs and two cke output signals , make SDRAM still in self-refresh state to prevent data missing.

- Static Memory Interface (ASRAM/Nor Flash)
  - Compatible with standard async SRAM or Nor Flash
  - Support up to 2 banks (chip selects), maximum 1MB address space per bank
  - For bank0, 8bits/16bits data width is software programmable; For bank1, 16bits data width is fixed
  - Support separately data and address bus, also support shared data and address bus to save IO numbers

- Nand Flash Interface
  - Support 8bits/16bits async nand flash, up to 8 banks
  - Support 8bits sync DDR nand flash, up to 8 banks
  - Support LBA nand flash in async or sync mode
  - Up to 60bits hardware ECC
  - For DDR nand flash, support DLL bypass and 1/4 or 1/8 clock adjust, maximum clock rate is 75MHz
  - For async nand flash, support configurable interface timing , maximum data rate is 16bit/cycle
  - Embedded special DMA interface to do data transfer
  - Also support data transfer together with general DMAC1 in SoC system

- eMMC Interface
  - Compatible with standard iNAND interface
  - Support MMC4.41 protocol
  - Provide eMMC boot sequence to receive boot data from external eMMC device
  - Support combined single FIFO(32x32bits) for both transmit and receive operations
  - Support FIFO over-run and under-run prevention by stopping card clock automatically
  - Support CRC generation and error detection
  - Embedded clock frequency division control to provide programmable baud rate
  - Support block size from 1 to 65535Bytes
  - 8bits data bus width

- SD/MMC Interface
  - Compatible with SD3.0, MMC ver4.41
  - Support combined single FIFO(32x32bits) for both transmit and receive operations
  - Support FIFO over-run and under-run prevention by stopping card clock automatically
  - Support CRC generation and error detection
  - Embedded clock frequency division control to provide programmable baud rate
  - Support block size from 1 to 65535Bytes
  - Data bus width is 4bits

1.1.5 System Component

- CRU (clock & reset unit)
  - Support clock gating control for individual components inside RK30xx
  - One oscillator with 24MHz clock input and 4 embedded PLLs
  - Up to 1.5GHz clock output for all PLLs
  - Support global soft-reset control for whole SOC, also individual soft-reset for every components

- PMU(power management unit)
  - 7 work modes(slow mode, normal mode, idle mode, deep-idle mode, stop mode, sleep mode, power-off mode) to save power by different frequency or automatical clock gating control or power domain on/off control
  - Lots of wakeup sources in different mode
  - 3 separate voltage domains
  - 9 separate power domains, which can be power up/down by software based on different application scenes

- Timer
  - 3 on-chip 32bits Timersin SoC with interrupt-based operation
  - Provide two operation modes: free-running and user-defined count
  - Support timer work state checkable
  - Fixed 24MHz clock input

- PWM
  - Four on-chip PWMs with interrupt-based operation
  - Programmable 4-bit pre-scalar from apb bus clock

- Embedded 32-bit timer/counter facility
- Support single-run or continuous-run PWM mode
- Provides reference mode and output various duty-cycle waveform

- WatchDog
  - 32 bits watchdog counter width
  - Counter clock is from apb bus clock
  - Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
  - WDT can perform two types of operations when timeout occurs:
    - Generate a system reset
    - First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
  - Programmable reset pulse length
  - Totally 16 defined-ranges of main timeout period

- Bus Architecture
  - 64-bit multi-layer AXI/AHB/APB composite bus architecture
  - 5 embedded AXI interconnect
    - CPU interconnect with three 64-bits AXI masters, two 64-bits AXI slaves, one 32-bits AHB master and lots of 32-bits AHB/APB slaves
    - PERI interconnect with two 64-bits AXI masters, one 64-bits AXI slave, one 32-bits AXI slave, four 32-bits AHB masters and lots of 32-bits AHB/APB slaves
    - Display interconnect with six 64-bits AXI masters and one 32-bits AHB slave
    - GPU interconnect with one 128-bits AXI master and one 32-bits APB slave ,they are point-to-point AXI-lite architecture
    - VCODEC interconnect also with one 64-bits AXI master and one 32-bits AHB slave ,they are point-to-point AXI-lite architecture
  - For each interconnect with AXI/AHB/APB composite bus, clocks for AXI/AHB/APB domains are always synchronous, and different integer ratio is supported for them.
  - Flexible different QoS solution to improve the utility of bus bandwidth

- Interrupt Controller
  - Support 3 PPI interrupt source and 76 SPI interrupt sources input from different components inside RK30xx
  - Support 16 softwre-triggered interrupts
  - Input interrupt level is fixed , only high-level sensitive
  - Two interrupt outputs (nFIQ and nIRQ) separatelyfor each Cortex-A9, both are low-level sensitive
  - Support different interrupt priority for each interrupt source, and they are always software-programmable

- DMAC
  - Micro-code programming based DMA
  - The specific instruction set provides flexibility for programming DMA transfers
  - Linked list DMA function is supported to complete scatter-gather transfer
  - Support internal instruction cache
  - Embedded DMA manager thread

- Support data transfer types with memory-to-memory, memory-to-peripheral, peripheral-to-memory
- Signals the occurrence of various DMA events using the interrupt output signals
- Mapping relationship between each channel and different interrupt outputs is software-programmable
- Two embedded DMA controller , DMAC0 is for cpu system, DMAC1 is for peri system
- DMAC0 features:
  - 6 channels totally
  - 11 hardware request from peripherals
  - 2 interrupt output
  - Dual APB slave interface for register config, designated as secure and non-secure
  - Support trustzone technology and programmable secure state for each DMA channel
- DMAC1 features:
  - 7 channels totally
  - 13 hardware request from peripherals
  - 2 interrupt output
  - Not support trustzone technology

- Security system
  - Support trustzone technology for the following components inside RK30xx
    - Cortex-A9, support security and non-security mode, switch by software
    - DMAC0, support some dedicated channels work only in security mode
    - eFuse, only accessed by Cortex-A9 in security mode
    - Internal memory , part of space is addressed only in security mode, detailed size is software-programmable together with TZMA(trustzone memory adapter) and TZPC(trustzone protection controller)

1.1.6 Video CODEC

- Shared internal memory and bus interface for video decoder and encoder②
- Video Decoder
  - Real-time video decoder of MPEG-1, MPEG-2, MPEG-4,H.263, H.264 , AVS , VC-1 , RV , VP6/VP8 , Sorenson Spark, MVC
  - Error detection and concealment support for all video formats
  - Output data format is YUV420 semi-planar, and YUV400(monochrome) is also supported for H.264
  - H.264 up to HP level 4.2 : 1080p@60fps (1920x1088)③
  - MPEG-4 up to ASP level 5 : 1080p@60fps (1920x1088)
  - MPEG-2 up to MP : 1080p@60fps (1920x1088)
  - MPEG-1 up to MP : 1080p@60fps (1920x1088)
  - H.263 : 576p@60fps (720x576)
  - Sorenson Spark : 1080p@60fps (1920x1088)
  - VC-1 up to AP level 3 : 1080p@30fps (1920x1088)
  - RV8/RV9/RV10 : 1080p@60fps (1920x1088)
  - VP6/VP8 : 1080p@60fps (1920x1088)
  - AVS : 1080p@60fps (1920x1088)
  - MVC : 1080p@60fps (1920x1088)

- For AVS, 4:4:4 sampling not supported
- For H.264, Image cropping not supported
- For MPEG-4,GMC(global motion compensation) not supported
- For VC-1, upscaling and range mapping are supported in image post-processor
- For MPEG-4 SP/H.263/Sorenson spark, using a modified H.264 in-loop filter to implement deblocking filter in post-processor unit

- Video Encoder
  - Support video encoder for H.264 (BP@level4.0, MP@level4.0, HP@level4.0), MVC and VP8
  - Only support I and P slices, not B slices
  - Support error resilience based on constrained intra prediction and slices
  - Input data format :
    - YCbCr 4:2:0 planar
    - YCbCr 4:2:0 semi-planar
    - YCbYCr 4:2:2
    - CbYCrY 4:2:2 interleaved
    - RGB444 and BGR444
    - RGB555 and BGR555
    - RGB565 and BGR565
    - RGB888 and BRG888
    - RGB101010 and BRG101010
  - Image size is from 96x96 to 1920x1088(Full HD)
  - Maximum frame rate is up to 30fps@1920x1080③
  - Bit rate supported is from 10Kbps to 20Mbps

1.1.7 JPEG CODEC

- JPEG decoder
  - Input JPEG file : YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 sampling formats
  - Output raw image : YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
  - Decoder size is from 48x48 to 8176x8176(66.8Mpixels)
  - Maximum data rate④ is up to 76million pixels per second

- JPEG encoder
  - Input raw image :
    - YCbCr 4:2:0 planar
    - YCbCr 4:2:0 semi-planar
    - YCbYCr 4:2:2
    - CbYCrY 4:2:2 interleaved
    - RGB444 and BGR444
    - RGB555 and BGR555
    - RGB565 and BGR565
    - RGB888 and BRG888
    - RGB101010 and BRG101010
  - Output JPEG file : JFIF file format 1.02 or Non-progressive JPEG
  - Encoder image size up to 8192x8192(64million pixels) from 96x32
  - Maximum data rate④ up to 90million pixels per second

1.1.8 Image Enhancement

- Image pre-processor

- Only used together with HD video encoder inside RK30xx , not support stand-alone mode
- Provides RGB to YCbCr 4:2:0 color space conversion, compatible with BT.601 , BT.709 or user defined coefficients
- Provides YCbCr4:2:2 to YCbCr4:2:0 color space conversion
- Support cropping operation from 8192x8192 to any supported encoding size
- Support rotation with 90 or 270 degrees

- Video stabilization
  - Work in combined mode with HD video encoder inside RK30xx and stand-alone mode
  - Adaptive motion compensation filter
  - Support scene detection from video sequence, encodes key frame when scene change noticed

- Image post-processor(embedded inside video decoder)
  - Combined with HD video decoder and JPEG decoder, post-processor can read input data directly from decoder output to reduce bus bandwidth
  - Also work as a stand-alone mode, its input data is from a camera interface or other image data stored in external memory
  - Input data format :
    - any format generated by video decoder in combined mode
    - YCbCr 4:2:0 semi-planar
    - YCbCr 4:2:0 planar
    - YCbYCr 4:2:2
    - YCrYCb 4:2:2
    - CbYCrY 4:2:2
    - CrYCbY 4:2:2
  - Ouput data format:
    - YCbCr 4:2:0 semi-planar
    - YCbYCr 4:2:2
    - YCrYCb 4:2:2
    - CbYCrY 4:2:2
    - CrYCbY 4:2:2
    - Fully configurable ARGB channel lengths and locations inside 32bits, such as ARGB8888,RGB565,ARGB4444 etc.
  - Input image size:
    - Combined mode : from 48x48 to 8176x8176 (66.8Mpixels)
    - Stand-alone mode : width from 48 to 8176,height from 48 to 8176, and maximum size limited to 16.7Mpixels
    - Step size is 16 pixels
  - Output image size: from 16x16 to 1920x1088 (horizontal step size 8,vertical step size 2)
  - Support image up-scaling :
    - Bicubic polynomial interpolation with a four-tap horizontal kernel and a two-tap vertical kernel
    - Arbitrary non-integer scaling ratio separately for both dimensions
    - Maximum output width is 3x input width
    - Maximum output height is 3x input height
  - Support image down-scaling:
    - Arbitrary non-integer scaling ratio separately for both dimensions
    - Unlimited down-scaling ratio
  - Support YUV to RGB color conversioin, compatible with BT.601-5,

BT.709 and user definable conversion coefficient
- Support dithering (2x2 ordered spatial dithering for 4,5,6bit RGB channel precision
- Support programmable alpha channel and alpha blending operation with the following overlay input formats:
  - 8bit alpha +YUV444, big endian channel order with AYUV8888
  - 8bit alpha +24bit RGB, big endian channel order with ARGB8888
- Support deinterlacing with conditional spatial deinterlace filtering, only compatible with YUV420 input format
- Support RGB image contrast / brightness / color saturation adjustment
- Support image cropping & digital zoom only for JPEG or stand-alone mode
- Support picture in pcture
- Support image rotation (horizontal flip, vertical flip, rotation 90,180 or 270 degrees)

- Image Post-Processor (IPP)(standalone)
  - memory to memory mode
  - input data format and size
    - RGB888 : 16x16 to 8191x8191
    - RGB565 : 16x16 to 8191x8191
    - YUV422/YUV420 : 16x16 to 8190x8190
    - YUV444 : 16x16 to 8190x8190
  - pre scaler
    - integer down-scaling(ratio: 1/2,1/3,1/4,1/5,1/6,1/7,1/8) with linear filter
    - deinterlace(up to 1080i) to support YUV422&YUV420 input format
  - post scaler
    - down-scaling with 1/2 ~ 1 arbitary non-integer ratio
    - up-scaling with 1~4 arbitary non-integer ratio
    - 4-tap vertical, 2-tap horizontal filter
    - The max output image width of post scaler is 4096
  - Support rotation with 90/180/270 degrees and x-mirror,y-mirror

1.1.9 Graphics Engine

- 3D Graphics Engine :
  - High performanceOpenGL ES1.1 and 2.0, OpenVG1.1 etc.
  - Embedded 4 shader cores with shared hierarchical tiler
  - Separate vertex(geometry) and fragment(pixel) processing for maximum parallel throughput
  - Provide MMU and L2 Cache with 128KB size
  - Triangle rate : 30M triangles/s
  - Pixel rate: 1.4G pixels/s @ 350MHz

- 2D Graphics Engine :
  - Pixel rate: 300M pixel/s without scale, 150M pixel/s with bilinear scale, 75M pixel/s with bicubic scale.
  - Bit Blit with Strength Blit, Simple Blit and Filter Blit
  - Color fill with gradient fill, and pattern fill
  - Line drawing with anti-aliasing and specified width
  - High-performance stretch and shrink
  - Monochrome expansion for text rendering
  - ROP2, ROP3, ROP4 full alpha blending and transparency
  - Alpha blending modes including Java 2 Porter-Duff compositing

blending rules , chroma key, and pattern mask
- 8K x 8K raster 2D coordinate system
- Arbitrary degrees rotation with anti-aliasing on every 2D primitive
- Programmable bicubic filter to support image scaling
- Blending, scaling and rotation are supported in one pass for stretch blit
- Source formats :
  - ABGR8888, XBGR888, ARGB8888, XRGB888
  - RGB888, RGB565
  - RGBA5551, RGBA4444
  - YUV420 planar, YUV420 semi-planar
  - YUV422 planar, YUV422 semi-planar
  - BPP8, BPP4, BPP2, BPP1
- Destination formats :
  - ABGR8888, XBGR888, ARGB8888, XRGB888
  - RGB888, RGB565
  - RGBA5551, RGBA4444
  - YUV420 planar, YUV420 semi-planar only in filter and pre-scale mode
  - YUV422 planar, YUV422 semi-planar only in filter and pre-scale mode

1.1.10 Video IN/OUT

- Camera Interface
  - 2 independent camera interface controller
  - Support up to 5M pixels
  - 8bits CCIR656(PAL/NTSC) interface
  - 8bits/10bits/12bits raw data interface
  - YUV422 data input format with adjustable YUV sequence
  - YUV422,YUV420 output format with separately Y and UV space
  - Support picture in picture (PIP)
  - Support simple image effects such as Arbitrary(sepia), Negative, Art freeze, Embossing etc.
  - Support static histogram statistics and white balance statistics
  - Support image crop with arbitrary windows
  - Support scale up/down from 1/8 to 8 with arbitrary non-integer ratio

- Display Interface
  - 2 independent display controller
  - Support LCD or TFT interfaces up to 1920x1080
  - Support HDMI 1.4a output up to 1080p@30fps
  - Support TV interface with ITU-R BT.656 (8bits, 480i/576i/1080i)
  - Parallel RGB LCD Interface : RGB888(24bits),RGB666(18bits),RGB565(15bits)
  - Serial RGB LCD Interface: 3x8bits with RGB delta support, 3x8bits followed by dummy data, 16bits followed by 8bits
  - MCU LCD interface : i-8080 with up to 24bits RGB
  - 5 display layers :
    - One background layer with programmable 24bits color
    - One video layer (win0)
      - RGB888, ARGB888, RGB565, YUV422, YUV420, AYUV
      - maximum resolution is 1920x1080
      - 1/8 to 8 scaling up/down engine with arbitrary non-integer ratio
      - 256 level alpha blending
      - Support transparency color key

- Support 3D display
- One video layer (win1)
  - RGB888, ARGB888, RGB565, YUV422, YUV420,AYUV
  - maximum resolution is 1920x1080
  - 1/8 to 8 scaling up/down engine with arbitrary non-integer ratio
  - 256 level alpha blending
  - Support transparency color key
- One OSD layer(win2)
  - RGB888, ARGB888, RGB565, 1/2/4/8BPP
  - 256 level alpha blending
  - transparency color key
- Hardware cursor(win3)
  - 2BPP
  - Maximum resolution 64x64
  - 3-color and transparent mode
  - 2-color + transparency + tran_invert mode
  - 16 level alpha blending

- Support 180 rotation in combined mode with LCDC or separately mode
- 3 x 256 x 8 bits display LUTs
- Win0 and Win1 layer overlay exchangeable
- Support color space conversion :
  YUV2RGB(rec601-mpeg/rec601-jpeg/rec709) and RGB2YUV
- Deflicker support for interlace output
- Support replication(16bits to 24bits) and dithering(24bits to 16bits/ 18bits) operation

- HDMI TX Interface
  - HDMI version 1.4a, HDCP revision 1.4 and DVI version 1.0 compliant transmitter
  - Supports DTV from 480i to 1080i/p HD resolution, and PC from VGA to UXGA by LCDC0 or LCDC1 in RK30xx
  - Supports 3D and 2k x 4k video resolution output
  - Programmable 2-way color space converter
  - Compliant with EIA/CEA-861D
  - Deep color supported up to 12bit per pixel.
  - xvYCC Enhanced Colorimetry
  - Gamut Metadata transmission
  - Supports RGB, YCbCr digital video input format includes ITU.656
  - 36bit RGB/YCbCr 4:4:4
    16/20/24bit YCbCr 4:2:2
    8/10/12bit YCbCr 4:2:2 (ITU.601 and 656)
  - Supports standard SPDIF for stereo or compressed audio up to 192KHz by SPDIF controller in RK30xx
  - Support PCM, Dolby digital, DTS digital audio transmission through 8ch I2S controller in RK30xx
  - Wide range channel speed up to 2.2Gbps

1.1.11 Audio Interface

- I2S/PCM with 8ch
  - Up to 8 channels (4xTX , 2xRX)
  - Audio resolution from 16bits to 32bits
  - Sample rate up to 192KHz
  - Provides master and slave work mode, software configurable
  - Support 3 I2S formats (normal , left-justified , right-justified)

- Support 4 PCM formats(early , late1 , late2 , late3)
- I2S and PCM mode cannot be used at the same time

- I2S/PCM with 2ch
  - 2 independent I2S/PCM interface with 2 channels
  - Up to 2 channels (2xTX, 2xRX)
  - Audio resolution from 16bits to 32bits
  - Sample rate up to 192KHz
  - Provides master and slave work mode, software configurable
  - Support 3 I2S formats (normal , left-justified , right-justified)
  - Support 4 PCM formats(early , late1 , late2 , late3)
  - I2S and PCM mode cannot be used at the same time

- SPDIF
  - Audio resolution : 16bits/20bits/24bits
  - Software configurable sample rates (48KHz, 44.1KHz, 32KHz)
  - Stereo voice replay with 2 channels

1.1.12 Connectivity

- SDIO interface
  - Compatible with SDIO 3.0 protocol
  - Support FIFO over-run and under-run prevention by stopping card clock automatically
  - 4bits data bus widths

- High-speed ADC & TS stream interface
  - Support single-channel 8bits/10bits interface
  - DMA-based and interrupt-based operation
  - Support 8bits TS stream interface
  - Support PID filter operation
    - Combined with high-speed ADC interface to implement filter from original TS data
    - Provide PID filter up to 64 channels PID simultaneously
    - Support sync-byte detection in transport packet head
    - Support packet lost mechanism in condition of limited bandwidth

- MAC 10/100MEthernet Controller
  - IEEE802.3u compliant Ethernet Media Access Controller(MAC)
  - Support only RMII(Reduced MII) mode
  - 10Mbps and 100Mbps compatible
  - Automatic retry and automatic collision frame deletion
  - Full duplex support with flow-control
  - Address filtering(broadcast, multicast, logical, physical)
  - Clock can be from RK30xx or external ethernet PHY

- SPI Controller
  - 2 on-chip SPI controller inside RK30xx
  - Support serial-master and serial-slave mode, software-configurable
  - DMA-based or interrupt-based operation
  - Embedded two 32x16bits FIFO for TX and RX operation respectively
  - Support 2 chip-selects output in serial-master mode

- Uart Controller
  - 4 on-chip uart controller inside RK30xx

- DMA-based or interrupt-based operation
- For UART1/UART2/UART3, Embedded two 32Bytes FIFO for TX and RX operation respectively
- For UART0, two 64Bytes FIFOs are embedded for TX/RX operation
- Support 5bit,6bit,7bit,8bit serial data transmit or receive
- Standard asynchronous communication bits such as start,stop and parity
- Support different input clock for uart operation to get up to 4Mbps or other special baud rate
- Support non-integer clock divides for baud clock generation
- Auto flow control mode is only for UART0, UART1, UART2

- I2C controller
  - 5 on-chip I2C controller in RK30xx
  - Multi-master I2C operation
  - Support 7bits and 10bits address mode
  - Software programmable clock frequency and transfer rate up to 400Kbit/s in the fast mode
  - Serial 8bits oriented and bidirectional data transfers can be made at up to 100Kbit/s in the standard mode

- GPIO
  - 6 groups of GPIO (GPIO0~GPIO4, GPIO6) , 32 GPIOs per group in GPIO0~GPIO4, and 16GPIOs in GPIO6, totally have 176 GPIOs
  - All of GPIOs can be used to generate interrupt to Cortex-A9
  - GPIO6 can be used to wakeup system from stop/sleep/power-off mode
  - All of pullup GPIOs are software-programmable for pullup resistor or not
  - All of pulldown GPIOs are software-programmable for pulldown resistor or not
  - All of GPIOs are always in input direction in default after power-on-reset

- USB Host2.0
  - Compatible with USB Host2.0 specification
  - Supports high-speed(480Mbps), full-speed(12Mbps) and low-speed(1.5Mbps) mode
  - Provides 16 host mode channels
  - Support periodic out channel in host mode

- USB OTG2.0
  - Compatible with USB OTG2.0 specification
  - Supports high-speed(480Mbps), full-speed(12Mbps) and low-speed(1.5Mbps) mode
  - Support up to 9 device mode endpoints in addition to control endpoint 0
  - Support up to 6 device mode IN endpoints including control endpoint 0
  - Endpoints 1/3/5/7 can be used only as data IN endpoint
  - Endpoints 2/4/6 can be used only as data OUT endpoint
  - Endpoints 8/9 can be used as data OUT and IN endpoint
  - Provides 9 host mode channels

1.1.13 Others

- Temperature Sensor
  - 2 bipolar-based temperature-sensing cell embedded

- 2-channel 12-bits SAR ADC
- Temperature accuracy sensed is ±5 degree
- SAR-ADC clock must be less than 50KHz
- Standby Current is about 180uA for analog and 40uA for digital logic
- Power Down Current is about 1uA for anolog and 5uA for digital logic

- SAR-ADC(Successive Approximation Register)
  - 4-channel single-ended 10-bit SAR analog-to-digital converter
  - Conversion speed range is up to 1 MSPS
  - SAR-ADC clock must be less than 1MHz
  - DNL is less than ±1 LSB , INL is less than ±2.0 LSB
  - Power down current is about 0.5uA for analog and digital logic
  - Power supply is 2.5V (±10%) for analog interface

- eFuse
  - 256bits (32x8) high-density electrical Fuse
  - Programming condition：VDDQ must be 2.5V(±10%)
  - Program time is about 10us(±1us)
  - Read condition：VDDQ must be 0V or floating
  - Support power-down and standby mode

- Operation Temperature Range
  - -40°C to +85°C

- Operation Voltage Range
  - Core supply: 1.1V (±10%)
  - IO supply：3.3V or 2.5V or 1.8V (±10%)

- Process
  - TSMC 40nm LP

- Package Type
  - TFBGA453LD (body: 19mm x 19mm ; ball size：0.4mm ; ball pitch：0.8mm)

- Power
  - TBA

Notes :① : DDR3/LPDDR/LPDDR2 are not used simultaneously as well as async and sync ddr nand flash

②: In RK30xx, Video decoder and encoder are not used simultaneously because of shared internal buffer

③: Actual maximum frame rate will depend on the clock frequency and system bus performance

④: Actual maximum data rate will depend on the clock frequency and JPEG compression rate

1.2 Block Diagram

The following diagram shows the basic block diagram for RK30xx.

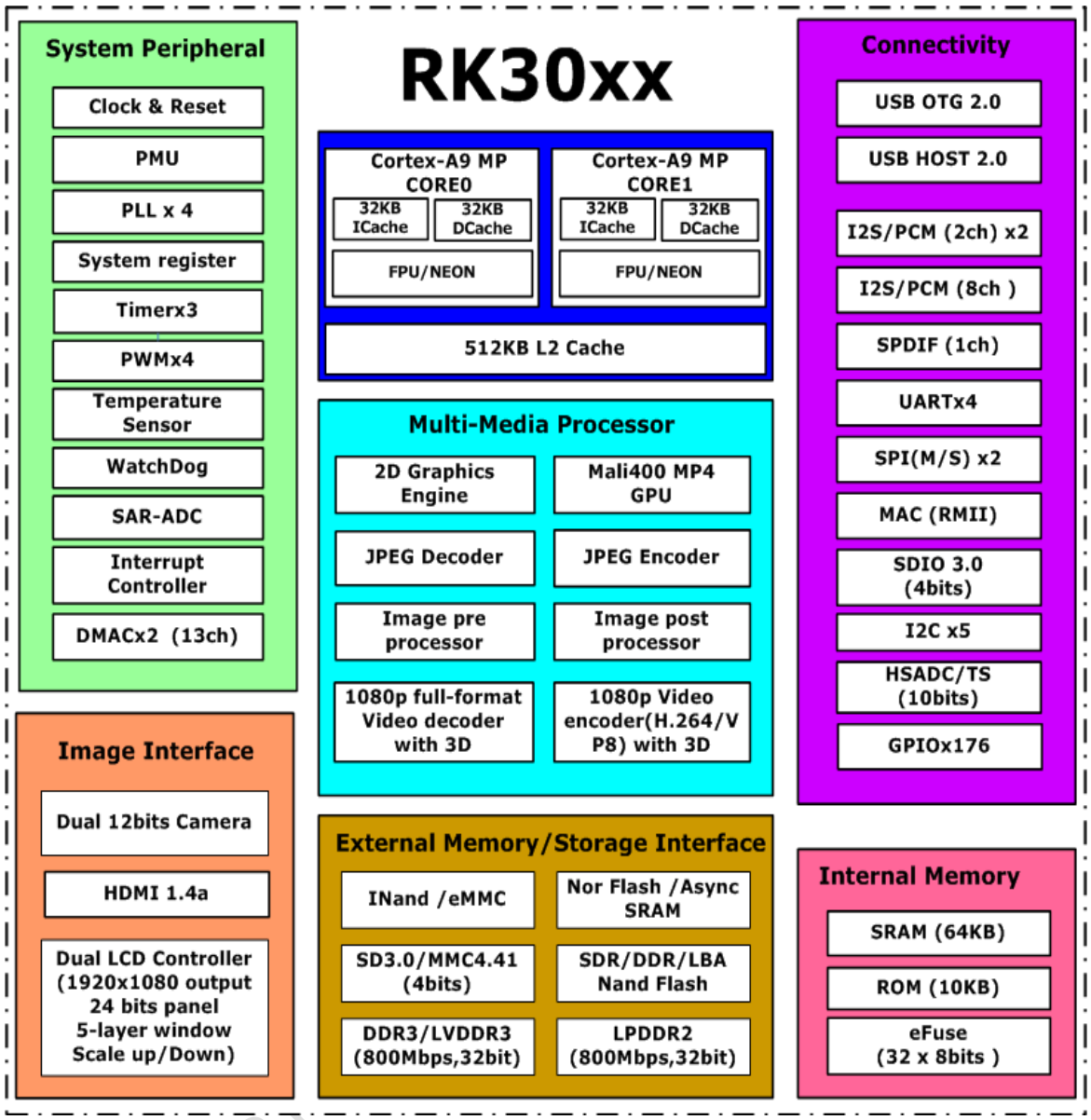


Fig. 1-1 RK30xx Block Diagram

1.3 Pin Description

In this chapter, the pin description will be divided into two parts, one is all power/ground descriptions in Table 1-1, include analog power/ground, another is all the function signals descriptions in Table 1-2, also include analog power/ground.

1.3.1 RK30xx power/ground IO descriptions

Table 1-1 RK30xx Power/Ground IO informations

Group	Ball #	Min(V)	Typ(V)	Max(V)	Descriptions

GND	H8,H9,H10,H11,H12,H13,H14, H15,H16,J8,J9,J10,J11,J12,J13, J14,J15,J16,K8,K9,K10,K11, K12,K13,K14,K15,K16,L8,L9, L10,L11,L12,L13,L14,L15,L16, M8,M9,M10,M11,M12,M13,M14, M15,M16,N8,N9,N10,N11,N12, N13,N14,N15,N16,P8,P9,P10, P11,P12,P13,P14,P15,P16,R8, R9,R10,R11,R12,R13,R14,R15, R16,T8,T9,T10,P7,R7,T7,T14, T15,T16,G12,G16,M17,D20, AB10,AA16,AA17	N/A	N/A	N/A	Internal Core Ground and Digital IO Ground
AVDD	N5,P5,P6,T5,T6,U5, U6,V5,U8,U9,V7,V8	0.99	1.1	1.21	Internal CPU Power (@ cpu frequency <= 1GHz)
		TBD	TBD	TBD	Internal CPU Power (@ cpu frequency <= 1.4GHz)
		TBD	TBD	TBD	Internal CPU Power (@ cpu frequency <= 600MHz)
CVDD	G11,G16,H7,J17, L17,N7,P17,T17	0.99	1.1	1.21	Internal Core Power w/o CPU and PMU logic
PVDD	U13	0.99	1.1	1.21	Internal PMU Domain Power
PVCC	V13	3	3.3	3.6	Digital GPIO Power for PMU Domain
VDDIO0	K17	3	3.3	3.6	Digital GPIO Power
VDDIO1	H17	3	3.3	3.6	
LCD0_VCC0	K7	3 1.62	3.3 1.8	3.6 1.98	LCDC0 Digital IO Power
LCD0_VCC1	J7	3 1.62	3.3 1.8	3.6 1.98	
LCD1_VCC	L7	3 1.62	3.3 1.8	3.6 1.98	LCDC1 Digital IO Power
CIF0_VCC	M7	3 1.62	3.3 1.8	3.6 1.98	Camera0 Digital IO Power
CIF1_VCC	N6	3	3.3	3.6	Camera1 Digital IO Power

		1.62	1.8	1.98	
SMC_VCC	R17	3 1.62	3.3 1.8	3.6 1.98	SMC Digital IO Power
FLASH_VCC	N17	3 1.62	3.3 1.8	3.6 1.98	Nand Flash Digital IO Power
AP0_VCC	U16	3 1.62	3.3 1.8	3.6 1.98	I2S1/UART0/SDIO Digital IO Power
AP1_VCC	V16	3 1.62	3.3 1.8	3.6 1.98	SPI0/UART1 Digital IO Power
MVDD	F10,F11,F14,F16,F17, G10,G13,G14	1.425 1.28 1.14	1.5 1.35 1.2	1.575 1.45 1.30	DDR3 Digital IO Power LVDDR3 Digital IO Power LPDDR2 Digital IO Power
AVSS_APLL	T11	N/A	N/A	N/A	ARM PLLAnalog Ground
AVDD_APLL	U10	0.99	1.1	1.21	ARM PLL Analog Power
AVSS_DPLL	T12	N/A	N/A	N/A	DDR PLL Analog Ground
AVDD_DPLL	U11	0.99	1.1	1.21	DDR PLLAnalog Power
AVSS_CGPLL	T13	N/A	N/A	N/A	CODEC/GENERAL PLL Analog Ground
AVDD_CGPLL	U12	0.99	1.1	1.21	CODEC/GENERAL PLL Analog Power
VDDA_SARADC VDDA_TSADC	V11	2.25	2.5	2.75	SAR-ADC/TS-ADC Analog Power
OTG_DVDD HOST_DVDD	U14	1.023	1.2	1.32	USB OTG2.0/Host2.0 Digital Power
OTG_VDD25 HOST_VDD25	U15	2.325	2.5	2.75	USB OTG2.0/Host2.0 Analog Power
OTG_VDD33 HOST_VDD33	V14	3.069	3.3	3.63	USB OTG2.0/Host2.0 Analog Power
EFUSE_VDDQ	W14	2.25 0	2.5 0	2.75 0	Program Power Supply for eFuse Read Power Supply for eFuse
HDMI_AVSS	B3,B6,C5,G8,G9	N/A	N/A	N/A	HDMI Analog Ground

HDMI_AVDD	F8	2.25	2.5	2.75	HDMI Analog Power Supply
HDMI_VDD	F7	0.99	1.1	1.21	HDMI Analog Power Supply

1.3.2 RK30xx function IO descriptions

Table 1-2 RK3066 IO descriptions

Pin Name	Ball #	func0	func1	func2	func3	Pad bype①	Drive ②	pull up/ down	Reset State③	Power supply⑤
Left Side④										
LCDC0_DATA[7]	E4	LCDC0_DATA[7]				I/O	8	N/A	I	LCD0_VCC0 LCD0_VCC1
LCDC0_DATA[8]	F4	LCDC0_DATA[8]				I/O	8	N/A	I	
LCDC0_DATA[9]	E3	LCDC0_DATA[9]				I/O	8	N/A	I	
LCDC0_DATA[10]	E2	LCDC0_DATA[10]				I/O	8	N/A	I	
LCDC0_DATA[11]	F3	LCDC0_DATA[11]				I/O	8	N/A	I	
LCDC0_DATA[12]	E1	LCDC0_DATA[12]				I/O	8	N/A	I	
LCDC0_DATA[13]	G5	LCDC0_DATA[13]				I/O	8	N/A	I	
LCDC0_DATA[14]	G4	LCDC0_DATA[14]				I/O	8	N/A	I	
LCDC0_DATA[15]	G6	LCDC0_DATA[15]				I/O	8	N/A	I	
LCDC0_DATA[16]	F2	LCDC0_DATA[16]				I/O	8	N/A	I	
LCDC0_DATA[17]	G3	LCDC0_DATA[17]				I/O	8	N/A	I	
LCDC0_DATA[18]	G2	LCDC0_DATA[18]				I/O	8	N/A	I	
LCDC0_DATA[19]	G1	LCDC0_DATA[19]				I/O	8	N/A	I	
LCDC0_DATA[20]	H4	LCDC0_DATA[20]				I/O	8	N/A	I	
LCDC0_DATA[21]	H3	LCDC0_DATA[21]				I/O	8	N/A	I	
LCDC0_DATA[22]	H2	LCDC0_DATA[22]				I/O	8	N/A	I	
LCDC0_DATA[23]	H5	LCDC0_DATA[23]				I/O	8	N/A	I	
GPIO2_A[0]	H6	GPIO2_A[0]	lcdc1_data0	smc_addr4		I/O	8	down	I	LCD1_VCC
GPIO2_A[1]	J2	GPIO2_A[1]	lcdc1_data1	smc_addr5		I/O	8	down	I	
GPIO2_A[2]	H1	GPIO2_A[2]	lcdc1_data2	smc_addr6		I/O	8	down	I	

GPIO2_B[5]	J3	GPIO2_B[5]	lcdc1_data13	smc_addr17	hsadc_data8	I/O	8	down	I	
GPIO2_B[6]	K6	GPIO2_B[6]	lcdc1_data14	smc_addr18	ts_sync	I/O	8	down	I	
GPIO2_B[7]	K5	GPIO2_B[7]	lcdc1_data15	smc_addr19	hsadc_data7	I/O	8	down	I	
GPIO2_C[0]	L5	GPIO2_C[0]	lcdc1_data16	gps_clk	hsadc_clkout	I/O	8	down	I	
GPIO2_C[1]	K4	GPIO2_C[1]	lcdc1_data17	smc_bls_n0	hsadc_data6	I/O	8	down	I	
GPIO2_C[2]	K3	GPIO2_C[2]	lcdc1_data18	smc_bls_n1	hsadc_data5	I/O	8	down	I	
GPIO2_C[3]	K2	GPIO2_C[3]	lcdc1_data19	spi1_clk	hsadc_data0	I/O	8	down	I	
GPIO2_C[4]	K1	GPIO2_C[4]	lcdc1_data20	spi1_csn0	hsadc_data1	I/O	8	down	I	
GPIO2_C[5]	L1	GPIO2_C[5]	lcdc1_data21	spi1_txd	hsadc_data2	I/O	8	down	I	
GPIO2_C[6]	L2	GPIO2_C[6]	lcdc1_data22	spi1_rxd	hsadc_data3	I/O	8	down	I	
GPIO2_C[7]	L3	GPIO2_C[7]	lcdc1_data23	spi1_csn1	hsadc_data4	I/O	8	down	I	
CIF0_DATAIN[2]	L6	CIF0_DATAIN[2]				I	8	down	I	CIF0_VCC
CIF0_DATAIN[3]	L4	CIF0_DATAIN[3]				I	8	down	I	
CIF0_DATAIN[4]	M3	CIF0_DATAIN[4]				I	8	down	I	
CIF0_DATAIN[5]	M2	CIF0_DATAIN[5]				I	8	down	I	
CIF0_DATAIN[6]	N1	CIF0_DATAIN[6]				I	8	down	I	
CIF0_DATAIN[7]	N2	CIF0_DATAIN[7]				I	8	down	I	
CIF0_DATAIN[8]	N3	CIF0_DATAIN[8]				I	8	down	I	
CIF0_DATAIN[9]	P1	CIF0_DATAIN[9]				I	8	down	I	
CIF0_VSYNC	N4	CIF0_VSYNC				I	8	down	I	
CIF0_HREF	P2	CIF0_HREF				I	8	down	I	
CIF0_CLKIN	P3	CIF0_CLKIN				I	8	down	I	
GPIO1_B[3]	R2	GPIO1_B[3]	cif0_clkout			I/O	4	down	I	
GPIO1_B[4]	R3	GPIO1_B[4]	cif0_data0			I/O	8	down	I	
GPIO1_B[5]	P4	GPIO1_B[5]	cif0_data1			I/O	8	down	I	

GPIO1_B[6]	T1	GPIO1_B[6]	cif0_data10			I/O	8	down	I	
GPIO1_B[7]	T2	GPIO1_B[7]	cif0_data11			I/O	8	down	I	
GPIO3_A[2]	T3	GPIO3_A[2]	i2c3_sda			I/O	8	up	I	
GPIO3_A[3]	U1	GPIO3_A[3]	i2c3_scl			I/O	8	up	I	
GPIO1_C[0]	U2	GPIO1_C[0]	cif1_data2	rmii_clkout	rmii_clkin	I/O	4	down	I	CIF1_VCC
GPIO1_C[1]	U3	GPIO1_C[1]	cif1_data3	rmii_tx_en		I/O	4	down	I	
GPIO1_C[2]	V2	GPIO1_C[2]	cif1_data4	rmii_txd1		I/O	4	down	I	
GPIO1_C[3]	V3	GPIO1_C[3]	cif1_data5	rmii_txd0		I/O	4	down	I	
GPIO1_C[4]	T4	GPIO1_C[4]	cif1_data6	rmii_rx_err		I/O	8	down	I	
GPIO1_C[5]	U4	GPIO1_C[5]	cif1_data7	rmii_crs_dvalid		I/O	8	down	I	
GPIO1_C[6]	W1	GPIO1_C[6]	cif1_data8	rmii_rxd1		I/O	8	down	I	
GPIO1_C[7]	V4	GPIO1_C[7]	cif1_data9	rmii_rxd0		I/O	8	down	I	
GPIO1_D[0]	W2	GPIO1_D[0]	cif1_vsync	mii_md		I/O	8	down	I	
GPIO1_D[1]	W3	GPIO1_D[1]	cif1_href	mii_mdclk		I/O	8	down	I	
GPIO1_D[2]	W4	GPIO1_D[2]	cif1_clkin			I/O	8	down	I	
GPIO1_D[7]	Y1	GPIO1_D[7]	cif1_clkout			I/O	4	down	I	
GPIO1_D[6]	Y2	GPIO1_D[6]	cif1_data11			I/O	8	down	I	
GPIO3_A[4]	Y4	GPIO3_A[4]	i2c4_sda			I/O	8	up	I	
GPIO3_A[5]	Y3	GPIO3_A[5]	i2c4_scl			I/O	8	up	I	
Bottom Side										
ARMP_power_feedback	V10	1.1V				P	N/A	N/A	N/A	
VDDA_SARADC	V11	2.5V				AP	N/A	N/A	N/A	SARADC Domain
SARADC_AIN[2]	W7	SARADC_AIN[2]				A	N/A	N/A	N/A	
SARADC_AIN[1]	W8	SARADC_AIN[1]				A	N/A	N/A	N/A	

SARADC_AIN[0]	W6	SARADC_AIN[0]				A	N/A	N/A	N/A	
AVSS_CGPLL	T13	Analog Ground				AG	N/A	N/A	N/A	PLL Domain
AVDD_CGPLL	U12	1.1V				AP	N/A	N/A	N/A	
AVSS_APLL	T11	Analog Ground				AG	N/A	N/A	N/A	
AVDD_APLL	U10	1.1V				AP	N/A	N/A	N/A	
AVDD_DPLL	U11	1.1V				AP	N/A	N/A	N/A	
AVSS_DPLL	T12	Analog Ground				AG	N/A	N/A	N/A	
CPU_PWROFF	AB1	CPU_PWROFF				O	8	down	O	PVCC
CORE_PWROFF	AC1	CORE_PWROFF				O	8	down	O	
GPIO6_B[0]	W10	GPIO6_B[0]				I/O	8	down	I	
GPIO6_B[1]	AB2	GPIO6_B[1]				I/O	8	down	I	
GPIO6_B[2]	AA2	GPIO6_B[2]				I/O	8	down	I	
GPIO6_B[3]	AC2	GPIO6_B[3]				I/O	8	down	I	
GPIO6_A[0]	W11	GPIO6_A[0]				I/O	8	up	I	
GPIO6_A[1]	AA3	GPIO6_A[1]				I/O	8	up	I	
GPIO6_A[2]	AB3	GPIO6_A[2]				I/O	8	up	I	
GPIO6_A[3]	AA5	GPIO6_A[3]				I/O	8	up	I	
GPIO6_A[4]	Y5	GPIO6_A[4]				I/O	8	up	I	
GPIO6_A[5]	Y6	GPIO6_A[5]				I/O	8	up	I	
CLK32K	AA4	CLK32K				I	N/A	down	I	
XIN24M	AC11	XIN24M				I	N/A	N/A	I	
XOUT24M	AB11	XOUT24M				O	N/A	N/A	O	
GPIO6_A[6]	Y8	GPIO6_A[6]				I/O	8	up	I	
GPIO6_A[7]	AC4	GPIO6_A[7]				I/O	8	up	I	
NPOR	Y7	NPOR				I	8	N/A	I	

OTG_DVDD	U14	OTG_DVDD				DP	N/A	N/A	N/A	USB Domain
OTG_ID	AB4	OTG_ID				A	N/A	N/A	N/A	
OTG_VBUS	AB5	OTG_VBUS				A	N/A	N/A	N/A	
OTG_VDD33	V14	OTG_VDD33				AP	N/A	N/A	N/A	
OTG_DP	AC16	OTG_DP				A	N/A	N/A	N/A	
OTG_DM	AB16	OTG_DM				A	N/A	N/A	N/A	
OTG_RKELVIN	AA6	OTG_RKELVIN				A	N/A	N/A	N/A	
OTG_VDD25	U15	OTG_VDD25				AP	N/A	N/A	N/A	
HOST_DP	AC17	HOST_DP				A	N/A	N/A	N/A	
HOST_DM	AB17	HOST_DM				A	N/A	N/A	N/A	
HOST_RKELVIN	W13	HOST_RKELVIN				A	N/A	N/A	N/A	
EFUSE_VDDQ	W14	EFUSE_VDDQ				AP	N/A	N/A	N/A	EFUSE Domain
GPIO1_A[4]	AA7	GPIO1_A[4]	uart1_sin	spi0_csn0		I/O	8	up	I	AP1_VCC
GPIO1_A[5]	AC5	GPIO1_A[5]	uart1_sout	spi0_clk		I/O	8	down	I	
GPIO1_A[6]	AB6	GPIO1_A[6]	uart1_cts_n	spi0_rxd		I/O	8	up	I	
GPIO1_A[7]	AB7	GPIO1_A[7]	uart1_rts_n	spi0_txd		I/O	8	up	I	
GPIO4_B[7]	AC7	GPIO4_B[7]	spi0_csn1			I/O	8	up	I	
GPIO3_D[2]	AA8	GPIO3_D[2]	sdmmc1_int_n			I/O	8	up	I	AP0_VCC
GPIO3_C[0]	AB8	GPIO3_C[0]	sdmmc1_cmd			I/O	4	up	I	
GPIO3_C[1]	AC8	GPIO3_C[1]	sdmmc1_data0			I/O	4	up	I	
GPIO3_C[2]	AB9	GPIO3_C[2]	sdmmc1_data1			I/O	4	up	I	
GPIO3_C[3]	AA9	GPIO3_C[3]	sdmmc1_data2			I/O	4	up	I	
GPIO3_C[4]	AA10	GPIO3_C[4]	sdmmc1_data3			I/O	4	up	I	
GPIO3_C[5]	AA12	GPIO3_C[5]	sdmmc1_clkout			I/O	4	down	I	

GPIO3_C[6]	Y10	GPIO3_C[6]	sdmmc1_detect_n			I/O	8	up	I	
GPIO3_C[7]	AA13	GPIO3_C[7]	sdmmc1_write_prt			I/O	8	down	I	
GPIO3_D[0]	Y13	GPIO3_D[0]	sdmmc1_pwr_en			I/O	8	down	I	
GPIO3_D[1]	AA14	GPIO3_D[1]	sdmmc1_backend_pwr			I/O	8	down	I	
GPIO0_C[4]	Y14	GPIO0_C[4]	i2s1_sdi			I/O	8	down	I	
Right Side										
GPIO0_C[0]	AC13	GPIO0_C[0]	i2s1_clk			I/O	4	down	I	AP0_VCC
GPIO0_C[1]	AB13	GPIO0_C[1]	i2s1_sclk			I/O	4	down	I	
GPIO0_C[2]	AC14	GPIO0_C[2]	i2s1_lrck_rx			I/O	4	down	I	
GPIO0_C[3]	AB14	GPIO0_C[3]	i2s1_lrck_tx			I/O	4	down	I	
GPIO0_C[5]	AB15	GPIO0_C[5]	i2s1_sdo			I/O	4	down	I	
GPIO1_A[0]	AB12	GPIO1_A[0]	uart0_sin			I/O	8	up	I	
GPIO1_A[1]	Y11	GPIO1_A[1]	uart0_sout			I/O	8	down	I	
GPIO1_A[2]	AA11	GPIO1_A[2]	uart0_cts_n			I/O	8	up	I	
GPIO1_A[3]	AC10	GPIO1_A[3]	uart0_rts_n			I/O	8	up	I	
GPIO4_C[0]	Y17	GPIO4_C[0]	smc_data0	trace_data0		I/O	4	up	I	SMC_VCC
GPIO4_C[1]	W17	GPIO4_C[1]	smc_data1	trace_data1		I/O	4	up	I	
GPIO4_C[2]	W16	GPIO4_C[2]	smc_data2	trace_data2		I/O	4	up	I	
GPIO4_C[3]	T19	GPIO4_C[3]	smc_data3	trace_data3		I/O	4	up	I	
GPIO4_C[4]	V17	GPIO4_C[4]	smc_data4	trace_data4		I/O	4	up	I	
GPIO4_C[5]	U19	GPIO4_C[5]	smc_data5	trace_data5		I/O	4	up	I	
GPIO4_C[6]	V19	GPIO4_C[6]	smc_data6	trace_data6		I/O	4	down	I	
GPIO4_C[7]	W18	GPIO4_C[7]	smc_data7	trace_data7		I/O	4	down	I	
GPIO4_D[0]	Y18	GPIO4_D[0]	smc_data8	trace_data8		I/O	4	down	I	
GPIO4_D[1]	AA18	GPIO4_D[1]	smc_data9	trace_data9		I/O	4	down	I	

GPIO4_D[2]	W20	GPIO4_D[2]	smc_data10	trace_data10		I/O	4	down	I	
GPIO4_D[3]	AB18	GPIO4_D[3]	smc_data11	trace_data11		I/O	4	down	I	
GPIO4_D[4]	U18	GPIO4_D[4]	smc_data12	trace_data12		I/O	4	down	I	
GPIO4_D[5]	AC19	GPIO4_D[5]	smc_data13	trace_data13		I/O	4	down	I	
GPIO4_D[6]	Y20	GPIO4_D[6]	smc_data14	trace_data14		I/O	4	down	I	
GPIO4_D[7]	AA20	GPIO4_D[7]	smc_data15	trace_data15		I/O	4	down	I	
GPIO0_C[7]	AB20	GPIO0_C[7]	trace_ctl	smc_addr3		I/O	4	down	I	
GPIO0_C[6]	T18	GPIO0_C[6]	trace_clk	smc_addr2		I/O	4	down	I	
GPIO0_D[5]	AC20	GPIO0_D[5]	i2s2_sdo	smc_addr1		I/O	4	down	I	
GPIO0_D[4]	V21	GPIO0_D[4]	i2s2_sdi	smc_addr0		I/O	4	down	I	
GPIO0_D[3]	W21	GPIO0_D[3]	i2s2_lrck_tx	smc_adv_n		I/O	4	up	I	
GPIO0_D[2]	Y21	GPIO0_D[2]	i2s2_lrck_rx	smc_oe_n		I/O	4	up	I	
GPIO0_D[0]	P19	GPIO0_D[0]	i2s2_clk	smc_csn0		I/O	4	up	I	
GPIO0_D[1]	P20	GPIO0_D[1]	i2s2_sclk	smc_we_n		I/O	4	up	I	
FLASH_DATA[0]	P18	FLASH_DATA[0]	emmc_data0			I/O	8	down	I	FLASH_VCC
FLASH_DATA[1]	AA21	FLASH_DATA[1]	emmc_data1			I/O	8	down	I	
FLASH_DATA[2]	AB21	FLASH_DATA[2]	emmc_data2			I/O	8	down	I	
FLASH_DATA[3]	W22	FLASH_DATA[3]	emmc_data3			I/O	8	down	I	
FLASH_DATA[4]	Y22	FLASH_DATA[4]	emmc_data4			I/O	8	down	I	
FLASH_DATA[5]	V20	FLASH_DATA[5]	emmc_data5			I/O	8	down	I	
FLASH_DATA[6]	AA22	FLASH_DATA[6]	emmc_data6			I/O	8	down	I	
FLASH_DATA[7]	AB22	FLASH_DATA[7]	emmc_data7			I/O	8	down	I	
FLASH_RDY	U20	FLASH_RDY				I/O	8	up	I	
FLASH_ALE	T20	FLASH_ALE				O	4	down	O	
FLASH_CLE	AC22	FLASH_CLE				O	4	down	O	

FLASH_RDN	N20	FLASH_RDN				O	8	up	O	
FLASH_WRN	AC23	FLASH_WRN				O	8	up	O	
FLASH_WP	AB23	FLASH_WP	emmc_pwr_en			O	4	down	O	
FLASH_CSN0	Y23	FLASH_CSN0				O	4	up	O	
GPIO4_B[0]	N18	GPIO4_B[0]	flash_csn1			I/O	4	up	I	
GPIO4_B[1]	U21	GPIO4_B[1]	flash_csn2	emmc_cmd		I/O	4	up	I	
GPIO4_B[2]	P21	GPIO4_B[2]	flash_csn3	emmc_rstn_out		I/O	4	up	I	
GPIO3_D[7]	N21	GPIO3_D[7]	flash_dqs	emmc_clkout		I/O	8	up	I	
GPIO3_B[6]	L20	GPIO3_B[6]	sdmmc0_detect_n			I/O	8	up	I	VCCIO0 VCCIO1
GPIO3_B[7]	N19	GPIO3_B[7]	sdmmc0_write_prt			I/O	8	down	I	
GPIO3_A[6]	W23	GPIO3_A[6]	sdmmc0_rstn_out			I/O	8	up	I	
GPIO3_A[7]	V22	GPIO3_A[7]	sdmmc0_pwr_en			I/O	8	down	I	
GPIO3_B[0]	U22	GPIO3_B[0]	sdmmc0_clkout			I/O	4	down	I	
GPIO3_B[1]	P22	GPIO3_B[1]	sdmmc0_cmd			I/O	4	up	I	
GPIO3_B[2]	M21	GPIO3_B[2]	sdmmc0_data0			I/O	4	up	I	
GPIO3_B[3]	U23	GPIO3_B[3]	sdmmc0_data1			I/O	4	up	I	
GPIO3_B[4]	T21	GPIO3_B[4]	sdmmc0_data2			I/O	4	up	I	
GPIO3_B[5]	K20	GPIO3_B[5]	sdmmc0_data3			I/O	4	up	I	
GPIO0_A[7]	L21	GPIO0_A[7]	i2s0_sdi			I/O	8	down	I	
GPIO0_B[0]	T22	GPIO0_B[0]	i2s0_clk			I/O	4	down	I	
GPIO0_B[1]	T23	GPIO0_B[1]	i2s0_sclk			I/O	4	down	I	
GPIO0_B[2]	R22	GPIO0_B[2]	i2s0_lrck_rx			I/O	4	down	I	
GPIO0_B[3]	R21	GPIO0_B[3]	i2s0_lrck_tx			I/O	4	down	I	
GPIO0_B[4]	K21	GPIO0_B[4]	i2s0_sdo0			I/O	4	down	I	
GPIO0_B[5]	P23	GPIO0_B[5]	i2s0_sdo1			I/O	4	down	I	

GPIO0_B[6]	N23	GPIO0_B[6]	i2s0_sdo2			I/O	4	up	I	
GPIO0_B[7]	M22	GPIO0_B[7]	i2s0_sdo3			I/O	4	up	I	
GPIO1_B[2]	L22	GPIO1_B[2]	spdif_tx			I/O	4	down	I	
GPIO1_B[0]	N22	GPIO1_B[0]	uart2_sin			I/O	8	up	I	
GPIO1_B[1]	AB19	GPIO1_B[1]	uart2_sout			I/O	8	down	I	
GPIO2_D[6]	L23	GPIO2_D[6]	i2c1_sda			I/O	8	up	I	
GPIO2_D[7]	K22	GPIO2_D[7]	i2c1_scl			I/O	8	up	I	
GPIO3_A[0]	K23	GPIO3_A[0]	i2c2_sda			I/O	8	up	I	
GPIO3_A[1]	Y19	GPIO3_A[1]	i2c2_scl			I/O	8	up	I	
GPIO3_D[3]	L19	GPIO3_D[3]	uart3_sin			I/O	8	up	I	
GPIO3_D[4]	K19	GPIO3_D[4]	uart3_sout			I/O	8	down	I	
GPIO3_D[5]	K18	GPIO3_D[5]	uart3_cts_n			I/O	8	up	I	
GPIO3_D[6]	L18	GPIO3_D[6]	uart3_rts_n			I/O	8	up	I	
GPIO0_A[0]	AA19	GPIO0_A[0]	hdmi_hot_plug_in			I/O	8	down	I	
GPIO0_A[1]	Y16	GPIO0_A[1]	hdmi_i2c_scl			I/O	8	up	I	
GPIO0_A[2]	H19	GPIO0_A[2]	hdmi_i2c_sda			I/O	8	up	I	
GPIO0_A[3]	AA15	GPIO0_A[3]	pwm0			I/O	8	down	I	
GPIO0_A[4]	H21	GPIO0_A[4]	pwm1			I/O	8	down	I	
GPIO0_A[5]	J22	GPIO0_A[5]	otg_drv_vbus			I/O	8	down	I	
GPIO0_A[6]	H18	GPIO0_A[6]	host_drv_vbus			I/O	8	down	I	
GPIO0_D[6]	H20	GPIO0_D[6]	pwm2			I/O	8	down	I	
GPIO0_D[7]	J21	GPIO0_D[7]	pwm3			I/O	8	down	I	
TDO	H22	TDO				O	8	N/A	O	
TCK	H23	TCK				I	8	up	I	
TRST_N	G20	TRST_N				I	8	down	I	

TDI	G22	TDI				I	8	up	I	
TMS	G21	TMS				I/O	8	up	I	
GPIO2_D[4]	G19	GPIO2_D[4]	i2c0_sda			I/O	8	up	I	
GPIO2_D[5]	G18	GPIO2_D[5]	i2c0_scl			I/O	8	up	I	
GPIO6_B[4]	G23	GPIO6_B[4]				I/O	8	up	I	
Top Side										
DQ[7]	F20	DQ[7]				I/O	N/A	N/A	I	
DQ[6]	F22	DQ[6]				I/O	N/A	N/A	I	
DQ[5]	F21	DQ[5]				I/O	N/A	N/A	I	
DQ[4]	F19	DQ[4]				I/O	N/A	N/A	I	
DQS[0]	E22	DQS[0]				I/O	N/A	N/A	I	
DQS_B[0]	E23	DQS_B[0]				I/O	N/A	N/A	I	
DQ[3]	E18	DQ[3]				I/O	N/A	N/A	I	
DQ[2]	E20	DQ[2]				I/O	N/A	N/A	I	
DQ[1]	E21	DQ[1]				I/O	N/A	N/A	I	
DQ[0]	E17	DQ[0]				I/O	N/A	N/A	I	MVDD
DM[0]	D23	DM[0]				I/O	N/A	N/A	I	
VREF	F13	VREF				P	N/A	N/A	N/A	
DQ[23]	D22	DQ[23]				I/O	N/A	N/A	I	
DQ[22]	D21	DQ[22]				I/O	N/A	N/A	I	
DQ[21]	B23	DQ[21]				I/O	N/A	N/A	I	
DQ[20]	A23	DQ[20]				I/O	N/A	N/A	I	
DQS[2]	B22	DQS[2]				I/O	N/A	N/A	I	
DQS_B[2]	A22	DQS_B[2]				I/O	N/A	N/A	I	
DQ[19]	D17	DQ[19]				I/O	N/A	N/A	I	

DQ[18]	E16	DQ[18]				I/O	N/A	N/A	I	
DQ[17]	C22	DQ[17]				I/O	N/A	N/A	I	
DQ[16]	B21	DQ[16]				I/O	N/A	N/A	I	
DM[2]	C21	DM[2]				I/O	N/A	N/A	I	
ZQ_PIN	G15	ZQ_PIN				I/O	N/A	N/A	I	
ODT[1]	C20	ODT[1]				O	N/A	N/A	O	
ODT[0]	B20	ODT[0]				O	N/A	N/A	O	
A[14]	D16	A[14]				O	N/A	N/A	O	
A[13]	C19	A[13]				O	N/A	N/A	O	
A[12]	D19	A[12]				O	N/A	N/A	O	
A[11]	A20	A[11]				O	N/A	N/A	O	
A[10]	D18	A[10]				O	N/A	N/A	O	
A[9]	C18	A[9]				O	N/A	N/A	O	
A[8]	B19	A[8]				O	N/A	N/A	O	
A[7]	A19	A[7]				O	N/A	N/A	O	
A[6]	B18	A[6]				O	N/A	N/A	O	
A[5]	C17	A[5]				O	N/A	N/A	O	
CK	B16	CK				O	N/A	N/A	O	
CK_B	A16	CK_B				O	N/A	N/A	O	
A[4]	B17	A[4]				O	N/A	N/A	O	
A[3]	C15	A[3]				O	N/A	N/A	O	
A[2]	E14	A[2]				O	N/A	N/A	O	
A[1]	A17	A[1]				O	N/A	N/A	O	
A[0]	B15	A[0]				O	N/A	N/A	O	
BA[2]	C14	BA[2]				O	N/A	N/A	O	

BA[1]	B14	BA[1]				O	N/A	N/A	O	
BA[0]	A14	BA[0]				O	N/A	N/A	O	
RAS_B	D14	RAS_B				O	N/A	N/A	O	
CAS_B	D13	CAS_B				O	N/A	N/A	O	
WE_B	C13	WE_B				O	N/A	N/A	O	
CS_B[1]	E13	CS_B[1]				O	N/A	N/A	O	
CS_B[0]	B13	CS_B[0]				O	N/A	N/A	O	
CKE1	D11	CKE1				O	N/A	N/A	O	
CKE0	A13	CKE0				O	N/A	N/A	O	
RESET	C12	RESET				O	N/A	N/A	O	
DQ[15]	B12	DQ[15]				I/O	N/A	N/A	I	
DQ[14]	C11	DQ[14]				I/O	N/A	N/A	I	
DQ[13]	E11	DQ[13]				I/O	N/A	N/A	I	
DQ[12]	D10	DQ[12]				I/O	N/A	N/A	I	
DQS[1]	B11	DQS[1]				I/O	N/A	N/A	I	
DQS_B[1]	A11	DQS_B[1]				I/O	N/A	N/A	I	
DQ[11]	B10	DQ[11]				I/O	N/A	N/A	I	
DQ[10]	A10	DQ[10]				I/O	N/A	N/A	I	
DQ[9]	C10	DQ[9]				I/O	N/A	N/A	I	
DQ[8]	B9	DQ[8]				I/O	N/A	N/A	I	
DM[1]	C9	DM[1]				I/O	N/A	N/A	I	
DQ[31]	C8	DQ[31]				I/O	N/A	N/A	I	
DQ[30]	E10	DQ[30]				I/O	N/A	N/A	I	
DQ[29]	D8	DQ[29]				I/O	N/A	N/A	I	
DQ[28]	E8	DQ[28]				I/O	N/A	N/A	I	

DQS[3]	B8	DQS[3]				I/O	N/A	N/A	I	
DQS_B[3]	A8	DQS_B[3]				I/O	N/A	N/A	I	
DQ[27]	A7	DQ[27]				I/O	N/A	N/A	I	
DQ[26]	B7	DQ[26]				I/O	N/A	N/A	I	
DQ[25]	D7	DQ[25]				I/O	N/A	N/A	I	
DQ[24]	C7	DQ[24]				I/O	N/A	N/A	I	
DM[3]	E7	DM[3]				I/O	N/A	N/A	I	
HDMI_VDDLA	F7	1.1V				AP	N/A	N/A	N/A	HDMI Domain
HDMI_PVDD	F8	2.5V				AP	N/A	N/A	N/A	
HDMI_REXT	C6	HDMI_REXT				A	N/A	N/A	N/A	
HDMI_TXC_N	B5	HDMI_TXC_N				A	N/A	N/A	N/A	
HDMI_TXC	A5	HDMI_TXC				A	N/A	N/A	N/A	
HDMI_AVDD25	F8	2.5V				AP	N/A	N/A	N/A	
HDMI_TX0_N	B4	HDMI_TX0_N				A	N/A	N/A	N/A	
HDMI_TX0	A4	HDMI_TX0				A	N/A	N/A	N/A	
HDMI_TX1_N	B2	HDMI_TX1_N				A	N/A	N/A	N/A	
HDMI_TX1	A2	HDMI_TX1				A	N/A	N/A	N/A	
HDMI_TX2_N	B1	HDMI_TX2_N				A	N/A	N/A	N/A	
HDMI_TX2	A1	HDMI_TX2				A	N/A	N/A	N/A	
LCDC0_HSYNC	D6	LCDC0_HSYNC				I/O	4	N/A	I	LCD0_VCC0
LCDC0_DCLK	E6	LCDC0_DCLK				I/O	12	N/A	I	
LCDC0_VSYNC	D5	LCDC0_VSYNC				I/O	4	N/A	I	
LCDC0_DEN	D4	LCDC0_DEN				I/O	4	N/A	I	
LCDC0_DATA[0]	C4	LCDC0_DATA[0]				I/O	8	N/A	I	
LCDC0_DATA[1]	C3	LCDC0_DATA[1]				I/O	8	N/A	I	

LCDC0_DATA[2]	C2	LCDC0_DATA[2]				I/O	8	N/A	I	
LCDC0_DATA[3]	D3	LCDC0_DATA[3]				I/O	8	N/A	I	
LCDC0_DATA[4]	D1	LCDC0_DATA[4]				I/O	8	N/A	I	
LCDC0_DATA[5]	F5	LCDC0_DATA[5]				I/O	8	N/A	I	
LCDC0_DATA[6]	D2	LCDC0_DATA[6]				I/O	8	N/A	I	

Notes :

①: *Pad types : I = input , O = output , I/O = input/output (bidirectional) ,*
AP = Analog Power , AG = Analog Ground
DP = Digital Power , DG = Digital Ground
A = Analog

②: *Output Drive Unit is mA , only Digital IO have drive value*

③: *Reset state : I = input without any pull resistor , O = output without any pull resistor ,*

④: *It is die location. For examples, "Left side" means that all the related IOs are always in left side of die*

⑤: *Power supply means that all the related IOs is in these IO power domain. If multiple powers is included, they are connected together in one IO power ring*

1.3.3 IO pin name descriptions

This sub-chapter will focus on the detailed function description of every pins based on different interface.

Table 1-3 RK30xx IO function description list

Interface	Pin Name	Direction	Description
Misc	XIN24M	I	Clock input of 24MHz crystal
	XOUT24M	O	Clock output of 24MHz crystal
	CLK32K	I	Clock input of 32.768KHz
	CPU_PWROFF	O	Request signal to external PMIC for power down CPU subsystem with dual-core Cortex-A9
	CORE_PWROFF	O	Request signal to external PMIC for SoC Core logic w/o Cortex-A9 subsystem and PMU logic
	BTMODE	I	Chip boot device select (BootRom or Nor Flash)
	NPOR	I	Power on reset for chip

Interface	Pin Name	Direction	Description
Debug	TRST_N	I	JTAG interface reset input
	TCK	I	JTAG interface clock input/SWD interface clock input
	TDI	I	JTAG interface TDI input
	TMS	I/O	JTAG interface TMS input/SWD interface data out
	TDO	O	JTAG interface TDO output

Interface	Pin Name	Direction	Description
ETM Trace	trace_clk	O	Cortex-A8 ETM trace port clk
	trace_ctl	O	Cortex-A8 ETM trace port control
	trace_data*i*(*i*=0~15)	O	Cortex-A8 ETM trace port data

Interface	Pin Name	Direction	Description
SD/MMC Host Controller	sdmmc_clkout	O	sdmmc card clock.
	sdmmc_cmd	I/O	sdmmc card command output and reponse input.
	sdmmc_data*i* (*i*=0~3)	I/O	sdmmc card data input and output.
	sdmmc_detect_n	I	sdmmc card detect signal, a 0 represents presence of card.
	sdmmc_write_prt	I	sdmmc card write protect signal, a 1 represents write is protected.
	sdmmc_rstn_out	O	sdmmc card reset signal
	sdmmc_pwr_en	O	sdmmc card power-enable control signal

Interface	Pin Name	Direction	Description
SDIO Host Controller	sdio_clkout	O	sdio card clock.
	sdio_cmd	I/O	sdio card command output and reponse input.
	sdio_data*i* (*i*=0~3)	I/O	sdio card data input and output.
	sdio_detect_n	I	sdio card detect signal, a 0 represents presence of card.
	sdio_write_prt	I	sdio card write protect signal, a 1 represents write is protected.
	sdio_pwr_en	O	sdio card power-enable control signal
	sdio_int_n	O	sdio card interrupt indication
	sdio_backend	O	the back-end power supply for embedded device

Interface	Pin Name	Direction	Description
eMMC Interface	emmc_clkout	O	emmc card clock.
	emmc_cmd	I/O	emmc card command output and reponse input.
	emmc_data*i* (*i*=0~7)	I/O	emmc card data input and output.
	emmc_pwr_en	O	emmc card power-enable control signal
	emmc_rstn_out	O	emmc card reset signal

Interface	Pin Name	Direction	Description
DMC	CK	O	Active-high clock signal to the memory device.
	CK_B	O	Active-low clock signal to the memory device.
	CKE*i* (*i*=0,1)	O	Active-high clock enable signal to the memory device for two chip select.
	CS_B*i* (*i*=0,1)	O	Active-low chip select signal to the memory device. AThere are two chip select.
	RAS_B	O	Active-low row address strobe to the memory device.
	CAS_B	O	Active-low column address strobe to the memory device.
	WE_B	O	Active-low write enable strobe to the memory device.
	BA[2:0]	O	Bank address signal to the memory device.
	A[15:0]	O	Address signal to the memory device.
	DQ[31:0]	I/O	Bidirectional data line to the memory device.
	DQS[3:0]	I/O	Active-high bidirectional data strobes to the memory device.

	DQS_B[3:0]	I/O	Active-low bidirectional data strobes to the memory device.
	DM[3:0]	O	Active-low data mask signal to the memory device.
	ODT*i* (*i*=0,1)	O	On-Die Termination output signal for two chip select.
	RET_EN	I	Active-low retention latch enable input
	RESET	O	DDR3 reset signal to the memory device
	VREF*i* (*i*=0,1,2,3)	I/O	Reference Voltage input for three regions of DDR IO
	ZQ_PIN	I/O	ZQ calibration pad which connects 240ohm±1% resistor

Interface	Pin Name	Direction	Description
SMC	smc_oe_n	O	SMC output enable signal.
	smc_bls_n*i* (*i*=0,1)	O	SMC byte lane strobe signal for two bytes.
	smc_we_n	O	SMC write enable signal.
	smc_csn*i* (*i*=0,1)	O	SMC chip enable signal.
	smc_adv_n	O	SMC address valid signal in shared mode
	smc_addr*i* (*i*=0~19)	O	SMC address signal.
	smc_data*i* (*i*=0~15)	I/O	SMC directional data line to memory device.

Interface	Pin Name	Direction	Description
NandC	FLASH_WP	O	Flash write-protected signal
	FLASH_ALE	O	Flash address latch enable signal
	FLASH_CLE	O	Flash command latch enable signal
	FLASH_WRN	O	Flash write enable and clock signal
	FLASH_RDN	O	Flash read enable and write/read signal
	FLASH_DATA[*i*](*i*=0~7)	I/O	Low 8bits of flash data inputs/outputs signal
	flash_data*i*(*i*=8~15)	I/O	High 8bits of flash data inputs/outputs signal
	flash_dqs	I/O	Flash data strobe signal
	FLASH_RDY	I	Flash ready/busy signal
	FLASH0_CSN	O	Flash chip enable signal for chip 0
	flash_csn*i*(*i*=1~7)	O	Flash chip enable signal for chip i, i=1~7

Interface	Pin Name	Direction	Description
HSADC Interface	hsadc_clkout	O	hsadc/tsi/gps reference clock
	hsadc_data*i* (*i*=0~9)	I	hsadc(i=0~9)/tsi(i=0~7)/gps data(i=0,1)
	ts_sync	I	ts synchronizer signal

Interface	Pin Name	Direction	Description
I2S/PCM0 Controller (8 channel)	i2s0_clk	O	I2S/PCM0 clock source
	i2s0_sclk	I/O	I2S/PCM0 serial clock
	i2s0_lrck_rx	I/O	I2S/PCM0 left & right channel signal for receiving serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode
	i2s0_sdi	I	I2S/PCM0 serial data input
	i2s0_sdo*i* *(i=0,1,2,3)*	O	I2S/PCM0 serial data ouput
	i2s0_lrck_tx	I/O	I2S/PCM0 left & right channel signal for transmitting serial data, synchronous left & right channel in I2S mode *(i=0)* and the beginning of a group of left & right channels in PCM mode *(i=0,1)*

Interface	Pin Name	Direction	Description
I2S/PCM1 Controller (2 channel)	i2s1_clk	O	I2S/PCM1 clock source
	i2s1_sclk	I/O	I2S/PCM1 serial clock
	i2s1_lrck_rx	I/O	I2S/PCM1 left & right channel signal for receiving serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode
	i2s1_sdi	I	I2S/PCM1 serial data input
	i2s1_sdo	O	I2S/PCM1 serial data ouput
	i2s1_lrck_tx	I/O	I2S/PCM1 left & right channel signal for transmitting serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode

Interface	Pin Name	Direction	Description
I2S/PCM2 Controller (2 channel)	i2s2_clk	O	I2S/PCM2 clock source
	i2s2_sclk	I/O	I2S/PCM2 serial clock
	i2s2_lrck_rx	I/O	I2S/PCM2 left & right channel signal for receiving serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode
	i2s2_sdi	I	I2S/PCM2 serial data input
	i2s2_sdo	O	I2S/PCM2 serial data ouput
	i2s2_lrck_tx	I/O	I2S/PCM2 left & right channel signal for transmitting serial data, synchronous left & right channel in I2S mode and the beginning of a group of left & right channels in PCM mode

Interface	Pin Name	Direction	Description
SPDIF transmitter	spdif_tx	O	spdif biphase data ouput

Interface	Pin Name	Direction	Description
SPI	spi*x*_clk*(x=0,1)*	I/O	spi serial clock

Controller	spi*x*_csn*y* *(x=0,1)(y=0,1)*	I/O	spi chip select signal,low active
	spi*x*_txd*(x=0,1)*	O	spi serial data output
	spi*x*_rxd*(x=0,1)*	I	spi serial data input

Interface	Pin Name	Direction	Description
LCDC0	LCDC0_DCLK	O	LCDC0 RGB interface display clock out, MCU i80 interface RS signal
	LCDC0_VSYNC	O	LCDC0 RGB interface vertival sync pulse, MCU i80 interface CSN signal
	LCDC0_HSYNC	O	LCDC0 RGB interface horizontial sync pulse, MCU i80 interface WEN signal
	LCDC0_DEN	O	LCDC0 RGB interface data enable, MCU i80 interface REN signal
	LCDC0_DATA[23:0]	I/O	LCDC0 data output/input

Interface	Pin Name	Direction	Description
LCDC1	lcdc1_dclk	O	LCDC1 RGB interface display clock out, MCU i80 interface RS signal
	lcdc1_vsync	O	LCDC1 RGB interface vertival sync pulse, MCU i80 interface CSN signal
	lcdc1_hsync	O	LCDC1 RGB interface horizontial sync pulse, MCU i80 interface WEN signal
	lcdc1_den	O	LCDC1 RGB interface data enable, MCU i80 interface REN signal
	lcdc1_data[23:0]	I/O	LCDC1 data output/input

Interface	Pin Name	Direction	Description
Camera IF0	CIF0_CLKIN	I	Camera0 interface input pixel clock
	cif0_clkout	O	Camera0 interface output work clock
	CIF0_VSYNC	I	Camera0 interface vertical sync signal
	CIF0_HREF	I	Camera0 interface horizontial sync signal
	cif0_data[1:0]	I	Camera0 interface low 2-bit input pixel data
	CIF0_DATAIN[9:2]	I	Camera0 interface middle 8-bit input pixel data
	cif0_data[11:10]	I	Camera0 interface high 2-bit input pixel data

Interface	Pin Name	Direction	Description
Camera IF1	cif1_clkin	I	Camera1 interface input pixel clock
	cif1_clkout	O	Camera1 interface output work clock
	cif1_vsync	I	Camera1 interface vertical sync signal

	cif1_href	I	Camera1 interface horizontial sync signal
	cif1_data[11:0]	I	Camera1 interface 12-bit input pixel data

Interface	Pin Name	Direction	Description
RMII	rmii_clkout	O	RMII REC_CLK output
	rmii_clkin	I	RMII REF_CLK input
	rmii_tx_en	O	rmii transfer enable
	rmii_txd1	O	rmii transfer data
	rmii_txd0	O	rmii transfer data
	rmii_rx_err	I	rmii receive error
	rmii_crs_dvalid	I	rmii carrier sense / receive data valid input
	rmii_rxd1	I	rmii receive data
	rmii_rxd0	I	rmii receive data
	mii_md	I/O	mii management interface data
	mii_mdclk	O	mii management interface clock

Interface	Pin Name	Direction	Description
PWM	pwm3	O	Pulse Width Modulation output
	pwm2	O	Pulse Width Modulation output
	pwm1	O	Pulse Width Modulation output
	pwm0	O	Pulse Width Modulation output

Interface	Pin Name	Direction	Description
I2C	i2c0_sda	I/O	I2C0 data
	i2c0_scl	I/O	I2C0 clock
	i2c1_sda	I/O	I2C1 data
	i2c1_scl	I/O	I2C1 clock
	i2c2_sda	I/O	I2C2 data
	i2c2_scl	I/O	I2C2 clock
	i2c3_sda	I/O	I2C3 data
	i2c3_scl	I/O	I2C3 clock
	i2c4_sda	I/O	I2C4 data
	i2c4_scl	I/O	I2C4 clock

Interface	Pin Name	Direction	Description
UART	uart0_sin	I	UART0 searial data input
	uart0_sout	O	UART0 searial data output
	uart0_cts_n	I	UART0 clear to send
	uart0_rts_n	O	UART0 request to send
	uart1_sin	I	UART1 searial data input

	uart1_sout	O	UART1 searial data output
	uart1_cts_n	O	UART1 clear to send
	uart1_rts_n	I	UART1 request to send
	uart2_sin	I	UART2 searial data input
	uart2_sout	O	UART2 searial data output
	uart3_sin	I	UART3 searial data input
	uart3_sout	O	UART3 searial data output
	uart3_cts_n	I	UART3 clear to send
	uart3_rts_n	O	UART3 request to send

Interface	Pin Name	Direction	Description
USB OTG 2.0	OTG_DM	N/A	USB OTG 2.0 Data signal DM
	OTG_RKELVIN	N/A	USB OTG 2.0 Transmitter Kelvin Connection to Resistor Tune Pin
	OTG_DP	N/A	USB OTG 2.0 Data signal DP
	OTG_VBUS	N/A	USB OTG 2.0 5-V power supply pin
	otg_drv_vbus	O	USB OTG 2.0 drive VBUS

Interface	Pin Name	Direction	Description
USB Host 2.0	HOST_DM	N/A	USB HOST 2.0 Data signal DM
	HOST_RKELVIN	N/A	USB HOST 2.0 Transmitter Kelvin Connection to Resistor Tune Pin
	HOST_DP	N/A	USB HOST 2.0 Data signal DP
	HOST_VBUS	N/A	USB HOST 2.0 5-V power supply pin
	host_drv_vbus	O	USB HOST 2.0 drive VBUS

Interface	Pin Name	Direction	Description
SAR-ADC	SARADC_AIN[i] (i=0~3)	N/A	SAR-ADC input signal for 4 channel

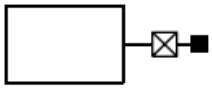
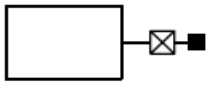
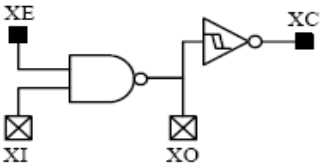
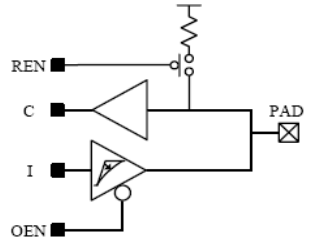
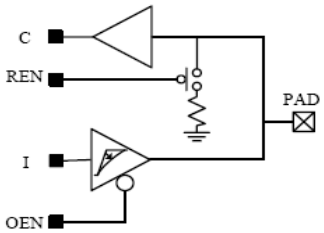
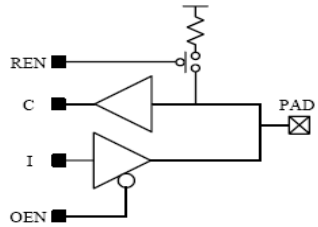
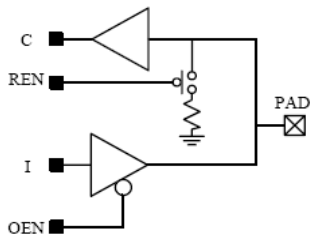
Interface	Pin Name	Direction	Description
eFuse	EFUSE_VDDQ	N/A	eFuse program and sense power

1.3.4 RK30xx IO Type

The following list shows IO type except DDR IO and all of Power/Ground IO .

Table 1-4 RK30xx IO Type List

Type	Diagram	Description	Pin Name

A		Analog IO Cell with IO voltage	EFUSE_VDDQ
B		Dedicated Power supply to Internal Macro with IO voltage	SARADC_AIN[3:0]
C		Crystal Oscillator with high enable	XIN24M/XOUT24M
D		Tri-state output pad with input, limited slew rate and enable controlled pull-up	Part of digital GPIO
E		Tri-state output pad with input, limited slew rate and enable controlled pull-down	Part of digital GPIO
F		Tri-state output pad with input, and enable controlled pull-up	Part of digital GPIO
G		Tri-state output pad with input, and enable controlled pull-down	Part of digital GPIO

1.4 Package information

RK3066 package is TFBGA453LD
(body: 19mm x 19mm ; ball size : 0.4mm ; ball pitch : 0.8mm)

1.4.1 Dimension

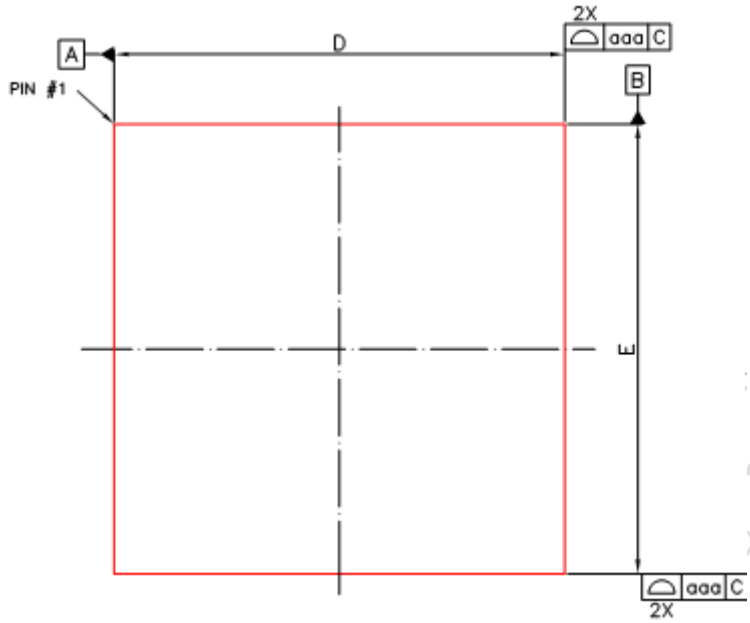


Fig. 1-2 RK3066 TFBGA453 Package Top View

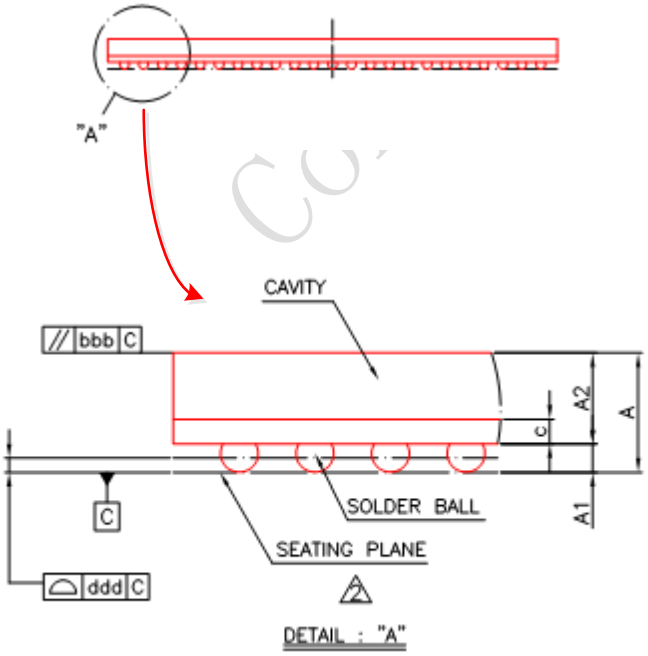


Fig. 1-3 RK3066 TFBGA453 Package Side View

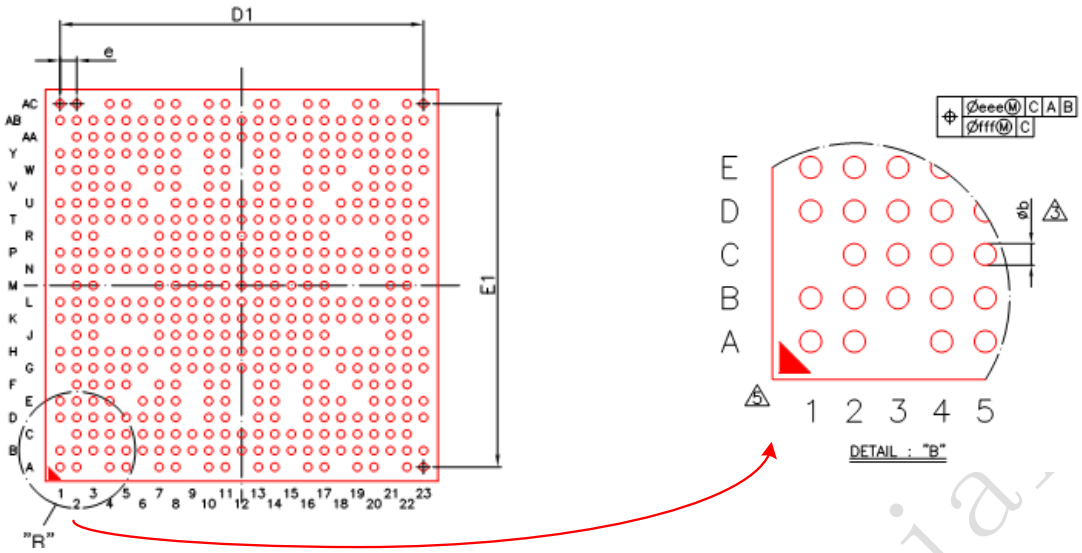


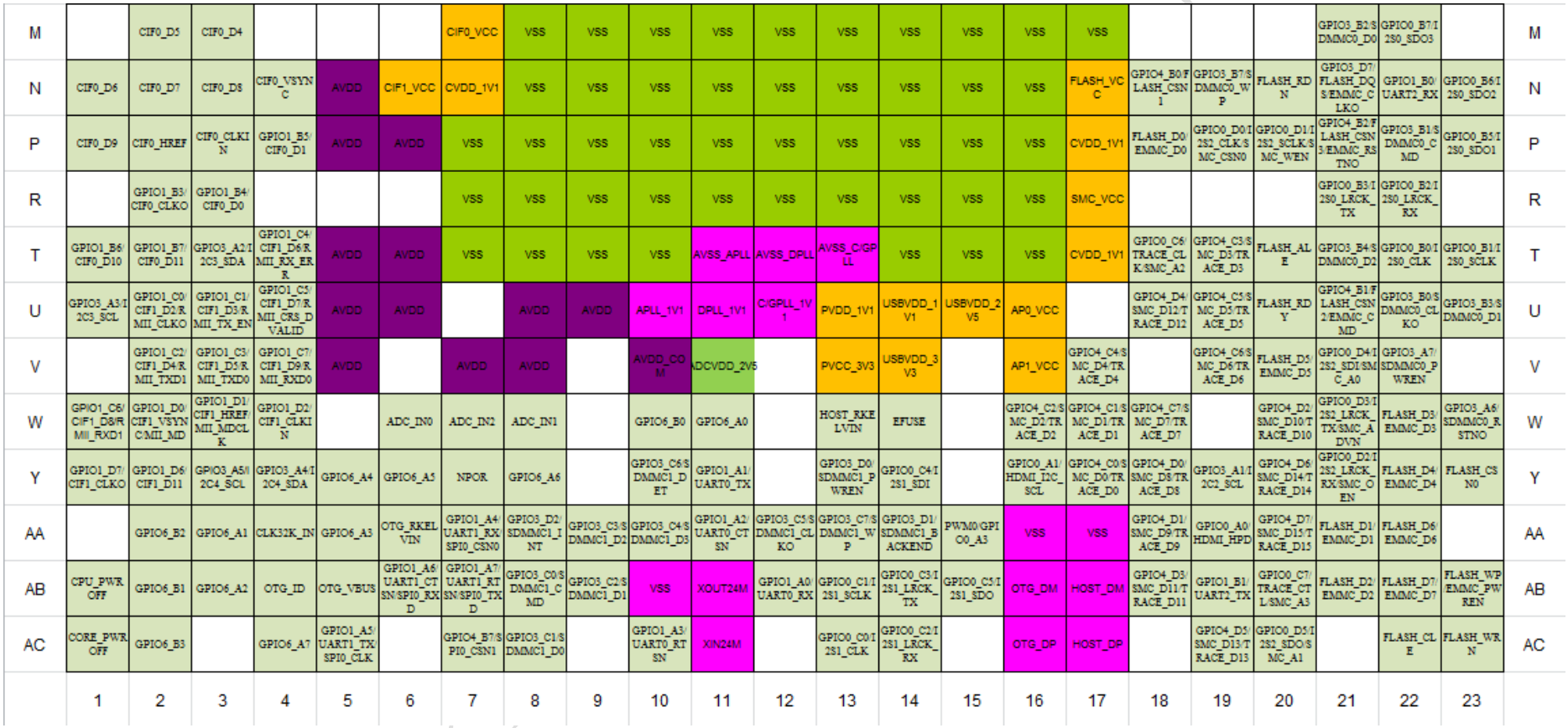
Fig. 1-4RK3066 TFBGA453 Package Bottom View

Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	---	---	1.40	---	---	0.055
A1	0.25	0.30	0.35	0.010	0.012	0.014
A2	0.91	0.96	1.01	0.036	0.038	0.040
b	0.35	0.40	0.45	0.014	0.016	0.018
c	0.22	0.26	0.30	0.009	0.010	0.012
D	18.90	19.00	19.10	0.744	0.748	0.752
E	18.90	19.00	19.10	0.744	0.748	0.752
D1	---	17.60	---	---	0.693	---
E1	---	17.60	---	---	0.693	---
e	---	0.80	---	---	0.031	---
aaa	0.15			0.006		
bbb	0.20			0.008		
ddd	0.15			0.006		
eee	0.15			0.006		
fff	0.08			0.003		
MD/ME	23/23			23/23		

Fig. 1-5 RK3066 TFBGA453 Package Dimension

1.4.2 Ball Map

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
A	HDMI_TX2	HDMI_TX1		HDMI_TX0	HDMI_TXC		MDQ27	MDQS_B3		MDQ10	MDQS_B1		MCKE0	MBA0		MCK_N	MA1		MA7	MA11		MDQS_B2	MDQ20	A
B	HDMI_TX2N	HDMI_TX1N	HDMI_AVSS	HDMI_TX0N	HDMI_TXCN	HDMI_AVSS	MDQ26	MDQS_3	MDQ8	MDQ11	MDQS_1	MDQ15	MCSN0	MBA1	MA0	MCK	MA4	MA6	MA8	MODT0	MDQ16	MDQS_2	MDQ21	B
C		LCD0_D2	LCD0_D1	LCD0_D0	HDMI_AVSS	HDMI_REXT	MDQ24	MDQ31	MDM1	MDQ9	MDQ14	MRESET	MWEN	MBA2	MA3	VSS	MA5	MA9	MA13	MODT1	MDM2	MDQ17		C
D	LCD0_D4	LCD0_D6	LCD0_D3	LCD0_DEN	LCD0_VSYNC	LCD0_HSYNC	MDQ25	MDQ29		MDQ12	MCKE1		MCASN	MRASN		MA14	MDQ19	MA10	MA12	VSS	MDQ22	MDQ23	MDM0	D
E	LCD0_D12	LCD0_D10	LCD0_D9	LCD0_D7		LCD0_DCLK	MDM3	MDQ28		MDQ30	MDQ13		MCSN1	MA2		MDQ18	MDQ0	MDQ3		MDQ2	MDQ1	MDQS_0	MDQS_B0	E
F		LCD0_D16	LCD0_D11	LCD0_D8	LCD0_D5		HDMIVDD_1V	DMIAVDD_2V5		MVDD	MVDD		MVREF	MVDD		MVDD	MVDD		MDQ4	MDQ7	MDQ5	MDQ6		F
G	LCD0_D19	LCD0_D18	LCD0_D17	LCD0_D14	LCD0_D13	LCD0_D15		HDMI_AVSS	HDMI_AVSS	MVDD	CVDD_1V1	VSS	MVDD	MVDD	MPZQ	CVDD_1V1		GPIO2_D5/I2C0_SCL	GPIO2_D4/I2C0_SDA	TRSTN	TMS/PLL_BYPASS	TDI/CPR_BYPASS	GPIO6_B4	G
H	GPIO2_A2/LCD1_D2/SMC_A6	LCD0_D22	LCD0_D21	LCD0_D20	LCD0_D23	GPIO2_A0/LCD1_D0/SMC_A4	CVDD_1V1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCCIO1	GPIO0_A6/HOST_DRV_VBUS	GPIO0_A2/HDMI_I2C_SDA	PWM2/GPIO0_D6	PWM1/GPIO0_A4	TDO	TCK/HSSCAN_SHIFT_CLOCK	H
J		GPIO2_A1/LCD1_D1/SMC_A5	GPIO2_B5/LCD1_D13/SMC_A17/TS_VALID				LCD0_VCC1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	CVDD_1V1				PWM3/GPIO0_D7	GPIO0_A5/OTG_DRV_VBUS		J
K	GPIO2_C4/LCD1_D20/SPI1_CSN0/TS_D1	GPIO2_C3/LCD1_D19/SPI1_CLK/TS_D0	GPIO2_C2/LCD1_D18/SMC_BIS_N1/TS_D5	GPIO2_C1/LCD1_D17/SMC_BIS_N0/TS_D6	GPIO2_B7/LCD1_D15/SMC_A19/TS_D7	GPIO2_B6/LCD1_D14/SMC_A18/TS_SYNC	LCD0_VCC0	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCCIO0	GPIO3_D5/UART3_CTSN	GPIO3_D4/UART3_TX	GPIO3_B5/SDMMC0_D3	GPIO0_B4/I2S0_SDO0	GPIO2_D7/I2C1_SCL	GPIO3_A0/I2C2_SDA	K
L	GPIO2_C5/LCD1_D21/SPI1_RXD/TS_D2	GPIO2_C6/LCD1_D22/SPI1_RXD/TS_D3	GPIO2_C7/LCD1_D23/SPI1_CSN1/TS_D4	CIF0_D3	GPIO2_C0/LCD1_D16/GPS_CLK/TS_CLKO	CIF0_D2	LCD1_VCC	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	CVDD_1V1	GPIO3_D6/UART3_RTSN	GPIO3_D3/UART3_RX	GPIO3_B6/SDMMC0_DET	GPIO0_A7/I2S0_SDI	GPIO1_B2/SPDIF_TX	GPIO2_D6/I2C1_SDA	L



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
M		CIF0_D5	CIF0_D4				CIF0_VCC	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS				GPIO3_B2/SDMMC0_D0	GPIO0_B7/I2S0_SDO3		M
N	CIF0_D6	CIF0_D7	CIF0_D8	CIF0_VSYNC	AVDD	CIF1_VCC	CVDD_1V1	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	FLASH_VCC	GPIO4_B0/FLASH_CSN1	GPIO3_B7/SDMMC0_WP	FLASH_RDN	GPIO3_D7/FLASH_DQS/EMMC_CLKO	GPIO1_B0/UART2_RX	GPIO0_B6/I2S0_SDO2	N
P	CIF0_D9	CIF0_HREF	CIF0_CLKIN	GPIO1_B5/CIF0_D1	AVDD	AVDD	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	CVDD_1V1	FLASH_D0/EMMC_D0	GPIO0_D0/I2S2_CLK/SMC_CSN0	GPIO0_D1/I2S2_SCLK/SMC_WEN	GPIO4_B2/FLASH_CSN3/EMMC_RSTNO	GPIO3_B1/SDMMC0_CMD	GPIO0_B5/I2S0_SDO1	P
R		GPIO1_B3/CIF0_CLKO	GPIO1_B4/CIF0_D0				VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	SMC_VCC				GPIO0_B3/I2S0_LRCK_TX	GPIO0_B2/I2S0_LRCK_RX		R
T	GPIO1_B6/CIF0_D10	GPIO1_B7/CIF0_D11	GPIO3_A2/I2C3_SDA	GPIO1_C4/CIF1_D6/RMII_RX_ERR	AVDD	AVDD	VSS	VSS	VSS	VSS	AVSS_APLL	AVSS_DPLL	AVSS_C/GPLL	VSS	VSS	VSS	CVDD_1V1	GPIO0_C6/TRACE_CLK/SMC_A2	GPIO4_C3/SMC_D3/TRACE_D3	FLASH_ALE	GPIO3_B4/SDMMC0_D2	GPIO0_B0/I2S0_CLK	GPIO0_B1/I2S0_SCLK	T
U	GPIO3_A3/I2C3_SCL	GPIO1_C0/CIF1_D2/RMII_CLKO	GPIO1_C1/CIF1_D3/RMII_TX_EN	GPIO1_C5/CIF1_D7/RMII_CRS_DVALID	AVDD	AVDD		AVDD	AVDD	APLL_1V1	DPLL_1V1	C/GPLL_1V1	PVDD_1V1	USBVDD_1V1	USBVDD_2V5	AP0_VCC		GPIO4_D4/SMC_D12/TRACE_D12	GPIO4_C5/SMC_D5/TRACE_D5	FLASH_RDY	GPIO4_B1/FLASH_CSN2/EMMC_CMD	GPIO3_B0/SDMMC0_CLKO	GPIO3_B3/SDMMC0_D1	U
V		GPIO1_C2/CIF1_D4/RMII_TXD1	GPIO1_C3/CIF1_D5/RMII_TXD0	GPIO1_C7/CIF1_D9/RMII_RXD0	AVDD		AVDD	AVDD		AVDD_COM	VDDVDD_2V5		PVCC_3V3	USBVDD_3V3		AP1_VCC	GPIO4_C4/SMC_D4/TRACE_D4		GPIO4_C6/SMC_D6/TRACE_D6	FLASH_D5/EMMC_D5	GPIO0_D4/I2S2_SDI/SMC_A0	GPIO3_A7/SDMMC0_PWREN		V
W	GPIO1_C6/CIF1_D8/RMII_RXD1	GPIO1_D0/CIF1_VSYNC/MII_MD	GPIO1_D1/CIF1_HREF/MII_MDCLK	GPIO1_D2/CIF1_CLKIN		ADC_IN0	ADC_IN2	ADC_IN1		GPIO6_B0	GPIO6_A0		HOST_RKELVIN	EFUSE		GPIO4_C2/SMC_D2/TRACE_D2	GPIO4_C1/SMC_D1/TRACE_D1	GPIO4_C7/SMC_D7/TRACE_D7		GPIO4_D2/SMC_D10/TRACE_D10	GPIO0_D3/I2S2_LRCK_TX/SMC_ADVN	FLASH_D3/EMMC_D3	GPIO3_A6/SDMMC0_RSTNO	W
Y	GPIO1_D7/CIF1_CLKO	GPIO1_D6/CIF1_D11	GPIO3_A5/I2C4_SCL	GPIO3_A4/I2C4_SDA	GPIO6_A4	GPIO6_A5	NPOR	GPIO6_A6		GPIO3_C6/SDMMC1_DET	GPIO1_A1/UART0_TX		GPIO3_D0/SDMMC1_PWREN	GPIO0_C4/I2S1_SDI		GPIO0_A1/HDMI_I2C_SCL	GPIO4_C0/SMC_D0/TRACE_D0	GPIO4_D0/SMC_D8/TRACE_D8	GPIO3_A1/I2C2_SCL	GPIO4_D6/SMC_D14/TRACE_D14	GPIO0_D2/I2S2_LRCK_RX/SMC_OEN	FLASH_D4/EMMC_D4	FLASH_CSN0	Y
AA		GPIO6_B2	GPIO6_A1	CLK32K_IN	GPIO6_A3	OTG_RKELVIN	GPIO1_A4/UART1_RX/SPI0_CSN0	GPIO3_D2/SDMMC1_INT	GPIO3_C3/SDMMC1_D2	GPIO3_C4/SDMMC1_D3	GPIO1_A2/UART0_CTSN	GPIO3_C5/SDMMC1_CLKO	GPIO3_C7/SDMMC1_WP	GPIO3_D1/SDMMC1_BACKEND	PWM0/GPIO0_A3	VSS	VSS	GPIO4_D1/SMC_D9/TRACE_D9	GPIO0_A0/HDMI_HPD	GPIO4_D7/SMC_D15/TRACE_D15	FLASH_D1/EMMC_D1	FLASH_D6/EMMC_D6		AA
AB	CPU_PWROFF	GPIO6_B1	GPIO6_A2	OTG_ID	OTG_VBUS	GPIO1_A6/UART1_CTSN/SPI0_RXD	GPIO1_A7/UART1_RTSN/SPI0_TXD	GPIO3_C0/SDMMC1_CMD	GPIO3_C2/SDMMC1_D1	VSS	XOUT24M	GPIO1_A0/UART0_RX	GPIO0_C1/I2S1_SCLK	GPIO0_C3/I2S1_LRCK_TX	GPIO0_C5/I2S1_SDO	OTG_DM	HOST_DM	GPIO4_D3/SMC_D11/TRACE_D11	GPIO1_B1/UART2_TX	GPIO0_C7/TRACE_CTL/SMC_A3	FLASH_D2/EMMC_D2	FLASH_D7/EMMC_D7	FLASH_WP/EMMC_PWREN	AB
AC	CORE_PWROFF	GPIO6_B3		GPIO6_A7	GPIO1_A5/UART1_TX/SPI0_CLK		GPIO4_B7/SPI0_CSN1	GPIO3_C1/SDMMC1_D0		GPIO1_A3/UART0_RTSN	XIN24M		GPIO0_C0/I2S1_CLK	GPIO0_C2/I2S1_LRCK_RX		OTG_DP	HOST_DP		GPIO4_D5/SMC_D13/TRACE_D13	GPIO0_D5/I2S2_SDO/SMC_A1		FLASH_CLE	FLASH_WRN	AC

Fig. 1-6 RK3066 Ball Mapping Diagram

1.4.3 Ball Pin Number Order

Table 1-5 RK3066Ball Pin Number Order Information

Ball #	Ball Pin Name	Ball #	Ball Pin Name
A1	HDMI_TX2	B1	HDMI_TX2N
A2	HDMI_TX1	B2	HDMI_TX1N
A3	NC	B3	HDMI_AVSS
A4	HDMI_TX0	B4	HDMI_TX0N
A5	HDMI_TXC	B5	HDMI_TXCN
A6	NC	B6	HDMI_AVSS
A7	MDQ27	B7	MDQ26
A8	MDQS_B3	B8	MDQS_3
A9	NC	B9	MDQ8
A10	MDQ10	B10	MDQ11
A11	MDQS_B1	B11	MDQS_1
A12	NC	B12	MDQ15
A13	MCKE0	B13	MCSN0
A14	MBA0	B14	MBA1
A15	NC	B15	MA0
A16	MCK_N	B16	MCK
A17	MA1	B17	MA4
A18	NC	B18	MA6
A19	MA7	B19	MA8
A20	MA11	B20	MODT0
A21	NC	B21	MDQ16
A22	MDQS_B2	B22	MDQS_2
A23	MDQ20	B23	MDQ21
C1	NC	D1	LCD0_D4
C2	LCD0_D2	D2	LCD0_D6
C3	LCD0_D1	D3	LCD0_D3
C4	LCD0_D0	D4	LCD0_DEN
C5	HDMI_AVSS	D5	LCD0_VSYNC
C6	HDMI_REXT	D6	LCD0_HSYNC
C7	MDQ24	D7	MDQ25
C8	MDQ31	D8	MDQ29
C9	MDM1	D9	NC
C10	MDQ9	D10	MDQ12
C11	MDQ14	D11	MCKE1
C12	MRESET	D12	NC
C13	MWEN	D13	MCASN
C14	MBA2	D14	MRASN
C15	MA3	D15	NC
C16	VSS	D16	MA14

C17	MA5	D17	MDQ19
C18	MA9	D18	MA10
C19	MA13	D19	MA12
C20	MODT1	D20	VSS
C21	MDM2	D21	MDQ22
C22	MDQ17	D22	MDQ23
C23	NC	D23	MDM0
E1	LCD0_D12	F1	NC
E2	LCD0_D10	F2	LCD0_D16
E3	LCD0_D9	F3	LCD0_D11
E4	LCD0_D7	F4	LCD0_D8
E5	NC	F5	LCD0_D5
E6	LCD0_DCLK	F6	NC
E7	MDM3	F7	HDMIVDD_1V1
E8	MDQ28	F8	HDMIAVDD_2V5
E9	NC	F9	NC
E10	MDQ30	F10	MVDD
E11	MDQ13	F11	MVDD
E12	NC	F12	NC
E13	MCSN1	F13	MVREF
E14	MA2	F14	MVDD
E15	NC	F15	NC
E16	MDQ18	F16	MVDD
E17	MDQ0	F17	MVDD
E18	MDQ3	F18	NC
E19	NC	F19	MDQ4
E20	MDQ2	F20	MDQ7
E21	MDQ1	F21	MDQ5
E22	MDQS_0	F22	MDQ6
E23	MDQS_B0	F23	NC
G1	LCD0_D19	H1	GPIO2_A2/LCD1_D2/SMC_A6
G2	LCD0_D18	H2	LCD0_D22
G3	LCD0_D17	H3	LCD0_D21
G4	LCD0_D14	H4	LCD0_D20
G5	LCD0_D13	H5	LCD0_D23
G6	LCD0_D15	H6	GPIO2_A0/LCD1_D0/SMC_A4
G7	NC	H7	CVDD_1V1
G8	HDMI_AVSS	H8	VSS
G9	HDMI_AVSS	H9	VSS
G10	MVDD	H10	VSS
G11	CVDD_1V1	H11	VSS
G12	VSS	H12	VSS

G13	MVDD	H13	VSS
G14	MVDD	H14	VSS
G15	MPZQ	H15	VSS
G16	CVDD_1V1	H16	VSS
G17	NC	H17	VCCIO1
G18	GPIO2_D5/I2C0_SCL	H18	GPIO0_A6/HOST_DRV_VBUS
G19	GPIO2_D4/I2C0_SDA	H19	GPIO0_A2/HDMI_I2C_SDA
G20	TRSTN	H20	PWM2/GPIO0_D6
G21	TMS/PLL_BYPASS	H21	PWM1/GPIO0_A4
G22	TDI/CPR_BYPASS	H22	TDO
G23	GPIO6_B4	H23	TCK/HSSCAN_SHIFT_CLOCK
J1	NC	K1	GPIO2_C4/LCD1_D20/SPI1_CSN0/TS_D1
J2	GPIO2_A1/LCD1_D1/SMC_A5	K2	GPIO2_C3/LCD1_D19/SPI1_CLK/TS_D0
J3	GPIO2_B5/LCD1_D13/SMC_A17/TS_VALID	K3	GPIO2_C2/LCD1_D18/SMC_BIS_N1/TS_D5
J4	NC	K4	GPIO2_C1/LCD1_D17/SMC_BIS_N0/TS_D6
J5	NC	K5	GPIO2_B7/LCD1_D15/SMC_A19/TS_D7
J6	NC	K6	GPIO2_B6/LCD1_D14/SMC_A18/TS_SYNC
J7	LCD0_VCC1	K7	LCD0_VCC0
J8	VSS	K8	VSS
J9	VSS	K9	VSS
J10	VSS	K10	VSS
J11	VSS	K11	VSS
J12	VSS	K12	VSS
J13	VSS	K13	VSS
J14	VSS	K14	VSS
J15	VSS	K15	VSS
J16	VSS	K16	VSS
J17	CVDD_1V1	K17	VCCIO0
J18	NC	K18	GPIO3_D5/UART3_CTSN
J19	NC	K19	GPIO3_D4/UART3_TX
J20	NC	K20	GPIO3_B5/SDMMC0_D3
J21	PWM3/GPIO0_D7	K21	GPIO0_B4/I2S0_SDO0
J22	GPIO0_A5/OTG_DRV_VBUS	K22	GPIO2_D7/I2C1_SCL
J23	NC	K23	GPIO3_A0/I2C2_SDA
L1	GPIO2_C5/LCD1_D21/SPI1_RXD/TS_D2	M1	NC
L2	GPIO2_C6/LCD1_D22/SPI1_RXD/TS_D3	M2	CIF0_D5
L3	GPIO2_C7/LCD1_D23/SPI1_CSN1/TS_D4	M3	CIF0_D4
L4	CIF0_D3	M4	NC
L5	GPIO2_C0/LCD1_D16/GPS_CLK/TS_CLKO	M5	NC
L6	CIF0_D2	M6	NC
L7	LCD1_VCC	M7	CIF0_VCC
L8	VSS	M8	VSS

L9	VSS	M9	VSS
L10	VSS	M10	VSS
L11	VSS	M11	VSS
L12	VSS	M12	VSS
L13	VSS	M13	VSS
L14	VSS	M14	VSS
L15	VSS	M15	VSS
L16	VSS	M16	VSS
L17	CVDD_1V1	M17	VSS
L18	GPIO3_D6/UART3_RTSN	M18	NC
L19	GPIO3_D3/UART3_RX	M19	NC
L20	GPIO3_B6/SDMMC0_DET	M20	NC
L21	GPIO0_A7/I2S0_SDI	M21	GPIO3_B2/SDMMC0_D0
L22	GPIO1_B2/SPDIF_TX	M22	GPIO0_B7/I2S0_SDO3
L23	GPIO2_D6/I2C1_SDA	M23	NC
N1	CIF0_D6	P1	CIF0_D9
N2	CIF0_D7	P2	CIF0_HREF
N3	CIF0_D8	P3	CIF0_CLKIN
N4	CIF0_VSYNC	P4	GPIO1_B5/CIF0_D1
N5	AVDD	P5	AVDD
N6	CIF1_VCC	P6	AVDD
N7	CVDD_1V1	P7	VSS
N8	VSS	P8	VSS
N9	VSS	P9	VSS
N10	VSS	P10	VSS
N11	VSS	P11	VSS
N12	VSS	P12	VSS
N13	VSS	P13	VSS
N14	VSS	P14	VSS
N15	VSS	P15	VSS
N16	VSS	P16	VSS
N17	FLASH_VCC	P17	CVDD_1V1
N18	GPIO4_B0/FLASH_CSN1	P18	FLASH_D0/EMMC_D0
N19	GPIO3_B7/SDMMC0_WP	P19	GPIO0_D0/I2S2_CLK/SMC_CSN0
N20	FLASH_RDN	P20	GPIO0_D1/I2S2_SCLK/SMC_WEN
N21	GPIO3_D7/FLASH_DQS/EMMC_CLKO	P21	GPIO4_B2/FLASH_CSN3/EMMC_RSTNO
N22	GPIO1_B0/UART2_RX	P22	GPIO3_B1/SDMMC0_CMD
N23	GPIO0_B6/I2S0_SDO2	P23	GPIO0_B5/I2S0_SDO1
R1	NC	T1	GPIO1_B6/CIF0_D10
R2	GPIO1_B3/CIF0_CLKO	T2	GPIO1_B7/CIF0_D11
R3	GPIO1_B4/CIF0_D0	T3	GPIO3_A2/I2C3_SDA
R4	NC	T4	GPIO1_C4/CIF1_D6/RMII_RX_ERR

R5	NC	T5	AVDD
R6	NC	T6	AVDD
R7	VSS	T7	VSS
R8	VSS	T8	VSS
R9	VSS	T9	VSS
R10	VSS	T10	VSS
R11	VSS	T11	AVSS_APLL
R12	VSS	T12	AVSS_DPLL
R13	VSS	T13	AVSS_C/GPLL
R14	VSS	T14	VSS
R15	VSS	T15	VSS
R16	VSS	T16	VSS
R17	SMC_VCC	T17	CVDD_1V1
R18	NC	T18	GPIO0_C6/TRACE_CLK/SMC_A2
R19	NC	T19	GPIO4_C3/SMC_D3/TRACE_D3
R20	NC	T20	FLASH_ALE
R21	GPIO0_B3/I2S0_LRCK_TX	T21	GPIO3_B4/SDMMC0_D2
R22	GPIO0_B2/I2S0_LRCK_RX	T22	GPIO0_B0/I2S0_CLK
R23	NC	T23	GPIO0_B1/I2S0_SCLK
U1	GPIO3_A3/I2C3_SCL	V1	NC
U2	GPIO1_C0/CIF1_D2/RMII_CLKO	V2	GPIO1_C2/CIF1_D4/RMII_TXD1
U3	GPIO1_C1/CIF1_D3/RMII_TX_EN	V3	GPIO1_C3/CIF1_D5/RMII_TXD0
U4	GPIO1_C5/CIF1_D7/RMII_CRS_DVALID	V4	GPIO1_C7/CIF1_D9/RMII_RXD0
U5	AVDD	V5	AVDD
U6	AVDD	V6	NC
U7	NC	V7	AVDD
U8	AVDD	V8	AVDD
U9	AVDD	V9	NC
U10	APLL_1V1	V10	AVDD_COM
U11	DPLL_1V1	V11	ADCVDD_2V5
U12	C/GPLL_1V1	V12	NC
U13	PVDD_1V1	V13	PVCC_3V3
U14	USBVDD_1V1	V14	USBVDD_3V3
U15	USBVDD_2V5	V15	NC
U16	AP0_VCC	V16	AP1_VCC
U17	NC	V17	GPIO4_C4/SMC_D4/TRACE_D4
U18	GPIO4_D4/SMC_D12/TRACE_D12	V18	NC
U19	GPIO4_C5/SMC_D5/TRACE_D5	V19	GPIO4_C6/SMC_D6/TRACE_D6
U20	FLASH_RDY	V20	FLASH_D5/EMMC_D5
U21	GPIO4_B1/FLASH_CSN2/EMMC_CMD	V21	GPIO0_D4/I2S2_SDI/SMC_A0
U22	GPIO3_B0/SDMMC0_CLKO	V22	GPIO3_A7/SDMMC0_PWREN
U23	GPIO3_B3/SDMMC0_D1	V23	NC

W1	GPIO1_C6/CIF1_D8/RMII_RXD1	Y1	GPIO1_D7/CIF1_CLKO
W2	GPIO1_D0/CIF1_VSYNC/MII_MD	Y2	GPIO1_D6/CIF1_D11
W3	GPIO1_D1/CIF1_HREF/MII_MDCLK	Y3	GPIO3_A5/I2C4_SCL
W4	GPIO1_D2/CIF1_CLKIN	Y4	GPIO3_A4/I2C4_SDA
W5	NC	Y5	GPIO6_A4
W6	ADC_IN0	Y6	GPIO6_A5
W7	ADC_IN2	Y7	NPOR
W8	ADC_IN1	Y8	GPIO6_A6
W9	NC	Y9	NC
W10	GPIO6_B0	Y10	GPIO3_C6/SDMMC1_DET
W11	GPIO6_A0	Y11	GPIO1_A1/UART0_TX
W12	NC	Y12	NC
W13	HOST_RKELVIN	Y13	GPIO3_D0/SDMMC1_PWREN
W14	EFUSE	Y14	GPIO0_C4/I2S1_SDI
W15	NC	Y15	NC
W16	GPIO4_C2/SMC_D2/TRACE_D2	Y16	GPIO0_A1/HDMI_I2C_SCL
W17	GPIO4_C1/SMC_D1/TRACE_D1	Y17	GPIO4_C0/SMC_D0/TRACE_D0
W18	GPIO4_C7/SMC_D7/TRACE_D7	Y18	GPIO4_D0/SMC_D8/TRACE_D8
W19	NC	Y19	GPIO3_A1/I2C2_SCL
W20	GPIO4_D2/SMC_D10/TRACE_D10	Y20	GPIO4_D6/SMC_D14/TRACE_D14
W21	GPIO0_D3/I2S2_LRCK_TX/SMC_ADVN	Y21	GPIO0_D2/I2S2_LRCK_RX/SMC_OEN
W22	FLASH_D3/EMMC_D3	Y22	FLASH_D4/EMMC_D4
W23	GPIO3_A6/SDMMC0_RSTNO	Y23	FLASH_CSN0
AA1	NC	AB1	CPU_PWROFF
AA2	GPIO6_B2	AB2	GPIO6_B1
AA3	GPIO6_A1	AB3	GPIO6_A2
AA4	CLK32K_IN	AB4	OTG_ID
AA5	GPIO6_A3	AB5	OTG_VBUS
AA6	OTG_RKELVIN	AB6	GPIO1_A6/UART1_CTSN/SPI0_RXD
AA7	GPIO1_A4/UART1_RX/SPI0_CSN0	AB7	GPIO1_A7/UART1_RTSN/SPI0_TXD
AA8	GPIO3_D2/SDMMC1_INT	AB8	GPIO3_C0/SDMMC1_CMD
AA9	GPIO3_C3/SDMMC1_D2	AB9	GPIO3_C2/SDMMC1_D1
AA10	GPIO3_C4/SDMMC1_D3	AB10	VSS
AA11	GPIO1_A2/UART0_CTSN	AB11	XOUT24M
AA12	GPIO3_C5/SDMMC1_CLKO	AB12	GPIO1_A0/UART0_RX
AA13	GPIO3_C7/SDMMC1_WP	AB13	GPIO0_C1/I2S1_SCLK
AA14	GPIO3_D1/SDMMC1_BACKEND	AB14	GPIO0_C3/I2S1_LRCK_TX
AA15	PWM0/GPIO0_A3	AB15	GPIO0_C5/I2S1_SDO
AA16	VSS	AB16	OTG_DM
AA17	VSS	AB17	HOST_DM
AA18	GPIO4_D1/SMC_D9/TRACE_D9	AB18	GPIO4_D3/SMC_D11/TRACE_D11
AA19	GPIO0_A0/HDMI_HPD	AB19	GPIO1_B1/UART2_TX

AA20	GPIO4_D7/SMC_D15/TRACE_D15	AB20	GPIO0_C7/TRACE_CTL/SMC_A3
AA21	FLASH_D1/EMMC_D1	AB21	FLASH_D2/EMMC_D2
AA22	FLASH_D6/EMMC_D6	AB22	FLASH_D7/EMMC_D7
AA23	NC	AB23	FLASH_WP/EMMC_PWREN
AC1	CORE_PWROFF	AC13	GPIO0_C0/I2S1_CLK
AC2	GPIO6_B3	AC14	GPIO0_C2/I2S1_LRCK_RX
AC3	NC	AC15	NC
AC4	GPIO6_A7	AC16	OTG_DP
AC5	GPIO1_A5/UART1_TX/SPI0_CLK	AC17	HOST_DP
AC6	NC	AC18	NC
AC7	GPIO4_B7/SPI0_CSN1	AC19	GPIO4_D5/SMC_D13/TRACE_D13
AC8	GPIO3_C1/SDMMC1_D0	AC20	GPIO0_D5/I2S2_SDO/SMC_A1
AC9	NC	AC21	NC
AC10	GPIO1_A3/UART0_RTSN	AC22	FLASH_CLE
AC11	XIN24M	AC23	FLASH_WRN
AC12	NC		

1.5 Electrical Specification

1.5.1 Absolute Maximum Ratings

Table 1-6 RK30xx absolute maximum ratings

Paramerters	Related Power Group	Max	Unit
DC supply voltage for Internal digital logic	AVDD, CVDD, PVDD, OTG_DVDD, HOST_DVDD	1.21	V
DC supply voltage for Digital GPIO (except for SAR-ADC, TS-ADC, PLL, USB, DDR IO)	LCD0_VCC0,LCD0_VCC1, LCD1_VCC, CIF0_VCC,CIF1_VCC, PVCC, AP0_VCC,AP1_VCC, SMC_VCC,FLASH_VCC, VCCIO_0,VCCIO_1	3.6	V
DC supply voltage for DDR IO	MVDD	1.65	V
DC supply voltage for Analog part of SAR-ADC/TS-ADC	VDDA_SARADC, VDDA_TSADC	2.75	V
DC supply voltage for Analog part of PLL	AVDD_APLL,AVDD_DPLL, AVDD_CGPLL	1.21	V
DC supply voltage for Analog part of USB OTG/Host2.0	OTG_VDD25,HOST_VDD25 OTG_VDD33,HOST_VDD33	2.75 3.63	V
DC supply voltage for Analog part of HDMI	HDMIVDD HDMIAVDD	1.21 2.75	V
DC supply voltage for Analog part of EFUSE	EFUSE_VDDQ	2.75	V
Analog Input voltage for SAR-ADC		2.75	V

Analog Input voltage for TS-ADC		2.75	V
Analog Input voltage for DP/DM/VBUS of USB OTG/Host2.0		5	V
Analog input voltage for RKELVIN/ID of USB OTG/Host2.0		2.75	V
Digital input voltage for input buffer of GPIO		3.6	V
Digital output voltage for output buffer of GPIO		3.6	V
Storage Temperature		125	℃

Absolute maximum ratings specify the values beyond which the device may be damaged permanently. Long-term exposure to absolute maximum ratings conditions may affect device reliability.

1.5.2 Recommended Operating Conditions

Table 1-7 RK30xx recommended operating conditions

Parameters	Symbol	Min	Typ	Max	Units
Internal digital logic Power （except USB OTG)	AVDD, CVDD, PVDD	0.99	1.10	**1.21**	V
Digital GPIO Power(3.3V)	VCCIO0,VCCIO1, EFUSE_VDDQ	3	3.3	3.6	V
Digital GPIO Power(3.3V/1.8V)	LCD0_VCC0, LCD0_VCC1, LCD1_VCC, SMC_VCC, CIF0_VCC,CIF1_VCC, PVCC, FLASH_VCC, AP0_VCC, AP1_VCC	3 1.62	3.3 1.8	3.6 1.98	V
DDR IO (DDR3 mode) Power	MVDD	1.425	1.5	1.575	V
DDR IO (LPDDR2 mode) Power	MVDD	1.14	1.2	1.30	V
DDR IO (LVDDR3 mode) Power	MVDD	1.28	1.35	1.45	V
DDR reference supply (VREF) Input	VREF	0.49*MVDD	0.5*MVDD	0.51*MVDD	V
DDR External termination voltage		VREF - 40mV	VREF	VREF + 40mV	V
PLLAnalog Power	AVDD_APLL, AVDD_DPLL, AVDD_CGPLL	0.99	1.1	1.21	V
SAR-ADC Analog Power	VDDA_SARADC	2.25	2.5	2.75	V
TS-ADC Analog Power	VDDA_TSADC	2.25	2.5	2.75	V
USB OTG/Host2.0 Digital Power	OTG_DVDD, HOST_DVDD	1.023	1.1	1.21	V
USB OTG/Host2.0 Analog Power(2.5V)	OTG_VDD25,HOST_VDD25	2.325	2.5	2.75	V
USB OTG/Host2.0 Analog Power(3.3V)	OTG_VDD33,HOST_VDD33	3.069	3.3	3.63	V
USB OTG/Host2.0 external resistor	REXT	42.768	43.2	43.632	Ohm

PLL input clock frequency		N/A	24	N/A	MHz
Operating Temperature		-40	25	85	℃

1.5.3 DC Characteristics

Table 1-8 RK30xx DC Characteristics

Parameters		Symbol	Min	Typ	Max	Units
Digital GPIO @3.3V	Input Low Voltage	V_{il}	-0.3	0	0.8	V
	Input High Voltage	V_{ih}	2	3.3	3.6	V
	Output Low Voltage	V_{ol}	N/A	0	0.4	V
	Output High Voltage	V_{oh}	2.4	3.3	N/A	V
	Threshold Point	V_t	1.34	1.46	1.6	V
	Threshold Point with Pullup Resistor Enabled	V_{tpu}	1.2	1.31	1.45	V
	Threshold Point with Pulldown Resistor Enabled	V_{tpd}	1.71	1.84	1.97	V
	Pullup Resistor	R_{pu}	41	60	91	Kohm
	Pulldown Resistor	R_{pd}	43	63	103	Kohm
Digital GPIO @1.8V	Input Low Voltage	V_{il}	-0.3	0	0.63	V
	Input High Voltage	V_{ih}	1.17	1.8	3.6	V
	Output Low Voltage	V_{ol}	N/A	0	0.45	V
	Output High Voltage	V_{oh}	1.35	1.8	N/A	V
	Threshold Point	V_t	0.77	0.84	0.92	V
	Threshold Point with Pullup Resistor Enabled	V_{tpu}	0.77	0.84	0.91	V
	Threshold Point with Pulldown Resistor Enabled	V_{tpd}	0.77	0.85	0.92	V
	Pullup Resistor	R_{pu}	79	129	218	Kohm
	Pulldown Resistor	R_{pd}	73	127	233	Kohm
DDR IO @DDR3 mode	Input High Voltage	V_{ih_ddr}	VREF + 0.1	1.5	MVDD	V
	Input Low Voltage	V_{il_ddr}	-0.3	0	VREF - 0.1	V
	Output High Voltage	V_{oh_ddr}	0.8 * MVDD	1.5	N/A	V
	Output Low Voltage	V_{ol_ddr}	N/A	0	0.2 * MVDD	V
	Input termination resistance(ODT) to VDDIO_BLi/2 (i=0~3)	R_{tt}	100 54 36	120 60 40	140 66 44	Ohm
DDR IO @LPDDR2 mode	Input High Voltage	V_{ih_ddr}	VREF + 0.13	1.2	MVDD	V
	Input Low Voltage	V_{il_ddr}	-0.3	0	VREF- 0.13	V
	Output High Voltage	V_{oh_ddr}	0.9 * MVDD	1.2	N/A	V
	Output Low Voltage	V_{ol_ddr}	N/A	0	0.1 * MVDD	V

1.5.4 Recommended Operating Frequency

Table 1-9 Recommended operating frequency for PD_ALIVE domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
XIN Oscillator	1.1V , 25 ℃	IO_XIN24M	24	24	24	MHz
	1.21V , -40 ℃		24	24	24	
	0.99V , 125 ℃		24	24	24	
DDR PLL	1.1V , 25 ℃	ddr_pll_clk	N/A	N/A	1333	MHz
	1.21V , -40 ℃		N/A	N/A	1976	
	0.99V , 125 ℃		N/A	N/A	835	
ARM PLL	1.1V , 25 ℃	arm_pll_clk	N/A	N/A	1250	MHz
	1.21V , -40 ℃		N/A	N/A	1683	
	0.99V , 125 ℃		N/A	N/A	800	
CODEC PLL	1.1V , 25 ℃	cocec_pll_clk	N/A	N/A	1108	MHz
	1.21V , -40 ℃		N/A	N/A	1572	
	0.99V , 125 ℃		N/A	N/A	734	
GENERAL PLL	1.1V , 25 ℃	general_pll_clk	N/A	N/A	1392	MHz
	1.21V , -40 ℃		N/A	N/A	1529	
	0.99V , 125 ℃		N/A	N/A	717	
UART1CLK	1.1V , 25 ℃	clk_uart1	N/A	N/A	273	MHz
	1.21V , -40 ℃		N/A	N/A	304	
	0.99V , 125 ℃		N/A	N/A	200	
TIMER2 CLK	1.1V , 25 ℃	clk_timer2	N/A	N/A	211	MHz
	1.21V , -40 ℃		N/A	N/A	349	
	0.99V , 125 ℃		N/A	N/A	132	

Table 1-10 Recommended operating frequency for A9 core

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
Cortex-A9	1.1V , 25 ℃	clk_core_pre	N/A	N/A	1131	MHz
	1.21V , -40 ℃		N/A	N/A	1333	
	0.99V , 125 ℃		N/A	N/A	758	
	1.1V , 25 ℃	clk_core_peri	N/A	N/A	178	MHz
	1.21V , -40 ℃		N/A	N/A	269	
	0.99V , 125 ℃		N/A	N/A	122	

Table 1-11 Recommended operating frequency for PD_CPU domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
CPU AXI interconnect	1.1V , 25 ℃	aclk_cpu	N/A	N/A	540	MHz
	1.21V , -40 ℃		N/A	N/A	877	
	0.99V , 125 ℃		N/A	N/A	358	
	1.1V , 25 ℃	hclk_cpu	N/A	N/A	467	MHz
	1.21V , -40 ℃		N/A	N/A	300	
	0.99V , 125 ℃		N/A	N/A	198	
	1.1V , 25 ℃	pclk_cpu	N/A	N/A	136	MHz

	1.21V , -40 ℃		N/A	N/A	205	
	0.99V , 125 ℃		N/A	N/A	95	
DMC	1.1V , 25 ℃	clk_ddr	N/A	N/A	588	MHz
	1.21V , -40 ℃		N/A	N/A	909	
	0.99V , 125 ℃		N/A	N/A	394	
Embedded SRAM	1.1V , 25 ℃	aclk_intmem	N/A	N/A	322	MHz
	1.21V , -40 ℃		N/A	N/A	500	
	0.99V , 125 ℃		N/A	N/A	216	
SPDIF	1.1V , 25 ℃	clk_spdif	N/A	N/A	356	MHz
	1.21V , -40 ℃		N/A	N/A	588	
	0.99V , 125 ℃		N/A	N/A	219	
Timer0/1	1.1V , 25 ℃	clk_timer0/ clk_timer1	N/A	N/A	177	MHz
	1.21V , -40 ℃		N/A	N/A	108	
	0.99V , 125 ℃		N/A	N/A	2500	
UART0	1.1V , 25 ℃	clk_uart0	N/A	N/A	263	MHz
	1.21V , -40 ℃		N/A	N/A	301	
	0.99V , 125 ℃		N/A	N/A	218	
I2S0/I2S1/I2S2	1.1V , 25 ℃	clk_i2s0/ clk_i2s1/ clk_i2s2	N/A	N/A	60	MHz
	1.21V , -40 ℃		N/A	N/A	69	
	0.99V , 125 ℃		N/A	N/A	52	

Table 1-12 Recommended operating frequency for PD_PERI domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
PERI AXI interconnect	1.1V , 25 ℃	aclk_periph	N/A	N/A	480	MHz
	1.21V , -40 ℃		N/A	N/A	725	
	0.99V , 125 ℃		N/A	N/A	314	
	1.1V , 25 ℃	hclk_periph	N/A	N/A	237	MHz
	1.21V , -40 ℃		N/A	N/A	370	
	0.99V , 125 ℃		N/A	N/A	150	
	1.1V , 25 ℃	pclk_periph	N/A	N/A	135	MHz
	1.21V , -40 ℃		N/A	N/A	213	
	0.99V , 125 ℃		N/A	N/A	94	
SMC	1.1V , 25 ℃	clk_smc	N/A	N/A	234	MHz
	1.21V , -40 ℃		N/A	N/A	395	
	0.99V , 125 ℃		N/A	N/A	159	
NANDC	1.1V , 25 ℃	hclk_nandc	N/A	N/A	251	MHz
	1.21V , -40 ℃		N/A	N/A	395	
	0.99V , 125 ℃		N/A	N/A	162	
USB Host	1.1V , 25 ℃	clk_otgphy1	N/A	N/A	161	MHz
	1.21V , -40 ℃		N/A	N/A	266	
	0.99V , 125 ℃		N/A	N/A	101	
USB OTG	1.1V , 25 ℃	clk_otgphy0	N/A	N/A	128	MHz
	1.21V , -40 ℃		N/A	N/A	212	

	0.99V , 125 ℃		N/A	N/A	80	
UART2/3	1.1V , 25 ℃	clk_uart2/ clk_uart3	N/A	N/A	278	MHz
	1.21V , -40 ℃		N/A	N/A	304	
	0.99V , 125 ℃		N/A	N/A	250	
SDMMC/SDIO	1.1V , 25 ℃	clk_sdmmc0/ clk_sdio	N/A	N/A	192	MHz
	1.21V , -40 ℃		N/A	N/A	311	
	0.99V , 125 ℃		N/A	N/A	121	
EMMC	1.1V , 25 ℃	clk_emmc	N/A	N/A	185	MHz
	1.21V , -40 ℃		N/A	N/A	299	
	0.99V , 125 ℃		N/A	N/A	117	
MAC	1.1V , 25 ℃	clk_mac_ref	N/A	N/A	54	MHz
	1.21V , -40 ℃		N/A	N/A	58	
	0.99V , 125 ℃		N/A	N/A	52	
SPI0/1	1.1V , 25 ℃	clk_spi0/ clk_spi1	N/A	N/A	91	MHz
	1.21V , -40 ℃		N/A	N/A	100	
	0.99V , 125 ℃		N/A	N/A	86	
SAR-ADC	1.1V , 25 ℃	clk_saradc	N/A	N/A	69	MHz
	1.21V , -40 ℃		N/A	N/A	78	
	0.99V , 125 ℃		N/A	N/A	59	
TS-ADC	1.1V , 25 ℃	clk_tsadc	N/A	N/A	22	MHz
	1.21V , -40 ℃		N/A	N/A	23	
	0.99V , 125 ℃		N/A	N/A	21	
HSADC	1.1V , 25 ℃	clk_hsadc	N/A	N/A	68	MHz
	1.21V , -40 ℃		N/A	N/A	77	
	0.99V , 125 ℃		N/A	N/A	58	

Table 1-13 Recommended operating frequency for PD_VIO domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
Display AXI interconnection	1.1V , 25 ℃	aclk_lcdc0	N/A	N/A	482	MHz
	1.21V , -40 ℃		N/A	N/A	739	
	0.99V , 125 ℃		N/A	N/A	314	
	1.1V , 25 ℃	aclk_lcdc1	N/A	N/A	476	MHz
	1.21V , -40 ℃		N/A	N/A	736	
	0.99V , 125 ℃		N/A	N/A	310	
	1.1V , 25 ℃	hclk_lcdc	N/A	N/A	377	MHz
	1.21V , -40 ℃		N/A	N/A	648	
	0.99V , 125 ℃		N/A	N/A	235	
LCDC0/1	1.1V , 25 ℃	dclk_lcdc0/ dclk_lcdc1	N/A	N/A	269	MHz
	1.21V , -40 ℃		N/A	N/A	448	
	0.99V , 125 ℃		N/A	N/A	168	
CIF0/1	1.1V , 25 ℃	pclkin_cif0/ pclkin_cif1	N/A	N/A	101	MHz
	1.21V , -40 ℃		N/A	N/A	103	
	0.99V , 125 ℃		N/A	N/A	103	

HDMI	1.1V , 25 ℃	clk_hdmi	N/A	N/A	417	MHz
	1.21V , -40 ℃		N/A	N/A	648	
	0.99V , 125 ℃		N/A	N/A	258	

Table 1-14 Recommended operating frequency PD_GPU domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
GPU	1.1V , 25 ℃	aclk_gpu	N/A	N/A	480	MHz
	1.21V , -40 ℃		N/A	N/A	717	
	0.99V , 125 ℃		N/A	N/A	318	

Table 1-15 Recommended operating frequency for PD_VIDEO domain

Parameter	Condition	Symbol	MIN	TYP	MAX	Unit
VIDEO	1.1V , 25 ℃	aclk_vepu	N/A	N/A	479	MHz
	1.21V , -40 ℃		N/A	N/A	738	
	0.99V , 125 ℃		N/A	N/A	316	
	1.1V , 25 ℃	hclk_vepu	N/A	N/A	126	MHz
	1.21V , -40 ℃		N/A	N/A	197	
	0.99V , 125 ℃		N/A	N/A	86	
	1.1V , 25 ℃	aclk_vdpu	N/A	N/A	417	MHz
	1.21V , -40 ℃		N/A	N/A	648	
	0.99V , 125 ℃		N/A	N/A	275	
	1.1V , 25 ℃	hclk_vdpu	N/A	N/A	115	MHz
	1.21V , -40 ℃		N/A	N/A	179	
	0.99V , 125 ℃		N/A	N/A	78	

1.5.5 Electrical Characteristics for General IO

Table 1-16RK30xx Electrical Characteristics for Digital General IO

Parameters		Symbol	Test condition	Min	Typ	Max	Units
Digital GPIO @3.3V	Input leakage current	I_i	Vin = 3.3V or 0V	-10	N/A	10	uA
	Tri-state output leakage current	I_{oz}	Vout = 3.3V or 0V	-10	N/A	10	uA
	High level input current	I_{ih}	Vin = 3.3V, pulldown disabled	TBD	N/A	TBD	uA
			Vin = 3.3V, pulldown enabled	32	52	77	uA
	Low level input current	I_{il}	Vin = 0V, pullup disabled	TBD	N/A	TBD	uA
			Vin = 0V, pullup enabled	36	55	80	uA
Digital GPIO @1.8V	Input leakage current	I_i	Vin = 1.8V or 0V	-10	N/A	10	uA
	Tri-state output leakage current	I_{oz}	Vout = 1.8V or 0V	-10	N/A	10	uA
	High level input current	I_{ih}	Vin = 1.8V, pulldown disabled	TBD	N/A	TBD	uA
			Vin = 1.8V, pulldown enabled	7.7	14	25	uA
	Low level input current	I_{il}	Vin = 0V, pullup disabled	TBD	N/A	TBD	uA
			Vin = 0V, pullup enabled	8.3	14	23	uA

1.5.6 Electrical Characteristics for PLL

Table 1-17 RK30xx Electrical Characteristics for PLL

Parameters	Symbol	Test condition	Min	Typ	Max	Units
Input clock frequency	F_{in}	$F_{in} = F_{ref} * NR$① @1.1V	0.183	24	1500	MHz
Comparison frequency	F_{ref}	$F_{ref} = F_{in} / NR$ @1.1V	0.183	N/A	1500	MHz
VCO operating range	F_{VCO}	$F_{VCO} = F_{ref} * NF$① @1.1V	300	N/A	1500	MHz
Output clock frequency	F_{out}	$F_{out} = F_{VCO} / NO$① @1.1V	18.75	N/A	1500	MHz
Lock time	T_{lt}	(NR * 500)/ F_{in} @1.1V	N/A	N/A	N/A	N/A
Power consumption	N/A	F_{out} = 750MHz, NO = 1 @1.1V	N/A	3	N/A	mA

Notes： ①: *NR is the input divider value;*
NF is the feedback divider value;
NO is the output divider value

1.5.7 Electrical Characteristics for SAR-ADC

Table 1-18 RK30xx Electrical Characteristics for SAR-ADC

Parameters	Symbol	Test condition	Min	Typ	Max	Units
ADC resolution	N		N/A	10	N/A	bits
Analog Supply Voltage	VDDIO		2.25	2.5	2.75	V
Digital Supply Voltage	VDDCore		1.0	1.1	1.2	V
Conversion speed	F_S	The duty cycle should be between 40%~60%	N/A	N/A	1	MSPS
Analog Supply Current	IAVDD		N/A	200	N/A	uA
Digital Supply Current	IDVDD		N/A	50	N/A	uA
Number of Channels	NCH AVDD		N/A	8	N/A	N/A
Differential Non Linearity	DNL		N/A	±1	N/A	LSB
Integral Non Linearity	INL		N/A	±2	N/A	LSB
Gain Error	E_{gain}		-8	N/A	8	LSB
Offset Error	E_{offset}		-8	N/A	8	mV
Power Down Current	ISHDN	From AVDD	N/A	0.5	N/A	uA
		From DVDD	N/A	0.5	N/A	uA
Power up time			N/A	7	N/A	$1/F_S$

1.5.8 Electrical Characteristics for TS-ADC

Table 1-19 RK30xx Electrical Characteristics for TS-ADC

Parameters	Symbol	Test condition	Min	Typ	Max	Units
Analog Supply Voltage	AVDD		2.25	2.5	2.75	V
Digital Supply Voltage	DVDD		1.0	1.1	1.2	V
TSADC Accuracy			N/A	N/A	±5	C
Quiescent Current	IQ	From AVDD	N/A	180	N/A	uA
		From DVDD	N/A	40	N/A	uA
Power Down Current	ISHDN	From AVDD	N/A	1	N/A	uA
		From DVDD	N/A	5	N/A	uA

Number of Channels (Differential)		7 external, 1 internal	N/A	8	N/A	N/A
Clock Frequency	F_{clk}		N/A	N/A	50.0	KHz
Power up time			N/A	N/A	7	TCLK
Latency			N/A	N/A	1	TCLK

1.5.9 Electrical Characteristics for USB OTG/Host2.0 Interface

Table 1-20 RK30xx Electrical Characteristics for USB OTG/Host2.0 Interface

Parameters		Test condition	Min	Typ	Max	Units
HS transmit, maximum transition density (all 0's data in DP/DM)	Current From OTG_DVDD	75℃ , OTG_VDD25 = HOST_VDD25 = 2.5V, OTG_VDD33 = HOST_VDD33 = 3.3V, OTG_DVDD = HOST_DVDD = 1.1V , 15-cm USB cable attached to DP/DM	N/A	5.35	N/A	mA
	Current From OTG_VDD33		N/A	2.50	N/A	mA
	Current From OTG_VDD25		N/A	21.2	N/A	mA
HS transmit, minimum transition density (all 1's data in DP/DM)	Current From OTG_DVDD		N/A	4.07	N/A	mA
	Current From OTG_VDD33		N/A	2.25	N/A	mA
	Current From OTG_VDD25		N/A	17	N/A	mA
HS idle mode	Current From OTG_DVDD		N/A	5.4	N/A	mA
	Current From OTG_VDD33		N/A	2.22	N/A	mA
	Current From OTG_VDD25		N/A	6.23	N/A	mA
FS transmit, maximum transition density (all 0's data in DP/DM)	Current From OTG_DVDD		N/A	3.25	N/A	mA
	Current From OTG_VDD33		N/A	16.5	N/A	mA
	Current From OTG_VDD25		N/A	6.47	N/A	mA
LS transmit, maximum transition density (all 0's data in DP/DM)	Current From OTG_DVDD		N/A	3.62	N/A	mA
	Current From OTG_VDD33		N/A	16.9	N/A	mA
	Current From OTG_VDD25		N/A	6.39	N/A	mA
Suspend mode	Current From OTG_DVDD		N/A	61.2	N/A	uA
	Current From OTG_VDD33		N/A	0.1	N/A	uA
	Current From OTG_VDD25		N/A	17.0	N/A	uA
Sleep mode	Current From OTG_DVDD		N/A	0.2	N/A	mA
	Current From OTG_VDD33		N/A	0.1	N/A	uA
	Current From OTG_VDD25		N/A	0.348	N/A	mA

1.5.10 Electrical Characteristics for HDMI

Table 1-21 RK30xx Electrical Characteristics for HDMI

Parameters	Symbol	Test condition	Min	Typ	Max	Units
2.5V Supply Voltage	VDDH	2.5±10%	2.25	2.5	2.75	V
1.1V Supply Voltage	VDDL	1.1±10%	0.99	1.1	1.21	V
REXT Resistance Error	DREXT	470ohms	-1	0	1	%
D2-0[9:0] Setup Time	TSU	To TMDS_CK	700	N/A	N/A	ps
D2-0[9:0] Hold Time	THLD	To TMDS_CK	700	N/A	N/A	ps
MSENS detect voltage for monitor connection	VCNCT	(VTXC+VTXC_N) / 2	2	N/A	N/A	V
MSENS detect voltage for monitor disconnection	VDISC	(VTXC+VTXC_N) / 2	N/A	N/A	0.4	V

MSENS response time	TMSENS		3	N/A	15	us
OSC minimum oscillation frequency when f(IDCK)=0Hz	FMINDDC	IDCK=L	20	30	40	MHz
Supply Current (2.5V)	IDDH		N/A	N/A	TBD	mA
Supply Current (1.1V)	IDDL		N/A	N/A	TBD	mA
Activation Time From Sleep	TACT		N/A	N/A	1	ms

1.5.11 Electrical Characteristics for DDR IO

Table 1-22 RK30xx Electrical Characteristics for DDR IO

Parameters		Symbol	Test condition	Min	Typ	Max	Units
DDR IO @DDR3 mode	DDR IO power standby current, ODT OFF		@ 1.5V , 125℃	N/A	0.02	14.47	uA
	Input leakage current, SSTL mode, unterminated		@ 1.5V , 125℃	N/A	0.02	5.06	uA
DDR IO @LPDDR2 mode	Input leakage current		@ 1.2V , 125℃	N/A	0.01	4.51	uA
	VDD quiescent current		@ 1.1V , 125℃	N/A	0.02	4.21	uA
	DDR IO power quiescent current		@ 1.2V , 125℃	N/A	0.02	12.31	uA

1.5.12 Electrical Characteristics for eFuse

Table 1-23 RK30xx Electrical Characteristics for eFuse

	Parameters	Symbol	Test condition	Min	Typ	Max	Units
Active mode	read current for eFuse digital core logic(1.1V)	I_{load_vdd}	STROBE high	2.537	4.049	5.695	mA
	read current for eFuse digital core logic (1.1V)	I_{active_vdd}	normal read 10MHz	1.498	2.368	3.308	mA
standby mode	standby current for eFuse digital core logic (1.1V)	$I_{standby\_vdd}$		0.057	0.255	0.492	uA

1.6 Hardware Guideline

1.6.1 Reference design for RK30xx oscillator PCB connection

RK30xx only use one oscillator, and its typical clock frequency is 24MHz. The oscillator will provide input clock to four on-chip PLLs.

- External reference circuit for oscillators with 24MHz input

In the following diagram , the value for Rf,Rd,C1,C2 must be adjusted a little to improve performance of oscillator based on real crystal model . Especially C1 and C2 value is advised to meet formula (C1 * C2)/(C1+C2) = ~8pF

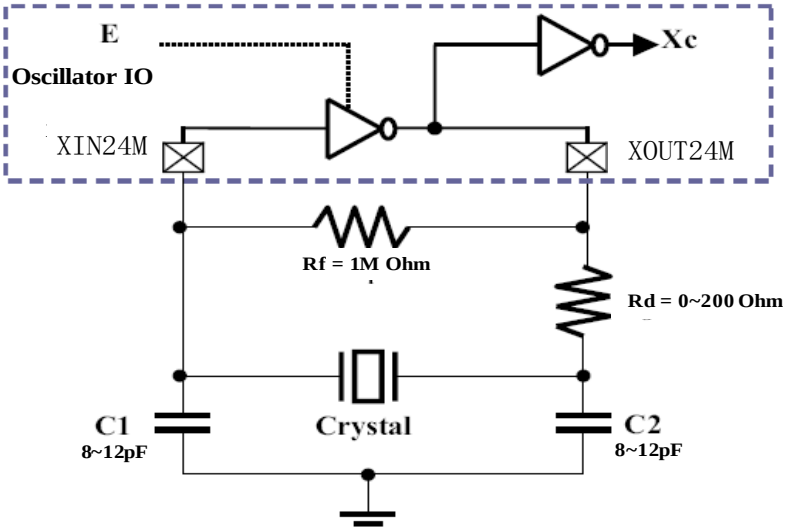


Fig. 1-7 External Reference Circuit for 24MHzOscillators

1.6.2 Reference design for PLL PCB connection

The following reference design is suitable for PLL in RK30xx.

The PLL's two analog supplies should be filtered with two series ferritebeads and two shunt 0.1uF and 0.01uF capacitors. The ferrite on VSS ispreferred but optional. Adding the ferrite on VSS converts supply noiseto substrate noise as seen by the PLL. The PLLs are designed to berelatively insensitive to supply and substrate noise, so the presenceof this ferrite is a second order issue.

The VDD/VSS is mapped to VDD_APLL/VSS_APLL, VDD_DPLL/VSS_DPLL and VDD_CGPLL/VSS_CGPLL.

The AVDD/AVSS is mapped to AVDD_APLL/AVSS_APLL, AVDD_DPLL/ AVSS_DPLL and AVDD_CGPLL/AVSS_CGPLL.

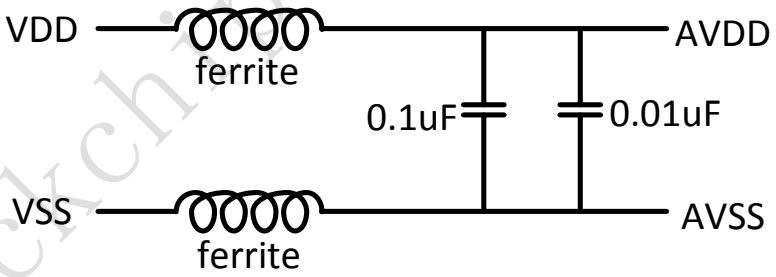


Fig. 1-8 External reference circuit for PLL

The ferrite beads should be similar one of the following from Murata:

Table 1-24 Ferrite Bead Selection

Part number	R@DC	Z@10MHz	Z@100MHz	size
BLM18EG601SN1	0.35	200	600	0603
BLM18PG471SN1	0.2	130	470	0603
BLM18KG601SN1	0.15	160	600	0603
BLM18AG601SN1	0.38	180	600	0603
BLM18AG102SN1	0.5	280	1000	0603

BLM18TG601TN1	0.45	190	600	0603
BLM15AG601SN1	0.6	200	600	0402
BLM15AX601SN1	0.34	190	600	0402
BLM15AX102SN1	0.49	250	1000	0402
BLM03AX601SN1	0.85	120	600	0201

Similar ferrite beads are also available from Panasonic. The keycharacteristics to select are:

- DC resistance less than 0.40 ohms
- impedance at 10MHz equal to or greater 180 ohms
- impedance at 100MHz equal to or greater than 600 ohms

The capacitors should be mounted as close to the package balls aspossible.

1.6.3 Reference design for USB OTG/Host2.0 connection

In RK30xx there are USB OTG and USB Host2.0 interface, in fact, same interface is for them.The following diagram shows external reference design. Of course, for USB Host2.0some signals can be removed based on different application.

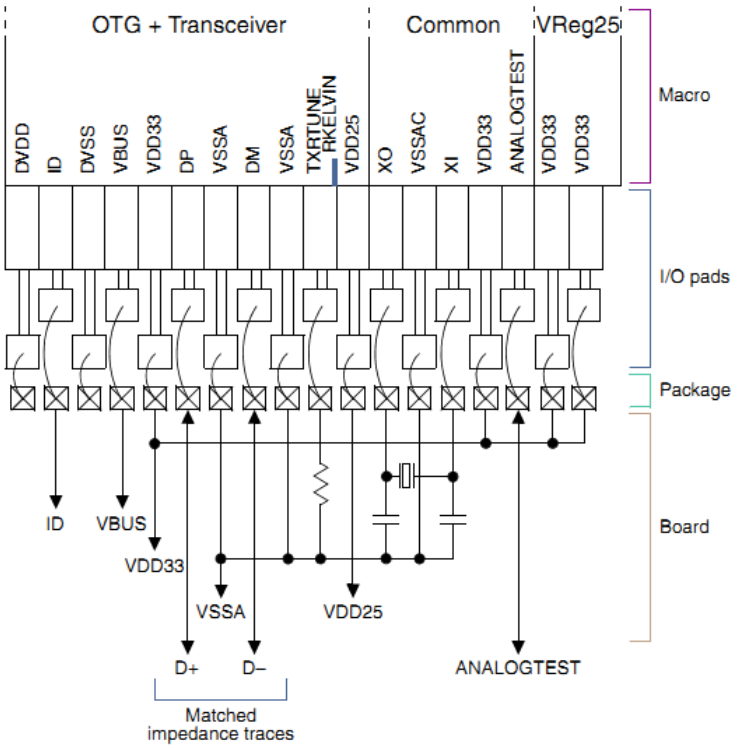
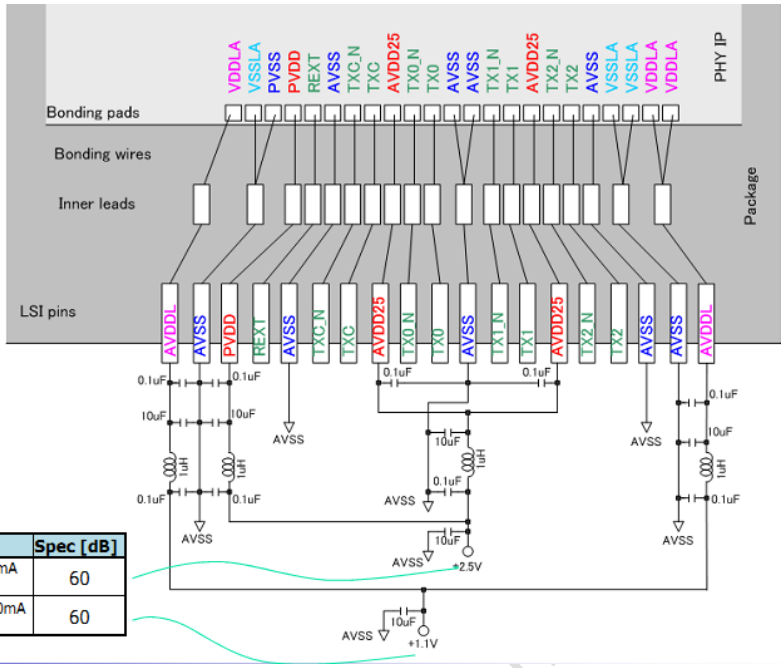


Fig. 1-9 RK30xx USB OTG/Host2.0 interface reference connection

1.6.4 Reference design for HDMI Tx PHY connection

In RK30xx, the following diagram shows external PCB reference design for HDMI Tx PHY.



Following table is the recommended supply power specifications.

PSRR (Ripple Rejection)

Power Supply	Conditions	Spec [dB]
+2.5V Power	Vin = 2.5V, Iload = 20mA f = 1KHz	60
+1.1V Power	Vin = 2.5V, Iload = 100mA f = 1KHz	60

Fig. 1-10RK30xx HDMI interface reference connection

1.6.5 RK30xx Power up/down sequence requirement

For all of the power supply in RK30xx, there is no any specific requirement of power up/down sequence except power supply between core logic and DDR3/LPDDR2 IO or digital GPIO , between USB OTG/Host2.0 power supply .

- Power supply sequence for core logic(CVDD/AVDD/PVDD) and DDR3/LPDDR2 IO (MVDD)

It is generally recommended that the core logic and DDR IO be powered-up together, and it is also acceptable for core logic supply to power-up a very short time before the DDR IO supply. If DDR IO supply must power-up before the core logic supply, it is advised to keep the time between these two events less than 100ms to limit excessive DDR IO current draws.

- Power supply sequence for core logic(CVDD/AVDD/PVDD) and digital GPIO power①

It is generally recommended that "turn on the higher GPIO voltage first and then the lower core voltage" so that the crowbar current would not occur on the power-up stage.

Also it is acceptable that "turn on the lower core voltage first and then higher GPIO voltage" only if the GPIO control pins are set to a fixed state. However, the ramp-up time for them can not be less than 10us.

There is no requirement on the power-down sequence for two above groups. Customers can decide which voltage to be down first based on the application need.

- Power supply sequence for USB OTG/Host2.0

Please follow the following sequence for power up and recommended ramp-up time is more than 10us

OTG_DVDD (1.1V)->OTG_VDD25 (2.5V)->OTG_VDD33 (3.3V)
HOST_DVDD (1.1V)->HOST_VDD25 (2.5V)->HOST_VDD33 (3.3V)

For power down sequence , just reverse with power up sequence .

OTG_VDD33 (3.3V)->OTG_VDD25 (2.5V)->OTG_DVDD (1.1V)
HOST_VDD33 (3.3V)->HOST_VDD25 (2.5V)->HOST_DVDD (1.1V)

Notes :①:digital GPIO power include LCD0_VCCj, LCD1_VCC, CIFj_VCC, PVCC, APj_VCC, SMC_VCC, FLASH_VCC, VCCIOj.

1.6.6 RK30xx Power on reset descriptions

The following figure shows power-on-reset sequence. External power-on-reset input signal NPOR is released after stabilization of oscillator input clock XIN24M. Internal signal sysrstn is generated after NPOR is filtered glitch , which can filter out 5 clock cycles(24MHz) for low pulse of NPOR, so 208ns low pulse of NPOR will not be recognized as valid power-on-reset signal for RK30xx.

To make PLLs work normally, the internal power down signal(pllpd) for PLLs must be high after power-on-reset, and maintains high level for more than 1us after sysrstn is deasserted. Then PLL reset signals(pllrstn) are asserted for about 10.6us, and PLLs start to lock when pllrstn deassert, and consume about to 1330us to lock.

So the system will wait about 1330us, then deactive internal reset signal chiprstn, which is used to control generation logic of all the clock inside CRU.

After 256 cycles or about 10.7us , rstn_pre for reset signal of all internal IPs will be deasserted , in other words, about 10.7us of clock has been generated before reset of every internal module is released.

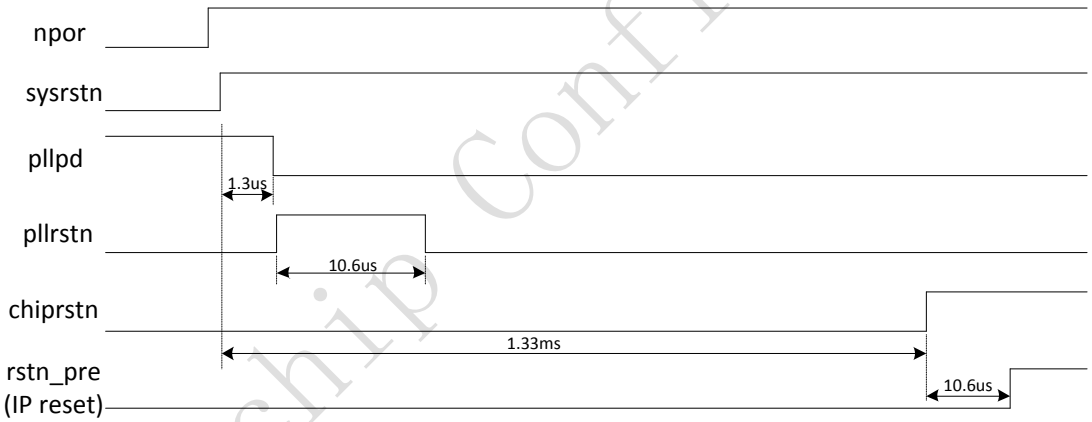


Fig. 1-11 RK30xx reset signals sequence

Chapter 2 System Overview

2.1 Address Mapping

Since RK30xx support to boot from internal bootrom, embedded SRAM or external nor flash , they have two types of address mapping, which is decided by BTMODE off-chip input signal .

Also they always support remap function by software programming. In the following description, remap is value for GRF_SOC_CON0 bit[12] .

- BTMODE=Low Level , remap function is disabled (default state)

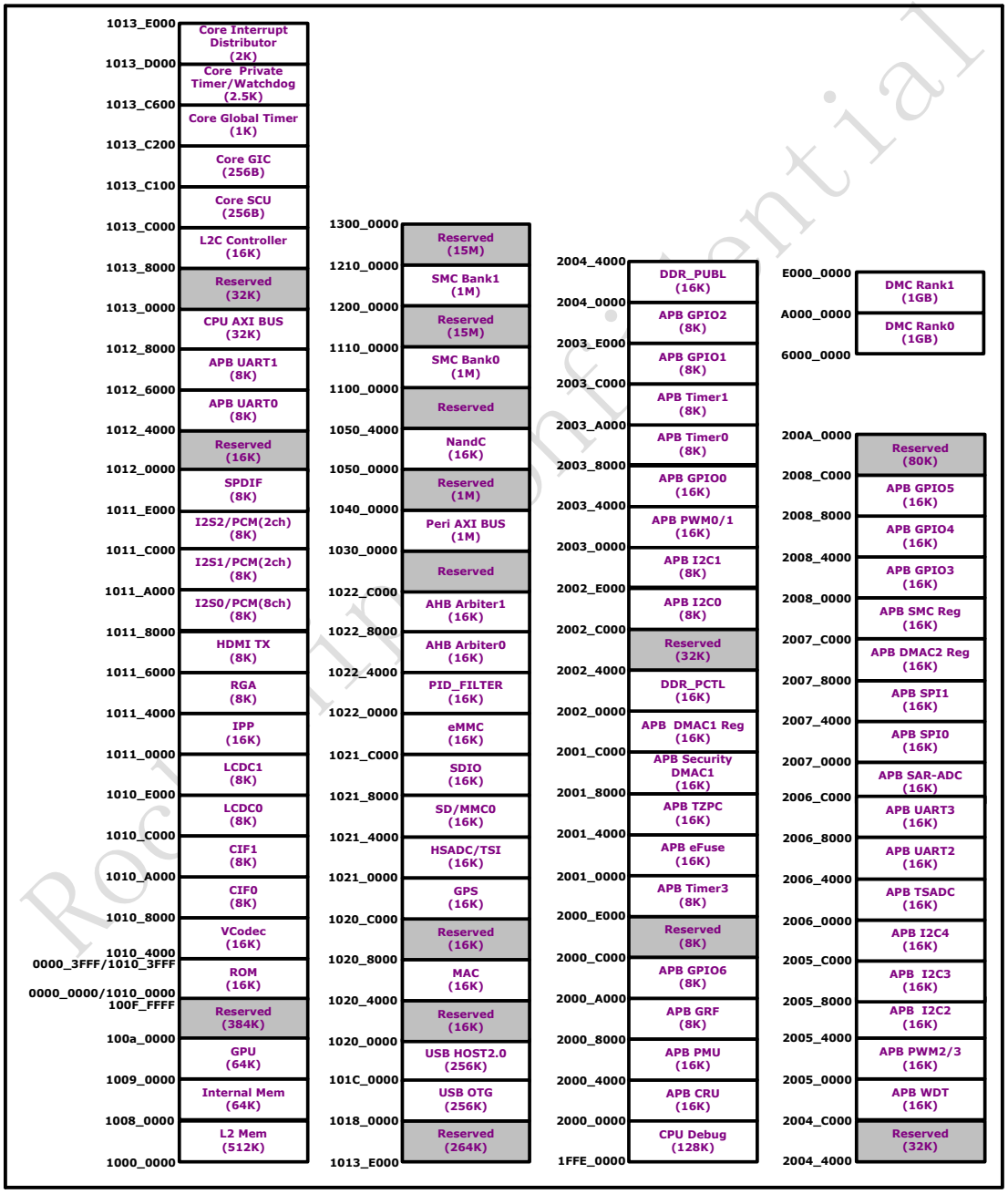


Fig. 2-1 RK30xx Address Mapping when BTMODE=low before remap

- BTMODE=Low Level , remap function is enable

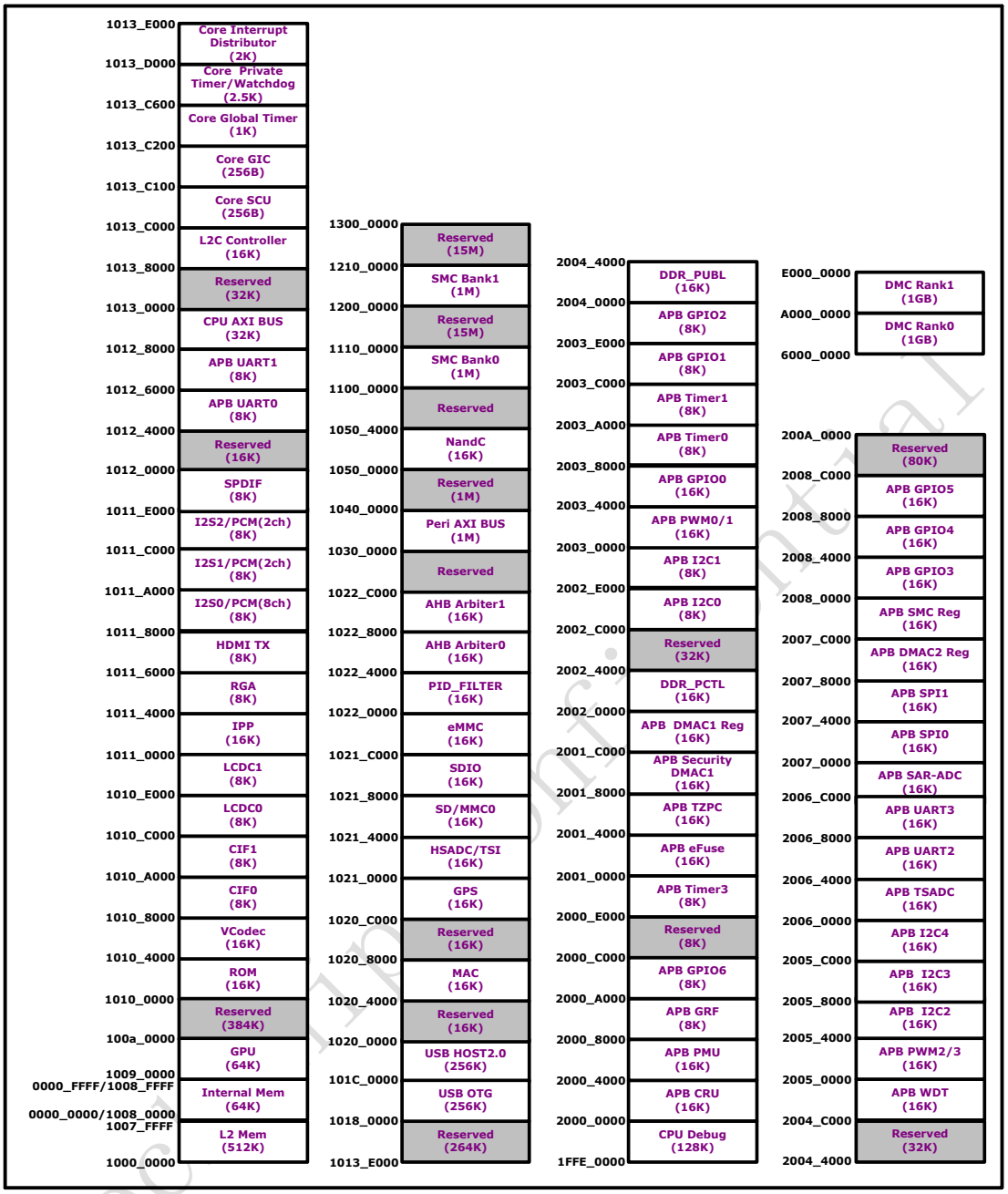


Fig. 2-2 RK30xx Address Mapping when BTMODE=lowafter remap

- BTMODE = High Level

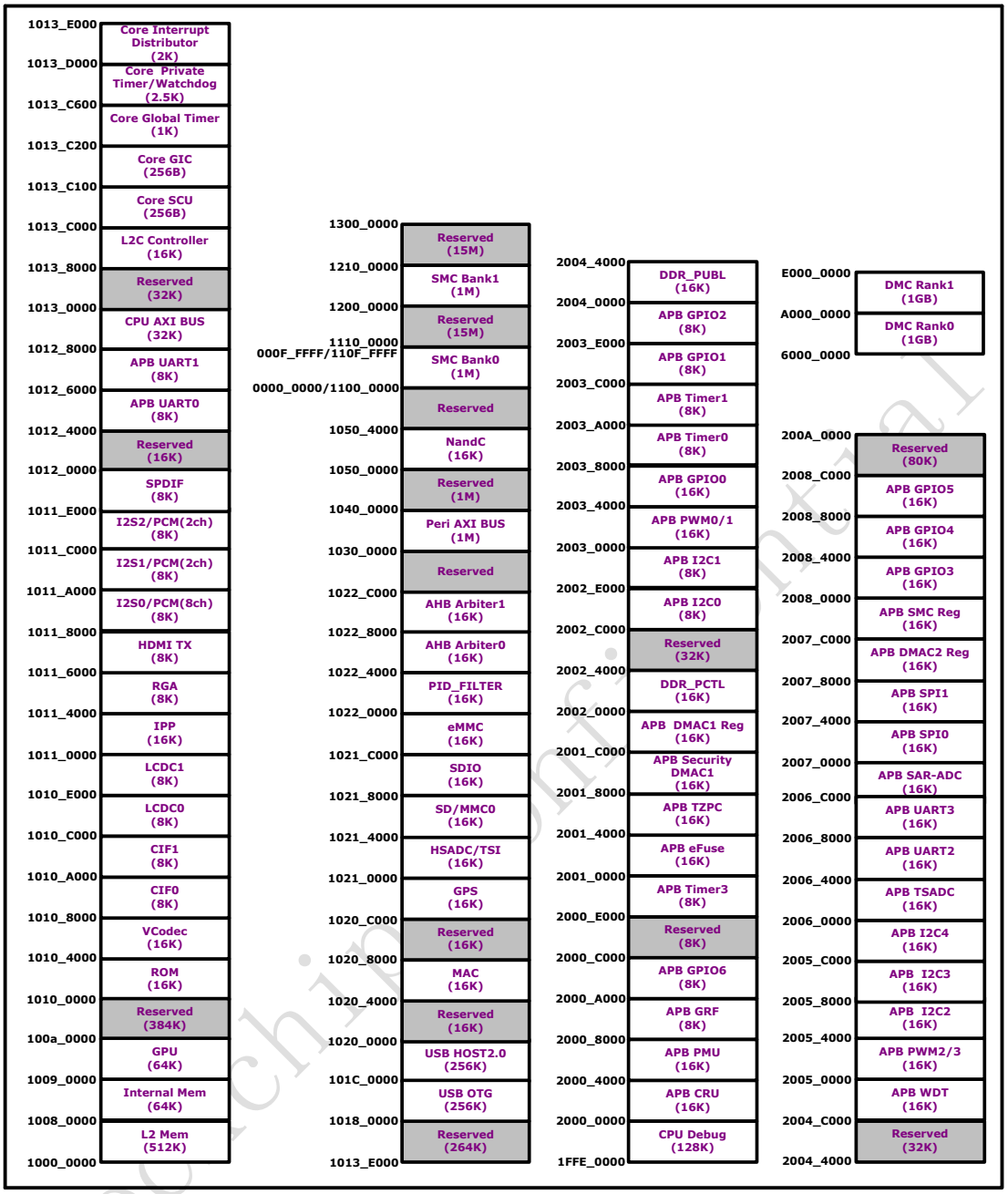


Fig. 2-3 RK30xx Address Mapping when BTMODE=high

2.2 System Boot

RK30xx provides system boot from off-chip devices such as 8bits/16bits async nand flash, spi and emmc memory. When boot code is not ready in these devices, also provide system code download into them by usb otg and uart interface. All of the boot code will be stored in internal boot rom or external 8bits nor flash device, which is decided by input level of external input pin BTMODE. The following is the whole boot procedure for boot code , which will be stored in bootrom or nor flash in advance.

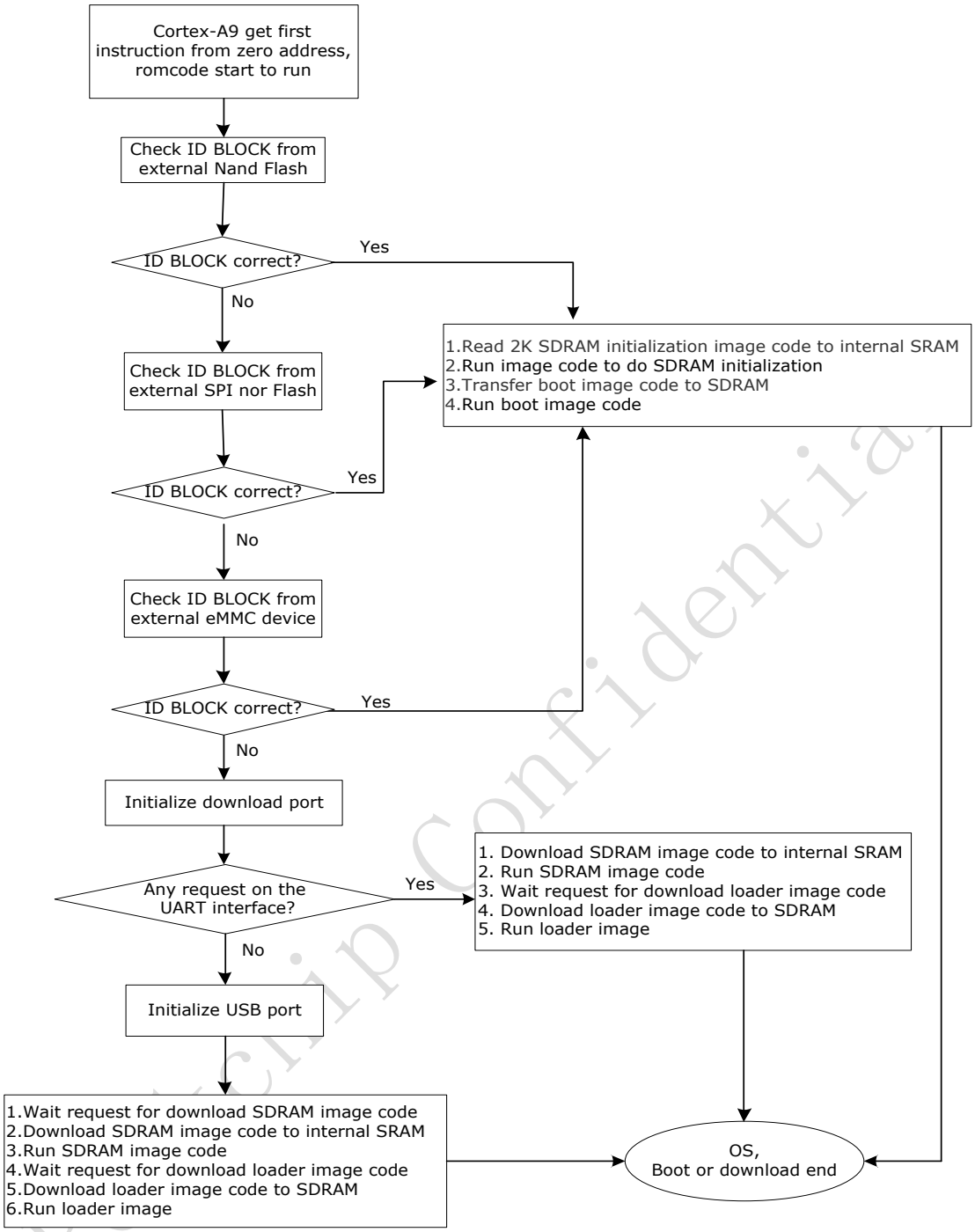


Fig. 2-4 RK30xx boot procedure flow

2.3 System Interrupt connection

RK30xx provides an general interrupt controller(GIC) for Cortex-A9 MPCore processor, which has 76 SPI interrupt sources and 3 PPI interrupt source and separately generates one nIRQ and one nFIQ to CPU. The triggered type for each interrupts is high level sensitive, not programmable. The detailed interrupt sources connection is in the following table. For detailed GIC setting, please refer to Chapter 12 .

Table 2-1 RK30xx Interrupt connection list

IRQ Type	IRQ ID	Source(spi)	Polarity
PPI	27	Golbal Timer	High level
	29	Private Timer	High level
	30	WDT	High level
SPI	32	DMAC1(0)	High level
	33	DMAC1(1)	High level
	34	DMAC2(0)	High level
	35	DMAC2(1)	High level
	36	DDR Controller	High level
	37	GPU(irqgp)	High level
	38	GPU(irqmmu)	High level
	39	GPU(irqpp)	High level
	40	Reserved	High level
	41	Video encoder	High level
	42	Video decoder	High level
	43	Camera IF0	High level
	44	Camera IF1	High level
	45	LCDC0	High level
	46	LCDC1	High level
	47	IPP	High level
	48	USB OTG	High level
	49	USB Host2.0	High level
	50	Reserved	High level
	51	MAC	High level
	52	I2S2/PCM(2ch)	High level
	53	TSADC	High level
	54	HS-ADC/TSI	High level
	55	SD/MMC0	High level
	56	SDIO	High level
	57	eMMC	High level
	58	SAR-ADC	High level
	59	NandC	High level
	60	Reserved	High level
	61	SMC	High level
	62	PID_FILTER	High level
	63	I2S0/PCM (8ch)	High level
	64	I2S1/PCM (2ch)	High level
	65	SPDIF	High level
	66	UART0	High level
	67	UART1	High level
	68	UART2	High level
	69	UART3	High level

	70	SPI0	High level
	71	SPI1	High level
	72	I2C0	High level
	73	I2C1	High level
	74	I2C2	High level
	75	I2C3	High level
	76	Timer0	High level
	77	Timer1	High level
	78	Timer2	High level
	79	PWM0	High level
	80	PWM1	High level
	81	PWM2	High level
	82	PWM3	High level
	83	WDT	High level
	84	I2C4	High level
	85	PMU(int)	High level
	86	GPIO0	High level
	87	GPIO1	High level
	88	GPIO2	High level
	89	GPIO3	High level
	90	GPIO4	High level
	91	Reserved	High level
	92	GPIO6	High level
	93	peri_ahb_usb arbiter	High level
	94	peri_ahb_emem arbiter	High level
	95	RGA	High level
	96	HDMI	High level
	97	Reserved	High level
	98	SD/MMC detect	High level
	99	SDIO detect	High level
	100	gpu_obsrv_mainfault	High level
	101	PMU(stop_exit_int)	High level
	102	observer_mainfault	High level
	103	vpu_obsrv_mainfault	High level
	104	peri_obsrv_mainfault	High level
	105	vio1_obsrv_mainfault	High level
	106	vio0_obsrv_mainfault	High level
	107	dmac_obsrv_mainfault	High level

2.4 System DMA hardware request connection

RK30xx provides 2 DMA controllers : DMAC0 inside cpu system and DMAC1 inside peri system. As for DMAC0, there are 11 hardware request ports . Another, 14 hardware request ports are used in DMAC1, the trigger type for each of them

is high level, not programmable. For detailed descriptions of DMAC0 and DMAC1, please refer to Chapter 10 and Chapter 11 .

Table 2-2 RK30xx DMAC0 Hardware request connection list

Req Number	Source	Polarity
0	Uart0 tx	High level
1	Uart0 rx	High level
2	Uart1 tx	High level
3	Uart1 rx	High level
4	I2S0/PCM(8ch) tx	High level
5	I2S0/PCM(8ch) rx	High level
6	I2S1/PCM(2ch) tx	High level
7	I2S1/PCM(2ch) rx	High level
8	SPDIF tx	High level
9	I2S2/PCM(2ch) tx	High level
10	I2S2/PCM(2ch) rx	High level

Table 2-3 RK30xx DMAC1 Hardware request connection list

Req Number	Source	Polarity
0	HS-ADC/TSI	High level
1	SD/MMC(0)	High level
2	N/A	High level
3	SDIO	High level
4	eMMC	High level
5	PID_FILTER	High level
6	Uart2 tx	High level
7	Uart2 rx	High level
8	Uart3 tx	High level
9	Uart3 rx	High level
10	SPI0 tx	High level
11	SPI0 rx	High level
12	SPI1 tx	High level
13	SPI1 rx	High level

Chapter 3 CRU (Clock & Reset Unit)

3.1 Overview

The CRU is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip. CRU generates system clock from PLL output clock or external clock source, and generate system reset from external power-on-reset, watchdog timer reset or software reset.

CRU supports the following features:

- Compliance to the AMBA APB interface
- Embedded four PLL
- Support only one crystals
- Flexible selection of clock source
- Supports the respectivegating of all clocks
- Supports the respective software reset of all modules

For detailed information about CRU, please refer to **RK30xx CRU.pdf**。

Chapter 4 PMU (Power Management Unit)

4.1 Overview

The PMU focuses on the power on/off switch for different power domain, which support the different system power saved mode to meet the chip high performance and lower power application requirement.

4.1.1 Features

- Support 3 voltage domains
- Support 8 separate power domains, which can be power up/down by software based on different application scenes
- Support seven work modes(normal mode, slow mode, idle mode, deep idle mode, stop mode, sleep mode and power off mode) to save power
- Support idle mode which only Cortex-A9 core clock gated, and wakeup by any interrupt from every on-chip component
- Support deep idle mode which only Cortex-A9 core power off, and wakeup by any interrupt from every on-chip component
- Support stop mode which almost modules clock gated, and wakeup by some periphrals or 16 different GPIOs
- Support sleep mode which the internal power is power off, and wakeup by some periphrals or 16 different GPIOs
- Support PD_SCU domain is externally turned off in sleep mode
- Support power off mode which the internal power is externally turned off, and wakeup by 16 different GPIOs
- Support clock of PD_ALIVE and PD_RTC switch to 32.768kHz optionally in some low power modes
- Support PLLs off in some low power modes
- Support OSC disable optionally in some low power modes
- Support hardware DDR self-refresh optionally in some low power modes
- Support select to boot from SRAM or ROM after wakeup in deep idle mode and sleep mode

For detailed information about PMU, please refer to **RK30xx PMU.pdf**。

Chapter 5 System Security

5.1 cOverview

The RK30xx use the TrustZone access control scheme to support the system security application requirement.

For detailed information about system security, please refer to **RK30xx system security.pdf**。

Chapter 6 System Debug

6.1 Overview

The RK30xx use the CoreSight Technology to support real-time debug access and trace for the multi-core. A standard infrastructure is implemented for the capture and transmission of trace data, combination of multiple data streams by funneling together, and then output of data to a trace port.

6.1.1 Features

- Invasive debug with core halted
- cross-triggering, the ECT provide a standard interconnect mechanism to pass debug or profiling events around the SOC
- Trace, capture and transmission trace data using PTM and TPIU
- Real-time access system memory and peripheral register without halting the CPU，using DAP AHB master

6.1.2 Debug components address map

The following table shows the debug components address in memory map:

Module	Base Address
DAP_ROM	0x1ffe0000
CTI4	0x1ffe1000
TPIU	0x1ffe2000
Trace Funnel	0x1ffe3000
CPUDBG0	0x1ffe4000
CPUPMU0	0x1fff0000
CPUDBG1	0x1fff2000
CPUPMU1	0x1fff3000
CTI0	0x1fff8000
CTI1	0x1fff9000
PTM0	0x1fffc000
PTM1	0x1fffd000

6.2 Block Diagram

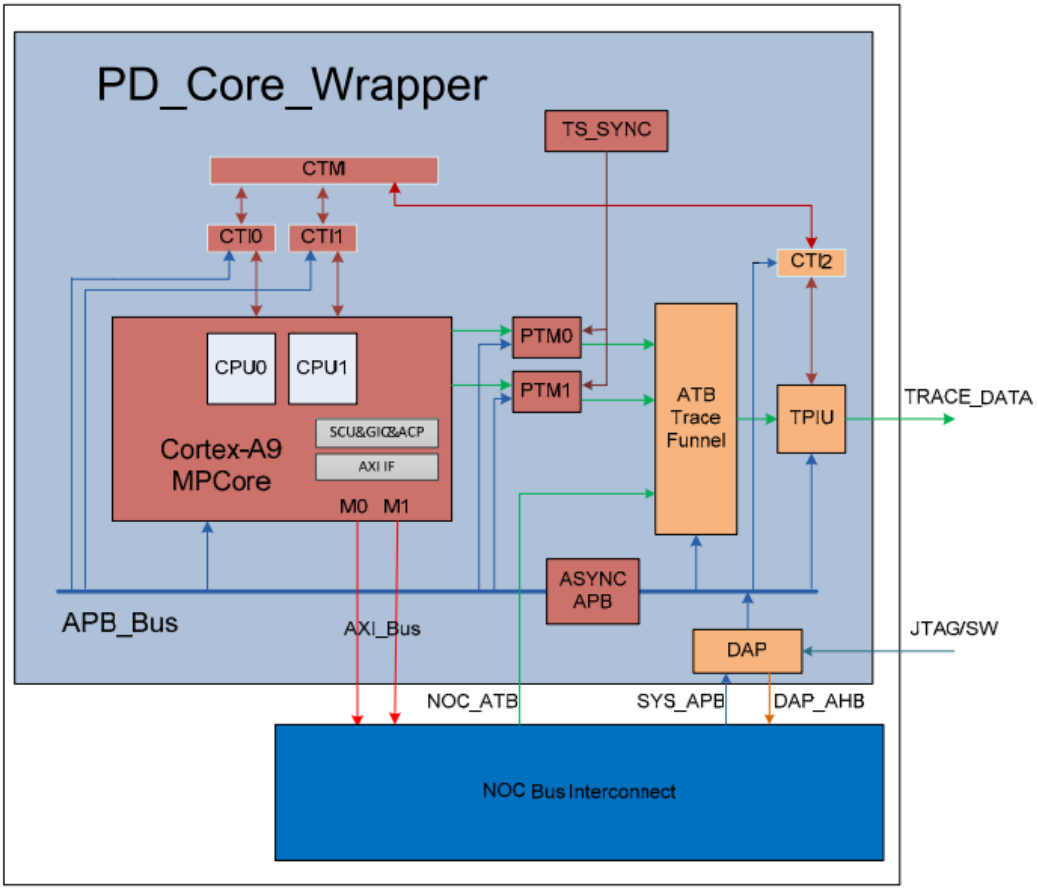


Fig. 6-1 RK30xx Debug system structure

6.3 Function description

6.3.1 DAP

The Debug Access Port (DAP) is an implementation of an ARM Debug Interface version 5.1 (ADIv5.1) comprising a number of components supplied in a single configuration. All the supplied components fit into the various architectural components for Debug Ports (DPs), which are used to access the DAP from an external debugger and Access Ports (APs), to access on-chip system resources.

The RK30xx DAP has following components:

- Serial Wire JTAG Debug Port(SWJ-DP)
- APB Access Port(APB-AP)
- APB-Mux
- AHB Access Port(AHB-AP)
- ROM table

The debug port is the host tools interface to access the DAP-Lit. This interface controls any access ports provided within the DAP-Lite. The DAP-Lite support a combined debug port which includes both JTAG and Serial Wire Debug(SWD), with a mechanism that supports switching between them.

The APB-AP acts as a bridge between SWJ-DP and APB bus which translate the Debug request to APB bus.

The APB-Mux enables external tools and system access to the debug APB. The APB-Mux encapsulates the multiple interface into a single deliverable

component, enable multi-master access to the Debug APB.

The AHB-AP implements the MEM-AP architecture to directly connect to an AHB based memory system. Connection to other memory systems is possible through suitable bridging local.

The DAP provides an internal ROM table connected to the master Debug APB port of the APB-Mux. The Debug ROM table is loaded at address 0x00000000 and 0x80000000 of this bus and is accessible from both APB-AP and the system APB input. Bit [31] of the address bus is not connected to the ROM Table, ensuring that both views read the same value. The ROM table stores the locations of the components on the Debug APB.

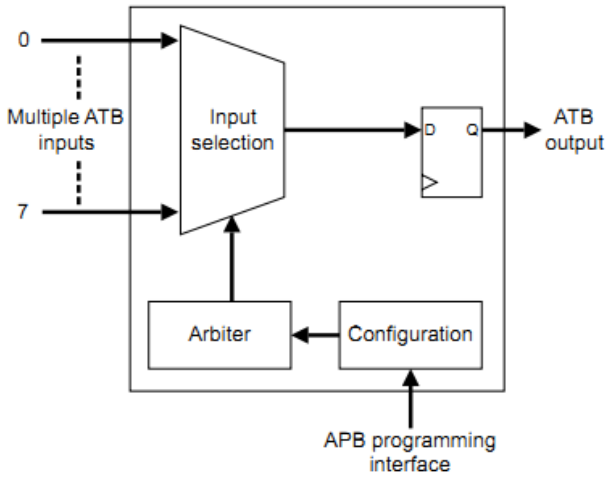
6.3.2 PTM

The PTM is a module that performs real-time instruction flow tracing based on the Program Flow Trace (PFT) architecture. The PTM generates information that trace tools use to reconstruct the execution of all or part of a program.

The PFT architecture assumes that the trace tools can access a copy of the codebeing traced. For this reason, the PTM generates trace only at certain points in program execution, called waypoints. This reduces the amount of trace data generated by the PTM compared to the ETM protocol. Waypoints are changes in the program flow or events, such as an exception. The trace tools use waypoints to follow the flow of program execution.

6.3.3 Trace funnel

The CSTF is used when there is more than one trace source. The CSTF combines multiple trace streams onto a single ATB bus.



6.3.4 TPIU

The TPIU acts as a bridge between the on-chip trace data, with separate IDs, to a data stream, encapsulating IDs where required, that is then captured by a Trace Port Analyzer (TPA). Figure 8-1 shows the main blocks of the TPIU and the clock domains.

The TPIU contains the following components:

- Formatter
  Inserts source ID signals into the data packet stream so that trace data can be re-associated with its trace source. See TPIU formatter and FIFO.
- Asynchronous FIFO

Enables trace data to be driven out at a speed that is not dependent on theon-chip bus clock.

- Register bank

Contains the management, control and status registers for triggers, flushing behavior and external control.

- Trace out

The trace out block serializes formatted data before it goes off-chip.

- Pattern Generator

The pattern generator unit provides a simple set of defined bit sequences or patterns that can be output over the Trace Port and be detected by the TPA or other associated Trace Capture Device (TCD). The TCD can use these patterns to indicate if it is possible to increase or to decrease the trace port clock speed.

- ATB interface

  The TPIU accepts trace data from a trace source, either direct from a trace source or using a Trace Funnel.

- APB interface

  The APB interface is the programming interface for the TPIU.

6.3.5 ECT (CTI & CTM)

The ECT for CoreSight consists of a number of CTIs and CTMs connected together. This enables ARM/ETM subsystems to interact. That is cross trigger, with each other. The debug system enables debug support for multiple cores, together with cross triggering between the cores and their respective ETMs.

The main function of the ECT (CTI and CTM) is to pass debug events from onecore to another. For example, the ECT can communicate debug state information from one core to another, so that program execution on both processors can be stopped at the same time if required.

- CTI (Cross Trigger Interface)

  The CTI combines and maps the trigger requests, and broadcasts them to all other interfaces on the ECT as channel events. When the CTI receives a channel event it maps this onto a trigger output. This enables subsystems to cross trigger with each other. The receiving and transmitting of triggers is performed through the trigger interface.

- CTM (Cross Trigger Matrix)

  This block controls the distribution of channel events. It provides Channel Interfaces (CIs) for connection to either CTIs or CTMs. This enables multiple CTIs to be linked together.

6.4 Register description

6.4.1 DAP APB-AP register summary

Name	Offset	Size	Reset Value	Description
DAP_CSW	0x000	W	0x00000002	Control/Status Word, CSW
DAP_TAR	0x004	W	0x00000000	Transfer Address, TAR
Reserved	0x008	W	NA	Reserved
DAP_DRW	0x00c	W	NA	Data Read/Write, DRW
DAP_BD0	0x010	W	NA	Bank Data 0, BD0
DAP_BD1	0x014	W	NA	Bank Data 1, BD1
DAP_BD2	0x018	W	NA	Bank Data 2, BD2
DAP_BD3	0x01c	W	NA	Bank Data 3, BD3
Reserved	0x20-0xf4	W	NA	Reserved
DAP_ROM_ADDR	0xf8	W	NA	Debug ROM Address, ROM
DAP_IDR	0xfc	W	0x14770002	Identification Register, IDR

6.4.2 DAP APB-AP Detailed Register Description

DAP_CSW

Address: APBAP_BASE + offset(0x000)

Control/Status Word

Bits	Attr	Reset Value	Description
31	RW	0x0	Software access enable. Drives DBGSWENABLE to enable or disable software access to the Debug APBbus in the APB multiplexor. b1 = Enable software access b0 = Disable software access. Reset value = b0. On exit from reset, defaults to b1 to enable software access.
31:12	RW	0x0	Reserved
11:8	R	0x0	Specifies the mode of operation. b0000 = Normal download/upload model b0001-b1111 = Reserved Reset value = b0000.
7	R	0x0	Transfer in progress. This field indicates if a transfer is currently in progress on theAPB master port.
6	R	0x0	Transfer in progress. This field indicates if a transfer is currently in progress on the APB master port. Indicates the status of the DEVICEEN input. • If APB-AP is connected to the Debug APB, that is, a bus connected only todebug and trace components, it must be permanently enabled by tyingDEVICEEN HIGH. This ensures that trace components can still beprogrammed when DBGEN is LOW. In practice, it is expected that theAPB-AP is almost always used in this way. • If APB-AP is connected to a system APB dedicated to the non-secure world,DEVICEEN must be connected to DBGEN. • If APB-AP is connected to a system APB dedicated to the secure world,DEVICEEN must be connected to SPIDEN.
5:4	RW	0x0	Auto address increment and packing mode on Read or Write data access. Does not increment if the transaction completes with an error response or the transaction isaborted.Auto address incrementing is not performed on access to banked data registers0x10-0x1C.The status of these bits is ignored in these cases. b11 = Reserved b10 = Reserved b01 = Increment b00 = Auto increment OFF. Increment occurs in word steps.

			Reset value = b00.
3	R	0x0	Reserved
2:0	R	0x2	Size of the access to perform. Fixed at b010 = 32 bits. Reset value = b010.

DAP_TAR
Address: APBAP_BASE + offset(0x004)
Transfer Address

Bits	Attr	Reset Value	Description
31:2	RW	0x0	Address[31:2] of the current transfer PADDR[31:2]=TAR[31:2] for accesses from Data Read/Write Register at 0x0C. PADDR[31:2]=TAR[31:4]+DAPADDR[3:2] for accesses from Banked Data Registers at 0x10-0x1C and 0x0C.
1:0	R	0x0	Reserved

DAP_DRW
Address: APBAP_BASE + offset(0x00c)
Data Read/Write

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Write mode: Data value to write for the current transfer. Read mode: Data value read from the current transfer.

DAP_BD0-DAP_BD3
Address: APBAP_BASE + offset(0x010) - APBAP_BASE + offset(0x01c)
Bank Data 0-3

Bits	Attr	Reset Value	Description
31:0	RW	0x0	If DAPADDR[7:4] = 0x0001, so accessing APB-AP registers in the range 0x10-0x1C, then thederived PADDR[31:0] is: • Write mode: Data value to write for the current transfer to external address TAR[31:4]+ DAPADDR[3:2] + 2'b00. • Read mode: Data value read from the current transfer from external address TAR[31:4]+ DAPADDR[3:2] + 2'b00. Auto address incrementing is not performed on DAP accesses to BD0-BD3. Reset value = 0x00000000

DAP_ROM_ADDR
Address: 0xf8
ROM address

Bits	Attr	Reset Value	Description
31:12	R	0x800000	Base address of the ROM Table The ROM provides a look-up table of allCoreSight Debug APB components. Read only. Set to 0xFFFFF if no ROM ispresent. In the initial CoreSight release this must

			be set to 0x80000.
11:0	R	0x000	Set to 0x000 if ROM is present. Set to 0xFFF if ROM table is not present. Inthe initial CoreSight release this must be set to 0x000.

DAP_IDR
Address: APBAP_BASE + offset(0x0fc)

Bits	Attr	Reset Value	Description
31:28	R	0x1	Revision. Reset value is 0x1 for APB-AP.
27:24	R	0x4	JEDEC bank. 0x4 indicates ARM Limited.
23:17	R	0x3b	JEDEC code. 0x3B indicates ARM Limited.
16	R	0x1	Memory AP. 0x1 indicates a standard register map is used.
15:8	R	0x00	Reserved
7:0	R	0x02	Identity value. Reset value is 0x02 for APB-AP.

6.4.3 DAP AHB-AP register summary

Name	Offset	Size	Reset Value	Description
DAP_AHB_CSW	0x000	W	0x00000002	Control/Status Word, CSW
DAP_AHB_TAR	0x004	W	0x00000000	Transfer Address, TAR
Reserved	0x008	W	NA	Reserved
DAP_AHB_DRW	0x00c	W	NA	Data Read/Write, DRW
DAP_AHB_BD0	0x010	W	NA	Bank Data 0, BD0
DAP_AHB_BD1	0x014	W	NA	Bank Data 1, BD1
DAP_AHB_BD2	0x018	W	NA	Bank Data 2, BD2
DAP_AHB_BD3	0x01c	W	NA	Bank Data 3, BD3
Reserved	0x20-0xf7	W	NA	Reserved
DAP_DEBUG_ROM	0xf8	W	NA	Debug ROM table
DAP_AHB_IDR	0xfc	W	0x14770002	Identification Register, IDR

6.4.4 DAP AHB-AP Detailed Register Description

DAP_AHB_CSW
Address: AHBAP_BASE + offset(0x000)
Control/Status Word

Bits	Attr	Reset Value	Description
31	-	-	Reserved
30	RW	0x0	Specifies that a secure transfer is requested. SProt HIGH indicates a non-secure transfer. SProt LOW indicates a secure transfer. • If this bit is LOW, and SPIDEN is HIGH, HPROT[6] is asserted LOW on an AHB transfer. • If this bit is LOW, and SPIDEN is LOW, HPROT[6] is asserted HIGH and the AHB transfer is not initiated. • If this bit is HIGH, the state of SPIDEN is ignored. HPROT[6] is HIGH.

			Reset value = b1. Non-secure
29	-	-	Reserved
28:24	RW	0x0	Specifies the protection signal encoding to be output on HPROT[4:0].
23	RO	0x0	Indicates the status of the SPIDEN port. If SPIStatus is LOW, no secure AHB transfers are carried out.
22:12	-	-	Reserved
11:8	RW	0x0	Specifies the mode of operation. b0000 = Normal download/upload model b0001-b1111 = Reserved
7	RO	0x0	Transfer in progress. This field indicates if a transfer is currently in progress on the AHB master port
6	RO	0x0	Indicates the status of the DBGEN port. If DbgStatus is LOW, no AHB transfers are carried out. 1 = AHB transfers permitted. 0 = AHB transfers not permitted.
5:4	RW	0x0	Auto address increment and packing mode on Read or Write data access. Only increments if the current transaction completes without an Error response and the transaction is not aborted. Auto address incrementing and packed transfers are not performed on access to Banked Data registers 0x10-0x1C. The status of these bits is ignored in these cases. Increments and wraps within a 1KB address boundary, for example, for word incrementing from 0x1400-0x17FC. If the start is at 0x14A0, then the counter increments to 0x17FC, wraps to 0x1400, then continues incrementing to 0x149C. b00 = Auto increment OFF. b01 = Increment, single. Single transfer from corresponding byte lane. b10 = Increment, packed Word = Same effect as single increment. Byte/Halfword: Packs four 8-bit transfers or two 16-bit transfers into a 32-bit DAP transfer. Multiple transactions are carried out on the AHB interface. b11 = Reserved SBZ, no transfer. Size of address increment is defined by the Size field, bits [2:0].
3	-	-	Reserved
2:0	RW	0x2	Size of the data access to perform: b000 = 8 bits b001 = 16 bits b010 = 32 bits b011-b111 = Reserved

DAP_AHB_TAR
Address: AHBAP_BASE + offset(0x004)
Control/Status Word

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Address of the current transfer.

DAP_AHB_DRW
Address: AHBAP_BASE + offset(0x00c)
Control/Status Word

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Write mode: Data value to write for the current transfer. Read mode: Data value read from the current transfer.

DAP_AHB_BD0- DAP_AHB_BD3
Address: APBAP_BASE + offset(0x010) - APBAP_BASE + offset(0x01c)
Bank Data 0-3

Bits	Attr	Reset Value	Description
31:0	RW	0x0	If DAPADDR[7:4] = 0x0001, so accessing AHB-AP registers in the range 0x10-0x1C, then thederived HADDR[31:0] is: • Write mode: Data value to write for the current transfer to external address TAR[31:4]+ DAPADDR[3:2] + 2'b00. • Read mode: Data value read from the current transfer from external address TAR[31:4]+ DAPADDR[3:2] + 2'b00. Auto address incrementing is not performed on DAP accesses to BD0-BD3. Banked transfers are only supported for word transfers. Non-word banked transfers are reserved and unpredictable. Transfer size is currently ignored for banked transfers

DAP_DEBUG_ROM
Address: 0xf8
ROM address

Bits	Attr	Reset Value	Description
31:0	RO	-	Base address of a ROM table. The ROM provides a look-up table for system components. Set to 0xFFFFFFFF in the AHB-AP in the initial release

DAP_AHB_IDR
Address: APBAP_BASE + offset(0x0fc)

Bits	Attr	Reset Value	Description
31:28	R	0x4	Revision. Reset value is 0x4 for AHB-AP.
27:24	R	0x4	JEDEC bank. 0x4 indicates ARM Limited.
23:17	R	0x3b	JEDEC code. 0x3B indicates ARM Limited.
16	R	0x1	Memory AP. 0x1 indicates a standard register map is used.
15:8	R	0x00	Reserved

7:0	R	0x01	Identity value. Reset value is 0x01 for AHB-AP.

6.4.5 DAP-ROM register summary

Name	Offset	Size	Reset Value	Description
DAP_ROMENTRY0	0x0000	W	0x00001003	CTI4 entry register
DAP_ROMENTRY1	0x0004	W	0x00002003	TPIU entry register
DAP_ROMENTRY2	0x0008	W	0x00003003	Trace Funnel register
DAP_ROMENTRY3	0x000c	W	0x00004003	Cortex-A9 ROM entry register
DAP_ROM_PERIPHID4	0x0fd0	W	0x00000004	Peripheral ID4
DAP_ROM_PERIPHID5	0x0fd4	W	0x00000000	Peripheral ID5
DAP_ROM_PERIPHID6	0x0fd8	W	0x00000000	Peripheral ID6
DAP_ROM_PERIPHID7	0x0fdc	W	0x00000000	Peripheral ID7
DAP_ROM_PERIPHID0	0x0fe0	W	0x000000c4	Peripheral ID0
DAP_ROM_PERIPHID1	0x0fe4	W	0x000000b4	Peripheral ID1
DAP_ROM_PERIPHID2	0x0fe8	W	0x0000006b	Peripheral ID2
DAP_ROM_PERIPHID3	0x0fec	W	0x00000020	Peripheral ID3
DAP_ROM_COMPONID0	0x0ff0	W	0x0000000d	Component ID0
DAP_ROM_COMPONID1	0x0ff4	W	0x00000010	Component ID1
DAP_ROM_COMPONID2	0x0ff8	W	0x00000005	Component ID2
DAP_ROM_COMPONID3	0x0ffc	W	0x000000b1	Component ID3

6.4.6 DAP-ROM Detailed Register Description

DAP_ROMENTRY0
Address: DAPROM_BASE + offset(0x0000)
TPIU entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00001003	TPIU entry register

DAP_ROMENTRY1
Address: DAPROM_BASE + offset(0x0004)
Cortex-A9 Debug entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00002003	Cortex-A9 Debug entry register

DAP_ROMENTRY2
Address: DAPROM_BASE + offset(0x0008)
Cortex-A9 ETM entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00003003	Cortex-A9 ETM entry register

DAP_ROMENTRY3
Address: DAPROM_BASE + offset(0x000c)
Cortex-A9 CTI entry register

Bits	Attr	Reset Value	Description
31:0	R	0x00004003	Cortex-A9 CTI entry register

DAP_ROM_PERIPHID4
Address: DAPROM_BASE + offset(0x0fd0)
Peripheral ID4

Bits	Attr	Reset Value	Description
31:0	R	0x00000004	Peripheral ID4

DAP_ROM_PERIPHID5
Address: DAPROM_BASE + offset(0x0fd4)
Peripheral ID5

Bits	Attr	Reset Value	Description
31:0	R	0x00000000	Peripheral ID5

DAP_ROM_PERIPHID6
Address: DAPROM_BASE + offset(0x0fd8)
Peripheral ID6

Bits	Attr	Reset Value	Description
31:0	R	0x00000000	Peripheral ID6

DAP_ROM_PERIPHID7
Address: DAPROM_BASE + offset(0x0fdc)
Peripheral ID7

Bits	Attr	Reset Value	Description
31:0	R	0x00000000	Peripheral ID7

DAP_ROM_PERIPHID0
Address: DAPROM_BASE + offset(0x0fe0)
Peripheral ID0

Bits	Attr	Reset Value	Description
31:0	R	0x000000c4	Peripheral ID0

DAP_ROM_PERIPHID1
Address: DAPROM_BASE + offset(0x0fe4)
Peripheral ID1

Bits	Attr	Reset Value	Description
31:0	R	0x000000b4	Peripheral ID1

DAP_ROM_PERIPHID2
Address: DAPROM_BASE + offset(0x0fe8)
Peripheral ID2

Bits	Attr	Reset Value	Description
31:0	R	0x0000006b	Peripheral ID2

DAP_ROM_PERIPHID3
Address: DAPROM_BASE + offset(0x0fec)
Peripheral ID3

Bits	Attr	Reset Value	Description
31:0	R	0x00000020	Peripheral ID3

DAP_ROM_COMPONID0

Address: DAPROM_BASE + offset(0x0ff0)
Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x0000000d	Component ID0

DAP_ROM_COMPONID1
Address: DAPROM_BASE + offset(0x0ff4)
Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x00000010	Component ID1

DAP_ROM_COMPONID2
Address: DAPROM_BASE + offset(0x0ff8)
Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x00000005	Component ID2

DAP_ROM_COMPONID3
Address: DAPROM_BASE + offset(0x0ffc)
Component ID0

Bits	Attr	Reset Value	Description
31:0	R	0x000000b1	Component ID3

6.4.7 PTM register summary

Name	Offset	Size	Reset Value	Description
PTM_ETMCR	0x0	W	0x401	Main control
PTM_ETMCCR	0x4	W	0x8d294004	Configuration code
PTM_ETMTE	0x8	W	0x0	Trigger event
PTM_ETMSR	0x10	W	0x0	Status
PTM_ETMSCR	0x14	W	0x0	System configuration
PTM_ETMTSSCR	0x18	W	0x0	TraceEnable Start/Stop control
PTM_ETMTEE	0x20	W	0x0	TraceEnable event
PTM_ETMTECR1	0x24	W	0x0	TraceEnable Control
PTM_ETMACVR1-8	0x40-0x5c	W	0x0	Address comparator value
PTM_ETMACTR1-8	0x80-0x9c	W	0x0	Address comparator access type
PTM_ETMCNTRLDVR1-2	0x140-0x144	W	0x0	Counter load value
PTM_ETMCNTENR1-2	0x150-0x154	W	0x0	Counter enable
PTM_ETMCNTRLDEVR1-2	0x160-0x164	W	0x0	Counter reload event
PTM_ETMCNTVR1-2	0x170-0x174	W	0x0	Counter value
PTM_SSTE1-SSTE6	0x180-0x194	W	0x0	Sequencer status transition event
PTM_CSS	0x19c	W	0x0	Current sequencer state

PTM_EOE1-PTM_EOE2	0x1a0-0x1a4	W	0x0	External output event
PTM_CICV1	0x1b0	W	0x0	Context ID comparator value
PTM_CICM	0x1bc	W	0x0	Context ID comparator mask
PTM_ETMSYNCFR	0x1e0	W	0x0	Synchronization frequency
PTM_ETMIDR	0x1e4	W	0x411cf301	ID register
PTM_ETMCCER	0x1e8	W	0x00c019a2	Configuration code extension
PTM_ETMEXTINSELR	0x1ec	W	0x0	Extended external input selection
PTM_TE	0x1f8	W	0x0	Timestamp Event
PTM_ETMAUXCR	0x1fc	W	0x0	Auxiliary control register
PTM_ETMTRACEIDR	0x200	W	0x0	CoreSight trace ID
PTM_OSLSR	0x304	W	0x0	OS lock status
PTM_ETMPDSR	0x314	W	0x1	Device power-down status
PTM_ITMISCOUT	0xedc	W	0x0	Miscellaneous outputs
PTM_ITMISCIN	0xee0	W	0x0	Miscellaneous inputs
PTM_ITTRIGGER	0xee8	W	0x0	Trigger register
PTM_ITATBDATA0	0xeec	W	0x0	ATB data 0
PTM_ITATBCTR2	0xef0	W	0x0	ATB control 2
PTM_ITATBID	0xef4	W	0x0	ATB identification
PTM_ITATBCTR0	0xef8	W	0x0	ATB control 0
PTM_ETMITCTRL	0xf00	W	0x0	Integration mode control
PTM_AS	0xfb8	W	0x0	Authentication status
PTM_DC	0xfc8	W	0x0	Device configuration
PTM_DT	0xfcc	W	0x0	Device type
PTM_PID4	0xfd0	W	0x4	Peripheral ID4
PTM_PID5	0xfd4	W	0x0	Peripheral ID5
PTM_PID6	0xfd8	W	0x0	Peripheral ID6
PTM_PID7	0xfdc	W	0x0	Peripheral ID7
PTM_PID0	0xfe0	W	0x50	Peripheral ID0
PTM_PID1	0xfe4	W	0xb9	Peripheral ID1
PTM_PID2	0xfe8	W	0x1b	Peripheral ID2
PTM_PID3	0xfec	W	0x0	Peripheral ID3
PTM_CID0	0xff0	W	0xd	Component ID0
PTM_CID1	0xff4	W	0x90	Component ID1
PTM_CID2	0xff8	W	0x5	Component ID2
PTM_CID3	0xffc	W	0xb1	Component ID3

6.4.8 PTM Detailed Register Description

PTM_ETMCR

Address: PTM_BASE + offset(0x000)

Main Control Register

Bits	Attr	Reset Value	Description
31:30	-	-	Reserved
29	RW	0x0	Return stack enable b0 = disabled b1 = enabled
28	RW	0x0	Timestamp enable b0 = disabled

			b1 = enabled
27:25	RW	0x0	Processor select
24	R	0x0	Reserved
23:16	-	-	Reserved
15:14	RW	0x0	ContextIDSize b00 = no context ID tracing b01 = context ID bits [7:0] traced b10 = context ID bits [15:0] traced b11 = context ID bits [31:0] traced. On reset, this bit is set to b00, no context ID tracing
13	-	-	Reserved
12	RW	0x0	CycleAccurate b0 = cycle counting disabled b1 = cycle counting enabled On reset this bit is set to b0, no cycle counting.
11	-	-	Reserved
10	RW	0x1	Programming Bit This bit must be set to b1 when the PTM is being programmed, see Modes of operation on page 2-3. On a PTM reset this bit is set to b1.
9	RW	0x0	Debug request control When set to b1 and the trigger event occurs, the PTMDBGRQ output is asserted until PTMDBGACK is observed. This enables a debugger to force the processor into Debug state. On PTM reset this bit is set to b0
8	RW	0x0	Branch Output When this bit is set to b1, addresses are output for all executed branches, both direct and indirect. On PTM reset this bit is set to b0.
7	R	0x0	Stall processor
6:1	-	-	Reserved
0	RW	0x1	PowerDown This bit enables external control of the PTM. This bit must be cleared by the trace software tools at the beginning of a debug session. When this bit is set to b0, both the PTM and the trace interface in the processor are enabled. To avoid corruption of trace data, this bit must not be set before the Programming Status bit in the PTM Status Register has been read as 1. On PTM reset this bit is set to b1.

PTM_ETMCCR

Address: PTM_BASE + offset(0x004)

Configuration Code Register

Bits	Attr	Reset Value	Description

31	RO	0x1	ID Register present Indicates that the ID Register is present.
30:28	-	-	Reserved
27	RO	0x1	Software access Indicates that software access is supported
26	RO	0x1	Trace stop/start block Indicates that the trace start/stop block is present.
25:24	RO	0x1	Number of Context ID comparators Specifies the number of Context ID comparators, one.
23	RO	0x0	FIFOFULL logic Indicates that it is not possible to stall the processor to prevent FIFO overflow
22:20	RO	0x2	Number of external outputs Specifies the number of external outputs, two.
19:17	RO	0x4	Number of external inputs Specifies the number of external inputs, four.
16	RO	0x1	Sequencer Indicates that the sequencer is present
15:13	RO	0x2	Number of counters Specifies the number of counters, two.
12:4	-	-	Reserved
3:0	RO	0x4	Number of pairs of address comparators Specifies the number of address comparator pairs, four.

PTM_ETMSCR
Address: PTM_BASE + offset(0x014)
System Configuration Register

Bits	Attr	Reset Value	Description
31:15	-	-	Reserved
14:12	RO	-	Number of supported processors minus 1.The value of this field is set by the MAXCORES[2:0] input to the PTM
11:9	-	-	Reserved
8	RO	-	Read Only, as b0 - FIFOFULL is not supported.
7:0	-	-	Reserved

PTM_ETMTSSCR
Address: PTM_BASE + offset(0x018)
TraceEnable Start/Stop Control Register

Bits	Attr	Reset Value	Description
31:24	-	-	Reserved
23:16	RW	0x0	When a bit is set to 1, it selects a single address comparator (8-1) as a stop address for the TraceEnable Start/Stop block. For example, if you set bit [16] to 1 it selects single address comparator 1 as a stop address.
15:8	-	-	Reserved

7:0	RW	0x0	When a bit is set to 1, it selects a single address comparator (8-1) as a start address for the TraceEnable Start/Stop block. For example, if you set bit [0] to 1 it selects single address comparator 1 as a start address.

PTM_ETMTECR1
Address: PTM_BASE + offset(0x024)
TraceEnable Control Register1

Bits	Attr	Reset Value	Description
31:26	-	-	Reserved
25	RW	0x0	Trace start/stop control enable. The possible values of this bit are: 0 Tracing is unaffected by the trace start/stop logic. 1 Tracing is controlled by the trace on and off addresses configured for the trace start/stop logic. The trace start/stop resource is not affected by the value of this bit.
24	RW	0x0	Exclude/include flag. The possible values of this bit are: 0 Include. The specified address range comparators indicate the regions where tracing can occur. No tracing occurs outside this region. 1 Exclude. The specified address range comparators indicate regions to be excluded from the trace. When outside an exclude region, tracing can occur
23:4	-	-	Reserved
3:0	RW	0x0	When a bit is set to 1, it selects an address range comparator, 4-1, for include/exclude control. For example, bit [0] set to 1 selects address range comparator 1

PTM_ETMACVR1-8
Address: PTM_BASE + offset(0x040-0x05c)
ETMACVR1-8

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Address comparator value register

PTM_ETMACTR1-8
Address: PTM_BASE + offset(0x080-0x09c)
ETMACTR1-8

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Address comparator access type register

PTM_ETMCNTRLDVR1-2
Address: PTM_BASE + offset(0x140-0x144)
ETMCNTRLDVR1-2

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Reload value

PTM_ETMCNTENR1-2
Address: PTM_BASE + offset(0x150-0x154)
ETMCNTENR1-2

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Enable Event

PTM_ETMCNTRLDEVR1-2
Address: PTM_BASE + offset(0x160-0x164)
ETMCNTRLDEVR1-2

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Reload Event

PTM_ETMCNTVR1-2
Address: PTM_BASE + offset(0x160-0x164)
ETMCNTVR1-2

Bits	Attr	Reset Value	Description
31:0	RW	0x0	Value

PTM_ETMSYNCFR
Address: PTM_BASE + offset(0x1e0)

Bits	Attr	Reset Value	Description
31:0	RW	0x0	The ETMSYNCFR holds the trace synchronization frequency value.Bits [2:0] of this register are not implemented and read as zero (RAZ).

PTM_ETMIDR
Address: PTM_BASE + offset(0x1e4)

Bits	Attr	Reset Value	Description
31:24	RO	0x41	Implementor code. This field reads as 0x41, ASCII code for A, indicating ARM Limited.
23:20	-	-	Reserved
19	RO	0x1	Support for Security Extensions. The value of this bit is 1, indicating that the processor implements the ARM architecture Security Extensions.
18	RO	0x1	Support for 32-bit Thumb instructions. The value of this bit is 1, indicating that a 32-bit Thumb instruction is traced as a single instruction.
17:12	-	-	Reserved
11:8	RO	0x3	Major architecture version number.
7:4	RO	0x0	Major architecture version number.
3:0	RO	-	Implementation revision

PTM_ETMCCER
Address: PTM_BASE + offset(0x1e8)

Bits	Attr	Reset Value	Description
31:26	RO	0x0	Reserved
25	RO	0x0	Timestamps not generated for DMB/DSB.
24	RO	0x0	MB/DSB instructions are not treated as waypoints.
23	RO	0x1	Return stack implemented.
22	RO	0x1	Timestamping implemented.
21:16	-	-	Reserved
15:13	RO	0x0	Specifies the number of instrumentation resources.
12	-	-	Reserved
11	RO	0x1	b1 - Indicates that all registers, except some Integration Test Registers, are readable.
10:3	RO	0x34	Specifies the size of the extended external input bus, 52.
2:0	RO	0x2	Specifies the number of extended external input selectors, 2.

PTM_ETMEXTINSELR

Address: PTM_BASE + offset(0x1ec)

Bits	Attr	Reset Value	Description
31:14	-	-	Reserved
13:8	RW	0x0	Second extended external input selector
7:6	-	-	Reserved
5:0	RW	0x0	First extended external input selector

PTM_ETMAUXCR

Address: PTM_BASE + offset(0x1fc)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3	RW	0x0	Force insertion of synchronization packets, regardless of current trace activity. Possible values for this bit are: b0 = Synchronization packets delayed when trace activity is high. This is the reset value. b1 = Synchronization packets inserted regardless of trace activity. This bit might be set if synchronization packets occur too far apart. Setting this bit might cause the trace FIFO to overflow more frequently when trace activity is high.
2	RW	0x0	Specifies whether the PTM issues waypoint update packets if there are more than 4096 bytes between waypoints. Possible values for this bit are: b0 = PTM always issues update packets if there are more than 4096 bytes between waypoints. This is the reset value. b1 = PTM does not issue waypoint update packets unless required to do so as the result of

			an exception or debug entry.
1	RW	0x0	Specifies whether the PTM issues a timestamp on a barrier instruction. Possible values for this bit are: b0 = PTM issues timestamps on barrier instructions. This is the reset value. b1 = PTM does not issue timestamps on barriers
0	RW	0x0	Specifies whether the PTM enters overflow state when synchronization is requested, and the previous synchronization sequence has not yet completed. This does not affect entry to overflow state when the FIFO becomes full. Possible values for this bit are: b0 = Forced overflow enabled. This is the reset value. b1 = Forced overflow disabled

PTM_ETMTRACEIDR

Address: PTM_BASE + offset(0x200)

Bits	Attr	Reset Value	Description
31:7	-	-	Reserved
6:0	RW	0x0	Before trace is generated, you must program this register with a non-reserved value. Reserved values are 0x00 and any value in the range 0x70-0x7F. The reset value of this register is 0x00

PTM_ETMPDSR

Address: PTM_BASE + offset(0x314)

Bits	Attr	Reset Value	Description
31:0	RO	0x1	This register always reads as 0x00000001, indicating that the PTM Trace Registers can be accessed.

PTM_OSLSR

Address: PTM_BASE + offset(0x304)

Bits	Attr	Reset Value	Description
31:0	RO	0x0	For the PTM, the OSLSR Reads As Zero (RAZ) to show that OS Locking is not implemented.

PTM_ITMISCOUT

Address: PTM_BASE + offset(0xedc)

Bits	Attr	Reset Value	Description
31:10	-	-	Reserved
9:8	WO	0x0	Drives the PTMEXTOUT[1:0] outputs
7:6	-	-	Reserved
5	WO	0x0	Drives the PTMIDLEnACK output
4	WO	0x0	Drives the PTMDBGREQ output
3:0	WO	0x0	Reserved

PTM_ITMISCIN

Address: PTM_BASE + offset(0xee0)

Bits	Attr	Reset Value	Description
31:7	-	-	Reserved
6	RO	0x0	Returns the value of the STANDBYWFI input
5	-	-	Reserved
4	RO	0x0	Returns the value of the PTMDBGACK input
3:0	RO	0x0	Returns the value of the EXTIN[3:0] inputs

PTM_ITTRIGGER

Address: PTM_BASE + offset(0xee8)

Bits	Attr	Reset Value	Description
31:1	-	-	Reserved
0	WO	0x0	Drives the PTMTRIGGER output

PTM_ITATBDATA0

Address: PTM_BASE + offset(0xeec)

Bits	Attr	Reset Value	Description
31:5	-	-	Reserved
4	WO	-	Drives the ATDATAM[31] output
3	WO	-	Drives the ATDATAM[23] output
2	WO	-	Drives the ATDATAM[15] output
1	WO	-	Drives the ATDATAM[7] output
0	WO	-	Drives the ATDATAM[0] output

PTM_ITATBCTR2

Address: PTM_BASE + offset(0xef0)

Bits	Attr	Reset Value	Description
31:2	-	-	Reserved
1	RO	-	Returns the value of the AFVALIDM input
0	RO	-	Returns the value of the ATREADYM input

PTM_ITATBID

Address: PTM_BASE + offset(0xef4)

Bits	Attr	Reset Value	Description
31:7	-	-	Reserved
6:0	WO	-	Drives the ATIDM[6:0] outputs

PTM_ITATBCTR0

Address: PTM_BASE + offset(0xef8)

Bits	Attr	Reset Value	Description
31:10	-	-	Reserved
9:8	WO	-	Drives the ATBYTESM outputs
7:2	-	-	Reserved
1	WO	-	Drives the AFREADYM output
0	WO	-	Drives the ATVALIDM output

PTM_ETMITCTRL
Address: PTM_BASE + offset(0xff0)

Bits	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	When bit [0] is set to 1, the PTM enters an integration mode. On reset this bit is cleared to 0. Before entering integration mode, the PTM must be powered up and in programming mode. This means bit [0] of the Main Control Register is set to 0, and bit [10] of the Main Control Register is set to 1. After leaving integration mode, the PTM must be reset before attempting to perform tracing.

PTM_PID4
Address: PTM_BASE + offset(0xfd0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x04	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_PID5
Address: PTM_BASE + offset(0xfd4)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_PID6
Address: PTM_BASE + offset(0xfd8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_PID7
Address: PTM_BASE + offset(0xfdc)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.

3:0	-	-	Reserved

PTM_PID0

Address: PTM_BASE + offset(0xfe0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x50	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_PID1

Address: PTM_BASE + offset(0xfe4)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0xb9	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_PID2

Address: PTM_BASE + offset(0xfe8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x1b	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_PID3

Address: PTM_BASE + offset(0xfec)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

PTM_CID0

Address: PTM_BASE + offset(0xff0)

Bits	Attr	Reset Value	Description
31:0	RO	0x0d	The component identification registers identify the PTM as a CoreSight component

PTM_CID1

Address: PTM_BASE + offset(0xff4)

Bits	Attr	Reset Value	Description
31:0	RO	0x90	The component identification registers identify the PTM as a CoreSight component

PTM_CID2
Address: PTM_BASE + offset(0xff8)

Bits	Attr	Reset Value	Description
31:0	RO	0x05	The component identification registers identify the PTM as a CoreSight component

PTM_CID3
Address: PTM_BASE + offset(0xffc)

Bits	Attr	Reset Value	Description
31:0	RO	0xb1	The component identification registers identify the PTM as a CoreSight component

6.4.9 Funnel register summary

Name	Offset	Size	Reset Value	Description
FUNNEL_FCR	0x0	W	0x300	CSTF Control Register
FUNNEL_PCR	0x4	W	0xfac688	CSTF Priority Control Register
FUNNEL_ITATBDATA0	0xeec	W	0x0	CSTF Integration Test Registers
FUNNEL_ITATBCTR2	0xef0	W	0x0	CSTF Integration Test Registers
FUNNEL_ITATBCTR1	0xef4	W	0x0	CSTF Integration Test Registers
FUNNEL_ITATBCTR0	0xef8	W	0x0	CSTF Integration Test Registers
FUNNEL_IMCR	0xff0	W	0x0	Integration Mode Control Register
FUNNEL_CTSR	0xfa0	W	0xf	Claim Tag Set Register
FUNNEL_CTCR	0xfa4	W	0x0	Claim Tag Clear Register
FUNNEL_LA	0xfb0	W	-	Lock Access
FUNNEL_LS	0xfb4	W	0x0	Lock Status
FUNNEL_AS	0xfb8	W	0x0	Authentication status
FUNNEL_DI	0xfc8	W	0x28	Device ID
FUNNEL_DTI	0xfcc	W	0x12	Device Type Identifier
FUNNEL_PID4	0xfd0	W	0x04	Peripheral ID4
FUNNEL_PID0	0xfe0	W	0x08	Peripheral ID0
FUNNEL_PID1	0xfe4	W	0xb9	Peripheral ID1
FUNNEL_PID2	0xfe8	W	0x1b	Peripheral ID2
FUNNEL_PID3	0xfec	W	0x00	Peripheral ID3
FUNNEL_CID0	0xff0	W	0x0d	Component ID0
FUNNEL_CID1	0xff4	W	0x90	Component ID1
FUNNEL_CID2	0xff8	W	0x05	Component ID2
FUNNEL_CID3	0xffc	W	0xb1	Component ID3

6.4.10 Funnel register details

FUNNEL_CR
Address: FUNNEL_BASE + offset(0x000)

Bits	Attr	Reset Value	Description
31:12	-	-	Reserved
11:8	RW	0x3	Minimum hold time[3:0] The formatting scheme can easily become inefficient if fast switching occurs, so, where possible, this must be minimized. If a source has nothing to transmit, then another source is selected irrespective of the minimum number of cycles. Reset is 0x3. The CSTF holds for the minimum hold time and one additional cycle. The maximum value that can be entered is 0xE and this equates to 15 cycles. 0xF is reserved.
7	RW	0x0	Enable Slave port 7 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.
6	RW	0x0	Enable Slave port 6 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.
5	RW	0x0	Enable Slave port 5 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.
4	RW	0x0	Enable Slave port 4 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.
3	RW	0x0	Enable Slave port 3 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.
2	RW	0x0	Enable Slave port 2 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.
1	RW	0x0	Enable Slave port 1 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection

			scheme. The reset value is all clear, that is, all ports disabled.
0	RW	0x0	Enable Slave port 0 Setting this bit enables this input, or slave, port. If the bit is not set then this has the effect of excluding the port from the priority selection scheme. The reset value is all clear, that is, all ports disabled.

FUNNEL_PCR

Address: FUNNEL_BASE + offset(0x004)

Bits	Attr	Reset Value	Description
31:24	-	-	Reserved
23:21	RW	0x0	PriPort 7 Priority value of the eighth port. The value written into this location is the value that you want to assign the eighth slave port.
20:18	RW	0x0	PriPort 6 7th port priority value.
17:15	RW	0x0	PriPort 5 6th port priority value.
14:12	RW	0x0	PriPort 4 5th port priority value.
11:9	RW	0x0	PriPort 3 4th port priority value.
8:6	RW	0x0	PriPort 2 3th port priority value.
5:3	RW	0x0	PriPort 1 2th port priority value.
2:0	RW	0x0	PriPort 0 Priority value of the first slave port. The value written into this location is the value that you want to assign the first slave port

FUNNEL_ITATBDATA0

Address: FUNNEL_BASE + offset(0xeec)

Bits	Attr	Reset Value	Description
31:5	-	-	Reserved
4	RW	0x0	the value of ATDATAS<31>
3	RW	0x0	the value of ATDATAS<23>
2	RW	0x0	the value of ATDATAS<15>
1	RW	0x0	the value of ATDATAS<7>
0	RW	0x0	the value of ATDATAS<0>

FUNNEL_ITATBCTR2

Address: FUNNEL_BASE + offset(0xef0)

Bits	Attr	Reset Value	Description
31:2	-	-	Reserved
1	RW	0x0	the value of AFVALIDM
0	RW	0x0	the value of ATREADYM

FUNNEL_ITATBCTR1
Address: FUNNEL_BASE + offset(0xef4)

Bits	Attr	Reset Value	Description
31:7	-	-	Reserved
6:0	RW	0x0	the value of ATIDS

FUNNEL_ITATBCTR0
Address: FUNNEL_BASE + offset(0xef4)

Bits	Attr	Reset Value	Description
31:10	-	-	Reserved
9:8	RW	0x0	the value of ATBYTESS<n>
7:2	-	-	Reserved
1	RW	0x0	the value of AFREADYS<n>
0	RW	0x0	Read the value of ATVALIDS<n>

FUNNEL_CTS- FUNNEL_CTC
Address: FUNNEL_BASE + offset(0xfa0-0xfa4)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	The CSTF implements a four-bit claim tag. The use of bits [3:0] is software defined.

FUNNEL_LA- FUNNEL_LS
Address: FUNNEL_BASE + offset(0xfb0-0xfb4)

Bits	Attr	Reset Value	Description
31:3	-	-	Reserved
2:0	RW	0x3	The CSTF implements two memory maps controlled through PADDRDBG31. When PADDRDBG31 is HIGH, the Lock Status Register reads as 0x0 indicating that no lock exists. WhenPADDRDBG31 is LOW, the Lock Status Register reads as 0x3 from reset. This indicates a 32-bit lock access mechanism is present and is locked.

FUNNEL_AS
Address: FUNNEL_BASE + offset(0xfb8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	Reports the required security level. This is set to 0x00 for functionality not implemented.

FUNNEL_DID
Address: FUNNEL_BASE + offset(0xfc8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RW	0x0	The CSTF implements a static priority scheme.

3:0	RW	0x8	This is the value of the Verilog define PORTCOUNT and represents the number of input ports connected. By default all 8 ports are connected. 0x0 and 0x1 are illegal values.

FUNNEL_DTID

Address: FUNNEL_BASE + offset(0xfcc)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x12	A value of 0x12 identifies this device as a trace link (0x2) and specifically as a funnel/router (0x1)

FUNNEL_PID4

Address: FUNNEL_BASE + offset(0xfd0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x04	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID5

Address: FUNNEL_BASE + offset(0xfd4)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID6

Address: FUNNEL_BASE + offset(0xfd8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID7

Address: FUNNEL_BASE + offset(0xfdc)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID0

Address: FUNNEL_BASE + offset(0xfe0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x50	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID1

Address: FUNNEL_BASE + offset(0xfe4)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0xb9	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID2

Address: FUNNEL_BASE + offset(0xfe8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x1b	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_PID3

Address: FUNNEL_BASE + offset(0xfec)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

FUNNEL_CID0

Address: FUNNEL_BASE + offset(0xff0)

Bits	Attr	Reset Value	Description
31:0	RO	0x0d	The component identification registers identify the PTM as a CoreSight component

FUNNEL_CID1

Address: FUNNEL_BASE + offset(0xff4)

Bits	Attr	Reset Value	Description
31:0	RO	0x90	The component identification registers identify the PTM as a CoreSight component

FUNNEL_CID2

Address: FUNNEL_BASE + offset(0xff8)

Bits	Attr	Reset Value	Description
31:0	RO	0x05	The component identification registers identify the PTM as a CoreSight component

FUNNEL_CID3

Address: FUNNEL_BASE + offset(0xffc)

Bits	Attr	Reset Value	Description
31:0	RO	0xb1	The component identification registers identify the PTM as a CoreSight component

6.4.11 CTI register summary

Name	Offset	Size	Reset Value	Description
CTI_CTICONTROL	0x000	W	0x0	CTI Control Register
CTI_CTIINTACK	0x010	W	-	CTI Interrupt Acknowledge Register
CTI_CTIAPPSET	0x014	W	0x0	CTI Application Trigger Set Register
CTI_CTIAPPCLEAR	0x018	W	0x0	CTI Application Trigger Clear Register
CTI_CTIAPPPULSE	0x01c	W	0x0	CTI Application Pulse Register
CTI_CTIINEN	0x020-0x03c	W	0x0	CTI Trigger to Channel Enable Registers, CTIINEN0-7
CTI_CTIOUTEN	0x0a0-0x0bc	W	0x0	CTI Channel to Trigger Enable Registers, CTIOUTEN0-7
CTI_CTITRIGINSTATUS	0x130	W	0x0	CTI Trigger In Status Register, CTITRIGINSTATUS
CTI_CTICHINSTATUS	0x138	W	-	CTI Channel In Status Register, CTICHINSTATUS
CTI_CTICHOUTSTATUS	0x13c	W	0x0	CTI Channel Out Status Register
CTI_CTIGATE	0x140	W	0xf	Enable CTI Channel Gate Register
CTI_ASICCTL	0x144	W	0x0	External Multiplexor Control Register
CTI_ITCHINACK	0xedc	W	0x0	ITCHINACK Register
CTI_ITTRIGINACK	0xee0	W	0x0	ITTRIGINACK Register
CTI_ITCHOUT	0xee4	W	0x0	ITCHOUT Register
CTI_ITTRIGOUT	0xee8	W	0x0	ITTRIGOUT Register
CTI_ITCHOUTACK	0xeec	W	0x0	ITCHOUTACK Register
CTI_ITTRIGOUTACK	0xef0	W	0x0	ITTRIGOUTACK Register
CTI_ITCHIN	0xef4	W	0x0	ITCHIN Register
CTI_ITTRIGIN	0xef8	W	0x0	ITTRIGIN Register
CTI_ITCTRL	0xf00	W	0x0	ITCTRL Register
CTI_CTSR	0xfa0	W	0xf	Claim Tag Set Register
CTI_CTCR	0xfa4	W	0x0	Claim Tag Clear Register

CTI_LA	0xfb0	W	-	Lock Access
CTI_LS	0xfb4	W	0x0	Lock Status
CTI_AS	0xfb8	W	0x0	Authentication status
CTI_DI	0xfc8	W	0x28	Device ID
CTI_DTI	0xfcc	W	0x12	Device Type Identifier
CTI_PID4	0xfd0	W	0x04	Peripheral ID4
CTI_PID5	0xfd4	W	0x00	Peripheral ID5
CTI_PID6	0xfd8	W	0x00	Peripheral ID6
CTI_PID7	0xfdc	W	0x00	Peripheral ID7
CTI_PID0	0xfe0	W	0x08	Peripheral ID0
CTI_PID1	0xfe4	W	0xb9	Peripheral ID1
CTI_PID2	0xfe8	W	0x1b	Peripheral ID2
CTI_PID3	0xfec	W	0x00	Peripheral ID3
CTI_CID0	0xff0	W	0x0d	Component ID0
CTI_CID1	0xff4	W	0x90	Component ID1
CTI_CID2	0xff8	W	0x05	Component ID2
CTI_CID3	0xffc	W	0xb1	Component ID3

6.4.12 CTI register details

CTI_CTICONTROL

Address: CTI_BASE + offset(0x000)

Bits	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	GLBEN Enables or disables the ECT: 0 = disabled (reset) 1 = enabled. When disabled, all cross triggering mapping logic functionality is disabled for this processor

CTI_CTIINTACK

Address: CTI_BASE + offset(0x010)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	WO	-	INTACK Acknowledges the corresponding CTITRIGOUT output: 1 = CTITRIGOUT is acknowledged and is cleared when MAPTRIGOUT is LOW. 0 = no effect. There is one bit of the register for each CTITRIGOUT output

CTI_CTIAPPSET

Address: CTI_BASE + offset(0x014)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	APPSET Setting a bit HIGH generates a channel event for the selected channel.

			Read: 0 = application trigger inactive (reset) 1 = application trigger active. Write: 0 = no effect 1 = generate channel event. There is one bit of the register for each channel

CTI_CTIAPPCLEAR

Address: CTI_BASE + offset(0x018)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	WO	-	Clears corresponding bits in the CTIAPPSET register. 1 = application trigger disabled in the CTIAPPSET register 0 = no effect. There is one bit of the register for each channel.

CTI_CTIAPPPULSE

Address: CTI_BASE + offset(0x01c)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	WO	-	APPULSE Setting a bit HIGH generates a channel event pulse for the selected channel. Write: 1 = channel event pulse generated for one CTICLK period 0 = no effect. There is one bit of the register for each channel.

CTI_CTIINEN0-7

Address: CTI_BASE + offset(0x020-0x03c)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	TRIGINEN Enables a cross trigger event to the corresponding channel when an CTITRIGIN is activated. 1 = enables the CTITRIGIN signal to generate an event on the respective channel of the CTM. There is one bit of the register for each of the four channels. For example in register CTIINEN0, TRIGINEN[0] set to 1 enables CTITRIGIN onto channel 0. 0 = disables the CTITRIGIN signal from generating an event on the respective channel of the CTM

CTI_CTIOUTEN0-7

Address: CTI_BASE + offset(0x0a0-0x0bc)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	TRIGOUTEN Changing the value of this bit from a 0 to a 1 enables a channel event for the corresponding channel to generate an CTITRIGOUT output: 0 = the channel input (CTICHIN) from the CTM is not routed to the CTITRIGOUT output 1 = the channel input (CTICHIN) from the CTM is routed to the CTITRIGOUT output. There is one bit for each of the four channels. For example in register CTIOUTEN0, enabling bit 0 enables CTICHIN[0] to cause a trigger event on the CTITRIGOUT[0] output.

CTI_CTITRIGINSTATUS

Address: CTI_BASE + offset(0x130)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	TRIGINSTATUS Shows the status of the CTITRIGIN inputs: 1 = CTITRIGIN is active 0 = CTITRIGIN is inactive. Because the register provides a view of the raw CTITRIGIN inputs, the reset value is unknown. There is one bit of the register for each trigger input.

CTI_CTITRIGOUTSTATUS

Address: CTI_BASE + offset(0x134)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	TRIGOUTSTATUS Shows the status of the CTITRIGOUT outputs. 1 = CTITRIGOUT is active 0 = CTITRIGOUT is inactive (reset). There is one bit of the register for each trigger output.

CTI_CTICHINSTATUS

Address: CTI_BASE + offset(0x138)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	CTICHINSTATUS Shows the status of the CTICHIN inputs: 1 = CTICHIN is active 0 = CTICHIN is inactive. Because the register provides a view of the raw CTICHIN inputs from the CTM, the reset value is unknown. There is one bit of the register for each channel input.

CTI_CTICHOUTSTATUS

Address: CTI_BASE + offset(0x13c)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	CTICHOUTSTATUS Shows the status of the CTICHOUT outputs. 1 = CTICHOUT is active 0 = CTICHOUT is inactive (reset). There is one bit of the register for each channel output.

CTI_CTIGATE

Address: CTI_BASE + offset(0x140)

Bits	Attr	Reset Value	description
31:4	-	-	Reserved
3	RW	0x0	CTIGATEEN3 Enable CTICHOUT3. Set to 0 to disable channel propagation.
2	RW	0x0	CTIGATEEN2 Enable CTICHOUT2. Set to 0 to disable channel propagation.
1	RW	0x0	CTIGATEEN1 Enable CTICHOUT1. Set to 0 to disable channel propagation.
0	RW	0x0	CTIGATEEN0 Enable CTICHOUT0. Set to 0 to disable channel propagation

CTI_ASICCTL

Address: CTI_BASE + offset(0x144)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	ASICCTL Implementation-defined ASIC control, value written to the register is output on ASICCTL[7:0]. If external multiplexing of trigger signals is implemented then the number of multiplexed signals on each trigger must be reflected within the Device ID Register. This is done within a Verilog define EXTMUXNUM.

CTI_ITCHINACK

Address: CTI_BASE + offset(0xedc)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	WO	-	CTCHINACK Set the value of the CTCHINACK outputs

CTI_ITTRIGINACK

Address: CTI_BASE + offset(0xee0)

Bits	Attr	Reset Value	Description

31:8	-	-	Reserved
7:0	WO	-	CTTRIGINACK Set the value of the CTTRIGINACK outputs

CTI_ITCHOUT

Address: CTI_BASE + offset(0xee4)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	WO	-	CTCHOUT Set the value of the CTCHOUT outputs

CTI_ITTRIGOUT

Address: CTI_BASE + offset(0xee8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	WO	-	CTTRIGOUT Set the value of the CTTRIGOUT outputs

CTI_ITCHOUTACK

Address: CTI_BASE + offset(0xeec)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RO	0x0	CTCHOUTACK Read the values of the CTCHOUTACK inputs

CTI_ITTRIGOUTACK

Address: CTI_BASE + offset(0xef0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RO	0x0	CTTRIGOUTACK Read the values of the CTTRIGOUTACK inputs

CTI_ITCHIN

Address: CTI_BASE + offset(0xef4)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RO	0x0	CTCHIN Read the values of the CTCHIN inputs

CTI_ITTRIGIN

Address: CTI_BASE + offset(0xef8)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RO	0x0	CTTRIGIN Read the values of the CTTRIGIN inputs

CTI_CTS- CTI_CTC

Address: CTI_BASE + offset(0xfa0-0xfa4)

Bits	Attr	Reset Value	Description

31:4	-	-	Reserved
3:0	RW	0x0	The CTI implements a four-bit claim tag. The use of bits [3:0] is software defined

CTI_LA- CTI_LS

Address: CTI_BASE + offset(0xfb0-0xfb4)

Bits	Attr	Reset Value	Description
31:3	-	-	Reserved
2:0	RW	0x3	The CTI implements two memory maps controlled through PADDRDBG31. When PADDRDBG31 is HIGH, the Lock Status Register reads as 0x0 indicating that no lock exists. WhenPADDRDBG31 is LOW, the Lock Status Register reads as 0x3 from reset. This indicates a 32-bit lock access mechanism is present and is locked.

CTI_AS

Address: CTI_BASE + offset(0xfb8)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3	RW	0x0	Current value of noninvasive debug enable signals
2	RW	0x0	Non-invasive debug controlled
1	RW	0x0	Current value of invasive debug enable signals
0	RW	0x0	Invasive debug controlled

CTI_DID

Address: CTI_BASE + offset(0xfc8)

Bits	Attr	Reset Value	Description
31:20	-	-	Reserved
19:16	RO	0x0	Number of ECT channels available.
15:8	RO	0x0	Number of ECT triggers available.
7:5	-	-	Reserved
4:0	RO	0x0	Indicates the number of multiplexing available on Trigger Inputs and Trigger Outputs using ASICCTL. Default value of 5'b00000 indicating no multiplexing present. Reflects the value of the Verilog `define EXTMUXNUM that you must alter accordingly.

CTI_DTID

Address: CTI_BASE + offset(0xfcc)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x14	0x14 indicates this device has a major type of debug control logic component (0x4) and sub-type corresponding to cross trigger (0x1).

CTI_PID4

Address: CTI_BASE + offset(0xfd0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
3:0	RO	0x3	The CTI is identified as an ARM component with a JEP106 identity at 0x3B and a JEP106 continuation code at 0x4 (fifth bank).

CTI_PID0

Address: CTI_BASE + offset(0xfe0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x0	Middle BCD value of Device number.
3:0	RO	0x6	Lower BCD value of Device number.

CTI_PID1

Address: CTI_BASE + offset(0xfe4)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0xb	The CTI is identified as an ARM component with a JEP106 identity at 0x3B and a JEP106 continuation code at 0x4 (fifth bank).
3:0	RO	0x9	Upper BCD value of Device number.

CTI_PID2

Address: CTI_BASE + offset(0xfe8)

Bits	Attr	Reset Value	Description
31:3	-	-	Reserved
2:0	RO	0x4	The CTI is identified as an ARM component with a JEP106 identity at 0x3B and a JEP106 continuation code at 0x4 (fifth bank).
3:0	-	-	Reserved

CTI_PID3

Address: CTI_BASE + offset(0xfec)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

CTI_CID0

Address: CTI_BASE + offset(0xff0)

Bits	Attr	Reset Value	Description
31:0	RO	0x0d	The component identification registers identify the PTM as a CoreSight component

CTI_CID1

Address: CTI_BASE + offset(0xff4)

Bits	Attr	Reset Value	Description
31:0	RO	0x90	The component identification registers identify the PTM as a CoreSight component

CTI_CID2
Address: CTI_BASE + offset(0xff8)

Bits	Attr	Reset Value	Description
31:0	RO	0x05	The component identification registers identify the PTM as a CoreSight component

CTI_CID3
Address: CTI_BASE + offset(0xffc)

Bits	Attr	Reset Value	Description
31:0	RO	0xb1	The component identification registers identify the PTM as a CoreSight component

6.4.13 TPIU register summary

Name	Offset	Size	Reset Value	Description
TPIU_SPSR	0x000	W	0x0000008a	Supported port size
TPIU_CPSR	0x004	W	0x00000001	Current port size
TPIU_STMR	0x100	W	0x11f	Supported trigger modes
TPIU_TCR	0x104	W	0x00	Trigger counter value
TPIU_TMR	0x108	W	0x00	Trigger multiplier
TPIU_STPMR	0x200	W	0x3000f	Supported test pattern/modes
TPIU_CTPMR	0x204	W	0x00000	Current test pattern/mode
TPIU_TTPRCR	0x208	W	0x00	TPIU Test pattern repeat counter
TPIU_FFSR	0x300	W	0x00000006	Formatter and flush status
TPIU_FFCR	0x304	W	0x00000100	Formatter and flush control
TPIU_FSCR	0x308	W	0x00000000	Formatter synchronization counter
TPIU_EXCTLPRIN	0x400	W	N/A	EXTCTL In Port
TPIU_EXCTLPROUT	0x404	W	0x00	EXTCTL Out Port
TPIU_ITTRFLINACK	0xee4	W	N/A	Integration Register
TPIU_ITTRFLIN	0xee8	W	N/A	Integration Register
TPIU_ITATBDATA0	0xeec	W	N/A	Integration Register
TPIU_ITATBCTR2	0xef0	W	N/A	Integration Register
TPIU_ITATBCTR0	0xef8	W	N/A	Integration Register
TPIU_ITAMCTL	0xf00	W	0x00000000	Integration Mode control register
TPIU_CTS	0xfa0	W	0x0000000f	Claim tag set
TPIU_CTC	0xfa4	W	0x00000000	Claim tag clear

TPIU_LA	0xfb0	W	N/A	Lock access
TPIU_LS	0xfb4	W	N/A	Local status
TPIU_AS	0xfb8	W	0x00000000	Authentication status
TPIU_DID	0xfc8	W	0x00000000	Device ID
TPIU_DTID	0xfcc	W	0x00000011	Device type identifier
TPIU_PID4	0xfd0	W	0x00000004	Peripheral ID4
TPIU_PID0	0xfe0	W	0x00000041	Peripheral ID0
TPIU_PID1	0xfe4	W	0x000000b9	Peripheral ID1
TPIU_PID2	0xfe8	W	0x0000000b	Peripheral ID2
TPIU_PID3	0xfec	W	0x00000000	Peripheral ID3
TPIU_CID0	0xff0	W	0x0000000d	Component ID0
TPIU_CID1	0xff4	W	0x00000090	Component ID1

6.4.14 TPIU detailed register description

TPIU_SPSR

Address: TPIU_BASE + offset(0x000)

Bits	Attr	Reset Value	Description
31:0	RW	0x0000008a	This register is read/write. Each bit location represents a single port size that is supported on the device, that is, 32-1 in bit locations [31:0]. If the bit is set then that port size is allowed. By default the RTL is designed to support all port sizes, set to 0xFFFFFFFF. This register reflects the value of the CSTPIU_SUPPORTSIZE_VAL Verilog `define value, currently not user modifiable, and is further constrained by the input tie-off TPMAXDATASIZE. The external tie-off, TPMAXDATASIZE, must be set during finalization of the ASIC to reflect the actual number of TRACEDATA signals being wired to physical pins. This is to ensure that tools do not attempt to select a port width that cannot be captured by an attached TPA. The value on TPMAXDATASIZE causes bits within the Supported Port Size register that represent wider widths to be clear, that is, unsupported

TPIU_CPSR

Address: TPIU_BASE + offset(0x004)

Bits	Attr	Reset Value	Description
31:0	RW	0x00000001	This register is read/write. The Current Port Size Register has the same format as the Supported Port Sizes register but only one bit is set, and all others must be zero. Writing values with more than one bit set or setting a bit that is not indicated as supported is not supported and causes unpredictable behavior. On reset this defaults to the smallest possible port size, 1 bit, and so reads as 0x00000001

TPIU_STMR
Address: TPIU_BASE + offset(0x100)

Bits	Attr	Reset Value	Description
31:18	-	-	Reserved
17	RO	0x0	Trigger Counter running. A trigger has occurred but the counter is not at zero.
16	RO	0x1	Triggered. A trigger has occurred and the counter has reached zero
15:9	-	-	Reserved
8	RO	0x0	8-bit wide counter register implemented.
7:5	-	-	Reserved
4	RO	0x1	Multiply the Trigger Counter by 65536 supported.
3	RO	0x1	Multiply the Trigger Counter by 256 supported.
2	RO	0x1	Multiply the Trigger Counter by 16 supported.
1	RO	0x1	Multiply the Trigger Counter by 4 supported
0	RO	0x1	Multiply the Trigger Counter by 2 supported

TPIU_TCR
Address: TPIU_BASE + offset(0x104)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	8-bit counter value for the number of words to be output from the formatter before a trigger is inserted. Reset value is 0x00.

TPIU_TMR
Address: TPIU_BASE + offset(0x108)

Bits	Attr	Reset Value	Description
31:5	-	-	Reserved
4	RO	0x0	Multiply the Trigger Counter by 65536 supported.
3	RO	0x0	Multiply the Trigger Counter by 256 supported.
2	RO	0x0	Multiply the Trigger Counter by 16 supported.
1	RO	0x0	Multiply the Trigger Counter by 4 supported
0	RO	0x0	Multiply the Trigger Counter by 2 supported

TPIU_STPMR
Address: TPIU_BASE + offset(0x200)

Bits	Attr	Reset Value	Description
31:18	-	-	Reserved
17	RO	0x1	Continuous mode
16	RO	0x1	Timed mode
15:4	-	-	Reserved
3	RO	0x1	FF/00 Pattern

2	RO	0x1	AA/55 Pattern
1	RO	0x1	Walking 0s Pattern
0	RO	0x1	Walking 1s Pattern

TPIU_CTPMR

Address: TPIU_BASE + offset(0x204)

Bits	Attr	Reset Value	Description
31:18	-	-	Reserved
17:16	RW	0x0	Mode select
15:4	-	-	Reserved
3:0	RW	0x0	Number of cycles

TPIU_TTPRCR

Address: TPIU_BASE + offset(0x208)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	8-bit counter value to indicate the number of TRACECLKIN cycles that a pattern runs for before switching to the next pattern. Default value is 0

TPIU_FFSR

Address: TPIU_BASE + offset(0x300)

Bits	Attr	Reset Value	Description
31:3	-	-	Reserved
2	RO	0x1	If this bit is set then TRACECTL is present. If no TRACECTL pin is available, that is, this bit is zero, then the data formatter must be used and only in continuous mode. This is constrained by the CSTPIU_TRACECTL_VAL Verilog `define, which is not user modifiable, and the external tie-off TPCTL. If either constraint reports zero/LOW then no TRACECTL is present and this inability to use the pin is reflected in this register.
1	RO	0x1	Formatter stopped. The formatter has received a stop request signal and all trace data and post-amble has been output. Any more trace data on the ATB interface is ignored and ATREADYS goes HIGH.
0	RO	0x0	Flush In Progress. This is an indication of the current state of AFVALIDS

TPIU_FFCR

Address: TPIU_BASE + offset(0x304)

Bits	Attr	Reset Value	Description
31:14	-	-	Reserved
13	RW	0x0	Stop the formatter after a Trigger Eventa is observed. Reset to disabled, or zero.

12	RW	0x0	Stop the formatter after a flush completes (return of AFREADYS). This forces the FIFO to drain off any part-completed packets. Setting this bit enables this function but this is clear on reset, or disabled.
11	-	-	Reserved
10	RW	0x0	Indicates a trigger on Flush completion on AFREADYS being returned.
9	RW	0x0	Indicate a trigger on a Trigger Event
8	RW	0x1	Indicate a trigger on TRIGIN being asserted.
7	-	-	Reserved
6	RW	0x0	Manually generate a flush of the system. Setting this bit causes a flush to be generated. This is cleared when this flush has been serviced. This bit is clear on reset.
5	RW	0x0	Generate flush using Trigger event. Set this bit to cause a flush of data in the system when a Trigger Eventa occurs. Reset value is this bit clear.
4	RW	0x0	Generate flush using the FLUSHIN interface. Set this bit to enable use of the FLUSHIN connection. This is clear on reset.
3:2	-	-	Reserved
1	RW	0x0	Continuous Formatting, no TRACECTL. Embed in trigger packets and indicate null cycles using Sync packets. Reset value is this bit clear. Can only be changed when FtStopped is HIGH.
0	RW	0x0	Enable Formatting. Do not embed Triggers into the formatted stream. Trace disable cycles and triggers are indicated by TRACECTL, where fitted. Reset value is this bit clear. Can only be changed when FtStopped is HIGH.

TPIU_FSCR

Address: TPIU_BASE + offset(0x308)

Bits	Attr	Reset Value	Description
31:12	-	-	Reserved
11:0	RW	0x0	12-bit counter value to indicate the number of complete frames between full synchronization packets. Default value is 64 (0x40).

TPIU_ITTRFLINACK

Address: TPIU_BASE + offset(0xee4)

Bits	Attr	Reset Value	Description
31:2	-	-	Reserved
1	WO	-	Set the value of FLUSHINACK
0	WO	-	Set the value of TRIGINACK

TPIU_ITTRFLIN

Address: TPIU_BASE + offset(0xee8)

Bits	Attr	Reset Value	Description
31:2	-	-	Reserved
1	RO	0x0	Read the value of FLUSHIN
0	RO	0x0	Read the value of TRIGIN

TPIU_ITATBDATA0

Address: TPIU_BASE + offset(0xeec)

Bits	Attr	Reset Value	Description
31:5	-	-	Reserved
4	RO	0x0	Read the value of ATDATAS[31]
3	RO	0x0	Read the value of ATDATAS[23]
2	RO	0x0	Read the value of ATDATAS[15]
1	RO	0x0	Read the value of ATDATAS[7]
0	RO	0x0	Read the value of ATDATAS[0]

TPIU_ITATBCTR2

Address: TPIU_BASE + offset(0xef0)

Bits	Attr	Reset Value	Description
31:2	-	-	Reserved
1	WO	0x0	Set the value of AFVALIDS
0	WO	0x0	Set the value of ATREADYS

TPIU_ITATBCTR1

Address: TPIU_BASE + offset(0xef4)

Bits	Attr	Reset Value	Description
31:7	-	-	Reserved
6:0	RO	0x0	Read the value of ATIDS

TPIU_ITATBCTR0

Address: TPIU_BASE + offset(0xef8)

Bits	Attr	Reset Value	Description
31:10	-	-	Reserved
9:8	RO	0x0	Read the value of ATBYTESS
7:2	-	-	Reserved
1	RO	0x0	Read the value of AFREADYS
0	RO	0x0	Read the value of ATVALIDS

TPIU_CTS-TPIU_CTC

Address: TPIU_BASE + offset(0xfa0-0xfa4)

Bits	Attr	Reset Value	Description
31:4	-	-	Reserved
3:0	RW	0x0	The TPIU implements a four-bit claim tag. The use of bits [3:0] is software defined

TPIU_LA-TPIU_LS

Address: TPIU_BASE + offset(0xfb0-0xfb4)

Bits	Attr	Reset Value	Description
31:3	-	0xb1	Reserved
2:0	RW	0x3	The TPIU implements two memory maps controlled through PADDRDBG31. When PADDRDBG31 is HIGH, the Lock Status Register reads as 0x0 indicating that no lock exists. WhenPADDRDBG31 is LOW, the Lock Status Register reads as 0x3 from reset. This indicates a 32-bit lock access mechanism is present and is locked.

TPIU_AS

Address: TPIU_BASE + offset(0xfb8)

Bits	Attr	Reset Value	Description
31:0	RO	0x0	Reports the required security level. The TPIU has a default value of 0x00 to indicate that this functionality is not implemented.

TPIU_DID

Address: TPIU_BASE + offset(0xfc8)

Bits	Attr	Reset Value	Description
31:12	-	-	Reserved
11	RO	0x0	Indicates Serial Wire Output (UART/NRZ) is not supported.
10	RO	0x0	Indicates Serial Wire Output (Manchester) is not supported.
9	RO	0x0	Indicates trace clock + data is supported.
8:6	RO	0x0	FIFO size in powers of 2. A value of 2 gives a FIFO size of 4 entries, 16 bytes.
5	RO	0x0	Indicates the relationship between ATCLK and TRACECLKIN. 0x1 indicates asynchronous.
4:0	RO	0x0	Hidden Level of Input multiplexing.When nonzero this value indicates the type/number of ATB multiplexing present on the input to the ATB. Currently only 0x00 is supported, that is, no multiplexing present. This value is used to assist topology detection of the ATB structure

TPIU_DTID

Address: TPIU_BASE + offset(0xfcc)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x11	0x11 indicates this device is a trace sink (0x1) and specifically a TPIU (0x1).

TPIU_PID4

Address: TPIU_BASE + offset(0xfd0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved

3:0	RO	0x3	The TPIU is identified as an ARM component with a JEP106 identity at 0x3B and a JEP106 continuation code at 0x4 (fifth bank).

TPIU_PID0

Address: TPIU_BASE + offset(0xfe0)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x1	Middle BCD value of Device number.
3:0	RO	0x2	Lower BCD value of Device number.

TPIU_PID1

Address: TPIU_BASE + offset(0xfe4)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0xb	The TPIU is identified as an ARM component with a JEP106 identity at 0x3B and a JEP106 continuation code at 0x4 (fifth bank).
3:0	RO	0x9	Upper BCD value of Device number.

TPIU_PID2

Address: TPIU_BASE + offset(0xfe8)

Bits	Attr	Reset Value	Description
31:3	-	-	Reserved
2:0	RO	0x4	The TPIU is identified as an ARM component with a JEP106 identity at 0x3B and a JEP106 continuation code at 0x4 (fifth bank).
3:0	-	-	Reserved

TPIU_PID3

Address: TPIU_BASE + offset(0xfec)

Bits	Attr	Reset Value	Description
31:8	-	-	Reserved
7:4	RO	0x00	The peripheral identification registers provide standard information required for all CoreSight components.
3:0	-	-	Reserved

TPIU_CID1

Address: TPIU_BASE + offset(0xff4)

Bits	Attr	Reset Value	Description
31:8	RO	0x0	Reserved
7:4	RO	0x9	The TPIU complies to the CoreSight class of components and this value is set to 0x9.
3:0	-	0x0	Reserved

*Notes: Attr: **RW**– Read/writable, **RO**– read only, **WO**– write only, **RWTC-**Readable and write "1" to clear the asserted bit from "1" to "0".*

6.5 Interface description

6.5.1 DAP SWJ-DP interface

The following figure is the DAP SWJ-DP interface, the SWJ-DP is a combined JTAG-DP and SW-DP that enable you connect either a Serial Write Debug(SWJ) to JTAG probe to a target.

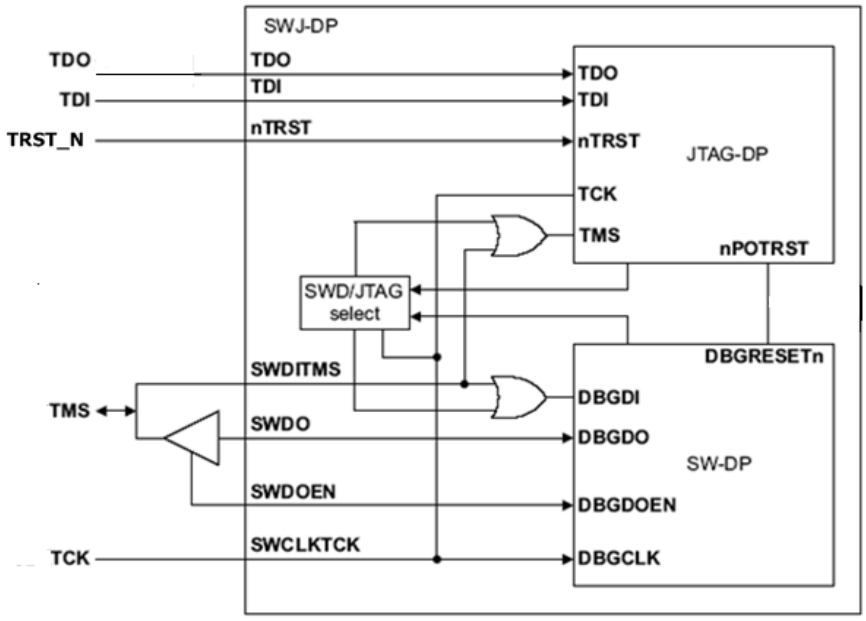


Fig. 6-2 DAP SWJ interface

Table 6-1SWJ interface

Module Pin	Direction	PAD Name	IOMUX Setting
TRST_N	I	TRST_N	Dedicated IO
TCK	I	TCK	Dedicated IO
TDI	I	TDI	Dedicated IO
TMS	IO	TMS	Dedicated IO
TDO	O	TDO	Dedicated IO

6.5.2 TPIU trace port interface

Table 6-2TPIU interface

Module Pin	Direction	PAD Name	IOMUX Setting
trace_data[0]	O	GPIO4_C[0]	GRF_GPIO4C_IOMUX[1:0]=0x2
trace_data[1]	O	GPIO4_C[1]	GRF_GPIO4C_IOMUX[3:2]=0x2
trace_data[2]	O	GPIO4_C[2]	GRF_GPIO4C_IOMUX[5:4]=0x2
trace_data[3]	O	GPIO4_C[3]	GRF_GPIO4C_IOMUX[7:6]=0x2
trace_data[4]	O	GPIO4_C[4]	GRF_GPIO4C_IOMUX[9:8]=0x2
trace_data[5]	O	GPIO4_C[5]	GRF_GPIO4C_IOMUX[11:10]=0x2
trace_data[6]	O	GPIO4_C[6]	GRF_GPIO4C_IOMUX[13:12]=0x2
trace_data[7]	O	GPIO4_C[7]	GRF_GPIO4C_IOMUX[15:14]=0x2

trace_data[8]	O	GPIO4_D[0]	GRF_GPIO4D_IOMUX[1:0]=0x2
trace_data[9]	O	GPIO4_D[1]	GRF_GPIO4D_IOMUX[3:2]=0x2
trace_data[10]	O	GPIO4_D[2]	GRF_GPIO4D_IOMUX[5:4]=0x2
trace_data[11]	O	GPIO4_D[3]	GRF_GPIO4D_IOMUX[7:6]=0x2
trace_data[12]	O	GPIO4_D[4]	GRF_GPIO4D_IOMUX[9:8]=0x2
trace_data[13]	O	GPIO4_D[5]	GRF_GPIO4D_IOMUX[11:10]=0x2
trace_data[14]	O	GPIO6_D[6]	GRF_GPIO4D_IOMUX[13:12]=0x2
trace_data[15]	O	GPIO6_D[7]	GRF_GPIO4D_IOMUX[15:14]=0x2
trace_clk	O	GPIO0_C[6]	GRF_GPIO0C_IOMUX[13:12]=0x1
trace_ctl	O	GPIO0_C[7]	GRF_GPIO0C_IOMUX[15:14]=0x1

Chapter 7 GRF (General Register Files)

7.1 Overview

The general register file will be used to do static set by software, which is composed of many registers for system control.

7.2 Function Description

The function of general register file is ：

- IOMUX control
- Control the state of gpio in power-down mode
- GPIO PAD pulldown and pullup control
- Used for common system control
- Used to record the system state

7.3 Register description、

7.3.1 Register Summary

Name	Offset	Size	Reset Value	Description
GRF_GPIO0L_DIR	0x0000	W	0x00000000	GPIO0A / GPIO0B output enable control
GRF_GPIO0H_DIR	0x0004	W	0x00000000	GPIO0C / GPIO0D output enable control
GRF_GPIO1L_DIR	0x0008	W	0x00000000	GPIO1A / GPIO0B output enable control
GRF_GPIO1H_DIR	0x000c	W	0x00000000	GPIO1C / GPIO1D output enable control
GRF_GPIO2L_DIR	0x0010	W	0x00000000	GPIO2A / GPIO2B output enable control
GRF_GPIO2H_DIR	0x0014	W	0x00000000	GPIO2C / GPIO2D output enable control
GRF_GPIO3L_DIR	0x0018	W	0x00000000	GPIO3A / GPIO3B output enable control
GRF_GPIO3H_DIR	0x001c	W	0x00000000	GPIO3C / GPIO3D output enable control
GRF_GPIO4L_DIR	0x0020	W	0x00000000	GPIO4A / GPIO4B output enable control
GRF_GPIO4H_DIR	0x0024	W	0x00000000	GPIO4C / GPIO4D output enable control
GRF_GPIO6L_DIR	0x0030	W	0x00000000	GPIO6A / GPIO6B output enable control
GRF_GPIO0L_DO	0x0038	W	0x00000000	GPIO0A / GPIO0B output control
GRF_GPIO0H_DO	0x003c	W	0x00000000	GPIO0C / GPIO0D output control

Name	Offset	Size	Reset Value	Description
GRF_GPIO1L_DO	0x0040	W	0x00000000	GPIO1A / GPIO1B output control
GRF_GPIO1H_DO	0x0044	W	0x00000000	GPIO1C / GPIO1D output control
GRF_GPIO2L_DO	0x0048	W	0x00000000	GPIO2A / GPIO2B output control
GRF_GPIO2H_DO	0x004c	W	0x00000000	GPIO2C / GPIO2D output control
GRF_GPIO3L_DO	0x0050	W	0x00000000	GPIO3A / GPIO3B output control
GRF_GPIO3H_DO	0x0054	W	0x00000000	GPIO3C / GPIO3D output control
GRF_GPIO4L_DO	0x0058	W	0x00000000	GPIO4A / GPIO4B output control
GRF_GPIO4H_DO	0x005c	W	0x00000000	GPIO4C / GPIO4D output control
GRF_GPIO6L_DO	0x0068	W	0x00000000	GPIO6A / GPIO6B output control
GRF_GPIO0L_EN	0x0070	W	0x00000000	GPIO0A / GPIO0B output enable
GRF_GPIO0H_EN	0x0074	W	0x00000000	GPIO0C / GPIO0D output enable
GRF_GPIO1L_EN	0x0078	W	0x00000000	GPIO1A / GPIO1B output enable
GRF_GPIO1H_EN	0x007c	W	0x00000000	GPIO1C / GPIO1D output enable
GRF_GPIO2L_EN	0x0080	W	0x00000000	GPIO2A / GPIO2B output enable
GRF_GPIO2H_EN	0x0084	W	0x00000000	GPIO2C / GPIO2D output enable
GRF_GPIO3L_EN	0x0088	W	0x00000000	GPIO3A / GPIO3B output enable
GRF_GPIO3H_EN	0x008c	W	0x00000000	GPIO3C / GPIO3D output enable
GRF_GPIO4L_EN	0x0090	W	0x00000000	GPIO4A / GPIO4B output enable
GRF_GPIO4H_EN	0x0094	W	0x00000000	GPIO4C / GPIO4D output enable
GRF_GPIO6L_EN	0x00a0	W	0x00000000	GPIO6A / GPIO6B output enable
GRF_GPIO0A_IOMUX	0x00a8	W	0x00000000	GPIO0A iomux control

Name	Offset	Size	Reset Value	Description
GRF_GPIO0B_IOMUX	0x00ac	W	0x00000000	GPIO0B iomux control
GRF_GPIO0C_IOMUX	0x00b0	W	0x00000000	GPIO0C iomux control
GRF_GPIO0D_IOMUX	0x00b4	W	0x00000000	GPIO0D iomux control
GRF_GPIO1A_IOMUX	0x00b8	W	0x00000000	GPIO1A iomux control
GRF_GPIO1B_IOMUX	0x00bc	W	0x00000000	GPIO1B iomux control
GRF_GPIO1C_IOMUX	0x00c0	W	0x00000000	GPIO1C iomux control
GRF_GPIO1D_IOMUX	0x00c4	W	0x00000000	GPIO1D iomux control
GRF_GPIO2A_IOMUX	0x00c8	W	0x00000000	GPIO2A iomux control
GRF_GPIO2B_IOMUX	0x00cc	W	0x00000000	GPIO2B iomux control
GRF_GPIO2C_IOMUX	0x00d0	W	0x00000000	GPIO2C iomux control
GRF_GPIO2D_IOMUX	0x00d4	W	0x00000000	GPIO2D iomux control
GRF_GPIO3A_IOMUX	0x00d8	W	0x00000000	GPIO3A iomux control
GRF_GPIO3B_IOMUX	0x00dc	W	0x00000000	GPIO3B iomux control
GRF_GPIO3C_IOMUX	0x00e0	W	0x00000000	GPIO3C iomux control
GRF_GPIO3D_IOMUX	0x00e4	W	0x00000000	GPIO3D iomux control
GRF_GPIO4A_IOMUX	0x00e8	W	0x00000000	GPIO4A iomux control
GRF_GPIO4B_IOMUX	0x00ec	W	0x00000000	GPIO4B iomux control
GRF_GPIO4C_IOMUX	0x00f0	W	0x00000000	GPIO4C iomux control
GRF_GPIO4D_IOMUX	0x00f4	W	0x00000000	GPIO4D iomux control
GRF_GPIO6B_IOMUX	0x010c	W	0x00000000	GPIO6B iomux control
GRF_GPIO0L_PULL	0x0118	W	0x00000000	GPIO0A / GPIO0B pull up/down control
GRF_GPIO0H_PULL	0x011c	W	0x00000000	GPIO0C / GPIO0D pull up/down control
GRF_GPIO1L_PULL	0x0120	W	0x00000000	GPIO0A / GPIO0B pull up/down control
GRF_GPIO1H_PULL	0x0124	W	0x00000000	GPIO1C / GPIO1D pull up/down control
GRF_GPIO2L_PULL	0x0128	W	0x00000000	GPIO2A / GPIO2B pull up/down control
GRF_GPIO2H_PULL	0x012c	W	0x00000000	GPIO2C / GPIO2D pull up/down control
GRF_GPIO3L_PULL	0x0130	W	0x00000000	GPIO3A / GPIO3B pull up/down control
GRF_GPIO3H_PULL	0x0134	W	0x00000000	GPIO3C / GPIO3D pull up/down control
GRF_GPIO4L_PULL	0x0138	W	0x00000000	GPIO4A / GPIO4B pull up/down control
GRF_GPIO4H_PULL	0x013c	W	0x00000000	GPIO4C / GPIO4D pull up/down control

Name	Offset	Size	Reset Value	Description
GRF_GPIO6L_PULL	0x0148	W	0x00000000	GPIO6A / GPIO6B pull up/down control
GRF_SOC_CON0	0x0150	W	0x00000008	soc control register
GRF_SOC_CON1	0x0154	W	0x00000006	soc control register
GRF_SOC_CON2	0x0158	W	0x00000000	soc control register
GRF_SOC_STATUS0	0x015c	W	0x00000000	soc status register
GRF_DMAC1_CON0	0x0160	W	0x00000002	DMAC1 control register
GRF_DMAC1_CON1	0x0164	W	0x00000000	DMAC1 control register
GRF_DMAC1_CON2	0x0168	W	0x00000000	DMAC1 control register
GRF_DMAC2_CON0	0x016c	W	0x0000fffe	DMAC2 control register
GRF_DMAC2_CON1	0x0170	W	0x00003fff	DMAC2 control register
GRF_DMAC2_CON2	0x0174	W	0x0000000f	DMAC2 control register
GRF_DMAC2_CON3	0x0178	W	0x00003fff	DMAC2 control register
GRF_UOC0_CON0	0x017c	W	0x0000c963	otg0 control register
GRF_UOC0_CON1	0x0180	W	0x000016fb	otg0 control register
GRF_UOC0_CON2	0x0184	W	0x00000408	otg0 control register
GRF_UOC1_CON0	0x0188	W	0x0000c963	otg1 control register
GRF_UOC1_CON1	0x018c	W	0x000016fb	otg1 control register
GRF_UOC1_CON2	0x0190	W	0x00000408	otg1 control register
GRF_UOC1_CON3	0x0194	W	0x0000001c	otg1 control register
GRF_DDRC_CON0	0x0198	W	0x00000000	DDRC control register
GRF_DDRC_STAT	0x019c	W	0x00000000	DDRC status
GRF_OS_REG0	0x01c8	W	0x00000000	OS register
GRF_OS_REG1	0x01cc	W	0x00000000	OS register
GRF_OS_REG2	0x01d0	W	0x00000000	OS register
GRF_OS_REG3	0x01d4	W	0x00000000	OS register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

7.3.2 Detail Register Description

GRF_GPIO0L_DIR
Address: Operational Base + offset (0x0000)
GPIO0A / GPIO0B output enable control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_dir GPIO0B output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio0a_dir GPIO0A output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO0H_DIR

Address: Operational Base + offset (0x0004)

GPIO0C / GPIO0D output enable control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0d_dir GPIO0D output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio0c_dir GPIO0C output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO1L_DIR
Address: Operational Base + offset (0x0008)
GPIO1A / GPIO0B output enable control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1b_dir GPIO1B output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio1a_dir GPIO1A output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO1H_DIR
Address: Operational Base + offset (0x000c)
GPIO1C / GPIO1D output enable control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1d_dir GPIO1D output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio1c_dir GPIO1C output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO2L_DIR

Address: Operational Base + offset (0x0010)

GPIO2A / GPIO2B output enable control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2b_dir GPIO2B output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio2a_dir GPIO2A output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO2H_DIR
Address: Operational Base + offset (0x0014)
GPIO2C / GPIO2D output enable control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio2d_dir GPIO2D output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio2c_dir GPIO2C output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO3L_DIR
Address: Operational Base + offset (0x0018)
GPIO3A / GPIO3B output enable control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio3b_dir GPIO3B output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio3a_dir GPIO3A output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO3H_DIR
Address: Operational Base + offset (0x001c)
GPIO3C / GPIO3D output enable control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio3d_dir GPIO3D output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio3c_dir GPIO3C output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO4L_DIR
Address: Operational Base + offset (0x0020)
GPIO0A / GPIO0B output enable control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio4b_dir GPIO4B output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio4a_dir GPIO4A output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO4H_DIR
Address: Operational Base + offset (0x0024)
GPIO4C / GPIO4D output enable control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio0d_dir GPIO0D output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio0c_dir GPIO0C output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO6L_DIR
Address: Operational Base + offset (0x0030)
GPIO6A / GPIO6B output enable control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	gpio6b_dir GPIO6B output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output
7:0	RW	0x00	gpio6a_dir GPIO6A output enable control When system is forced into power down mode, Values written to this register independently control the direction of the corresponding data bit in GPIO bits . 0: Input (default) 1:Output

GRF_GPIO0L_DO
Address: Operational Base + offset (0x0038)
GPIO0A / GPIO0B output control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_do GPIO0B output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

Bit	Attr	Reset Value	Description
7:0	RW	0x00	gpio0a_do GPIO0A output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO0H_DO
Address: Operational Base + offset (0x003c)
GPIO0C / GPIO0D output control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0d_do GPIO0D output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio0c_do GPIO0C output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO1L_DO
Address: Operational Base + offset (0x0040)
GPIO1A / GPIO1B output control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1b_do GPIO1B output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio1a_do GPIO1A output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO1H_DO
Address: Operational Base + offset (0x0044)
GPIO1C / GPIO1D output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1d_do GPIO1D output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio1c_do GPIO1C output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO2L_DO

Address: Operational Base + offset (0x0048)

GPIO2A / GPIO2B output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2b_do GPIO2B output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio2a_do GPIO2A output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO2H_DO

Address: Operational Base + offset (0x004c)

GPIO2C / GPIO2D output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2d_do GPIO2D output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio2c_do GPIO2C output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO3L_DO

Address: Operational Base + offset (0x0050)

GPIO3A / GPIO3B output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio3b_do GPIO3B output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio3a_do GPIO3A output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO3H_DO

Address: Operational Base + offset (0x0054)

GPIO3C / GPIO3D output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio3d_do GPIO3D output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio3c_do GPIO3C output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO4L_DO

Address: Operational Base + offset (0x0058)

GPIO4A / GPIO4B output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4b_do GPIO4B output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio4a_do GPIO4A output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO4H_DO

Address: Operational Base + offset (0x005c)

GPIO4C / GPIO4D output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4d_do GPIO4D output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio4c_do GPIO4C output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO6L_DO
Address: Operational Base + offset (0x0068)
GPIO6A / GPIO6B output control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio6b_do GPIO6B output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.
7:0	RW	0x00	gpio6a_do GPIO6A output control When system is forced into power down mode, Values written to this register are output on the I/O signals for GPIO if the corresponding data direction bits for GPIO are set to output mode.

GRF_GPIO0L_EN
Address: Operational Base + offset (0x0070)
GPIO0A / GPIO0B output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_en GPIO0B output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio0a_en GPIO0A output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO0H_EN
Address: Operational Base + offset (0x0074)
GPIO0C / GPIO0D output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0d_en GPIO0D output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio0c_en GPIO0C output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO1L_EN
Address: Operational Base + offset (0x0078)
GPIO1A / GPIO1B output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1b_en GPIO1B output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio1a_en GPIO1A output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO1H_EN
Address: Operational Base + offset (0x007c)
GPIO1C / GPIO1D output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1d_en GPIO1D output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio1c_en GPIO1C output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO2L_EN
Address: Operational Base + offset (0x0080)
GPIO2A / GPIO2B output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2b_en GPIO2B output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio2a_en GPIO2A output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO2H_EN

Address: Operational Base + offset (0x0084)

GPIO2C / GPIO2D output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2d_en GPIO2D output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio2c_en GPIO2C output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO3L_EN
Address: Operational Base + offset (0x0088)
GPIO3A / GPIO3B output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio3b_en GPIO3B output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio3a_en GPIO3A output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO3H_EN
Address: Operational Base + offset (0x008c)
GPIO3C / GPIO3D output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio3d_en GPIO3D output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio3c_en GPIO3C output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO4L_EN
Address: Operational Base + offset (0x0090)
GPIO4A / GPIO4B output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4b_en GPIO4B output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio4a_en GPIO4A output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO4H_EN
Address: Operational Base + offset (0x0094)
GPIO4C / GPIO4D output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4d_en GPIO4D output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio4c_en GPIO4C output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO6L_EN

Address: Operational Base + offset (0x00a0)

GPIO6A / GPIO6B output enable

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio6b_en GPIO6B output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register
7:0	RW	0x00	gpio6a_en GPIO6A output enable When system is forced into power down mode, Values written to this register will enable whether GPIO status will be controlled by GRF register

GRF_GPIO0A_IOMUX
Address: Operational Base + offset (0x00a8)
GPIO0A iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio0a7_sel GPIO0A[7] iomux select 1: i2s 8ch sdi 0: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio0a6_sel GPIO0A[6] iomux select 1: host_drv_vbus 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio0a5_sel GPIO0A[5] iomux select 1: otg_drv_vbus 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio0a4_sel GPIO0A[4] iomux select 1: pwm1 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio0a3_sel GPIO0A[3] iomux select 1: pwm0 0: gpio
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio0a2_sel GPIO0A[2] iomux select 1: hdmi_i2c_sda 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio0a1_sel GPIO0A[1] iomux select 1: hdmi_i2c_scl 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio0a0_sel GPIO0A[0] iomux select 1: hdmi_hot_plug_in 0: gpio

GRF_GPIO0B_IOMUX
Address: Operational Base + offset (0x00ac)
GPIO0B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio0b7_sel GPIO0B[7] iomux select 1: i2s 8ch sdo3 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio0b6_sel GPIO0B[6] iomux select 1: i2s 8ch sdo2 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio0b5_sel GPIO0B[5] iomux select 1: i2s 8ch sdo1 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio0b4_sel GPIO0B[4] iomux select 1: i2s 8ch sdo0 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio0b3_sel GPIO0B[3] iomux select 1: i2s 8ch lrck tx 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio0b2_sel GPIO0B[2] iomux select 1: i2s 8ch lrck_rx 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio0b1_sel GPIO0B[1] iomux select 1: i2s 8ch sclk 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio0b0_sel GPIO0B[0] iomux select 1: i2s_8ch_clk 0: gpio

GRF_GPIO0C_IOMUX
Address: Operational Base + offset (0x00b0)
GPIO0C iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio0c7_sel GPIO0C[7] iomux select 01: trace ctl 10: smc_addr3 00: gpio
13:12	RW	0x0	gpio0c6_sel GPIO0C[6] iomux select 01: trace clk 10: smc_addr2 00: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio0c5_sel GPIO0C[5] iomux select 1: i2s1 2ch sdo 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio0c4_sel GPIO0C[4] iomux select 1: i2s1 2ch sdi 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio0c3_sel GPIO0C[3] iomux select 1: i2s1 2ch lrck tx 0: gpio
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio0c2_sel GPIO0C[2] iomux select 1: i2s 8ch lrck_rx 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio0c1_sel GPIO0C[1] iomux select 1: i2s1 2ch sclk 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio0c0_sel GPIO0C[0] iomux select 1: i2s1 2ch clk 0: gpio

GRF_GPIO0D_IOMUX
Address: Operational Base + offset (0x00b4)
GPIO0D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio0d7_sel GPIO0D[7] iomux select 1: pwm3 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio0d6_sel GPIO0D[6] iomux select 1: pwm2 0: gpio
11:10	RW	0x0	gpio0d5_sel GPIO0D[5] iomux select 01: i2s2 2ch sdo 10: smc_addr1 00: gpio
9:8	RW	0x0	gpio0d4_sel GPIO0D[4] iomux select 01: i2s1 2ch sdi 10: smc_addr0 00: gpio
7:6	RW	0x0	gpio0d3_sel GPIO0D[3] iomux select 01: i2s1 2ch lrck tx 10: smc_adv_n 00: gpio
5:4	RW	0x0	gpio0d2_sel GPIO0D[2] iomux select 01: i2s1 2ch lrck_rx 10: smc_oe_n 0: gpio
3:2	RW	0x0	gpio0d1_sel GPIO0D[1] iomux select 01: i2s2 2ch sclk 10: smc_we_n 00: gpio
1:0	RW	0x0	gpio0d0_sel GPIO0D[0] iomux select 01: i2s2 2ch clk 10: smc_csn0 00: gpio

GRF_GPIO1A_IOMUX
Address: Operational Base + offset (0x00b8)
GPIO1A iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1a7_sel GPIO1A[7] iomux select 01: uart1_rts_n 10: spi0_txd 00: gpio
13:12	RW	0x0	gpio1a6_sel GPIO1A[6] iomux select 01: uart1_cts_n 10: spi0_rxd 00: gpio
11:10	RW	0x0	gpio1a5_sel GPIO1A[5] iomux select 01: uart1_sout 10: spi0_clk 00: gpio
9:8	RW	0x0	gpio1a4_sel GPIO1A[4] iomux select 01: uart1_sin 10: spi0_csn0 00: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio1a3_sel GPIO1A[3] iomux select 1: uart0_rts_n 0: gpio
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio1a2_sel GPIO1A[2] iomux select 1: uart0_cts_n 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio1a1_sel GPIO1A[1] iomux select 1: uart0 sout 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio1a0_sel GPIO1A[0] iomux select 1: uart0_sin 0: gpio

GRF_GPIO1B_IOMUX
Address: Operational Base + offset (0x00bc)
GPIO1B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio1b7_sel GPIO1B[7] iomux select 1: cif_data11 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio1b6_sel GPIO1B[6] iomux select 1: cif_data10 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio1b5_sel GPIO1B[5] iomux select 1: cif0_data1 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio1b4_sel GPIO1B[4] iomux select 1: cif0_data0 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio1b3_sel GPIO1B[3] iomux select 1: cif0_clkout 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio1b2_sel GPIO1B[2] iomux select 1: spdif_tx 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio1b1_sel GPIO1B[1] iomux select 1: uart2_sout 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio1b0_sel GPIO1B[0] iomux select 1: uart2_sin 0: gpio

GRF_GPIO1C_IOMUX
Address: Operational Base + offset (0x00c0)
GPIO1C iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio1c7_sel GPIO1C[7] iomux select 01: cif_data9 10: rmii_rxd0 00: gpio
13:12	RW	0x0	gpio1c6_sel GPIO1C[6] iomux select 01: cif_data8 10: rmii_rxd1 00: gpio
11:10	RW	0x0	gpio1c5_sel GPIO1C[5] iomux select 01:cif_data7 10:rmii_crs_dvalid 00: gpio
9:8	RW	0x0	gpio1c4_sel GPIO1C[4] iomux select 01:cif_data6 10:rmii_rx_err 00: gpio
7:6	RW	0x0	gpio1c3_sel GPIO1C[3] iomux select 01: cif_data5 10: rmii_txd0 00: gpio

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio0c2_sel GPIO0C[2] iomux select 01: cif1_data4 10: rmii_txd1 00:gpio
3:2	RW	0x0	gpio1c1_sel GPIO1C[1] iomux select 01: cif_data3 10: rmii_tx_en 00: gpio
1:0	RW	0x0	gpio1c0_sel GPIO1C[0] iomux select 01: cif1_data2 10: rmii_clkout 11: rmii_clkin 00: gpio

GRF_GPIO1D_IOMUX
Address: Operational Base + offset (0x00c4)
GPIO1D iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio1d7_sel GPIO1D[7] iomux select 1: cif1_clkout 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio1d6_sel GPIO1D[6] iomux select 1: cif1_data11 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio1d5_sel GPIO1D[5] iomux select 1: cif1_data10 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio1d4_sel GPIO1D[4] iomux select 1: cif1_data1 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio1d3_sel GPIO1D[3] iomux select 1:cif1_data0 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio1d2_sel GPIO1D[2] iomux select 1: cif1_clkin 0: gpio
3:2	RW	0x0	gpio1d1_sel GPIO1D[1] iomux select 01: cif1_href 10: mii_mdclk 00: gpio
1:0	RW	0x0	gpio1d0_sel GPIO1D[0] iomux select 01: cif1_vsync 10: mii_md 00: gpio

GRF_GPIO2A_IOMUX
Address: Operational Base + offset (0x00c8)
GPIO2A iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2a7_sel GPIO2A[7] iomux select 01: lcdc1_data7 10: smc_addr11 00: gpio
13:12	RW	0x0	gpio2a6_sel GPIO2A[6] iomux select 01:lcdc1_data6 10: smc_addr10 00: gpio
11:10	RW	0x0	gpio2a5_sel GPIO2A[5] iomux select 01: lcdc1_data5 10: smc_addr9 00: gpio
9:8	RW	0x0	gpio2a4_sel GPIO2A[4] iomux select 01:lcdc1_data4 10:smc_addr8 00: gpio
7:6	RW	0x0	gpio2a3_sel GPIO2A[3] iomux select 01: lcdc_data3 10: smc_addr7 00: gpio

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio2a2_sel GPIO2A[2] iomux select 01: lcdc_data2 10: smc_addr6 00: gpio
3:2	RW	0x0	gpio2a1_sel GPIO2A[1] iomux select 01: lcdc1_data1 10:smc_addr5 00: gpio
1:0	RW	0x0	gpio2a0_sel GPIO2A[0] iomux select 01: lcdc1_data0 10: smc_addr4 00: gpio

GRF_GPIO2B_IOMUX
Address: Operational Base + offset (0x00cc)
GPIO2B iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2b7_sel GPIO2B[7] iomux select 01: lcdc1_data15 10: smc_addr19 11: hsadc_data7 00: gpio

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio2b6_sel GPIO2B[6] iomux select 01: lcdc1_data14 10:smc_addr18 11:ts_sync 00: gpio
11:10	RW	0x0	gpio2b5_sel GPIO2B[5] iomux select 01:lcdc1_data13 10: smc_addr17 11:hsadc_data8 00: gpio
9:8	RW	0x0	gpio2b4_sel GPIO2B[4] iomux select 01: lcdc1_data12 10: smc_addr16 11: hsadc_data9 00: gpio
7:6	RW	0x0	gpio2b3_sel GPIO2B[3] iomux select 01: lcdc1_data11 10: smc_addr15 00: gpio
5:4	RW	0x0	gpio2b2_sel GPIO2B[2] iomux select 01: lcdc1_data10 10: smc_addr14 0: gpio
3:2	RW	0x0	gpio2b1_sel GPIO2B[1] iomux select 01: lcdc1_data9 10: smc_addr13 00: gpio
1:0	RW	0x0	gpio2b0_sel GPIO2B[0] iomux select 01: lcdc1_data8 10: smc_addr12 00: gpio

GRF_GPIO2C_IOMUX
Address: Operational Base + offset (0x00d0)
GPIO2C iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio2c7_sel GPIO2C[7] iomux select 01: lcdc1_data23 10: spi1_csn1 11:hsadc_data4 00: gpio
13:12	RW	0x0	gpio2c6_sel GPIO2C[6] iomux select 01: lcdc1_data22 10: spi1_rxd 11:hsadc_data3 00: gpio
11:10	RW	0x0	gpio2c5_sel GPIO2C[5] iomux select 01: lcdc1_data21 10: spi1_txd 11:hsadc_data2 00: gpio
9:8	RW	0x0	gpio2c4_sel GPIO2C[4] iomux select 01: lcdc1_data20 10:spi1_csn0 11:hsadc_data1 00: gpio

Bit	Attr	Reset Value	Description
7:6	RW	0x0	gpio2c3_sel GPIO2C[3] iomux select 01: lcdc1_data19 10:spi1_clk 11:hsadc_data0 00: gpio
5:4	RW	0x0	gpio2c2_sel GPIO2C[2] iomux select 01: lcdc1_data18 10: smc_bls_n1 11:hsadc_data5 00: gpio
3:2	RW	0x0	gpio2c1_sel GPIO2C[1] iomux select 01: lcdc1_data17 10: smc_bls_n0 11:hsadc_data6 00: gpio
1:0	RW	0x0	gpio2c0_sel GPIO2C[0] iomux select 01:lcdc_data16 10:gps_clk 11:hsadc_clkout 00: gpio

GRF_GPIO2D_IOMUX
Address: Operational Base + offset (0x00d4)
GPIO2D iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio2d7_sel GPIO2D[7] iomux select 1: i2c1_scl 0: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio2d6_sel GPIO2D[6] iomux select 1: i2c1_sda 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio2d5_sel GPIO2D[5] iomux select 1: i2c0_scl 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio2d4_sel GPIO2D[4] iomux select 1:i2c0_sda 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio2d3_sel GPIO2D[3] iomux select 1: lcdc1_vsync 0: gpio
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio2d2_sel GPIO2D[2] iomux select 1: lcdc1_hsync 0: gpio
3:2	RW	0x0	gpio2d1_sel GPIO2D[1] iomux select 01: lcdc1_den 10: smc_csn1 00: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio2d0_sel GPIO2D[0] iomux select 1: lcdc1_dclk 0: gpio

GRF_GPIO3A_IOMUX
Address: Operational Base + offset (0x00d8)
GPIO3A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio3a7_sel GPIO3A[7] iomux select 1: sdmmc0_write_prt 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio3a6_sel GPIO3A[6] iomux select 1:sdmmc0_rstn_out 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio3a5_sel GPIO3A[5] iomux select 1: i2c4_scl 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio3a4_sel GPIO3A[4] iomux select 1: i2c4_sda 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio3a3_sel GPIO3A[3] iomux select 1: i2c3_scl 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio3a2_sel GPIO3A[2] iomux select 1: i2c3_sda 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio3a1_sel GPIO3A[1] iomux select 1: i2c2_scl 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio3a0_sel GPIO3A[0] iomux select 1: i2c2_sda 0: gpio

GRF_GPIO3B_IOMUX
Address: Operational Base + offset (0x00dc)
GPIO3B iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio3b7_sel GPIO3B[7] iomux select 1: sdmmc0_write_prt 0: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio3b6_sel GPIO3B[6] iomux select 1: sdmmc0_detect_n 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio3b5_sel GPIO3B[5] iomux select 1: sdmmc0_data3 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio3b4_sel GPIO3B[4] iomux select 1: sdmmc0_data2 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio3b3_sel GPIO3B[3] iomux select 1: sdmmc0_data1 0: gpio
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio3b2_sel GPIO3B[2] iomux select 1: sdmmc0_data0 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio3b1_sel GPIO3B[1] iomux select 1: sdmmc0_cmd 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio3b0_sel GPIO3B[0] iomux select 1: sdmmc0_clkout 0: gpio

GRF_GPIO3C_IOMUX
Address: Operational Base + offset (0x00e0)
GPIO3C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio3c7_sel GPIO3C[7] iomux select 1: sdmmc1_write_prt 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio3c6_sel GPIO3C[6] iomux select 1: sdmmc1_detect_n 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio3c5_sel GPIO3C[5] iomux select 1: sdmmc1_clkout 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio3c4_sel GPIO3C[4] iomux select 1: sdmmc1_data3 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio3c3_sel GPIO3C[3] iomux select 1: sdmmc1_data2 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio3c2_sel GPIO3C[2] iomux select 1: sdmmc1_data1 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio3c1_sel GPIO3C[1] iomux select 1: sdmmc1_data0 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio3c0_sel GPIO3C[0] iomux select 1: smmc1_cmd 0: gpio

GRF_GPIO3D_IOMUX
Address: Operational Base + offset (0x00e4)
GPIO3D iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio3d7_sel GPIO3D[7] iomux select 01: flash_dqs 10: emmc_clkout 00: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio3d6_sel GPIO3D[6] iomux select 1: uart3_rts_n 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio3d5_sel GPIO3D[5] iomux select 1: uart3_cts_n 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio3d4_sel GPIO3D[4] iomux select 1: uart3_sout 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio3d3_sel GPIO3D[3] iomux select 1:uart3_sin 0: gpio
5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	gpio3d2_sel GPIO3D[2] iomux select 1: sdmmc1_int_n 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio3d1_sel GPIO3D[1] iomux select 1: sdmmc1_backend_pwr 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio3d0_sel GPIO3D[0] iomux select 1: sdmmc1_pwr_en 0: gpio

GRF_GPIO4A_IOMUX
Address: Operational Base + offset (0x00e8)
GPIO4A iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio4a7_sel GPIO4A[7] iomux select 1: flash_data15 0: gpio
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	gpio4a6_sel GPIO4A[6] iomux select 1: flash_data14 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio4a5_sel GPIO4A[5] iomux select 1: flash_data13 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio4a4_sel GPIO4A[4] iomux select 1: flash_data12 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio4a3_sel GPIO4A[3] iomux select 1: flash_data11 0: gpio
5	RO	0x0	reserved
4	RW	0x0	gpio4a2_sel GPIO4A[2] iomux select 1: flash_data10 0: gpio
3	RO	0x0	reserved
2	RW	0x0	gpio4a1_sel GPIO4A[1] iomux select 1: flash_data9 0: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio4a0_sel GPIO4A[0] iomux select 1: flash_data8 0: gpio

GRF_GPIO4B_IOMUX
Address: Operational Base + offset (0x00ec)
GPIO4B iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio4b7_sel GPIO4B[7] iomux select 1: spi0_csn1 0: gpio
13	RO	0x0	reserved
12	RW	0x0	gpio4b6_sel GPIO4B[6] iomux select 1: flash_csn7 0: gpio
11	RO	0x0	reserved
10	RW	0x0	gpio4b5_sel GPIO4B[5] iomux select 1: flash_csn6 0: gpio
9	RO	0x0	reserved
8	RW	0x0	gpio4b4_sel GPIO4B[4] iomux select 1: flash_csn5 0: gpio
7	RO	0x0	reserved
6	RW	0x0	gpio4b3_sel GPIO4B[3] iomux select 1: flash_csn4 0: gpio

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio4b2_sel GPIO4B[2] iomux select 01: flash_csn3 10:emmc_rstn_out 00: gpio
3:2	RW	0x0	gpio4b1_sel GPIO4B[1] iomux select 01: flash_csn2 10: emmc_cmd 00: gpio
1	RO	0x0	reserved
0	RW	0x0	gpio4b0_sel GPIO4B[0] iomux select 1: flash_csn1 0: gpio

GRF_GPIO4C_IOMUX
Address: Operational Base + offset (0x00f0)
GPIO4C iomux control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4c7_sel GPIO4[7] iomux select 01: smc_data7 10: trace_data7 00: gpio

Bit	Attr	Reset Value	Description
13:12	RW	0x0	gpio4c6_sel GPIO4C[6] iomux select 01: smc_data6 10: trace_data6 00: gpio
11:10	RW	0x0	gpio4c5_sel GPIO4C[5] iomux select 01:smc_data5 10:trace_data5 00:gpio
9:8	RW	0x0	gpio4c4_sel GPIO4C[4] iomux select 01: smc_data4 10: trace_data4 00: gpio
7:6	RW	0x0	gpio4c3_sel GPIO4C[3] iomux select 01: smc_data3 10: trace_data3 00: gpio
5:4	RW	0x0	gpio4c2_sel GPIO4C[2] iomux select 01: smc_data2 10: trace_data2 00: gpio
3:2	RW	0x0	gpio4c1_sel GPIO4C[1] iomux select 01:smc_data1 10:trace_data1 00: gpio
1:0	RW	0x0	gpio4c0_sel GPIO4C[0] iomux select 01: smc_data0 10:trace_data0 00: gpio

GRF_GPIO4D_IOMUX
Address: Operational Base + offset (0x00f4)
GPIO4D iomux control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:14	RW	0x0	gpio4d7_sel GPIO4D[7] iomux select 10: smc_data15 01: trace_data15 00: gpio
13:12	RW	0x0	gpio4d6_sel GPIO4D[6] iomux select 01: smc_data14 10: trace_data14 00: gpio
11:10	RW	0x0	gpio4d5_sel GPIO4D[5] iomux select 01: smc_data13 10: trace_data13 00: gpio
9:8	RW	0x0	gpio4d4_sel GPIO4D[4] iomux select 01: smc_data12 10: trace_data12 00: gpio
7:6	RW	0x0	gpio4d3_sel GPIO4D[3] iomux select 01: smc_data11 10: trace_data11 00: gpio

Bit	Attr	Reset Value	Description
5:4	RW	0x0	gpio4d2_sel GPIO4D[2] iomux select 01: smc_data10 10: trace_data10 00: gpio
3:2	RW	0x0	gpio4d1_sel GPIO4D[1] iomux select 01: smc_data9 10:trace_data9 00: gpio
1:0	RW	0x0	gpio4d0_sel GPIO4D[0] iomux select 01: smc_data8 10: trace_data8 00: gpio

GRF_GPIO6B_IOMUX
Address: Operational Base + offset (0x010c)
GPIO6B iomux control

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	gpio6b7_sel GPIO6B[7] iomux select 1: test_clock_out 0: gpio
13:0	RO	0x0	reserved

GRF_GPIO0L_PULL
Address: Operational Base + offset (0x0118)
GPIO0A / GPIO0B pull up/down control

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_pull GPIO0B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio0a_pull GPIO0A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO0H_PULL
Address: Operational Base + offset (0x011c)
GPIO0C / GPIO0D pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0d_pull GPIO0D pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio0c_pull GPIO0C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO1L_PULL
Address: Operational Base + offset (0x0120)
GPIO0A / GPIO0B pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio0b_pull GPIO0B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio0a_pull GPIO0A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO1H_PULL
Address: Operational Base + offset (0x0124)
GPIO1C / GPIO1D pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio1d_pull GPIO1d pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio1c_pull GPIO1C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO2L_PULL
Address: Operational Base + offset (0x0128)
GPIO2A / GPIO2B pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2b_pull GPIO2B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio2a_pull GPIO2A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO2H_PULL

Address: Operational Base + offset (0x012c)

GPIO2C / GPIO2D pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio2d_pull GPIO2d pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio2c_pull GPIO2C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO3L_PULL

Address: Operational Base + offset (0x0130)

GPIO3A / GPIO3B pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio3b_pull GPIO3B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio3a_pull GPIO3A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO3H_PULL

Address: Operational Base + offset (0x0134)

GPIO3C / GPIO3D pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio3d_pull GPIO3d pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio3c_pull GPIO3C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO4L_PULL
Address: Operational Base + offset (0x0138)
GPIO4A / GPIO4B pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4b_pull GPIO4B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio4a_pull GPIO4A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO4H_PULL

Address: Operational Base + offset (0x013c)

GPIO4C / GPIO4D pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio4d_pull GPIO4d pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio4c_pull GPIO4C pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_GPIO6L_PULL
Address: Operational Base + offset (0x0148)
GPIO6A / GPIO6B pull up/down control

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RW	0x00	gpio6b_pull GPIO6B pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable
7:0	RW	0x00	gpio6a_pull GPIO6A pull up/down enable Values written to this register independently control Pullup/Pulldown or not for the corresponding data bit in GPIO. 0: pull up/down enable, PAD type will decide to be up or down ,not related with this value 1: pull up/down disable

GRF_SOC_CON0

Address: Operational Base + offset (0x0150)

soc control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit 0~bit 15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x0	hdmi_ext_sel external hdmi select 0: hdmi in SoC is used 1: hdmi in SoC is not used
14	RW	0x0	hdmi_video_sel hdmi video source select 0: hdmi get rgb datas from LCDC0 1: hdmi get rga datas from LCDC1
13	RW	0x0	smc_mux_con When high, the smc memory interface operates in multiplexed address/data mode.
12	RW	0x0	soc_remap remap bit control When soc_remap = 1, the bootrom is mapped to address 0x10100000 and internal memory is mapped to address 0x0.
11	RW	0x0	emmc_flash_sel emmc flash select used for iomux IO_FLASH_DATA[7:0] , IO_FLASH_WP are selected for emmc instead of flash
10:7	RW	0x0	tzpc_revision

Bit	Attr	Reset Value	Description
6:5	RW	0x0	l2cache_acc L2 Cache access control 00: used as L2 cache accessed by CPU. 10: 256KB is for L2 cahce accessed by CPU , another 256KB is for internal memory accessed by AXI 11/01: used as internal mmeory accessed by AXI
4:3	RW	0x1	l2rd_wait L2 Cache read wait cycle 00: the rdata of L2 cache will be captured at the next cycle after read command valid when used as internal memory. 01: the rdata of L2 cache will be captured one cycle later after read command valid when used as internal memory. 10: the rdata of L2 cache will be captured two cycle later after read command valid when used as internal memory. 11: the rdata of L2 cache will be captured three cycle later after read command valid when used as internal memory.
2:1	RW	0x0	imemrd_wait IMEM read wait cycle 00: the rdata of internal memory will be captured at the next cycle after read command valid 01: the rdata of internal memory will be captured one cycle later after read command valid 10: the rdata of internal memory will be captured two cycle later after read command valid 11: the rdata of internal memory will be captured three cycle later after read command valid
0	RO	0x0	reserved

GRF_SOC_CON1

Address: Operational Base + offset (0x0154)

soc control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit 0~bit 15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:11	RW	0x00	rki2c_sel 0: use old i2c 1: rki2c is used instead of old i2c
10	RW	0x0	vpu_sel vdpu vepu clock select 0: select vepu aclk as vpu main clock 1: select vdpu aclk as vpu main clock
9	RW	0x0	peri_emem_pause peri emem ahb bus arbiter pause control PERI AHB bus arbiter pause control
8	RW	0x0	peri_usb_pause peri usb ahb bus arbiter pause control USB AHB bus arbiter pause control
7	RO	0x0	reserved
6	RW	0x0	smc_mux_mode_0 When high, the smc memory interface operates in multiplexed address/data mode.
5:4	RW	0x0	smc_sram_mw_0 Sets the memory width for smc chip select0, on memory interface 0, when remap_0(grf_soc_con[14]) is high, the encoding is : 00: 8 bit 01: 16 bit 10: 32bit 11: reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	smc_remap_0 If high, remap smc controller chip select 0 , on memory interface 0 , to address 0x0
2	RW	0x1	smc_a_gt_m0_sync Indicate that if aclk is faster than and synchronous to mclk0 in smc controller 1: aclk is faster than and synochrounous to mclk0
1	RW	0x0	emac_speed 0:10MHz speed mode 1:100MHz speed mode
0	RW	0x0	emac_mode 0: rmii 1: mii

GRF_SOC_CON2
Address: Operational Base + offset (0x0158)
soc control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved
7	RW	0x0	msch4_mainddr3 When DDR3 is used , software should configure this bit to 1.
6	RO	0x0	reserved
5	RW	0x0	sw_addr15_en 1 : The axiaddr[15] of L2C axi bus 1 is switched with axiaddr[12] of L2C axi bus 1

Bit	Attr	Reset Value	Description
4	RW	0x0	sw_addr16_en 1 : The axiaddr[16] of L2C axi bus 1 is switched with axiaddr[13] of L2C axi bus 1
3	RW	0x0	sw_addr17_en 1 : The axiaddr[17] of L2C axi bus 1 is switched with axiaddr[14] of L2C axi bus 1
2	RW	0x0	bank2_to_rank_en 1: bank[2] connect to rank
1	RW	0x0	rank_to_row15_en 1: ddr rank bit connect to row[15]
0	RW	0x0	upctl_c_active_in ddr clock active in. External signal from system that flags if a hardware low power request can be accepted or should always be denied. 0: may be accepted 1: will be denied

GRF_SOC_STATUS0
Address: Operational Base + offset (0x015c)
soc status register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:23	RW	0x0	host20_linestate host 2.0 linestate status
22	RW	0x0	host20_bvalid host 2.0 bvalid status
21	RW	0x0	host20_vbusvalid host 2.0 vbus valid status
20	RW	0x0	otg0_iddig otg iddig status 0: indicate otg work as host 1: indicate otg work as device
19:18	RW	0x0	otg_linestate otg linestate status
17	RW	0x0	otg_bvalid otg bvalid status

Bit	Attr	Reset Value	Description
16	RW	0x0	otg_vbusvalid otg vbus valid status
15:8	RO	0x00	cru_rfslip_status RFSLIP and FBSLIP status : generalpll_fbslip, codecpll_fbslip, ddrpll_fbslip, armpll_fbslip, generalpll_rfslip, codecpll_rfslip, ddrpll_rfslip, armpll_rfslip. RFSLIP - Reference cycle slip output, indicating out of lock for PLL FBSLIP - Feedback cycle slip output, indicating out of lock for PLL
7:4	RO	0x0	pll_lock Pll lock status :generalpll_lock, codecpll_lock,armpll_lock,ddrpll_lock
3	RO	0x0	reserved
2:0	RO	0x0	timer_en_status

GRF_DMAC0_CON0
Address: Operational Base + offset (0x0160)
DMAC0 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RW	0x0	dmac0_boot_addr dmac0_boot_addr[18:16] DMAC0 boot_addr[18:16] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.
3	RO	0x0	reserved
2:1	RW	0x1	dmac0_drtype DMAC0 type of acknowledgement or request for peripheral signals: 00 : single level request 01 ： burst level request 10 ： acknowledging a flush request 11 ： reserved
0	RW	0x0	dmac0_boot_from_pc DMAC0 boot_from_pc input control Controls the location in which the DMAC0 executes its initial instruction, after it exits from reset : 0= DMAC0 waits for an instruction from APB interface 1= DMAC manager thread executes the instruction that is located at the address that boot_addr[31:0] provided.

GRF_DMAC0_CON1
Address: Operational Base + offset (0x0164)
DMAC0 control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	dmac0_boot_addr dmac0_boot_addr[15:0] DMAC0 boot_addr[15:0] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

GRF_DMAC0_CON2

Address: Operational Base + offset (0x0168)

DMAC0 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:8	RW	0x00	dmac0_boot_irq_ns dmac0_boot_irq_ns[11:4] DMAC1 boot_irq_ns[11:4] input control Controls the security state of an event-interrupt resource , when the DMAC1 exits from reset. Note : DMAC1 don't support secure feature, these bits don't need to be configured.
7:0	RW	0x00	dmac0_boot_periph_ns dmac0_boot_periph_ns[11:4] DMAC1 boot_peri_ns[11:4] input control Controls the security state of a peripheral request interface, when the DMAC1 exits from reset. Note: DMAC1 don't support secure feature, these bits don't need to be configured.

GRF_DMAC1_CON0
Address: Operational Base + offset (0x016c)
DMAC1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:2	RW	0x3fff	dmac1_boot_irq_ns DMAC1 boot_irq_ns input control Controls the security state of an event-interrupt resource , when the DMAC1 exits from reset. Note: DMAC1 don't support secure feature , these bits don't need to be configured.

Bit	Attr	Reset Value	Description
1	RW	0x1	dmac1_boot_manager_ns DMAC1 boot_manager_ns input control When the DMAC1 exits from reset , this signal controls the security state of the DMA manager thread: 0 = assigns DMA manager to the secure state 1 = assigns DMA manager to the Non-secure state
0	RW	0x0	dmac1_boot_from_pc DMAC1 boot_from_pc input control Controls the location in which the DMAC1 executes its initial instruction, after it exits from reset : 0= DMAC1 waits for an instruction from APB interface 1= DMAC manager thread executes the instruction that is located at the address that boot_addr[31:0] provided.

GRF_DMAC1_CON1
Address: Operational Base + offset (0x0170)
DMAC1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x3fff	dmac1_boot_addr dmac1_boot_addr[27:12] DMAC1 boot_addr[27:12] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

GRF_DMAC1_CON2

Address: Operational Base + offset (0x0174)

DMAC1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:10	RO	0x0	reserved
9:6	RW	0x0	dmac1_boot_periph_ns dmac1_boot_periph_ns[19:16] DMAC1 boot_peri_ns[19:16] input control Controls the security state of a peripheral request interface , when the DMAC1 exits from reset. Note: DMAC1 don't support secure feature, these bits don't need to be configured.
5:4	RW	0x0	dmac1_drtype DMAC1 type of acknowledgement or request for peripheral signals: 00 : single level request 01 ： burst level request 10 ： acknowledging a flush request 11 ： reserved

Bit	Attr	Reset Value	Description
3:0	RW	0xf	dmac1_boot_addr dmac1_boot_addr[31:28] DMAC1 boot_addr[31:28] input control Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

GRF_DMAC1_CON3
Address: Operational Base + offset (0x0178)
DMAC1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x3fff	dmac1_boot_perph_ns DMAC1 boot_peri_ns[19:16] input control Controls the security state of a peripheral request interface , when the DMAC1 exits from reset. Note: DMAC1 don't support secure feature, these bits don't need to be configured.

GRF_UOC0_CON0
Address: Operational Base + offset (0x017c)
otg0 control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	usbphy_txbitstuff_enh High-Byte Transmit Bit-Stuffing Enable Function: This controller signal controls bit stuffing on DATAINH[7:0] when OPMODE[1:0] = 2'b11. 1: Bit stuffing is enabled. 0: Bit stuffing is disabled.
14	RW	0x1	usbphy_txbitstuff_en Low-Byte Transmit Bit-Stuffing Enable Function: This controller signal controls bit stuffing on DATAINH[7:0] when OPMODE[1:0] = 2'b11. 1: Bit stuffing is enabled. 0: Bit stuffing is disabled.
13	RW	0x0	usbphy_siddq IDDQ Test Enable Function: This test signal enables you to perform IDDQ testing by powering down all analog blocks. 1: The analog blocks are powered down. 0: The analog blocks are powered up.

Bit	Attr	Reset Value	Description
12	RW	0x0	usbphy_port_reset Per-Port Reset Function: When asserted, this customer-specific signal resets the corresponding port transmit and receive logic without disabling the clocks within the USB 2.0 nanoPHY. 1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles.
11:10	RW	0x2	usbphy_refclk_sel Reference Clock Select for PLL Block Function: This signal selects the reference clock source for the PLL block. 11: The PLL uses CLKCORE as reference. 10: The PLL uses CLKCORE as reference. 01: The XO block uses an external, 2.5-V clock supplied on the XO pin. 00: The XO block uses the clock from a crystal.
9:8	RW	0x1	usbphy_refclk_div Reference Clock Frequency Select Function: This bus selects the USB 2.0 nanoPHY reference clock frequency. 11: 19.2 MHz 10: 48 MHz 01: 24 MHz 00: 12 MHz

Bit	Attr	Reset Value	Description
7:5	RW	0x3	usbphy_otg_tune VBUS Valid Threshold Adjustment Function: This bus adjusts the voltage level for the VBUS Valid threshold. 111: + 9% 110: + 6% 101: + 3% 100: Design default 011:- 3% 010:- 6% 001: - 9% 000: - 12%
4	RW	0x0	usbphy_otg_disable OTG Block Disable
3:1	RW	0x1	usbphy_compdistune Disconnect Threshold Adjustment Function: This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host. 111: + 4.5% 110: + 3% 101: + 1.5% 100: Design default 011: - 1.5% 010: - 3% 001: - 4.5% 000: - 6%
0	RW	0x1	usb_phy_common_on_n Common Block Power-Down Control Function: This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 PHY is in Suspend or Sleep mode. 1: In Suspend mode, the XO, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 0: In Suspend mode, the XO, Bias, and PLL blocks remain powered in Suspend mode. In Sleep mode, if the reference clock is a crystal, the XO block remains powered.

GRF_UOC0_CON1
Address: Operational Base + offset (0x0180)
otg0 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	usbphy_txrise_tune HS Transmitter Rise/Fall Time Adjustment Function: This bus adjusts the rise/fall times of the high-speed waveform. 1: - 8% 0: Design default
13:12	RW	0x1	usbphy_txhsxv_tune Transmitter High-Speed Crossover Adjustment Function: This bus adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. 11: Default setting 10: + 15 mV 01: - 15 mV 00: Reserved

Bit	Attr	Reset Value	Description
11:8	RW	0x6	usbphy_txvref_tune HS DC Voltage Level Adjustment Function: This bus adjusts the high-speed DC level voltage. 1111: + 8.75% 1110: + 7.5% 1101: + 6.25% 1100: + 5% 1011: + 3.75% 1010: + 2.5% 1001: + 1.25% 1000: Design default 0111: -1.25% 0110: -2.5% 0101: - 3.75% 0100: - 5% 0011: - 6.25% 0010: - 7.5% 0001: - 8.75% 0000: - 10%
7:4	RW	0xf	usbphy_txfsls_tune FS/LS Source Impedance Adjustment Function: This bus adjusts the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 1111: - 5% 0111: -2.5% 0011: Design default 0001: + 2.5% 0000: + 5%
3	RW	0x1	usbphy_txfreemphasis_tune HS Transmitter Pre-Emphasis Enable Function: This signal controls the pre-emphasis for a J-K or K-J state transition in HS mode. 1: The HS Transmitter pre-emphasis is enabled. 0 (design default): The HS Transmitter pre-emphasis is disabled

Bit	Attr	Reset Value	Description
2:0	RW	0x3	usbphy_sqrxtune Squelch Threshold Adjustment Function: This bus adjusts the voltage level for the threshold used to detect valid high-speed data. 111: - 20% 110: -15% 101: -10% 100: -5% 011: Design default 010: + 5% 001: + 10% 000: + 15%

GRF_UOC0_CON2
Address: Operational Base + offset (0x0184)
otg0 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable
15:13	RO	0x0	reserved
12:11	RW	0x0	scaledown Scale-Down Mode 00: Disables all scale-downs. Actual timing values are used. Required for synthesis. 01: Enables scale-down of all timing values except Device mode suspend and resume. These include: - Speed enumeration. - HNP/SRP. - Host mode suspend and resume. 10: Enables scale-down of Device mode suspend and resume timing values only. 11: Enables bit 0 and bit 1 scale-down timing values.
10	RW	0x1	sleepm Sleep Assertion 1: Normal operating mode 0: Sleep mode

Bit	Attr	Reset Value	Description
9	RW	0x0	vregtune 2.5-V Voltage Regulator HS Boost Adjustment
8	RW	0x0	utmi_termselect USB Termination Select 1: Full-speed terminations are enabled. 0: High-speed terminations are enabled.
7:6	RW	0x0	utmi_xcvrselect Transceiver Select 11: Sends an LS packet on an FS bus or receives an LS packet. 10: LS Transceiver 01: FS Transceiver 00: HS Transceiver
5:4	RW	0x0	utmi_opmode UTMI+ Operational Mode Function: This controller bus selects the UTMI+ operational mode. 11: Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 00 while OPMODE[1:0] is set to 11, USB PHY behavior is undefined. 10: Disable bit stuffing and NRZI encoding 01: Non-Driving 00: Normal
3	RW	0x1	utmi_suspend_n Suspend Assertion 1: Normal operating mode 0: Suspend mode
2	RW	0x0	usbphy_soft_con_sel 0: software control usb phy disable 1 : software control usb phy enable
1	RW	0x0	usbphy_vbus_vld_extsel External VBUS Valid Select Function: This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. 1: The VBUSVLDEXT input is used. 0: The internal Session Valid comparator is used.

Bit	Attr	Reset Value	Description
0	RW	0x0	usbphy_vbus_vld_ext External VBUS Valid Indicator Function: This signal is valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXT enables the pullup resistor on the D+ line. 1: The VBUS signal is valid, and the pull-up resistor on D+ is enabled. 0: The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.

GRF_UOC1_CON0
Address: Operational Base + offset (0x0188)
otg1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RW	0x1	usbphy_txbitstuff_enh High-Byte Transmit Bit-Stuffing Enable Function: This controller signal controls bit stuffing on DATAINH[7:0] when OPMODE[1:0] = 2'b11. 1: Bit stuffing is enabled. 0: Bit stuffing is disabled.

Bit	Attr	Reset Value	Description
14	RW	0x1	usbphy_txbitstuff_en Low-Byte Transmit Bit-Stuffing Enable Function: This controller signal controls bit stuffing on DATAINH[7:0] when OPMODE[1:0] = 2'b11. 1: Bit stuffing is enabled. 0: Bit stuffing is disabled.
13	RW	0x0	usbphy_siddq IDDQ Test Enable Function: This test signal enables you to perform IDDQ testing by powering down all analog blocks. 1: The analog blocks are powered down. 0: The analog blocks are powered up.
12	RW	0x0	usbphy_port_reset Per-Port Reset Function: When asserted, this customer-specific signal resets the corresponding port transmit and receive logic without disabling the clocks within the USB 2.0 nanoPHY. 1: The transmit and receive finite state machines (FSMs) are reset, and the line_state logic combinatorially reflects the state of the single-ended receivers. 0: The transmit and receive FSMs are operational, and the line_state logic becomes sequential after 11 PHYCLOCK cycles.
11:10	RW	0x2	usbphy_refclk_sel Reference Clock Select for PLL Block Function: This signal selects the reference clock source for the PLL block. 11: The PLL uses CLKCORE as reference. 10: The PLL uses CLKCORE as reference. 01: The XO block uses an external, 2.5-V clock supplied on the XO pin. 00: The XO block uses the clock from a crystal.

Bit	Attr	Reset Value	Description
9:8	RW	0x1	usbphy_refclk_div Reference Clock Frequency Select Function: This bus selects the USB 2.0 nanoPHY reference clock frequency. 11: 19.2 MHz 10: 48 MHz 01: 24 MHz 00: 12 MHz
7:5	RW	0x3	usbphy_otg_tune VBUS Valid Threshold Adjustment Function: This bus adjusts the voltage level for the VBUS Valid threshold. 111: + 9% 110: + 6% 101: + 3% 100: Design default 011:- 3% 010:- 6% 001: - 9% 000: - 12%
4	RW	0x0	usbphy_otg_disable OTG Block Disable
3:1	RW	0x1	usbphy_compdistune Disconnect Threshold Adjustment Function: This bus adjusts the voltage level for the threshold used to detect a disconnect event at the host. 111: + 4.5% 110: + 3% 101: + 1.5% 100: Design default 011: - 1.5% 010: - 3% 001: - 4.5% 000: - 6%

Bit	Attr	Reset Value	Description
0	RW	0x1	usb_phy_common_on_n Common Block Power-Down Control Function: This signal controls the power-down signals in the XO, Bias, and PLL blocks when the USB 2.0 PHY is in Suspend or Sleep mode. 1: In Suspend mode, the XO, Bias, and PLL blocks are powered down. In Sleep mode, the Bias and PLL blocks are powered down. 0: In Suspend mode, the XO, Bias, and PLL blocks remain powered in Suspend mode. In Sleep mode, if the reference clock is a crystal, the XO block remains powered.

GRF_UOC1_CON1
Address: Operational Base + offset (0x018c)
otg1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15	RO	0x0	reserved
14	RW	0x0	usbphy_txrise_tune HS Transmitter Rise/Fall Time Adjustment Function: This bus adjusts the rise/fall times of the high-speed waveform. 1: - 8% 0: Design default

Bit	Attr	Reset Value	Description
13:12	RW	0x1	usbphy_txhsxv_tune Transmitter High-Speed Crossover Adjustment Function: This bus adjusts the voltage at which the DP and DM signals cross while transmitting in HS mode. 11: Default setting 10: + 15 mV 01: - 15 mV 00: Reserved
11:8	RW	0x6	usbphy_txvref_tune HS DC Voltage Level Adjustment Function: This bus adjusts the high-speed DC level voltage. 1111: + 8.75% 1110: + 7.5% 1101: + 6.25% 1100: + 5% 1011: + 3.75% 1010: + 2.5% 1001: + 1.25% 1000: Design default 0111: -1.25% 0110: -2.5% 0101: - 3.75% 0100: - 5% 0011: - 6.25% 0010: - 7.5% 0001: - 8.75% 0000: - 10%
7:4	RW	0xf	usbphy_txfsls_tune FS/LS Source Impedance Adjustment Function: This bus adjusts the low- and full-speed single-ended source impedance while driving high. The following adjustment values are based on nominal process, voltage, and temperature. 1111: - 5% 0111: -2.5% 0011: Design default 0001: + 2.5% 0000: + 5%

Bit	Attr	Reset Value	Description
3	RW	0x1	usbphy_txfreemphasis_tune HS Transmitter Pre-Emphasis Enable Function: This signal controls the pre-emphasis for a J-K or K-J state transition in HS mode. 1: The HS Transmitter pre-emphasis is enabled. 0 (design default): The HS Transmitter pre-emphasis is disabled
2:0	RW	0x3	usbphy_sqrxtune Squelch Threshold Adjustment Function: This bus adjusts the voltage level for the threshold used to detect valid high-speed data. 111: - 20% 110: -15% 101: -10% 100: -5% 011: Design default 010: + 5% 001: + 10% 000: + 15%

GRF_UOC1_CON2
Address: Operational Base + offset (0x0190)
otg1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10	RW	0x1	sleepm Sleep Assertion 1: Normal operating mode 0: Sleep mode
9	RW	0x0	vregtune 2.5-V Voltage Regulator HS Boost Adjustment
8	RW	0x0	utmi_termselect USB Termination Select 1: Full-speed terminations are enabled. 0: High-speed terminations are enabled.
7:6	RW	0x0	utmi_xcvrselect Transceiver Select 11: Sends an LS packet on an FS bus or receives an LS packet. 10: LS Transceiver 01: FS Transceiver 00: HS Transceiver
5:4	RW	0x0	utmi_opmode UTMI+ Operational Mode Function: This controller bus selects the UTMI+ operational mode. 11: Normal operation without SYNC or EOP generation. If the XCVRSEL bus is not set to 00 while OPMODE[1:0] is set to 11, USB PHY behavior is undefined. 10: Disable bit stuffing and NRZI encoding 01: Non-Driving 00: Normal
3	RW	0x1	utmi_suspend_n Suspend Assertion 1: Normal operating mode 0: Suspend mode
2	RW	0x0	usbphy_soft_con_sel 0: software control usb phy disable 1 : software control usb phy enable

Bit	Attr	Reset Value	Description
1	RW	0x0	usbphy_vbus_vld_extsel External VBUS Valid Select Function: This signal selects the VBUSVLDEXT input or the internal Session Valid comparator to indicate when the VBUS signal on the USB cable is valid. 1: The VBUSVLDEXT input is used. 0: The internal Session Valid comparator is used.
0	RW	0x0	usbphy_vbus_vld_ext External VBUS Valid Indicator Function: This signal is valid in Device mode and only when the VBUSVLDEXTSEL signal is set to 1. VBUSVLDEXT indicates whether the VBUS signal on the USB cable is valid. In addition, VBUSVLDEXT enables the pullup resistor on the D+ line. 1: The VBUS signal is valid, and the pull-up resistor on D+ is enabled. 0: The VBUS signal is not valid, and the pull-up resistor on D+ is disabled.

GRF_UOC1_CON3
Address: Operational Base + offset (0x0194)
otg1 control register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:6	RW	0x0	scaledown Scale-Down Mode 00: Disables all scale-downs. Actual timing values are used. Required for synthesis. 01: Enables scale-down of all timing values except Device mode suspend and resume. These include: - Speed enumeration. - HNP/SRP. - Host mode suspend and resume. 10: Enables scale-down of Device mode suspend and resume timing values only. 11: Enables bit 0 and bit 1 scale-down timing values.
5	RW	0x0	utmiotg_idpullup Analog ID Input Sample Enable Function: This controller signal controls ID line sampling. 1: ID pin sampling is enabled, and the IDDIG output is valid. 0: ID pin sampling is disabled, and the IDDIG output is not valid.
4	RW	0x1	utmiotg_dppulldown D+ Pull-Down Resistor Enable
3	RW	0x1	utmiotg_dmpulldown D- Pull-Down Resistor Enable
2	RW	0x1	utmiotg_drvvbus Drive VBUS 1: The VBUS Valid comparator is enabled. 0: The VBUS Valid comparator is disabled.
1	RW	0x0	utmisrp_chrgvbus VBUS Input Charge Enable
0	RW	0x0	utmisrp_dischrgvbus VBUS Input Discharge Enable

GRF_DDRC_CON0
Address: Operational Base + offset (0x0198)
DDRC control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~ bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:13	RO	0x0	reserved
12:11	RW	0x0	dto_lb DTO Loopback Enable DTO I/O internal loopback enable
10:9	RW	0x0	dto_te DTO On-Die Termination Enable DTO I/O on-die termination enable
8:7	RW	0x0	dto_pdr DTO Power Down Receiver DTO I/O receiver power down
6:5	RW	0x0	dto_pdd DTO Power Down Driver DTO I/O driver power down
4:3	RW	0x0	dto_iom DTO I/O Mode DTO I/O mode select
2:1	RW	0x0	dto_oe DTO Output Enable DTO I/O output enable
0	RW	0x0	ato_ae Analog Test Enable Enables, if set, the analog test output I/O. Connects to the AE pin of the analog test output I/O

GRF_DDRC_STAT
Address: Operational Base + offset (0x019c)
DDRC status

Bit	Attr	Reset Value	Description
31:21	RW	0x000	gpu_idle gpu idle staus
20:19	RO	0x0	reserved
18:16	RO	0x0	ddrupctl_stat Current state of the protocol controller 3'b000 = Init_mem 3'b001 = Config 3'b010 = Config_req 3'b011 = Access 3'b100 = Access_req 3'b101 = Low_power 3'b110 = Low_power_entry_req 3'b111 = Low_power_exit_req
15:0	RO	0x0000	ddrupctl_bbflags Bank busy indication NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on when precharge, activates, reads/writes. Bit0 indication Bank0 busy, bit1 indication Bank1 busy, and so on.

GRF_OS_REG0
Address: Operational Base + offset (0x01c8)
software OS register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	os_reg software OS register

GRF_OS_REG1
Address: Operational Base + offset (0x01cc)
software OS register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	os_reg software OS register

GRF_OS_REG2
Address: Operational Base + offset (0x01d0)
software OS register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	os_reg software OS register

GRF_OS_REG3
Address: Operational Base + offset (0x01d4)
software OS register

Bit	Attr	Reset Value	Description
31:16	WO	0x0000	write_enable bit0~bit15 write enable When bit 16=1, bit 0 can be written by software . When bit 16=0, bit 0 cannot be written by software; When bit 17=1, bit 1 can be written by software . When bit 17=0, bit 1 cannot be written by software; ..... When bit 31=1, bit 15 can be written by software . When bit 31=0, bit 15 cannot be written by software;
15:0	RW	0x0000	os_reg software OS register

Chapter 8 Embedded Processor: Cortex-A9

8.1 Overview

The Cortex™-A9 MP subsystem of the device is based on the symmetricmultiprocessor (SMP) architecture, thus the dual Cortex-A9 MPU subsystem delivers higher performanceand optimal power management, debug and emulation capabilities.

The Cortex-A9 MP subsystem incorporates two Cortex-A9 central processing units (CPUs), level 2(L2) cache shared between the two CPUs, and uses PL310 as L2 cache controller. Each CPU has 32KBof level 1 (L1) instruction cache, 32KB of L1 data cache, separate dedicated power domain, and includes one Neon™ and Vector Floating Point Unit (VFPv3) coprocessors. The Cortex-A9 MP subsystemalso includes standard CoreSight™ components to support SMP debug and emulation, snoop control unit(SCU), interrupt controller (GIC), and clock and reset manager.

The Cortex-A9 MP subsystem supports following feature:

- Coretex-A9 Processor
  - Cortex-A9 core revision r3p0
  - SMP architecture
  - Superscalar, variable length, out-of-order pipeline with dynamic branch prediction, 8-stage pipeline
  - Full implementation of the ARM architecture v7-A instruction setARM Neon Advanced SIMD (single instruction, multiple data) support for accelerated media andsignal processing computation.
  - Include VFPv3 hardware to support single and double-precision add, subtract, divide, multiply andaccumulate, and square root operations.
  - 32KB L1 instruction and 32KB L1 data cache – 32-byte line size, 4-way set associative
  - Memory management unit (MMU)
  - SCU ensures memory coherency between the two CPUs
  - Integrated timer and watchdog timer per CPU
  - Interrupt controller with 128 hardware interrupt inputs
- PL310 L2 cache controller (revision r3p2) with 512KB cache size
  - 16-way set associative
  - 32-byte line size
  - Two slave ports and two master ports
  - Includes four 256-bit line-fill-buffers (LFBs) shared by the master ports
  - Each slave port includes two 256-bit line-read-buffers (LRBs)
  - Includes four 256-bit store buffers with merge capability
  - Lockdown by line supported
  - Lockdown by master ID supported
  - Speculative Read supported
  - Address filter
- CoreSight
  - Program trace macrocell (PTM)
  - Emulation logic (cross-triggers)
  - TPIU and AMBA advanced trace bus (ATB) trace port

8.2 Block Diagram

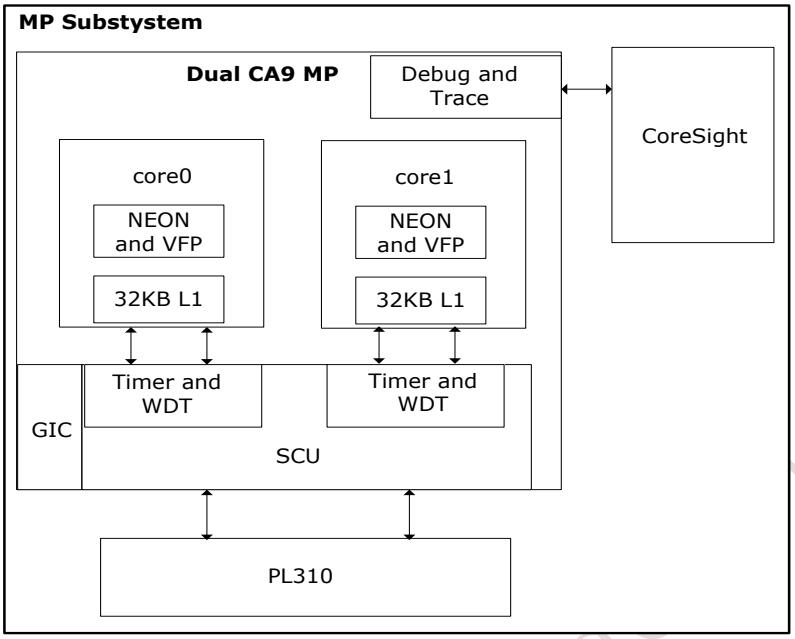


Fig. 8-1MP Subsystem architecture

8.3 Function description

The SCU connects dual Cortex-A9 processors to the memory system through theAXI interfaces.The SCU functions are tomaintain data cache coherency between the Cortex-A9 processors; initiate L2 AXI memory accesses; arbitrate between Cortex-A9 processors requesting L2 accesses.

The Interrupt Controller is compliant with the ARM Generic Interrupt Controller Architecture Specification 1.0. Please refer to Chapter 12 GIC.

The global timer is accessible to all Cortex-A9 processors in the cluster. Each Cortex-A9 processor has a private 64-bit comparator that is used to assert a private interrupt when the global timer has reached the comparator value. All the Cortex-A9 processors in a design use the banked ID, ID27, for this interrupt. ID27 is sent to the GIC as a Private Peripheral Interrupt.The global timer is a 64-bit incrementing counter with an auto-incrementing feature. It continues incrementing after sending interrupts.

The private timer and watchdog can only be accessed by the corresponding processor.It has a 32-bit counter that generates an interrupt when it reaches zero.

8.4 Register description

8.4.1 Registers Summary

Cortex-A9 MP SCU Registers Summary

Name	Offset	Size	Reset	Description
MP_SCU_CTRL	0x0000	W	0x00000000	Global timer counter low 32bits register
MP_SCU_CFG	0x0004	W	0x00000000	Global timer counter high 32bits register

Name	Offset	Size	Reset	Description
MP_SCU_PWR_STATUS	0x0008	W	0x00000000	Global timer control register
MP_SCU_INVALIDATE	0x000c	W	0x00000000	Global timer interrupt status register
MP_SCU_FILTER_START	0x0010	W	0x00000000	Global timer comparator low 32bits register
MP_SCU_FILTER_END	0x0014	W	0x00000000	Global timer comparator high 32bits register
MP_SCU_SAC	0x0018	W	0x00000000	Global timer auto increment register
MP_SCU_SNSAC	0x001c	W	0x00000000	Global timer auto increment register

Cortex-A9 MP Global Timer Registers Summary

Name	Offset	Size	Reset Value	Description
MP_GTIMER_COUNTER_LOW	0x0000	W	0x00000000	Global timer counter low 32bits register
MP_GTIMER_COUNTER_HIGH	0x0004	W	0x00000000	Global timer counter high 32bits register
MP_GTIMER_CONTROL	0x0008	W	0x00000000	Global timer control register
MP_GTIMER_INT_STATUS	0x000c	W	0x00000000	Global timer interrupt status register
MP_GTIMER_COMPARE_LOW	0x0010	W	0x00000000	Global timer comparator low 32bits register
MP_GTIMER_COMPARE_HIGH	0x0014	W	0x00000000	Global timer comparator high 32bits register
MP_GTIMER_AUTO_INCR	0x0018	W	0x00000000	Global timer auto increment register

Cortex-A9 MP Private Timer Registers Summary

Name	Offset	Size	Reset Value	Description
MP_PTIMER_TIMER_LOAD	0x0000	W	0x00000000	Private timer load register
MP_PTIMER_TIMER_COUNTER	0x0004	W	0x00000000	Private timer counter register

Name	Offset	Size	Reset Value	Description
MP_PTIMER_TIMER_CONTROL	0x0008	W	0x00000000	Private timer control register
MP_PTIMER_TIMER_INT_STATUS	0x000c	W	0x00000000	Private timer interrupt status register
MP_PTIMER_WDT_LOAD	0x0020	W	0x00000000	Private watchdogload register
MP_PTIMER_WDT_COUNTER	0x0024	W	0x00000000	Private watchdog counter register
MP_PTIMER_WDT_CONTROL	0x0028	W	0x00000000	Private watchdog control register
MP_PTIMER_WDT_INT_STATUS	0x002c	W	0x00000000	Private watchdog interrupt status register
MP_PTIMER_WDT_RESET_STATUS	0x0030	W	0x00000000	Private watchdog reset status register
MP_PTIMER_WDT_DISABLE	0x0034	W	0x00000000	private watchdog disable

L2C Registers Summary

Name	Offset	Size	Reset Value	Description
L2C_reg0_cache_id	0x0000	W	0x410000c6	Cache ID Register
L2C_reg0_cache_type	0x0004	W	0x00000000	Cache Type Register
L2C_reg1_control	0x0100	W	0x00000000	
L2C_reg1_aux_control	0x0104	W	0x00000000	Auxiliary Control Register
L2C_reg1_tag_ram_control	0x0108	W	0x00000000	Tag RAM Latency Control Registers
L2C_reg1_data_ram_control	0x010c	W	0x00000000	Data RAM Latency Control Registers
L2C_reg2_ev_counter_ctrl	0x0200	W	0x00000000	Event Counter Control Register

Name	Offset	Size	Reset Value	Description
L2C_reg2_ev_counter1_cfg	0x0204	W	0x00000000	Event Counter Configuration Registers
L2C_reg2_ev_counter0_cfg	0x0208	W	0x00000000	Event Counter Configuration Registers
L2C_reg2_ev_counter1	0x020c	W	0x00000000	Event counter value registers
L2C_reg2_ev_counter0	0x0210	W	0x00000000	Event counter value registers
L2C_reg2_int_mask	0x0214	W	0x00000000	
L2C_reg2_int_mask_status	0x0218	W	0x00000000	
L2C_reg2_int_raw_status	0x021c	W	0x00000000	
L2C_reg2_int_clear	0x0220	W	0x00000000	
L2C_reg7_cache_sync	0x0730	W	0x00000000	
L2C_reg7_inv_pa	0x0770	W	0x00000000	
L2C_reg7_inv_way	0x077c	W	0x00000000	
L2C_reg7_clean_pa	0x07b0	W	0x00000000	
L2C_reg7_clean_index	0x07b8	W	0x00000000	
L2C_reg7_clean_way	0x07bc	W	0x00000000	
L2C_reg7_clean_inv_pa	0x07f0	W	0x00000000	
L2C_reg7_clean_inv_index	0x07f8	W	0x00000000	
L2C_reg7_clean_inv_way	0x07fc	W	0x00000000	
L2C_reg9_d_lockdown0	0x0900	W	0x00000000	
L2C_reg9_i_lockdown0	0x0904	W	0x00000000	
L2C_reg9_d_lockdown1	0x0908	W	0x00000000	
L2C_reg_i_lockdown1	0x090c	W	0x00000000	
L2C_reg9_d_lockdown2	0x0910	W	0x00000000	
L2C_reg9_i_lockdown2	0x0914	W	0x00000000	
L2C_reg9_d_lockdown3	0x0918	W	0x00000000	
L2C_reg9_i_lockdwon3	0x091c	W	0x00000000	
L2C_reg9_d_lockdown4	0x0920	W	0x00000000	
L2C_reg9_i_lockdwon4	0x0924	W	0x00000000	
L2C_reg9_d_lockdwon5	0x0928	W	0x00000000	
L2C_reg9_i_lockdwon5	0x092c	W	0x00000000	
L2C_reg9_d_lockdwon6	0x0930	W	0x00000000	
L2C_reg9_i_lockdwon6	0x0934	W	0x00000000	
L2C_reg9_d_lockdwon7	0x0938	W	0x00000000	
L2C_reg9_i_lockdwon7	0x093c	W	0x00000000	
L2C_reg9_lock_line_en	0x0950	W	0x00000000	

Name	Offset	Size	Reset Value	Description
L2C_reg9_unlock_way	0x0954	W	0x00000000	
L2C_reg12_addr_filtering_start	0x0c00	W	0x00000000	
L2C_reg12_addr_filtering_end	0x0c04	W	0x00000000	
L2C_reg15_debug_ctrl	0x0f40	W	0x00000000	Debug Register o
L2C_reg15_prefetch_ctrl	0x0f60	W	0x00000000	Prefetch Control Register
L2C_reg15_power_ctrl	0x0f80	W	0x00000000	Power Control Register

8.4.2 Detail Registers Description

MP_SCU_CTRL
Address: Operational Base + offset (0x0000)
SCU Control Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	ic_standby_en when set, this stops the Interrupt Controller clock when no interrupts are pending, and no CPU is performing a read/write request.
5	RW	0x0	scu_stanby_en When set, SCU CLK is turned off when all processors are in WFI mode, and there is no remaining activity in the SCU. The clock is turned on when any processor leaves WFI mode.
4	RW	0x0	force all device to port0 enable When set, all requests from processors with AxCACHE = NonCacheable Bufferable are forced to be issued on the AXI Master port M0.
3	RW	0x0	scu speculative linefills enable When set, coherent linefill requests are sent speculatively to the L2C-310 in parallel with the tag look-up. If the tag look-up misses, the confirmed linefill is sent to the L2C-310 and gets RDATA earlier because the data request was already initiated by the speculative request. This feature works only if the L2C-310 is present in the design.

Bit	Attr	Reset Value	Description
2	RW	0x0	scu rams parity enable 1 = Parity on. 0 = Parity off. This is the default setting. This bit is always zero if support for parity is not implemented.
1	RW	0x0	address filtering enable 1 = Addressing filtering on. 0 = Addressing filtering off. The default value is the value of FILTEREN sampled when nSCURESET is deasserted.
0	RW	0x0	1 = SCU enable. 0 = SCU disable. This is the default setting.

MP_SCU_CFG

Address: Operational Base + offset (0x0004)

SCU Configuration Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	tagram_size Bits [11:10] indicate Cortex-A9 processor CPU1 tag RAM size if present. Bits [9:8] indicate Cortex-A9 processor CPU0 tag RAM size. The encoding is as follows: b11 = reserved b10 = 64KB cache, 256 indexes per tag RAM b01 = 32KB cache, 128 indexes per tag RAM b00 = 16KB cache, 64 indexes per tag RAM.
7:4	RW	0x0	cpus_smp 0 = this Cortex-A9 processor is in AMP mode not taking part in coherency or not present. 1 = this Cortex-A9 processor is in SMP mode taking part in coherency. Bit 5 is for CPU1 Bit 4 is for CPU0.
3:2	RO	0x0	reserved
1:0	RW	0x0	cpu_numbers Number of CPUs present in the Cortex-A9 MPCore processor b01 = two Cortex-A9 processors, CPU0 and CPU1 b00 = one Cortex-A9 processor, CPU0.

MP_SCU_PWR_STATUS
Address: Operational Base + offset (0x0008)
SCU CPU Power Status Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:8	RW	0x0	cpu1_status Power status of the Cortex-A9 processor.
7:2	RO	0x0	reserved
1:0	RW	0x0	cpu0_status Power status of the Cortex-A9 processor.

MP_SCU_INVALIDATE
Address: Operational Base + offset (0x000c)
SCU Invalidate All Registers

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
7:4	RW	0x0	cpu1_ways Specifies the ways that must be invalidated for CPU1. Writing to these bits has no effect if the Cortex-A9 MPCore processor has fewer than two processors.
3:0	RW	0x0	cpu0_ways Specifies the ways that must be invalidated for CPU0.

MP_SCU_FILTER_START
Address: Operational Base + offset (0x0010)
Filtering Start Address Register

Bit	Attr	Reset Value	Description
31:20	RW	0x000	filter start address Start address for use with master port 1 in a two-master port configuration when address filtering is enabled. The default value is the value of FILTERSTART sampled on exit from reset. The value on the pin gives the upper address bits with 1MB granularity.
19:0	RO	0x000	reserved

MP_SCU_FILTER_END
Address: Operational Base + offset (0x0014)
Filtering End Address Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:20	RW	0x000	filter end address End address for use with master port 1 in a two-master port configuration, whenaddress filtering is enabled. The default value is the value of FILTEREND sampled on exit from reset. The value on the pin gives the upper address bits with 1MB granularity.
19:0	RO	0x000	reserved

MP_SCU_SAC
Address: Operational Base + offset (0x0018)
SCU Access Control Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	RW	0x0	cpu3 access Field0002 Description
2	RW	0x0	cpu2 access Field0001 Description
1	RW	0x0	cpu1 access 0 = CPU1 cannot access the components. 1 = CPU1 can access the components. This is the default.
0	RW	0x0	cpu0 access 0 = CPU0 cannot access the components. 1 = CPU0 can access the components. This is the default.

MP_SCU_SNSAC
Address: Operational Base + offset (0x001c)
SCU Non-secure Access Control Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x0	global access controll for CPU<n> Non-secure access to the global timer for CPU<n>. <n> is 1 for bit[1] <n> is 0 for bit[0]. 0 = Secure accesses only. This is the default value. 1 = Secure accesses and Non-Secure accesses.

Bit	Attr	Reset Value	Description
7:4	RW	0x0	private access controll for CPU<n> Non-secure access to the private timer and watchdog for CPU<n>. <n> is 1 for bit[1] <n> is 0 for bit[0]. 0 = Secure accesses only. Non-secure reads return 0. This is the default value. 1 = Secure accesses and Non-secure accesses.
3:0	RW	0x0	component access control for CPU<n> Non-secure access to the components for CPU<n>. <n> is 1 for bit[1] <n> is 0 for bit[0]. 0 = CPU cannot write the componentsa 1 = CPU can access the componentsa.

MP_GTIMER_COUNTER_LOW
Address: Operational Base + offset (0x0000)
Global timer counter low 32bits register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	load lower 32-bit timer counter register

MP_GTIMER_COUNTER_HIGH
Address: Operational Base + offset (0x0004)
Global timer counter high 32bits register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	counter upper 32-bit timer counter register

MP_GTIMER_CONTROL
Address: Operational Base + offset (0x0008)
Global timer control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	prescaler The prescaler modifies the clock period for the decrementing event for the Counter Register.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	auto increment This bit is banked per Cortex-A9 processor. 1'b0: single shot mode. When the counter reaches the comparator value, sets the event flag. It is the responsibility of software to update the comparator value to get further events. 1'b1: auto increment mode. Each time the counter reaches the comparator value, the comparator register is incremented with the auto-increment register, so that further events can be set periodically without any software updates.
2	RW	0x0	IRQ enable This bit is banked per Cortex-A9 processor. If set, the interrupt ID 27 is set as pending in the Interrupt Distributor when the event flag is set in the Timer Status Register
1	RW	0x0	compare enable This bit is banked per Cortex-A9 processor. If set, it allows the comparison between the 64-bit Timer Counter and the related 64-bit Comparator Register.
0	RW	0x0	Timer enable 1'b0 = Timer is disabled and the counter does not increment. All registers can still be read and written 1'b1 = Timer is enabled and the counter increments normally

MP_GTIMER_INT_STATUS
Address: Operational Base + offset (0x000c)
Global timer interrupt status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	event flag This is a banked register for all Cortex-A9 processors present. The event flag is a sticky bit that is automatically set when the Counter Register reaches the Comparator Register value. If the timer interrupt is enabled, Interrupt ID 27 is set as pending in the Interrupt Distributor after the event flag is set. The event flag is cleared when written to 1.

MP_GTIMER_COMPARE_LOW
Address: Operational Base + offset (0x0010)
Global timer comparator low 32bits register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	lower 32-bit Comparator Value Register

MP_GTIMER_COMPARE_HIGH
Address: Operational Base + offset (0x0014)
Global timer comparator high 32bits register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	upper 32-bit Comparator Value Register

MP_GTIMER_AUTO_INCR
Address: Operational Base + offset (0x0018)
Global timer auto increment register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Autoincrement This 32-bit register gives the increment value of the Comparator Register when the Auto-increment bit is set in the Timer Control Register. Each Cortex-A9 processor present has its own Auto-increment Register. If the comp enable and auto-increment bits are set when the global counter reaches the Comparator Register value, the comparator is incremented by the auto-increment value, so that a new event can be set periodically. The global timer is not affected and goes on incrementing.

MP_PTIMER_TIMER_LOAD
Address: Operational Base + offset (0x0000)

Private timer load register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	The Timer Load Register contains the value copied to the Timer Counter Register when it decrements down to zero with auto reload mode enabled. Writing to the Timer Load Register means that you also write to the Timer Counter Register.

MP_PTIMER_TIMER_COUNTER

Address: Operational Base + offset (0x0004)

Private timer counter register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	counter

MP_PTIMER_TIMER_CONTROL

Address: Operational Base + offset (0x0008)

Private timer control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	prescaler The prescaler modifies the clock period for the decrementing event for the Counter Register.
7:3	RO	0x0	reserved
2	RW	0x0	IRQ enable If set, the interrupt ID 29 is set as pending in the Interrupt Distributor when the event flag is set in the Timer Status Register
1	RW	0x0	auto reload 1'b0 = Single shot mode. Counter decrements down to zero, sets the event flag and stops. 1'b1 = Auto-reload mode. Each time the Counter Register reaches zero, it is reloaded with the value contained in the Timer Load Register.
0	RW	0x0	Timer enable 1'b0 = Timer is disabled and the counter does not decrement. All registers can still be read and written 1'b1 = Timer is enabled and the counter decrements normally

MP_PTIMER_TIMER_INT_STATUS
Address: Operational Base + offset (0x000c)
Private timer interrupt status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	event flag The event flag is a sticky bit that is automatically set when the Counter Register reaches zero. If the timer interrupt is enabled, Interrupt ID 29 is set as pending in the Interrupt Distributor after the event flag is set. The event flag is cleared when written to 1.

MP_PTIMER_WDT_LOAD
Address: Operational Base + offset (0x0020)
Private watchdogload register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	The Watchdog Load Register contains the value copied to the Watchdog Counter Register when it decrements down to zero with auto reload mode enabled, in Timer mode. Writing to the Watchdog Load Register means that you also write to the Watchdog Counter Register

MP_PTIMER_WDT_COUNTER
Address: Operational Base + offset (0x0024)
Private watchdog counter register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	It decrements if the Watchdog is enabled using the Watchdog enable bit in the Watchdog Control Register.

MP_PTIMER_WDT_CONTROL
Address: Operational Base + offset (0x0028)
Private watchdog control register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RW	0x00	The prescaler modifies the clock period for the decrementing event for the Counter Register.
7:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RW	0x0	watchdog mode 1'b0 = Timer mode, default Writing a zero to this bit has no effect. You must use the Watchdog Disable Register to put the watchdog into timer mode. 1'b1 = Watchdog mode
2	RW	0x0	IT enable If set, the interrupt ID 30 is set as pending in the Interrupt Distributor when the event flag is set in the watchdog Status Register. In watchdog mode this bit is ignored
1	RW	0x0	auto reload 1'b0 = Single shot mode. Counter decrements down to zero, sets the event flag and stops. 1'b1 = Auto-reload mode. Each time the Counter Register reaches zero, it is reloaded with the value contained in the Load Register and then continues decrementing.
0	RW	0x0	watchdog enable Global watchdog enable 1'b0 = Watchdog is disabled and the counter does not decrement. All registers can still be read and /or written 1'b1 = Watchdog is enabled and the counter decrements normally

MP_PTIMER_WDT_INT_STATUS
Address: Operational Base + offset (0x002c)
Private watchdog interrupt status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	event flag The event flag is a sticky bit that is automatically set when the Counter Register reaches zero in timer mode. If the watchdog interrupt is enabled, Interrupt ID 30 is set as pending in the Interrupt Distributor after the event flag is set. The event flag is cleared when written with a value of 1. Trying to write a zero to the event flag or a one when it is not set has no effect.

MP_PTIMER_WDT_RESET_STATUS
Address: Operational Base + offset (0x0030)
Private watchdog reset status register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	private watchdog reset flag The reset flag is a sticky bit that is automatically set when the Counter Register reaches zero and a reset request is sent accordingly. (In watchdog mode)

MP_PTIMER_WDT_DISABLE
Address: Operational Base + offset (0x0034)
private watchdog disable

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	WO	0x0	private watchdog disable Use the Watchdog Disable Register to switch from watchdog to timer mode. The software must write 0x12345678 then 0x87654321 successively to the Watchdog Disable Register so that the watchdog mode bit in the Watchdog Control Register is set to zero.

L2C_reg0_cache_id
Address: Operational Base + offset (0x0000)
Cache ID Register

Bit	Attr	Reset Value	Description
31:24	RW	0x41	Implementer
23:16	RO	0x0	reserved
15:10	RW	0x00	CacheID
9:6	RW	0x3	PartNumber
5:0	RW	0x06	RTL release

L2C_reg0_cache_type
Address: Operational Base + offset (0x0004)
Cache Type Register

Bit	Attr	Reset Value	Description
31	RW	0x0	data_banking 0 = Data banking not implemented. 1 = Data banking implemented
30:29	RO	0x0	reserved

Bit	Attr	Reset Value	Description
28:25	RW	0x0	ctype 11xy, where: x=1 if pl310_LOCKDOWN_BY_MASTER is defined, otherwise 0 y=1 if pl310_LOCKDOWN_BY_LINE is defined, otherwise 0.
24:12	RO	0x0	reserved
11:7	RW	0x00	Isize Read from Auxiliary Control Register[19:17]
6	RW	0x0	associativity Read from Auxiliary Control Register[16]
5:2	RO	0x0	reserved
1:0	RW	0x0	line_length 00-32 bytes

L2C_reg1_control
Address: Operational Base + offset (0x0100)
reg1_control

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	l2_en 0 = L2 Cache is disabled. This is the default value. 1 = L2 Cache is enabled

L2C_reg1_aux_control
Address: Operational Base + offset (0x0104)
Auxiliary Control Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	early_bresp_en 0 = Early BRESP disabled. This is the default. 1 = Early BRESP enabled.
29	RW	0x0	inst_prf_en 0 = Instruction prefetching disabled. This is the default. 1 = Instruction prefetching enabled.
28	RW	0x0	data_prf_en 0 = Data prefetching disabled. This is the default. 1 = Data prefetching enabled

Bit	Attr	Reset Value	Description
27	RW	0x0	ns_int_ac 0 = Interrupt Clear, 0x220, and Interrupt Mask, 0x214, can only be modified or read with secure accesses. This is the default. 1 = Interrupt Clear, 0x220, and Interrupt Mask, 0x214, can be modified or read with secure or non-secure accesses.
26	RW	0x0	ns_lock_en 0 = Lockdown registers cannot be modified using non-secure accesses. This is the default. 1 = Non-secure accesses can write to the lockdown registers.
25	RW	0x0	crp 0 = pseudo-random replacement using lfsr. 1 = round-robin replacement. This is the default.
24:23	RW	0x0	fwa b00 = Use AWCACHE attributes for WA. This is the default. b01 = Force no allocate, set WA bit always 0. b10 = Override AWCACHE attributes, set WA bit always 1, all cacheable write misses become write allocated. b11 = Internally mapped to 00.
22	RW	0x0	sav_en 0 = Treats shared accesses as specified in Shareable attribute. This is the default. 1 = Shared attribute internally ignored.
21	RW	0x0	parity_en 0 = Disabled. This is the default. 1 = Enabled
20	RW	0x0	evmb_en 0 = Disabled. This is the default. 1 = Enable

Bit	Attr	Reset Value	Description
19:17	RW	0x0	way_size b000 = Reserved, internally mapped to 16KB. b001 = 16KB. b010 = 32KB. b011 = 64KB. b100 = 128KB. b101 = 256KB. b110 = 512KB. b111 = Reserved, internally mapped to 512 KB.
16	RW	0x0	associativity 0 = 8-way. 1 = 16-way.
15:14	RO	0x0	reserved
13	RW	0x0	sai_en 0 = Shared invalidate behavior disabled. This is the default. 1 = Shared invalidate behavior enabled, if Shared Attribute Override Enable bit not set.
12	RW	0x0	excc 0 = Disabled. This is the default. 1 = Enabled,
11	RW	0x0	sbdl_en 0 = Store buffer device limitation disabled. Device writes can take all slots in store buffer. This is the default. 1= Store buffer device limitation enabled. Device writes cannot take all slots in store buffer when connected to the Cortex-A9 MPCore processor. There is always one available slot to service Normal Memory
10	RO	0x0	stronly_priority 0 = Strongly Ordered and Device reads have lower priority than cacheable accesses when arbitrated in the L2CC (L2C-310) master ports. This is the default. 1 = Strongly Ordered and Device reads get the highest priority when arbitrated in the L2CC (L2C-310) master ports.
9:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	full_line_zero 0 = Full line of write zero behavior disabled. This is the default. 1 = Full line of write zero behavior Enabled

L2C_reg1_tag_ram_control
Address: Operational Base + offset (0x0108)
Tag RAM Latency Control Registers

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:8	RW	0x0	write_ac_latency b000 = 1 cycle of latency, there is no additional latency. b001 = 2 cycles of latency. b010 = 3 cycles of latency. b011 = 4 cycles of latency. b100 = 5 cycles of latency. b101 = 6 cycles of latency. b110 = 7 cycles of latency. b111 = 8 cycles of latency.
7	RO	0x0	reserved
6:4	WO	0x0	read_ac_latency b000 = 1 cycle of latency, there is no additional latency. b001 = 2 cycles of latency. b010 = 3 cycles of latency. b011 = 4 cycles of latency. b100 = 5 cycles of latency. b101 = 6 cycles of latency. b110 = 7 cycles of latency. b111 = 8 cycles of latency
3	RO	0x0	reserved
2:0	WO	0x0	setup_latency b000 = 1 cycle of latency, there is no additional latency. b001 = 2 cycles of latency. b010 = 3 cycles of latency. b011 = 4 cycles of latency. b100 = 5 cycles of latency. b101 = 6 cycles of latency. b110 = 7 cycles of latency. b111 = 8 cycles of latency.

L2C_reg1_data_ram_control
Address: Operational Base + offset (0x010c)
Data RAM Latency Control Registers

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:8	RW	0x0	write_ac_latency b000 = 1 cycle of latency, there is no additional latency. b001 = 2 cycles of latency. b010 = 3 cycles of latency. b011 = 4 cycles of latency. b100 = 5 cycles of latency. b101 = 6 cycles of latency. b110 = 7 cycles of latency. b111 = 8 cycles of latency.
7	RO	0x0	reserved
6:4	RW	0x0	read_ac_latency b000 = 1 cycle of latency, there is no additional latency. b001 = 2 cycles of latency. b010 = 3 cycles of latency. b011 = 4 cycles of latency. b100 = 5 cycles of latency. b101 = 6 cycles of latency. b110 = 7 cycles of latency. b111 = 8 cycles of latency.
3	RO	0x0	reserved
2:0	RO	0x0	setup_latency b000 = 1 cycle of latency, there is no additional latency. b001 = 2 cycles of latency. b010 = 3 cycles of latency. b011 = 4 cycles of latency. b100 = 5 cycles of latency. b101 = 6 cycles of latency. b110 = 7 cycles of latency. b111 = 8 cycles of latency.

L2C_reg2_ev_counter_ctrl
Address: Operational Base + offset (0x0200)
Event Counter Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:1	RO	0x0	counter_rst Always Read as zero. The following counters are reset when a 1 is written to the following bits: bit[2] = Event Counter1 reset bit[1] = Event Counter0 reset.
0	RO	0x0	ev_cnt_en 0 = Event Counting Disable. This is the default. 1 = Event Counting Enable

L2C_reg2_ev_counter1_cfg
Address: Operational Base + offset (0x0204)
Event Counter Configuration Registers

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:2	RW	0x0	ev_source
1:0	RW	0x0	ev_cnt_int_gen b00 = Disabled. This is the default. b01 = Enabled: Increment condition. b10 = Enabled: Overflow condition. b11 = Interrupt generation is disabled

L2C_reg2_ev_counter0_cfg
Address: Operational Base + offset (0x0208)
Event Counter Configuration Registers

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:2	RW	0x0	en_source
1:0	RO	0x0	ev_cnt_int_gen

L2C_reg2_ev_counter1
Address: Operational Base + offset (0x020c)
Event counter value registers

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cnt_val Total of the event selected. If a counter reaches its maximum value, it saturates at that value until it is reset.

L2C_reg2_ev_counter0
Address: Operational Base + offset (0x0210)

Event counter value registers

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cnt_val Total of the event selected. If a counter reaches its maximum value, it saturates at that value until it is reset.

L2C_reg2_int_mask
Address: Operational Base + offset (0x0214)
reg2_int_mask

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	DECERR_MSK 1 = Enabled. 0 = Masked. This is the default.
7	RW	0x0	SLVERR_MSK 1 = Enabled. 0 = Masked. This is the default.
6	RW	0x0	ERRRD_MSK 1 = Enabled. 0 = Masked. This is the default.
5	RW	0x0	ERRRT_MSK 1 = Enabled. 0 = Masked. This is the default.
4	RW	0x0	ERRWD_MSK 1 = Enabled. 0 = Masked. This is the default.
3	RW	0x0	ERRWT_MSK 1 = Enabled. 0 = Masked. This is the default.
2	RW	0x0	PARRD_MSK 1 = Enabled. 0 = Masked. This is the default.
1	RW	0x0	PARRT_MSK 1 = Enabled. 0 = Masked. This is the default.
0	RO	0x0	ECNTR_MSK 1 = Enabled. 0 = Masked. This is the default.

L2C_reg2_int_mask_status
Address: Operational Base + offset (0x0218)
reg2_int_mask_status

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	DECERR_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the inputlines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
7	RW	0x0	SLVERR_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
6	RW	0x0	ERRRD_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
5	RW	0x0	ERRRT_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
4	RW	0x0	ERRWD_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
3	RW	0x0	ERRWT_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked

Bit	Attr	Reset Value	Description
2	RW	0x0	PARRD_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
1	RW	0x0	PARRT_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked
0	RO	0x0	ECNTR_MSKST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, either no interrupt has been generated, or the interrupt is masked

L2C_reg2_int_raw_status
Address: Operational Base + offset (0x021c)
reg2_int_raw_status

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8	RW	0x0	DECERR_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
7	RW	0x0	SLVERR_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
6	RW	0x0	ERRRD_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.

Bit	Attr	Reset Value	Description
5	RW	0x0	ERRRT_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
4	RW	0x0	ERRWD_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
3	RW	0x0	ERRWT_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
2	RW	0x0	PARRD_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
1	RW	0x0	PARRT_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.
0	RO	0x0	ECNTR_RAWST Bits read can be HIGH or LOW: HIGH If the bits read HIGH, they reflect the status of the input lines triggering an interrupt. LOW If the bits read LOW, no interrupt has been generated.

L2C_reg2_int_clear
Address: Operational Base + offset (0x0220)
reg2_int_clear

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	DECERR_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
7	RW	0x0	SLVERR_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
6	RW	0x0	ERRRD_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
5	RW	0x0	ERRRT_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
4	RW	0x0	ERRWD_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
3	RW	0x0	ERRWT_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
2	RW	0x0	PARRD_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect
1	RW	0x0	PARRT_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect

Bit	Attr	Reset Value	Description
0	RO	0x0	ECNTR_CL When a bit is written as 1, it clears the corresponding bit in the Raw Interrupt Status Register. When a bit is written as 0, it has no effect

L2C_reg7_cache_sync
Address: Operational Base + offset (0x0730)
reg7_cache_sync

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	sync Drain the STB. Operation complete when all buffers, LRB, LFB, STB, and EB, are empty

L2C_reg7_inv_pa
Address: Operational Base + offset (0x0770)
reg7_inv_pa

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	set_en Specific L2 cache line is marked as not valid.

L2C_reg7_inv_way
Address: Operational Base + offset (0x077c)
reg7_inv_way

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	Invalidate all data in specified ways, including dirty data. An Invalidate by way while selecting all cache ways is equivalent to invalidating all cache entries. Completes as a background task with the way, or ways, locked, preventing allocation.

L2C_reg7_clean_pa
Address: Operational Base + offset (0x07b0)
reg7_clean_pa

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	set_en set enable bits Write the specific L2 cache line to L3 main memory if the line is marked as valid and dirty. The line is marked as not dirty. The valid bit is unchanged.

L2C_reg7_clean_index
Address: Operational Base + offset (0x07b8)
reg7_clean_index

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	set_en set enable bits Write the specific L2 cache line within the specified way to L3 main memory if the line is marked as valid and dirty. The line is marked as not dirty. The valid bit is unchanged

L2C_reg7_clean_way
Address: Operational Base + offset (0x07bc)
reg7_clean_way

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clear_en clear enable bits Writes each line of the specified L2 cache ways to L3 main memory if the line is marked as valid and dirty. The lines are marked as not dirty. The valid bits are unchanged. Completes as a background task with the way, or ways, locked, preventing allocation

L2C_reg7_clean_inv_pa
Address: Operational Base + offset (0x07f0)
reg7_clean_inv_pa

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clear_en clear enable bits Write the specific L2 cache line to L3 main memory if the line is marked as valid and dirty. The line is marked as not valid.

L2C_reg7_clean_inv_index
Address: Operational Base + offset (0x07f8)
reg7_clean_inv_index

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clear_en Write the specific L2 cache line within the specified way to L3 main memory if the line is marked as valid and dirty. The line is marked as not valid

L2C_reg7_clean_inv_way
Address: Operational Base + offset (0x07fc)
reg7_clean_inv_way

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	clear_en Writes each line of the specified L2 cache ways to L3 main memory if the line is marked as valid and dirty. The lines are marked as not valid. Completes as a background task with the way, or ways, locked, preventing allocation.

L2C_reg9_d_lockdown0
Address: Operational Base + offset (0x0900)
reg9_d_lockdown0

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock000 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdown0
Address: Operational Base + offset (0x0904)
reg9_i_lockdown0

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	instlock000 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdown1
Address: Operational Base + offset (0x0908)
reg9_d_lockdown1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock001 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg_i_lockdown1
Address: Operational Base + offset (0x090c)
reg_i_lockdown1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	instlock001 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdown2
Address: Operational Base + offset (0x0910)
reg9_d_lockdown2

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock002 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdown2
Address: Operational Base + offset (0x0914)
reg9_i_lockdown2

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	instlock002 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdown3
Address: Operational Base + offset (0x0918)
reg9_d_lockdown3

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock003 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdwon3
Address: Operational Base + offset (0x091c)
reg9_i_lockdwon3

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	instlock003 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdown4
Address: Operational Base + offset (0x0920)
reg9_d_lockdown4

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	datalockoo4 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdwon4
Address: Operational Base + offset (0x0924)
reg9_i_lockdwon4

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	instlock004 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdwon5
Address: Operational Base + offset (0x0928)
reg9_d_lockdwon5

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock005 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdwon5
Address: Operational Base + offset (0x092c)
reg9_i_lockdwon5

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	instlock005 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdwon6
Address: Operational Base + offset (0x0930)
reg9_d_lockdwon6

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock006 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdwon6
Address: Operational Base + offset (0x0934)
reg9_i_lockdwon6

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	instlock006 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_d_lockdwon7
Address: Operational Base + offset (0x0938)
reg9_d_lockdwon7

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	datalock007 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_i_lockdwon7
Address: Operational Base + offset (0x093c)
reg9_i_lockdwon7

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	instlock007 each bit has the following meaning: 0 allocation can occur in the corresponding way. 1 there is no allocation in the corresponding way

L2C_reg9_lock_line_en
Address: Operational Base + offset (0x0950)
reg9_lock_line_en

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	lock_down_by_line_en 0 = Lockdown by line disabled. This is the default. 1 = Lockdown by line enabled.

L2C_reg9_unlock_way
Address: Operational Base + offset (0x0954)
reg9_unlock_way

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	unlock_all_lines_by_way_op For all bits: 0 = Unlock all lines disabled. This is the default. 1 = Unlock all lines operation in progress for the corresponding way.

L2C_reg12_addr_filtering_start
Address: Operational Base + offset (0x0c00)
reg12_addr_filtering_start

Bit	Attr	Reset Value	Description
31:20	RW	0x000	address_filtering_start Address filtering start address for bits [31:20] of the filtering address
19:1	RO	0x0	reserved
0	RW	0x0	filter_en 0 = Address filtering disabled. 1 = Address filtering enabled

L2C_reg12_addr_filtering_end
Address: Operational Base + offset (0x0c04)

reg12_addr_filtering_end

Bit	Attr	Reset Value	Description
31:20	RW	0x000	address_filtering_end Address filtering end address for bits [31:20] of the filtering address.
19:0	RO	0x000	reserved

L2C_reg15_debug_ctrl
Address: Operational Base + offset (0x0f40)
Debug Register o

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	SPNIDEN Reads value of SPNIDEN input.
1	RW	0x0	DWB 0 = Enable write-back behavior. This is the default. 1 = Force write-through behavior.
0	RW	0x0	DCL 0 = Enable cache linefills. This is the default. 1 = Disable cache linefills.

L2C_reg15_prefetch_ctrl
Address: Operational Base + offset (0x0f60)
Prefetch Control Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	double_linefill_en You can set the following options for this register bit: 0 The L2CC always issues 4x64-bit read bursts to L3 on reads that miss in the L2 cache. This is the default. 1 The L2CC issues 8x64-bit read bursts to L3 on reads that miss in the L2 cache.
29	RW	0x0	inst_prf_en You can set the following options for this register bit: 0 Instruction prefetching disabled. This is the default. 1 Instruction prefetching enabled.

Bit	Attr	Reset Value	Description
28	RW	0x0	data_prf_en You can set the following options for this register bit: 0 Data prefetching disabled. This is the default. 1 Data prefetching enabled.
27	RW	0x0	double_linefill_on_wrap You can set the following options for this register bit: 0 Double linefill on WRAP read enabled. This is the default. 1 Double linefill on WRAP read disabled.
26:25	RO	0x0	reserved
24	RW	0x0	prf_drop_en You can set the following options for this register bit: 0 The L2CC does not discard prefetch reads issued to L3. This is the default. 1 The L2CC discards prefetch reads issued to L3 when there is a resource conflict with explicit reads
23	RW	0x0	incr_db_lf_en incr_db_lf_en You can set the following options for this register bit: 0 The L2CC does not issue INCR 8x64-bit read bursts to L3 on reads that miss in the L2 cache. This is the default. 1 The L2CC can issue INCR 8x64-bit read bursts to L3 on reads that miss in the L2 cache.
22	RO	0x0	reserved
21	RW	0x0	excl_seq_en You can set the following options for this register bit: 0 Read and write portions of a non-cacheable exclusive sequence have the same AXI ID when issued to L3. This is the default. 1 Read and write portions of a non-cacheable exclusive sequence do not have the same AXI ID when issued to L3.
20:5	RO	0x0	reserved
4:0	RW	0x00	prf_offset Default = b00000

L2C_reg15_power_ctrl
Address: Operational Base + offset (0x0f80)
Power Control Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	dynamic_clk_gating_en 1 = Enabled. 0 = Masked. This is the default
0	RW	0x0	standby_mode_en 1 = Enabled. 0 = Masked. This is the default.

8.5 Application Notes

8.5.1 Address filtering

When address_filtering_enable is set, all accesses with address >= address_filtering_start and < address_filtering_end are automatically directed to M1. All other accesses are directed to M0.

Because the input pins provide the reset values of the address filtering registers, it is not expected that the values of these registers are changed dynamically after reset. Furthermore, changing these values without special attention can lead to unpredictable behavior.

It is recommended that you program the Address Filtering End Register before the Address Filtering Start Register to avoid unpredictable behavior between the two writes.

8.5.2 L2 Cache initialization

A typical cache controller start-up programming sequence consists of the following register operations:

1. Write to the Auxiliary, Tag RAM Latency, Data RAM Latency, Prefetch, and PowerControl registers using a read-modify-write to set up global configurations:
   - associativity, Way Size
   - latencies for RAM accesses
   - allocation policy
   - prefetch and power capabilities.

2. Secure write to the Invalidate by Way, offset 0x77C, to invalidate all entries in cache:
   - Write 0xFFFF to 0x77C
   - Poll cache maintenance register until invalidate operation is complete.
3. Write to the Lockdown D and Lockdown I Register 9 if required.
4. Write to interrupt clear register to clear any residual raw interrupts set.
5. Write to the Interrupt Mask Register if you want to enable interrupts.
6. Write to Control Register 1 with the LSB set to 1 to enable the cache.

If you write to the Auxiliary, Tag RAM Latency, or Data RAM Latency Control

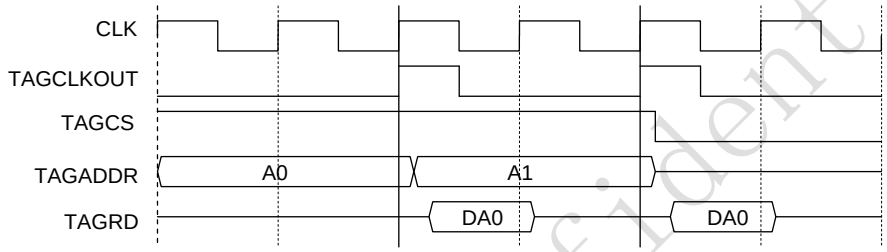
Register with the L2 cache enabled, this results in a SLVERR. You must disable the L2 cache by writing to the Control Register 1 before writing to the Auxiliary, Tag RAM Latency, or Data RAM Latency Control Register.

8.5.3 L2 ram latency programming

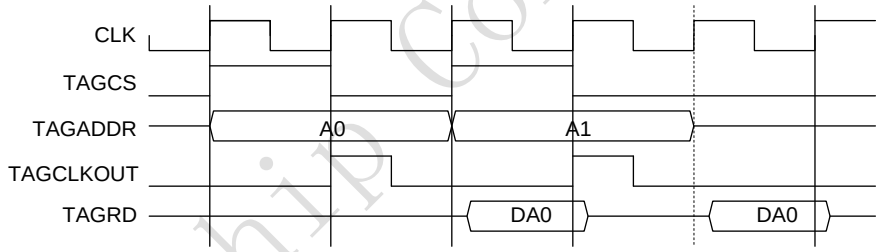
Programmable RAM latencies enable the cache controller to manage RAMs requiring several clock cycles for dealing with accesses. For each RAM, there are three programmable latencies:

- setup
- read access
- write access.

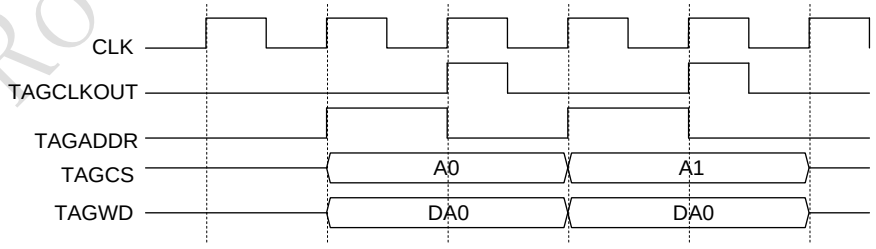
Setup latency is the number of cycles that the RAM control signals remain valid prior to the RAM clock edge.Following figure shows a timing diagram where the tag RAM setup latency has been programmed with the value 0x1.



Read access latency is the number of cycles taken by the read data to become valid after the RAM clock edge.Following figure shows a timing diagram where the tag RAM read latency has been programmed with the value 0x1.



Write access latency is the minimum number of cycles between a RAM clock edge for a write access and the next RAM clock edge corresponding to another access, read or write. Following figure shows a timing diagram where the tag RAM write access latency has been programmed with the value 0x1.



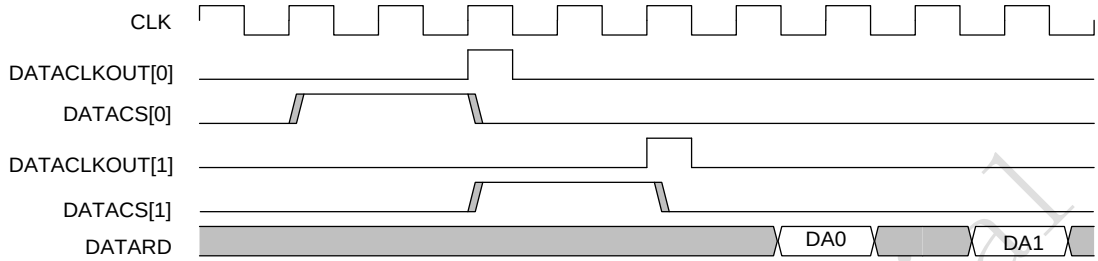
In typical use, the tag ram latency constrained for setup/read/write and correspond register values should be set are like following:

Item	Setup latency	Read latency	Write latency
Cycles	1	1	1
Register filed value	0x0	0x0	0x0

As the tag ram's min period width is 0.8ns, the write latency must be set to 2 cycles if the core is over-frequency to 1.25Ghz.

The data ram setup/read/write latency is like the tag ram. The only difference is data ram use the banking technology. Following figure shows the benefit of the banking when two consecutive reads targetingdifferent banks are treated with the following programmed latencies:

- Data RAM setup latency = 2 cycles, programmed value = 0x1
- Data RAM read latency = 4 cycles, programmed value = 0x3.



In typical use, the data ram latency constrained for setup/read/write and correspond register values should be set are like following:

Item	Setup latency	Read latency	Write latency
Cycles	2	4	1
Register filed value	0x1	0x3	0x0

The data ram read latency is constrained as 4-cycles. Following table shows the data ram read latency's most reasonable value under different core frequency.

Core frequency mhz	0~380	380~560	560~750	>750
Cycles	2	3	4	4
Register filed value	0x0	0x1	0x2	0x3

The data ram setup latency is constrained as 2-cycles. Following table shows the data ram read latency's most reasonable value under different core frequency.

Core frequency mhz	0~560	560~750	>750
Cycles	1	2	2
Register filed value	0x0	0x1	0x1

8.5.4 L2 data ram mutiplexing

The data ram size is 512KB and reside in the pd_cpu power domain. There are 3 ways to use the data ram:

- 512KB used as l2c data ram
- 512KB used as share memory
- 256KB used as l2c data ram, the rest 256KB used as share memory

It's default to use the 512KB ram as l2c data ram.

The register grf_soc_con0[6:5] need to be set to 0x01 , when use 512KB as share memory. And need to be set to 0x10, when use 256KB as share memory.

The register grf_cpu_con0[11:9] need to be set to 0x001 , when use 256KB as share memory.

Chapter 9 AXI interconnect

The chip-level interconnect consists of one cpu_sys interconnect and peri_sys interconnects. It enables communication among themodules and subsystems in the device.

The cpu_sys interconnect handles many types of data transfers, especially exchanges with system-on-chip (SoC)/external memories. It transfers data with amaximum width of 128 bits from the initiator to the target. It is a little-endian platform.

The peri_sys interconnect belong to peri system which is responsible for peripheral devices control such as usb device , flash device , uart , spi etc.

Chapter 10 DMAC0(DMA Controller)

10.1 Overview

This device supports 2 Direct Memory Access (DMA) tops, one for cpu system (DMAC0), and the other one for Peripheral system(DMAC1).Both of these two dma support transfers between memory and memory, peripheral and memory.

DMAC0 supports TrustZone technology and is under secure state after reset.The secure state can be changed by configuring TZPC module.

DMAC0 is mainly used for data transfer of the following slaves: I2S0/I2S1/ SPDIF/UART0/Embedded SRAM and transfer data from/to external DDR SDRAM.

Following table shows the DMAC0 peripheral request mapping scheme.

Table 10-1DMAC0 Request Mapping Table

Req number	Source	Polarity
0	Uart0 tx	High level
1	Uart0 rx	High level
2	Uart1 tx	High level
3	Uart1 rx	High level
4	I2S0/PCM(8ch) tx	High level
5	I2S0/PCM(8ch) rx	High level
6	I2S1/PCM(2ch) tx	High level
7	I2S1/PCM(2ch) rx	High level
8	SPDIF tx	High level
9	I2S2/PCM(2ch) tx	High level
10	I2S2/PCM(2ch) rx	High level

DMAC0 supports the following features:

- Supports Trustzone technology.
- Supports 10 perihpral request.
- Up to 64bits data size.
- 6 channel at the same time.
- Up to burst 16.
- 1 interrupt output and one abort output.
- Supports 32 MFIFO depth.

10.2 Block Diagram

Figure 10-1 shows the block diagram of DMAC0

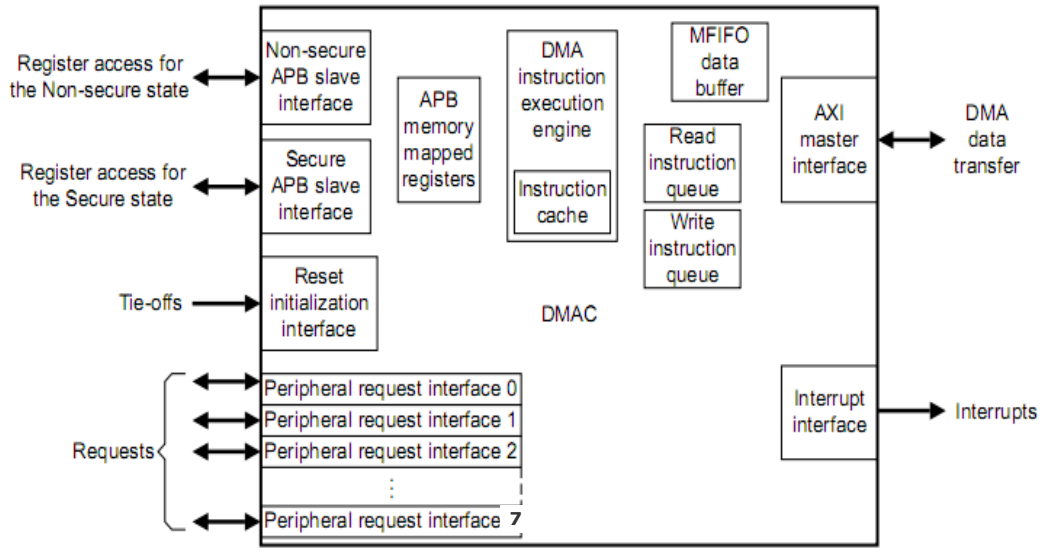


Fig. 10-1 Block diagram of dmac0

As the DMAC0 supports Trustzone technology, so dual APB interfaces enable the operation of the DMAC0 to be partitioned into the Secure state and Non-secure state. You can use the APB interfaces to access status registers and also directly execute instructions in the DMAC0.The default interface after reset is secure apb interface.

10.3 Function Description

10.3.1 Introduction

The DMAC contains an instruction processing block that enables it to process program code that controls a DMA transfer. The program code is stored in a region of system memory that the DMAC accesses using its AXI interface. The DMAC stores instructions temporarily in a cache.

DMAC0 supports 7 channels,each channel capable of supporting a single concurrent thread of DMA operation. In addition, a single DMA manager thread exists, and you can use it to initialize the DMA channel threads. The DMAC executes up to one instruction for each AXI clock cycle. To ensure that it regularly executes each active thread, it alternates by processing the DMA manager thread and then a DMA channel thread. It
uses a round-robin process when selecting the next active DMA channel thread to execute.

The DMAC uses variable-length instructions that consist of one to six bytes. It provides a separate Program Counter (PC) register for each DMA channel. When a thread requests an instruction from an address, the cache performs a look-up. If a cache hit occurs, then the cache immediately provides the data. Otherwise, the thread is stalled while the DMAC uses the AXI interface to perform a cache line fill. If an instruction is greater than 4 bytes, or spans the end of a cache line, the DMAC performs multiple cache accesses to fetch the instruction.

When a cache line fill is in progress, the DMAC enables other threads to access the cache, but if another cache miss occurs, this stalls the pipeline until the first line fill is complete.

When a DMA channel thread executes a load or store instruction, the DMAC adds the instruction to the relevant read or write queue. The DMAC uses these queues as an instruction storage buffer prior to it issuing the instructions on the AXI bus. The DMAC also contains a Multi First-In-First-Out (MFIFO) data buffer that it uses to store data that it reads, or writes, during a DMA transfer.

10.3.2 Operating states

Figure shows the operating states for the DMA manager thread and DMA channel threads.

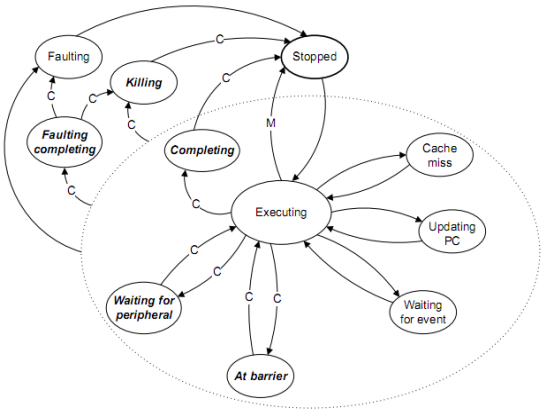


Fig. 10-2 DMAC0 operation states

Note:
arcs with no letter designator indicate state transitions for the DMA manager and DMA channel threads, otherwise use is restricted as follows:
C DMA channel threads only.
M DMA manager thread only.

After the DMAC exits from reset, it sets all DMA channel threads to the stopped state, and the status of boot_from_pc(tie-off interface of dmac) controls the DMA manager thread state:
boot_from_pc is LOW :DMA manager thread moves to the Stopped state.
boot_from_pc is HIGH :DMA manager thread moves to the Executing state.

10.4 Register Description

10.4.1 Register summary

Name	Offset	Size	Reset Value	Description
DMAC0_DSR	0x0000	W	0x0	DMA Status Register.
DMAC0_DPC	0x0004	W	0x0	DMA Program Counter Register.
-	-	-	-	reserved
DMAC0_INTEN	0x0020	W	0x0	Interrupt Enable Register
DMAC0_EVENT_RIS	0x0024	W	0x0	Event Status Register.
DMAC0_INTMIS	0x0028	W	0x0	Interrupt Status Register
DMAC0_INTCLR	0x002C	W	0x0	Interrupt Clear Register
DMAC0_FSRD	0x0030	W	0x0	Fault Status DMA Manager Register.
DMAC0_FSRC	0x0034	W	0x0	Fault Status DMA Channel

				Register.
DMAC0_FTRD	0x0038	W	0x0	Fault Type DMA Manager Register.
-	-	-	-	reserved
DMAC0_FTR0	0x0040	W	0x0	Fault type for DMA Channel 0
DMAC0_FTR1	0x0044	W	0x0	Fault type for DMA Channel 1
DMAC0_FTR2	0x0048	W	0x0	Fault type for DMA Channel 2
DMAC0_FTR3	0x004C	W	0x0	Fault type for DMA Channel 3
DMAC0_FTR4	0x0050	W	0x0	Fault type for DMA Channel 4
DMAC0_FTR5	0x0054	W	0x0	Fault type for DMA Channel 5
-	-	-	-	reserved
DMAC0_CSR0	0x0100	W	0x0	Channel Status for DMA Channel 0
DMAC0_CSR1	0x0108	W	0x0	Channel Status for DMA Channel 1
DMAC0_CSR2	0x0110	W	0x0	Channel Status for DMA Channel 2
DMAC0_CSR3	0x0118	W	0x0	Channel Status for DMA Channel 3
DMAC0_CSR4	0x0120	W	0x0	Channel Status for DMA Channel 4
DMAC0_CSR5	0x0128	W	0x0	Channel Status for DMA Channel 5
DMAC0_CPC0	0x0104	W	0x0	Channel PC for DMA Channel 0
DMAC0_CPC1	0x010c	W	0x0	Channel PC for DMA Channel 1
DMAC0_CPC2	0x0114	W	0x0	Channel PC for DMA Channel 2
DMAC0_CPC3	0x011c	W	0x0	Channel PC for DMA Channel 3
DMAC0_CPC4	0x0124	W	0x0	Channel PC for DMA Channel 4
DMAC0_CPC5	0x012c	W	0x0	Channel PC for DMA Channel 5
DMAC0_SAR0	0x0400	W	0x0	Source Address for DMA Channel 0
DMAC0_SAR1	0x0420	W	0x0	Source Address for DMA Channel 1
DMAC0_SAR2	0x0440	W	0x0	Source Address for DMA Channel 2
DMAC0_SAR3	0x0460	W	0x0	Source Address for DMA Channel 3
DMAC0_SAR4	0x0480	W	0x0	Source Address for DMA Channel 4
DMAC0_SAR5	0x04a0	W	0x0	Source Address for DMA Channel 5
DMAC0_DAR0	0x0404	W	0x0	Dest Address for DMAChannel 0
DMAC0_DAR1	0x0424	W	0x0	Dest Address for DMAChannel 1
DMAC0_DAR2	0x0444	W	0x0	Dest Address for DMAChannel 2
DMAC0_DAR3	0x0464	W	0x0	Dest Address for DMAChannel 3
DMAC0_DAR4	0x0484	W	0x0	Dest Address for DMAChannel 4
DMAC0_DAR5	0x04a4	W	0x0	Dest Address for DMAChannel 5

DMAC0_CCR0	0x0408	W	0x0	Channel Control for DMA Channel 0
DMAC0_CCR1	0x0428	W	0x0	Channel Control for DMA Channel 1
DMAC0_CCR2	0x0448	W	0x0	Channel Control for DMA Channel 2
DMAC0_CCR3	0x0468	W	0x0	Channel Control for DMA Channel 3
DMAC0_CCR4	0x0488	W	0x0	Channel Control for DMA Channel 4
DMAC0_CCR5	0x04a8	W	0x0	Channel Control for DMA Channel 5
DMAC0_LC0_0	0x040C	W	0x0	Loop Counter 0 for DMA Channel 0
DMAC0_LC0_1	0x042C	W	0x0	Loop Counter 0 for DMA Channel 1
DMAC0_LC0_2	0x044C	W	0x0	Loop Counter 0 for DMA Channel 2
DMAC0_LC0_3	0x046C	W	0x0	Loop Counter 0 for DMA Channel 3
DMAC0_LC0_4	0x048C	W	0x0	Loop Counter 0 for DMA Channel 4
DMAC0_LC0_5	0x04aC	W	0x0	Loop Counter 0 for DMA Channel 5
DMAC0_LC1_0	0x0410	W	0x0	Loop Counter 1 for DMA Channel 0
DMAC0_LC1_1	0x0430	W	0x0	Loop Counter 1 for DMA Channel 1
DMAC0_LC1_2	0x0450	W	0x0	Loop Counter 1 for DMA Channel 2
DMAC0_LC1_3	0x0470	W	0x0	Loop Counter 1 for DMA Channel 3
DMAC0_LC1_4	0x0490	W	0x0	Loop Counter 1 for DMA Channel 4
DMAC0_LC1_5	0x04b0	W	0x0	Loop Counter 1 for DMA Channel 5
-	-	-	-	reserved
DMAC0_DBGSTDMAC0_ATUS	0x0D00	W	0x0	Debug Status Register.
DMAC0_DBGCMD	0x0D04	W	0x0	Debug Command Register.
DMAC0_DBGINST0	0x0D08	W	0x0	Debug Instruction-0 Register.
DMAC0_DBGINST1	0x0D0C	W	0x0	Debug Instruction-1 Register.
DMAC0_CR0	0x0E00	W		Configuration Register 0.
DMAC0_CR1	0x0E04	W		Configuration Register 1.
DMAC0_CR2	0x0E08	W		Configuration Register 2.
DMAC0_CR3	0x0E0C	W		Configuration Register 3.
DMAC0_CR4	0x0E10	W		Configuration Register 4.
DMAC0_CRDn	0x0E14	W		Configuration Register Dn.
DMAC0_WD	0x0E80	W	0x0	Watchdog Register.

Notes:
*Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

10.4.2 Detail Register Description

DMAC0_DSR
Address:Operational Base+0x0
DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	-	-	Reserved
9	R	0x0	Provides the security status of the DMA manager thread: 0 = DMA manager operates in the Secure state 1 = DMA manager operates in the Non-secure state.
8:4	R	0x0	When the DMA manager thread executes a DMAWFE instruction, it waits for the following event to occur: b00000 = event[0] b00001 = event[1] b00010 = event[2] … b11111 = event[31].
3:0	R	0x0	The operating state of the DMA manager: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101-b1110 = reserved b1111 = Faulting.

DMAC0_DPC
Address:Operational Base+0x4
DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA manager thread

DMAC0_INTEN
Address:Operational Base+0x20
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	Program the appropriate bit to control how the DMAC responds when it executes DMASEV: Bit [N] = 0 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request. Bit [N] = 1 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interrupt request.

DMAC0_EVENT_RIS
Address:Operational Base+0x24
Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Returns the status of the event-interrupt resources: Bit [N] = 0 Event N is inactive or irq[N] is LOW. Bit [N] = 1 Event N is active or irq[N] is HIGH.

DMAC0_INTMIS
Address:Operational Base+0x28
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the status of the interrupts that are active in the DMAC: Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW. Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH

DMAC0_INTCLR
Address:Operational Base+0x2c
Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	Controls the clearing of the irq outputs: Bit [N] = 0 The status of irq[N] does not change. Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change.

DMAC0_FSRD
Address:Operational Base+0x30
Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the fault status of the DMA manager. Read as: 0 = the DMA manager thread is not in the Faulting state 1 = the DMA manager thread is in the Faulting state.

DMAC0_FSRC
Address:Operational Base+0x34
Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Each bit provides the fault status of the corresponding channel. Read as: Bit [N] = 0 No fault is present on DMA channel N. Bit [N] = 1 DMA channel N is in the Faulting or Faulting completing state.

DMAC0_FTRD
Address:Operational Base+0x38
Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	-	-	reserved
30	R	0x0	If the DMA manager aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface.
29:17	-	-	reserved
16	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA manager performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response
15:6	-	-	reserved
5	R	0x0	Indicates if the DMA manager was attempting to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAWFE or DMASEV 1 = a DMA manager thread in the Non-secure state attempted to execute either: • DMAWFE to wait for a secure event • DMASEV to create a secure event or secure interrupt
4	R	0x0	ndicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAGO 1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state.
3:2	-	-	reserved
1	R	0x0	Indicates if the DMA manager was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand.
0	R	0x0	Indicates if the DMA manager was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction.

DMAC0_FTR0~DMAC0_FTR5
Address:Operational Base+0x40
Operational Base+0x44
Operational Base+0x48
Operational Base+0x4c

Operational Base+0x50
Operational Base+0x54
Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	R	0x0	Indicates if the DMA channel has locked-up because of resource starvation: 0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	R	0x0	If the DMA channel aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	-	-	reserved
18	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	R	0x0	Indicates the AXI response that the DMAC receives on the BRESP bus, after the DMA channel thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort.
16	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort.
15:14	-	-	reserved
13	R	0x0	Indicates if the MFIFO did not contain the data to enable the DMAC to perform the DMAST: 0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort.
12	R	0x0	Indicates if the MFIFO prevented the DMA channel thread from executing DMALD or DMAST. Depending on the instruction: DMALD 0 = MFIFO contains sufficient space 1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to

			enable DMAST to complete. This fault is an imprecise abort
11:8	-	-	reserved
7	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to program the CCRn Register to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: • DMAWFP to wait for a secure peripheral • DMALDP or DMASTP to notify a secure peripheral • DMAFLUSHP to flush a secure peripheral. This fault is a precise abort.
5	R	0x0	Indicates if the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: • DMAWFE to wait for a secure event • DMASEV to create a secure event or secure interrupt. This fault is a precise abort.
4:2	-	-	reserved
1	R	0x0	Indicates if the DMA channel thread was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort.
0	R	0x0	Indicates if the DMA channel thread was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMAC0_CSR0~DMAC0_CSR5

Address:Operational Base+0x100

Operational Base+0x108

Operational Base+0x110
Operational Base+0x118
Operational Base+0x120
Operational Base+0x128

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21	R	0x0	The channel non-secure bit provides the security of the DMA channel: 0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	-	-	reserved
15	R	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the periph operand was set: 0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	R	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the burst or single operand were set: 0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set.
13:9	-	-	reserved
8:4	R	0x0	If the DMA channel is in the Waiting for event state or the Waiting for peripheral state then these bits indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31
3:0	R	0x0	The channel status encoding is: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101 = At barrier b0110 = reserved b0111 = Waiting for peripheral b1000 = Killing

			b1001 = Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMAC0_CPC0~DMAC0_CPC5

Address:Operational Base+0x104
Operational Base+0x10c
Operational Base+0x114
Operational Base+0x11c
Operational Base+0x124
Operational Base+0x12c

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA channel n thread

DMAC0_SAR0~DMAC0_SAR5

Address:Operational Base+0x400
Operational Base+0x420
Operational Base+0x440
Operational Base+0x460
Operational Base+0x480
Operational Base+0x4a0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the source data for DMA channel n

DMAC0_DAR0~DMAC0_DAR5

Address:Operational Base+0x404
Operational Base+0x424
Operational Base+0x444
Operational Base+0x464
Operational Base+0x484
Operational Base+0x4a4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the Destinationdata for DMA channel n

DMAC0_CCR0~DMAC0_CCR5

Address:Operational Base+0x408
Operational Base+0x428
Operational Base+0x448
Operational Base+0x468
Operational Base+0x488
Operational Base+0x4a8

Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	-	-	reserved
27:25	R	0x0	Programs the state of AWCACHE[3,1:0]a when the DMAC writes the destination data. Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW

			1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	R	0x0	Programs the state of AWPROT[2:0]a when the DMAC writes the destination data. Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	R	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it writes the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size
17:15	R	0x0	For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination: b000 = writes 1 byte per beat b001 = writes 2 bytes per beat b010 = writes 4 bytes per beat b011 = writes 8 bytes per beat b100 = writes 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	R	0x0	Programs the burst type that the DMAC performs when it writes the destination data: 0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	R	0x0	Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data. Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH.
10:8	R	0x0	Programs the state of ARPROT[2:0]a when the

			DMAC reads the source data. Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH.
7:4	R	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it reads the source data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size
3:1	R	0x0	For each beat within a burst, it programs the number of bytes that the DMAC reads from the source: b000 = reads 1 byte per beat b001 = reads 2 bytes per beat b010 = reads 4 bytes per beat b011 = reads 8 bytes per beat b100 = reads 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size
0	R	0x0	Programs the burst type that the DMAC performs when it reads the source data: 0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMAC0_LC0_0~DMAC0_LC0_5

Address:Operational Base+0x40c
Operational Base+0x42c
Operational Base+0x44c
Operational Base+0x46c
Operational Base+0x48c
Operational Base+0x4ac

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 0 iterations

DMAC0_LC1_0~DMAC0_LC1_5
Address:Operational Base+0x410
Operational Base+0x430
Operational Base+0x450
Operational Base+0x470
Operational Base+0x490
Operational Base+0x4b0
Loop Counter 1 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 1 iterations

DMAC0_DBGSTATUS
Address:Operational Base+0xd00
Debug Status Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved
1:0	R	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved.

DMAC0_DBGCMD
Address:Operational Base+0xd04
Debug Command Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved
1:0	W	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved

DMAC0_DBGINST0
Address:Operational Base+0xd08
Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 1
23:16	W	0x0	Instruction byte 0
15:11	-	-	reserved
10:8	W	0x0	DMA channel number: b000 = DMA channel 0 b001 = DMA channel 1 b010 = DMA channel 2 … b111 = DMA channel 7
7:1	-	-	reserved
0	W	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC0_DBGINST1
Address:Operational Base+0xd0c
Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 5
23:16	W	0x0	Instruction byte 4
15:8	W	0x0	Instruction byte 3
7:0	W	0x0	Instruction byte 2

DMAC0_CR0
Address:Operational Base+0xe00
Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21:17	R	0x2	Number of interrupt outputs that the DMAC provides: b00000 = 1 interrupt output, irq[0] b00001 = 2 interrupt outputs, irq[1:0] b00010 = 3 interrupt outputs, irq[2:0] . . . b11111 = 32 interrupt outputs, irq[31:0].
16:12	R	0x7	Number of peripheral request interfaces that the DMAC provides: b00000 = 1 peripheral request interface b00001 = 2 peripheral request interfaces b00010 = 3 peripheral request interfaces . . . b11111 = 32 peripheral request interfaces.
11:7	-	-	reserved
6:4	R	0x5	Number of DMA channels that the DMAC supports: b000 = 1 DMA channel b001 = 2 DMA channels b010 = 3 DMA channels . . . b111 = 8 DMA channels.
3	-	-	reserved
2	R	0x0	Indicates the status of the boot_manager_ns signal when the DMAC exited from reset: 0 = boot_manager_ns was LOW 1 = boot_manager_ns was HIGH.
1	R	0x0	Indicates the status of the boot_from_pc signal when the DMAC exited from reset: 0 = boot_from_pc was LOW 1 = boot_from_pc was HIGH
0	R	0x1	Supports peripheral requests: 0 = the DMAC does not provide a peripheral

			request interface 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.

DMAC0_CR1
Address:Operational Base+0xe04
Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:4	R	0x5	[7:4] num_i-cache_lines Number of i-cache lines: b0000 = 1 i-cache line b0001 = 2 i-cache lines b0010 = 3 i-cache lines … b1111 = 16 i-cache lines.
3	-	-	reserved
2:0	R	0x7	The length of an i-cache line: b000-b001 = reserved b010 = 4 bytes b011 = 8 bytes b100 = 16 bytes b101 = 32 bytes b110-b111 = reserved

DMAC0_CR2
Address:Operational Base+0xe08
Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC0_CR3
Address:Operational Base+0xe0c
Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the security state of an event-interrupt resource: Bit [N] = 0 Assigns event<N> or irq[N] to the Secure state. Bit [N] = 1 Assigns event<N> or irq[N] to the Non-secure state.

DMAC0_CR4
Address:Operational Base+0xe10
Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	R	0x6	Provides the security state of the peripheral request interfaces: Bit [N] = 0 Assigns peripheral request interface N to the Secure state. Bit [N] = 1 Assigns peripheral request interface N to the Non-secure state

DMAC0_CRDn
Address:Operational Base+0xe14
DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	-	-	reserved
29:20	R	0x20	The number of lines that the data buffer contains: b000000000 = 1 line b000000001 = 2 lines … b111111111 = 1024 lines
19:16	R	0x9	The depth of the read queue: b0000 = 1 line b0001 = 2 lines . . . b1111 = 16 lines.
15	-	-	reserved
14:12	R	0x4	Read issuing capability that programs the number of outstanding read transactions: b000 = 1 b001 = 2 … b111 = 8
11:8	R	0x7	The depth of the write queue: b0000 = 1 line b0001 = 2 lines … b1111 = 16 lines.
7	-	-	reserved
6:4	R	0x3	Write issuing capability that programs the number of outstanding write transactions: b000 = 1 b001 = 2 … b111 = 8
3	-	-	reserved
2:0		0x3	The data bus width of the AXI interface: b000 = reserved b001 = reserved b010 = 32-bit b011 = 64-bit b100 = 128-bit b101-b111 = reserved.

DMAC0_WD
Address:Operational Base+0xe80
DMA Watchdog Register

Bit	Attr	Reset Value	Description
-	-	-	reserved
0	RW	0x0	Controls how the DMAC responds when it detects a lock-up condition:

			0 = the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1 = the DMAC sets irq_abort HIGH.

10.5 Timing Diagram

Following picture shows the relationship between dma_req and dma_ack.

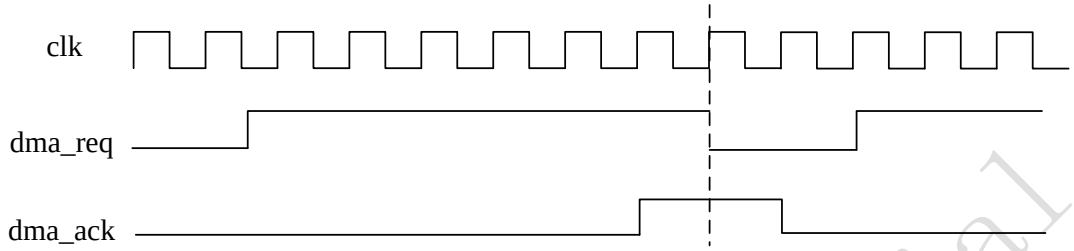


Fig. 10-3 DMAC0 request and acknowledge timing

10.6 Interface Description

DMAC0 has the following tie-off signals.It can be configured by GRF register or TZPC register.(Please refer to these two chapters to find how to configure)

interface	Reset value	Control source
boot_addr	0x0	GRF
boot_from_pc	0x0	GRF
boot_manager_ns	0x1	TZPC
boot_irq_ns	0x6	TZPC
boot_periph_ns	0xff	TZPC

boot_addr

Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

boot_from_pc

Controls the location in which the DMAC executes its initial instruction, after it exits from reset:

0 = DMAC waits for an instruction from either APB interface

1 = DMA manager thread executes the instruction that is located at the address that

boot_manager_ns

When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:

0 = assigns DMA manager to the Secure state

1 = assigns DMA manager to the Non-secure state.

boot_irq_ns

Controls the security state of an event-interrupt resource, when the DMAC exits from reset:

boot_irq_ns[x] is LOW

The DMAC assigns event<x> or irq[x] to the Secure state.

boot_irq_ns[x] is HIGH

The DMAC assigns event<x> or irq[x] to the Non-secure state.

boot_periph_ns

Controls the security state of a peripheral request interface, when the DMAC exits from reset:

boot_periph_ns[x] is LOW

The DMAC assigns peripheral request interface x to the Secure state.

boot_periph_ns[x] is HIGH

The DMAC assigns peripheral request interface x to the Non-secure state.

10.7 Application Notes

10.7.1 Using the APB slave interfaces

You must ensure that you use the appropriate APB interface, depending on the security state in which the boot_manager_ns initializes the DMAC to operate. For example, if the DMAC is in the Secure state, you must issue the instruction using the secure APB interface, otherwise the DMAC ignores the instruction. You can use the secure APB interface, or the non-secure APB interface, to start or restart a DMA channel when the DMAC is in the Non-secure state.

The necessary steps to start a DMA channel thread using the debug instruction registers as following:

1. Create a program for the DMA channel.

2. Store the program in a region of system memory.

3. Poll the DBGSTATUS Register to ensure that debug is idle, that is, the dbgstatus bit is 0.

4. Write to the DBGINST0 Register and enter the:
• Instruction byte 0 encoding for DMAGO.
• Instruction byte 1 encoding for DMAGO.
• Debug thread bit to 0. This selects the DMA manager thread.

5. Write to the DBGINST1 Register with the DMAGO instruction byte [5:2] data, see Debug Instruction-1 Register o. You must set these four bytes to the address of the first instruction in the program, that was written to system memory in step 2.

6. Writing zero to the DBGCMD Register. The DMAC starts the DMA channel thread and sets the dbgstatus bit to 1.

10.7.2 Security usage

When the DMAC exits from reset, the status of the configuration signals that tie-off signals which descripted in chapter 10.6.

DMA manager thread is in the Secure state

If the DNS bit is 0, the DMA manager thread operates in the Secure state and it only performs secure instruction fetches. When a DMA manager thread in the Secure state processes:

DMAGO

It uses the status of the ns bit, to set the security state of the DMA

channel thread by writing to the CNS bit for that channel.

DMAWFE
It halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit.

DMASEV
It sets the corresponding bit in the INT_EVENT_RIS Register, irrespective of the security state of the corresponding INS bit.

DMA manager thread is in the Non-secure state
If the DNS bit is 1, the DMA manager thread operates in the Non-secure state, and it only performs non-secure instruction fetches. When a DMA manager thread in the Non-secure state processes:

DMAGO
The DMAC uses the status of the ns bit, to control if it starts a DMA channel thread. If:

ns = 0
The DMAC does not start a DMA channel thread and instead it:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager
3. Sets the dmago_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

ns = 1
The DMAC starts a DMA channel thread in the Non-secure state and programs the CNS bit to be non-secure.

DMAWFE
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:

INS = 0
The event is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.
3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

INS = 1
The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV
The DMAC uses the status of the corresponding INS bit, in the CR3Register, to control if it creates the event-interrupt. If:

INS = 0
The event-interrupt resource is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the FSRD Register, see Fault Status DMA Manager Register.

3. Sets the mgr_evnt_err bit in the FTRD Register, see Fault Type DMA Manager Register.
4. Moves the DMA manager to the Faulting state.

INS = 1
The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMA channel thread is in the Secure state
When the CNS bit is 0, the DMA channel thread is programmed to operate in the Secure state and it only performs secure instruction fetches.
When a DMA channel thread in the Secure state processes the following instructions:

DMAWFE
The DMAC halts execution of the thread until the event occurs. When the event occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMASEV
The DMAC creates the event-interrupt, irrespective of the security state of the corresponding INS bit, in the CR3 Register.

DMAWFP
The DMAC halts execution of the thread until the peripheral signals a DMA request. When this occurs, the DMAC continues execution of the thread, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMALDP, DMASTP
The DMAC sends a message to the peripheral to communicate that data transfer is complete, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

DMAFLUSHP
The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status, irrespective of the security state of the corresponding PNS bit, in the CR4 Register.

When a DMA channel thread is in the Secure state, it enables the DMAC to perform secure and non-secure AXI accesses

DMA channel thread is in the Non-secure state

When the CNS bit is 1, the DMA channel thread is programmed to operate in the Non-secure state and it only performs non-secure instruction fetches.
When a DMA channel thread in the Non-secure state processes the following instructions:

DMAWFE
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it waits for the event. If:

INS = 0 The event is in the Secure state. The DMAC:

1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

INS = 1 The event is in the Non-secure state. The DMAC halts execution of the thread and waits for the event to occur.

DMASEV
The DMAC uses the status of the corresponding INS bit, in the CR3 Register, to control if it creates the event. If:

INS = 0 The event-interrupt resource is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_evnt_err bit in the FTRn Register, see Fault Type DMA Channel Registers .
4. Moves the DMA channel to the Faulting completing state.

INS = 1 The event-interrupt resource is in the Non-secure state. The DMAC creates the event-interrupt.

DMAWFP
The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it waits for the peripheral to signal a request. If:

PNS = 0 The peripheral is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1 The peripheral is in the Non-secure state. The DMAC halts execution of the thread and waits for the peripheral to signal a request.

DMALDP, DMASTP
The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends an acknowledgement to the peripheral. If:

PNS = 0 The peripheral is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Register.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1 The peripheral is in the Non-secure state. The DMAC sends a message to the peripheral to communicate when the data transfer is complete.

DMAFLUSHP
The DMAC uses the status of the corresponding PNS bit, in the CR4 Register, to control if it sends a flush request to the peripheral. If:

PNS = 0 The peripheral is in the Secure state. The DMAC:
1. Executes a NOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Registe.
3. Sets the ch_periph_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel to the Faulting completing state.

PNS = 1 The peripheral is in the Non-secure state. The DMAC clears the state of the peripheral and sends a message to the peripheral to resend its level status.

When a DMA channel thread is in the Non-secure state, and a DMAMOV CCR instruction attempts to program the channel to perform a secure AXI transaction, the DMAC:

1. Executes a DMANOP.
2. Sets the appropriate bit in the FSRC Register that corresponds to the DMA channel number. See Fault Status DMA Channel Registe.
3. Sets the ch_rdwr_err bit in the FTRn Register, see Fault Type DMA Channel Registers.
4. Moves the DMA channel thread to the Faulting completing state.

10.7.3 Programming restrictions

Fixed unaligned bursts
The DMAC does not support fixed unaligned bursts. If you program the following conditions, the DMAC treats this as a programming error:
Unaligned read
• src_inc field is 0 in the CCRn Register
• the SARn Register contains an address that is not aligned to the size of data that the src_burst_size field contain
Unaligned write
• dst_inc field is 0 in the CCRn Register
• the DARn Register contains an address that is not aligned to the size of data that the dst_burst_size field contains

Endian swap size restrictions
If you program the endian_swap_size field in the CCRn Register, to enable a DMA channel to perform an endian swap then you must set the corresponding SARn Register and the corresponding DARn Register to contain an address that is aligned to the value that the endian_swap_size field contains.

Updating DMA channel control registers during a DMA cyclerestrictions

Prior to the DMAC executing a sequence of DMALD and DMAST instructions, the values you program in to the CCRn Register, SARn Register, and DARn Register control the data byte lane manipulation that the DMAC performs when it transfers the data from the source address to the destination address. You'd better not update these registers during a DMA cycle.

Resource sharing between DMA channels

DMA channel programs share the MFIFO data storage resource. You must not start a set ofconcurrently running DMA channel programs with a resource requirement that exceeds theconfigured size of the MFIFO. If you exceed this limit then the DMAC might lock up and generate a Watchdog abort.

10.7.4 Unaligned transfers may be corrupted

For a configuration with more than one channel, if any of channels 1 to 7 is performing transfers between certain types of misaligned source and destination addresses, then the output data may be corrupted by the action of channel 0.

Data corruption might occur if all of the following are true:
1. Two beats of AXI read data are received for one of channels 1 to 7.
2. Source and destination address alignments mean that each read data beat is split across two lines in the data buffer (see Splitting data, below).
3. There is one idle cycle between the two read data beats .
4. Channel 0 performs an operation that updates channel control information during this idle cycle (see Updates to channel control information, below)

Splitting data
Depending upon the programmed values for the DMA transfer, one beat of read data from the AXI interface mneed to be split across two lines in the internal data buffer. This occurs when the read data beat contains datbytes which will be written to addresses that wrap around at the AXI interface data width, so that these bytes could not be transferred by a single AXI write data beat of the full interface width.

Most applications of DMA-330 do not split data in this way, so are NOT vulnerable to data corruption from thisdefect.
The following cases are NOT vulnerable to data corruption because they do not split data:
• Byte lane offset between source and destination addresses is 0
When source and destination addresses have the same byte lane alignment, the offset is 0 and a wrap
operation that splits data cannot occur.
• Byte lane offset between source and destination addresses is a multiple of source size

Source size in CCRn	Allowed offset between SARn and DARn
SS8	any offset allowed.
SS16	0,2,4,6,8,10,12,14
SS32	0,4,8,12
SS64	0,8

10.7.5 Interrupt shares between channel.

As the DMAC0 does not record which channel (or list of channels) have asserted an interrupt. So it will depend on your program and whether any of the visible information for that program can be used to determine progress, and help

identify the interrupt source.

There are 4 likely information sources that can be used to determine the progress made by a program:
- Program counter (PC)
- Source address
- Destination address
- Loop counters (LC)

For example, a program might emit an interrupt each time that it iterates around a loop. In this case, the interrupt service routine (ISR) would need to store the loop value of each channel when it is called, and then compare against the new value when it is next called. A change in value would indicate that the program has progressed.

The ISR must be carefully written to ensure that no interrupts are lost. The sequence of operations is as follows:
1. Disable interrupts
2. Immediately clear the interrupt in DMA-330
3. Check the relevant registers for both channels to determine which must be serviced
4. Take appropriate action for the channels
5. Re-enable interrupts and exit ISR

10.7.6 Instruction sets

Table 10-2 DMAC Instruction sets

Mnemonic	Instruction	Thread usage: • M = DMA manager • C = DMA channel
DMAADDH	Add Halfword	C
DMAEND	End	M/C
DMAFLUSHP	Flush and notify Peripheral	C
DMAGO	Go	M
DMAKILL	Kill	C
DMALD	Load	C
DMALDP	Load Peripheral	C
DMALP	Loop	C
DMALPEND	Loop End	C
DMALPFE	Loop Forever	C
DMAMOV	Move	C
DMANOP	No operation	M/C
DMARMB	Read Memory Barrier	C
DMASEV	Send Event	M/C
DMAST	Store	C
DMASTP	Store and notify Peripheral	C
DMASTZ	Store Zero	C
DMAWFE	Wai t For Event M	M/C
DMAWFP	Wait For Peripheral	C
DMAWMB	Write Memory Barrier	C
DMAADNH	Add Negative Halfword	C

10.7.7 Assembler directives

In this document, only DMMADNH instruction is took as an example to show the way the instruction assembled. *For the other instructions , please refer to pl330_trm.pdf.*

DMAADNH
Add Negative Halfword adds an immediate negative 16-bit value to the SARn Register or DARn Register, for the DMA channel thread. This enables the DMA C to support 2D DMA operations, or reading or writing an area of memory in a different order to naturally incrementing addresses. See Source Address Regi sters and Destination Address Registers.

The immediate unsigned 16-bit value is one-extended to 32 bits, to create a v alue that is the two's complement representation of a negative number betwe en -65536 and -1, before the DMAC adds it to the address using 32-bit additi on. The DMAC discards the carry bit so that addresses wrap from 0xFFFFFFFF to 0x00000000. The net effect is to subtract between 65536 and 1 from the c urrent value in the Source or Destination Address Register.

Following table shows the instruction encoding.

Imm[15:8]	Imm[7:0]	0	1	0	1	1	1	ra	0

Assembler syntax

DMAADNH <address_register>, <16-bit immediate>

where:
<address_register>
Selects the address register to use. It must be either:
SAR
SARn Register and sets ra to 0.
DAR
DARn Register and sets ra to 1.

<16-bit immediate>
The immediate value to be added to the <address_register>.

You should specify the 16-bit immediate as the number that is to be represen ted in the instruction encoding. For example, DMAADNH DAR, 0xFFF0 causes the value 0xFFFFFFF0 to be added to the current value of the Destination Add ress Register, effectively subtracting 16 from the DAR.

You can only use this instruction in a DMA channel thread.

10.7.8 MFIFO usage

For MFIFO usage , please refer to pl330_trm.pdf

Chapter 11 DMAC1(DMA Controller)

11.1 Overview

DMAC1 does not support TrustZone technology and work under non-secure state only.

DMAC1 is mainly used for data transfer of the following slaves: SMC , HSADC, PID_FILTER, SD/MMC, SDIO, eMMC, HIF, UART1, UART2, UART3, SPI0, SPI1.

Following table shows the DMAC1 request mapping scheme.

Table 11-1DMAC1 Request Mapping Table

Req number	Source	Polarity
0	HSADC/TSI	High level
1	SD/MMC	High level
2	N/A	
3	SDIO	High level
4	eMMC	High level
5	PID_FILTER	High level
6	Uart2 tx	High level
7	Uart2 rx	High level
8	Uart3 tx	High level
9	Uart3 rx	High level
10	Spi0 tx	High level
11	Spi0 rx	High level
12	Spi1 tx	High level
13	Spi1 rx	High level

DMAC1supports the following features:

- Supports 14 perihpral request.
- Up to 64bits data size.
- 7 channel at the same time.
- Up to burst 16.
- 1 interrupts output and one abort output.
- Supports 64 MFIFO depth.

11.2 Block Diagram

Figure 11-1 shows the block diagram of DMAC1

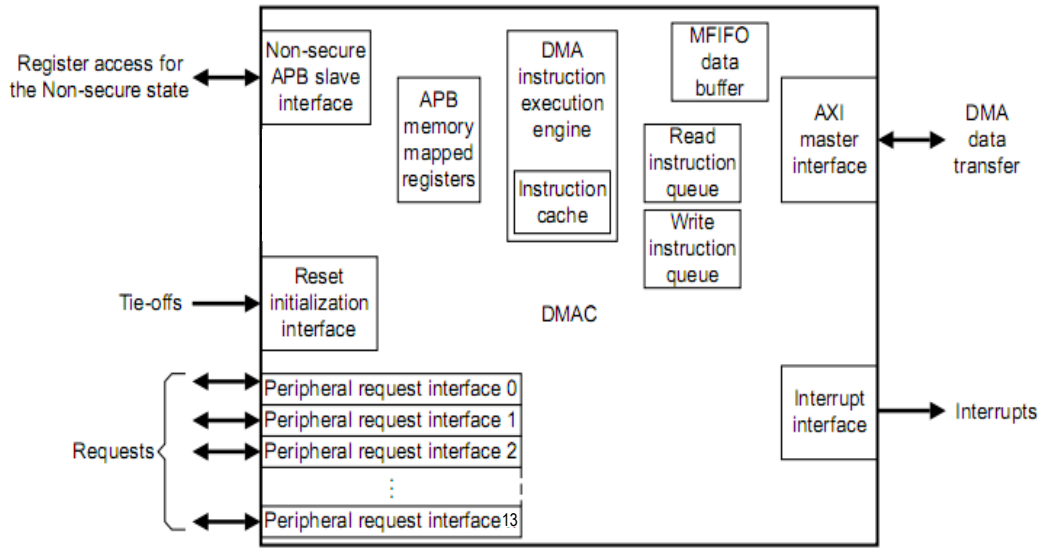


Fig. 11-1 Block diagram of dmac1

11.3 Function Description

Please refer to chapter 10.3 for the similar description.

11.4 Register Description

11.4.1 Register summary

Name	Offset	Size	Reset Value	Description
DMAC1_DSR	0x0000	W	0x0	DMA Status Register.
DMAC1_DPC	0x0004	W	0x0	DMA Program Counter Register.
-	-	-	-	reserved
DMAC1_INTEN	0x0020	W	0x0	Interrupt Enable Register
DMAC1_EVENT_RIS	0x0024	W	0x0	Event Status Register.
DMAC1_INTMIS	0x0028	W	0x0	Interrupt Status Register
DMAC1_INTCLR	0x002C	W	0x0	Interrupt Clear Register
DMAC1_FSRD	0x0030	W	0x0	Fault Status DMA Manager Register.
DMAC1_FSRC	0x0034	W	0x0	Fault Status DMA Channel Register.
DMAC1_FTRD	0x0038	W	0x0	Fault Type DMA Manager Register.
-	-	-	-	reserved
DMAC1_FTR0	0x0040	W	0x0	Fault type for DMA Channel 0
DMAC1_FTR1	0x0044	W	0x0	Fault type for DMA Channel 1
DMAC1_FTR2	0x0048	W	0x0	Fault type for DMA Channel 2
DMAC1_FTR3	0x004C	W	0x0	Fault type for DMA Channel 3
DMAC1_FTR4	0x0050	W	0x0	Fault type for DMA Channel 4
DMAC1_FTR5	0x0054	W	0x0	Fault type for DMA Channel 5
DMAC1_FTR6	0x0058	W	0x0	Fault type for DMA Channel 6
-	-	-	-	reserved
DMAC1_CSR0	0x0100	W	0x0	Channel Status for DMA Channel 0

DMAC1_CSR1	0x0108	W	0x0	Channel Status for DMA Channel 1
DMAC1_CSR2	0x0110	W	0x0	Channel Status for DMA Channel 2
DMAC1_CSR3	0x0118	W	0x0	Channel Status for DMA Channel 3
DMAC1_CSR4	0x0120	W	0x0	Channel Status for DMA Channel 4
DMAC1_CSR5	0x0128	W	0x0	Channel Status for DMA Channel 5
DMAC1_CSR6	0x0130	W	0x0	Channel Status for DMA Channel 6
DMAC1_CPC0	0x0104	W	0x0	Channel PC for DMA Channel 0
DMAC1_CPC1	0x010c	W	0x0	Channel PC for DMA Channel 1
DMAC1_CPC2	0x0114	W	0x0	Channel PC for DMA Channel 2
DMAC1_CPC3	0x011c	W	0x0	Channel PC for DMA Channel 3
DMAC1_CPC4	0x0124	W	0x0	Channel PC for DMA Channel 4
DMAC1_CPC5	0x012c	W	0x0	Channel PC for DMA Channel 5
DMAC1_CPC6	0x0134	W	0x0	Channel PC for DMA Channel 6
DMAC1_SAR0	0x0400	W	0x0	Source Address for DMA Channel 0
DMAC1_SAR1	0x0420	W	0x0	Source Address for DMA Channel 1
DMAC1_SAR2	0x0440	W	0x0	Source Address for DMA Channel 2
DMAC1_SAR3	0x0460	W	0x0	Source Address for DMA Channel 3
DMAC1_SAR4	0x0480	W	0x0	Source Address for DMA Channel 4
DMAC1_SAR5	0x04a0	W	0x0	Source Address for DMA Channel 5
DMAC1_SAR6	0x04c0	W	0x0	Source Address for DMA Channel 6
DMAC1_DAR0	0x0404	W	0x0	Dest Address for DMAChannel 0
DMAC1_DAR1	0x0424	W	0x0	Dest Address for DMAChannel 1
DMAC1_DAR2	0x0444	W	0x0	Dest Address for DMAChannel 2
DMAC1_DAR3	0x0464	W	0x0	Dest Address for DMAChannel 3
DMAC1_DAR4	0x0484	W	0x0	Dest Address for DMAChannel 4
DMAC1_DAR5	0x04a4	W	0x0	Dest Address for DMAChannel 5
DMAC1_DAR6	0x04c4	W	0x0	Dest Address for DMAChannel 6
DMAC1_CCR0	0x0408	W	0x0	Channel Control for DMA Channel 0
DMAC1_CCR1	0x0428	W	0x0	Channel Control for DMA Channel 1
DMAC1_CCR2	0x0448	W	0x0	Channel Control for DMA Channel 2

DMAC1_CCR3	0x0468	W	0x0	Channel Control for DMA Channel 3
DMAC1_CCR4	0x0488	W	0x0	Channel Control for DMA Channel 4
DMAC1_CCR5	0x04a8	W	0x0	Channel Control for DMA Channel 5
DMAC1_CCR6	0x04c8	W	0x0	Channel Control for DMA Channel 6
DMAC1_LC0_0	0x040C	W	0x0	Loop Counter 0 for DMA Channel 0
DMAC1_LC0_1	0x042C	W	0x0	Loop Counter 0 for DMA Channel 1
DMAC1_LC0_2	0x044C	W	0x0	Loop Counter 0 for DMA Channel 2
DMAC1_LC0_3	0x046C	W	0x0	Loop Counter 0 for DMA Channel 3
DMAC1_LC0_4	0x048C	W	0x0	Loop Counter 0 for DMA Channel 4
DMAC1_LC0_5	0x04aC	W	0x0	Loop Counter 0 for DMA Channel 5
DMAC1_LC0_6	0x04cC	W	0x0	Loop Counter 0 for DMA Channel 6
DMAC1_LC1_0	0x0410	W	0x0	Loop Counter 1 for DMA Channel 0
DMAC1_LC1_1	0x0430	W	0x0	Loop Counter 1 for DMA Channel 1
DMAC1_LC1_2	0x0450	W	0x0	Loop Counter 1 for DMA Channel 2
DMAC1_LC1_3	0x0470	W	0x0	Loop Counter 1 for DMA Channel 3
DMAC1_LC1_4	0x0490	W	0x0	Loop Counter 1 for DMA Channel 4
DMAC1_LC1_5	0x04b0	W	0x0	Loop Counter 1 for DMA Channel 5
DMAC1_LC1_6	0x04d0	W	0x0	Loop Counter 1 for DMA Channel 6
-	-	-	-	reserved
DMAC1_DBGSTDMAC1_ATUS	0x0D00	W	0x0	Debug Status Register.
DMAC1_DBGCMD	0x0D04	W	0x0	Debug Command Register.
DMAC1_DBGINST0	0x0D08	W	0x0	Debug Instruction-0 Register.
DMAC1_DBGINST1	0x0D0C	W	0x0	Debug Instruction-1 Register.
DMAC1_CR0	0x0E00	W		Configuration Register 0.
DMAC1_CR1	0x0E04	W		Configuration Register 1.
DMAC1_CR2	0x0E08	W		Configuration Register 2.
DMAC1_CR3	0x0E0C	W		Configuration Register 3.
DMAC1_CR4	0x0E10	W		Configuration Register 4.
DMAC1_CRDn	0x0E14	W		Configuration Register Dn.
DMAC1_WD	0X0E80	W		Watchdog Register

Notes:
*Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

11.4.2 Detail Register Description

DMAC1_DSR
Address:Operational Base+0x0
DMA Manager Status Register

Bit	Attr	Reset Value	Description
31:10	-	-	Reserved
9	R	0x0	Provides the security status of the DMA manager thread: 0 = DMA manager operates in the Secure state 1 = DMA manager operates in the Non-secure state.
8:4	R	0x0	When the DMA manager thread executes a DMAWFE instruction, it waits for the following event to occur: b00000 = event[0] b00001 = event[1] b00010 = event[2] … b11111 = event[31].
3:0	R	0x0	The operating state of the DMA manager: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event b0101-b1110 = reserved b1111 = Faulting.

DMAC1_DPC
Address:Operational Base+0x4
DMA Program Counter Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA manager thread

DMAC1_INTEN
Address:Operational Base+0x20
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	Program the appropriate bit to control how the DMAC responds when it executes DMASEV: Bit [N] = 0 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC signals event N to all of the threads. Set bit [N] to 0 if your system design does not use irq[N] to signal an interrupt request. Bit [N] = 1 If the DMAC executes DMASEV for the event-interrupt resource N then the DMAC sets irq[N] HIGH. Set bit [N] to 1 if your system designer requires irq[N] to signal an interrupt

			request.

DMAC1_EVENT_RIS
Address:Operational Base+0x24
Event-Interrupt Raw Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Returns the status of the event-interrupt resources: Bit [N] = 0 Event N is inactive or irq[N] is LOW. Bit [N] = 1 Event N is active or irq[N] is HIGH.

DMAC1_INTMIS
Address:Operational Base+0x28
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the status of the interrupts that are active in the DMAC: Bit [N] = 0 Interrupt N is inactive and therefore irq[N] is LOW. Bit [N] = 1 Interrupt N is active and therefore irq[N] is HIGH

DMAC1_INTCLR
Address:Operational Base+0x2c
Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	Controls the clearing of the irq outputs: Bit [N] = 0 The status of irq[N] does not change. Bit [N] = 1 The DMAC sets irq[N] LOW if the INTEN Register programs the DMAC to signal an interrupt. Otherwise, the status of irq[N] does not change.

DMAC1_FSRD
Address:Operational Base+0x30
Fault Status DMA Manager Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the fault status of the DMA manager. Read as: 0 = the DMA manager thread is not in the Faulting state 1 = the DMA manager thread is in the Faulting state.

DMAC1_FSRC
Address:Operational Base+0x34
Fault Status DMA Channel Register

Bit	Attr	Reset Value	Description
31:0	R	0x0	Each bit provides the fault status of the corresponding channel. Read as: Bit [N] = 0 No fault is present on DMA channel N.

			Bit [N] = 1 DMA channel N is in the Faulting or Faulting completing state.

DMAC1_FTRD
Address:Operational Base+0x38
Fault Type DMA Manager Register

Bit	Attr	Reset Value	Description
31	-	-	reserved
30	R	0x0	If the DMA manager aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface.
29:17	-	-	reserved
16	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA manager performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response
15:6	-	-	reserved
5	R	0x0	Indicates if the DMA manager was attempting to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAWFE or DMASEV 1 = a DMA manager thread in the Non-secure state attempted to execute either: • DMAWFE to wait for a secure event • DMASEV to create a secure event or secure interrupt
4	R	0x0	ndicates if the DMA manager was attempting to execute DMAGO with inappropriate security permissions: 0 = DMA manager has appropriate security to execute DMAGO 1 = a DMA manager thread in the Non-secure state attempted to execute DMAGO to create a DMA channel operating in the Secure state.
3:2	-	-	reserved
1	R	0x0	Indicates if the DMA manager was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand.
0	R	0x0	Indicates if the DMA manager was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction.

DMAC1_FTR0~DMAC1_FTR6
Address:Operational Base+0x40

Operational Base+0x44
Operational Base+0x48
Operational Base+0x4c
Operational Base+0x50
Operational Base+0x54
Operational Base+0x58

Fault Type DMA Channel Register

Bit	Attr	Reset Value	Description
31	R	0x0	Indicates if the DMA channel has locked-up because of resource starvation: 0 = DMA channel has adequate resources 1 = DMA channel has locked-up because of insufficient resources. This fault is an imprecise abort
30	R	0x0	If the DMA channel aborts, this bit indicates if the erroneous instruction was read from the system memory or from the debug interface: 0 = instruction that generated an abort was read from system memory 1 = instruction that generated an abort was read from the debug interface. This fault is an imprecise abort but the bit is only valid when a precise abort occurs.
29:19	-	-	reserved
18	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs a data read: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort
17	R	0x0	Indicates the AXI response that the DMAC receives on the BRESP bus, after the DMA channel thread performs a data write: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is an imprecise abort.
16	R	0x0	Indicates the AXI response that the DMAC receives on the RRESP bus, after the DMA channel thread performs an instruction fetch: 0 = OKAY response 1 = EXOKAY, SLVERR, or DECERR response. This fault is a precise abort.
15:14	-	-	reserved
13	R	0x0	Indicates if the MFIFO did not contain the data to enable the DMAC to perform the DMAST: 0 = MFIFO contains all the data to enable the DMAST to complete 1 = previous DMALDs have not put enough data in the MFIFO to enable the DMAST to complete. This fault is a precise abort.
12	R	0x0	Indicates if the MFIFO prevented the DMA channel thread from executing DMALD or DMAST. Depending on the instruction: DMALD 0 = MFIFO contains sufficient space

			1 = MFIFO is too small to hold the data that DMALD requires. DMAST 0 = MFIFO contains sufficient data 1 = MFIFO is too small to store the data to enable DMAST to complete. This fault is an imprecise abort
11:8	-	-	reserved
7	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to program the CCRn Register to perform a secure read or secure write: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to perform a secure read or secure write. This fault is a precise abort
6	R	0x0	Indicates if a DMA channel thread, in the Non-secure state, attempts to execute DMAWFP, DMALDP, DMASTP, or DMAFLUSHP with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: • DMAWFP to wait for a secure peripheral • DMALDP or DMASTP to notify a secure peripheral • DMAFLUSHP to flush a secure peripheral. This fault is a precise abort.
5	R	0x0	Indicates if the DMA channel thread attempts to execute DMAWFE or DMASEV with inappropriate security permissions: 0 = a DMA channel thread in the Non-secure state is not violating the security permissions 1 = a DMA channel thread in the Non-secure state attempted to execute either: • DMAWFE to wait for a secure event • DMASEV to create a secure event or secure interrupt. This fault is a precise abort.
4:2	-	-	reserved
1	R	0x0	Indicates if the DMA channel thread was attempting to execute an instruction operand that was not valid for the configuration of the DMAC: 0 = valid operand 1 = invalid operand. This fault is a precise abort.
0	R	0x0	Indicates if the DMA channel thread was attempting to execute an undefined instruction: 0 = defined instruction 1 = undefined instruction. This fault is a precise abort

DMAC1_CSR0~DMAC1_CSR6

Address:Operational Base+0x100
Operational Base+0x108
Operational Base+0x110
Operational Base+0x118
Operational Base+0x120
Operational Base+0x128
Operational Base+0x130

Channel Status Registers

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21	R	0x0	The channel non-secure bit provides the security of the DMA channel: 0 = DMA channel operates in the Secure state 1 = DMA channel operates in the Non-secure state
20:16	-	-	reserved
15	R	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the periph operand was set: 0 = DMAWFP executed with the periph operand not set 1 = DMAWFP executed with the periph operand set
14	R	0x0	When the DMA channel thread executes DMAWFP this bit indicates if the burst or single operand were set: 0 = DMAWFP executed with the single operand set 1 = DMAWFP executed with the burst operand set.
13:9	-	-	reserved
8:4	R	0x0	If the DMA channel is in the Waiting for event state or the Waiting for peripheral state then these bits indicate the event or peripheral number that the channel is waiting for: b00000 = DMA channel is waiting for event, or peripheral, 0 b00001 = DMA channel is waiting for event, or peripheral, 1 b00010 = DMA channel is waiting for event, or peripheral, 2 . . . b11111 = DMA channel is waiting for event, or peripheral, 31
3:0	R	0x0	The channel status encoding is: b0000 = Stopped b0001 = Executing b0010 = Cache miss b0011 = Updating PC b0100 = Waiting for event

			b0101 = At barrier b0110 = reserved b0111 = Waiting for peripheral b1000 = Killing b1001 = Completing b1010-b1101 = reserved b1110 = Faulting completing b1111 = Faulting

DMAC1_CPC0~DMAC1_CPC6

Address:Operational Base+0x104
Operational Base+0x10c
Operational Base+0x114
Operational Base+0x11c
Operational Base+0x124
Operational Base+0x12c
Operational Base+0x134

Channel Program Counter Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Program counter for the DMA channel n thread

DMAC1_SAR0~DMAC1_SAR6

Address:Operational Base+0x400
Operational Base+0x420
Operational Base+0x440
Operational Base+0x460
Operational Base+0x480
Operational Base+0x4a0
Operational Base+0x4c0

Source Address Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the source data for DMA channel n

DMAC1_DAR0~DMAC1_DAR5

Address:Operational Base+0x404
Operational Base+0x424
Operational Base+0x444
Operational Base+0x464
Operational Base+0x484
Operational Base+0x4a4

DestinationAddress Registers

Bit	Attr	Reset Value	Description
31:0	R	0x0	Address of the Destinationdata for DMA channel n

DMAC1_CCR0~DMAC1_CCR6

Address:Operational Base+0x408
Operational Base+0x428
Operational Base+0x448
Operational Base+0x468
Operational Base+0x488
Operational Base+0x4a8
Operational Base+0x4c8

Channel Control Registers

Bit	Attr	Reset Value	Description
31:28	-	-	reserved
27:25	R	0x0	Programs the state of AWCACHE[3,1:0]a when the DMAC writes the destination data. Bit [27] 0 = AWCACHE[3] is LOW 1 = AWCACHE[3] is HIGH. Bit [26] 0 = AWCACHE[1] is LOW 1 = AWCACHE[1] is HIGH. Bit [25] 0 = AWCACHE[0] is LOW 1 = AWCACHE[0] is HIGH
24:22	R	0x0	Programs the state of AWPROT[2:0]a when the DMAC writes the destination data. Bit [24] 0 = AWPROT[2] is LOW 1 = AWPROT[2] is HIGH. Bit [23] 0 = AWPROT[1] is LOW 1 = AWPROT[1] is HIGH. Bit [22] 0 = AWPROT[0] is LOW 1 = AWPROT[0] is HIGH
21:18	R	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it writes the destination data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size
17:15	R	0x0	For each beat within a burst, it programs the number of bytes that the DMAC writes to the destination: b000 = writes 1 byte per beat b001 = writes 2 bytes per beat b010 = writes 4 bytes per beat b011 = writes 8 bytes per beat b100 = writes 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC writes out of the MFIFO when it executes a DMAST instruction is the product of dst_burst_len and dst_burst_size.
14	R	0x0	Programs the burst type that the DMAC performs when it writes the destination data: 0 = Fixed-address burst. The DMAC signals AWBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals AWBURST[0] HIGH.
13:11	R	0x0	Set the bits to control the state of ARCACHE[2:0]a when the DMAC reads the source data.

			Bit [13] 0 = ARCACHE[2] is LOW 1 = ARCACHE[2] is HIGH. Bit [12] 0 = ARCACHE[1] is LOW 1 = ARCACHE[1] is HIGH. Bit [11] 0 = ARCACHE[0] is LOW 1 = ARCACHE[0] is HIGH.
10:8	R	0x0	Programs the state of ARPROT[2:0]a when the DMAC reads the source data. Bit [10] 0 = ARPROT[2] is LOW 1 = ARPROT[2] is HIGH. Bit [9] 0 = ARPROT[1] is LOW 1 = ARPROT[1] is HIGH. Bit [8] 0 = ARPROT[0] is LOW 1 = ARPROT[0] is HIGH.
7:4	R	0x0	For each burst, these bits program the number of data transfers that the DMAC performs when it reads the source data: b0000 = 1 data transfer b0001 = 2 data transfers b0010 = 3 data transfers . . . b1111 = 16 data transfers. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size
3:1	R	0x0	For each beat within a burst, it programs the number of bytes that the DMAC reads from the source: b000 = reads 1 byte per beat b001 = reads 2 bytes per beat b010 = reads 4 bytes per beat b011 = reads 8 bytes per beat b100 = reads 16 bytes per beat b101-b111 = reserved. The total number of bytes that the DMAC reads into the MFIFO when it executes a DMALD instruction is the product of src_burst_len and src_burst_size
0	R	0x0	Programs the burst type that the DMAC performs when it reads the source data: 0 = Fixed-address burst. The DMAC signals ARBURST[0] LOW. 1 = Incrementing-address burst. The DMAC signals ARBURST[0] HIGH

DMAC1_LC0_0~DMAC1_LC0_6

Address:Operational Base+0x40c
Operational Base+0x42c
Operational Base+0x44c
Operational Base+0x46c
Operational Base+0x48c

Operational Base+0x4ac
Operational Base+0x4cc

Loop Counter 0 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 0 iterations

DMAC1_LC1_0~DMAC1_LC1_6

Address:Operational Base+0x410
Operational Base+0x430
Operational Base+0x450
Operational Base+0x470
Operational Base+0x490
Operational Base+0x4b0
Operational Base+0x4e0

Loop Counter 1 Registers

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	R	0x0	Loop counter 1 iterations

DMAC1_DBGSTATUS

Address:Operational Base+0xd00
Debug Status Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved
1:0	R	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved.

DMAC1_DBGCMD

Address:Operational Base+0xd04
Debug Command Register

Bit	Attr	Reset Value	Description
31:2	-	-	reserved
1:0	W	0x0	The debug encoding is as follows: b00 = execute the instruction that the DBGINST [1:0] Registers contain b01 = reserved b10 = reserved b11 = reserved

DMAC1_DBGINST0

Address:Operational Base+0xd08
Debug Instruction-0 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 1
23:16	W	0x0	Instruction byte 0
17:11	-	-	reserved

10:8	W	0x0	DMA channel number: b000 = DMA channel 0 b001 = DMA channel 1 b010 = DMA channel 2 … b111 = DMA channel 7
7:1	-	-	reserved
0	W	0x0	The debug thread encoding is as follows: 0 = DMA manager thread 1 = DMA channel.

DMAC1_DBGINST1

Address:Operational Base+0xd0c

Debug Instruction-1 Register

Bit	Attr	Reset Value	Description
31:24	W	0x0	Instruction byte 5
23:16	W	0x0	Instruction byte 4
15:8	W	0x0	Instruction byte 3
7:0	W	0x0	Instruction byte 2

DMAC1_CR0

Address:Operational Base+0xe00

Configuration Register 0

Bit	Attr	Reset Value	Description
31:22	-	-	reserved
21:17	R	0x2	Number of interrupt outputs that the DMAC provides: b00000 = 1 interrupt output, irq[0] b00001 = 2 interrupt outputs, irq[1:0] b00010 = 3 interrupt outputs, irq[2:0] . . . b11111 = 32 interrupt outputs, irq[31:0].
16:12	R	0x7	Number of peripheral request interfaces that the DMAC provides: b00000 = 1 peripheral request interface b00001 = 2 peripheral request interfaces b00010 = 3 peripheral request interfaces . . . b11111 = 32 peripheral request interfaces.
11:7	-	-	reserved
6:4	R	0x5	Number of DMA channels that the DMAC supports: b000 = 1 DMA channel b001 = 2 DMA channels b010 = 3 DMA channels . . . b111 = 8 DMA channels.

3	-	-	reserved
2	R	0x0	Indicates the status of the boot_manager_ns signal when the DMAC exited from reset: 0 = boot_manager_ns was LOW 1 = boot_manager_ns was HIGH.
1	R	0x0	Indicates the status of the boot_from_pc signal when the DMAC exited from reset: 0 = boot_from_pc was LOW 1 = boot_from_pc was HIGH
0	R	0x1	Supports peripheral requests: 0 = the DMAC does not provide a peripheral request interface 1 = the DMAC provides the number of peripheral request interfaces that the num_periph_req field specifies.

DMAC1_CR1

Address:Operational Base+0xe04

Configuration Register 1

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:4	R	0x5	[7:4] num_i-cache_lines Number of i-cache lines: b0000 = 1 i-cache line b0001 = 2 i-cache lines b0010 = 3 i-cache lines … b1111 = 16 i-cache lines.
3	-	-	reserved
2:0	R	0x7	The length of an i-cache line: b000-b001 = reserved b010 = 4 bytes b011 = 8 bytes b100 = 16 bytes b101 = 32 bytes b110-b111 = reserved

DMAC1_CR2

Address:Operational Base+0xe08

Configuration Register 2

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the value of boot_addr[31:0] when the DMAC exited from reset

DMAC1_CR3

Address:Operational Base+0xe0c

Configuration Register 3

Bit	Attr	Reset Value	Description
31:0	R	0x0	Provides the security state of an event-interrupt resource: Bit [N] = 0 Assigns event<N> or irq[N] to the Secure state. Bit [N] = 1 Assigns event<N> or irq[N] to the Non-secure state.

DMAC1_CR4
Address:Operational Base+0xe10
Configuration Register 4

Bit	Attr	Reset Value	Description
31:0	R	0x6	Provides the security state of the peripheral request interfaces: Bit [N] = 0 Assigns peripheral request interface N to the Secure state. Bit [N] = 1 Assigns peripheral request interface N to the Non-secure state

DMAC1_CRDn
Address:Operational Base+0xe14
DMA Configuration Register

Bit	Attr	Reset Value	Description
31:30	-	-	reserved
29:20	R	0x20	The number of lines that the data buffer contains: b000000000 = 1 line b000000001 = 2 lines … b111111111 = 1024 lines
19:16	R	0x9	The depth of the read queue: b0000 = 1 line b0001 = 2 lines . . . b1111 = 16 lines.
15	-	-	reserved
14:12	R	0x4	Read issuing capability that programs the number of outstanding read transactions: b000 = 1 b001 = 2 … b111 = 8
11:8	R	0x7	The depth of the write queue: b0000 = 1 line b0001 = 2 lines … b1111 = 16 lines.
7	-	-	reserved
6:4	R	0x3	Write issuing capability that programs the number of outstanding write transactions: b000 = 1 b001 = 2 … b111 = 8
3	-	-	reserved
2:0		0x3	The data bus width of the AXI interface: b000 = reserved b001 = reserved b010 = 32-bit

			b011 = 64-bit b100 = 128-bit b101-b111 = reserved.

DMAC1_WD
Address:Operational Base+0xe80
DMA Watchdog Register

Bit	Attr	Reset Value	Description
31:1	-	-	reserved
0	RW	0x0	Controls how the DMAC responds when it detects a lock-up condition: 0 = the DMAC aborts all of the contributing DMA channels and sets irq_abort HIGH 1 = the DMAC sets irq_abort HIGH.

11.5 Timing Diagram

Please refer to chapter 10.5 for the similar description.

11.6 Interface Description

DMAC1 has the following tie-off signals.It can be configured by GRF register.(Please refer to the chapter to find how to configure)

DMAC1

interface	Reset value	Control source
boot_addr	0x0	GRF
boot_from_pc	0x0	GRF
boot_manager_ns	0x0	GRF
boot_irq_ns	0xf	GRF
boot_periph_ns	0xfffff	GRF

boot_addr
Configures the address location that contains the first instruction the DMAC executes, when it exits from reset.

boot_from_pc
Controls the location in which the DMAC executes its initial instruction, after it exits from reset:
0 = DMAC waits for an instruction from either APB interface
1 = DMA manager thread executes the instruction that is located at the address that

boot_manager_ns
When the DMAC exits from reset, this signal controls the security state of the DMA manager thread:
0 = assigns DMA manager to the Secure state
1 = assigns DMA manager to the Non-secure state.

boot_irq_ns
Controls the security state of an event-interrupt resource, when the DMAC exits from reset:
boot_irq_ns[x] is LOW

The DMAC assigns event<x> or irq[x] to the Secure state.

boot_irq_ns[x] is HIGH

The DMAC assigns event<x> or irq[x] to the Non-secure state.

boot_periph_ns

Controls the security state of a peripheral request interface, when the DMAC exits from reset:

boot_periph_ns[x] is LOW

The DMAC assigns peripheral request interface x to the Secure state.

boot_periph_ns[x] is HIGH

The DMAC assigns peripheral request interface x to the Non-secure state.

11.7 Application Notes

Please refer to chapter 10.3 for the similar description.

Chapter 12 GIC(General Interrupt Controller)

12.1 Overview

The interrupt controller(GIC) in This device has two interfaces, the distributor interface connects to the interrupt source,the cpu interface connects to Cortex-A8. The GIC supports Security Extensions.

It supports the following features:

- Supports 72 vectored IRQ interrupts
- Supports 64 interrupts priority levels
- Programmable interrupt priority level masking
- Generates IRQ and FIQ
- Generates Software interrupt
- Supports Security Extensions

12.2 Block Diagram

Fig.12-1 shows the block diagram of gic

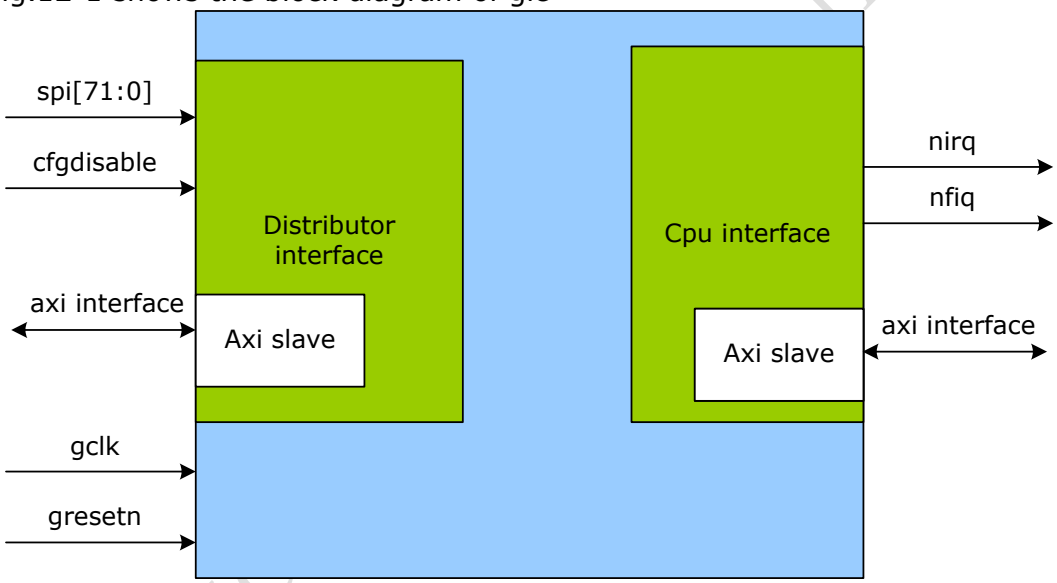


Fig. 12-1Block diagram of GIC

The diagram shows that GIC has two axi interfaces independently and has two base address for these two interfaces.

These two interfaces connect to CPU L1 AXI interconnect, and the connected ports are secure after reset. So, after reset GIC can only be accessed by secure transaction. These two ports in CPU L1 AXI interconnect can be configured to non-secure , Please refer to chapter 9.3 for detailed information.

12.3 Function Description

This GIC architecture splits logically into a Distributor block and one CPU interface block, as Figure 12-1 shows.

Distributor

This performs interrupt prioritization and distribution to the CPU interface that connect to the processor in the system.

CPU interface

CPU interface performs priority masking and preemption handling for a connected processor in the system.

12.3.1 The Distributor

The Distributor centralizes all interrupt sources, determines the priority of each interrupt, and for CPU interface dispatches the interrupt with the highest priority to the interface for priority masking and preemption handling.

The Distributor provides a programming interface for:

- Globally enabling the forwarding of interrupts to the CPU interface
- Enabling or disabling each interrupt
- Setting the priority level of each interrupt
- Setting the target processor list of each interrupt
- Setting each peripheral interrupt to be level-sensitive or edge-triggered
- If the GIC implements the Security Extensions, setting each interrupt as either
- Secure or Non-secure
- Sending an SGI to processor.
- Visibility of the state of each interrupt
- A mechanism for software to set or clear the pending state of a peripheral
- interrupt.

Interrupt ID

Interrupts from sources are identified using ID numbers. CPU interface can see up to 88 interrupts.

The GIC assigns interrupt these 88 ID numbers as follows:

- Interrupt numbers ID32-ID1 are used for SPIs(shared peripheral interrupts).
- ID0-ID15 are used for SGIs(software generated interrupts).
- ID16-ID31 are not used

The GIC architecture reserves interrupt ID numbers 1022-1023 for special purposes.

ID1022

The GIC returns this value to a processor in response to an interrupt acknowledge only whenall of the following apply:

- The interrupt acknowledge is a Secure read
- The highest priority pending interrupt is Non-secure
- The AckCtl bit in the Secure ICCICR is set to 0
- The priority of the interrupt is sufficient for it to be signalled to the processor.

Interrupt ID 1022 informs Secure software that there is a Non-secure interrupt of sufficient priority to be signalled to the processor, that must be handled by Non-secure software. In this situation the Secure software might alter its schedule to permit Non-secure software to handle the interrupt, to minimize the interrupt latency.

ID1023

This value is returned to a processor, in response to an interrupt acknowledge, if there is no pending interrupt with sufficient priority for it to be signalled to the processor.

On a processor that implements the Security Extensions, Secure software

treats values of 1022 and 1023 as spurious interrupts.

12.3.2 CPU interface

CPU interface block provides the interface for a processor that operates with the GIC. CPU interface provides a programming interface for:

- Enabling the signalling of interrupt requests by the CPU interface
- Acknowledging an interrupt
- Indicating completion of the processing of an interrupt
- Setting an interrupt priority mask for the processor
- Defining the preemption policy for the processor
- Determining the highest priority pending interrupt for the processor.

When enabled, CPU interface takes the highest priority pending interrupt for its connected processor and determines whether the interrupt has sufficient priority for it to signal the interrupt request to the processor.

To determine whether to signal the interrupt request to the processor the CPU interface considers the interrupt priority mask and the preemption settings for the processor. At any time, the connected processor can read the priority of its highest priority active interrupt from a CPU interface register.

The processor acknowledges the interrupt request by reading the CPU interface Interrupt Acknowledge register. The CPU interface returns one of:

The ID number of the highest priority pending interrupt, if that interrupt is of sufficient priority to generate an interrupt exception on the processor. This is the normal response to an interrupt acknowledge.

Exceptionally, an ID number that indicates a spurious interrupt.

When the processor acknowledges the interrupt at the CPU interface, the Distributor changes the status of the interrupt from pending to either active, or active and pending. At this point the CPU interface can signal another interrupt to the processor, to preempt interrupts that are active on the processor. If there is no pending interrupt with sufficient priority for signalling to the processor, the interface deasserts the interrupt request signal to the processor.

When the interrupt handler on the processor has completed the processing of an interrupt, it writes to the CPU interface to indicate interrupt completion. When this happens, the distributor changes the status of the interrupt either:

- from active to inactive
- from active and pending to pending.

12.3.3 Interrupt handling state machine

The distributor maintains a state machine for each supported interrupt on CPU interface. Following figure shows an instance of this state machine, and the possible state transitions.

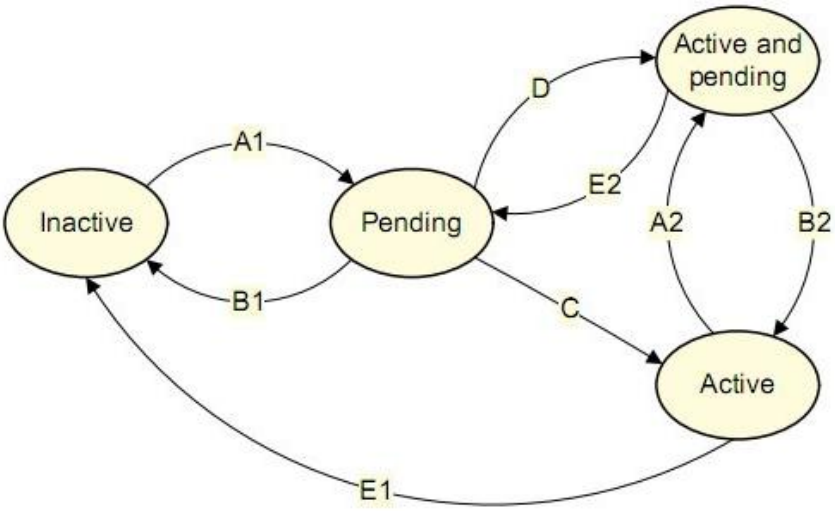


Fig. 12-2 GIC Interrupt handling state machine

Transition A1 or A2, add pending status
For an SGI:
• Occurs on a write to an ICDSGIR that specifies the processor as a target.
• If the GIC implements the Security Extensions and the write to the ICDSGIR is Secure, the transition occurs only if the security configuration of the specified SGI, for the CPU interface, corresponds to the ICDSGIR.SATT bit value.

For an SPI, occurs if either:
• a peripheral asserts an interrupt signal
• software writes to an ICDISPR.

Transition B1 or B2, remove pending status
Not applicable to SGIs:
• a pending SGI must transition through the active state, or reset, to remove its pending status.
• an active and pending SGI must transition through the pending state, or reset, to remove its pending status.

For an SPI, occurs if either:
• the level-sensitive interrupt is pending only because of the assertion of an input signal, and that signal is deasserted
• the interrupt is pending only because of the assertion of an edge-triggered interrupt signal, or a write to an ICDISPR, and software writes to the corresponding ICDICPR.

Transition C
If the interrupt is enabled and of sufficient priority to be signalled to the processor, occurs when software reads from the ICCIAR.

Transition D
For an SGI, occurs if the associated SGI is enabled and the Distributor forwards it to the CPU interface at the same time that the processor reads the ICCIAR to acknowledge a previous instance of the SGI. Whether this transition occurs depends on the timing of the read of the ICCIAR relative to the reforwarding of the SGI.

For an SPI:

• Occurs if all the following apply:
— The interrupt is enabled.
— Software reads from the ICCIAR. This read adds the active state to the interrupt.
— For a level-sensitive interrupt, the interrupt signal remains asserted. This is usually the case, because the peripheral does not deassert the interrupt until the processor has serviced the interrupt.

• For an edge-triggered interrupt, whether this transition occurs depends on the timing of the read of the ICCIAR relative to the detection of the reassertion of the interrupt. Otherwise the read of the ICCIAR causes transition C, possibly followed by transition A2.

Transition E1 or E2, remove active status
Occurs when software writes to the ICCEOIR.

12.4 Register Description

12.4.1 GIC Distributor interface register summary

Name	Offset	Size	Reset	Description
GICD_ICDDCR	0x000	W	0x0	Distributor Control Register
GICD_ICDICTR	0x004	W		Interrupt Controller Type Register
GICD_ICDIIDR	0x008	W		Distributor Implementer Identification Register
GICD_ICDISR	0x080	W		Interrupt Security Registers
-	-	-	-	reserved
GICD_ICDISER	0x100-0x17C	W		Interrupt Set-Enable Registers
GICD_ICDICER	0x180-0x1FC	W		Interrupt Clear-Enable Registers
GICD_ICDISPR	0x200-0x27C	W	0x0	Interrupt Set-Pending Registers
GICD_ICDICPR	0x280-0x2FC	W	0x0	Interrupt Clear-Pending Registers
GICD_ICDABR	0x300-0x37C	W	0x0	Active Bit Registers
-	-	-	-	reserved
GICD_ICDIPR	0x400-0x7F8	B	0x0	Interrupt Priority Registers
-	-	-	-	reserved
GICD_ICDIPTR	0x800-0x81C	B		Interrupt Processor Targets
-	-	-	-	reserved
GICD_ICDICFR	0xC00-0xCFC	W		Interrupt Configuration Registers
-	-	-	-	reserved
GICD_ICDSGIR	0xF00	W		Software Generated Interrupt Register

Notes:
*Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

12.4.2 GIC Distributor interface detail register description

GICD_ICDDCR
Address:Operational Base+0x0

Distributor Control Register

Bit	Attr	Reset Value	Description
31:1	-	-	reserved
0	RW	0x0	Global enable for monitoring peripheral interrupt signals and forwarding pending interrupts to the CPU interface. 0 The GIC ignores all peripheral interrupt signals, and does not forward pending interrupts to the Cpu interface. 1 The GIC monitors the peripheral interrupt signals, and forwards pending interrupts to the Cpu interface.

GICD_ICDICTR

Address:Operational Base+0x4

Interrupt Controller Type Register

Bit	Attr	Reset Value	Description
31:11	-	-	reserved
10	R	0x1	Indicates whether the GIC implements the Security Extensions. 0 Security Extensions not implemented. 1 Security Extensions implemented
9:8	-	-	reserved
7:5	R	0x0	Indicates the number of implemented Cpu interface. The number of implemented Cpu interface is one more than the value of this field, for example if this field is 0b011, there are four Cpu interface. In this product ,only one cpu interface is implemented.
4:0	R	0x2	Indicates the maximum number of interrupts that the GIC supportsa. If the value of this field is N, the maximum number of interrupts is 32(N+1). The interrupt ID range is from 0 to one less than the number of IDs. For example: 0b00011 Up to 128 interrupt lines, interrupt IDs 0-127. The maximum number of interrupts is 1020 (0b11111).

GICD_ICDIIDR

Address:Operational Base+0x8

Distributor Implementer Identification Register

Bit	Attr	Reset Value	Description
31:24	R	0x0	product identifier.
23:20	-	-	reserved
19:16	R	0x0	variant number. Typically, this field is used to distinguish product variants, or major revisions of a product
15:12	R	0x0	revision number. Typically, this field is used to distinguish minor revisions of a product
11:0	R	0x0	Contains the JEP106 code of the company that implemented the GIC Distributor:a

			Bits [11:8] The JEP106 continuation code of the implementer. Bits [7] Always 0. Bits [6:0] The JEP106 identity code of the implementer.

GICD_ICDISR
Address:Operational Base+0x80
Interrupt Security Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: 0 The corresponding interrupt is Secure. 1 The corresponding interrupt is Non-secure.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDISR number, M, is given by M = N DIV 32
• the offset of the required ICDISR is (0x080 + (4*M))
• the bit number of the required Security status bit in this register is N MOD 32.

GICD_ICDISER
Address:Operational Base+0x100
Interrupt Set-Enable Registers

Bit	Attr	Reset Value	Description
31:0	RW		For SPIs, for each bit: Reads 0 The corresponding interrupt is disabled. 1 The corresponding interrupt is enabled. Writes 0 Has no effect. 1 Enables the corresponding interrupt. A subsequent read of this bit returns the value 1.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDISER number, M, is given by M = N DIV 32
• the offset of the required ICDISER is (0x100 + (4*M))
• the bit number of the required Set-enable bit in this register is N MOD 32.

GICD_ICDICER
Address:Operational Base+0x180
Interrupt Clear-Enable Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For SPI, for each bit: Reads 0 The corresponding interrupt is disabled. 1 The corresponding interrupt is enabled. Writes 0 Has no effect. 1 Disables the corresponding interrupt. A subsequent read of this bit returns the value 0.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDICER number, M, is given by M = N DIV 32
• the offset of the required ICDICER is (0x180 + (4*M))
• the bit number of the required Clear-enable bit in this register is N MOD 32.

GICD_ICDISPR
Address:Operational Base+0x200
Interrupt Set-Pending Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: Reads 0 The corresponding interrupt is not pending on any processor. 1 • For SGIs, the corresponding interrupt is pending on this processor. • For SPIs, the corresponding interrupt is pending on at least one processor. Writes For SPIs: 0 Has no effect. 1 The effect depends on whether the interrupt is edge-triggered or level-sensitive: Edge-triggered Changes the status of the corresponding interrupt to: • pending if it was previously inactive • active and pending if it was previously active.Has no effect if the interrupt is already pending. Level sensitive If the corresponding interrupt is not pendinga, changes the status of the corresponding interrupt to: • pending if it was previously inactive • active and pending if it was previously active. If the interrupt is already pending: • because of a write to the ICDISPR, the write has no effect • because the corresponding interrupt signal is asserted, the write has no effect on the status of the interrupt, but the interrupt remains pendinga if the interrupt signal is deasserted.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDISPR number, M, is given by M = N DIV 32
• the offset of the required ICDISPR is (0x200 + (4*M))
• the bit number of the required Set-pending bit in this register is N MOD 32.

GICD_ICDICPR
Address:Operational Base+0x280
Interrupt Clear-Pending Registers

Bit	Attr	Reset Value	Description
31:0	RW	0x0	For each bit: Reads

			0 The corresponding interrupt is not pending on any processor 1 • For SGIs, the corresponding interrupt is pendinga on this processor. • For SPIs, the corresponding interrupt is pendinga on aleast one processor. Writes For SPIs: 0 Has no effect. 1 The effect depends on whether the interrupt is edge-triggeredor level-sensitive: Edge-triggered Changes the status of the corresponding interrupto: • inactive if it was previously pending • active if it was previously active and pending.Has no effect if the interrupt is not pending. Level-sensitive If the corresponding interrupt is pendinga only because of a write to the ICDISPR, the write changes the status of the interrupt to: • inactive if it was previously pending • active if it was previously active and pending.Otherwise the interrupt remains pending if the interrupt signal remains asserted.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDICPR number, M, is given by M = N DIV 32
• the offset of the required ICDICPR is (0x280 + (4*M))
• the bit number of the required Set-pending bit in this register is N MOD 32.

GICD_ICDABR
Address:Operational Base+0x300
Active Bit Registers

Bit	Attr	Reset Value	Description
31:0	R		For each bit: 0 Corresponding interrupt is not active. 1 Corresponding interrupt is active.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDABR number, M, is given by M = N DIV 32
• the offset of the required ICDABR is (0x300 + (4*M))
• the bit number of the required Active bit in this register is N MOD 32.

GICD_ICDIPR
Address:Operational Base+0x400
Interrupt Priority Registers

Bit	Attr	Reset Value	Description
7:0	RW	0x0	The lower the value, the greater the priority of the corresponding interrupt.

For interrupt ID N:
• the corresponding ICDIPR number, M, is given by M = N
• the offset of the required ICDIPR is (0x400 + M)

GICD_ICDIPTR
Address:Operational Base+0x800
Interrupt Processor Targets Registers

Bit	Attr	Reset Value	Description
7:0	RW	0x1	This register is not used. As in our product ,there is only one processor.

GICD_ICDICFR
Address:Operational Base+0xc00
Interrupt Configuration Registers

Bit	Attr	Reset Value	Description
2F+1	RW	0x0	F=0,1,2,3….15 The encoding is: 0 Corresponding interrupt is level-sensitive. 1 Corresponding interupt is edge-triggered.

For interrupt ID N, when DIV and MOD are the integer division and modulo operations:
• the corresponding ICDICFR number, M, is given by M = N DIV 16
• the offset of the required ICDIPTR is (0xC00 + (4*M))
• the required Priority field in this register, F, is given by F = N MOD 16, where field 0 refers to register bits [1:0], field 1 refers to bits [3:2], and so on, up to field 15 refers to bits [31:30]

GICD_ICDSGIR
Address:Operational Base+0xf00
Software Generated Interrupt Register

Bit	Attr	Reset Value	Description
31:26	-	-	reserved
25:24	W	0x0	0b00 Send the interrupt to the Cpu interface specified in the CPUTargetList fielda. 0b01 Send the interrupt to all Cpu interface except the CPU interface that requested the interrupt. 0b10 Send the interrupt only to the CPU interface that requested the interrupt. 0b11 Reserved
23:16	W	0x0	When TargetList Filter = 0b00, defines the Cpu interface the Distributor must send the interrupt to. Each bit of CPUTargetList[7:0] refers to the corresponding CPU interface, for example CPUTargetList[0] corresponds to CPU interface 0. Setting a bit to 1 sends the interrupt to the corresponding interface.
15	W	0x0	If the GIC implements the Security Extensions, this field is writable only using a Secure access.

			Any Non-secure write to the ICDSGIR issues an SGI only if the specified SGI is programmed as Non-secure, regardless of the value of bit [15] of the write. Specifies the required security value of the SGI: 0 Send the SGI specified in the SGIINTID field to a specified CPU interface only if the SGI is configured as Secure on that interface. 1 Send the SGI specified in the SGIINTID field to a specified Cpu interface only if the SGI is configured as Non-secure on that interface
14:4	-	-	reserved
3:0	W	0x0	The Interrupt ID of the SGI to send to the specified Cpu interface. The value of this field is the Interrupt ID, in the range 0-15, for example a value of 0b0011 specifies Interrupt ID 3

12.4.3 GIC CPU interface register summary

Name	Offset	Size	Reset Value	Description
GICC_ICCICR	0x00	W	0x0	CPU Interface Control Register
GICC_ICCPMR	0x04	W	0x0	Interrupt Priority Mask Register
GICC_ICCBPR	0x08	W	0x0	Binary Point Register
GICC_ICCIAR	0x0C	W	0x3ff	Interrupt Acknowledge Register
GICC_ICCEOIR	0x10	W	-	End of Interrupt Register
GICC_ICCRPR	0x14	W	0xff	Running Priority Register
GICC_ICCHPIR	0x18	W	0x3ff	Highest Pending Interrupt Register
GICC_ICCABPR	0x1C	W	0x0	Aliased Binary Point Register
GICC_ICCIIDR	0xFC	W	0x0	CPU Interface Identification Register

Notes:
*Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

12.4.4 GIC CPU interface detail register description

GICC_ICCICR
Address:Operational Base+0x0
CPU Interface Control Register

Bit	Attr	Reset Value	Description
31:5	-	-	reserved
4	RW	0x0	Controls whether the CPU interface uses the Secure or Non-secure Binary Point Register for preemption. 0 To determine any preemption, use: • the Secure Binary Point Register for Secure

			interrupts • the Non-secure Binary Point Register for Non-secure interrupts. 1 To determine any preemption use the Secure Binary Point Register for both Secure and Non-secure interrupts.
3	RW	0x0	Controls whether the GIC signals Secure interrupts to a target processor using the FIQ or the IRQ signal. 0 Signal Secure interrupts using the IRQ signal. 1 Signal Secure interrupts using the FIQ signal. The GIC always signals Non-secure interrupts using the IRQ signal.
2	RW	0x0	Controls whether a Secure read of the ICCIAR, when the highest priority pending interrupt is Non-secure, causes the CPU interface to acknowledge the interrupt. 0 If the highest priority pending interrupt is Non-secure, a Secure read of the ICCIAR returns an Interrupt ID of 1022. The read does not acknowledge the interrupt, and the pending status of the interrupt is unchanged. 1 If the highest priority pending interrupt is Non-secure, a Secure read of the ICCIAR returns the Interrupt ID of the Non-secure interrupt. The read acknowledges the interrupt, and the status of the interrupt becomes active, or active and pending.
1	RW	0x0	An alias of the Enable bit in the Non-secure ICCICR. This alias bit means Secure software can enable the signalling of Non-secure interrupts. 0 Disable signalling of Non-secure interrupts. 1 Enable signalling of Non-secure interrupts
0	RW	0x0	Global enable for the signalling of Secure interrupts by the Cpu interface to the connected processors. 0 Disable signalling of Secure interrupts. 1 Enable signalling of Secure interrupts

GICC_ICCPMR

Address:Operational Base+0x4

Interrupt Priority Mask Register

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	RW	0x0	The priority mask level for the CPU interface. If the priority of an interrupt is higher than the value indicated by this field, the interface signals

			the interrupt to the processor. If the GIC supports fewer than 256 priority levels then some bits are RAZ/WI, as follows: 128 supported levels Bit [0] = 0. 64 supported levels Bit [1:0] = 0b00. 32 supported levels Bit [2:0] = 0b000. 16 supported levels Bit [3:0] = 0b0000

GICC_ICCBPR

Address:Operational Base+0x8

Binary Point Register

Bit	Attr	Reset Value	Description
31:3	-	-	reserved
2:0	RW	0x0	The value of this field controls how the 8-bit interrupt priority field is split into a group priority field, used to determine interrupt preemption, and a subpriority field.

GICC_ICCIAR

Address:Operational Base+0xc

Interrupt Acknowledge Register

Bit	Attr	Reset Value	Description
31:13	-	-	reserved
12:10	RO	0x0	For SGIs in a multiprocessor implementation, this field identifies the processor that requested the interrupt. It returns the number of the CPU interface that made the request, for example a value of 3 (0b011) means the request was generated by a write to the IDCSFGIR on CPU interface 3. For all other interrupts this field is RAZ.
9:0	RO	0x0	The interrupt ID.

GICC_ICCEOIR

Address:Operational Base+0x10

End of Interrupt Register

Bit	Attr	Reset Value	Description
31:13	-	-	reserved
12:10	WO	0x0	On a multiprocessor implementation, on completion of the processing of an SGI, this field contains the CPUID value from the corresponding ICCIAR access.
9:0	WO	0x0	The ACKINTID value from the corresponding ICCIAR access.

GICC_ICCRPR

Address:Operational Base+0x14

Running Priority Register

Bit	Attr	Reset Value	Description
31:8	-	-	reserved
7:0	RO	0x0	The priority value of the highest priority interrupt that is active on the CPU interface.

GICC_ICCABPR
Address:Operational Base+0x18
Aliased Binary Point Register

Bit	Attr	Reset Value	Description
31:3	-	-	reserved
2:0	RW	0x0	Provides an alias of the Non-secure ICCBPR.

GICC_ICCHPIR
Address:Operational Base+0x1c
Highest Pending Interrupt Register

Bit	Attr	Reset Value	Description
31:10	-	-	reserved
9:0	R	0x0	The interrupt ID of the highest priority pending interrupt.

GICC_ICCIIDR
Address:Operational Base+0xfc
CPU Interface Identification Register

Bit	Attr	Reset Value	Description
31:20	R	0x0	An IMPLEMENTATION DEFINED product identifier.
19:16	R	0x0	For an implementation that complies with this specification, the value is 0x1
15:12	R	0x0	An IMPLEMENTATION DEFINED revision number for the CPU interface.
11:0	R	0x0	Contains the JEP106 code of the company that implemented the GIC CPU interface:b Bits [11:8] The JEP106 continuation code of the implementer. Bit [7] Always 0. Bits [6:0] The JEP106 identity code of the implementer.

12.5 Interface Description

Both distributor interface and cpu interface are secure accessed only after reset,Register inside the CPU L1 AXI interconnect needs to be configured to change to non-secure access.

When the tie_off signal cfgsisable is HIGH, it enhances the security of the GIC by preventing write accesses to security-critical configuration registers.This signal is low after reset, it can be configured through TZPC registers.

12.6 Application Notes

12.6.1 General handling of interrupts

The GIC operates on interrupts as follows:

1. The GIC determines whether each interrupt is enabled. An interrupt that is not enabled has no further effect on the GIC.(Enables an interrupt by writing to the appropriate ICDISER bit, disables an interrupt by writing to the appropriate ICDICER bit)

2. For each enabled interrupt that is pending, the Distributor determines the targeted processor.

3. For processor, the Distributor determines the highest priority pending interrupt, based on the priority information it holds for each interrupt, and forwards the interrupt to the CPU interface.

4. The CPU interface compares the interrupt priority with the current interrupt priority for the processor, determined by a combination of the Priority Mask Register, the current preemption settings, and the highest priority active interrupt for the processor. If the interrupt has sufficient priority, the GIC signals an interrupt exception request to the processor.

5. When the processor takes the interrupt exception, it reads the ICCIAR in its CPU interface to acknowledge the interrupt. This readreturns an Interrupt ID that the processor uses to select the correct interrupt handler. When it recognizes this read, the GIC changes the state of the interrupt:
• if the pending state of the interrupt persists when the interrupt becomes active, or if the interrupt is generated again, from pending to active and pending.
• otherwise, from pending to active

6.When the processor has completed handling the interrupt, it signals this completion by writing to the ICCEOIR in the GIC

Generating an SGI
A processor generates an SGI by writing to an ICDSGIR.

12.6.2 Interrupt prioritization

Software configures interrupt prioritization in the GIC by assigning a priority value to each interrupt source. Priority values are 8-bit unsigned binary.
In this product, GIC implements 64 priority levels. So only the highest 6 bits are valid, the lower 2 bits read as zero.
In the GIC prioritization scheme, lower numbers have higher priority, that is, the lower the assigned priority value the higher the priority of the interrupt. The highest interrupt priority always has priority field value 0.

The ICDIPRs hold the priority value for each supported interrupt. To determine the number of priority bits implemented write 0xFF to an ICDIPR priority field and read back the value stored.

Preemption
A CPU interface supports forwarding of higher priority pending interrupts to a target processor before an active interrupt completes. A pending interrupt is only forwarded if it has a higher priority than all of:
• the priority of the highest priority active interrupt on the target processor, the running priority for the processor, see Running Priority Register (ICCRPR) .
• the priority mask, see Priority masking.
• the priority group, see Priority grouping.

Preemption occurs at the time when the processor acknowledges the new interrupt, and starts to service it in preference to the previously active interrupt or the currently running process. When this occurs, the initial active interrupt is said to have been preempted. Starting to service an interrupt while another

interrupt is still active is sometimes described as interrupt nesting.

Priority masking
The ICCPMR for a CPU interface defines a priority threshold for the target processor, see Interrupt Priority Mask Registe. The GIC only signals pending interrupts with a higher priority than this threshold value to the target processor. A value of zero, the register reset value, masks all interrupts to the associated processor.
The GIC always masks an interrupt that has the largest supported priority field value. This provides an additional means of preventing an interrupt being signalled to any processor.

Priority grouping
Priority grouping splits each priority value into two fields, the group priority and the subpriority fields. The GIC uses the group priority field to determine whether a pending interrupt has sufficient priority to preempt a currently active interrupt.

The binary point field in the ICCBPR controls the split of the priority bits into the two parts. This 3-bit field specifies how many of the least significant bits of the 8-bit interrupt priority field are excluded from the group priority field, as following table shows.

Binary point value	Group priority field	Subpriority field	Field with binary point
0	[7:1]	[0]	ggggggg.s
1	[7:2]	[1:0]	gggggg.ss
2	[7:3]	[2:0]	ggggg.sss
3	[7:4]	[3:0]	gggg.ssss
4	[7:5]	[4:0]	ggg.sssss
5	[7:6]	[5:0]	gg.ssssss
6	[7]	[6:0]	g.sssssss
7	No preemption	[7:0]	.ssssssss

Where multiple pending interrupts share the same group priority, the GIC uses the subpriority field to resolve the priority within a group.

12.6.3 The effect of the Security Extensions on interrupt handling

If a GIC CPU interface implements the Security Extensions, it provides two interrupt output signals, IRQ and FIQ:
• The CPU interface always uses the IRQ exception request for Non-secure interrupts
• Software can configure the CPU interface to use either IRQ or FIQ exception requests for Secure interrupts.

Security Extensions support
Software can detect support for the Security Extensions by reading the ICDICTR.SecurityExtn bit, see Interrupt Controller Type Register (ICDICTR). Secure software makes Secure writes to the ICDISRs to configure each interrupt as Secure or Non-secure, see Interrupt Security Registers (ICDISRn).

In addition:
• The banking of registers provides independent control of Secure and

Non-secure interrupts.
- The Secure copy of the ICCICR has additional fields to control the processing of Secure and Non-secure interrupts, see CPU Interface Control Register (ICCICR) These fields are:

— the SBPR bit, that affects the preemption of Non-secure interrupts.
— the FIQEn bit, that controls whether the interface signals Secure interrupts to the processor using the IRQ or FIQ interrupt exception requests.
— the AckCtl bit, that affects the acknowledgment of Non-secure interrupts.
— the EnableNS bit, that controls whether Non-secure interrupts are signaled to the processor, and is an alias of the Enable bit in the Non-secure ICCICR.
- The Non-secure copy of the ICCBPR is aliased as the ICCABPR, see Aliased Binary Point Register (ICCABPR). This is a Secure register, meaning it is only accessible by Secure accesses.

Effect of the Security Extensions on interrupt acknowledgement
When a processor takes an interrupt, it acknowledges the interrupt by reading the ICCIAR. A read of the ICCIAR always acknowledges the highest priority pending interrupt for the processor performing the read.

If the highest priority pending interrupt is a Secure interrupt, the processor must make a Secure read of the ICCIAR to acknowledge it.

By default, the processor must make a Non-secure read of the ICCIAR to acknowledge a Non-secure interrupt. If he AckCtl bit in the Secure ICCICR is set to 1 the processor can make a Secure read of the ICCIAR to acknowledge a Non-secure interrupt.
If the read of the ICCIAR does not match the security of the interrupt, taking account of the AckCtl bit value for a Non-secure interrupt, the ICCIAR read does not acknowledge any interrupt and returns the value:
- 1022 for a Secure read when the highest priority interrupt is Non-secure
- 1023 for a Non-secure read when the highest priority interrupt is Secure.

12.6.4 The effect of Security Extensions on interrupt priority

If the GIC supports the Security Extensions:
- Secure software must program the ICDISRs to configure each supported interrupt as either Secure or Non-secure, see Interrupt Security Registers (ICDISRn).
- the GIC provides Secure and Non-secure views of the interrupt priority settings

Software views of interrupt priority
When a processor reads the priority value of an interrupt, the GIC returns either the Secure or the Non-secure view of that value, depending on whether the access is Secure or Non-secure. This section describes the two views of interrupt priority, and the relationship between them.

In this product, 64 priority levels are implemented.

F	E	D	C	B	A	0	0

Fig. 12-3Secure view of the priority field for a Secure interrupt
Fig 12-3 shows the Secure view of a priority value field for a Secure interrupt.

A-F is the priority value. The low-order bits of the priority fields are RAZ/WI

E	D	C	B	A	0	0	0

Fig. 12-4 Non-secure view of the priority field for a Non-secure interrupt

Fig 12-4 shows the Non-secure view of a priority value field for a Non-secure interrupt. A-E is the priority value. The low-order bits of the priority fields are RAZ/WI

The Non-secure view of a priority value does not show how the value is stored in the Distributor. Taking the value from a Non-secure write to a priority field, before storing the value the Distributor:
• right-shifts the value by one bit
• sets bit [7] of the value to 1.

This translation means the priority value for the Non-secure interrupt is in the top half of the possible value range, meaning the interrupt priority is in the bottom half of the priority range.
A Secure read of the priority value for a Non-secure interrupt returns the value stored in the distributor. Fig12-5 shows this Secure view of the priority value field for a Non-secure interrupt that has had its priority value field set by a Non-secure access, or has had a priority value with bit [7] == 1 set by a Secure access:

1	E	D	C	B	A	0	0

Fig. 12-5 Secure read of the priority field for a Non-secure interrupt

Chapter 13 DMC (Dynamic Memory Interface)

13.1 Overview

The DMC includes two section: dynamic ram protocol controller(PCTL) and phy controller (PHYCTL).

The PCTL SoC application bus interface supports a lowest-latency native application interface (NIF). To maximize data transfer efficiency, NIF commands transfer data without flow control. To simplify command processing, the NIF accepts addresses in rank, bank, row, column format.

The PHYCTL provides control features to ease the customer implementation of digitally controlled features of the PHY such as initialization, DQS gate training, and programmable configuration controls. The PHYCTL has built-in self test features to provide support for production testing of the compatible PHY. It also provides a DFI 2.1 interface to the PHY.

The DMC supports the following features:

- Complete, integrated, single-vendor DDR2, DDR3, mDDR, LPDDR2 solution .
- DFI 2.1 interface compatibility
- Up to 1066 Mbps in 1:1 frequency ratio, using a 533MHz controller clock and 533MHz memory clock.
- Support for x8, x16, and x32 memories, for a total memory data path width of up to 72 bits
- Up to 2 memory ranks; devices within a rank tie to a common chip select
- Up to 8 open memory banks, maximum of eight per rank
- Per-NIF transaction controllable bank management policies: open-page, close-page
- Low area, low power architecture with minimal buffering on the data, avoiding duplication of storage resources within the system
- PCTL NIF slave interface facilitates easy integration with an external scheduler or standard on-chip buses
- Efficient DDR protocol implementation with in-order column (Read and Write) commands and out- of-order Activate and Precharge commands
- Three clock cycles best case command latency (best case is when a command is to an open page and the shift array in the PCTL is empty).
- 1T or 2T memory command timing
- Automatic clock stop, power-down and self-refresh entry and exit.
  Clock stop is mDDR/LPDDR2 only
- Software and hardware driven self-refresh entry and exit
- Programmable memory initialization
- Partial population of memories, where not all DDR byte lanes are populated with memory chips
- Programmable per rank memory ODT (On-Die Termination) support for reads and writes
- APB interface for controller software-accessible registers
- Programmable data training interface:

Assists in training of the data eye of the memory channel

Provides a method for testing large sections of memory

- Support for industry standard UDIMMs (Unbuffered DIMMs) and RDIMMs (Registered DIMMs)
- Automatic DQS gate training and drift compensation
- At-speed built-in-self-test (BIST) loopback testing on both the address and data channels for DDR PHYs
- PHY control and configuration registers
- Optional, additional JTAG interface to configure registers
- DFI 2.1 interface

13.2 Block Diagram

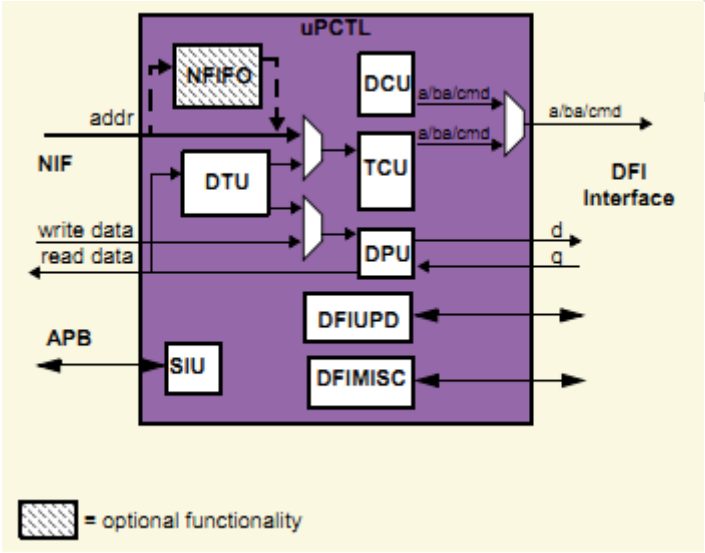


Fig. 13-1Protocol controller architecture

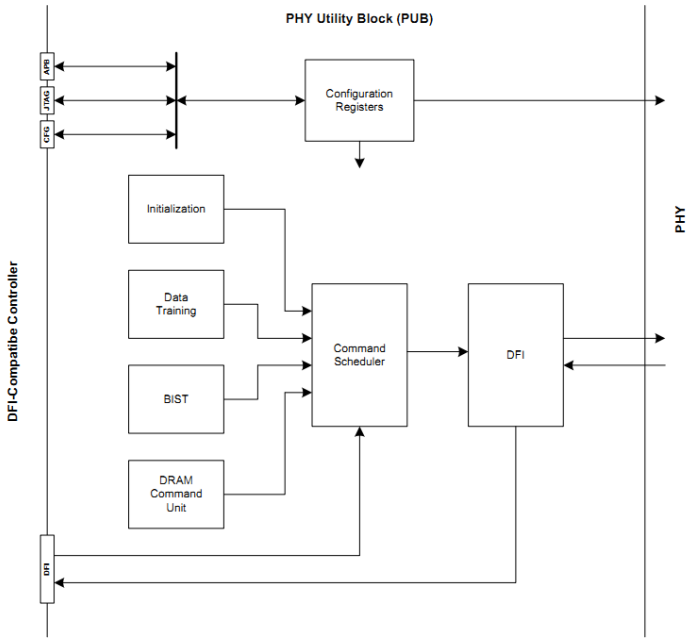


Fig. 13-2PHY controller architecture

13.3 Function description

PCTL operations are defined in terms of the current state of the Operational State Machine. Software can move PCTL in any of the operational states by issuing commands via the SCTL register. Transitions from one operational state to the other occur pass through a "transitional" state. Transitional states are exited automatically by the PCTL after all the necessary actions required to change operational state have been completed. The current operational state of PCTL is reported by the STAT register and is also available from the p_ctl_stat output.

PCTL supports the following operational states:

- Init_mem - This state is the default state entered after reset. All writable registers can be programmed. While in this state software can program PCTL and initialize the PHY and the memories. The memories are not refreshed and data that has previously been written to the memories may be lost as a result. The Init_mem state is also used when it is desirable to stop any automatic PCTL function that directly affects the memories, like Power Down and Refresh, or when a software reset of the memory subsystem has to be executed.

- Config - This state is used to suspend temporarily the normal NIF traffic and allow software to reprogram PCTL and memories if necessary, while still keeping active the periodic generation of Refresh cycles to the memories. Power Down entry and exit sequences are possible while in Config state.

- Access - This is the operational state where NIF transactions are accepted by the PCTL and converted into memory read and writes. None of the registers can be programmed except SCFG, SCTL, ECCCLR and DTU* registers.

- Low_power - Memories are in self refresh mode. The PCTL does not generate refresh cycles while in this state.

Access and Low_power states can also be entered and exited by the hardware low power signals (c_*). In case of conflicting software and hardware low-power commands, the resulting operational state taken by the controller can be either one of the two conflicting requests.
Figure 13-3 illustrates the operational and transitional states.

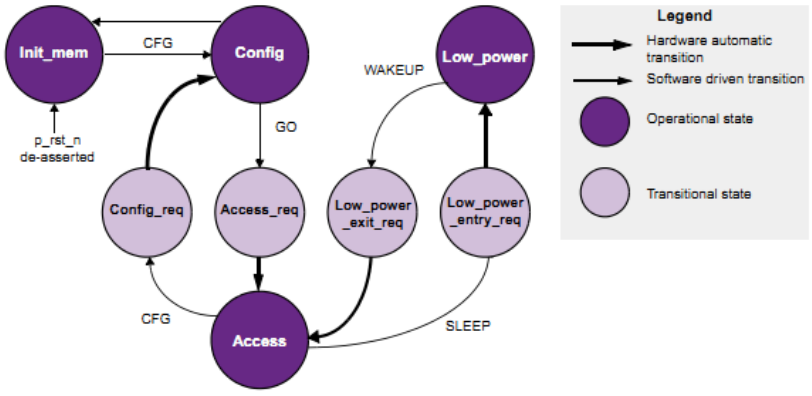


Fig. 13-3Protocol controller architecture

The PHYCTL provides control features to ease the customer implementation of digitally controlled PHY features such as read DQS training, data eye training, output impedance calibration, and so on. The PHTCTL has built-in self test features to provide support for production testing of PHY. It also provides a DFI 2.1 interface to the PHY. The PHYCTL performs, in sequence, various tasks required by the PHY before it can commence normal DDR operations. SDRAM memory read/write access through the DDR PHY is primarily through a DFI 2.1 interface on the PHYCTL. Therefore, the memory controller used with the PHY must be DFI 2.1 compatible.

Access to the PHYCTL internal control features and registers is through a dedicated configuration port, which can be either APB or CFG (generic configuration interface). An optional JTAG interface can also be compiled in as an additional second configuration port to co-exist with either the APB or CFG main configuration ports. The PHYCTL is driven off two clocks, the controller clock (ctl_clk) and the configuration clock pclk for an APB interface.

The controller clock is the same clock driving the memory controller and will be the same frequency as the SDRAM clock (ck). The configuration clock can run at a frequency equal to or less than the controller clock. The configuration clock drives all non-DDR timing logic, such as configuration registers, PHYinitialization, output impedance, and so on.

13.4 DDR PHY

13.4.1 DDR PHY Overview

In order to facilitate robust system timing and ease of use, DMC interface and control architecture utilizes a mixture of soft-IP and hard-IP design elements. The main control logic (Memory Controller) is supplied as soft-IP. The PHY is comprised of hard-IP components that include double-data rate InterfaceTiming Modules(ITM), input and output path DLLs, and application-specific SSTL I/Os.

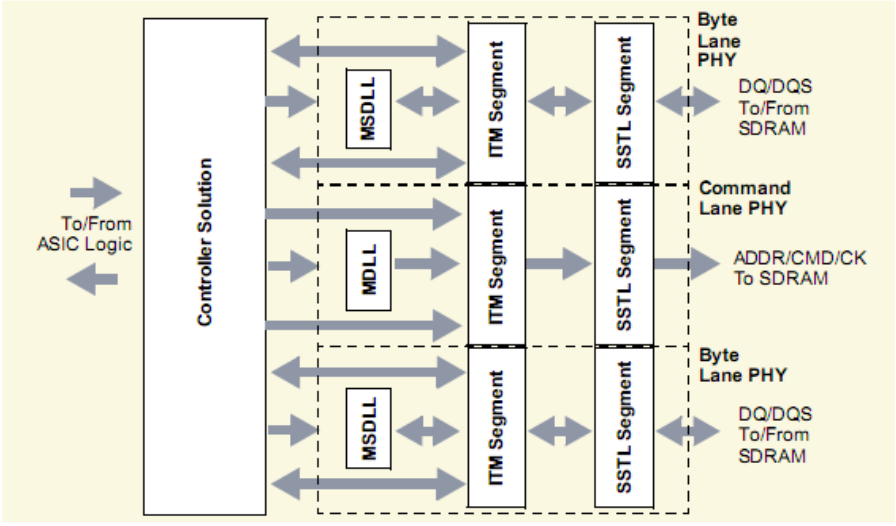


Fig. 13-4 DDR PHY architecture

In order to maximize system timing margins on the command/write path, inputs to the SDRAM are provided with the clock or data strobe centered in the associated data eye. The ITM components perform timing translation for the various signal groups of the interface. The hardened ITM approach ensures minimal pin to pin skew while allowing optimal circuit design for drive and capture circuitry. A DLL is utilized to facilitate the clock centering. In the Command Lane, a master DLL (MDLL) is utilized. In the Byte Lane, the master portion of a master/slave DLL macrocell (MSDLL) is utilized.

On the read path, read data from the SDRAM is arriving from the SDRAM edge aligned with the data strobes. In order to maintain maximum system timing margins on the input path, the data strobes are translated to the center of the data eye. The MSDLL macrocell associated with each Byte Lane contains a master DLL and 2 slave DLLs (mirror delay lines). The slave DLL portion of the MSDLL is utilized to facilitate the clock centering. DQS and DQS_b strobe inputs each utilize one of these slave DLL functions. The captured double data rate inputs are then converted to single data rate and passed onto the DDR Controller RTL logic. The ITM facilitates both data capture and DDR to SDR conversion.

The physical interface between the DDR controller and DDR SDRAMs uses DDR-specific SSTL I/O buffers with programmable on-die termination (ODT). These I/Os operate at either 1.8V for LPDDR/DDRII interfacing (SSTL_18).

DMC interface and control architecture follows a common signal grouping philosophy. A Byte Lane is a complete eight-bit data unit consisting of the associated DQ, DM, and DQS/DQS_b signals. A 32-bit system would consist of four Byte Lanes. A Command Lane is a complete command and address unit including also clock signals. There would normally be only one Command Lane in a particular DDR SDRAM interface. All clock and data signals relative to a Lane, either Byte or Command, are isolated to within that Lane only. Timing critical clock and data signals do not traverse between Lanes. Implementation of a memory interface involves placing the Command Lane components, placing the Byte Lane components, and standard synthesis/place and route to complete the design.

Each SSTL cell communicating with the SDRAM has an associated ITM component. The ITM library consists of individual components designed specifically for signal groups of address and command, data & data mask, and data strobes. In order to ensure low pin to pin skews and facilitate ease of implementation, the ITM components are tileable. DLL output clock distribution is embedded within the ITM components.

13.4.2 Lane-Based Architecture

Byte Lane PHY

The data bus interface to the external memory is organized into self-contained units referred to herein as Byte Lanes. The external memory components are designed to support Byte Lanes for optimal system timing. The partitioning of the data word into discrete Byte Lanes allows pin to pin skew to be managed across a much smaller group of signals than would typically be required.

All components of the Byte Lane PHY are designed to permit connectivity by abutment. The ITM connects by abutment to the SSTL I/O, and the DLL connects by abutment to the ITM.

The SDRAM contains data strobes associated with each 8 bits of data and there is a timing skew allowance between the main clock signal to the SDRAM and its data strobe inputs during a Write command (tDQSS). 8bit memory components provide a single DQS.

A Byte Lane consists of the following I/O slots:

- ✧ 8 data bits (DQ)
- ✧ data strobe bits (DQS / DQS_b)
- ✧ 1 data mask bit (DM)
- ✧ I/O power and ground cells
- ✧ Core power and ground cells

Each functional I/O slot has an associated ITM module, including DQ, DM, and DQS/DQS_b. The ITMs provide a mechanism for monitoring read timing drift, which can be used to adjust timing to maintain optimum system margins. Drift

analysis and compensation is performed by the controller on a per Byte Lane basis. The ITM components contain the functions to monitor DQS drift and permit timing adjustment, the controller provides the analysis and control for these functions. These functions operate dynamically for each data bit of every user-issued Read command. There are no overhead penalties in channel bandwidth or utilization incurred by the use of these functions.

The memory interface (PHY) architecture is based on the concept of independent, but related, signal groups to provide the highest level of system timing performance. In order to maintain robust system timing, all clock and data signals relevant to a Byte Lane remain within that Byte Lane. These signals are not shared between other Byte Lanes or between a Byte Lane and a Command Lane. Alternate approaches require clock distribution networks that span the full length of the interface including all address, command, and data signals. These large clock distribution networks are difficult for the user to design and implement, and add an additional component of pin to pin skew to the critical timing budget.

A DLL macrocell (MSDLL) consisting of a master DLL and 2 slave DLLs (mirror delay lines) is utilized at each Byte Lane to facilitate optimal PHY timing for drive and capture of DDR data streams, and allows the Lanes to be independent. The master DLL section provides outputs for DDR data stream creation to the SDRAMs and acts as a reference for the slave delay line sections. The slave delay line sections translate the incoming DQS/DQS_b into the center of the read data eye to maximize read system timing margins.

The user is permitted to fine tune the relationship of the DQS and DQ signals to maximize read system timing margin. The DLL includes adjustability of the slave delay lines for the DQS and DQS_b signals, which provide byte-wide timing adjustments. The ITMs include adjustability of the read DQS/DQS_b strobe timing, which provides byte-wide timing adjustments. The ITMs include adjustability of the read DQ signal timing, which provides per-bit timing adjustability. To permit Lane-independent timing adjustments, DLL adjustment bits are provided by the controller per Byte Lane and ITM adjustment bits are provided per bit.

DMC interface and control solution allows memory systems with a word width narrower than the design. Our system is designed with a 32 bit data width and it can then be utilized with either 16 bit or 32 bit memory systems. The controller contains register settings to allow the desired operational mode to be set in the final device.

The DDR-specific SSTL I/Os include programmable ODT and output impedence selection. The ODT and output impedances can be be dynamically calibrated to compensate for variations in voltage and temperature. The ODT feature can be disabled by the controller. When ODT is enabled by the controller, the SSTL I/O automatically enables its internal ODT circuitry when in input mode and disable this circuitry when in output mode, as determined by the output enable signal. The initial programming and subsequent calibration of the of the ODT and output impedance is achieved through the use of an impedance control loop that can be triggered to calibrate the ODT and output impedance values at the I/Os based on the desired impedance value when compared to an precision external resistor. All the necessary pieces of the impedance control loop are included in the SSTL I/O library.

There are four Byte Lanes in our chip of 32 bit memory system.

Command Lane PHY

The control and address interface to the external memory is organized into a self-contained unit referred to herein as a Command Lane. DMC interface

contains a single Command Lane and four Byte Lanes.

All components of the Command Lane PHY are designed to permit connectivity by abutment. The ITM connects by abutment to the SSTL I/O, and the DLL connects by abutment to the ITM.

A typical Command Lane consists of the following I/O slots:

- ✧ Memory clocks (CK/CK_b)
- ✧ Command signals (RAS_b, CAS_b, WE_b)
- ✧ 1 or more clock enable (CKE)
- ✧ 1 or more on-die termination (ODT)
- ✧ 2 chip select (CS_b)
- ✧ 3 bank address (BA)
- ✧ 16 row/column address (A)
- ✧ I/O power and ground cells
- ✧ Core power and ground cells

The system clock input is used to provide the source clock for the memory interface. Memory controller supports 2 SDRAM ranks. There is one CKE, ODT, and CS_b signal provided for each rank.

Each functional I/O slot has an associated ITM module, with exception of the system clock input. A master DLL (MDLL) is utilized with the Command Lane to facilitate optimal PHY timing for drive of DDR data streams, and allows the Lane to be independent. The DLL macrocells provide two 0 degree phase outputs, one which can be used to drive the controller logic. The Command Lane MDLL is used for this purpose.

To permit Lane-independent timing adjustments, DLL and ITM adjustment bits are provided by the controller separately for Command and Byte Lanes.

13.4.3 Master DLL(MDLL)

Master DLL for DDRII, and LPDDR applications is a Delay Locked Loop that takes an input reference clock (clk_in) and generates four clock outputs, each delayed in quarter clock cycle (90º) increments. These four clock phases (clk_0, clk_90, clk_180, clk_270) can be generated with very high accuracy and low jitter across a wide range of frequencies.

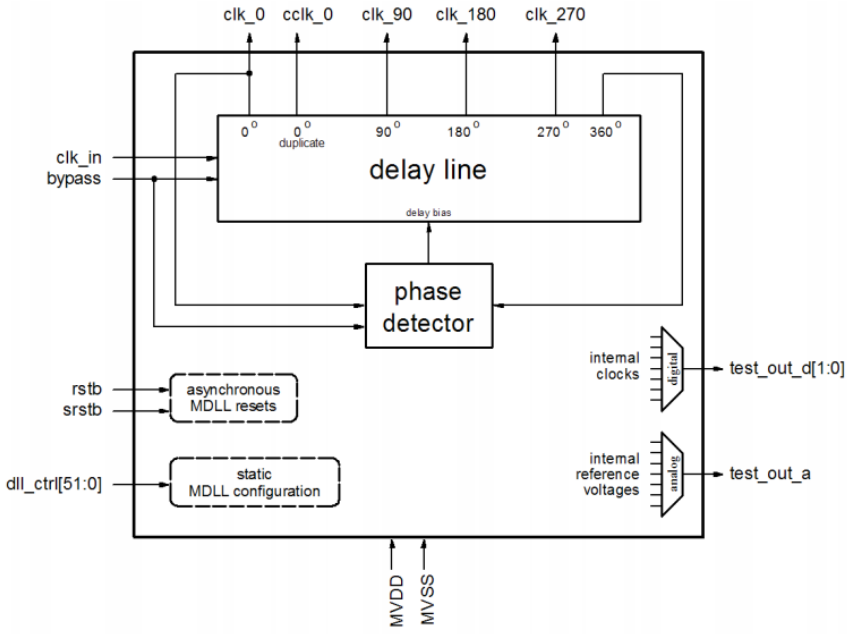


Fig. 13-5 DDR PHY master DLL architecture diagram

A number of test modes and configuration settings are included:

- ✧ A bypass mode shuts down all analog circuitry, and directly buffers the input clock and strobes with appropriate delays and inversions to the output clocks and strobes. This mode can be used for low speed functional or IDDQ testing.
- ✧ A digital test output (test_out_d) provides direct observability of several internal reference clock and timing nodes.
- ✧ An analog test output (test_out_a) provides direct observability of several internal reference voltages.

Master DLL Control for Trim and Test

The performance and testing of the MDLL can be accessed through the dll_ctrl bus.

Table 13-1DDR PHYtrim and test MDLL control

Static Input		Field	Description
dll_ctrl	[1:0]		Reserved
	[4:2]	ipump_trm[2:0]	Charge pump current trim
	[5]	test_ctrl_en	Test control enable for analog and digital test outputs
	[8:6]	test_ctrl_d[2:0]	Digital test control. Selects the digital signal to be viewed at the digital test output
	[10:9]	test_ctrl_a[1:0]	Analog test control. Selects the analog signal to be viewed at the analog test output
	[11]		Reserved
	[14:12]	bias_trm[2:0]	Bias generator frequency trim
	[19,15]	fdtrm[1:0]	Bypass mode fixed delay trim
	[22:20]	bias_trm[6:4]	Bias generator control voltage trim
	[23]	bps200	Bypass frequency select
	[28:24]		Reserved
	[29]		Reserved
	[37:30]		Reserved
	[43:38]	fb_trm[5:0]	Feedback delay adjust
	[49:44]		Reserved
	[50]	test_hizb_a	Analog test output tri-stated control
	[51]		Reserved

Charge Pump Current Trim:

Table 13-2 charge pump current trim in dll_ctrl

Field	Setting	Function	Suggested Default
ipump_trm[2:0]	000	Maximum current	000
	111	Minimum current	

Digital Test Control:

Table 13-3DLL digital test control in dll_ctrl

test_ctrl_en	test_ctrl_d[2:0]	Function	Suggested Default
0	xxx	digital test outputs disabled (drive '0')	0,000
1	000	0° output clock (clk_0)	
1	001	90° output clock (clk_90)	
1	010	180° output clock (clk_180)	
1	011	270° output clock (clk_270)	

1	100	360° internal clock (clk_360_int)	
1	101	Speed-up pulse (spdup)	
1	110	Slow-down pulse (slwdn)	
1	111	Asic output clock (cclk_0)	

Analog Test Control:

Table 13-4DLL analog test control in dll_ctrl

test_hizb_a	test_ctrl_en	test_ctrl_a [1:0]	Function	Suggested Default
0	x	xx	Tri-state	0,0,00
1	0	xx	MVSS	
1	1	00	Filter output (Vc)	
1	1	01	Replica bias output for NMOS (Vbn)	
1	1	10	Replica bias output for PMOS (Vbp)	
1	1	11	MVDD	

Bias Generator Trim:

The bias generator trim capability can be used to adjust the behavior of the bias voltages being supplied to the delay line. Characteristics of the DLL that may warrant an adjustment of this trim value include the inability to lock due to a slow clock (suggest decreasing Vc adjust), inability to lock due to fast clock (suggest increasing Vc adjust) and increase noise margin on bias voltages (suggest decreasing Fmax adjust).The bit fields described in the following table can be set to any value between 000(binary) and 111(binary).

Table 13-5 bias generator trim in dll_ctrl

Field	Setting	Function	Suggested Default
bias_trm[2:0]	000	Fmax trim: minimum adjust	111
	111	Fmax trim: maximum adjust	
bias_trm[6:4]	000	Vc level trim: minimum adjust	011
	111	Vc level trim: maximum adjust	

Feedback Trim:

The feedback trim capability can be used in the event that an adjustment is desired in the phase detector feedback of the DLL. Characteristics of the DLL that may warrant an adjustment of this trim value include non-optimal phase alignment. The lower 3 bits (2:0) are used for feed-back delay trimming and the upper 3 bits (5:3) are used for feed-forward delay trimming. The feed-back trimming is used to decrease total delay, decreasing the amount of delay between phase outputs. The feed-forward trimming is used to increase total delay, increasing the amount of delay between phase outputs. For each 3-bit field, the inputs can be set to any value between 000(binary) and 111(binary).

Table 13-6MDLL feedback trim in dll_ctrl

Field	Setting	Function	Suggested Default
fb_trm[5:3] (feed-forward path)	000	Minimum additional delay	000
	111	Maximum additional delay	
fb_trm[2:0] (feed-back path)	000	Minimum additional delay	000
	111	Maximum additional delay	

Bypass Mode

The DLL has a bypass mode which allows phased clocks to be generated with analog locking circuitry disabled. This mode may be used for low-speed functional testing and for IDDq testing. Bypass mode can also be used when operating with LPDDR SDRAMs. When bypass mode is enabled, all analog circuitry is disabled, and all static current paths are shut down.

Bypass mode has two settings for the clk_90 delay to optimize it for two different frequency ranges.

Table 13-7 MDLL bypass mode frequency range in dll_ctrl

Field	Setting	Function	Suggested Default
bps200	0	0 to 100MHz	0
	1	0 to 200MHz	

It is also possible to trim the 90-degree delay using the fdtrm control bits.

Table 13-8fdtrm control bits in dll_ctrl

Field	Setting	Function	Suggested Default
fdtrm[1:0]	00	nominal delay	00
	01	nominal delay – 10%	
	10	nominal delay + 10%	
	11	nominal delay + 20%	

13.4.4 Master-Slave DLL(MSDLL)

Master-Slave DLL for DDRII, and LPDDR applications is an integrated Delay Locked Loop and a pair of slave delays.The Delay Locked Loop (DLL) takes an input reference clock (clk_in), and generates four clock outputs, each delayed in quarter clock cycle (90º) increments. These four clock phases (clk_0, clk_90, clk_180, clk_270) can be generated with very high accuracy and low jitter across a wide range of frequencies.

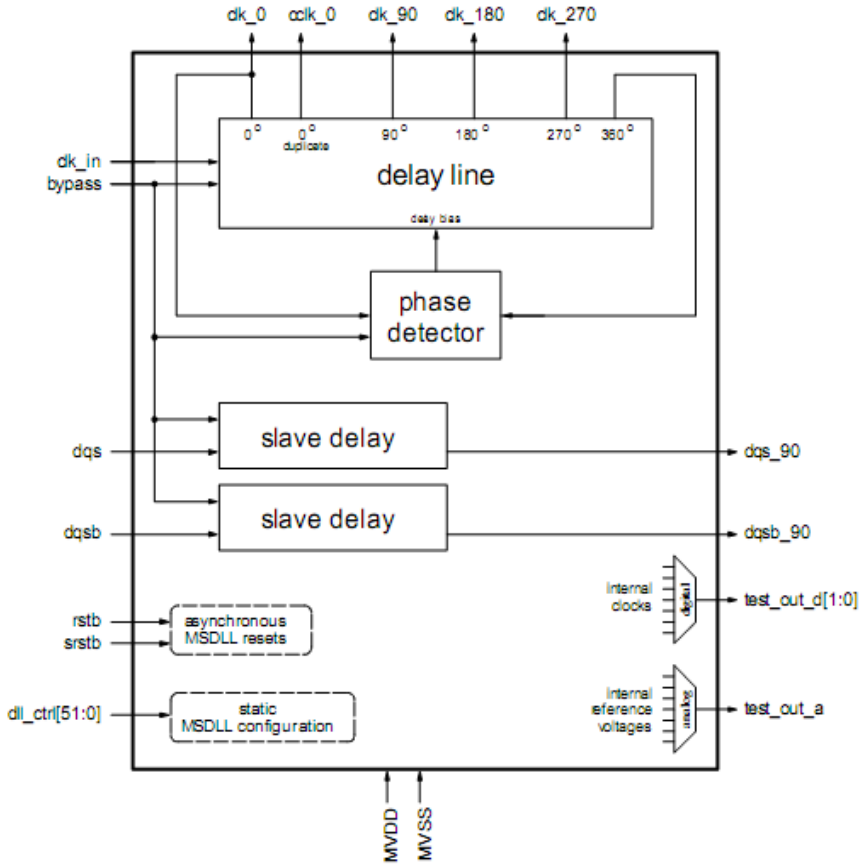


Fig. 13-6 DDR PHY master-slave DLL architecture diagram

The slave delay pair uses timing reference from the delay line to provide a highly accurate 90º delay to dqs and dqsb inputs (generating dqs_90 and dqsb_90 respectively).

A number of test modes and configuration settings are included:

- ✧ A bypass mode shuts down all analog circuitry, and directly buffers the input clock and strobes with appropriate delays and inversions to the output clocks and strobes. This mode can be used for low speed functional or IDDQ testing.
- ✧ A digital test output (test_out_d) provides direct observability of several internal reference clock and timing nodes.
- ✧ An analog test output (test_out_a) provides direct observability of several internal reference voltages.

The primary application for MSDLL is a DDRII Byte Lane PHY with Interface Timing Modules (ITMs)..

MSDLL Control for Trim and Test

The performance and testing of the MSDLL can be accessed through the dll_ctrl bus. Many of these controls are the same as the MDLL, therefore, this section only describes the settings that are different.

Table 13-9 DDR PHYMSDLL control for trim and test

Static Input		Field	Description
dll_ctrl	[1:0]		Reserved
	[4:2]	ipump_trm[2:0]	Charge pump current trim
	[5]	test_ctrl_en	Test control enable for analog and digital test outputs
	[8:6]	test_ctrl_d[2:0]	Digital test control. Selects the digital signal to be viewed at the digital test output
	[10:9]	test_ctrl_a[1:0]	Analog test control. Selects the analog signal to be viewed at the analog test output
	[11]	test_ctrl_switch	Test control switch. Selects the analog and digital test signals of master or slave
	[14:12]	bias_trm[2:0]	Master bias generator frequency trim
	[19,15]	fdtrm[1:0]	Master bypass fixed delay trim
	[18:16]	bias_trm[6:4]	Master bias generator control voltage trim
	[22:20]	sl_bias_trm[2:0]	Slavebias generator control voltage trim
	[23]	bps200	Bypass frequency select
	[26:24]	sl_bias_trm[6:4]	Slave bias generator control voltage trim
	[28:27]	fdtrm_sl[1:0]	Slave bypass fixed delay trim
	[29]	lock_det_en	Lock detector enable
	[31:30]		Reserved
	[37:32]	sl_fb_trm[5:0]	Slave feedback delay adjust
	[43:38]	fb_trm[5:0]	Master feedback delay adjust
	[45:44]	sl_bypass_start_up[1:0]	Slave auto-startup bypass
	[49:46]	sl_phase_trm[3:0]	Slave phase lock trim
	[50]	test_hizb_a	Analog test output tri-stated control
	[51]		Reserved

MSDLL Digital Test Control:

Table 13-10MSDLL digital test control in dll_ctrl

test_ctrl_en	test_ctrl_switch	test_ctrl_d[2:0]	Function	Suggested Default
0	x	xx	digital test outputs disabled (drive '0')	0,0,000
1	0	000	0°output clock (clk_0)	
1		001	90°output clock (clk_90)	
1		010	180° output clock (clk_180)	
1		011	270° output clock (clk_270)	
1		100	360° internal clock (clk_360_int)	
1		101	Master speed-up pulse (spdup)	
1		110	Master slow-down pulse (slwdn)	
1		111	Output clock (cclk_0)	
1	1	000	Input signal dqs	
1		001	Slave input clock reference (clk_90_in)	
1		010	Slave internal feedback clock (clk_0_out)	
1		011	Output signal dqsb_90	
1		100	Output signal dqs_90	
1		101	Slave speed-up pulse (spdup)	
1		110	Slave slow-down pulse (slwdn)	
1		111	Auto-lock enable signal	

MSDLL Analog Test Control:

Table 13-11MSDLL analog test control in dll_ctrl

test_hizb_a	test_ctrl_en	test_ctrl_switch	test_ctrl_a[1:0]	Function	Suggested Default
0	x	x	xx	Tri-state	0,0,0,00
1	0	x	xx	MVSS	
1	1	0	00	Master Filter output (Vc)	
1	1		01	Master Replica bias output for NMOS (Vbn)	
1	1		10	Master Replica bias output for PMOS (Vbp)	
1	1		11	MVDD	
1	1	1	00	Slave Filter output (Vc)	
1	1		01	Slave Replica bias output for NMOS (Vbn)	
1	1		10	Slave Replica bias output for PMOS (Vbp)	
1	1		11	MVDD	

MSDLL Lock Detector Enable:

This setting enables start of the slave DLL section after the master DLL section has reached lock. Characteristics of the DLL that may warrant an adjustment of this trim value include the slave DLL delay remaining in it's reset state (minimum delay, much less than 90 degrees) after the DLL lock time.

Table 13-12MSDLL lock detector enable in dll_ctrl

Field	Setting	Function	Suggested Default
lock_det_en	0	Disable lock detector	0
	1	Enable lock detector	

Slave Auto-Startup Bypass:
By default, the slave DLL automatically starts to lock during the time the master is locking, after the master has begun to approach lock. This setting permits the user to manually start-up the slave DLL. To bypass the automatic startup, this setting should be set to '10'. Once the specified number of clocks has passed for the master DLL to achieve lock, the user sets this field to '11' to permit the slave DLL to startup. The user then waits for the specified number of clocks for the slave DLL to lock before proceeding. Characteristics of the slave DLL that might warrant a manual startup of the slave DLL include the inability for the slave DLL to produce a consistent and/or correct phase difference between the input signal and the output signal.

Table 13-13 slave auto_startup bypass in dll_ctrl

sl_bypass_start_up[1:0]	Function	Suggested Default
0X	Slave DLL automatically starts up	00
10	Slave DLL's automatic startup is disabled; the phase detector is disabled	
11	Slave DLL's automatic startup is disabled; the phase detector is enabled	

Slave DLL Phase Trim:
Selects the phase difference between the input signal and the corresponding output signal of the slave DLL. This setting applies to the dqs to dqs_90 and dqsb to dqsb_90 paths. The nominal phase difference is 90 degrees. Users may select to modify this value to account for factors external to the DLL, which require the DLL to produce a delay of greater than or less than the nominal 90 degrees. When modifying the value of these bits, the user does not need to issue a reset to the DLL but should wait the equivalent of the DLL lock time before the slave DLL circuitry is used (such as, receiving Read data from an SDRAM) to ensure the DLL has adequate time to stabilize with the new settings.

Table 13-14 slave DLL phase trim in dll_ctrl

sl_phase_trm[3:0]	Phase Difference (degrees)	Suggested Default
0000	90	0000
0001	72	
0010	54	
0011	36	
0100	108	
0101	90	
0110	72	
0111	54	
1000	126	
1001	108	
1010	90	
1011	72	
1100	144	
1101	126	
1110	108	
1111	90	

MSDLL Bypass Mode

The DLL bypass mode, when enabled, shuts down all analog delay paths and phase detection circuitry and generates output clocks as directly buffered and

inverted versions of clk_in. Bypass mode can be used for low-speed functional testing or for IDDQ testing. Bypass mode can also be used when operating with LPDDR SDRAMs. When bypass mode is enabled, all analog circuitry is disabled, and all static current paths are shut down. Phased outputs are generated during bypass with inverters and standard delays:

clk_0 = buffered clk_in
clk_90 = delayed version of clk_0
clk_180 = inverted clk_0
clk_270 = inverted clk_90
cclk_0 = buffered clk_in
dqs_90 = delayed version of dqs
dqsb_90 = delayed version of dqsb

Bypass mode has two settings for the clk_90 delay to optimize it for two different frequency ranges same as MDLL.

It is also possible to trim the MDLL 90 degree delay using the fdtrm control bits same as MDLL. And it is also possible to trim the MSDLL 90 degree delay using the fdtrm_sl control bits same as fdtrm.

13.4.5 DQS Gating

DDRII systems use a bidirectional data strobe which is driven by the host during memory writes, and by the SDRAM during memory reads. During active read commands, the ITMS basically acts as a buffer for the incoming DQS/DQS_b. A turn-around time exists between operations when neither device is driving the bus, and the strobe traces are held by termination circuitry at a mid-rail voltage.

While the DQS lines are held at mid-rail during inactive periods, an unknown value X is being received by the SSTL inputs. To prevent X from causing false transitions and other negative effects within the read path, the input read dqs strobe path is disabled when there is no active read data. The ITMS provides the functions to enable/disable this path, while the control of these functions is provided by the memory controller logic. A basic view of the enable/disable requirements is shown in following figure.

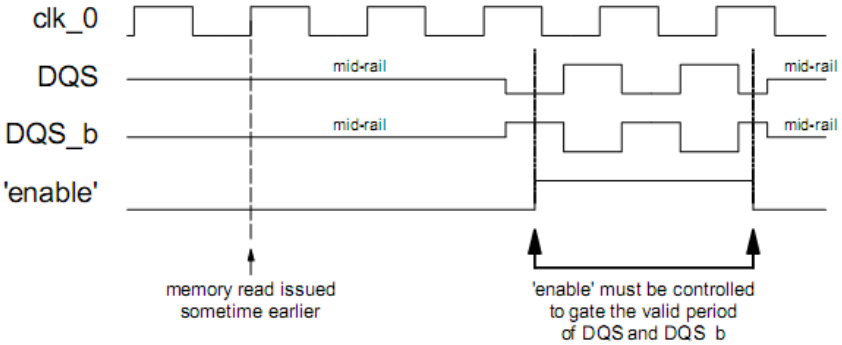


Fig. 13-7Strobe Gating Requirements During Read Operations

After a read is issued, the SDRAM drives DQS and DQS_b for a number of clock cycles equal to the read burst length. Differing SDRAM CAS latencies, clock cycle times, board trace lengths, and other analog factors between controller and SDRAM result in a variable latency between when the read was issued, and when the returning DQS/DQS_b strobes reach the ITMS. The goal of DQS gating is to control a window, which enables and disables the input read dqs path only when the DQS lines are active, not when they are at mid-rail. There is a pre-amble and post-amble surrounding the active DQS edges that is used as the point to perform the enabling and disabling of this window.

There are two windowing schemes supported by the ITMS - passive windowing and active windowing - which are selected by input dqs_config.

Passive Windowing

In the passive windowing mode (dqs_config = 1), the controller asserts dqs_en at the start of the window and de-asserts dqs_en at the end of the window. This provides the course (clock-cycle) position of the enable and disable edges. Fine tuning (1/4 clock cycle) of the window placement is selected by phase_sel[1:0].

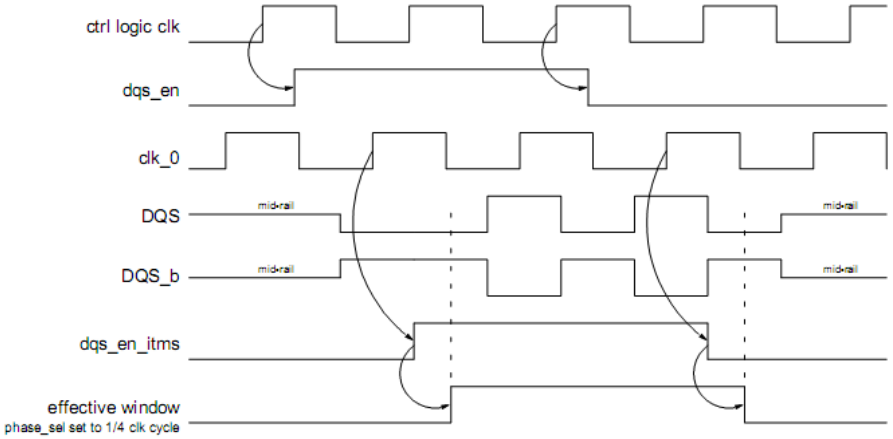


Fig. 13-8DQS gating – passive windowing mode

The phase_sel[1:0] settings are provided.

Table 13-15 phase selection for dqs gating

phase_sel[1:0] Phase Selection		
Setting	Selected Phase	Offset
00	clk_90 (90 deg)	1/4 clock cycle
01	clk_180 (180 deg)	1/2 clock cycle
10	clk_270 (270 deg)	3/4 clock cycle
11	clk_0 (360 deg)	1 clock cycle

Active Windowing

The active windowing mode addresses the fact that the postamble is shorter than the preamble. The optimal window position for the preamble and postamble are not necessarily the same. In the active windowing mode (dqs_config = 0), the controller asserts dqs_en for one clock cycle at the start of the window and asserts dqs_dis for one clock cycle at the end of the window. Internal to ITMS, the assertion of dqs_dis is shifted by a further 180 degrees to account for the fact that DQS_b occurs 180 degrees later than DQS. This provides the course (clock-cycle) position of the enable and disable edges.

Fine tuning (1/4 clock cycle) of the window placement is selected by phase_sel[1:0]. The effective window is opened in the same manner as in the passive windowing mode, such as dqs_en assertion plus the phase_sel offset. To close the window, the controller asserts dqs_dis to inform the ITMS to expect the last DQS_b rising edge of the burst. The phase_sel setting is applied to this to set the effective time at which to expect the last DQS_b rising edge. The last DQS_b rising edge of the burst is also the last data of the burst. This last DQS_b rising edge is used to close the window. Thus, the window is self-closing.

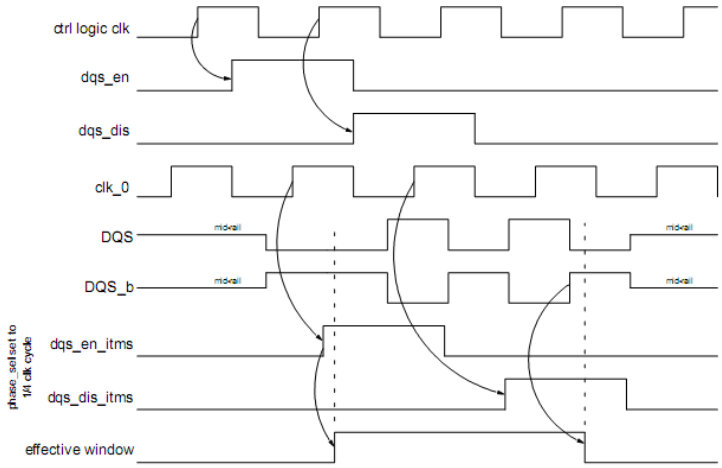


Fig. 13-9DQS gating – active windowing mode

13.4.6 Dynamic Strobe Drift Detection

DDRII systems can have a long round-trip path from the controller clock output (CK), to the SDRAM, and back to the controller data strobe input (DQS). The sum of potential variations in this path can exceed 25% of a clock cycle at high frequencies (>300MHz), so some compensation should be made if the path delay increases or decreases slowly, but significantly, during normal operation.

The ITMS component has a two-bit strobe drift indicator (dqs_drift), which changes value in grey code if the returning strobe drifts across internal 90° timing reference boundaries. The absolute value of this indicator is not important, but the change in value over time is.

Table 13-16 dynamic strobe drift indicators

dqs_drift[1:0]		DQS Drift Direction	Required Changes
Old Value	New Value		
00	01	forward	increase read data latency by 90 degrees
	10	backward	decrease read data latency by 90 degrees
01	11	forward	increase read data latency by 90 degrees
	00	backward	decrease read data latency by 90 degrees
10	00	forward	increase read data latency by 90 degrees
	11	backward	decrease read data latency by 90 degrees
11	10	forward	increase read data latency by 90 degrees
	01	backward	decrease read data latency by 90 degrees

13.5 Register description

13.5.1 Registers Summary

Name	Offset	Size	Reset Value	Description
DDR_PCTL_SCFG	0x0000	W	0x00000300	State Configuration Register
DDR_PCTL_SCTL	0x0004	W	0x00000000	Operational State Control Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_STAT	0x0008	W	0x00000000	Operational State Status Register
DDR_PCTL_INTRSTAT	0x000c	W	0x00000000	Interrupt Status Register
DDR_PCTL_MCMD	0x0040	W	0x00100000	Memory Command Register
DDR_PCTL_POWCTL	0x0044	W	0x00000000	Power Up Control Register
DDR_PCTL_POWSTAT	0x0048	W	0x00000000	Power Up Status Register
DDR_PCTL_CMDTSTAT	0x004c	W	0x00000000	Command Timers Status Register
DDR_PCTL_CMDTSTATEN	0x0050	W	0x00000000	Command Timers Status Enable Register
DDR_PCTL_MRRCFG0	0x0060	W	0x00000000	Mode Register Read Configuration 0
DDR_PCTL_MRRSTAT0	0x0064	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MRRSTAT1	0x0068	W	0x00000000	Mode Register Read Status 0 Register
DDR_PCTL_MCFG	0x0080	W	0x00040020	Memory Configuration Register
DDR_PCTL_PPCFG	0x0084	W	0x00000000	Partially Populated Memories Configuration Register
DDR_PCTL_MSTAT	0x0088	W	0x00000000	Memory Status Register
DDR_PCTL_LPDDR2ZQCFG	0x008c	W	0xab0a560a	LPDDR2 ZQ Configuration Register
DDR_PCTL_MCFG1	0x0090	W	0x00000000	Memory Configuration 1 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DTUPDES	0x0094	W	0x00000000	DTU Status Register
DDR_PCTL_DTUNA	0x0098	W	0x00000000	DTU Number of Addresses Created Register
DDR_PCTL_DTUNE	0x009c	W	0x00000000	DTU Number of Errors Register
DDR_PCTL_DTUPRD0	0x00a0	W	0x00000000	DTU Parallel Read 0 Register
DDR_PCTL_DTUPRD1	0x00a4	W	0x00000000	DTU Parallel Read 1 Register
DDR_PCTL_DTUPRD2	0x00a8	W	0x00000000	DTU Parallel Read 2 Register
DDR_PCTL_DTUPRD3	0x00ac	W	0x00000000	DTU Parallel Read 3 Register
DDR_PCTL_DTUAWDT	0x00b0	W	0x00000290	DTU Address Width Register
DDR_PCTL_TOGCNT1U	0x00c0	W	0x00000064	Toggle Counter 1us Register
DDR_PCTL_TINIT	0x00c4	W	0x000000c8	t_init Timing Register
DDR_PCTL_TRSTH	0x00c8	W	0x00000000	t_rsth Timing Register
DDR_PCTL_TOGCNT100N	0x00cc	W	0x00000001	Toggle Counter 100ns
DDR_PCTL_TREFI	0x00d0	W	0x00000001	t_refi Timing Register
DDR_PCTL_TMRD	0x00d4	W	0x00000001	t_mrd Timing Register
DDR_PCTL_TRFC	0x00d8	W	0x00000001	
DDR_PCTL_TRP	0x00dc	W	0x00010006	t_trp Timing Register
DDR_PCTL_TRTW	0x00e0	W	0x00000002	t_rtw Timing Register
DDR_PCTL_TAL	0x00e4	W	0x00000000	AL Register
DDR_PCTL_TCL	0x00e8	W	0x00000004	CL Timing Register
DDR_PCTL_TCWL	0x00ec	W	0x00000003	CWL Timing Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_TRAS	0x00f0	W	0x00000010	t_ras Timing Register
DDR_PCTL_TRC	0x00f4	W	0x00000016	t_rc Timing Register
DDR_PCTL_TRCD	0x00f8	W	0x00000006	t_rcd Timing Register
DDR_PCTL_TRRD	0x00fc	W	0x00000004	t_rrd Timing Register
DDR_PCTL_TRTP	0x0100	W	0x00000003	t_rtp Timing Register
DDR_PCTL_TWR	0x0104	W	0x00000006	t_wr Register
DDR_PCTL_TWTR	0x0108	W	0x00000004	t_wtr Timing Register
DDR_PCTL_TEXSR	0x010c	W	0x00000001	t_exsr Timing Register
DDR_PCTL_TXP	0x0110	W	0x00000001	t_xp Timing Register
DDR_PCTL_TXPDLL	0x0114	W	0x00000000	t_xpdll Timing Register
DDR_PCTL_TZQCS	0x0118	W	0x00000000	t_zqcs Timing Register
DDR_PCTL_TZQCSI	0x011c	W	0x00000000	t_zqcsi Timing Register
DDR_PCTL_TDQS	0x0120	W	0x00000001	t_dqs Timing Register
DDR_PCTL_TCKSRE	0x0124	W	0x00000000	t_cksre Timing Register
DDR_PCTL_TCKSRX	0x0128	W	0x00000000	t_cksrx Timing Register
DDR_PCTL_TCKE	0x012c	W	0x00000003	t_cke Timing Register
DDR_PCTL_TMOD	0x0130	W	0x00000000	t_mod Timing Register
DDR_PCTL_TRSTL	0x0134	W	0x00000000	Reset Low Timing Register
DDR_PCTL_TZQCL	0x0138	W	0x00000000	t_zqcl Timing Register
DDR_PCTL_TMRR	0x013c	W	0x00000002	t_mrr Timing Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_TCKESR	0x0140	W	0x00000004	t_ckesr Timing Register
DDR_PCTL_TDPD	0x0144	W	0x00000000	t_dpd Timing Register
DDR_PCTL_DTUWACTL	0x0200	W	0x00000000	DTU Write Address Control
DDR_PCTL_DTURACTL	0x0204	W	0x00000000	DTU Read Address Control Register
DDR_PCTL_DTUCFG	0x0208	W	0x00000000	DTU Configuration Control Register
DDR_PCTL_DTUECTL	0x020c	W	0x00000000	DTU Execute Control Register
DDR_PCTL_DTUWD0	0x0210	W	0x00000000	DTU Write Data #0 Register
DDR_PCTL_DTUWD1	0x0214	W	0x00000000	DTU Write Data #1 Register
DDR_PCTL_DTUWD2	0x0218	W	0x00000000	DTU Write Data #2 Register
DDR_PCTL_DTUWD3	0x021c	W	0x00000000	DTU Write Data #3 Register
DDR_PCTL_DTUWDM	0x0220	W	0x00000000	DTU Write Data Mask Register
DDR_PCTL_DTURD0	0x0224	W	0x00000000	DTU Read Data #0 Register
DDR_PCTL_DTURD1	0x0228	W	0x00000000	DTU Read Data #1 Register
DDR_PCTL_DTURD2	0x022c	W	0x00000000	DTU Read Data #2 Register
DDR_PCTL_DTURD3	0x0230	W	0x00000000	DTU Read Data #3 Register
DDR_PCTL_DTULFSRWD	0x0234	W	0x00000000	DTU LFSR Seed for Write Data Generation Register
DDR_PCTL_DTULFSRRD	0x0238	W	0x00000000	DTU LFSR Seed for Read Data Generation Register
DDR_PCTL_DTUEAF	0x023c	W	0x00000000	DTU Error Address FIFO Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITCTRLDELAY	0x0240	W	0x00000002	DFI tctrl_delay Register
DDR_PCTL_DFIODTCFG	0x0244	W	0x00000000	DFI ODT Configuration
DDR_PCTL_DFIODTCFG1	0x0248	W	0x06060000	DFI ODT Timing Configuration 1 (for Latency and Length)
DDR_PCTL_DFIODTRANKMAP	0x024c	W	0x00008421	DFI ODT Rank Mapping
DDR_PCTL_DFITPHYWRDATA	0x0250	W	0x00000001	DFI tphy_wrdata Register
DDR_PCTL_DFITPHYWRLAT	0x0254	W	0x00000001	DFI tphy_wrlat Register
DDR_PCTL_DFITRDDATAEN	0x0260	W	0x00000001	DFI trddata_en Register
DDR_PCTL_DFITPHYRDLAT	0x0264	W	0x0000000f	DFI tphy_rdlat Register
DDR_PCTL_DFITPHYUPDTYPE0	0x0270	W	0x00000010	DFI tphyupd_type0 Register
DDR_PCTL_DFITPHYUPDTYPE1	0x0274	W	0x00000010	DFI tphyupd_type1 Register
DDR_PCTL_DFITPHYUPDTYPE2	0x0278	W	0x00000010	DFI tphyupd_type2 Register
DDR_PCTL_DFITPHYUPDTYPE3	0x027c	W	0x00000010	DFI tphyupd_type3 Register
DDR_PCTL_DFITCTRLUPDMIN	0x0280	W	0x00000010	DFI tctrlupd_min Register
DDR_PCTL_DFITCTRLUPDMAX	0x0284	W	0x00000040	DFI tctrlupd_max Register
DDR_PCTL_DFITCTRLUPDDLY	0x0288	W	0x00000008	DFI tctrlupddly Register
DDR_PCTL_DFIUPDCFG	0x0290	W	0x00000003	DFI Update Configuration Register
DDR_PCTL_DFITREFMSKI	0x0294	W	0x00000000	DFI Masked Refresh Interval

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITCTRLUPDI	0x0298	W	0x00000000	DFI tctrlupd_interval Register
DDR_PCTL_DFITRCFG0	0x02ac	W	0x00000000	DFI Training Configuration 0 Register
DDR_PCTL_DFITRSTAT0	0x02b0	W	0x00000000	DFI Training Status 0 Register
DDR_PCTL_DFITRWRLVLEN	0x02b4	W	0x00000000	DFI Training dfi_wrlvl_en Register
DDR_PCTL_DFITRRDLVLEN	0x02b8	W	0x00000000	DFI Training dfi_rdlvl_en Register
DDR_PCTL_DFITRRDLVLGATEEN	0x02bc	W	0x00000000	DFI Training dfi_rdlvl_gate_en Register
DDR_PCTL_DFISTSTAT0	0x02c0	W	0x00000000	DFI Status Status 0 Register
DDR_PCTL_DFISTCFG0	0x02c4	W	0x00000000	DFI Status Configuration 0 Register
DDR_PCTL_DFISTCFG1	0x02c8	W	0x00000000	DFI Status Configuration 1 Register
DDR_PCTL_DFITDRAMCLKEN	0x02d0	W	0x00000002	DFI tdram_clk_enable Register
DDR_PCTL_DFITDRAMCLKDIS	0x02d4	W	0x00000002	DFI tdram_clk_disable Register
DDR_PCTL_DFISTCFG2	0x02d8	W	0x00000000	DFI Status Configuration 2 Register
DDR_PCTL_DFISTPARCLR	0x02dc	W	0x00000000	DFI Status Parity Clear Register
DDR_PCTL_DFISTPARLOG	0x02e0	W	0x00000000	DFI Status Parity Log Register
DDR_PCTL_DFILPCFG0	0x02f0	W	0x00070000	DFI Low Power Configuration 0 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITRWRLVLRESP0	0x0300	W	0x00000000	DFI Training dfi_wrlvl_resp Status 0 Register
DDR_PCTL_DFITRWRLVLRESP1	0x0304	W	0x00000000	DFI Training dfi_wrlvl_resp Status 1 Register
DDR_PCTL_DFITRWRLVLRESP2	0x0308	W	0x00000000	DFI Training dfi_wrlvl_resp Status 2 Register
DDR_PCTL_DFITRRDLVLRESP0	0x030c	W	0x00000000	DFI Training dfi_rdlvl_resp Status 0 Register
DDR_PCTL_DFITRRDLVLRESP1	0x0310	W	0x00000000	DFI Training dfi_rdlvl_resp Status 1 Register
DDR_PCTL_DFITRRDLVLRESP2	0x0314	W	0x00000000	DFI Training dfi_rdlvl_resp Status 2 Register
DDR_PCTL_DFITRWRLVLDELAY0	0x0318	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 0 Register
DDR_PCTL_DFITRWRLVLDELAY1	0x031c	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 1 Register
DDR_PCTL_DFITRWRLVLDELAY2	0x0320	W	0x00000000	DFI Training dfi_wrlvl_delay Configuration 2 Register
DDR_PCTL_DFITRRDLVLDELAY0	0x0324	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 0 Register
DDR_PCTL_DFITRRDLVLDELAY1	0x0328	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 1 Register
DDR_PCTL_DFITRRDLVLDELAY2	0x032c	W	0x00000000	DFI Training dfi_rdlvl_delay Configuration 2 Register

Name	Offset	Size	Reset Value	Description
DDR_PCTL_DFITRRDLVLGATEDELAY0	0x0330	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 0
DDR_PCTL_DFITRRDLVLGATEDELAY1	0x0334	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 1
DDR_PCTL_DFITRRDLVLGATEDELAY2	0x0338	W	0x00000000	DFI Training dfi_rdlvl_gate_delay Configuration 2
DDR_PCTL_DFITRCMD	0x033c	W	0x00000000	DFI Training Command Register
DDR_PCTL_IPVR	0x03f8	W	0x00000000	IP Version Register
DDR_PCTL_IPTR	0x03fc	W	0x44574300	IP Type Register

Name	Offset	Size	Reset Value	Description
DDR_PHYCTL_RIDR	0x0000	W	0x00100140	Revision Identification Register
DDR_PHYCTL_PIR	0x0004	W	0x00000000	PHY Initialization Register
DDR_PHYCTL_PGCR	0x0008	W	0x01bc2e04	PHY General Configuration Register
DDR_PHYCTL_PGSR	0x000c	W	0x00000000	PHY General Status Register
DDR_PHYCTL_DLLGCR	0x0010	W	0x03737000	DLL General Control Register
DDR_PHYCTL_ACDLLCR	0x0014	W	0x40000000	AC DLL Control Register
DDR_PHYCTL_PTR0	0x0018	W	0x0022af9b	PHY Timing Register 0
DDR_PHYCTL_PTR1	0x001c	W	0x0604111d	PHY Timing Register 1
DDR_PHYCTL_PTR2	0x0020	W	0x042da072	PHY Timing Register 2
DDR_PHYCTL_ACIOCR	0x0024	W	0x33c03812	AC I/O Configuration Register
DDR_PHYCTL_DXCCR	0x0028	W	0x00000800	DATX8 Common Configuration Register
DDR_PHYCTL_DSGCR	0x002c	W	0xfa00001f	DDR System General Configuration Register
DDR_PHYCTL_DCR	0x0030	W	0x0000000b	DRAM Configuration Register

Name	Offset	Size	Reset Value	Description
DDR_PHYCTL_DTPR0	0x0034	W	0x3092666e	DRAM Timing Parameters Register 0
DDR_PHYCTL_DTPR1	0x0038	W	0x09830090	DRAM Timing Parameters Register 1
DDR_PHYCTL_DTPR2	0x003c	W	0x1001a0c8	DRAM Timing Parameters Register 2
DDR_PHYCTL_MR0	0x0040	W	0x00000a52	Mode Register 0
DDR_PHYCTL_MR1	0x0044	W	0x00000000	Mode Register 1
DDR_PHYCTL_MR2	0x0048	W	0x00000000	Mode Register 2
DDR_PHYCTL_MR3	0x004c	W	0x00000000	Mode Register 3
DDR_PHYCTL_ODTCR	0x0050	W	0x00210000	ODT Configuration Register
DDR_PHYCTL_DTAR	0x0054	W	0x00000000	Data Training Address Register
DDR_PHYCTL_DTDR0	0x0058	W	0xdd22ee11	Data Training Data Register 0
DDR_PHYCTL_DTDR1	0x005c	W	0x7788bb44	Data Training Data Register 1
DDR_PHYCTL_DCUAR	0x00c0	W	0x00000000	DCU Address Register
DDR_PHYCTL_DCUDR	0x00c4	W	0x00000000	DCU Data Register
DDR_PHYCTL_DCURR	0x00c8	W	0x00000000	DCU Run Register
DDR_PHYCTL_DCULR	0x00cc	W	0x00000000	DCU Loop Register
DDR_PHYCTL_DCUGCR	0x00d0	W	0x00000000	DCU General Configuration Register
DDR_PHYCTL_DCUTPR	0x00d4	W	0x00000000	DCU Timing Parameters Registers
DDR_PHYCTL_DCUSR0	0x00d8	W	0x00000000	DCU Status Register 0
DDR_PHYCTL_DCUSR1	0x00dc	W	0x00000000	DCU Status Register 1
DDR_PHYCTL_BISTRR	0x0100	W	0x00000000	BIST Run Register
DDR_PHYCTL_BISTMSKR0	0x0104	W	0x00000000	BIST Mask Register 0
DDR_PHYCTL_BISTMSKR1	0x0108	W	0x00000000	BIST Mask Register 1
DDR_PHYCTL_BISTWCR	0x010c	W	0x00000020	BIST Word Count Register
DDR_PHYCTL_BISTLSR	0x0110	W	0x1234abcd	BIST LFSR Seed Register
DDR_PHYCTL_BISTAR0	0x0114	W	0x00000000	BIST Address Register 0
DDR_PHYCTL_BISTAR1	0x0118	W	0x0000000c	BIST Address Register 1
DDR_PHYCTL_BISTAR2	0x011c	W	0x7fffffff	BIST Address Register 2
DDR_PHYCTL_BISTUDPR	0x0120	W	0xffff0000	BIST User Data Pattern Register
DDR_PHYCTL_BISTGSR	0x0124	W	0x00000000	BIST General Status Register

Name	Offset	Size	Reset Value	Description
DDR_PHYCTL_BISTWER	0x0128	W	0x00000000	BIST Word Error Register
DDR_PHYCTL_BISTBER0	0x012c	W	0x00000000	BIST Bit Error Register 0
DDR_PHYCTL_BISTBER1	0x0130	W	0x00000000	BIST Bit Error Register 1
DDR_PHYCTL_BISTBER2	0x0134	W	0x00000000	BIST Bit Error Register 2
DDR_PHYCTL_BISTWCSR	0x0138	W	0x00000000	BIST Word Count Status Register
DDR_PHYCTL_BISTFWR0	0x013c	W	0x00000000	BIST Fail Word Register 0
DDR_PHYCTL_BISTFWR1	0x0140	W	0x00000000	BIST Fail Word Register 1
DDR_PHYCTL_ZQ0CR0	0x0180	W	0x0000014a	ZQ 0 Impedance Control Register 0
DDR_PHYCTL_ZQ0CR1	0x0184	W	0x0000007b	ZQ 0 Impedance Control Register 1
DDR_PHYCTL_ZQ0SR0	0x0188	W	0x00000000	ZQ 0 Impedance Status Register 0
DDR_PHYCTL_ZQ0SR1	0x018c	W	0x00000000	ZQ 0 Impedance Status Register 1
DDR_PHYCTL_ZQ1CR0	0x0190	W	0x0000014a	ZQ 1 Impedance Control Register 0
DDR_PHYCTL_ZQ1CR1	0x0194	W	0x0000007b	ZQ 1 Impedance Control Register 1
DDR_PHYCTL_ZQ1SR0	0x0198	W	0x00000000	ZQ 1 Impedance Status Register 0
DDR_PHYCTL_ZQ1SR1	0x019c	W	0x00000000	ZQ 1 Impedance Status Register 1
DDR_PHYCTL_ZQ2CR0	0x01a0	W	0x0000014a	ZQ 2 Impedance Control Register 0
DDR_PHYCTL_ZQ2CR1	0x01a4	W	0x0000007b	ZQ 2 Impedance Control Register 1
DDR_PHYCTL_ZQ2SR0	0x01a8	W	0x00000000	ZQ 2 Impedance Status Register 0
DDR_PHYCTL_ZQ2SR1	0x01ac	W	0x00000000	ZQ 2 Impedance Status Register 1
DDR_PHYCTL_ZQ3CR0	0x01b0	W	0x0000014a	ZQ 3 Impedance Control Register 0
DDR_PHYCTL_ZQ3CR1	0x01b4	W	0x0000007b	ZQ 3 Impedance Control Register 1
DDR_PHYCTL_ZQ3SR0	0x01b8	W	0x00000000	ZQ 3 Impedance Status Register 0

Name	Offset	Size	Reset Value	Description
DDR_PHYCTL_ZQ3SR1	0x01bc	W	0x00000000	ZQ 3 Impedance Status Register 1
DDR_PHYCTL_DX0GCR	0x01c0	W	0x00000681	DATX8 0 General Configuration Register
DDR_PHYCTL_DX0GSR0	0x01c4	W	0x00000000	DATX8 0 General Status Register 0
DDR_PHYCTL_DX0GSR1	0x01c8	W	0x00000000	DATX8 0 General Status Register 1
DDR_PHYCTL_DX0DLLCR	0x01cc	W	0x40000000	DATX8 0 DLL Control Register
DDR_PHYCTL_DX0DQTR	0x01d0	W	0xffffffff	DATX8 0 DQ Timing Register
DDR_PHYCTL_DX0DQSTR	0x01d4	W	0x3db55000	DATX8 0 DQS Timing Register
DDR_PHYCTL_DX1GCR	0x0200	W	0x00000681	DATX8 1 General Configuration Register
DDR_PHYCTL_DX1GSR0	0x0204	W	0x00000000	DATX8 1 General Status Register 0
DDR_PHYCTL_DX1GSR1	0x0208	W	0x00000000	DATX8 1 General Status Register 1
DDR_PHYCTL_DX1DLLCR	0x020c	W	0x40000000	DATX8 1 DLL Control Register
DDR_PHYCTL_DX1DQTR	0x0210	W	0xffffffff	DATX8 1 DQ Timing Register
DDR_PHYCTL_DX1DQSTR	0x0214	W	0x3db55000	DATX8 1 DQS Timing Register
DDR_PHYCTL_DX2GCR	0x0240	W	0x00000681	DATX8 2 General Configuration Register
DDR_PHYCTL_DX2GSR0	0x0244	W	0x00000000	DATX8 2 General Status Register 0
DDR_PHYCTL_DX2GSR1	0x0248	W	0x00000000	DATX8 2 General Status Register 1
DDR_PHYCTL_DX2DLLCR	0x024c	W	0x40000000	DATX8 2 DLL Control Register
DDR_PHYCTL_DX2DQTR	0x0250	W	0xffffffff	DATX8 2 DQ Timing Register
DDR_PHYCTL_DX2DQSTR	0x0254	W	0x3db55000	DATX8 2 DQS Timing Register
DDR_PHYCTL_DX3GCR	0x0280	W	0x00000681	DATX8 3 General Configuration Register

Name	Offset	Size	Reset Value	Description
DDR_PHYCTL_DX3GSR0	0x0284	W	0x00000000	DATX8 3 General Status Register 0
DDR_PHYCTL_DX3GSR1	0x0288	W	0x00000000	DATX8 3 General Status Register 1
DDR_PHYCTL_DX3DLLCR	0x028c	W	0x40000000	DATX8 3 DLL Control Register
DDR_PHYCTL_DX3DQTR	0x0290	W	0xffffffff	DATX8 3 DQ Timing Register
DDR_PHYCTL_DX3DQSTR	0x0294	W	0x3db55000	DATX8 3 DQS Timing Register

13.5.2 Detail Registers Description

DDR_PCTL_SCFG
Address: Operational Base + offset (0x0000)
State Configuration Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:8	RW	0x3	bbflags_timing The n_bbflags is a NIF output vector which provides combined information about the status of each memory bank. The de-assertion is based on whenprecharge,activates,reads/writesare scheduled by the TCU block. It may be possible to de-assert n_bbflags earlier than calculated by the TCU block. Programming bbflags_timing is used to achieve this.The maximum recommended value is: PCTL_TCU_SED_P - TRP.t_rp. The programmed value is the maximum number of "early" cycles that n_bbflags maybe de-asserted. The actual achieved de-assertion depends on the traffic profile. In 1:2 mode the maximum allowed programmable value is 4'b0111 In 1:1 mode the value can be 4'b1111
7:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	hw_low_power_en Enables the hardware low-power interface. Allows the system to request via hardware (c_sysreq input) to enter the memories into Self-Refresh. The handshaking between the request and acknowledge hardware low power signals (c_sysreq and c_sysack, respectively) is always performed, but the PCTL response depends on the value set on this register field and by the value driven on the c_active_in input pin. 1'b0 = Disabled. Requests are always denied and PCTL is unaffected by c_sysreq 1'b1 = Enabled. Requests are accepted or denied, depending on the current operational state of PCTL and on the value of c_active_in.

DDR_PCTL_SCTL
Address: Operational Base + offset (0x0004)
Operational State Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x0	state_cmd Issues an operational state transition request to the PCTL. 3'b000 = INIT (move to Init_mem from Config) 3'b001 = CFG (move to Config from Init_mem or Access) 3'b010 = GO (move to Access from Config) 3'b011 = SLEEP (move to Low_power from Access) 3'b100 = WAKEUP (move to Access from Low_power) Others = Reserved

DDR_PCTL_STAT
Address: Operational Base + offset (0x0008)
Operational State Status Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:4	RO	0x0	lp_trig Reports the status of what triggered an entry to Low_power state. Is only set if in Low_power state. The individual bits report the following: - lp_trig[2]: Software driven due to SCTL.state_cmd==SLEEP. - lp_trig[1]: Hardware driven due to Hardware Low Power Interface. - lp_trig[0]: Hardware driven due to Auto Self Refresh (MCFG1.sr_idle>0). Note, if more than one trigger happens at the exact same time, more than one bit of lp_trig may be asserted high.
3	RO	0x0	reserved
2:0	RO	0x0	ctl_stat Returns the current operational state of the PCTL. 3'b000 = Init_mem 3'b001 = Config 3'b010 = Config_req 3'b011 = Access 3'b100 = Access_req 3'b101 = Low_power 3'b110 = Low_power_entry_req 3'b111 = Low_power_exit_req Others = Reserved

DDR_PCTL_INTRSTAT

Address: Operational Base + offset (0x000c)

Interrupt Status Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	parity_intr Indicates that a DFI parity error has been detected 1'b0 = No error 1'b1 = Parity error
0	RO	0x0	ecc_intr Indicates that an ECC error has been detected 1'b0 = No error 1'b1 = Parity error

DDR_PCTL_MCMD
Address: Operational Base + offset (0x0040)
Memory Command Register

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	start_cmd Start command. When this bit is set to 1, the command operation defined in the cmd_opcode field is started. This bit is automatically cleared by the PCTL after the command is finished. The application can poll this bit to determine when PCTL is ready to accept another command. This bit cannot be cleared to 1'b0 by software.
30:28	RO	0x0	reserved
27:24	RW	0x0	cmd_add_del Set the additional delay associated with each command to 2^n internal timers clock cycles, where n is the bit field value. If n=0, the delay is 0. Max value is n=10.
23:20	RW	0x1	rank_sel Rank select for the command to be executed. 4'b0001 = Rank 0 4'b0010 = Rank 1 4'b0100 = Rank 2 4'b1000 = Rank 3 4'b0000 = Reserved Multiple 1'b1s in rank_sel mean multiple ranks are selected, which is useful broadcasting commands in parallel to multiple ranks during initialization and configuration of the memories. If MCMD.cmd_opcode=RSTL, all ranks should be selected as it cannot be performed to individual ranks

Bit	Attr	Reset Value	Description
19:17	RW	0x0	bank_addr Mode Register address driven on the memory bank address bits, BA1, BA0, during a Mode Register Set operation, defined by cmd_opcode=MRS. For other values of cmd_opcode, this field is ignored. 3'b000 = MR0 (MR in DDR2) 3'b001 = MR1 (EMR in DDR2) 3'b010 = MR2 (EMR(2) in DDR2) 3'b011 = MR3 (EMR(3) in DDR2) Others = Reserved
16:4	RW	0x0000	cmd_addr Mode Register value driven on the memory address bits, A12 to A0, during a Mode Register Set operation defined by cmd_opcode=MRS. For other values of cmd_opcode this field is ignored. Refer to the memory specification for the correct settings of the various bits of this field during a MRS operation. If LPDDR2, this fields is merged into bank_addr - lpddr2_addr

Bit	Attr	Reset Value	Description
3:0	RW	0x0	cmd_opcode Command to be issued to the memory. 4'b000 = Deselect. This is only used for timing purposes, no actual direct Deselect command is passed to the memories. 4'b0001 = Precharge All (PREA) 4'b0010 = Refresh (REF) 4'b0011 = Mode Register Set (MRS) - is MRW in LPDDR2, MRS otherwise 4'b0100 = ZQ Calibration Short (ZQCS, only applies to LPDDR2/DDR3) 4'b0101 = ZQ Calibration Long (ZQCL, only applies to LPDDR2/DDR3) 4'b0110 = Software Driven Reset (RSTL, only applies to DDR3) 4'b0111 = Reserved 4'b1000 - Mode Register Read (MRR) - is MRR in LPDDR2, is SRR in mDRR and is MPR in DDR3 4'b1001 - Deep Power Down Entry (DPDE, only applies to mDDR/LPDDR2) Others - Reserved

DDR_PCTL_POWCTL
Address: Operational Base + offset (0x0044)
Power Up Control Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	R/WSC	0x0	power_up_start Start the memory power up sequence. When this bit is set to 1'b1, PCTL starts the CKE and RESET# power up sequence to the memories. This bit is automatically cleared by PCTL after the sequence is completed. This bit cannot be cleared to 1'b0 by software.

DDR_PCTL_POWSTAT
Address: Operational Base + offset (0x0048)
Power Up Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RO	0x0	power_up_done Returns the status of the memory power-up sequence. 1'b0 = Power-up sequence has not been performed. 1'b1 = Power-up sequence has been performed.

DDR_PCTL_CMDTSTAT
Address: Operational Base + offset (0x004c)
Command Timers Status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	cmd_tstat Returns the status of the timers for memory commands. This ANDs all the command timers together. 1'b0 = One or more command timers has not expired. 1'b1 = All command timers have expired.

DDR_PCTL_CMDTSTATEN
Address: Operational Base + offset (0x0050)
Command Timers Status Enable Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	cmd_tstat_en Enables the generation of the status of the timers for memory commands. Is enabled before CMDTSTAT register is read. 1'b0 - Disabled 1'b1 - Enabled

DDR_PCTL_MRRCFG0
Address: Operational Base + offset (0x0060)
Mode Register Read Configuration 0

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	mrr_byte_sel Selects which byte's data to store when performing an MRR command via MCMD. LegalValues: 0 .. 8

DDR_PCTL_MRRSTAT0
Address: Operational Base + offset (0x0064)
Mode Register Read Status 0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat3 MRR/MPR read data beat 3
23:16	RO	0x00	mrrstat_beat2 MRR/MPR read data beat 2
15:8	RO	0x00	mrrstat_beat1 MRR/MPR read data beat 1
7:0	RO	0x00	mrrstat_beat0 MRR/MPR read data beat 0

DDR_PCTL_MRRSTAT1
Address: Operational Base + offset (0x0068)
Mode Register Read Status 0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	mrrstat_beat7 MRR/MPR read data beat 7
23:16	RO	0x00	mrrstat_beat6 MRR/MPR read data beat 6
15:8	RO	0x00	mrrstat_beat5 MRR/MPR read data beat 5
7:0	RO	0x00	mrrstat_beat4 MRR/MPR read data beat 4

DDR_PCTL_MCFG
Address: Operational Base + offset (0x0080)
Memory Configuration Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	mddr_lpddr2_clock_stop_idle Clock stop idle period in n_clk cycles. Memories are placed into clock stop mode if the NIF is idle for mddr_lpddr2_clkstop_idle n_clk cycles. The automatic clock stop function is disabled when mddr_lpddr2_clkstop_idle=0. Clock stop mode is only applicable in mDDR/LPDDR2.

Bit	Attr	Reset Value	Description
23:22	RW	0x0	mddr_lpddr2_en mDDR/LPDDR2 Enable. Enables support for mDDR or LPDDR2. 2'b00 = mDDR/LPDDR2 Disabled 2'b10 = mDDR Enabled 2'b11 = LPDDR2 Enabled Others= Reserved.
21:20	RW	0x0	mddr_lpddr2_bl mDDR/LPDDR2 Burst Length. The BL setting must be consistent with the value programmed into the BL field of MR. 2'b00 = BL2, Burst length of 2 (MR.BL=3'b001, mDDR only) 2'b01 = BL4, Burst length of 4 (MR.BL=3'b010, for mDDR and LPDDR2) 2'b10 = BL8, Burst length of 8 (MR.BL=3'b011, for mDDR and LPDDR2) 2'b11 = BL16, Burst length of 16 (MR.BL=3'b100, for mDDR and LPDDR2) This value is effective only if MCFG.mddr_lpddr2_en[1]=1'b1. Otherwise, MCFG.mem_bl is used to define PCTL's Burst Length (for DDR2/DDR3).
19:18	RW	0x1	tfaw_cfg Sets tFAW to be 4, 5 or 6 times tRRD. 2'b00 = set tFAW=4*tRRD 2'b01 = set tFAW=5*tRRD 2'b10 = set tFAW=6*tRRD
17	RW	0x0	pd_exit_mode Selects the mode for Power Down Exit. For DDR2/DDR3, the power down exit mode setting in PCTL must be consistent with the value programmed into the power down exit mode bit of MR0. For mDDR/LPDDR2, only fast exit mode is valid. 1'b0 = slow exit 1'b1 = fast exit
16	RW	0x0	pd_type Sets the Power down type. 1'b0 = Precharge Power Down 1'b1 = Active Power Down

Bit	Attr	Reset Value	Description
15:8	RW	0x00	pd_idle Power-down idle period in n_clk cycles. Memories are placed into power-down mode if the NIF is idle for pd_idle n_clk cycles. The automatic power down function is disabled when pd_idle=0.
7	RO	0x0	reserved
6	RW	0x0	lpddr2_s4 Enables LPDDR2-S4 support. 1'b0 = LPDDR2-S4 disabled (LPDDR2-S2 enabled) 1'b1 = LPDDR2-S4 enabled
5	RW	0x1	ddr3_en Select DDR2 or DDR3 protocol. Ignored, if mDDR or LPDDR2 support is enabled. 1'b0 = DDR2 Protocol Rules 1'b1 = DDR3 Protocol Rules
4	RW	0x0	stagger_cs For multi-rank commands from the DCU, stagger the assertion of CS_N to odd and even ranks by one n_clk cycle. This is useful when using RDIMMs, when multi-rank commands may be interpreted as writes to control words in the register chip. 1'b0 = Do not stagger CS_N 1'b1 = Stagger CS_N
3	RW	0x0	two_t_en Enables 2T timing for memory commands. 1'b0= Disabled 1'b1 = Enabled
2	RW	0x0	bl8int_en Setting this bit enables the BL8 interrupt function of DDR2. This is the capability to early terminate a BL8 after only 4 DDR beats by issuing the next command two cycles earlier. This functionality is only available for DDR2 memories and this setting is ignored for mDDR/LPDDR2 and DDR3. 1'b0 = Disabled 1'b1 = Enabled

Bit	Attr	Reset Value	Description
1	RW	0x0	cke_or_en This bit is intended to be set for 4-rank RDIMMs, which have a 2-bit CKE input. If set, dfi_cke[0] is asserted to enable either of the even ranks (0 and 2), while dfi_cke[1] is asserted to enable either of the odd ranks (1 and 3). dfi_cke[3:2] are inactive (0) 1'b0: Disabled 1'b1: Enabled
0	RW	0x0	mem_bl DDR Burst Length. The BL setting in DDR2 / DDR3 must be consistent with the value programmed into the BL field of MR0. 1'b0 = BL4, Burst length of 4 (MR0.BL=3'b010, DDR2 only) 1'b1 = BL8, Burst length of 8 (MR0.BL=3'b011 for DDR2, MR0.BL=2'b00 for DDR3)

DDR_PCTL_PPCFG
Address: Operational Base + offset (0x0084)
Partially Populated Memories Configuration Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:1	RW	0x00	rpmem_dis Reduced Population Disable bits. Setting these bits disables the corresponding NIF/DDR data lanes from writing or reading data. Lane 0 is always present, hence only 8 bits are required for the remaining lanes including the ECC lane. In 1:2 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[63:32], bit 1 [95:64] etc. In 1:1 mode bit 0 of rpmem_dis covers n_wdata/n_rdata/m_ctl_d/m_phy_q[31:16], bit 2 [47:32] etc. There are no restrictions on which byte lanes can be disabled, other than byte lane 0 is required. Gaps between enabled byte lanes are allowed For each bit: 1'b0 = lane exists 1'b1 = lane is disabled

Bit	Attr	Reset Value	Description
0	RW	0x0	ppmem_en Partially Population Enable bit. Setting this bit enables the partial population of external memories where the entire application bus is routed to a reduced size memory system. The lower half of the SDRAM data bus, bit 0 up to bit PCTL_M_DW/2-1, is the active portion when Partially Populated memories are enabled. 1'b0 = Disabled 1'b1 = Enabled

DDR_PCTL_MSTAT
Address: Operational Base + offset (0x0088)
Memory Status Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	self_refresh Indicates if PCTL, through auto self refresh, has placed the memories in Self Refresh. 1'b0 = Memory is not in Self Refresh 1'b1 = Memory is in Self Refresh
1	RO	0x0	clock_stop Indicates if PCTL has placed the memories in Clock Stop. 1'b0 = Memory is not in Clock Stop 1'b1 = Memory is in Clock Stop
0	RO	0x0	power_down Indicates if PCTL has placed the memories in Power Down. 1'b0 = Memory is not in Power Down 1'b1 = Memory is in Power-Down

DDR_PCTL_LPDDR2ZQCFG
Address: Operational Base + offset (0x008c)
LPDDR2 ZQ Configuration Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:24	RW	0xab	zqcl_op Value to drive on memory address bits [19:12] for an automatic hardware generated ZQCL command (LPDDR2). Corresponds to OP7 .. OP0 of Mode Register Write (MRW) command which is used to send ZQCL command to memory.
23:16	RW	0x0a	zqcl_ma Value to drive on memory address bits [11:4] for an automatic hardware generated ZQCL command (LPDDR2). Corresponds to MA7 .. MA0 of Mode Register Write (MRW) command which is used to send ZQCL command to memory.
15:8	RW	0x56	zqcs_op Value to drive on memory address bits [19:12] for an automatic hardware generated ZQCS command (LPDDR2). Corresponds to OP7 .. OP0 of Mode Register Write (MRW) command which is used to send ZQCS command to memory.
7:0	RW	0x0a	zqcs_ma Value to drive on memory address bits [11:4] for an automatic hardware generated ZQCS command (LPDDR2). Corresponds to MA7 .. MA0 of Mode Register Write (MRW) command which is used to send ZQCS command to memory.

DDR_PCTL_MCFG1
Address: Operational Base + offset (0x0090)
Memory Configuration 1 Register

Bit	Attr	Reset Value	Description
31	RW	0x0	hw_exit_idle_en When this bit is programmed to 1'b1 the c_active_in pin can be used to exit from the automatic clock stop , power down or self-refresh modes.
30:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:16	RW	0x00	hw_idle Hardware idle period. The c_active output is driven high if the NIF is idle in Access state for hw_idle * 32 * n_clk cycles. The hardware idle function is disabled when hw_idle=0.
15:8	RO	0x0	reserved
7:0	RW	0x00	sr_idle Self Refresh idle period. Memories are placed into Self-Refresh mode if the NIF is idle in Access state for sr_idle * 32 * n_clk cycles. The automatic self refresh function is disabled when sr_idle=0.

DDR_PCTL_DTUPDES
Address: Operational Base + offset (0x0094)
DTU Status Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RO	0x0	dtu_rd_missing Indicates if one or more read beats of data did not return from memory.
12:9	RO	0x0	dtu_eaffl Indicates the number of entries in the FIFO that is holding the log of error addresses for data comparison
8	RO	0x0	dtu_random_error Indicates that the random data generated had some failures when written and read to the memories
7	RO	0x0	dtu_err_b7 Detected at least 1 bit error for bit 7 in the programmable data buffers
6	RO	0x0	dtu_err_b6 Detected at least 1 bit error for bit 6 in the programmable data buffers
5	RO	0x0	dtu_err_b5 Detected at least 1 bit error for bit 5 in the programmable data buffers
4	RO	0x0	dtu_err_b4 Detected at least 1 bit error for bit 4 in the programmable data buffers

Bit	Attr	Reset Value	Description
3	RO	0x0	dtu_err_b3 Detected at least 1 bit error for bit 3 in the programmable data buffers
2	RO	0x0	dtu_err_b2 Detected at least 1 bit error for bit 2 in the programmable data buffers
1	RO	0x0	dtu_err_b1 Detected at least 1 bit error for bit 1 in the programmable data buffers
0	RO	0x0	dtu_err_b0 Detected at least 1 bit error for bit 0 in the programmable data buffers

DDR_PCTL_DTUNA
Address: Operational Base + offset (0x0098)
DTU Number of Addresses Created Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_address Indicates the number of addresses that were created on the NIF interface during random data generation.

DDR_PCTL_DTUNE
Address: Operational Base + offset (0x009c)
DTU Number of Errors Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dtu_num_errors Indicates the number of errors that were detected on the readback of the NIF data during random data generation.

DDR_PCTL_DTUPRD0
Address: Operational Base + offset (0x00a0)
DTU Parallel Read 0 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_1 Allows all the bit ones from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	dtu_allbits_0 Allows all the bit zeros from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD1
Address: Operational Base + offset (0x00a4)
DTU Parallel Read 1 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_3 Allows all the bit threes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_2 Allows all the bit twos from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD2
Address: Operational Base + offset (0x00a8)
DTU Parallel Read 2 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_5 Allows all the bit fives from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_4 Allows all the bit fours from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUPRD3

Address: Operational Base + offset (0x00ac)
DTU Parallel Read 3 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	dtu_allbits_7 Allows all the bit sevens from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.
15:0	RO	0x0000	dtu_allbits_6 Allows all the bit sixes from each of the 16 received read bytes to be read in parallel. Used as part of read data eye training where a transition is required to be monitored to train the eye.

DDR_PCTL_DTUAWDT
Address: Operational Base + offset (0x00b0)
DTU Address Width Register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:9	RW	0x1	number_ranks Number of supported memory ranks. 2'b00 = 1 rank 2'b01 = 2 ranks 2'b10 = 3 ranks 2'b11 = 4 ranks
8	RO	0x0	reserved
7:6	RW	0x2	row_addr_width Width of the memory row address bits. 2'b00 = 13 bits wide 2'b01 = 14 bits wide 2'b10 = 15 bits wide 2'b11 = 16 bits wide
5	RO	0x0	reserved
4:3	RW	0x2	bank_addr_width Width of the memory bank address bits. 2'b00 = 2 bits wide (4 banks) 2'b01 = 3 bits wide (8 banks) Others = Reserved
2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	column_addr_width Width of the memory column address bits. 2'b00 = 7 bits wide 2'b01 = 8 bits wide 2'b10 = 9 bits wide 2'b11 = 10 bits wide

DDR_PCTL_TOGCNT1U
Address: Operational Base + offset (0x00c0)
Toggle Counter 1us Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x064	toggle_counter_1u The number of internal timers clock cycles

DDR_PCTL_TINIT
Address: Operational Base + offset (0x00c4)
t_init Timing Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x0c8	t_init Defines the time period (in us) to hold dfi_cke and dfi_reset_n stable during the memory power up sequence. The value programmed must correspond to at least 200us. The actual time period defined is TINIT * TOGCNT1U * internal timers clock .period

DDR_PCTL_TRSTH
Address: Operational Base + offset (0x00c8)
t_rsth Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:0	RW	0x000	t_rsth Defines the time period (in us) to hold the dfi_reset_n signal high after it is de-asserted during the DDR3 Power Up/Reset sequence. The value programmed for DDR3 must correspond to minimum 500us of delay. For mDDR and DDR2, this register should be programmed to 0.The actual time period defined is TRSTH * TOGCNT1U * internal timers clock period.

DDR_PCTL_TOGCNT100N
Address: Operational Base + offset (0x00cc)
Toggle Counter 100ns

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x01	toggle_counter_100n The number of internal timers clock cycles.

DDR_PCTL_TREFI
Address: Operational Base + offset (0x00d0)
t_refi Timing Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x01	t_refi Defines the time period (in 100ns units) of the Refresh interval. The actual time period defined is TREFI * TOGCNT100N * internal timers clock period.

DDR_PCTL_TMRD
Address: Operational Base + offset (0x00d4)
t_mrd Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x1	t_mrd Mode Register Set command cycle time in memory clock cycles. mDDR: Time from MRS to any valid command. LPDDR2: Time from MRS (MRW) to any valid command. DDR2: Time from MRS to any valid command. DDR3: Time from MRS to MRS command. mDDR Legal Values: 2 LPDDR2 Legal Values: 5 DDR2 Legal Values: 2..3 DDR3 Legal Values: 2..4

DDR_PCTL_TRFC
Address: Operational Base + offset (0x00d8)

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x001	t_rfc Refresh to Active/Refresh command time in memory clock cycles. mDDR Legal Values: 7..28 LPDDR2 Legal Values: 15..112 DDR2 Legal Values: 15..131 DDR3 Legal Values: 36.. 374

DDR_PCTL_TRP
Address: Operational Base + offset (0x00dc)
t_trp Timing Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RW	0x1	prea_extra Additional cycles required for a Precharge All (PREA) command - in addition to t_rp. In terms of memory clock cycles mDDR Value: 0 LPDDR2 Value: Value that corresponds (tRPab -tRPpb). Rounded up in terms of memory clock cycles. Values can be 0, 1, 2. DDR2 Value: 1 if 8 Banks, 0 otherwise DDR3 Value: 0
15:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x6	t_rp Precharge period in memory clock cycles. For LPDDR2, this should be set to TRPpb. mDDR Legal Values: 2..3 LPDDR2 Legal Values: 3..13 DDR2 Legal Values: 3..7 DDR3 Legal Values: 5..14

DDR_PCTL_TRTW
Address: Operational Base + offset (0x00e0)
t_rtw Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	t_rtw Read to Write turnaround time in memory clock cycles. mDDR Legal Values: 3..11 LPDDR2 Legal Values: 1..11 DDR2 Legal Values: 2..10 DDR3 Legal Values: 2..10

DDR_PCTL_TAL
Address: Operational Base + offset (0x00e4)
AL Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	t_al Additive Latency in memory clock cycles. For DDR2 this must match the value programmed into the AL field of MR1. For DDR3 this must be 0, CL-1, CL-2 depending weather the AL value in MR1 is 0,1, or 2 respectively. CL is the CAS latency programmed into MR0. For mDDR and LPDDR2, there is no AL field in the mode registers, and this setting should be set to 0 mDDR Legal Values: 0 LPDDR2 Legal Values: 0 DDR2 Legal Values: AL DDR3 Legal Values: 0, CL-1, CL-2 (depending on AL=0,1,2 in MR1)

DDR_PCTL_TCL
Address: Operational Base + offset (0x00e8)
CL Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_cl CAS Latency in memory clock cycles. If mDDR/DDR2/DDR3, the PCTL setting must match the value programmed into the CL field of MR0. If LPDDR2, the PCTL setting must match RL (ReadLatency), where RL is the value programmed into the "RL & W" field of MR2 mDDR/DDR2/3 Legal Value: CL LPDDR2 Legal Value: RL

DDR_PCTL_TCWL
Address: Operational Base + offset (0x00ec)
CWL Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_cwl CAS Write Latency in memory clock cycles. For mDDR, the setting must be 1. For LPDDR2 the setting must match WL (Write Latency), where WL is the value programmed into the "RL & WL" field of MR2. For DDR2 the setting must match CL-1, where CL is the value programmed into the CL field of MR0. For DDR3, the setting must match the value programmed in the memory CWL field of MR2. mDDR Legal Value: 1 LPDDR2 Legal Values: WL DDR2 Legal Value: CL-1 DDR3 Legal Value: CWL

DDR_PCTL_TRAS
Address: Operational Base + offset (0x00f0)
t_ras Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x10	t_ras Activate to Precharge command time in memory clock cycles. mDDR Legal Values: 4..8 LPDDR2 Legal Values: 7..23 DDR2 Legal Values: 8..24 DDR3 Legal Values: 15..38

DDR_PCTL_TRC
Address: Operational Base + offset (0x00f4)
t_rc Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x16	t_rc Row Cycle time in memory clock cycles. Specifies the minimum Activate to Activate distance for accesses to same bank. mDDR Legal Values: 5..11 LPDDR2 Legal Values: 10..36 DDR2 Legal Values: 11..31 DDR3 Legal Values: 20..52

DDR_PCTL_TRCD
Address: Operational Base + offset (0x00f8)
t_rcd Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x6	t_rcd Row to Column delay in memory clock cycles. Specifies the minimum Activate to Column distance. mDDR Legal Values: 2..3 LPDDR2 Legal Values: 3..13 DDR2 Legal Values: 3..7 DDR3 Legal Values: 5..14

DDR_PCTL_TRRD
Address: Operational Base + offset (0x00fc)
t_rrd Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x4	t_rrd Row-to-Row delay in memory clock cycles. Specifies the minimum Activate-to-Activate distance for consecutive accesses to different banks in the same rank. mDDR Legal Values: 1..2 LPDDR2 Legal Values: 2..6 DDR2 Legal Values: 2..6 DDR3 Legal Values: 4..8

DDR_PCTL_TRTP
Address: Operational Base + offset (0x0100)
t_rtp Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x3	t_rtp Read to Precharge time in memory clock cycles. Specifies the minimum distance Read to Precharge for consecutive accesses to same bank. mDDR Value: 0 LPDDR2 Legal Values: 2..4 DDR2 Legal Values: 2..4 DDR3 Legal Values: 3..8

DDR_PCTL_TWR
Address: Operational Base + offset (0x0104)
t_wr Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x06	t_wr Write recovery time in memory clock cycles. When using close page the PCTL setting must be consistent with the WR field setting of MR0. mDDR Legal Values: 2..3 LPDDR2 Legal Values: 3..8 DDR2 Legal Values: 3..8 DDR3 Legal Values: 6..16

DDR_PCTL_TWTR
Address: Operational Base + offset (0x0108)
t_wtr Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_wtr Write to Read turnaround time, in memory clock cycles. mDDR Legal Values: 1..2 LPDDR2 Legal Values: 2..4 DDR2 Legal Values: 2..4 DDR3 Legal Values: 3..8

DDR_PCTL_TEXSR
Address: Operational Base + offset (0x010c)
t_exsr Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x001	t_exsr Exit Self Refresh to first valid command delay, in memory clock cycles. For mDDR, this should be programmed to match tXSR. For LPDDR2, this should be programmed to match tXSR. For DDR2, this should be programmed to match tXSRD (SRE to read-related command) as defined by the memory devicespecification. For DDR3, this should be programmed to match tXSDLL (SRE to a command requiring DLL locked) as defined by the memory device specification. mDDR Legal Values: 17..40 LPDDR2 Legal Values: 17..117 DDR2 Typical Value: 200 DDR3 Typical Value: 512

DDR_PCTL_TXP
Address: Operational Base + offset (0x0110)
t_xp Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x1	t_xp Exit Power Down to first valid command delay when DLL is on (fast exit), measured in memory clock cycles. Legal Values: 1..7

DDR_PCTL_TXPDLL
Address: Operational Base + offset (0x0114)
t_xpdll Timing Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	t_xpdll Exit Power Down to first valid command delay when DLL is off (slow exit), measured in memory clock cycles. mDDR/LPDDR2 Value: 0 DDR2/DDR3 Legal Values: 3..63

DDR_PCTL_TZQCS
Address: Operational Base + offset (0x0118)
t_zqcs Timing Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_zqcs SDRAM ZQ Calibration Short period, in memory clock cycles. Should be programmed to match the tZQCS timing value as defined in the memory specification. mDDR Value: 0 LPDDR2 Legal Values: 15..48 DDR2 Value: 0 DDR3 Typical Value: 64

DDR_PCTL_TZQCSI
Address: Operational Base + offset (0x011c)
t_zqcsi Timing Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	t_zqcsi SDRAM ZQCS interval, measured in Refresh interval units. The total time period defined is TZQCSI*TREFI * TOGCNT100N * internal timers clock period. Programming a value of 0 in t_zqcsi disables the auto-ZQCS functionality in PCTL. mDDR Value: 0 LPDDR2 Legal Values: 0..4294967295 DDR2 Value: 0 DDR3 Legal Values: 0..4294967295

DDR_PCTL_TDQS
Address: Operational Base + offset (0x0120)
t_dqs Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x1	t_dqs Additional data turnaround time in memory clock cycles for accesses to different ranks. Used to increase the distance between column commands to different ranks, allowing more tolerance as the driver source changes on the bidirectional DQS and/or DQ signals. mDDR Legal Values: 1..7 LPDDR2 Legal Values: 1..7 DDR2 Legal Values: 1..7 DDR3 Legal Values: 1..7

DDR_PCTL_TCKSRE
Address: Operational Base + offset (0x0124)
t_cksre Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	t_cksre In DDR3, this is the time after Self Refresh Entry that CKE is held high before going low. In memory clock cycles. Specifies the clock disable delay after SRE. This should be programmed to match the greatest value between 10ns and 5 memory clock periods. mDDR Value: 0 LPDDR2 Value: 0 DDR2 Value: 0 DDR3 Legal Values: 5..15

DDR_PCTL_TCKSRX
Address: Operational Base + offset (0x0128)
t_cksrx Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_cksrx In DDR3, this is the time (before Self Refresh Exit) that CKE is maintained high before issuing SRX. In memory clock cycles. Specifies the clock stable time before SRX. This should be programmed to match the greatest value between 10ns and 5 memory clock periods. mDDR Value: 0 LPDDR2 Value: 0 DDR2 Value: 0 DDR3 Legal Values: 5..15

DDR_PCTL_TCKE
Address: Operational Base + offset (0x012c)
t_cke Timing Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	RW	0x3	t_cke CKE minimum pulse width in memory clock cycles. mDDR Legal Value: 2 LPDDR2 Legal Values: 3 DDR2 Legal Value: 3 DDR3 Legal Values: 3..6

DDR_PCTL_TMOD
Address: Operational Base + offset (0x0130)
t_mod Timing Register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x00	t_mod In DDR3 mode, this is the time from MRS to any valid non-MRS command (except DESELECT or NOP) in memory clock cycles. mDDR Value: 0 LPDDR2 Value: 0 DDR2 Value: 0 DDR3 Legal Values: 0..31

DDR_PCTL_TRSTL
Address: Operational Base + offset (0x0134)
Reset Low Timing Register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:0	RW	0x00	t_rstl Memory Reset Low time, in memory clock cycles. Defines the time period to hold dfi_reset_n signal low during a software driven DDR3 Reset Operation. The value programmed must correspond to at least 100ns of delay. mDDR Value: 0 LPDDR2 Value: 0 DDR2 Value: 0 DDR3 Legal Values: 1..127

DDR_PCTL_TZQCL
Address: Operational Base + offset (0x0138)
t_zqcl Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9:0	RW	0x000	t_zqcl SDRAM ZQ Calibration Long period in memory clock cycles. If LPDDR2, should be programmed to tZQCL. If DDR3, should be programmed to match the memory tZQinit timing value for the first ZQCL command during memory initialization; should be programmed to match tZQoper timing value after reset and initialization. mDDR Value: 0 LPDDR2 Legal Values: 60..192 DDR2 Value: 0 DDR3 Legal Values: 0..1023

DDR_PCTL_TMRR
Address: Operational Base + offset (0x013c)
t_mrr Timing Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x02	t_mrr Time for a Mode Register Read (MRR command from MCMD).

DDR_PCTL_TCKESR
Address: Operational Base + offset (0x0140)
t_ckesr Timing Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x4	t_ckesr Minimum CKE low width for Self Refresh entry to exit timing in memory clock cycles. Recommended settings: - mDDR : t_ckesr = 0 - LPDDR2 : t_ckesr = tCKESR setting from memories, rounded up in terms of memory cycles. - DDR2 : t_ckesr = 0 - DDR3 : t_ckesr = t_cke + 1 mDDR Value: 0 LPDDR2 Legal Values: 3..8 DDR2 Value: 0 DDR3 Legal Values: 4..7

DDR_PCTL_TDPD
Address: Operational Base + offset (0x0144)
t_dpd Timing Register

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RW	0x000	t_dpd Minimum Deep Power Down time. Is in terms of us. When a MCMD.DPDE command occurs, TDPD time is waited before MCMD.start_cmd can be cleared. MCMD_cmd_add_del (if any) does not start until TDPD has completed. This ensures TDPD requirement for the memory is not violated. The actual time period defined is TDPD* TOGCNT1U * internal timers clock period. Only applies for mDDR and LPDDR2 as Deep Power Down (DPD) is only valid for these memory types. For mDDR, tDPD=0, while for LPDDR2, tDPD=500 us. For LPDDR2, if 500 us is waited externally by system, then set tDPD=0. mDDR Value: 0 LPDDR2 Legal Values: 0 or 500 DDR2 Legal Value: 0 DDR3 Legal Values: 0

DDR_PCTL_DTUWACTL
Address: Operational Base + offset (0x0200)
DTU Write Address Control

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_wr_rank Write rank to where data is to be targeted
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_wr_row Write row to where data is to be targeted
12:10	RW	0x0	dtu_wr_bank Write bank to where data is to be targeted
9:0	RW	0x000	dtu_wr_col FWrite column to where data is to be targeted

DDR_PCTL_DTURACTL
Address: Operational Base + offset (0x0204)
DTU Read Address Control Register

Bit	Attr	Reset Value	Description
31:30	RW	0x0	dtu_rd_rank Read rank from where data comes
29	RO	0x0	reserved
28:13	RW	0x0000	dtu_rd_row Read row from where data comes
12:10	RW	0x0	dtu_rd_bank Read bank from where data comes
9:0	RW	0x000	dtu_rd_col Read column from where data comes

DDR_PCTL_DTUCFG
Address: Operational Base + offset (0x0208)
DTU Configuration Control Register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:16	RW	0x00	dtu_row_increments Number of times to increment the row address when generating random data, up to a maximum of 127 times.
15	RW	0x0	dtu_wr_multi_rd When set puts the DTU into write once multiple reads mode.
14	RW	0x0	dtu_data_mask_en Controls whether random generated data masks are transmitted. Unless enabled all data bytes are written to memory and expected to be read from memory.
13:10	RW	0x0	dtu_target_lane Selects one of the byte lanes for data comparison into the programmable read data buffer.
9	RW	0x0	dtu_generate_random Generate transfers using random data, otherwise generate transfers from the programmable write data buffers.
8	RW	0x0	dtu_incr_banks When the column address rolls over increment the bank address until we reach and conclude bank 7.

Bit	Attr	Reset Value	Description
7	RW	0x0	dtu_incr_cols Increment the column address until we saturate. Return to zero if DTUCFG.dtu_incr_banks is set to 1 and we are not at bank 7.
6:1	RW	0x00	dtu_nalen Length of the NIF transfer sequence that is passed through the PCTL for each created address.
0	RW	0x0	dtu_enable When set, allows the DTU module to take ownership of the NIF interface: 1: DTU enabled 0: DTU disabled

DDR_PCTL_DTUECTL
Address: Operational Base + offset (0x020c)
DTU Execute Control Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	R/WSC	0x0	wr_multi_rd_rst When set, resets the DTU in write once multiple reads mode, to allow a new write to be performed. This bit automatically clears.
1	R/WSC	0x0	run_error_reports When set, initiates the calculation of the error status bits. This bit automatically clears when the re-calculation is done. This is only used in debug mode to verify the comparison logic.
0	R/WSC	0x0	run_dtu When set, initiates the running of the DTU read and write transfer. This bit automatically clears when the transfers are completed

DDR_PCTL_DTUWD0
Address: Operational Base + offset (0x0210)
DTU Write Data #0 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte3 Write data byte
23:16	RW	0x00	dtu_wr_byte2 Write data byte

Bit	Attr	Reset Value	Description
15:8	RW	0x00	dtu_wr_byte1 Write data byte
7:0	RW	0x00	dtu_wr_byte0 Write data byte

DDR_PCTL_DTUWD1
Address: Operational Base + offset (0x0214)
DTU Write Data #1 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte7 Write data byte
23:16	RW	0x00	dtu_wr_byte6 Write data byte
15:8	RW	0x00	dtu_wr_byte5 Write data byte
7:0	RW	0x00	dtu_wr_byte4 Write data byte

DDR_PCTL_DTUWD2
Address: Operational Base + offset (0x0218)
DTU Write Data #2 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte11 Write data byte
23:16	RW	0x00	dtu_wr_byte10 Write data byte
15:8	RW	0x00	dtu_wr_byte9 Write data byte
7:0	RW	0x00	dtu_wr_byte8 Write data byte

DDR_PCTL_DTUWD3
Address: Operational Base + offset (0x021c)
DTU Write Data #3 Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	dtu_wr_byte15 Write data byte
23:16	RW	0x00	dtu_wr_byte14 Write data byte
15:8	RW	0x00	dtu_wr_byte13 Write data byte

Bit	Attr	Reset Value	Description
7:0	RW	0x00	dtu_wr_byte12 Write data byte

DDR_PCTL_DTUWDM
Address: Operational Base + offset (0x0220)
DTU Write Data Mask Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	dm_wr_byte0 Write data mask bit, one bit for each byte. Each bit should be 0 for a byte lane that contains valid write data.

DDR_PCTL_DTURD0
Address: Operational Base + offset (0x0224)
DTU Read Data #0 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte3 Read byte
23:16	RO	0x00	dtu_rd_byte2 Read byte
15:8	RO	0x00	dtu_rd_byte1 Read byte
7:0	RO	0x00	dtu_rd_byte0 Read byte

DDR_PCTL_DTURD1
Address: Operational Base + offset (0x0228)
DTU Read Data #1 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte7 Read byte
23:16	RO	0x00	dtu_rd_byte6 Read byte
15:8	RO	0x00	dtu_rd_byte5 Read byte
7:0	RO	0x00	dtu_rd_byte4 Read byte

DDR_PCTL_DTURD2
Address: Operational Base + offset (0x022c)
DTU Read Data #2 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte11 Read byte
23:16	RO	0x00	dtu_rd_byte10 Read byte
15:8	RO	0x00	dtu_rd_byte9 Read byte
7:0	RO	0x00	dtu_rd_byte8 Read byte

DDR_PCTL_DTURD3
Address: Operational Base + offset (0x0230)
DTU Read Data #3 Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	dtu_rd_byte15 Read byte
23:16	RO	0x00	dtu_rd_byte14 Read byte
15:8	RO	0x00	dtu_rd_byte13 Read byte
7:0	RO	0x00	dtu_rd_byte12 Read byte

DDR_PCTL_DTULFSRWD
Address: Operational Base + offset (0x0234)
DTU LFSR Seed for Write Data Generation Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_wseed This is the initial seed for the random write data generation LFSR (linear feedback shift register), shared with the write mask generation.

DDR_PCTL_DTULFSRRD
Address: Operational Base + offset (0x0238)
DTU LFSR Seed for Read Data Generation Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dtu_lfsr_rseed This is the initial seed for the random read data generation LFSR (linear feedback shift register), this is shared with the read mask generation. The read data mask is reconstructed the same as the write data mask was created, allowing the " on the fly comparison" ignore bytes which were not written.

DDR_PCTL_DTUEAF
Address: Operational Base + offset (0x023c)
DTU Error Address FIFO Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	ea_rank Indicates the rank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
29	RO	0x0	reserved
28:13	RO	0x0000	ea_row Indicates the row that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.
12:10	RO	0x0	ea_bank Indicates the bank that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes
9:0	RO	0x000	ea_column Indicates the column address that the error occurred in during random data generation. There could be a number of entries in this FIFO. If FIFO is empty one reads zeroes.

DDR_PCTL_DFITCTRLDELAY
Address: Operational Base + offset (0x0240)
DFI tctrl_delay Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x2	tctrl_delay Specifies the number of DFI clock cycles after an assertion or deassertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFIODTCFG
Address: Operational Base + offset (0x0244)
DFI ODT Configuration

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	rank3_odt_default Default ODT value of rank 3 when there is no read/write activity
27	RW	0x0	rank3_odt_write_sel Enable/disable ODT for rank 3 when a write access is occurring on this rank
26	RW	0x0	rank3_odt_write_nse Enable/disable ODT for rank 3 when a write access is occurring on a different rank
25	RW	0x0	rank3_odt_read_sel Enable/disable ODT for rank 3 when a read access is occurring on this rank
24	RW	0x0	rank3_odt_read_nsel Enable/disable ODT for rank 3 when a read access is occurring on a different rank
23:21	RO	0x0	reserved
20	RW	0x0	rank2_odt_default Default ODT value of rank 2 when there is no read/write activity
19	RW	0x0	rank2_odt_write_sel Enable/disable ODT for rank 2 when a write access is occurring on this rank
18	RW	0x0	rank2_odt_write_nse Enable/disable ODT for rank 2 when a write access is occurring on a different rank

Bit	Attr	Reset Value	Description
17	RW	0x0	rank2_odt_read_sel Enable/disable ODT for rank 2 when a read access is occurring on this rank
16	RW	0x0	rank2_odt_read_nsel Enable/disable ODT for rank 2 when a read access is occurring on a different rank
15:13	RO	0x0	reserved
12	RW	0x0	rank1_odt_default Default ODT value of rank 1 when there is no read/write activity
11	RW	0x0	rank1_odt_write_sel Enable/disable ODT for rank 1 when a write access is occurring on this rank
10	RW	0x0	rank1_odt_write_nse Enable/disable ODT for rank 1 when a write access is occurring on a different rank
9	RW	0x0	rank1_odt_read_sel Enable/disable ODT for rank 1 when a read access is occurring on this rank
8	RW	0x0	rank1_odt_read_nsel Enable/disable ODT for rank 1 when a read access is occurring on a different rank
7:5	RO	0x0	reserved
4	RW	0x0	rank0_odt_default Default ODT value of rank 0 when there is no read/write activity
3	RW	0x0	rank0_odt_write_sel Enable/disable ODT for rank 0 when a write access is occurring on this rank
2	RW	0x0	rank0_odt_write_nse Enable/disable ODT for rank 0 when a write access is occurring on a different rank
1	RW	0x0	rank0_odt_read_sel Enable/disable ODT for rank 0 when a read access is occurring on this rank
0	RW	0x0	rank0_odt_read_nsel Enable/disable ODT for rank 0 when a read access is occurring on a different rank

DDR_PCTL_DFIODTCFG1
Address: Operational Base + offset (0x0248)
DFI ODT Timing Configuration 1 (for Latency and Length)

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:24	RW	0x6	ODT length for BL8 read transfers Length of dfi_odt signal for BL8 reads. This is in terms of SDR cycles. For BL4 reads, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
23:19	RO	0x0	reserved
18:16	RW	0x6	ODT length for BL8 write transfers Length of dfi_odt signal for BL8 writes. This is in terms of SDR cycles. For BL4 writes, the length of dfi_odt is always 2 cycles shorter than the value in this register field.
15:13	RO	0x0	reserved
12:8	RW	0x00	ODT latency for reads Latency after a read command that dfi_odt is set. This is in terms of SDR cycles.
7:5	RO	0x0	reserved
4:0	RW	0x00	ODT latency for writes Latency after a write command that dfi_odt is set. This is in terms of SDR cycles

DDR_PCTL_DFIODTRANKMAP

Address: Operational Base + offset (0x024c)

DFI ODT Rank Mapping

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:12	RW	0x8	Rank mapping for dfi_odt[3] Determines whether dfi_odt[3] should be asserted when the PCTLrequires to terminate each rank Bit 15 = 1: dfi_odt[3] will be asserted to terminate rank 3 Bit 14 = 1: dfi_odt[3] will be asserted to terminate rank 2 Bit 13 = 1: dfi_odt[3] will be asserted to terminate rank 1 Bit 12 = 1: dfi_odt[3] will be asserted to terminate rank 0 This field exists only if PCTL_M_NRANKS = 4

Bit	Attr	Reset Value	Description
11:8	RW	0x4	Rank mapping for dfi_odt[2] Determines which rank access(es) will cause dfi_odt[2] to be asserted Bit 11 = 1: dfi_odt[2] will be asserted to terminate rank 3 Bit 10 = 1: dfi_odt[2] will be asserted to terminate rank 2 Bit 9 = 1: dfi_odt[2] will be asserted to terminate rank 1 Bit 8 = 1: dfi_odt[2] will be asserted to terminate rank 0 This field exists only if PCTL_M_NRANKS = 4
7:4	RW	0x2	Rank mapping for dfi_odt[1] Determines which rank access(es) will cause dfi_odt[1] to be asserted Bit 7= 1: dfi_odt[1] will be asserted to terminate rank 3 Bit 6= 1: dfi_odt[1] will be asserted to terminate rank 2 Bit 5= 1: dfi_odt[1] will be asserted to terminate rank 1 Bit 4= 1: dfi_odt[1] will be asserted to terminate rank 0 This field exists only if PCTL_M_NRANKS >
3:0	RW	0x1	Rank mapping for dfi_odt[0] Determines which rank access(es) will cause dfi_odt[0] to be asserted Bit 3= 1: dfi_odt[0] will be asserted to terminate rank 3 Bit 2= 1: dfi_odt[0] will be asserted to terminate rank 2 Bit 1= 1: dfi_odt[0] will be asserted to terminate rank 1 Bit 0= 1: dfi_odt[0] will be asserted to terminate rank 0

DDR_PCTL_DFITPHYWRDATA
Address: Operational Base + offset (0x0250)
DFI tphy_wrdata Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x01	tphy_wrdata Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted to when the associated write data is driven on the dfi_wrdata signal. This has no impact on performance, only adjusts the relative time between enable and data transfer.

DDR_PCTL_DFITPHYWRLAT
Address: Operational Base + offset (0x0254)
DFI tphy_wrlat Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	tphy_wrlat Specifies the number of DFI clock cycles between when a write command is sent on the DFI control interface and when the dfi_wrdata_en signal is asserted.

DDR_PCTL_DFITRDDATAEN
Address: Operational Base + offset (0x0260)
DFI trddata_en Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x01	trddata_en Specifies the number of DFI clock cycles from the assertion of a read command on the DFI to the assertion of the dfi_rddata_en signal.

DDR_PCTL_DFITPHYRDLAT
Address: Operational Base + offset (0x0264)
DFI tphy_rdlat Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x0f	tphy_rdlat Specifies the maximum number of DFI clock cycles allowed from the assertion of the dfi_rddata_en signal to the assertion of the dfi_rddata_valid signal.

DDR_PCTL_DFITPHYUPDTYPE0

Address: Operational Base + offset (0x0270)
DFI tphyupd_type0 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type0 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x0. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE1
Address: Operational Base + offset (0x0274)
DFI tphyupd_type1 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type1 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x1. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE2
Address: Operational Base + offset (0x0278)
DFI tphyupd_type2 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type2 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x2. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITPHYUPDTYPE3
Address: Operational Base + offset (0x027c)
DFI tphyupd_type3 Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x010	tphyupd_type3 Specifies the maximum number of DFI clock cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x3. The dfi_phyupd_req signal may de-assert at any cycle after the assertion of the dfi_phyupd_ack signal.

DDR_PCTL_DFITCTRLUPDMIN
Address: Operational Base + offset (0x0280)
DFI tctrlupd_min Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0010	tctrlupd_min Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted.

DDR_PCTL_DFITCTRLUPDMAX
Address: Operational Base + offset (0x0284)
DFI tctrlupd_max Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0040	tctrlupd_max Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert.

DDR_PCTL_DFITCTRLUPDDLY
Address: Operational Base + offset (0x0288)
DFI tctrlupddly Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x8	tctrlupd_dly Delay in DFI clock cycles between time a PCTL-initiated update could be started and time PCTL-initated update actually starts (dfi_ctrlupd_req going high).

DDR_PCTL_DFIUPDCFG

Address: Operational Base + offset (0x0290)
DFI Update Configuration Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x1	dfi_phyupd_en Enables the support for acknowledging PHY-initiated updates: 1'b0 = Disabled 1'b1 = Enabled
0	RW	0x1	dfi_ctrlupd_en Enables the generation of PCTL-initiated updates: 1'b0 = Disabled 1'b1 = Enabled

DDR_PCTL_DFITREFMSKI
Address: Operational Base + offset (0x0294)
DFI Masked Refresh Interval

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	trefmski Time period of the masked Refresh interva This value is only used if TREFI==0. Defines the time period (in 100ns units) of the masked Refresh (REFMSK) interval. The actual time period defined is DFITREFMSKI* TOGCNT100N * internal timers clock period.

DDR_PCTL_DFITCTRLUPDI
Address: Operational Base + offset (0x0298)
DFI tctrlupd_interval Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	tctrlupd_interval DFI PCTL-initiated updates interval, measured in terms of Refresh interval units. If TREFI!=0, the time period is defined as DFITCTRLUPDI*TREFI * TOGCNT100N * internal timers clock period. If TREFI==0 and DFITREFMSKI!=0, the period changes to DFITCTRLUPDI*DFITREFMSKI* * TOGCNT100N * internal timers clock period. Programming a value of 0 is the same as programming a value of 1; for instance, a PCTL-initiated update occurs every Refresh interval.

DDR_PCTL_DFITRCFG0
Address: Operational Base + offset (0x02ac)
DFI Training Configuration 0 Register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:16	RW	0x0	dfi_wrlvl_rank_sel Determines the value to drive on the output signal dfi_wrlvl_cs_n. The value on dfi_wrlvl_cs_n is the inverse of the setting in this field.
15:13	RO	0x0	reserved
12:4	RW	0x000	dfi_rdlvl_edge Determines the value to drive on the output signal dfi_rdlvl_edge. The value on dfi_rdlvl_edge is the same as the setting in this field.
3:0	RW	0x0	dfi_rdlvl_rank_sel Determines the value to drive on the output signal dfi_rdlvl_cs_n. The value on dfi_rdlvl_cs_n is the inverse of the setting in this field.

DDR_PCTL_DFITRSTAT0
Address: Operational Base + offset (0x02b0)
DFI Training Status 0 Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved

Bit	Attr	Reset Value	Description
17:16	RO	0x0	dfi_wrlvl_mode Reports the value of the input signal dfi_wrlvl_mode.
15:10	RO	0x0	reserved
9:8	RO	0x0	dfi_rdlvl_gate_mode Reports the value of the input signal dfi_rdlvl_gate_mode.
7:2	RO	0x0	reserved
1:0	RO	0x0	dfi_rdlvl_mode Reports the value of the input signal dfi_rdlvl_mode.

DDR_PCTL_DFITRWRLVLEN
Address: Operational Base + offset (0x02b4)
DFI Training dfi_wrlvl_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_wrlvl_en Determines the value to drive on the output signal dfi_wrlvl_en.

DDR_PCTL_DFITRRDLVLEN
Address: Operational Base + offset (0x02b8)
DFI Training dfi_rdlvl_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_en Determines the value to drive on the output signal dfi_rdlvl_en.

DDR_PCTL_DFITRRDLVLGATEEN
Address: Operational Base + offset (0x02bc)
DFI Training dfi_rdlvl_gate_en Register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved
8:0	RW	0x000	dfi_rdlvl_gate_en Determines the value to drive on the output signal dfi_rdlvl_gate_en.

DDR_PCTL_DFISTSTAT0
Address: Operational Base + offset (0x02c0)
DFI Status Status 0 Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:16	RO	0x000	dfi_data_byte_disable Reports the value of the output signal dfi_data_byte_disable.
15:6	RO	0x0	reserved
5:4	RO	0x0	dfi_freq_ratio Reports the value of the output signal dfi_freq_ratio.
3:2	RO	0x0	reserved
1	RO	0x0	dfi_init_start Reports the value of the output signal dfi_init_start.
0	RO	0x0	dfi_init_complete Reports the value of the input signal dfi_init_complete.

DDR_PCTL_DFISTCFG0

Address: Operational Base + offset (0x02c4)

DFI Status Configuration 0 Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	dfi_data_byte_disable_en Enables the driving of the dfi_data_byte_disable signal. The value driven on dfi_data_byte_disable is dependent on the setting of PPCFG register. 1'b0 - Drive dfi_data_byte_disable to default value of all zeroes. 1'b1 - Drive dfi_data_byte_disable according to value as defined by PPCFG register setting. Note: should be set to 1only after PPCFG is correctly set.

Bit	Attr	Reset Value	Description
1	RW	0x0	dfi_freq_ratio_en Enables the driving of the dfi_freq_ratio signal. When enabled, the dfi_freq_ratio value driven is dependent on configuration parameter PCTL_FREQ_RATIO: 2'b00 is driven when PCTL_FREQ_RATIO=1; 2'b01 is driven when PCTL_FREQ_RATIO=2. 1'b0 - Drive dfi_freq_ratio to default value of 2'b00. 1'b1 - Drive dfi_freq_ratio value according to how configuration parameter is set.
0	RW	0x0	dfi_init_start Sets the value of the dfi_init_start signal. 1'b0 - dfi_init_start is driven low 1'b1 - dfi__init_start is driven high

DDR_PCTL_DFISTCFG1
Address: Operational Base + offset (0x02c8)
DFI Status Configuration 1 Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	dfi_dram_clk_disable_en_dpd Enables support of the dfi_dram_clk_disable signal with Deep Power Down (DPD). DPD is only for mDDR/LPDDR2. 1'b0 - Disable dfi_dram_clk_disable support in relation to DPD 1'b1 - Enable dfi_dram_clk_disable support in relation to DPD
0	RW	0x0	dfi_dram_clk_disable_en Enables support of the dfi_dram_clk_disable signal with Self Refresh (SR). 1'b0 - Disable dfi_dram_clk_disable support in relation to SR 1'b1 - Enable dfi_dram_clk_disable support in relation to SR

DDR_PCTL_DFITDRAMCLKEN
Address: Operational Base + offset (0x02d0)
DFI tdram_clk_enable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x2	tdram_clk_enable Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phasealigned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFITDRAMCLKDIS
Address: Operational Base + offset (0x02d4)
DFI tdram_clk_disable Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x2	tdram_clk_disable Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phasealigned, this timing parameter should be rounded up to the next integer value.

DDR_PCTL_DFISTCFG2
Address: Operational Base + offset (0x02d8)
DFI Status Configuration 2 Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	parity_en Enables the DFI parity generation feature (driven on output signal dfi_parity_in) 1'b0 - Disable DFI parity generation 1'b1 - Enable DFI parity generation
0	RW	0x0	parity_intr_en Enable interrupt generation for DFI parity error (from input signal dfi_parity_error). 1'b0 - Disable interrupt 1'b1 - Enable interrupt

DDR_PCTL_DFISTPARCLR

Address: Operational Base + offset (0x02dc)
DFI Status Parity Clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	R/WSC	0x0	parity_log_clr Set this bit to 1'b1 to clear the DFI Status Parity Log register (DFISTPARLOG). 1'b0 = Do not clear DFI status Parity Log register 1'b1 = Clear DFI status Parity Log register
0	R/WSC	0x0	parity_intr_clr Set this bit to 1'b1 to clear the interrupt generated by an DFI parity error (as enabled by DFISTCFG2.parity_intr_en). It also clears the INTRSTAT.parity_intr register field. It is automatically cleared by hardware when the interrupt has been cleared.

DDR_PCTL_DFISTPARLOG
Address: Operational Base + offset (0x02e0)
DFI Status Parity Log Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	parity_err_cnt Increments any time the DFI parity logic detects a parity error(s) (on dfi_parity_error).

DDR_PCTL_DFILPCFG0
Address: Operational Base + offset (0x02f0)
DFI Low Power Configuration 0 Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:28	RW	0x0	dfi_lp_wakeup_dpd Value to drive on dfi_lp_wakeup signal when Deep Power Down mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000 - 16 cycles 4'b0001 - 32 cycles 4'b0010 - 64 cycles 4'b0011 - 128 cycles 4'b0100 - 256 cycles 4'b0101 - 512 cycles 4'b0110 - 1024 cycles 4'b0111 - 2048 cycles 4'b1000 - 4096 cycles 4'b1001 - 8192 cycles 4'b1010 - 16384 cycles 4'b1011 - 32768 cycles 4'b1100 - 65536 cycles 4'b1101 - 131072 cycles 4'b1110 - 262144 cycles 4'b1111 - Unlimited
27:25	RO	0x0	reserved
24	RW	0x0	dfi_lp_en_dpd Enables DFI Low Power interface handshaking during Deep Power Down Entry/Exit. 1'b0 - Disabled 1'b1 - Enabled
23:20	RO	0x0	reserved
19:16	RW	0x7	dfi_tlp_resp Setting for tlp_resp time. Same value is used for both Power Down and Self refresh and Deep Power Down modes. DFI 2.1 specification, recommends using value of 7 always.

Bit	Attr	Reset Value	Description
15:12	RW	0x0	dfi_lp_wakeup_sr Value to drive on dfi_lp_wakeup signal when Self Refresh mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000 - 16 cycles 4'b0001 - 32 cycles 4'b0010 - 64 cycles 4'b0011 - 128 cycles 4'b0100 - 256 cycles 4'b0101 - 512 cycles 4'b0110 - 1024 cycles 4'b0111 - 2048 cycles 4'b1000 - 4096 cycles 4'b1001 - 8192 cycles 4'b1010 - 16384 cycles 4'b1011 - 32768 cycles 4'b1100 - 65536 cycles 4'b1101 - 131072 cycles 4'b1110 - 262144 cycles 4'b1111 - Unlimited
11:9	RO	0x0	reserved
8	RW	0x0	dfi_lp_en_sr Enables DFI Low Power interface handshaking during Self Refresh Entry/Exit. 1'b0 - Disabled 1'b1 - Enabled

Bit	Attr	Reset Value	Description
7:4	RW	0x0	dfi_lp_wakeup_pd Value to drive on dfi_lp_wakeup signal when Power Down mode is entered. Determines the DFI's tlp_wakeup time: 4'b0000 - 16 cycles 4'b0001 - 32 cycles 4'b0010 - 64 cycles 4'b0011 - 128 cycles 4'b0100 - 256 cycles 4'b0101 - 512 cycles 4'b0110 - 1024 cycles 4'b0111 - 2048 cycles 4'b1000 - 4096 cycles 4'b1001 - 8192 cycles 4'b1010 - 16384 cycles 4'b1011 - 32768 cycles 4'b1100 - 65536 cycles 4'b1101 - 131072 cycles 4'b1110 - 262144 cycles 4'b1111 - Unlimited
3:1	RO	0x0	reserved
0	RW	0x0	dfi_lp_en_pd Enables DFI Low Power interface handshaking during Power Down Entry/Exit. 1'b0 - Disabled 1'b1 - Enabled

DDR_PCTL_DFITRWRLVLRESP0
Address: Operational Base + offset (0x0300)
DFI Training dfi_wrlvl_resp Status 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wrlvl_resp0 Reports the status of the dif_wrlvl_resp[31:0] signal.

DDR_PCTL_DFITRWRLVLRESP1
Address: Operational Base + offset (0x0304)
DFI Training dfi_wrlvl_resp Status 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_wrlvl_resp1 Reports the status of the dif_wrlvl_resp[63:32] signal.

DDR_PCTL_DFITRWRLVLRESP2
Address: Operational Base + offset (0x0308)
DFI Training dfi_wrlvl_resp Status 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_wrlvl_resp2 Reports the status of the dif_wrlvl_resp[71:64] signal.

DDR_PCTL_DFITRRDLVLRESP0
Address: Operational Base + offset (0x030c)
DFI Training dfi_rdlvl_resp Status 0 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp0 Reports the status of the dif_rdlvl_resp[31:0] signal.

DDR_PCTL_DFITRRDLVLRESP1
Address: Operational Base + offset (0x0310)
DFI Training dfi_rdlvl_resp Status 1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	dfi_rdlvl_resp1 Reports the status of the dif_rdlvl_resp[63:32] signal.

DDR_PCTL_DFITRRDLVLRESP2
Address: Operational Base + offset (0x0314)
DFI Training dfi_rdlvl_resp Status 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RO	0x00	dfi_rdlvl_resp2 Reports the status of the dif_rdlvl_resp[71:64] signal.

DDR_PCTL_DFITRWRLVLDELAY0
Address: Operational Base + offset (0x0318)
DFI Training dfi_wrlvl_delay Configuration 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay0 Sets the value to be driven on the signal dfi_wrlvl_delay_x[31:0].

DDR_PCTL_DFITRWRLVLDELAY1
Address: Operational Base + offset (0x031c)
DFI Training dfi_wrlvl_delay Configuration 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_wrlvl_delay1 Sets the value to be driven on the signal dfi_wrlvl_delay_x[63:32].

DDR_PCTL_DFITRWRLVLDELAY2
Address: Operational Base + offset (0x0320)
DFI Training dfi_wrlvl_delay Configuration 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_wrlvl_delay2 Sets the value to be driven on the signal dfi_wrlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLDELAY0
Address: Operational Base + offset (0x0324)
DFI Training dfi_rdlvl_delay Configuration 0 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay0 Sets the value to be driven on the signal dfi_rdlvl_delay_x[31:0].

DDR_PCTL_DFITRRDLVLDELAY1
Address: Operational Base + offset (0x0328)
DFI Training dfi_rdlvl_delay Configuration 1 Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_delay1 Sets the value to be driven on the signal dfi_rdlvl_delay_x[63:32].

DDR_PCTL_DFITRRDLVLDELAY2
Address: Operational Base + offset (0x032c)
DFI Training dfi_rdlvl_delay Configuration 2 Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_delay2 Sets the value to be driven on the signal dfi_rdlvl_delay_x[71:64].

DDR_PCTL_DFITRRDLVLGATEDELAY0

Address: Operational Base + offset (0x0330)
DFI Training dfi_rdlvl_gate_delay Configuration 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay0 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[31:0].

DDR_PCTL_DFITRRDLVLGATEDELAY1
Address: Operational Base + offset (0x0334)
DFI Training dfi_rdlvl_gate_delay Configuration 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dfi_rdlvl_gate_delay1 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[63:32].

DDR_PCTL_DFITRRDLVLGATEDELAY2
Address: Operational Base + offset (0x0338)
DFI Training dfi_rdlvl_gate_delay Configuration 2

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	dfi_rdlvl_gate_delay2 Sets the value to be driven on the signal dfi_rdlvl_gate_delay_x[71:64].

DDR_PCTL_DFITRCMD
Address: Operational Base + offset (0x033c)
DFI Training Command Register

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	dfitrcmd_start DFI Training Command Start. When this bit is set to 1, the command operation defined in the dfitrcmd_opcode field is started. This bit is automatically cleared by the PCTL after the command is finished. The application can poll this bit to determine when PCTL is ready to accept another command. This bit cannot be cleared to 1b0 by software.
30:13	RO	0x0	reserved
12:4	RW	0x000	dfitrcmd_en DFI Training Command Enable. Selects which bits of chosen DFI Training command to drive to 1'b1.
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RW	0x0	dfitrcmd_opcode DFI Training Command Opcode. Select which DFI Training command to generate for one n_clk cycle: 2'b00 - dfi_wrlvl_load 2'b01 - dfi_wrlvl_strobe 2'b10 - dfi_rdlvl_load 2'b11 - Reserved.

DDR_PCTL_IPVR
Address: Operational Base + offset (0x03f8)
IP Version Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ip_version ASCII value for each number in the version, followed by a *.

DDR_PCTL_IPTR
Address: Operational Base + offset (0x03fc)
IP Type Register

Bit	Attr	Reset Value	Description
31:0	RO	0x44574300	ip_type Contains the IP's identification code, which is an ASCII value to identify the component and it is currently set to the string "DWC". This value never changes.

DDR_PHYCTL_RIDR
Address: Operational Base + offset (0x0000)
Revision Identification Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	UDRID User-Defined Revision ID: General purpose revision identification set by the user.
23:20	RO	0x1	PHYMJR PHY Major Revision: Indicates major revision of the PHY such addition of the features that make the new version not compatible with previous versions.

Bit	Attr	Reset Value	Description
19:16	RO	0x0	PHYMDR PHY Moderate Revision: Indicates moderate revision of the PHY such as addition of new features. Normally the new version is still compatible with previous versions.
15:12	RO	0x0	PHYMNR PHY Minor Revision: Indicates minor update of the PHY such as bug fixes. Normally no new features are included.
11:8	RO	0x1	PUBMJR PUB Major Revision: Indicates major revision of the PUB such addition of the features that make the new version not compatible with previous versions.
7:4	RO	0x4	PUBMDR PUB Moderate Revision: Indicates moderate revision of the PUB such as addition of new features. Normally the new version is still compatible with previous versions.
3:0	RO	0x0	PUBMNR PUB Minor Revision: Indicates minor update of the PUB such as bug fixes. Normally no new features are included.

DDR_PHYCTL_PIR
Address: Operational Base + offset (0x0004)
PHY Initialization Register

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	INITBYP Initialization Bypass: Bypasses or stops, if set, all initialization routines currently running, including PHY initialization, DRAM initialization, and PHY training. Initialization may be triggered manually using INIT and the other relevant bits of the PIR register. This bit is self-clearing.
30	R/WSC	0x0	ZCALBYP Impedance Calibration Bypass: Bypasses or stops, if set, impedance calibration of all ZQ control blocks that automatically triggers after reset. Impedance calibration may be triggered manually using INIT and ZCAL bits of the PIR register. This bit is self-clearing.

Bit	Attr	Reset Value	Description
29	R/WSC	0x0	LOCKBYP DLL Lock Bypass: Bypasses or stops, if set, the waiting of DLLs to lock. DLL lock wait is automatically triggers after reset. DLL lock wait may be triggered manually using INIT and DLLLOCK bits of the PIR register. This bit is self-clearing.
28	R/WSC	0x0	CLRSR Clear Status Registers: A write of '1' to this bit will clear (reset to '0' all status registers, including PGSR and DXnGSR. The clear status register bit is self-clearing. This bit is primarily for debug purposes and is typically not needed during normal functional operation. It can be used when PGSR.IDONE=1, to manually clear the PGSR status bits, although starting a new init process will automatically clear the PGSR status bits. Or it can be used to manually clear the DXnGSR status bits, although starting a new data training process will automatically clear the DXnGSR status bits.
27:19	RO	0x0	reserved
18	RW	0x0	CTLDINIT Controller DRAM Initialization: Indicates if set that DRAM initialization will be performed by the controller. Otherwise if not set it indicates that DRAM initialization will be performed using the built-in initialization sequence or using software through the configuration port.
17	RW	0x0	DLLBYP DLL Bypass: A setting of 1 on this bit will put all PHY DLLs in bypass mode. A bypassed DLL is also powered down (disabled).

Bit	Attr	Reset Value	Description
16	RW	0x0	ICPC Initialization Complete Pin Configuration: Specifies how the DFI 2.1 initialization complete output pin should be used to indicate the status of initialization. Valid value are: 0 = Asserted after PHY initialization (DLL locking and impedance calibration) is complete. 1 = Asserted after PHY initialization is complete and the triggered the PHYCTL initialization (DRAM initialization, data training, or initialization trigger with no selected initialization) is complete.
15:9	RO	0x0	reserved
8	RW	0x0	EYETRN Read Data Eye Training: Executes a PHYCTL training routine to maximize the read data eye. This is not implemented in this version of the PHYCTL.
7	RW	0x0	QSTRN Read DQS Training: Executes a PHYCTL training routine to determine the optimum position of the read data DQS strobe for maximum system timing margins.
6	RW	0x0	DRAMINIT DRAM Initialization: Executes the DRAM initialization sequence.
5	RW	0x0	DRAMRST DRAM Reset (DDR3 Only): Issues a reset to the DRAM (by driving the DRAM reset pin low) and wait 200us. This can be triggered in isolation or with the full DRAM initialization (DRAMINIT). For the later case, the reset is issued and 200us is waited before starting the full initialization sequence.
4	RW	0x0	ITMSRST Interface Timing Module Soft Reset: Soft resets the interface timing modules for the data and data strobes, i.e., it asserts the ITM soft reset (srstb) signal.

Bit	Attr	Reset Value	Description
3	RW	0x0	ZCAL Impedance Calibrate: Performs PHY impedance calibration.
2	RW	0x0	DLLLOCK DLL Lock: Waits for the PHY DLLs to lock.
1	RW	0x0	DLLSRST DLL Soft Rest: Soft resets all PHY DLLs by driving the DLL soft reset pin.
0	RW	0x0	INIT Initialization Trigger: A write of '1' to this bit triggers the DDR system initialization, including PHY initialization, DRAM initialization, and PHY training. The exact initialization steps to be executes are specified in bits 1 to 6 of this register. A bit setting of 1 means the step will be executed as part of the initialization sequence, while a setting of □initialization) is complete.

DDR_PHYCTL_PGCR
Address: Operational Base + offset (0x0008)
PHY General Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	LBMODE Loopback Mode: Indicates if set that the PHY/PUB is in loopback mode
30	RW	0x0	LBGDQS Loopback DQS Gating: Selects the DQS gating mode that should be used when the PHY is in loopback mode, including BIST loopback mode. Valid values are: 0 = DQS gate training will be triggered on the PUB 1 = DQS gate is set manually using software
29	RW	0x0	LBDQSS Loopback DQS Shift: Selects how the read DQS is shifted during loopback to ensure that the read DQS is centered into the read data eye. Valid values are: 0 = PUB sets the read DQS delay to 0; DQS is already shifted 90 degrees by write path 1 = The read DQS shift is set manually through software

Bit	Attr	Reset Value	Description
28:25	RW	0x0	RFSHDT Refresh During Training: A non-zero value specifies that a burst of refreshes equal to the number specified in this field should be sent to the SDRAM after training each rank except the last rank.
24	RW	0x1	PDDISDX Power Down Disabled Byte: Indicates if set that the DLL and I/Os of a disabled byte should be powered down.
23:22	RW	0x2	ZCKSEL Impedance Clock Divider Select: Selects the divide ratio for the clock used by the impedance control logic relative to the clock used by the memory controller and SDRAM. Valid values are: 00 = Divide by 2 01 = Divide by 8 10 = Divide by 32 11 = Divide by 64
21:18	RW	0xf	RANKEN Rank Enable: Specifies the ranks that are enabled for data-training. Bit 0 controls rank 0, bit 1 controls rank 1, bit 2 controls rank 2, and bit 3 controls rank 3. Setting the bit to '0' enables the rank, and setting it to '1' disables the rank.
17:16	RW	0x0	IODDRM I/O DDR Mode (D3F I/O Only): Selects the DDR mode for the I/Os.
15	RW	0x0	IOLB I/O Loop-Back Select: Selects where inside the I/O the loop-back of signals happens. Valid values are: 0 = Loopback is after output buffer; output enable must be asserted 1 = Loopback is before output buffer; output enable is don't care
14	RW	0x0	CKINV CK Invert: Specifies if set that CK/CK# should be inverted. Otherwise CK/CK# toggles with normal polarity.

Bit	Attr	Reset Value	Description
13:12	RW	0x2	CKDV CK Disable Value: Specifies the static value that should be driven on CK/CK# pair(s) when the pair(s) is disabled. CKDV[0] specifies the value for CK and CKDV[1] specifies the value for CK#
11:9	RW	0x7	CKEN CK Enable: Controls whether the CK going to the SDRAM is enabled (toggling) or disabled (static value defined by CKDV). One bit for each of the three CK pairs.
8:5	RW	0x0	DTOSEL Digital Test Output Select: Selects the PHY digital test output that should be driven onto PHY digital test output (phy_dto) pin: Valid values are: 0000 = DATX8 0 DLL digital test output 0001 = DATX8 1 DLL digital test output 0010 = DATX8 2 DLL digital test output 0011 = DATX8 3 DLL digital test output 0100 = DATX8 4 DLL digital test output 0101 = DATX8 5 DLL digital test output 0110 = DATX8 6 DLL digital test output 0111 = DATX8 7 DLL digital test output 1000 = DATX8 8 DLL digital test output 1001 = AC DLL digital test output 1010 - 01111 = Reserved
4:3	RW	0x0	DFTLMT DQS Drift Limit: Specifies the expected limit of drift on read data strobes. A drift of this value or greater is reported as a drift error through the host port error flag. Valid values are: 00 = No limit (no error reported) 01 = 90 deg drift 10 = 180 deg drift 11 = 270 deg or more drift Note: Although reported through the error flag, this is not an error requiring any action. It is simply an indicator that the drift is greater than expected.

Bit	Attr	Reset Value	Description
2	RW	0x1	DFTCMP DQS Drift Compensation: Enables or disables DQS drift compensation. Valid values are: 0 = Disables data strobe drift compensation 1 = Enables data strobe drift compensation By default, drift compensation is enabled. Note: Drift compensation must be disabled for LPDDR2.
1	RW	0x0	DQSCFG DQS Gating Configuration: Selects one of the two DQS gating schemes: 0 = DQS gating is shut off using the rising edge of DQS_b (active windowing mode) 1 = DQS gating blankets the whole burst (passive windowing mode). Note: Passive windowing must be used for LPDDR2.
0	RW	0x0	ITMDMD ITM DDR Mode: Selects whether ITMS uses DQS and DQS# or it only uses DQS. Valid values are: 0 = ITMS uses DQS and DQS# 1 = ITMS uses DQS only Note: The only valid value for DDR is 1.

DDR_PHYCTL_PGSR

Address: Operational Base + offset (0x000c)

PHY General Status Register

Bit	Attr	Reset Value	Description
31	RO	0x0	TQ Temperature Output (LPDDR Only): Connected to the DRAM TQ pin which is defined to go high when the LPDDR device temperature equals to or exceeds 85C, otherwise it is low.
30:8	RO	0x0	reserved
7	RO	0x0	DFTERR DQS Drift Error: If set, indicates that at least one of the read data strobes has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR).

Bit	Attr	Reset Value	Description
6	RO	0x0	DTIERR Data Training Intermittent Error: If set, indicates that there was an intermittent error during data training, such as a pass was followed by a fail then followed by another pass.
5	RO	0x0	DTERR Data Training Error: If set, indicates that a valid DQS gating window could not be found during data training.
4	RO	0x0	DTDONE Data Training Done: Indicates, if set, that the PHY has finished doing data training.
3	RO	0x0	DIDONE DRAM Initialization Done: Indicates if set that DRAM initialization has completed.
2	RO	0x0	ZCDONE Impedance Calibration Done: Indicates if set that impedance calibration has completed.
1	RO	0x0	DLDONE DLL Lock Done: Indicates if set that DLL locking has completed.
0	RO	0x0	IDONE Initialization Done: Indicates if set that the DDR system initialization has completed. This bit is set after all the selected initialization routines in PIR register have completed.

DDR_PHYCTL_DLLGCR
Address: Operational Base + offset (0x0010)
DLL General Control Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	LOCKDET Master lock detector enable.
28	RO	0x0	reserved
27:20	RW	0x37	SBIAS Slave Bias Trim: Used to trim the bias for the slave DLL.
19:12	RW	0x37	MBIAS Master Bias Trim: Used to trim the bias for the master DLL.

Bit	Attr	Reset Value	Description
11	RW	0x0	TESTSW Test Switch: Selects the test signals of either the master DLL ('0') or the slave DLL ('1').
10:9	RW	0x0	ATC Analog Test Control: Selects the analog signal to be output on the DLL analog test output (test_out_a) when TESTEN is high (Output is Vss when TESTEN is low). The test output either comes from the master DLL or the slave DLL, depending on the setting of the test switch (TESTSW). Both master DLL and slave DLL output similar analog test signals. Valid settings for analog test control are: 00 = Filter output (Vc) 01 = Replica bias output for NMOS (Vbn) 10 = Replica bias output for PMOS (Vbp) 11 = Vdd 00

Bit	Attr	Reset Value	Description
8:6	RW	0x0	DTC Digital Test Control: Selects the digital signal to be output on the DLL digital test output (test_out_d[1]) when TESTEN is high (Output is '0' when TESTEN is low). Valid settings for master DLL (such as, when TESTSW = '0'): 000 = 0 output clock (clk_0) 001 = 90 output clock (clk_90) 010 = 180 output clock (clk_180) 011 = 270 output clock (clk_270) 100 = 360 internal clock (clk_360_int) 101 = Speed-up pulse (spdup) 110 = Slow-down pulse (slwdn) 111 = 0 MCTL logic clock (cclk_0) Valid settings for slave DLL (such as when TESTSW = '1'): 000 = Input DQS strobe (dqs) 001 = Input clock reference (clk_90_in) 010 = Internal feedback clock (clk_0_out) 011 = 90 output DQS_b strobe (dqsb_90) 100 = 90 output DQS strobe (dqs_90) 101 = Speed-up pulse (spdup) 110 = Slow-down pulse (slwdn) 111 = Auto-lock enable signal
5	RW	0x0	TESTEN Test Enable: Enables digital and analog test outputs selected by DTC and ATC respectively.
4:2	RW	0x0	IPUMP Charge Pump Current Trim: Used to trim charge pump current: 000 = maximum current 111 = minimum current
1:0	RW	0x0	DRES Delta Resistor Trim: Used to trim reference current versus resistor value variation: 00 = Rnom 01 = Rnom - 20% 1x = Rnom + 20%

DDR_PHYCTL_ACDLLCR
Address: Operational Base + offset (0x0014)
AC DLL Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DLLDIS DLL Disable: A disabled DLL is bypassed. Default ('0') is DLL enabled.
30	RW	0x1	DLLSRST DLL Soft Rest: Soft resets the AC DLL by driving the DLL soft reset pin.
29:20	RO	0x0	reserved
19	RW	0x0	SDLBMODE Slave DLL Loopback Mode: If this bit is set, the slave DLL is put in loopback mode in which there is no 90 degrees phase shift on read DQS/DQS#. This bit must be set when operating the byte PHYs in loopback mode such as during BIST loopback.
18	RW	0x0	ATESTEN Analog Test Enable: Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.
17:14	RW	0x0	SDPHASE Slave DLL Phase Trim: Selects the phase difference between the input clock and the corresponding output clock of the slave DLL. Valid settings: 0000 = 90 0001 = 72 0010 = 54 0011 = 36 0100 = 108 0101 = 90 0110 = 72 0111 = 54 1000 = 126 1001 = 108 1010 = 90 1011 = 72 1100 = 144 1101 = 126 1110 = 108 1111 = 90

Bit	Attr	Reset Value	Description
13:12	RW	0x0	SSTART Slave Auto Start-Up: Used to control how the slave DLL starts up relative to the master DLL locking: 0X = Slave DLL automatically starts up once the master DLL has achieved lock. 10 = The automatic startup of the slave DLL is disabled; the phase detector is disabled. 11 = The automatic startup of the slave DLL is disabled; the phase detector is enabled.
11:9	RW	0x0	MFWDLY Master Feed-Forward Delay Trim: Used to trim the delay in the master DLL feed-forward path: 000 = minimum delay 111 = maximum delay
8:6	RW	0x0	MFBDLY Master Feed-Back Delay Trim: Used to trim the delay in the master DLL feedback path: 000 = minimum delay 111 = maximum delay
5:3	RW	0x0	SFWDLY Slave Feed-Forward Delay Trim: Used to trim the delay in the slave DLL feed-forward path: 000 = minimum delay 111 = maximum delay
2:0	RW	0x0	SFBDLY Slave Feed-Back Delay Trim: Used to trim the delay in the slave DLL feedback path: 000 = minimum delay 111 = maximum delay

DDR_PHYCTL_PTR0

Address: Operational Base + offset (0x0018)

PHY Timing Register 0

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21:18	RW	0x8	tITMSRST ITM Soft Reset Time: Number of controller clock cycles that the ITM soft reset pin must remain asserted when the soft reset is applied to the ITMs. This must correspond to a value that is equal to or more than 8 controller clock cycles. Default value corresponds to 8 controller clock cycles.
17:6	RW	0xabe	tDLLLOCK DLL Lock Time: Number of clock cycles for the DLL to stabilize and lock, i.e. number of clock cycles from when the DLL reset pin is de-asserted to when the DLL has locked and is ready for use. Refer to the PHY databook for the DLL lock time. Default value corresponds to 5.12us at 533MHz.
5:0	RW	0x1b	tDLLSRST DLL Soft Reset Time: Number of controller clock cycles that the DLL soft reset pin must remain asserted when the soft reset is triggered through the PHY Initialization Register (PIR). This must correspond to a value that is equal to or more than 50ns or 8 controller clock cycles, whichever is bigger. Default value corresponds to 50ns at 533MHz.

DDR_PHYCTL_PTR1
Address: Operational Base + offset (0x001c)
PHY Timing Register 1

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26:19	RW	0xc0	tDINIT1 DRAM Initialization Time 1: DRAM initialization time corresponding to the following: DDR3 = CKE high time to first command (tRFC +10 ns or 5 tCK, whichever value is larger) DDR2 = CKE high time to first command (400 ns) DDR = CKE high time to first command (400 ns or 1 tCK) LPDDR2 = CKE low time with power and clock stable (100 ns) Default value corresponds to DDR3 360ns at 533MHz.
18:0	RW	0x4111d	tDINIT0 DRAM Initialization Time 0: DRAM initialization time corresponding to the following: DDR3 = CKE low time with power and clock stable (500 us) DDR2 = CKE low time with power and clock stable (200 us) DDR = CKE low time with power and clock stable (200 us) LPDDR = CKE high time to first command (200 us) LPDDR2 = CKE high time to first command (200 us) Default value corresponds to DDR3 500 us at 533MHz.

DDR_PHYCTL_PTR2
Address: Operational Base + offset (0x0020)
PHY Timing Register 2

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved

Bit	Attr	Reset Value	Description
26:17	RW	0x216	tDINIT3 DRAM Initialization Time 3: DRAM initialization time corresponding to the following: LPDDR2 = Time from ZQ initialization command to first command (1 us) Default value corresponds to the LPDDR2 1 us at 533MHz.
16:0	RW	0x1a072	tDINIT2 DRAM Initialization Time 2: DRAM initialization time corresponding to the following: DDR3 = Reset low time (200 us on power-up or 100 ns after power-up) LPDDR2 = Time from reset command to end of auto initialization (1 us + 10 us = 11 us) Default value corresponds to DDR3 200 us at 533MHz.

DDR_PHYCTL_ACIOCR
Address: Operational Base + offset (0x0024)
AC I/O Configuration Register

Bit	Attr	Reset Value	Description
31:30	RW	0x0	ACSR Address/Command Slew Rate (D3F I/O Only): Selects slew rate of the I/O for all address and command pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.
29	RW	0x1	RSTIOM SDRAM Reset I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for SDRAM Reset.
28	RW	0x1	RSTPDR SDRAM Reset Power Down Receiver: Powers down, when set, the input receiver on the I/O for SDRAM RST# pin.
27	RW	0x0	RSTPDD SDRAM Reset Power Down Driver: Powers down, when set, the output driver on the I/O for SDRAM RST# pin.

Bit	Attr	Reset Value	Description
26	RW	0x0	RSTODT SDRAM Reset On-Die Termination: Enables, when set, the on-die termination on the I/O for SDRAM RST# pin.
25:22	RW	0xf	RANKPDR Rank Power Down Receiver: Powers down, when set, the input receiver on the I/O CKE[3:0], ODT[3:0], and CS#[3:0] pins. RANKPDR[0] controls the power down for CKE[0], ODT[0], and CS#[0], RANKPDR[1] controls the power down for CKE[1], ODT[1], and CS#[1], and so on.
21:18	RW	0x0	CSPDD CS# Power Down Driver: Powers down, when set, the output driver on the I/O for CS#[3:0] pins. PDD[0] controls the power down for CS#[0], PDD[1] controls the power down for CS#[1], and so on. CKE and ODT driver power down is controlled by DSGCR register.
17:14	RW	0x0	RANKODT Rank On-Die Termination: Enables, when set, the on-die termination on the I/O for CKE[3:0], ODT[3:0], and CS#[3:0] pins. RANKODT[0] controls the on-die termination for CKE[0], ODT[0], and CS#[0], RANKODT[1] controls the on-die termination for CKE[1], ODT[1], and CS#[1], and so on.
13:11	RW	0x7	CKPDR CK Power Down Receiver: Powers down, when set, the input receiver on the I/O for CK[0], CK[1], and CK[2] pins, respectively
10:8	RW	0x0	CKPDD CK Power Down Driver: Powers down, when set, the output driver on the I/O for CK[0], CK[1], and CK[2] pins, respectively.
7:5	RW	0x0	CKODT CK On-Die Termination: Enables, when set, the on-die termination on the I/O for CK[0], CK[1], and CK[2] pins, respectively

Bit	Attr	Reset Value	Description
4	RW	0x1	ACPDR AC Power Down Receiver: Powers down, when set, the input receiver on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.
3	RW	0x0	ACPDD AC Power Down Driver: Powers down, when set, the output driver on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.
2	RW	0x0	ACODT Address/Command On-Die Termination: Enables, when set, the on-die termination on the I/O for RAS#, CAS#, WE#, BA[2:0], and A[15:0] pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.
1	RW	0x1	ACOE Address/Command Output Enable: Enables, when set, the output driver on the I/O for all address and command pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.
0	RW	0x0	ACIOM Address/Command I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for all address and command pins, as well as the optional DIMM PAR_IN pin and LPDDR TPD pin.

DDR_PHYCTL_DXCCR

Address: Operational Base + offset (0x0028)

DATX8 Common Configuration Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:12	RW	0x0	DXSR Data Slew Rate (D3F I/O Only): Selects slew rate of the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros.

Bit	Attr	Reset Value	Description
11:8	RW	0x8	DQSNRES DQS# Resistor: Selects the on-die pull-up/pull-down resistor for DQS# pins. Same encoding as DQSRES. Note: DQS# resistor must be connected for LPDDR2
7:4	RW	0x0	DQSRES DQS Resistor: Selects the on-die pull-down/pull-up resistor for DQS pins. DQSRES[3] selects pull-down (when set to 0) or pull-up (when set to 1). DQSRES[2:0] selects the resistor value as follows: 000 = Open: On-die resistor disconnected 001 = 688 ohms 010 = 611 ohms 011 = 550 ohms 100 = 500 ohms 101 = 458 ohms 110 = 393 ohms 111 = 344 ohms Note: DQS resistor must be connected for LPDDR2
3	RW	0x0	DXPDR Data Power Down Receiver: Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the PDR configuration bit of the individual DATX8.
2	RW	0x0	DXPDD Data Power Down Driver: Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the PDD configuration bit of the individual DATX8.
1	RW	0x0	DXIOM Data I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the IOM configuration bit of the individual DATX8.

Bit	Attr	Reset Value	Description
0	RW	0x0	DXODT Data On-Die Termination: Enables, when set, the on-die termination on the I/O for DQ, DM, and DQS/DQS# pins of all DATX8 macros. This bit is ORed with the ODT configuration bit of the individual DATX8.

DDR_PHYCTL_DSGCR

Address: Operational Base + offset (0x002c)

DDR System General Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x1	CKEOE SDRAM CKE Output Enable: Enables, when set, the output driver on the I/O for SDRAM CKE pins.
30	RW	0x1	RSTOE SDRAM Reset Output Enable: Enables, when set, the output driver on the I/O for SDRAM RST# pin.
29	RW	0x1	ODTOE SDRAM ODT Output Enable: Enables, when set, the output driver on the I/O for SDRAM ODT pins.
28	RW	0x1	CKOE SDRAM CK Output Enable: Enables, when set, the output driver on the I/O for SDRAM CK/CK# pins.
27	RW	0x1	TPDOE SDRAM TPD Output Enable (LPDDR Only): Enables, when set, the output driver on the I/O for SDRAM TPD pin.
26	RW	0x0	TPDPD DRAM TPD Power Down Driver (LPDDR Only): Powers down, when set, the output driver on the I/O for SDRAM TPD pin. Note that the power down of the receiver on the I/O for SDRAM TPD pin is controlled by ACIOCR[ACPDR] register bit.

Bit	Attr	Reset Value	Description
25	RW	0x1	NL2OE Non-LPDDR2 Output Enable: Enables, when set, the output driver on the I/O for non-LPDDR2 (ODT, RAS#, CAS#, WE#, and BA) pins. This may be used when a chip that is designed for both LPDDR2 and other DDR modes is being used in LPDDR2 mode. For these pins, the I/O output enable signal (OE) is an AND of this bit and the respective output enable bit in ACIOCR or DSGCR registers.
24	RW	0x0	NL2PD Non-LPDDR2 Power Down: Powers down, when set, the output driver and the input receiver on the I/O for non-LPDDR2 (ODT, RAS#, CAS#, WE#, and BA) pins. This may be used when a chip that is designed for both LPDDR2 and other DDR modes is being used in LPDDR2 mode. For these pins, the I/O power down signal (PDD or PDR) is an OR of this bit and the respective power-down bit in ACIOCR register.
23:20	RW	0x0	ODTPDD ODT Power Down Driver: Powers down, when set, the output driver on the I/O for ODT[3:0] pins. ODTPDD[0] controls the power down for ODT[0], ODTPDD[1] controls the power down for ODT[1], and so on.
19:16	RW	0x0	CKEPDD CKE Power Down Driver: Powers down, when set, the output driver on the I/O for CKE[3:0] pins. CKEPDD[0] controls the power down for CKE[0], CKEPDD[1] controls the power down for CKE[1], and so on.
15:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:8	RW	0x0	DQSGE DQS Gate Early: Specifies the number of clock cycles for which the DQS gating must be enabled earlier than its normal position. Only applicable when using PDQSR I/O cell, passive DQS gating and no drift compensation. This field is recommended to be set to zero for all DDR types other than LPDDR2. For LPDDR2 it should be set to (tDQSCKmax - tDQSCK) divide by clock period and rounded up.
7:5	RW	0x0	DQSGX DQS Gate Extension: Specifies the number of clock cycles for which the DQS gating must be extended beyond the normal burst length width. Only applicable when using PDQSR I/O cell, passive DQS gating and no drift compensation. This field is recommended to be set to zero for all DDR types other than LPDDR2. For LPDDR2 it should be set to (tDQSCKmax - tDQSCK) divide by clock period and rounded up.
4	RW	0x1	LPDLLPD Low Power DLL Power Down: Specifies if set that the PHY should respond to the DFI low power opportunity request and power down the DLL of the byte if the wakeup time request satisfies the DLL lock time.
3	RW	0x1	LPIOPD Low Power I/O Power Down: Specifies if set that the PHY should respond to the DFI low power opportunity request and power down the I/Os of the byte.
2	RW	0x1	ZUEN Impedance Update Enable: Specifies if set that the PHY should perform impedance calibration (update) whenever there is a controller initiated DFI update request. Otherwise the PHY will ignore an update request from the controller.

Bit	Attr	Reset Value	Description
1	RW	0x1	BDISEN Byte Disable Enable: Specifies if set that the PHY should respond to DFI byte disable request. Otherwise the byte disable from the DFI is ignored in which case bytes can only be disabled using the DXnGCR register.
0	RW	0x1	PUREN PHY Update Request Enable: Specifies if set, that the PHY should issue PHY-initiated DFI update request when there is DQS drift of more than 3/4 of a clock cycle within one continuous (back-to-back) read burst. By default the PHY issues PHY-initiated update requests and the controller should respond otherwise the PHY may return erroneous values. The option to disable it is provided only for silicon evaluation and testing

DDR_PHYCTL_DCR
Address: Operational Base + offset (0x0030)
DRAM Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	TPD Test Power Down (LPDDR Only): If set will place the DRAM in deep power down mode.
30	RW	0x0	RDIMM Registered DIMM: Indicates if set that a registered DIMM is used. In this case, the PUB increases the SDRAM write and read latencies (WL/RL) by 1. This only applies to PUB internal SDRAM transactions. Transactions generated by the controller must make its own adjustments to WL/RL when using a registered DIMM.

Bit	Attr	Reset Value	Description
29	RW	0x0	UDIMM Un-buffered DIMM Address Mirroring: Indicates if set that there is address mirroring on the second rank of an un-buffered DIMM (the rank connected to CS#[1]). In this case, the PUB re-scrambles the bank and address when sending mode register commands to the second rank. This only applies to PUB internal SDRAM transactions. Transactions generated by the controller must make its own adjustments when using an un-buffered DIMM. DCR[NOSRA] must be set if address mirroring is enabled.
28	RW	0x0	DDR2T DDR 2T Timing: Indicates if set that 2T timing should be used by PUB internally generated SDRAM transactions.
27	RW	0x0	NOSRA No Simultaneous Rank Access: Specifies if set that simultaneous rank access on the same clock cycle is not allowed. This means that multiple chip select signals should not be asserted at the same time. This may be required on some DIMM systems.
26:10	RO	0x0	reserved
9:8	RW	0x0	DDRTYPE DDR Type: Selects the DDR type for the specified DDR mode. Valid values for LPDDR2 are: 00 = LPDDR2-S4 01 = LPDDR2-S2 10 = LPDDR2-NVM 11 = Reserved
7	RW	0x0	MPRDQ Multi-Purpose Register (MPR) DQ (DDR3 Only): Specifies the value that is driven on non-primary DQ pins during MPR reads. Valid values are: 0 = Primary DQ drives out the data from MPR (0-1-0-1); non-primary DQs drive '0' 1 = Primary DQ and non-primary DQs all drive the same data from MPR (0-1-0-1)

Bit	Attr	Reset Value	Description
6:4	RW	0x0	PDQ Primary DQ (DDR3 Only): Specifies the DQ pin in a byte that is designated as a primary pin for Multi-Purpose Register (MPR) reads. Valid values are 0 to 7 for DQ[0] to DQ[7], respectively.
3	RW	0x1	DDR8BNK DDR 8-Bank: Indicates if set that the SDRAM used has 8 banks. tRPA = tRP+1 and tFAW are used for 8-bank DRAMs, other tRPA = tRP and no tFAW is used. Note that a setting of 1 for DRAMs that have fewer than 8 banks still results in correct functionality but less tighter DRAM command spacing for the parameters described here.
2:0	RW	0x3	DDRMD DDR Mode: SDRAM DDR mode. Valid values are: 000 = LPDDR (Mobile DDR) 001 = DDR 010 = DDR2 011 = DDR3 100 = LPDDR2 (Mobile DDR2) 101 = Reserved

DDR_PHYCTL_DTPR0
Address: Operational Base + offset (0x0034)
DRAM Timing Parameters Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	tCCD Read to read and write to write command delay. Valid values are: 0 = BL/2 for DDR2 and 4 for DDR3 1 = BL/2 + 1 for DDR2 and 5 for DDR3
30:25	RW	0x18	tRC Activate to activate command delay (same bank). Valid values are 2 to 42.
24:21	RW	0x4	tRRD Activate to activate command delay (different banks). Valid values are 1 to 8.
20:16	RW	0x12	tRAS Activate to precharge command delay. Valid values are 2 to 31.

Bit	Attr	Reset Value	Description
15:12	RW	0x6	tRCD Activate to read or write delay. Minimum time from when an activate command is issued to when a read or write to the activated row can be issued. Valid values are 2 to 11.
11:8	RW	0x6	tRP Precharge command period: The minimum time between a precharge command and any other command. Note that the Controller automatically derives tRPA for 8-bank DDR2 devices by adding 1 to tRP . Valid values are 2 to 11.
7:5	RW	0x3	tWTR Internal write to read command delay. Valid values are 1 to 6.
4:2	RW	0x3	tRTP Internal read to precharge command delay. Valid values are 2 to 6. Note that even though RTP does not apply to JEDEC DDR devices, this parameter must still be set to a minimum value of 2 for DDR because the Controller always uses the DDR2 equation, AL + BL/2 + max(RTP,2) - 2, to compute the read to precharge timing (which is BL/2 for JEDEC DDR).
1:0	RW	0x2	tMRD Load mode cycle time: The minimum time between a load mode register command and any other command. For DDR3 this is the minimum time between two load mode register commands. Valid values for DDR2 are 2 to 3. For DDR3, the value used for tMRD is 4 plus the value programmed in these bits, i.e. tMRD value for DDR3 ranges from 4 to 7.

DDR_PHYCTL_DTPR1

Address: Operational Base + offset (0x0038)

DRAM Timing Parameters Register 1

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved

Bit	Attr	Reset Value	Description
29:27	RW	0x1	tDQSCKmax Maximum DQS output access time from CK/CK# (LPDDR2 only). This value is used for implementing read-to-write spacing. Valid values are 1 to 7.
26:24	RW	0x1	tDQSCK DQS output access time from CK/CK# (LPDDR2 only). This value is used for computing the read latency. Valid values are 1 to 7.. This value is derived from the corresponding parameter in the SDRAM datasheet divided by the clock cycle time without rounding up. The fractional remainder is automatically adjusted for by data training in quarter clock cycle units. If data training is not performed then this fractional remainder must be converted to quarter clock cycle units and the gating registers (DXnDQSTR) adjusted accordingly.
23:16	RW	0x83	tRFC Refresh-to-Refresh: Indicates the minimum time, in clock cycles, between two refresh commands or between a refresh and an active command. This is derived from the minimum refresh interval from the datasheet, tRFC(min), divided by the clock cycle time. The default number of clock cycles is for the largest JEDEC tRFC(min) parameter value supported.
15:12	RO	0x0	reserved
11	RW	0x0	tRTODT Read to ODT delay (DDR3 only). Specifies whether ODT can be enabled immediately after the read post-amble or one clock delay has to be added. Valid values are: 0 = ODT may be turned on immediately after read post-amble 1 = ODT may not be turned on until one clock after the read post-amble If tRTODT is set to 1, then the read-to-write latency is increased by 1 if ODT is enabled.

Bit	Attr	Reset Value	Description
10:9	RW	0x0	tMOD Load mode update delay (DDR3 only). The minimum time between a load mode register command and a non-load mode register command. Valid values are: 00 = 12 01 = 13 10 = 14 11 = 15
8:3	RW	0x12	tFAW 4-bank activate period. No more than 4-bank activate commands may be issued in a given tFAW period. Only applies to 8-bank devices. Valid values are 2 to 31.
2	RW	0x0	tRTW Read to Write command delay. Valid values are: 0 = standard bus turn around delay 1 = add 1 clock to standard bus turn around delay This parameter allows the user to increase the delay between issuing Write commands to the SDRAM when preceded by Read commands. This provides an option to increase bus turn-around margin for high frequency systems.
1:0	RW	0x0	tAOND_tAOFD ODT turn-on/turn-off delays (DDR2 only). The delays are in clock cycles. Valid values are: 00 = 2/2.5 01 = 3/3.5 10 = 4/4.5 11 = 5/5.5 Most DDR2 devices utilize a fixed value of 2/2.5. For non-standard SDRAMs, the user must ensure that the operational Write Latency is always greater than or equal to the ODT turn-on delay. For example, a DDR2 SDRAM with CAS latency set to 3 and CAS additive latency set to 0 has a Write Latency of 2. Thus 2/2.5 can be used, but not 3/3.5 or higher.

DDR_PHYCTL_DTPR2
Address: Operational Base + offset (0x003c)
DRAM Timing Parameters Register 2

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:19	RW	0x200	tDLLK DLL locking time. Valid values are 2 to 1023.
18:15	RW	0x3	tCKE CKE minimum pulse width. Also specifies the minimum time that the SDRAM must remain in power down or self refresh mode. For DDR3 this parameter must be set to the value of tCKESR which is usually bigger than the value of tCKE. Valid values are 2 to 15.
14:10	RW	0x08	tXP Power down exit delay. The minimum time between a power down exit command and any other command. This parameter must be set to the maximum of the various minimum power down exit delay parameters specified in the SDRAM datasheet, i.e. max(tXP , tXARD, tXARDS) for DDR2 and max(tXP , tXPDLL) for DDR3. Valid values are 2 to 31.
9:0	RW	0x0c8	tXS Self refresh exit delay. The minimum time between a self refresh exit command and any other command. This parameter must be set to the maximum of the various minimum self refresh exit delay parameters specified in the SDRAM datasheet, i.e. max(tXSNR, tXSRD) for DDR2 and max(tXS, tXSDLL) for DDR3. Valid values are 2 to 1023.

DDR_PHYCTL_MR0
Address: Operational Base + offset (0x0040)
Mode Register 0

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	PD Power-Down Control: Controls the exit time for power-down modes. Refer to SDRAM datasheet for details on power-down modes. Valid values are: 0 = Slow exit (DLL off) 1 = Fast exit (DLL on)
11:9	RW	0x5	WR Write Recovery: This is the value of the write recovery in clock cycles. It is calculated by dividing the datasheet write recovery time, tWR (ns) by the datasheet clock cycle time, tCK (ns) and rounding up a non-integer value to the next integer. Valid values are: 001 = 5 010 = 6 011 = 7 100 = 8 101 = 10 110 = 12 All other settings are reserved and should not be used. NOTE: tWR (ns) is the time from the first SDRAM positive clock edge after the last data-in pair of a write command, to when a precharge of the same bank can be issued.
8	RW	0x0	DR DLL Reset: Writing a '1' to this bit will reset the SDRAM DLL. This bit is self-clearing, i.e. it returns back to '0' after the DLL reset has been issued.
7	RW	0x0	TM Operating Mode: Selects either normal operating mode (0) or test mode (1). Test mode is reserved for the manufacturer and should not be used.

Bit	Attr	Reset Value	Description
6:4	RW	0x5	CL_1 CAS Latency: The delay, in clock cycles, between when the SDRAM registers a read command to when data is available. Valid values are: 0010 = 5 0100 = 6 0110 = 7 1000 = 8 1010 = 9 1100 = 10 1110 = 11 All other settings are reserved and should not be used.
3	RW	0x0	BT Burst Type: Indicates whether a burst is sequential (0) or interleaved (1).
2	RW	0x0	CL_0 CAS Latency: merged with bit6-4
1:0	RW	0x2	BL Burst Length: Determines the maximum number of column locations that can be accessed during a given read or write command. Valid values are: Valid values for DDR3 are: 00 = 8 (Fixed) 01 = 4 or 8 (On the fly) 10 = 4 (Fixed) 11 = Reserved

DDR_PHYCTL_MR1
Address: Operational Base + offset (0x0044)
Mode Register 1

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12	RW	0x0	QOFF Output Enable/Disable: When '0' all outputs function normal; when '1' all SDRAM outputs are disabled removing output buffer current. This feature is intended to be used for IDD characterization of read current and should not be used in normal operation.

Bit	Attr	Reset Value	Description
11	RW	0x0	TDQS Termination Data Strobe: When enabled ('1') TDQS provides additional termination resistance outputs that may be useful in some system configurations. Refer to the SDRAM datasheet for details.
10	RO	0x0	reserved
9	RW	0x0	RTT_2 On Die Termination: Selects the effective resistance for SDRAM on die termination. Valid values are: 000 = ODT disabled 001 = RZQ/4 010 = RZQ/2 011 = RZQ/6 100 = RZQ/12 101 = RZQ/8 All other settings are reserved and should not be used.
8	RO	0x0	reserved
7	RW	0x0	LEVEL Write Leveling Enable: Enables write-leveling when set.
6	RW	0x0	RTT_1 On Die Termination: merged with bit9.
5	RW	0x0	DIC_1 Output Driver Impedance Control: Controls the output drive strength. Valid values are: 00 = Reserved for RZQ/6 01 = RZQ7 10 = Reserved 11 = Reserved
4:3	RW	0x0	AL Posted CAS Additive Latency: Setting additive latency that allows read and write commands to be issued to the SDRAM earlier than normal (refer to SDRAM datasheet for details). Valid values are: 00 = 0 (AL disabled) 01 = CL - 1 10 = CL - 2 11 = Reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	RTT_0 On Die Termination: merged with bit9.
1	RW	0x0	DIC_0 Output Driver Impedance Control: Controls the output drive strength. Merged with bit5.
0	RW	0x0	DE DLL Enable/Disable: Enable (0) or disable (1) the DLL. DLL must be enabled for normal operation.

DDR_PHYCTL_MR2

Address: Operational Base + offset (0x0048)

Mode Register 2

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:9	RW	0x0	RTTWR Dynamic ODT: Selects RTT for dynamic ODT. Valid values are: 00 = Dynamic ODT off 01 = RZQ/4 10 = RZQ/2 11 = Reserved
8	RO	0x0	reserved
7	RW	0x0	SRT Self-Refresh Temperature Range: Selects either normal ('0') or extended ('1') operating temperature range during self-refresh.
6	RW	0x0	ASR Auto Self-Refresh: When enabled (‘1’), SDRAM automatically provide self-refresh power management functions for all supported operating temperature values. Otherwise the SRT bit must be programmed to indicate the temperature range.

Bit	Attr	Reset Value	Description
5:3	RW	0x0	CWL CAS Write Latency: The delay, in clock cycles, between when the SDRAM registers a write command to when write data is available. Valid values are: 000 = 5 (tCK = 2.5ns) 001 = 6 (2.5ns > tCK = 1.875ns) 010 = 7 (1.875ns > tCK = 1.5ns) 011 = 8 (1.5ns > tCK = 1.25ns) All other settings are reserved and should not be used
2:0	RW	0x0	PASR Partial Array Self Refresh: Specifies that data located in areas of the array beyond the specified location will be lost if self refresh is entered. Valid settings for 4 banks are: 000 = Full Array 001 = Half Array (BA[1:0] = 00 & 01) 010 = Quarter Array (BA[1:0] = 00) 011 = Not defined 100 = 3/4 Array (BA[1:0] = 01, 10, & 11) 101 = Half Array (BA[1:0] = 10 & 11) 110 = Quarter Array (BA[1:0] = 11) 111 = Not defined Valid settings for 8 banks are: 000 = Full Array 001 = Half Array (BA[2:0] = 000, 001, 010 & 011) 010 = Quarter Array (BA[2:0] = 000, 001) 011 = 1/8 Array (BA[2:0] = 000) 100 = 3/4 Array (BA[2:0] = 010, 011, 100, 101, 110 & 111) 101 = Half Array (BA[2:0] = 100, 101, 110 & 111) 110 = Quarter Array (BA[2:0] = 110 & 111) 111 = 1/8 Array (BA[2:0] 111)

DDR_PHYCTL_MR3

Address: Operational Base + offset (0x004c)

Mode Register 3

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	MPR Multi-Purpose Register Enable: Enables, if set, that read data should come from the Multi-Purpose Register. Otherwise read data come from the DRAM array.
1:0	RW	0x0	MPRLOC Multi-Purpose Register (MPR) Location: Selects MPR data location: Valid value are: 00 = Predefined pattern for system calibration All other settings are reserved and should not be used.

DDR_PHYCTL_ODTCR
Address: Operational Base + offset (0x0050)
ODT Configuration Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:20	RW	0x2	WRODT1 Write ODT: Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a write command is sent to rank 0. WRODT0, WRODT1 specify ODT settings when a write is to rank 0, rank 1 respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 1. Default is to enable ODT only on rank being written to.
19:16	RW	0x1	WRODT0 Write ODT: Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a write command is sent to rank 0. WRODT0, WRODT1 specify ODT settings when a write is to rank 0, rank 1 respectively. The four bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 1. Default is to enable ODT only on rank being written to.
15:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:4	RW	0x0	RDODT1 Read ODT: Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a read command is sent to rank 1. RDODT0, RDODT1 specify ODT settings when a read is to rank 0, and rank 1, respectively. The two bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 1. Default is to disable ODT during reads.
3:0	RW	0x0	RDODT0 Read ODT: Specifies whether ODT should be enabled ('1') or disabled ('0') on each of the up to four ranks when a read command is sent to rank 0. RDODT0, RDODT1 specify ODT settings when a read is to rank 0, and rank 1, respectively. The two bits of each field each represent a rank, the LSB being rank 0 and the MSB being rank 1. Default is to disable ODT during reads.

DDR_PHYCTL_DTAR
Address: Operational Base + offset (0x0054)
Data Training Address Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DTMPR Data Training Using MPR (DDR3 Only): Specifies, if set, that data-training should use the SDRAM Multi-Purpose Register (MPR) register. Otherwise data-training is performed by first writing to some locations in the SDRAM and then reading them back.
30:28	RW	0x0	DTBANK Data Training Bank Address: Selects the SDRAM bank address to be used during data training.
27:12	RW	0x0000	DTROW Data Training Row Address: Selects the SDRAM row address to be used during data training.

Bit	Attr	Reset Value	Description
11:0	RW	0x000	DTCOL Data Training Column Address: Selects the SDRAM column address to be used during data training. The lower four bits of this address must always be '0000'

DDR_PHYCTL_DTDR0
Address: Operational Base + offset (0x0058)
Data Training Data Register 0

Bit	Attr	Reset Value	Description
31:24	RW	0xdd	DTBYTE3 Data Training Data: The fourth 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
23:16	RW	0x22	DTBYTE2 Data Training Data: The third 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
15:8	RW	0xee	DTBYTE1 Data Training Data: The second 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
7:0	RW	0x11	DTBYTE0 Data Training Data: The first 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.

DDR_PHYCTL_DTDR1
Address: Operational Base + offset (0x005c)
Data Training Data Register 1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:24	RW	0x77	DTBYTE7 Data Training Data: The eighth4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
23:16	RW	0x88	DTBYTE6 Data Training Data: The seventh 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
15:8	RW	0xbb	DTBYTE5 Data Training Data: The sixth 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.
7:0	RW	0x44	DTBYTE4 Data Training Data: The fifth 4 bytes of data used during data training. This same data byte is used for each Byte Lane. Default sequence is a walking 1 while toggling data every data cycle.

DDR_PHYCTL_DCUAR

Address: Operational Base + offset (0x00c0)

DCU Address Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11	RW	0x0	ATYPE Access Type: Specifies the type of access to be performed using this address. Valid values are: 0 = Write access 1 = Read access

Bit	Attr	Reset Value	Description
10	RW	0x0	INCA Increment Address: Specifies, if set, that the cache address specified in WADDR and SADDR should be automatically incremented after each access of the cache. The increment happens in such a way that all the slices of a selected word are first accessed before going to the next word.
9:8	RW	0x0	CSEL Cache Select: Selects the cache to be accessed. Valid values are: 00 = Command cache 01 = Expected data cache 10 = Read data cache 11 = Reserved
7:4	RW	0x0	CSADDR Cache Slice Address: Address of the cache slice to be accessed.
3:0	RW	0x0	CWADDR Cache Word Address: Address of the cache word to be accessed.

DDR_PHYCTL_DCUDR
Address: Operational Base + offset (0x00c4)
DCU Data Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CDATA Cache Data: Data to be written to or read from a cache. This data corresponds to the cache word slice specified by the DCU Address Register.

DDR_PHYCTL_DCURR
Address: Operational Base + offset (0x00c8)
DCU Run Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	XCEN Expected Compare Enable: Indicates if set that read data coming back from the SDRAM should be should be compared with the expected data.

Bit	Attr	Reset Value	Description
22	RW	0x0	RCEN Read Capture Enable: Indicates if set that read data coming back from the SDRAM should be captured into the read data cache.
21	RW	0x0	SCOF Stop Capture On Full: Specifies if set that the capture of read data should stop when the capture cache is full.
20	RW	0x0	SONF Stop On Nth Fail: Specifies if set that the execution of commands and the capture of read data should stop when there are N read data failures. The number of failures is specified by NFAIL. Otherwise commands execute until the end of the program or until manually stopped using a STOP command.
19:12	RW	0x00	NFAIL Number of Failures: Specifies the number of failures after which the execution of commands and the capture of read data should stop if SONF bit of this register is set. Execution of commands and the capture of read data will stop after (NFAIL+1) failures if SONF is set.
11:8	RW	0x0	EADDR End Address: Cache word address where the execution of command should end.
7:4	RW	0x0	SADDR Start Address: Cache word address where the execution of commands should begin.
3:0	RW	0x0	DINST DCU Instruction: Selects the DCU command to be executed: Valid values are: 0000 = NOP: No operation 0001 = Run: Triggers the execution of commands in the command cache. 0010 = Stop: Stops the execution of commands in the command cache. 0011 = Stop Loop: Stops the execution of an infinite loop in the command cache. 0100 = Reset: Resets all DCU run time registers. 0101 - 1111 Reserved

DDR_PHYCTL_DCULR
Address: Operational Base + offset (0x00cc)
DCU Loop Register

Bit	Attr	Reset Value	Description
31:28	RW	0x0	XLEADDR Expected Data Loop End Address: The last expected data cache word address that contains valid expected data. Expected data should looped between 0 and this address.
27:18	RO	0x0	reserved
17	RW	0x0	IDA Increment DRAM Address: Indicates if set that DRAM addresses should be incremented every time a DRAM read/write command inside the loop is executed.
16	RW	0x0	LINF Loop Infinite: Indicates if set that the loop should be executed indefinitely until stopped by the STOP command. Otherwise the loop is execute LCNT times.
15:8	RW	0x00	LCNT Loop Count: The number of times that the loop should be executed if LINF is not set.
7:4	RW	0x0	LEADDR Loop End Address: Command cache word address where the loop should end.
3:0	RW	0x0	LSADDR Loop Start Address: Command cache word address where the loop should start.

DDR_PHYCTL_DCUGCR
Address: Operational Base + offset (0x00d0)
DCU General Configuration Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RCSW Read Capture Start Word: The capture and compare of read data should start after Nth word. For example setting this value to 12 will skip the first 12 read data.

DDR_PHYCTL_DCUTPR

Address: Operational Base + offset (0x00d4)
DCU Timing Parameters Registers

Bit	Attr	Reset Value	Description
31:24	RW	0x00	tDCUT3 DCU Generic Timing Parameter 3.
23:16	RW	0x00	tDCUT2 DCU Generic Timing Parameter 2.
15:8	RW	0x00	tDCUT1 DCU Generic Timing Parameter 1.
7:0	RW	0x00	tDCUT0 DCU Generic Timing Parameter 0.

DDR_PHYCTL_DCUSR0
Address: Operational Base + offset (0x00d8)
DCU Status Register 0

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RO	0x0	CFULL Capture Full: Indicates if set that the capture cache is full.
1	RO	0x0	CFAIL Capture Fail: Indicates if set that at least one read data word has failed.
0	RO	0x0	RDONE Run Done: Indicates if set that the DCU has finished executing the commands in the command cache. This bit is also set to indicate that a STOP command has successfully been executed and command execution has stopped.

DDR_PHYCTL_DCUSR1
Address: Operational Base + offset (0x00dc)
DCU Status Register 1

Bit	Attr	Reset Value	Description
31:24	RO	0x00	LPCNT Loop Count: Indicates the value of the loop count. This is useful when the program has stooped because of failures to assess how many reads were executed before first fail.
23:16	RO	0x00	FLCND Fail Count: Number of read words that have failed.

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	RDCNT Read Count: Number of read words returned from the SDRAM.

DDR_PHYCTL_BISTRR
Address: Operational Base + offset (0x0100)
BIST Run Register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:23	RW	0x0	BCKSEL BIST CK Select: Selects the CK to be used for capturing loopback data on the address/command lane. Valid values are: 000 = CK[0] 001 = CK[1] 010 = CK[2] 011 = Reserved 100 = CK#[0] 101 = CK#[1] 110 = CK#[2] 111 = Reserved
22:19	RW	0x0	BDXSEL BIST DATX8 Select: Select the byte lane for comparison of loopback/read data. Valid values are 0 to 8.
18:17	RW	0x0	BDPAT BIST Data Pattern: Selects the data pattern used during BIST. Valid values are: 00 = Walking 0 01 = Walking 1 10 = LFSR-based pseudo-random 11 = User programmable
16	RW	0x0	BDMEN BIST Data Mask Enable: Enables if set that the data mask BIST should be included in the BIST run, i.e. data pattern generated and loopback data compared. This is valid only for loopback mode.
15	RW	0x0	BACEN BIST AC Enable: Enables the running of BIST on the address/command lane PHY. This bit is exclusive with BDXEN, i.e. both cannot be set to '1' at the same time.

Bit	Attr	Reset Value	Description
14	RW	0x0	BDXEN BIST DATX8 Enable: Enables the running of BIST on the data byte lane PHYs. This bit is exclusive with BACEN, i.e. both cannot be set to '1' at the same time.
13	RW	0x0	BSONF BIST Stop On Nth Fail: Specifies if set that the BIST should stop when an nth data word or address/command comparison error has been encountered.
12:5	RW	0x00	NFAIL Number of Failures: Specifies the number of failures after which the execution of commands and the capture of read data should stop if BSONF bit of this register is set. Execution of commands and the capture of read data will stop after (NFAIL+1) failures if BSONF is set.
4	RW	0x0	BINF BIST Infinite Run: Specifies if set that the BIST should be run indefinitely until when it is either stopped or a failure has been encountered. Otherwise BIST is run until number of BIST words specified in the BISTWCR register has been generated.
3	RW	0x0	BMODE BIST Mode: Selects the mode in which BIST is run. Valid values are: 0 = Loopback mode: Address, commands and data loop back at the PHY I/Os. 1 = DRAM mode: Address, commands and data go to DRAM for normal memory accesses.
2:0	RW	0x0	BINST BIST Instruction: Selects the BIST instruction to be executed: Valid values are: 000 = NOP: No operation 001 = Run: Triggers the running of the BIST. 010 = Stop: Stops the running of the BIST. 011 = Reset: Resets all BIST run-time registers, such as error counters. 100 - 111 Reserved

DDR_PHYCTL_BISTMSKR0
Address: Operational Base + offset (0x0104)
BIST Mask Register 0

Bit	Attr	Reset Value	Description
31:28	RW	0x0	ODTMSK Mask bit for each of the up to 4 ODT bits.
27:24	RW	0x0	CSMSK Mask bit for each of the up to 4 CS# bits.
23:20	RW	0x0	CKEMSK Mask bit for each of the up to 4 CKE bits.
19	RW	0x0	WEMSK Mask bit for the WE#.
18:16	RW	0x0	BAMSK Mask bit for each of the up to 3 bank address bits.
15:0	RW	0x0000	AMSK Mask bit for each of the up to 16 address bits.

DDR_PHYCTL_BISTMSKR1
Address: Operational Base + offset (0x0108)
BIST Mask Register 1

Bit	Attr	Reset Value	Description
31	RW	0x0	TPDMSK Mask bit for the TPD. LPDDR Only
30	RW	0x0	PARMSK Mask bit for the PAR_IN. Only for DIMM parity support.
29:20	RO	0x0	reserved
19	RW	0x0	CASMSK Mask bit for the CAS.
18	RW	0x0	RASMSK Mask bit for the RAS.
17:16	RW	0x0	DMMSK Mask bit for the data mask (DM) bit.
15:0	RW	0x0000	DQMSK Mask bit for each of the 8 data (DQ) bits.

DDR_PHYCTL_BISTWCR
Address: Operational Base + offset (0x010c)
BIST Word Count Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0020	BWCNT BIST Word Count: Indicates the number of words to generate during BIST. This must be a multiple of DRAM burst length (BL) divided by 2, e.g. for BL=8, valid values are 4, 8, 12, 16, and so on.

DDR_PHYCTL_BISTLSR
Address: Operational Base + offset (0x0110)
BIST LFSR Seed Register

Bit	Attr	Reset Value	Description
31:0	RW	0x1234abcd	SEED LFSR seed for pseudo-random BIST patterns.

DDR_PHYCTL_BISTAR0
Address: Operational Base + offset (0x0114)
BIST Address Register 0

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:28	RW	0x0	BBANK BIST Bank Address: Selects the SDRAM bank address to be used during BIST.
27:12	RW	0x0000	BROW BIST Row Address: Selects the SDRAM row address to be used during BIST.
11:0	RW	0x000	BCOL BIST Column Address: Selects the SDRAM column address to be used during BIST. The lower bits of this address must be "0000" for BL16, "000" for BL8, "00" for BL4 and "0" for BL2.

DDR_PHYCTL_BISTAR1
Address: Operational Base + offset (0x0118)
BIST Address Register 1

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:4	RW	0x000	BAINC BIST Address Increment: Selects the value by which the SDRAM address is incremented for each write/read access. This value must be at the beginning of a burst boundary, i.e. the lower bits must be "0000" for BL16, "00" for BL8, "00" for BL4 and "0" for BL2.
3:2	RW	0x3	BMRANK BIST Maximum Rank: Specifies the maximum SDRAM rank to be used during BIST. The default value is set to maximum ranks minus 1. Example default shown here is for a 4-rank system
1:0	RW	0x0	BRANK BIST Rank: Selects the SDRAM rank to be used during BIST. Valid values range from 0 to maximum ranks minus 1.

DDR_PHYCTL_BISTAR2
Address: Operational Base + offset (0x011c)
BIST Address Register 2

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:28	RW	0x7	BMBANK BIST Maximum Bank Address: Specifies the maximum SDRAM bank address to be used during BIST before the address increments to the next rank.
27:12	RW	0xffff	BMROW BIST Maximum Row Address: Specifies the maximum SDRAM row address to be used during BIST before the address increments to the next bank.
11:0	RW	0xfff	BMCOL BIST Maximum Column Address: Specifies the maximum SDRAM column address to be used during BIST before the address increments to the next row.

DDR_PHYCTL_BISTUDPR
Address: Operational Base + offset (0x0120)
BIST User Data Pattern Register

Bit	Attr	Reset Value	Description
31:16	RW	0xffff	BUDP1 BIST User Data Pattern 1: Data to be applied on odd DQ pins during BIST.
15:0	RW	0x0000	BUDP0 BIST User Data Pattern 0: Data to be applied on even DQ pins during BIST.

DDR_PHYCTL_BISTGSR

Address: Operational Base + offset (0x0124)

BIST General Status Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	CASBER CAS Bit Error: Indicates the number of bit errors on CAS.
29:28	RO	0x0	RASBER RAS Bit Error: Indicates the number of bit errors on RAS.
27:24	RO	0x0	DMBER DM Bit Error: Indicates the number of bit errors on data mask (DM) bit. DMBER[1:0] are for the first DM beat, and DMBER[3:2] are for the second DM beat.
23:22	RO	0x0	TPDBER TPD Bit Error (LPDDR Only): Indicates the number of bit errors on TPD.
21:20	RO	0x0	PARBER PAR_IN Bit Error (DIMM Only): Indicates the number of bit errors on PAR_IN.
19:3	RO	0x0	reserved
2	RO	0x0	BDXERR BIST Data Error: indicates if set that there is a data comparison error in the byte lane.
1	RO	0x0	BACERR BIST Address/Command Error: indicates if set that there is a data comparison error in the address/command lane.
0	RO	0x0	BDONE BIST Done: Indicates if set that the BIST has finished executing. This bit is reset to zero when BIST is triggered.

DDR_PHYCTL_BISTWER
Address: Operational Base + offset (0x0128)
BIST Word Error Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	DXWER Byte Word Error: Indicates the number of word errors on the byte lane. An error on any bit of the data bus including the data mask bit increments the error count.
15:0	RO	0x0000	ACWER Address/Command Word Error: Indicates the number of word errors on the address/command lane. An error on any bit of the address/command bus increments the error count.

DDR_PHYCTL_BISTBER0
Address: Operational Base + offset (0x012c)
BIST Bit Error Register 0

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	ABER Address Bit Error: Each group of two bits indicate the bit error count on each of the up to 16 address bits. [1:0] is the error count for A[0], [3:2] for A[1], and so on.

DDR_PHYCTL_BISTBER1
Address: Operational Base + offset (0x0130)
BIST Bit Error Register 1

Bit	Attr	Reset Value	Description
31:24	RO	0x00	ODTBER ODT Bit Error: Each group of two bits indicates the bit error count on each of the up to 4 ODT bits. [1:0] is the error count for ODT[0], [3:2] for ODT[1], and so on.
23:16	RO	0x00	CSBER CS# Bit Error: Each group of two bits indicates the bit error count on each of the up to 4 CS# bits. [1:0] is the error count for CS#[0], [3:2] for CS#[1], and so on.

Bit	Attr	Reset Value	Description
15:8	RO	0x00	CKEBER CKE Bit Error: Each group of two bits indicates the bit error count on each of the up to 4 CKE bits. [1:0] is the error count for CKE[0], [3:2] for CKE[1], and so on.
7:6	RO	0x0	WEBER WE# Bit Error: Indicates the number of bit errors on WE#.
5:0	RO	0x00	BABER Bank Address Bit Error: Each group of two bits indicates the bit error count on each of the up to 3 bank address bits. [1:0] is the error count for BA[0], [3:2] for BA[1], and so on.

DDR_PHYCTL_BISTBER2
Address: Operational Base + offset (0x0134)
BIST Bit Error Register 2

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DQBER Data Bit Error: The first 16 bits indicate the error count for the first data beat (i.e. the data driven out on DQ[7:0] on the rising edge of DQS). The second 16 bits indicate the error on the second data beat (i.e. the error count of the data driven out on DQ[7:0] on the falling edge of DQS). For each of the 16-bit group, the first 2 bits are for DQ[0], the second for DQ[1], and so on.

DDR_PHYCTL_BISTWCSR
Address: Operational Base + offset (0x0138)
BIST Word Count Status Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	DXWCNT Byte Word Count: Indicates the number of words received from the byte lane.
15:0	RO	0x0000	ACWCNT Address/Command Word Count: Indicates the number of words received from the address/command lane.

DDR_PHYCTL_BISTFWR0

Address: Operational Base + offset (0x013c)
BIST Fail Word Register 0

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RO	0x0	ODTWEBS Bit status during a word error for each of the up to 4 ODT bits.
27:24	RO	0x0	CSWEBS Bit status during a word error for each of the up to 4 CS# bits.
23:20	RO	0x0	CKEWEBS Bit status during a word error for each of the up to 4 CKE bits.
19	RO	0x0	WEWEBS Bit status during a word error for the WE#.
18:16	RO	0x0	BAWEBS Bit status during a word error for each of the up to 3 bank address bits.
15:0	RO	0x0000	AWEBS Bit status during a word error for each of the up to 16 address bits.

DDR_PHYCTL_BISTFWR1
Address: Operational Base + offset (0x0140)
BIST Fail Word Register 1

Bit	Attr	Reset Value	Description
31	RO	0x0	TPDWEBS Bit status during a word error for the TPD. LPDDR Only
30	RO	0x0	PARWEBS Bit status during a word error for the PAR_IN. Only for DIMM parity support.
29:20	RO	0x0	reserved
19	RO	0x0	CASWEBS Bit status during a word error for the CAS.
18	RO	0x0	RASWEBS Bit status during a word error for the RAS.
17:16	RO	0x0	DMWEBS Bit status during a word error for the data mask (DM) bit. DMWEBS [0] is for the first DM beat, and DMWEBS [1] is for the second DM beat.

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	DQWEBS Bit status during a word error for each of the 8 data (DQ) bits. The first 8 bits indicate the status of the first data beat (i.e. the status of the data driven out on DQ[7:0] on the rising edge of DQS). The second 8 bits indicate the status of the second data beat (i.e. the status of the data driven out on DQ[7:0] on the falling edge of DQS). For each of the 8-bit group, the first bit is for DQ[0], the second bit is for DQ[1], and so on.

DDR_PHYCTL_ZQ0CR0
Address: Operational Base + offset (0x0180)
ZQ 0 Impedance Control Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	ZQPD ZQ Power Down: Powers down, if set, the PZQ cell.
30	RW	0x0	ZCAL Impedance Calibration Trigger: A write of '1' to this bit triggers impedance calibration to be performed by the impedance control logic. The impedance calibration trigger bit is self-clearing and returns back to '0' when the calibration is complete.
29	RW	0x0	ZCALBYP Impedance Calibration Bypass: Disables, if set, impedance calibration of this ZQ control block when impedance calibration is triggered globally using the ZCAL bit of PIR. Impedance calibration of this ZQ block may be triggered manually using ZCAL.
28	RW	0x0	ZDEN Impedance Over-ride Enable: When this bit is set, it allows users to directly drive the impedance control using the dataprogrammed in the ZQDATA field. Otherwise, the control is generated automatically by the impedance control logic

Bit	Attr	Reset Value	Description
27:0	RW	0x000014a	ZDATA Impedance Over-Ride Data: Data used to directly drive the impedance control. ZDATA field mapping for D3R I/Os is as follows: ZDATA[27:20] is reserved and returns zeros on reads ZDATA[19:15] is used to select the pull-up on-die termination impedance ZDATA[14:10] is used to select the pull-down on-die termination impedance ZDATA[9:5] is used to select the pull-up output impedance ZDATA[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ0CR1
Address: Operational Base + offset (0x0184)
ZQ 0 Impedance Control Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x7b	ZPROG Impedance Divide Ratio: Selects the external resistor divide ratio to be used to set the output impedance and the on-die termination as follows: ZPROG[7:4] = On-die termination divide select ZPROG[3:0] = Output impedance divide select

DDR_PHYCTL_ZQ0SR0
Address: Operational Base + offset (0x0188)
ZQ 0 Impedance Status Register 0

Bit	Attr	Reset Value	Description
31	RO	0x0	ZDONE Impedance Calibration Done: Indicates that impedance calibration has completed.
30	RO	0x0	ZERR Impedance Calibration Error: If set, indicates that there was an error during impedance calibration.

Bit	Attr	Reset Value	Description
29:28	RO	0x0	reserved
27:0	RO	0x0000000	ZCTRL Impedance Control: Current value of impedance control. ZCTRL field mapping for D3R I/Os is as follows: ZCTRL[27:20] is reserved and returns zeros on reads ZCTRL[19:15] is used to select the pull-up on-die termination impedance ZCTRL[14:10] is used to select the pull-down on-die termination impedance ZCTRL[9:5] is used to select the pull-up output impedance ZCTRL[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ0SR1
Address: Operational Base + offset (0x018c)
ZQ 0 Impedance Status Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	OPU On-die termination (ODT) pull-up calibration status. Similar status encodings as ZPD.
5:4	RO	0x0	OPD On-die termination (ODT) pull-down calibration status. Similar status encodings as ZPD.
3:2	RO	0x0	ZPU Output impedance pull-up calibration status. Similar status encodings as ZPD.
1:0	RO	0x0	ZPD Output impedance pull-down calibration status. Valid status encodings are: 00 = Completed with no errors 01 = Overflow error 10 = Underflow error 11 = Calibration in progress

DDR_PHYCTL_ZQ1CR0

Address: Operational Base + offset (0x0190)
ZQ 1 Impedance Control Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	ZQPD ZQ Power Down: Powers down, if set, the PZQ cell.
30	RW	0x0	ZCAL Impedance Calibration Trigger: A write of '1' to this bit triggers impedance calibration to be performed by the impedance control logic. The impedance calibration trigger bit is self-clearing and returns back to '0' when the calibration is complete.
29	RW	0x0	ZCALBYP Impedance Calibration Bypass: Disables, if set, impedance calibration of this ZQ control block when impedance calibration is triggered globally using the ZCAL bit of PIR. Impedance calibration of this ZQ block may be triggered manually using ZCAL.
28	RW	0x0	ZDEN Impedance Over-ride Enable: When this bit is set, it allows users to directly drive the impedance control using the data programmed in the ZQDATA field. Otherwise, the control is generated automatically by the impedance control logic
27:0	RW	0x000014a	ZDATA Impedance Over-Ride Data: Data used to directly drive the impedance control. ZDATA field mapping for D3R I/Os is as follows: ZDATA[27:20] is reserved and returns zeros on reads ZDATA[19:15] is used to select the pull-up on-die termination impedance ZDATA[14:10] is used to select the pull-down on-die termination impedance ZDATA[9:5] is used to select the pull-up output impedance ZDATA[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ1CR1
Address: Operational Base + offset (0x0194)
ZQ 1 Impedance Control Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x7b	ZPROG Impedance Divide Ratio: Selects the external resistor divide ratio to be used to set the output impedance and the on-die termination as follows: ZPROG[7:4] = On-die termination divide select ZPROG[3:0] = Output impedance divide select

DDR_PHYCTL_ZQ1SR0
Address: Operational Base + offset (0x0198)
ZQ 1 Impedance Status Register 0

Bit	Attr	Reset Value	Description
31	RO	0x0	ZDONE Impedance Calibration Done: Indicates that impedance calibration has completed.
30	RO	0x0	ZERR Impedance Calibration Error: If set, indicates that there was an error during impedance calibration.
29:28	RO	0x0	reserved
27:0	RO	0x0000000	ZCTRL Impedance Control: Current value of impedance control. ZCTRL field mapping for D3R I/Os is as follows: ZCTRL[27:20] is reserved and returns zeros on reads ZCTRL[19:15] is used to select the pull-up on-die termination impedance ZCTRL[14:10] is used to select the pull-down on-die termination impedance ZCTRL[9:5] is used to select the pull-up output impedance ZCTRL[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ1SR1

Address: Operational Base + offset (0x019c)
ZQ 1 Impedance Status Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	OPU On-die termination (ODT) pull-up calibration status. Similar status encodings as ZPD.
5:4	RO	0x0	OPD On-die termination (ODT) pull-down calibration status. Similar status encodings as ZPD.
3:2	RO	0x0	ZPU Output impedance pull-up calibration status. Similar status encodings as ZPD.
1:0	RO	0x0	ZPD Output impedance pull-down calibration status. Valid status encodings are: 00 = Completed with no errors 01 = Overflow error 10 = Underflow error 11 = Calibration in progress

DDR_PHYCTL_ZQ2CR0
Address: Operational Base + offset (0x01a0)
ZQ 2 Impedance Control Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	ZQPD ZQ Power Down: Powers down, if set, the PZQ cell.
30	RW	0x0	ZCAL Impedance Calibration Trigger: A write of '1' to this bit triggers impedance calibration to be performed by the impedance control logic. The impedance calibration trigger bit is self-clearing and returns back to '0' when the calibration is complete.
29	RW	0x0	ZCALBYP Impedance Calibration Bypass: Disables, if set, impedance calibration of this ZQ control block when impedance calibration is triggered globally using the ZCAL bit of PIR. Impedance calibration of this ZQ block may be triggered manually using ZCAL.

Bit	Attr	Reset Value	Description
28	RW	0x0	ZDEN Impedance Over-ride Enable: When this bit is set, it allows users to directly drive the impedance control using the data programmed in the ZQDATA field. Otherwise, the control is generated automatically by the impedance control logic
27:0	RW	0x000014a	ZDATA Impedance Over-Ride Data: Data used to directly drive the impedance control. ZDATA field mapping for D3R I/Os is as follows: ZDATA[27:20] is reserved and returns zeros on reads ZDATA[19:15] is used to select the pull-up on-die termination impedance ZDATA[14:10] is used to select the pull-down on-die termination impedance ZDATA[9:5] is used to select the pull-up output impedance ZDATA[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ2CR1
Address: Operational Base + offset (0x01a4)
ZQ 2 Impedance Control Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x7b	ZPROG Impedance Divide Ratio: Selects the external resistor divide ratio to be used to set the output impedance and the on-die termination as follows: ZPROG[7:4] = On-die termination divide select ZPROG[3:0] = Output impedance divide select

DDR_PHYCTL_ZQ2SR0
Address: Operational Base + offset (0x01a8)
ZQ 2 Impedance Status Register 0

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	RO	0x0	ZDONE Impedance Calibration Done: Indicates that impedance calibration has completed.
30	RO	0x0	ZERR Impedance Calibration Error: If set, indicates that there was an error during impedance calibration.
29:28	RO	0x0	reserved
27:0	RO	0x0000000	ZCTRL Impedance Control: Current value of impedance control. ZCTRL field mapping for D3R I/Os is as follows: ZCTRL[27:20] is reserved and returns zeros on reads ZCTRL[19:15] is used to select the pull-up on-die termination impedance ZCTRL[14:10] is used to select the pull-down on-die termination impedance ZCTRL[9:5] is used to select the pull-up output impedance ZCTRL[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ2SR1
Address: Operational Base + offset (0x01ac)
ZQ 2 Impedance Status Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	OPU On-die termination (ODT) pull-up calibration status. Similar status encodings as ZPD.
5:4	RO	0x0	OPD On-die termination (ODT) pull-down calibration status. Similar status encodings as ZPD.
3:2	RO	0x0	ZPU Output impedance pull-up calibration status. Similar status encodings as ZPD.

Bit	Attr	Reset Value	Description
1:0	RO	0x0	ZPD Output impedance pull-down calibration status. Valid status encodings are: 00 = Completed with no errors 01 = Overflow error 10 = Underflow error 11 = Calibration in progress

DDR_PHYCTL_ZQ3CR0
Address: Operational Base + offset (0x01b0)
ZQ 3 Impedance Control Register 0

Bit	Attr	Reset Value	Description
31	RW	0x0	ZQPD ZQ Power Down: Powers down, if set, the PZQ cell.
30	RW	0x0	ZCAL Impedance Calibration Trigger: A write of '1' to this bit triggers impedance calibration to be performed by the impedance control logic. The impedance calibration trigger bit is self-clearing and returns back to '0' when the calibration is complete.
29	RW	0x0	ZCALBYP Impedance Calibration Bypass: Disables, if set, impedance calibration of this ZQ control block when impedance calibration is triggered globally using the ZCAL bit of PIR. Impedance calibration of this ZQ block may be triggered manually using ZCAL.
28	RW	0x0	ZDEN Impedance Over-ride Enable: When this bit is set, it allows users to directly drive the impedance control using the data programmed in the ZQDATA field. Otherwise, the control is generated automatically by the impedance control logic

Bit	Attr	Reset Value	Description
27:0	RW	0x000014a	ZDATA Impedance Over-Ride Data: Data used to directly drive the impedance control. ZDATA field mapping for D3R I/Os is as follows: ZDATA[27:20] is reserved and returns zeros on reads ZDATA[19:15] is used to select the pull-up on-die termination impedance ZDATA[14:10] is used to select the pull-down on-die termination impedance ZDATA[9:5] is used to select the pull-up output impedance ZDATA[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ3CR1
Address: Operational Base + offset (0x01b4)
ZQ 3 Impedance Control Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x7b	ZPROG Impedance Divide Ratio: Selects the external resistor divide ratio to be used to set the output impedance and the on-die termination as follows: ZPROG[7:4] = On-die termination divide select ZPROG[3:0] = Output impedance divide select

DDR_PHYCTL_ZQ3SR0
Address: Operational Base + offset (0x01b8)
ZQ 3 Impedance Status Register 0

Bit	Attr	Reset Value	Description
31	RO	0x0	ZDONE Impedance Calibration Done: Indicates that impedance calibration has completed.
30	RO	0x0	ZERR Impedance Calibration Error: If set, indicates that there was an error during impedance calibration.

Bit	Attr	Reset Value	Description
29:28	RO	0x0	reserved
27:0	RO	0x0000000	ZCTRL Impedance Control: Current value of impedance control. ZCTRL field mapping for D3R I/Os is as follows: ZCTRL[27:20] is reserved and returns zeros on reads ZCTRL[19:15] is used to select the pull-up on-die termination impedance ZCTRL[14:10] is used to select the pull-down on-die termination impedance ZCTRL[9:5] is used to select the pull-up output impedance ZCTRL[4:0] is used to select the pull-down output impedance

DDR_PHYCTL_ZQ3SR1
Address: Operational Base + offset (0x01bc)
ZQ 3 Impedance Status Register 1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:6	RO	0x0	OPU On-die termination (ODT) pull-up calibration status. Similar status encodings as ZPD.
5:4	RO	0x0	OPD On-die termination (ODT) pull-down calibration status. Similar status encodings as ZPD.
3:2	RO	0x0	ZPU Output impedance pull-up calibration status. Similar status encodings as ZPD.
1:0	RO	0x0	ZPD Output impedance pull-down calibration status. Valid status encodings are: 00 = Completed with no errors 01 = Overflow error 10 = Underflow error 11 = Calibration in progress

DDR_PHYCTL_DX0GCR
Address: Operational Base + offset (0x01c0)
DATX8 0 General Configuration Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	RTTOAL RTT On Additive Latency: Indicates when the ODT control of DQ/DQS SSTL I/Os is set to the value in DQODT/DQSODT during read cycles. Valid values are: 0 = ODT control is set to DQSODT/DQODT almost two cycles before read data preamble 1 = ODT control is set to DQSODT/DQODT almost one cycle before read data preamble
12:11	RW	0x0	RTTOH RTT Output Hold: Indicates the number of clock cycles (from 0 to 3) after the read data postamble for which ODT control should remain set to DQSODT for DQS or DQODT for DQ/DM before disabling it (setting it to '0' when using dynamic ODT control. ODT is disabled almost RTTOH clock cycles after the read postamble
10	RW	0x1	DQRTT DQ Dynamic RTT Control: Indicates, if set, that the ODT control of DQ/DM SSTL I/Os be dynamically controlled by setting it to the value in DQODT during reads and disabling it (setting it to '0' during any other cycle. If this bit is not set, then the ODT control of DQ SSTL I/Os is always set to the value in DQODT.
9	RW	0x1	DQSRTT DQS Dynamic RTT Control: Indicates, if set, that the ODT control of DQS SSTL I/Os be dynamically controlled by setting it to the value in DQSODT during reads and disabling it (setting it to 0 during any other cycle. If this bit is not set, then the ODT control of DQS SSTL I/Os is always set to the value in DQSODT field.

Bit	Attr	Reset Value	Description
8:7	RW	0x1	DSEN Write DQS Enable: Controls whether the write DQS going to the SDRAM is enabled (toggling) or disabled (static value) and whether the DQS is inverted. DQS# is always the inversion of DQS. These values are valid only when DQS/DQS# output enable is on, otherwise the DQS/DQS# is tristated. Valid settings are: 00 = DQS disabled (Driven to constant 0) 01 = DQS toggling with inverted polarity 10 = DQS toggling with normal polarity (This should be the default setting) 11 = DQS disabled (Driven to constant 1)
6	RW	0x0	DQSRPD DQSR Power Down: Powers down, if set, the PDQSR cell. This bit is ORed with the common PDR configuration bit
5	RW	0x0	DXPDR Data Power Down Receiver: Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDR configuration bit
4	RW	0x0	DXPDD Data Power Down Driver: Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDD configuration bit
3	RW	0x0	DXIOM Data I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the IOM configuration bit of the individual DATX8
2	RW	0x0	DQODT Data On-Die Termination: Enables, when set, the on-die termination on the I/O for DQ and DM pins of the byte. This bit is ORed with the common DATX8 ODT configuration bit

Bit	Attr	Reset Value	Description
1	RW	0x0	DQSODT DQS On-Die Termination: Enables, when set, the on-die termination on the I/O for DQS/DQS# pin of the byte. This bit is ORed with the common DATX8 ODT configuration bit
0	RW	0x1	DXEN Data Byte Enable: Enables if set the DATX8 and SSTL I/Os used on the data byte. Setting this bit to '0' disables the byte, i.e. the byte SSTL I/Os are put in power-down mode and the DLL in the DATX8 is put in bypass mode.

DDR_PHYCTL_DX0GSR0
Address: Operational Base + offset (0x01c4)
DATX8 0 General Status Register 0

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:13	RO	0x000	DTPASS Data Training Pass Count: The number of passing configurations during DQS gate training. Bits [2:0] are for rank 0, bits [5:3] for rank 1, and so on.
12	RO	0x0	reserved
11:8	RO	0x0	DTIERR Data Training Intermittent Error: If set, indicates that there was an intermittent error during data training of the byte, such as a pass was followed by a fail then followed by another pass. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
7:4	RO	0x0	DTERR Data Training Error: If set, indicates that a valid DQS gating window could not be found during data training of the byte. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
3:0	RO	0x0	DTDONE Data Training Done: Indicates, if set, that the byte has finished doing data training. Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX0GSR1
Address: Operational Base + offset (0x01c8)

DATX8 0 General Status Register 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RO	0x00	DQSDFT DQS Drift: Used to report the drift on the read data strobe of the data byte. Valid settings are: 00 = No drift 01 = 90 deg drift 10 = 180 deg drift 11 = 270 deg drift or more Bits [1:0] are for rank 0, bits [3:2] for rank 1, and so on.
3:0	RO	0x0	DFTERR DQS Drift Error: If set, indicates that the byte read data strobe has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR). Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX0DLLCR

Address: Operational Base + offset (0x01cc)

DATX8 0 DLL Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DLLDIS DLL Disable: A disabled DLL is bypassed. Default ('0') is DLL enabled.
30	RW	0x1	DLLSRST DLL Soft Rest: Soft resets the byte DLL by driving the DLL soft reset pin.
29:20	RO	0x0	reserved
19	RW	0x0	SDLBMODE Slave DLL Loopback Mode: If this bit is set, the slave DLL is put in loopback mode in which there is no 90 degrees phase shift on read DQS/DQS#. This bit must be set when operating the byte PHYs in loopback mode such as during BIST loopback. Applicable only to PHYs that have this feature. Refer to PHY databook.

Bit	Attr	Reset Value	Description
18	RW	0x0	ATESTEN Analog Test Enable: Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.
17:14	RW	0x0	SDPHASE Slave DLL Phase Trim: Selects the phase difference between the input clock and the corresponding output clock of the slave DLL. Valid settings: 0000 = 90 0001 = 72 0010 = 54 0011 = 36 0100 = 108 0101 = 90 0110 = 72 0111 = 54 1000 = 126 1001 = 108 1010 = 90 1011 = 72 1100 = 144 1101 = 126 1110 = 108 1111 = 90
13:12	RW	0x0	SSTART Slave Auto Start-Up: Used to control how the slave DLL starts up relative to the master DLL locking: 0X = Slave DLL automatically starts up once the master DLL has achieved lock. 10 = The automatic startup of the slave DLL is disabled; the phase detector is disabled. 11 = The automatic startup of the slave DLL is disabled; the phase detector is enabled.
11:9	RW	0x0	MFWDLY Master Feed-Forward Delay Trim: Used to trim the delay in the master DLL feed-forward path: 000 = minimum delay 111 = maximum delay

Bit	Attr	Reset Value	Description
8:6	RW	0x0	MFBDLY Master Feed-Back Delay Trim: Used to trim the delay in the master DLL feedback path: 000 = minimum delay 111 = maximum delay
5:3	RW	0x0	SFWDLY Slave Feed-Forward Delay Trim: Used to trim the delay in the slave DLL feed-forward path: 000 = minimum delay 111 = maximum delay
2:0	RW	0x0	SFBDLY Slave Feed-Back Delay Trim: Used to trim the delay in the slave DLL feedback path: 000 = minimum delay 111 = maximum delay

DDR_PHYCTL_DX0DQTR
Address: Operational Base + offset (0x01d0)
DATX8 0 DQ Timing Register

Bit	Attr	Reset Value	Description
31:28	RW	0xf	DQDLY7 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
27:24	RW	0xf	DQDLY6 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
23:20	RW	0xf	DQDLY5 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
19:16	RW	0xf	DQDLY4 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
15:12	RW	0xf	DQDLY3 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
11:8	RW	0xf	DQDLY2 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree.Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
7:4	RW	0xf	DQDLY1 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
3:0	RW	0xf	DQDLY0 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

DDR_PHYCTL_DX0DQSTR

Address: Operational Base + offset (0x01d4)

DATX8 0 DQS Timing Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:26	RW	0xf	DMDLY DM Delay: Used to adjust the delay of the data mask relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. The lower two bits of the DQMDLY controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
25:23	RW	0x3	DQSNDLY DQS# Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps
22:20	RW	0x3	DQSDLY DQS Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps

Bit	Attr	Reset Value	Description
19:18	RW	0x1	R3DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
17:16	RW	0x1	R2DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
15:14	RW	0x1	R1DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
13:12	RW	0x1	R0DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
11:9	RW	0x0	R3DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x0	R2DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
5:3	RW	0x0	R1DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	R0DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

DDR_PHYCTL_DX1GCR
Address: Operational Base + offset (0x0200)
DATX8 1 General Configuration Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	RTTOAL RTT On Additive Latency: Indicates when the ODT control of DQ/DQS SSTL I/Os is set to the value in DQODT/DQSODT during read cycles. Valid values are: 0 = ODT control is set to DQSODT/DQODT almost two cycles before read data preamble 1 = ODT control is set to DQSODT/DQODT almost one cycle before read data preamble

Bit	Attr	Reset Value	Description
12:11	RW	0x0	RTTOH RTT Output Hold: Indicates the number of clock cycles (from 0 to 3) after the read data postamble for which ODT control should remain set to DQSODT for DQS or DQODT for DQ/DM before disabling it (setting it to '0' when using dynamic ODT control. ODT is disabled almost RTTOH clock cycles after the read postamble
10	RW	0x1	DQRTT DQ Dynamic RTT Control: Indicates, if set, that the ODT control of DQ/DM SSTL I/Os be dynamically controlled by setting it to the value in DQODT during reads and disabling it (setting it to '0' during any other cycle. If this bit is not set, then the ODT control of DQ SSTL I/Os is always set to the value in DQODT.
9	RW	0x1	DQSRTT DQS Dynamic RTT Control: Indicates, if set, that the ODT control of DQS SSTL I/Os be dynamically controlled by setting it to the value in DQSODT during reads and disabling it (setting it to 0 during any other cycle. If this bit is not set, then the ODT control of DQS SSTL I/Os is always set to the value in DQSODT field.
8:7	RW	0x1	DSEN Write DQS Enable: Controls whether the write DQS going to the SDRAM is enabled (toggling) or disabled (static value) and whether the DQS is inverted. DQS# is always the inversion of DQS. These values are valid only when DQS/DQS# output enable is on, otherwise the DQS/DQS# is tristated. Valid settings are: 00 = DQS disabled (Driven to constant 0) 01 = DQS toggling with inverted polarity 10 = DQS toggling with normal polarity (This should be the default setting) 11 = DQS disabled (Driven to constant 1)
6	RW	0x0	DQSRPD DQSR Power Down: Powers down, if set, the PDQSR cell. This bit is ORed with the common PDR configuration bit

Bit	Attr	Reset Value	Description
5	RW	0x0	DXPDR Data Power Down Receiver: Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDR configuration bit
4	RW	0x0	DXPDD Data Power Down Driver: Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDD configuration bit
3	RW	0x0	DXIOM Data I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the IOM configuration bit of the individual DATX8
2	RW	0x0	DQODT Data On-Die Termination: Enables, when set, the on-die termination on the I/O for DQ and DM pins of the byte. This bit is ORed with the common DATX8 ODT configuration bit
1	RW	0x0	DQSODT DQS On-Die Termination: Enables, when set, the on-die termination on the I/O for DQS/DQS# pin of the byte. This bit is ORed with the common DATX8 ODT configuration bit
0	RW	0x1	DXEN Data Byte Enable: Enables if set the DATX8 and SSTL I/Os used on the data byte. Setting this bit to '0' disables the byte, i.e. the byte SSTL I/Os are put in power-down mode and the DLL in the DATX8 is put in bypass mode.

DDR_PHYCTL_DX1GSR0

Address: Operational Base + offset (0x0204)

DATX8 1 General Status Register 0

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:13	RO	0x000	DTPASS Data Training Pass Count: The number of passing configurations during DQS gate training. Bits [2:0] are for rank 0, bits [5:3] for rank 1, and so on.
12	RO	0x0	reserved
11:8	RO	0x0	DTIERR Data Training Intermittent Error: If set, indicates that there was an intermittent error during data training of the byte, such as a pass was followed by a fail then followed by another pass. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
7:4	RO	0x0	DTERR Data Training Error: If set, indicates that a valid DQS gating window could not be found during data training of the byte. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
3:0	RO	0x0	DTDONE Data Training Done: Indicates, if set, that the byte has finished doing data training. Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX1GSR1
Address: Operational Base + offset (0x0208)
DATX8 1 General Status Register 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RO	0x00	DQSDFT DQS Drift: Used to report the drift on the read data strobe of the data byte. Valid settings are: 00 = No drift 01 = 90 deg drift 10 = 180 deg drift 11 = 270 deg drift or more Bits [1:0] are for rank 0, bits [3:2] for rank 1, and so on.

Bit	Attr	Reset Value	Description
3:0	RO	0x0	DFTERR DQS Drift Error: If set, indicates that the byte read data strobe has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR). Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX1DLLCR

Address: Operational Base + offset (0x020c)

DATX8 1 DLL Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DLLDIS DLL Disable: A disabled DLL is bypassed. Default ('0') is DLL enabled.
30	RW	0x1	DLLSRST DLL Soft Rest: Soft resets the byte DLL by driving the DLL soft reset pin.
29:20	RO	0x0	reserved
19	RW	0x0	SDLBMODE Slave DLL Loopback Mode: If this bit is set, the slave DLL is put in loopback mode in which there is no 90 degrees phase shift on read DQS/DQS#. This bit must be set when operating the byte PHYs in loopback mode such as during BIST loopback. Applicable only to PHYs that have this feature. Refer to PHY databook.
18	RW	0x0	ATESTEN Analog Test Enable: Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.

Bit	Attr	Reset Value	Description
17:14	RW	0x0	SDPHASE Slave DLL Phase Trim: Selects the phase difference between the input clock and the corresponding output clock of the slave DLL. Valid settings: 0000 = 90 0001 = 72 0010 = 54 0011 = 36 0100 = 108 0101 = 90 0110 = 72 0111 = 54 1000 = 126 1001 = 108 1010 = 90 1011 = 72 1100 = 144 1101 = 126 1110 = 108 1111 = 90
13:12	RW	0x0	SSTART Slave Auto Start-Up: Used to control how the slave DLL starts up relative to the master DLL locking: 0X = Slave DLL automatically starts up once the master DLL has achieved lock. 10 = The automatic startup of the slave DLL is disabled; the phase detector is disabled. 11 = The automatic startup of the slave DLL is disabled; the phase detector is enabled.
11:9	RW	0x0	MFWDLY Master Feed-Forward Delay Trim: Used to trim the delay in the master DLL feed-forward path: 000 = minimum delay 111 = maximum delay
8:6	RW	0x0	MFBDLY Master Feed-Back Delay Trim: Used to trim the delay in the master DLL feedback path: 000 = minimum delay 111 = maximum delay

Bit	Attr	Reset Value	Description
5:3	RW	0x0	SFWDLY Slave Feed-Forward Delay Trim: Used to trim the delay in the slave DLL feed-forward path: 000 = minimum delay 111 = maximum delay
2:0	RW	0x0	SFBDLY Slave Feed-Back Delay Trim: Used to trim the delay in the slave DLL feedback path: 000 = minimum delay 111 = maximum delay

DDR_PHYCTL_DX1DQTR
Address: Operational Base + offset (0x0210)
DATX8 1 DQ Timing Register

Bit	Attr	Reset Value	Description
31:28	RW	0xf	DQDLY7 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
27:24	RW	0xf	DQDLY6 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
23:20	RW	0xf	DQDLY5 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
19:16	RW	0xf	DQDLY4 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
15:12	RW	0xf	DQDLY3 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
11:8	RW	0xf	DQDLY2 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
7:4	RW	0xf	DQDLY1 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
3:0	RW	0xf	DQDLY0 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

DDR_PHYCTL_DX1DQSTR

Address: Operational Base + offset (0x0214)

DATX8 1 DQS Timing Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:26	RW	0xf	DMDLY DM Delay: Used to adjust the delay of the data mask relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. The lower two bits of the DQMDLY controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
25:23	RW	0x3	DQSNDLY DQS# Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps
22:20	RW	0x3	DQSDLY DQS Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps

Bit	Attr	Reset Value	Description
19:18	RW	0x1	R3DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
17:16	RW	0x1	R2DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
15:14	RW	0x1	R1DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
13:12	RW	0x1	R0DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
11:9	RW	0x0	R3DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x0	R2DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
5:3	RW	0x0	R1DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	R0DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

DDR_PHYCTL_DX2GCR
Address: Operational Base + offset (0x0240)
DATX8 2 General Configuration Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	RTTOAL RTT On Additive Latency: Indicates when the ODT control of DQ/DQS SSTL I/Os is set to the value in DQODT/DQSODT during read cycles. Valid values are: 0 = ODT control is set to DQSODT/DQODT almost two cycles before read data preamble 1 = ODT control is set to DQSODT/DQODT almost one cycle before read data preamble

Bit	Attr	Reset Value	Description
12:11	RW	0x0	RTTOH RTT Output Hold: Indicates the number of clock cycles (from 0 to 3) after the read data postamble for which ODT control should remain set to DQSODT for DQS or DQODT for DQ/DM before disabling it (setting it to '0' when using dynamic ODT control. ODT is disabled almost RTTOH clock cycles after the read postamble
10	RW	0x1	DQRTT DQ Dynamic RTT Control: Indicates, if set, that the ODT control of DQ/DM SSTL I/Os be dynamically controlled by setting it to the value in DQODT during reads and disabling it (setting it to '0' during any other cycle. If this bit is not set, then the ODT control of DQ SSTL I/Os is always set to the value in DQODT.
9	RW	0x1	DQSRTT DQS Dynamic RTT Control: Indicates, if set, that the ODT control of DQS SSTL I/Os be dynamically controlled by setting it to the value in DQSODT during reads and disabling it (setting it to 0 during any other cycle. If this bit is not set, then the ODT control of DQS SSTL I/Os is always set to the value in DQSODT field.
8:7	RW	0x1	DSEN Write DQS Enable: Controls whether the write DQS going to the SDRAM is enabled (toggling) or disabled (static value) and whether the DQS is inverted. DQS# is always the inversion of DQS. These values are valid only when DQS/DQS# output enable is on, otherwise the DQS/DQS# is tristated. Valid settings are: 00 = DQS disabled (Driven to constant 0) 01 = DQS toggling with inverted polarity 10 = DQS toggling with normal polarity (This should be the default setting) 11 = DQS disabled (Driven to constant 1)
6	RW	0x0	DQSRPD DQSR Power Down: Powers down, if set, the PDQSR cell. This bit is ORed with the common PDR configuration bit

Bit	Attr	Reset Value	Description
5	RW	0x0	DXPDR Data Power Down Receiver: Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDR configuration bit
4	RW	0x0	DXPDD Data Power Down Driver: Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDD configuration bit
3	RW	0x0	DXIOM Data I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the IOM configuration bit of the individual DATX8
2	RW	0x0	DQODT Data On-Die Termination: Enables, when set, the on-die termination on the I/O for DQ and DM pins of the byte. This bit is ORed with the common DATX8 ODT configuration bit
1	RW	0x0	DQSODT DQS On-Die Termination: Enables, when set, the on-die termination on the I/O for DQS/DQS# pin of the byte. This bit is ORed with the common DATX8 ODT configuration bit
0	RW	0x1	DXEN Data Byte Enable: Enables if set the DATX8 and SSTL I/Os used on the data byte. Setting this bit to '0' disables the byte, i.e. the byte SSTL I/Os are put in power-down mode and the DLL in the DATX8 is put in bypass mode.

DDR_PHYCTL_DX2GSR0

Address: Operational Base + offset (0x0244)

DATX8 2 General Status Register 0

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:13	RO	0x000	DTPASS Data Training Pass Count: The number of passing configurations during DQS gate training. Bits [2:0] are for rank 0, bits [5:3] for rank 1, and so on.
12	RO	0x0	reserved
11:8	RO	0x0	DTIERR Data Training Intermittent Error: If set, indicates that there was an intermittent error during data training of the byte, such as a pass was followed by a fail then followed by another pass. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
7:4	RO	0x0	DTERR Data Training Error: If set, indicates that a valid DQS gating window could not be found during data training of the byte. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
3:0	RO	0x0	DTDONE Data Training Done: Indicates, if set, that the byte has finished doing data training. Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX2GSR1
Address: Operational Base + offset (0x0248)
DATX8 2 General Status Register 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RO	0x00	DQSDFT DQS Drift: Used to report the drift on the read data strobe of the data byte. Valid settings are: 00 = No drift 01 = 90 deg drift 10 = 180 deg drift 11 = 270 deg drift or more Bits [1:0] are for rank 0, bits [3:2] for rank 1, and so on.

Bit	Attr	Reset Value	Description
3:0	RO	0x0	DFTERR DQS Drift Error: If set, indicates that the byte read data strobe has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR). Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX2DLLCR

Address: Operational Base + offset (0x024c)

DATX8 2 DLL Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DLLDIS DLL Disable: A disabled DLL is bypassed. Default ('0') is DLL enabled.
30	RW	0x1	DLLSRST DLL Soft Rest: Soft resets the byte DLL by driving the DLL soft reset pin.
29:20	RO	0x0	reserved
19	RW	0x0	SDLBMODE Slave DLL Loopback Mode: If this bit is set, the slave DLL is put in loopback mode in which there is no 90 degrees phase shift on read DQS/DQS#. This bit must be set when operating the byte PHYs in loopback mode such as during BIST loopback. Applicable only to PHYs that have this feature. Refer to PHY databook.
18	RW	0x0	ATESTEN Analog Test Enable: Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.

Bit	Attr	Reset Value	Description
17:14	RW	0x0	SDPHASE Slave DLL Phase Trim: Selects the phase difference between the input clock and the corresponding output clock of the slave DLL. Valid settings: 0000 = 90 0001 = 72 0010 = 54 0011 = 36 0100 = 108 0101 = 90 0110 = 72 0111 = 54 1000 = 126 1001 = 108 1010 = 90 1011 = 72 1100 = 144 1101 = 126 1110 = 108 1111 = 90
13:12	RW	0x0	SSTART Slave Auto Start-Up: Used to control how the slave DLL starts up relative to the master DLL locking: 0X = Slave DLL automatically starts up once the master DLL has achieved lock. 10 = The automatic startup of the slave DLL is disabled; the phase detector is disabled. 11 = The automatic startup of the slave DLL is disabled; the phase detector is enabled.
11:9	RW	0x0	MFWDLY Master Feed-Forward Delay Trim: Used to trim the delay in the master DLL feed-forward path: 000 = minimum delay 111 = maximum delay
8:6	RW	0x0	MFBDLY Master Feed-Back Delay Trim: Used to trim the delay in the master DLL feedbackpath: 000 = minimum delay 111 = maximum delay

Bit	Attr	Reset Value	Description
5:3	RW	0x0	SFWDLY Slave Feed-Forward Delay Trim: Used to trim the delay in the slave DLL feed-forward path: 000 = minimum delay 111 = maximum delay
2:0	RW	0x0	SFBDLY Slave Feed-Back Delay Trim: Used to trim the delay in the slave DLL feedback path: 000 = minimum delay 111 = maximum delay

DDR_PHYCTL_DX2DQTR
Address: Operational Base + offset (0x0250)
DATX8 2 DQ Timing Register

Bit	Attr	Reset Value	Description
31:28	RW	0xf	DQDLY7 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
27:24	RW	0xf	DQDLY6 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
23:20	RW	0xf	DQDLY5 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
19:16	RW	0xf	DQDLY4 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
15:12	RW	0xf	DQDLY3 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
11:8	RW	0xf	DQDLY2 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
7:4	RW	0xf	DQDLY1 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
3:0	RW	0xf	DQDLY0 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

DDR_PHYCTL_DX2DQSTR

Address: Operational Base + offset (0x0254)

DATX8 2 DQS Timing Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:26	RW	0xf	DMDLY DM Delay: Used to adjust the delay of the data mask relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. The lower two bits of the DQMDLY controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
25:23	RW	0x3	DQSNDLY DQS# Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps
22:20	RW	0x3	DQSDLY DQS Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps

Bit	Attr	Reset Value	Description
19:18	RW	0x1	R3DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
17:16	RW	0x1	R2DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
15:14	RW	0x1	R1DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
13:12	RW	0x1	R0DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
11:9	RW	0x0	R3DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x0	R2DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
5:3	RW	0x0	R1DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	R0DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

DDR_PHYCTL_DX3GCR
Address: Operational Base + offset (0x0280)
DATX8 3 General Configuration Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	RTTOAL RTT On Additive Latency: Indicates when the ODT control of DQ/DQS SSTL I/Os is set to the value in DQODT/DQSODT during read cycles. Valid values are: 0 = ODT control is set to DQSODT/DQODT almost two cycles before read data preamble 1 = ODT control is set to DQSODT/DQODT almost one cycle before read data preamble

Bit	Attr	Reset Value	Description
12:11	RW	0x0	RTTOH RTT Output Hold: Indicates the number of clock cycles (from 0 to 3) after the read data postamble for which ODT control should remain set to DQSODT for DQS or DQODT for DQ/DM before disabling it (setting it to '0' when using dynamic ODT control. ODT is disabled almost RTTOH clock cycles after the read postamble
10	RW	0x1	DQRTT DQ Dynamic RTT Control: Indicates, if set, that the ODT control of DQ/DM SSTL I/Os be dynamically controlled by setting it to the value in DQODT during reads and disabling it (setting it to '0' during any other cycle. If this bit is not set, then the ODT control of DQ SSTL I/Os is always set to the value in DQODT.
9	RW	0x1	DQSRTT DQS Dynamic RTT Control: Indicates, if set, that the ODT control of DQS SSTL I/Os be dynamically controlled by setting it to the value in DQSODT during reads and disabling it (setting it to 0 during any other cycle. If this bit is not set, then the ODT control of DQS SSTL I/Os is always set to the value in DQSODT field.
8:7	RW	0x1	DSEN Write DQS Enable: Controls whether the write DQS going to the SDRAM is enabled (toggling) or disabled (static value) and whether the DQS is inverted. DQS# is always the inversion of DQS. These values are valid only when DQS/DQS# output enable is on, otherwise the DQS/DQS# is tristated. Valid settings are: 00 = DQS disabled (Driven to constant 0) 01 = DQS toggling with inverted polarity 10 = DQS toggling with normal polarity (This should be the default setting) 11 = DQS disabled (Driven to constant 1)
6	RW	0x0	DQSRPD DQSR Power Down: Powers down, if set, the PDQSR cell. This bit is ORed with the common PDR configuration bit

Bit	Attr	Reset Value	Description
5	RW	0x0	DXPDR Data Power Down Receiver: Powers down, when set, the input receiver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDR configuration bit
4	RW	0x0	DXPDD Data Power Down Driver: Powers down, when set, the output driver on I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the common PDD configuration bit
3	RW	0x0	DXIOM Data I/O Mode: Selects SSTL mode (when set to 0) or CMOS mode (when set to 1) of the I/O for DQ, DM, and DQS/DQS# pins of the byte. This bit is ORed with the IOM configuration bit of the individual DATX8
2	RW	0x0	DQODT Data On-Die Termination: Enables, when set, the on-die termination on the I/O for DQ and DM pins of the byte. This bit is ORed with the common DATX8 ODT configuration bit
1	RW	0x0	DQSODT DQS On-Die Termination: Enables, when set, the on-die termination on the I/O for DQS/DQS# pin of the byte. This bit is ORed with the common DATX8 ODT configuration bit
0	RW	0x1	DXEN Data Byte Enable: Enables if set the DATX8 and SSTL I/Os used on the data byte. Setting this bit to '0' disables the byte, i.e. the byte SSTL I/Os are put in power-down mode and the DLL in the DATX8 is put in bypass mode.

DDR_PHYCTL_DX3GSR0
Address: Operational Base + offset (0x0284)
DATX8 3 General Status Register 0

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:13	RO	0x000	DTPASS Data Training Pass Count: The number of passing configurations during DQS gate training. Bits [2:0] are for rank 0, bits [5:3] for rank 1, and so on.
12	RO	0x0	reserved
11:8	RO	0x0	DTIERR Data Training Intermittent Error: If set, indicates that there was an intermittent error during data training of the byte, such as a pass was followed by a fail then followed by another pass. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
7:4	RO	0x0	DTERR Data Training Error: If set, indicates that a valid DQS gating window could not be found during data training of the byte. Bit [0] is for rank 0, bit 1 for rank 1, and so on.
3:0	RO	0x0	DTDONE Data Training Done: Indicates, if set, that the byte has finished doing data training. Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX3GSR1
Address: Operational Base + offset (0x0288)
DATX8 3 General Status Register 1

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:4	RO	0x00	DQSDFT DQS Drift: Used to report the drift on the read data strobe of the data byte. Valid settings are: 00 = No drift 01 = 90 deg drift 10 = 180 deg drift 11 = 270 deg drift or more Bits [1:0] are for rank 0, bits [3:2] for rank 1, and so on.

Bit	Attr	Reset Value	Description
3:0	RO	0x0	DFTERR DQS Drift Error: If set, indicates that the byte read data strobe has drifted by more than or equal to the drift limit set in the PHY General Configuration Register (PGCR). Bit [0] is for rank 0, bit 1 for rank 1, and so on.

DDR_PHYCTL_DX3DLLCR

Address: Operational Base + offset (0x028c)

DATX8 3 DLL Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	DLLDIS DLL Disable: A disabled DLL is bypassed. Default ('0') is DLL enabled.
30	RW	0x1	DLLSRST DLL Soft Rest: Soft resets the byte DLL by driving the DLL soft reset pin.
29:20	RO	0x0	reserved
19	RW	0x0	SDLBMODE Slave DLL Loopback Mode: If this bit is set, the slave DLL is put in loopback mode in which there is no 90 degrees phase shift on read DQS/DQS#. This bit must be set when operating the byte PHYs in loopback mode such as during BIST loopback. Applicable only to PHYs that have this feature. Refer to PHY databook.
18	RW	0x0	ATESTEN Analog Test Enable: Enables the analog test signal to be output on the DLL analog test output (test_out_a). The DLL analog test output is tri-stated when this bit is '0'.

Bit	Attr	Reset Value	Description
17:14	RW	0x0	SDPHASE Slave DLL Phase Trim: Selects the phase difference between the input clock and the corresponding output clock of the slave DLL. Valid settings: 0000 = 90 0001 = 72 0010 = 54 0011 = 36 0100 = 108 0101 = 90 0110 = 72 0111 = 54 1000 = 126 1001 = 108 1010 = 90 1011 = 72 1100 = 144 1101 = 126 1110 = 108 1111 = 90
13:12	RW	0x0	SSTART Slave Auto Start-Up: Used to control how the slave DLL starts up relative to the master DLL locking: 0X = Slave DLL automatically starts up once the master DLL has achieved lock. 10 = The automatic startup of the slave DLL is disabled; the phase detector isdisabled. 11 = The automatic startup of the slave DLL is disabled; the phase detector is enabled.
11:9	RW	0x0	MFWDLY Master Feed-Forward Delay Trim: Used to trim the delay in the master DLL feed-forward path: 000 = minimum delay 111 = maximum delay
8:6	RW	0x0	MFBDLY Master Feed-Back Delay Trim: Used to trim the delay in the master DLL feedback path: 000 = minimum delay 111 = maximum delay

Bit	Attr	Reset Value	Description
5:3	RW	0x0	SFWDLY Slave Feed-Forward Delay Trim: Used to trim the delay in the slave DLL feed-forward path: 000 = minimum delay 111 = maximum delay
2:0	RW	0x0	SFBDLY Slave Feed-Back Delay Trim: Used to trim the delay in the slave DLL feedback path: 000 = minimum delay 111 = maximum delay

DDR_PHYCTL_DX3DQTR
Address: Operational Base + offset (0x0290)
DATX8 3 DQ Timing Register

Bit	Attr	Reset Value	Description
31:28	RW	0xf	DQDLY7 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
27:24	RW	0xf	DQDLY6 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
23:20	RW	0xf	DQDLY5 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
19:16	RW	0xf	DQDLY4 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
15:12	RW	0xf	DQDLY3 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
11:8	RW	0xf	DQDLY2 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps
7:4	RW	0xf	DQDLY1 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
3:0	RW	0xf	DQDLY0 DQ Delay: Used to adjust the delay of the data relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. Every four bits of this register control the delay of a different data bit in the byte. DQDLY0 controls the delay of data bit [0], DQDLY1 controls data bit [1], and so on. The lower two bits of the DQDLY for each DQ bit controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

DDR_PHYCTL_DX3DQSTR

Address: Operational Base + offset (0x0294)

DATX8 3 DQS Timing Register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:26	RW	0xf	DMDLY DM Delay: Used to adjust the delay of the data mask relative to the nominal delay that is matched to the delay of the data strobes through the slave DLL and clock tree. The lower two bits of the DQMDLY controls the delay for the data clocked by DQS, while the higher two bits control the delay for the data clocked by DQS_b. Valid settings for each 2-bit control field are: 00 = nominal delay 01 = nominal delay + 1 step 10 = nominal delay + 2 steps 11 = nominal delay + 3 steps

Bit	Attr	Reset Value	Description
25:23	RW	0x3	DQSNDLY DQS# Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps
22:20	RW	0x3	DQSDLY DQS Delay: Used to adjust the delay of the data strobes relative to the nominal delay that is matched to the delay of the data bit through the slave DLL and clock tree. DQSDLY control the delay on DQS strobe and DQSNDLY control the delay on DQS#. Valid values are: 000 = nominal delay - 3 steps 001 = nominal delay - 2 steps 010 = nominal delay - 1 step 011 = nominal delay 100 = nominal delay + 1 step 101 = nominal delay + 2 steps 110 = nominal delay + 3 steps 111 = nominal delay + 4 steps

Bit	Attr	Reset Value	Description
19:18	RW	0x1	R3DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
17:16	RW	0x1	R2DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
15:14	RW	0x1	R1DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
13:12	RW	0x1	R0DGPS Rank n DQS Gating Phase Select: Selects the clock used to enable the data strobes during read so that the value of the data strobes before and after the preamble/postamble are filtered out. The RnDGPS fields are initially set by the PHYCTL during automatic DQS data training and subsequently updated during data strobe drift compensation. However, these values can be overwritten by a direct write to this register, and the automatic update during DQS drift compensation can be disabled using the PHY General Configuration Register (PGCR). Every two bits of this register control the DQS gating for each of the (up to) four ranks. R0DGPS controls the DQS gating for rank 0, R1DGPS controls rank 1, and so on. Valid values for each 2-bit RnDGPS field are: 00 = 90 deg clock (clk90) 01 = 180 deg clock (clk180) 10 = 270 deg clock (clk270) 11 = 360 deg clock (clk0)

Bit	Attr	Reset Value	Description
11:9	RW	0x0	R3DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
8:6	RW	0x0	R2DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
5:3	RW	0x0	R1DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

Bit	Attr	Reset Value	Description
2:0	RW	0x0	R0DGSL Rank n DQS Gating System Latency: Used to increase the number of clock cycles needed to expect valid DDR read data by up to five extra clock cycles. This is used to compensate for board delays and other system delays. Power-up default is 000 (i.e. no extra clock cycles required). The SL fields are initially set by the PHYCTL during automatic DQS data training but these values can be overwritten by a direct write to this register. Every three bits of this register control the latency of each of the (up to) four ranks. R0DGSL controls the latency of rank 0, R1DGSL controls rank 1, and so on. Valid values are: 000 = No extra clock cycles 001 = 1 extra clock cycle 010 = 2 extra clock cycles 011 = 3 extra clock cycles 100 = 4 extra clock cycles 101 = 5 extra clock cycles 110 = Reserved 111 = Reserved

13.6 Timing Diagram

13.6.1 DDR3 Read/Write Access Timing

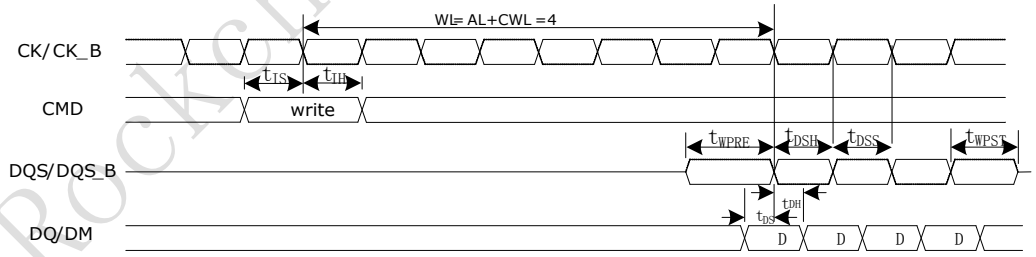


Fig. 13-10DDR3 burst write operation: AL=0,CWL=4, BC4

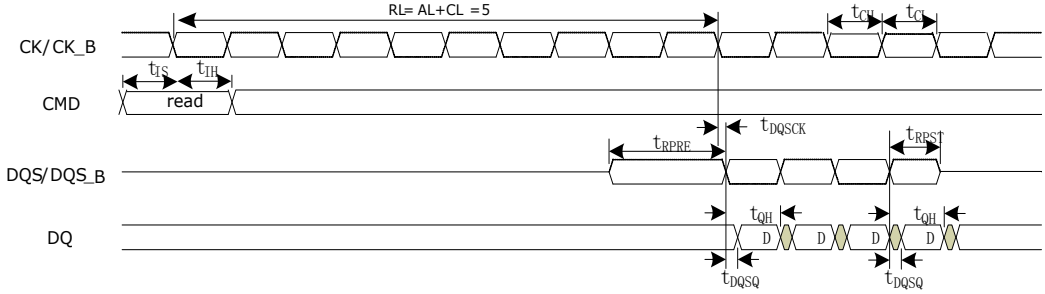


Fig. 13-11DDR3 burst read operation: AL=0,CL=5, BC4

Table 13-17 meaning of the parameter in Fig.13-10 and Fig.13-11

Parameter	Description	DDR3-800		unit
		min	max	
t_{CH}	CK HIGH pulse width	0.43	-	tCK
t_{CL}	CK LOW pulse width	0.43	-	tCK
t_{DS}	DQ and DM input setup time (differential strobe)	75	-	ps
t_{DH}	DQ and DM input hold time (differential strobe)	150	-	ps
t_{DSS}	DQS falling edge to CK setup time	0.2	-	tCK
t_{DSH}	DQS falling edge hold time from CK	0.2	-	tCK
t_{IS}	Address and control input setup time	200	-	ps
t_{IH}	Address and control input hold time	275	-	ps
t_{WPRE}	Write preamble	0.9	-	tCK
t_{WPST}	Write postamble	0.3	-	tCK
t_{RPRE}	Read preamble	0.9	1.1	tCK
t_{RPST}	Read postamble	0.3	0.5	tCK
t_{DQSCK}	DQS output access time from CK/CK_n	-400	+400	ps
t_{DQSQ}	DQS-DQ skew for DQS and associated DQ signals	-	200	ps
t_{QH}	DQ/DQS output hold time from DQS	0.38	-	tCK

13.6.2 LPDDR2 Read/Write Access Timing

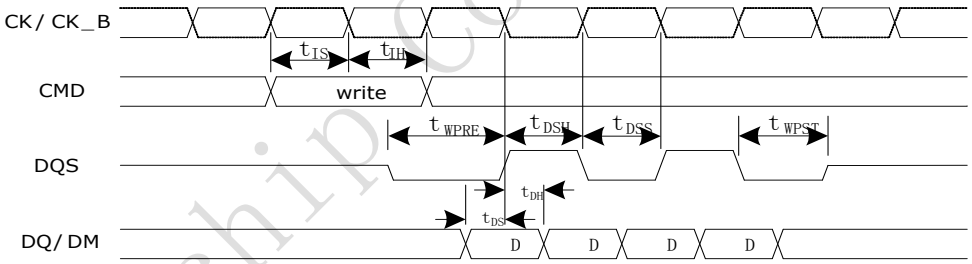


Fig. 13-12LPDDR2 burst write operation: WL=1,BL=4

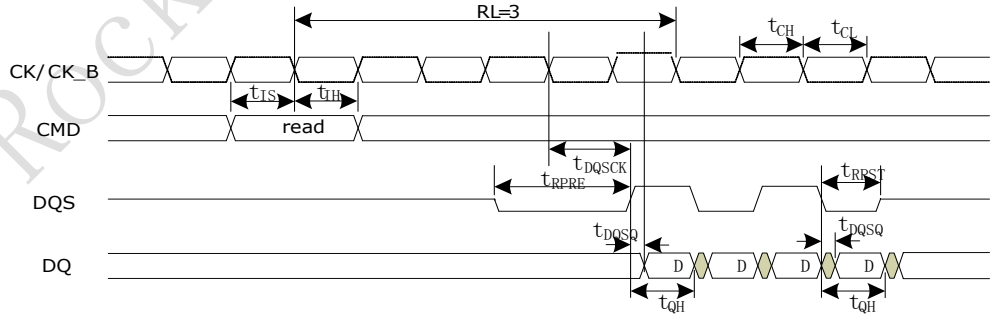


Fig. 13-13LPDDR2 burst read operation: RL=3, BL=4

Table 13-18 meaning of the parameter in Fig.13-12 and Fig.13-13

Parameter	Description	LPDDR2 S4-800		unit
		min	max	
t_{CH}	CK HIGH pulse width	0.43	0.57	tCK
t_{CL}	CK LOW pulse width	0.43	0.57	tCK

t_{DS}	DQ and DM inputsetup time	0.27	-	ns
t_{DH}	DQ and DM input hold time	0.27	-	ns
t_{DSS}	DQS falling edge to CK setup time	0.2	-	tCK
t_{DSH}	DQS falling edge hold time from CK	0.2	-	tCK
t_{IS}	Address and control input setup time	0.29	-	ns
t_{IH}	Address and control input hold time	0.29	-	ns
t_{WPRE}	Write preamble	0.35	-	tCK
t_{WPST}	Write postamble	0.4	-	tCK
t_{RPRE}	Read preamble	0.9	-	tCK
t_{RPST}	Read postamble	t_{CL} -0.05	t_{CL} -0.05	tCK
t_{DQSCK}	DQS output access time from CK/CK_n	2.5	5.5	ns
t_{DQSQ}	DQS-DQ skew for DQS and associated DQ signals	0.24	0.28	ns
t_{QH}	DQ/DQS output hold time from DQS	t_{QHP}- t_{QHS}	-	ns
RL	Read Latency	6		
WL	Write Latency	3		

13.7 Interface description

DDR IOs are listed as following Table.

Pin Name	Description
CK	Active-high clock signal to the memory device.
CK_B	Active-low clock signal to the memory device.
CKEi (i=0,1)	Active-high clock enable signal to the memory device for two chip select.
CS_Bi (i=0,1)	Active-low chip select signal to the memory device. AThere are two chip select.
RAS_B	Active-low row address strobe to the memory device.
CAS_B	Active-low column address strobe to the memory device.
WE_B	Active-low write enable strobe to the memory device.
BA[2:0]	Bank address signal to the memory device.
A[15:0]	Address signal to the memory device.
DQ[31:0]	Bidirectional data line to the memory device.
DQS[3:0]	Active-high bidirectional data strobes to the memory device.
DQS_B[3:0]	Active-low bidirectional data strobes to the memory device.
DM[3:0]	Active-low data mask signal to the memory device.
ODTi (i=0,1)	On-Die Termination output signal for two chip select.
RET_EN	Active-low retention latch enable input.
VREFi (i=0,1,2)	Reference Voltage input for three regions of DDR IO.
ZQ_PIN	ZQ calibration pad which connects 240ohm±1% resistor.
RESET	DDR3 reset signal.

13.8 Application Notes

13.8.1 State transition of PCTL

To operate PCTL, the programmer must be familiar with the available operational states and how to transition to each state from the current state.

Every software programmable register is accessible only during certain operational states. For information about what registers are accessible in each state, refer to "Software Registers,"which provides this information in each register description. The general rule is that the PCTL must be in the Init_mem or Config states to successfully write most of the registers.

The following tables provide the programming sequences for moving to the various states of the state machine.

Moving to the Init_mem State

Step	Application	PCTL
1	Read STAT register	ReturnsthecurrentPCTLstate.
2	If STAT.ctl_stat = Init_mem, go toEND.	
3	If STAT.ctl_stat =Config, go toStep9.	
4	If STAT.ctl_stat =Access, go toStep8.	
5	If STAT.ctl_stat = Low_power, go toStep7.	
6	Goto Step1.	PCTLis ina Transitionalstate and not inany ofthe previous operationalstates.
7	WriteWAKEUPtoSCTL.state_cmd andpoll STAT.ctl_stat= Access.	IssuesSRX,movestotheAccessstate,updates STAT.ctl_stat =Accesswhen complete.
8	WriteCFG to SCTL.state_cmd andpoll STAT.ctl_stat= Config.	PCTLstalls the NIF;completesany pending transaction; issuesPREAifrequired; movesintothe Config state; updates STAT.ctl_stat =Config whencomplete.
9	WriteINITto SCTL.state_cmd andpoll STAT.ctl_stat=Init_mem	Moves intotheInit_mem stateandupdatesSTAT.ctl_stat =Init_mem.
END		PCTLis inInit_memstate.

Moving to Config State

Step	Application	PCTL
1	Read STAT register.	Returns thecurrentPCTLstate.
2	If STAT.ctl_stat= Config, goto END.	
3	If STAT.ctl_stat= Low_power,gotoStep6.	
4	If STAT.ctl_stat= Init_mem or Access,gotoStep7.	
5	GotoStep1.	PCTLisinatransitionalstateandisnotinanyof the previous operational states.

6	WriteWAKEUPtoSCTL.state_cmdand poll STAT.ctl_stat= Access.	Issues SRX,movesto theAccess state,and updates STAT.ctl_stat= Accesswhen complete.
7	WriteCFGtoSCTL.state_cmdand poll STAT.ctl_stat= Config.	PCTLstalls the NIF; completesanypending transaction; issuesPREAif required;moves into the Configstate; and updates STAT.ctl_stat = Config whencomplete.
END		PCTLis in Configstate.

Moving to Access State

Step	Application	PCTL
1	Read STATregister	Returns the current PCTLstate.
2	IfSTAT.ctl_stat= Access,goto END.	
3	IfSTAT.ctl_stat= Config, gotoStep9	
4	IfSTAT.ctl_stat= Init_mem, gotoStep8	
5	IfSTAT.ctl_stat= Low_power,go to Step7.	
6	GotoStep1.	PCTLisinatransitionalstateandisnot inanyofthe previous operationalstates.
7	WriteWAKEUPtoSCTL.state_cmdandpoll STAT.ctl_stat= Access.GotoEND	Issues SRX,movestotheAccess state,updates STAT.ctl_stat= Accesswhen complete.
8	WriteCFGtoSCTL.state_cmdandpoll STAT.ctl_stat= Config.	MovesintotheConfigstate,updates STAT.ctl_stat= Config when complete.
9	WriteGOtoSCTL.state_cmd andpoll STAT.ctl_stat= Access.	MovesintotheAccessstate, updates STAT.ctl_stat= Accesswhen complete.
END		PCTLis inAccess state.

Moving to Low Power State

Step	Application	PCTL
1	Read STAT register.	Returnscurrent PCTL state.
2	If STAT.ctl_stat =Low_power, goto END.	
3	If STAT.ctl_stat = Access,gotoStep9	
4	If STAT.ctl_stat = Config,gotoStep8	
5	If STAT.ctl_stat = Init_mem,go toStep7.	
6	Goto Step1.	PCTL is intransitionalstateandis notin anyofthe previous operationalstates.
7	WriteCFG to SCTL.state_cmdand poll STAT.ctl_stat= Config.	Movesintothe Configstate, updates STAT.ctl_stat = Config when complete.
8	WriteGOtoSCTL.state_cmdandpoll STAT.ctl_stat= Access.	MovesintotheAccessstate,updates STAT.ctl_stat =Access whencomplete.

9	WriteSLEEPtoSCTL.state_cmdandpoll STAT.ctl_stat =Low_power.	IssuesPDXif necessary;completes anypending transactions;issuesPREAcommand; finally, issuesSRE andupdates STAT.ctl_stat = Low_power.
END		PCTLis inLowPowerstate

13.8.2 Initialization

Figure 13-14 shows a high-level illustration of the initialization sequence of the PHY. A detailed sequence description and timing diagrams are described in the following. This section assumes a generic configuration port and therefore cfg_clk and cfg_rst_n are shown as the configuration clock and reset, respectively. These signals must be replaced by pclk and presetn if the design is compiled to use the APB configuration port.

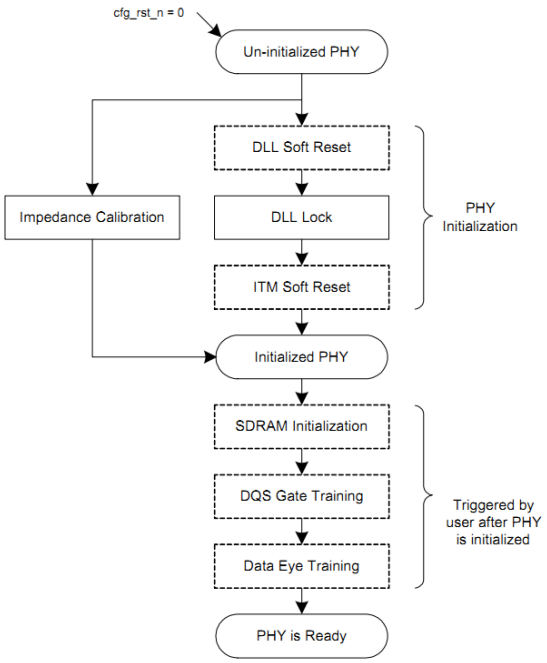


Fig. 13-14Protocol controller architecture

PHY Initialization

The initialization sequence has two phases. The first phase happens automatically at reset and is as follows:

1. Before and during configuration reset (i.e. if cfg_rst_n is asserted), the PHY is un-initialized and remains in this state until the reset is de-asserted.
2. At reset de-assertion, the PHY moves into the DLL initialization (lock) phase. This phase may be bypassed at any time by writing a '1' to the DLL initialization bypass register bit (PIR[LOCKBYP]).
3. In parallel to DLL initialization, the impedance calibration phase also starts at reset de-assertion.

This phase can also be bypassed by writing a '1' to the impedance calibration bypass register bit-PIR[ZCALBYP].

4. If the PHY initialization sequence was triggered by the user, a soft reset may optionally be selected to be issued to the ITMs. Initialization that is automatically triggered on reset does not issue a soft reset to the ITMs because the components will already have been reset by the main reset.

5. Once the DLL initialization and impedance calibration phases are done and after the ITMs are reset, the PHY is initialized. Note that if these phases were bypassed, it is up to the user to perform them in software or trigger them at a later time before the PHY can be used.

SDRAM Initialization

The second phase of initialization starts after the PHY is initialized. Each step of this phase is triggered by the user or memory controller and is as follows:

1. Prior to normal operation, DDR SDRAMs must be initialized. The PHYCTL has a built-in SDRAM initialization routine that may be triggered by software or memory controller by writing to the PHY Initialization Register (PIR). The initialization routine built into the PHYCTL is generic and does not require any knowledge of the type or configuration of external SDRAMs to be properly executed. The routine is designed with the relevant JEDEC specifications for the fastest & slowest SDRAMs supported by the PHYCTL to result in a universal initialization sequence. This generic sequence is applicable to DDR3, DDR2, LPDDR2, LPDDR, and DDR SDRAMs.

It is recommended to use the built-in PHYCTL routine to initialize the SDRAM. However, there may be cases such as during system debug when the built-in PHYCTL DRAM initialization is not triggered and DRAM initialization is performed by software or the controller. In these cases the system must first wait for the PHY to initialize, i.e. DLL locked and impedance calibration done, then it must write a '1' to PIR[INIT] bit with PIR[CTLDINT] set to '1' (for controller initialization) or '0' (for software or PHYCTL initialization) to inform the PHYCTL that DRAM initialization will be done later, by software, the controller or by re-triggering on the PHYCTL. The software or controller then executes the initialization sequence by sending relevant commands to the DRAM, respecting the various timing requirements of the initialization sequence.

2. After the SDRAM is initialized, the user or memory controller performs, or triggers the PHYCTL to perform DQS gate training (“Built-in DQS Gate Training” on page 114). The SDRAM must be initialized before triggering DQS gate training.

3. The user or memory controller performs, or triggers the PHYCTL to

perform read data eye training. Note that the current version of the PHYCTL does not have the read eye training designed in.

4. The PHY is now ready for SDRAM read/write accesses.

DDR3 Initialization Sequence

The initialization steps for DDR3 SDRAMs are as follows:

1. Optionally maintain RESET# low for a minimum of either 200 us (power-up initialization) or 100ns (power-on initialization). The PHYCTL drives RESET# low from the beginning of reset assertion and therefore this step may be skipped when DRAM initialization is triggered if enough time may already have expired to satisfy the RESET# low time.
2. After RESET# is de-asserted, wait a minimum of 500 us with CKE low.
3. Apply NOP and drive CKE high.
4. Wait a minimum of tXPR.
5. Issue a load Mode Register 2 (MR2) command.
6. Issue a load Mode Register 3 (MR3) command.
7. Issue a load Mode Register (MR1) command (to set parameters and enable DLL).
8. Issue a load Mode Register (MR0) command to set parameters and reset DLL.
9. Issue ZQ calibration command.

10. Wait 512 SDRAM clock cycles for the DLL to lock (tDLLK) and ZQ calibration (tZQinit) to finish. This wait time is relative to Step 8, i.e. relative to when the DLL reset command was issued onto the SDRAM command bus.

LPDDR2 Initialization Sequence

The initialization steps for LPDDR2 SDRAMs are as follows:

1. Wait a minimum of 100 ns (tINIT1) with CKE driven low.
2. Apply NOP and set CKE high.
3. Wait a minimum of 200 us (tINIT3).
4. Issue a RESET command.
5. Wait a minimum of 1 us + 10 us (tINIT4 + tINIT5).
6. Issue a ZQ calibration command.
7. Wait a minimum of 1 us (tZQINIT).
8. Issue a Write Mode Register to MR1.
9. Issue a Write Mode Register to MR2
10. Issue a Write Mode Register to MR3

Initialization Triggerred and bypass

All initialization steps shown in Figure 3-1 on page 34 can be triggered using the PHY Initialization Register (PIR) as described in "PHY Initialization Register (PIR)" on page 47. Writing a '1' to PIR[INIT] register bit will start initialization, with the routines to be run being selected by the corresponding PIR register bits. If multiple routines are selected, they are run in the order shown in Figure 3-1 on page 34. This is also the order of the select bits in PIR register. The completion of the routines is indicated in the PHY General Status Register (PGSR) with the corresponding done status bits (see "PHY General Status Register (PGSR)" on page 52). The PGSR[IDONE] bit indicates the overall completion of the initialization sequence. An initialization done status register bit is cleared (reset to '0') when the corresponding routine is re-triggered.

The de-assertion of reset will automatically trigger the PHYCTL to perform DLL initialization (locking) and impedance calibration. Once the DLL has locked and impedance calibration has completed, the SDRAM initialization and DQS gating may be triggered or performed by software or memory controller.

Since the PHYCTL allows the selection of individual routines to be run when initialization is triggered using PIR register, only those routines that automatically trigger on reset de-assertion have individual bypass capability. This means that DLL locking and/or impedance calibration may be bypassed any time by writing a '1' to the corresponding bypass register bit in the PIR register. Once a routine is bypassed, it is internally registered as completed and the corresponding done status register bit is set in the PGSR register.

It is up to the user to re-trigger or perform the bypassed routine at a later time before the PHY can be used. The PIR[INITBYP] register bit provides the option to bypass the whole initialization sequence.

13.8.3 MDLL and MSDLL Reset Requirements

Reset issued to the MDLL and MSDLL must always meet the following requirements:

1. Reset must always be asserted for a minimum of 50ns to ensure proper reset of the DLL.
2. On power-up, reset must be held for a minimum of 50ns after MVDD has been raised to its full value.
3. After reset has been asserted and then de-asserted, a number of clock cycles must pass for the DLL to achieve lock.
4. The input clock to the DLL must be stable for a minimum of 50ns before DLL

reset is de-asserted.

The following additional requirements apply when transitioning to/from bypass mode:

1. There must be at least 50ns between reset de-assertion and DLL bypass mode entry.
2. The DLL bypass pin must be asserted for at least 1000ns.
3. Reset must always be issued after the DLL mode has changed from bypass to normal mode.
4. A minimum of 100ns is required between bypass de-assertion and reset assertion.
5. Reset must be issued whenever DLL control/trim/option input bits are modified, with the exception of:
   a. Analog/digital test controls
   b. Slave DLL phase trim (if applicable).

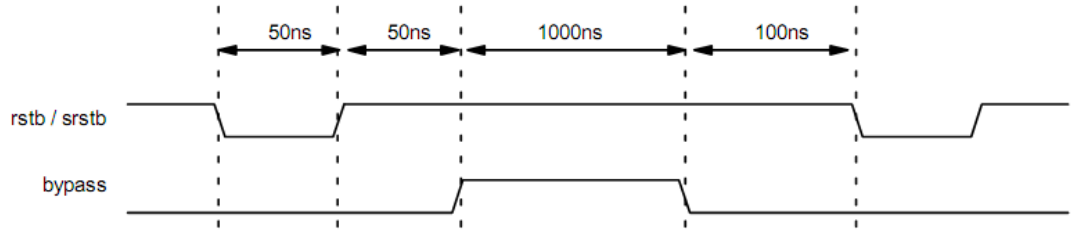


Fig. 13-15DLL reset requirements

13.8.4 Data Training

Built-in DQS Gate Training

The PHYCTL has a built-in DQS gate training routine that may be triggered by software or memory controller using the PIR register.

DQS gate training returns a number of status, including the done and error status. There are two types of errors. The first type is when no valid window was found for the byte. This is indicated by DTERR register bit in DXnGSR and PGSR registers. This is usually an indication of bad configuration. The second type is when some passing configurations were found but these were interspersed by failures. This is not expected in a working system. A typical window is signified by consecutive passes followed by consecutive failures, e.g. FPPPPPF and not FPPFPPF. This type of error is called an intermittent error and is indicated by the DTIERR register bit in DXnGSR and PGSR registers. Provided for debug purpose is the status of how many passing configurations were found for each byte on each rank. This is indicated by DTPASS field in the DXnGSR register.

Software DQS Gate Training

DQS gate training may also be executed in software using the controller and/or the PUB DCU. Figure 13-16 shows the DQS gate training software algorithm. This is followed by a description of the main phases of the training.

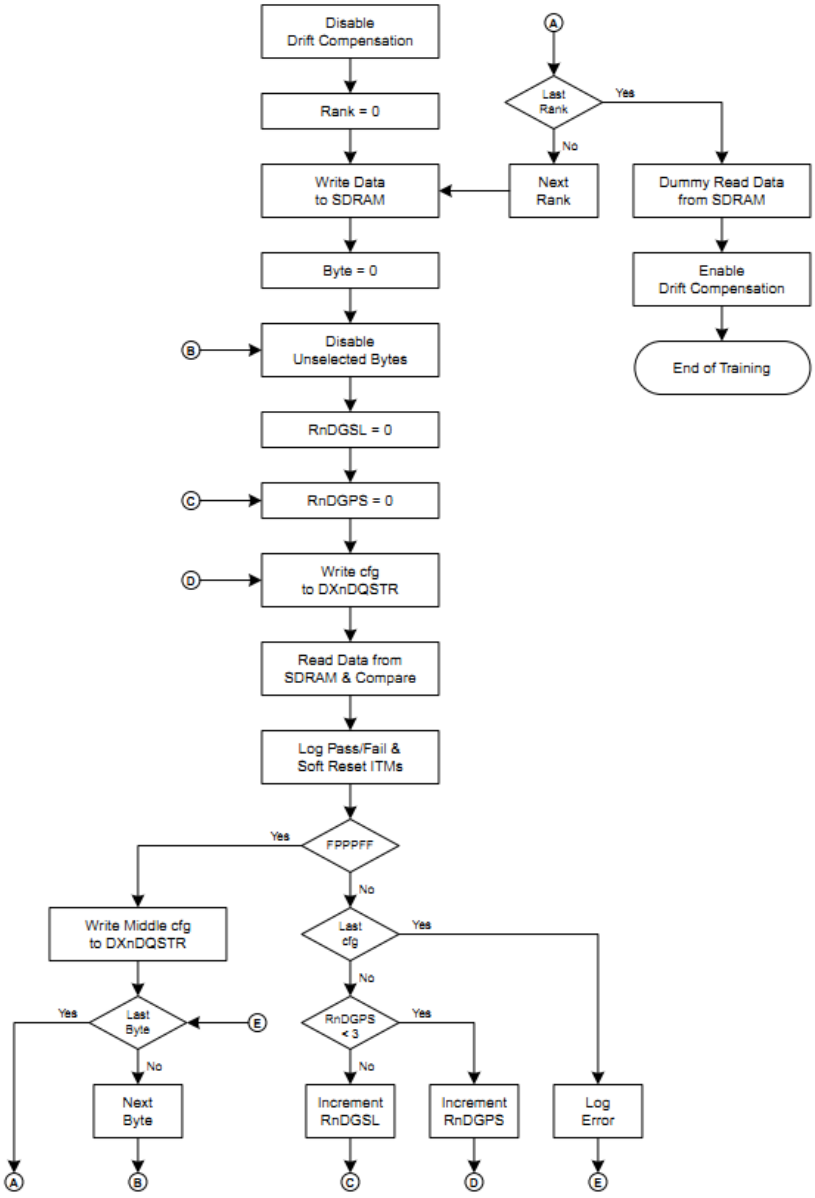


Fig. 13-16DLL reset requirements

The software DQS gate training phases are as follows:
1. Disable drift compensation by writing '0' to PGCR.DFTCMP register.
2. Start with rank 0, i.e. rank 0 is selected for training.
3. Execute a minimum of two writes to the SDRAM. Any type of data and any SDRAM address can be used for DQS gate training. It is however not recommended to use data that is all zeroes since this may mask read data comparison. The data mask must be set to 0 to enable writing of all bytes. The number of writes must be chosen such that it results in a minimum of eight data beats at the SDRAM. This means at least two write commands when using SDRAM burst length of 4.
4. Start with byte 0 (i.e. byte 0 is selected for training).
5. Disable all the other bytes except the byte that has been selected for training. Bytes are enabled/disabled by writing 1/0 to DXnDGCR.DXEN.
6. Start with the selected rank byte DQS gating system latency (DXnDQSTR.RnDGSL) of 0.
7. Start with the selected rank byte DQS gating phase select

(DXnDQSTR.RnDGPS) of 0.
8. Write the selected DQS gating configurations (RnDGSL and RnDGPS) to DXnDQSTR register of the selected byte, making sure the fields for the unselected ranks remain unchanged.
9. Execute reads from the SDRAM locations previously written. The number of reads must be equal to the number of writes used in Step 3. Compare the read data with the expected (written) data and log the pass/fail status as a sequence or history of flags for each trained RnDGSL/RnDGPS configuration (e.g. FFPPPPFF). A fail is either when there is a data miscompare or when fewer data than expected is returned. Note that a controller that is designed to always wait for the correct number of read data may need a time-out in case the trained configuration results in fewer data than expected. This is not as issue when using the PUB DCU because it does not wait for the expected number of reads; rather the read count status will indicate if fewer reads were returned.
10. Once the read data has been compared and the pass/fail status logged, issue an ITM soft reset to clear the status of the read data logic in the PHY. This is important because the ITM read data FIFO pointers may be in the wrong state at the end of training an RnDGSL/RnDGPS configuration that resulted in wrong DQS gating window.
11. If two consecutive fails and some passes exist, then this is the end of the training for this rank byte. In this case, do the following:

- Select the middle of the passes and write the values to the corresponding fields of DXnDQSTR register, making sure the fields for the unselected ranks remain unchanged
- If this is not the last byte, then select the next byte and go to Step 5
- If this is the last byte but not the last rank, then select the next rank and go to Step 3
- If this is the last byte and the last rank, then go to Step 12 to do final clean-up before the end of the DQS gate training.
- If the condition of two consecutive fails and some passes does not exist, then this signals that more RnDGSL/RnDGPS configurations need to be trained for this rank byte. If this is the case, do the following:
  ■if RnDGPS is less than 3, then increment RnDGPS and go to Step 8
  ■if RnDGPS is equal to 3 but RnDGSL is less than 7, then increment RnDGSL and go to Step 7
  ■if RnDGPS is equal to 3 and RnDGSL is equal to 7, then log an error because this is a signal that something in the system is very wrong such that no passing configuration is possible for this rank byte. With such an error condition, you can either terminate the whole training to investigate the system or you can go to train the next byte.

12. Once the training of all ranks and all bytes is finished, issue one or more dummy reads to the same SDRAM locations. This will flush out the DQS drift compensation logic in the PHY and therefore avoid reporting any false drift events caused by previous DQS gating settings.
13. Once the dummy reads have completed, re-enable drift compensation by writing 1 to PGCR.DFTCMP register. This is the end of DQS gate training. Regular memory operations can now commence.

13.8.5 Impedance Calibration

The impedance calibration circuit, which controls the impedance values for ODT and driver output impedance, consists of the following components:

- ✧ ZQ calibration cell - PZQ
- ✧ External RZQ precision resistor
- ✧ Impedance control logic - zctrl

- ✧ VREF cell (for code encoding and level shifting)
- ✧ Functional I/O cells

The connectivity of these components is shown as follow figure:

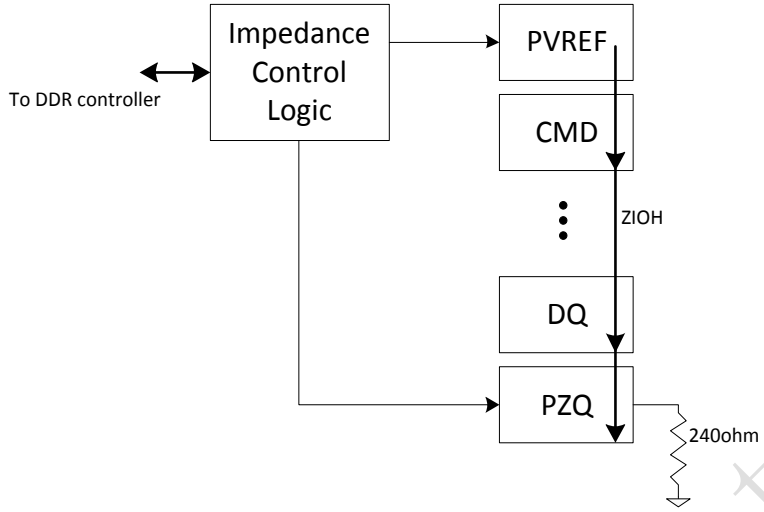


Fig. 13-17Impedance Calibration Circuit

A single calibration cell (PZQ) is used for the interface. The user connects the PZQ pin through an external 240ohm±1% resistor to ground. One or multiple VREF cells exist in the interface, depending on the total data width of the interface. The ZCTRL bus from the impedance control logic is connected to all VREF cells in the interface. It is not permitted to have a VREF cell in the interface that is not connected to the impedance control logic.

The impedance control logic sends an impedance code through the ZCTRL bus to the VREF cells. The VREF cells encodes this data, level shifts it to the VDDQ power domain, and sends it to both the functional I/O cells and the PZQ cell through the ZIOH bus embedded within the SSTL cells. The PZQ cell also receives the desired divide ratios from the Memory Controller or the user logic. The PZQ cell compares the impedance control code received from the PVREF cell with the external resistor, taking into account the selected divide ratio. The PZQ cell then sends ZCOMP back to the impedance control logic to relay information about impedance matching. The impedance control logic then sends a new impedance code to the PVREF cells. This results in a closed-loop system.

The four impedance elements are calibrated sequentially:

- ✧ Pull-up termination impedance
- ✧ Pull-down termination impedance
- ✧ Pull-up output impedance
- ✧ Pull-down output impedance

The ZPROG bus is used to signal which element is being calibrated. The state machine is implemented on the Impedance Controller RTL block.

The impedance control logic connects to the Memory Controller or customer logic to allow full controllability and observability of the loop operation.

The impedance control loop operates with a low bandwidth as compared to the memory system, thus the impedance control logic contains a clock divider to permit operation at a reduced clock frequency.

There are three basic modes of operation:

- ✧ Direct Calibration -uses ZPROG settings.
- ✧ Override Setting - uses ctrl_ovrd_data settings.
- ✧ Custom Calibration - extends calibration beyond the values available on ZPROG

Direct Calibration

In this mode, the user is setting independently the value for ODT (ZPROG[7:4]) and Output Impedance (ZPROG[3:0]) and runs the calibration sequence:

1. Output impedance pulldown
2. Output impedance pull-up
3. On-Die termination (ODT) pull-down
4. ODT pull-up

Override Setting

In this mode, the user is not using the calibration loop, and instead directly controls the impedance control using zctrl_ovrd_data[19:0] bus, which is parsed in four nibbles that independently control driver pull-down/up and ODT pull-down/up impedance in 31 steps.

For example, assuming one step is associated to current I and the calibration voltage is VREF, the programmed impedance for index N is:

ZPROG = K * VREF/(N * I)

K is correction factor, which is approximately equal to 1.Based on the formula, it can be concluded that if index N is increased, then the impedance is decreased.

Custom Calibration

This mode is a two-step procedure combining the previous two modes.

1. The user provides a Direct Calibration using a convenient value and records the Impedance control results from status register.
2. The user applies the correction factor that provides the custom impedance.

The following example assumes that it is required to program Driver Output Impedance to 18 ohms.

1. The user performs a Direct Calibration for driver Zo=36 ohms. For example, assume the result shows that Driver pull-up index is 12, and Driver pull-down index is 13.
2. Calculate and apply the Override Data for 18 ohm impedance adjustment as follows:

(<cal_value>/<req_value>) * <cal_index>
Driver pull-down (36/18) * 13 = 26
Driver pull-up (36/18) * 12 = 24

13.8.6 Retention Functional

The purpose of the retention function is to retain a known state on the signals to the SDRAMs while the system is placed in a low power mode, specifically when the core VDD supply is powered down. The general concept is that an external input signal (RET_EN) is driven low to put the SSTL I/O cells into retention mode shortly before the core VDD supply is powered down. The user must set the SSTL I/O outputs in the state required during power down before asserting RET_EN. This ensures that the output state of all SSTL I/Os are held static in the desired state while core VDD is power down. After core VDD is restored, the user must re-initialize the core logic to a known state before de-asserting the RET_EN signal.

Following figure provides the I/O cell arrangement with retention.

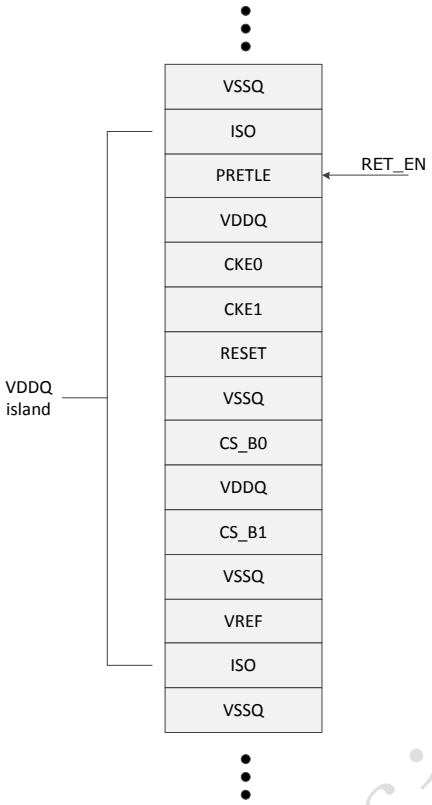


Fig. 13-18I/O cell arrangement with retention

IOs between two ISO is a VDDQ island, they will maintain power on when other IOs are powered down by RET_EN active.

Following figure provides a sequence of events to enter and exit retention.

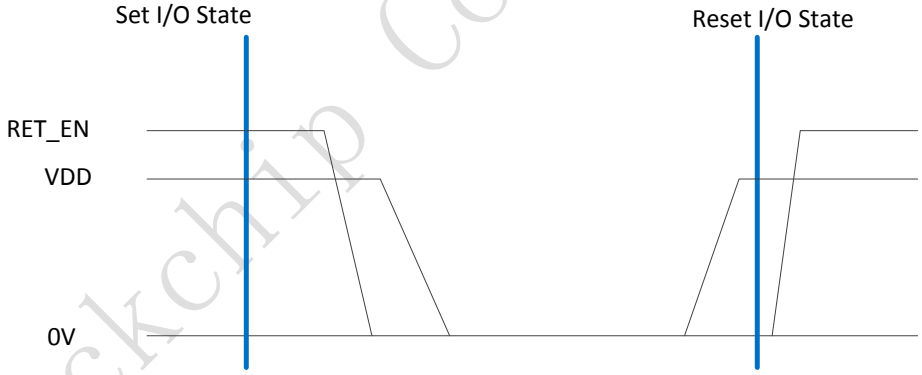


Fig. 13-19Sequence of Events to Enter and Exit Retention

CKE Retention Mode

An alternative CKE retention mode is supported. This scheme works by placing the SDRAMs into self-refresh mode and then driving the CKE signal low. Core VDD and VDDQ can then both be powered down except for a small VDDQ island supplying the CKE output cell. Two of the special 5um spacer cells ISO are used to break the VDDQ rail in order to create a separate CKE VDDQ island, which is kept powered while core VDD and the main VDDQ are powered down.

The sequence of events is as follows:

1. Enter self-refresh mode using the Self-Refresh Command
2. Set CKE low
3. Stop CK/CKB
4. Assert RET_EN (low)
5. Power-Off
6. Power-On

7. After reset is released, execute initialization
8. De-assert RET_EN (high)
9. Start CK/CKB
10. Set CKE high
11. Exit self-refresh mode

13.8.7 Low Power Operation

Low_power state can be entered/exited via following ways:

- Software control of PCTL State machine (highest priority)
- Hardware Low Power Interface (middle priority)
- Auto Self Refresh feature (lowest priority)

Note the priority of requests from Access to Low_power is highlighted above. The STAT.lp_trig register field reports which of the 3 requests caused the entry to Low_power state.

Software control of PCTL State

The application can request via software to enter the memories into Self Refresh state by issuing the SLEEP command by programming SCTL.PCTL responds to the software request by moving into the Low_power operational state and issuing the SRE command to the memories. Note that the Low_power state can only be reached from the Access state.

In a similar fashion, the application requests to exit the memories from Self Refresh by issuing a WAKEUP command by programming SCTL.. PCTL responds to the WAKEUP command issuing SRX and restoring normal NIF address channel operation.

Hardware Low Power Interface

The hardware low power interface can also be used to enter/exit Self Refresh. The functionality is enabled by setting SCFG.hw_low_power_en=1. Once that bit is set, the input c_sysreq has the ability to trigger entry into the

Low Power configuration state just like the software methodology (SCTL.state_cmd=SLEEP). A hardware Low Power entry trigger will be ignored/denied if the input c_active_in=1 or n_avalid=1. It may be accepted if c_active_in=0 and n_avalid=0, depending on the current state of the PCTL. When SCFG.hw_low_power_en=1, the outputs c_sysack and c_active provide feedback as required by the AXI low power interface specification (this interface's operation is defined by the AXI specification). c_sysack acknowledges the request to go into the Low_power state, and c_active indicates when the PCTL is actually in the Low_power state.

The c_active output could also be used by an external Low Power controller to decide when to request a transistion to low power. When MCFG1.hw_idle > 0, c_active = 1'b0 indicates that the NIF has been idle for at least MCFG1.hw_idle * 32 * n_clk cycles while in the Access state.

When in low power the c_active output can used by an external Low Power controller to trigger a low power exit. c_active will be driven high when either c_active_in or n_avalid are high. The path from c_active_in and n_valid to c_active is asynhronous so even if the clocks have been removed c_active will assert. The Low Power controller should re-enable the clocks when c_active is driven high while in the Low_power state.

Auto Clock Stop/Power Down/Self Refresh

The Clock Stop and/or Power Down and/or Self Refresh sequence is automatically started by PCTL when the NIF address channel is idle for a number

of cycles, depending on the programmed value in MCFG.mddr_lpddr2_clkstop_idle and MCFG.pd_idle and MCFG1.sr_idle. Following table outlines the effect of these settings in conjunction with NIF being idle.

mddr_lpddr2_ clkstop_idle	pd_idle	sr_idle	Memory modes	Memory Type
0	0	0	none	All
>0	0	0	Clock Stop	mDDR/LPDDR2 only
0	>0	0	Power Down	All
>0	>0	0	Clock Stop -> Power Down[1]	mDDR/LPDDR2 only
0	0	>0	Self Refresh	All
>0	0	>0	Clock Stop -> Self Refresh[2]	mDDR/LPDDR2 only
0	>0	>0	Power Down -> Self Refresh[3]	All
>0	>0	>0	Clock Stop -> Power Down -> Self Refresh[4]	mDDR/LPDDR2 only

Note:

1. Clock Stop is entered if NIF is idle for mddr_lpddr2_clkstop_idle. Following on from that, if NIF continues to be idle fora further pd_idle cycles, Clock Stop is exited and Power Down is entered.

*2. Clock Stop is entered if NIF is idle for mddr_lpddr2_clkstop_idle. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Clock Stop is exited and Self Refresh is entered.*

*3. Power Down is entered if NIF is idle for pd_idle. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self Refresh is entered.*

*4. Clock Stop is entered if NIF is idle for mddr_lpddr2_clkstop_idle. Following on from that, if NIF continues to be idle for a further pd_idle cycles, Clock Stop is exited and Power Down is entered. Following on from that, if NIF continues to be idle for a further sr_idle*32 cycles, Power Down is exited and Self Refresh is entered.*

Removing PCTL's n_clk

In LPDDR2 and DDR3, the relationship between SRE/SRX and stopping/starting the memory clock (CK)are formalized and are accounted for automatically by PCTL. With LPDDR2 and DDR3, CK should onlybe stopped after PCTL has reached the Low_power state. The current operational state can be verified byreading STAT.ctl_stat. The CK must be started and stable before the Software or Hardware Low Power Interface attempts to take the memory out of Self Refresh.

PCTL's n_clk can be safely removed when PCTL is in Low Power state. The sequences outlined in Table P2-17 or Table P2-18 should be followed for safe operation:

Step	Application	PCTL
1	Write SLEEP to SCTL.state_cmd and poll STAT.ctl_stat = LOW_POWER.	Tells PCTL to move memories into Self Refresh and waits until this completes.
2	Write TREFI=0. Also, write DFITCRLUPDI=0 and DFIREFMSKI=0, if they are not already 0.	Stops any MC-driven DFI updates occurring internally with PCTL
3	Wait a minimum interval which is equivalent to the PCTL's Refresh Interval (previous value of TREFI*TOGCNT100N*internal timers clock period;	Ensures any already scheduled PHY/PVT updates have completed successfully.
4	Stop toggling n_clk to PCTL.	n_clk logic inside PCTL is stopped.
end		

Step	Application	PCTL
1	Drive c_active_in low	Confirms that system external to PCTL can accept a Low- power request
2	Drive c_sysreq low	System Low-power request
3	Wait for PCTL to drive c_sysack low	PCTL Low-power request acknowledgement
4	Check value of c_active when Step 3 occurs. - if c_active=1, request denied. Cannot remove n_clk. Go to END. - if c_active=0, request accepted.	PCTL low-power request status response
5	Stop toggling n_clk to PCTL	n_clk logic inside PCTL is stopped
end		

Deep Power-Down

Compared with DDR2/DDR3, mDDR and LPDDR2 has an additional low power mode (Deep Power Down). :

❖ Software-driven Deep Power Down Entry – on reception of DPDE from the application, PCTL drives CKE low for TDPD.t_dpd. After TDPD, MCMD.start_cmd will be cleared to 1'b0. The following are recommended values for TDPD:

- ✦ mDDR: TDPD=0
- ✦ LPDDR2: dependent on if the system wants to immediately power off the PCTL after Deep Power down is entered::
  - ✧ If PCTL not Powered off: TDPD=500μs
  - ✧ Else if PCTL is Poweredoff: TDPD=0 - up to higher level system to meet tDPD requirement.

❖ To Exit Deep Power Mode, full initialization of the memories must be performed.

13.8.8 PHY Power Down

The PHYCTL includes several registers for putting certain components of the PHY in power down mode. The PHTCTL also supports DFI-initiated power-down of its components using the DFI low-power protocol.

Several components of the PHY can be powered down using PHYCTL registers. There are separate power-down register bits for the address/command lane and for each byte lane. Also there are separate controls for powering down the I/Os versus powering down the DLL. Following table describes the registers that are used to power down various components of the PHY.

Register Name	Bit Field	Description
PIR	DLLBYP	Bypasses, and hence disables or powers down all PHY DLLs.
ACDLLCR	DLLDIS	Disables (powers down) the address/command lane DLL
ACIOCR	*PDD	Powers down the output drivers for address/command lane signal I/Os. Different groups of signals have dedicated driver power-down control registers to allow finer selection of signals to power down, especially that some signals, such as CKE and RST#, are required to remain powered up when the SDRAM is in self-refresh mode. Each rank CS# signal and each CK/CK# pair has dedicated driver power down control registers, with the other rank-specific

		signals (CKE and ODT) of each rank being controlled by separate power down control registers in a separate PUB register (DSGCR). There is also a dedicated driver power down control register for SDRAM reset signal. However, the rest of the signals going to the SDRAM (address, bank address, RAS#, CAS#, WE#, and PAR_IN) share a common driver power down register just dedicated for this group. The LPDDR TPD signal has a dedicated output driver power down control register in a separate PUB register (DSGCR).
ACIOCR	*PDR	Powers down the input receivers for address/command lane signal I/Os. Different groups of signals have dedicated receiver power-down control registers toallow finer selection of signals to power down. Each rank and each CK/CK# pair has dedicated receiver power down control register, with all rank-specific signals (CKE, ODT, and CS#) of each rank sharing a common, but rank-specific, receiver power down control register. There is also a dedicated receiver power down control register for SDRAM reset pins. However, the rest of the signals going to the SDRAM (address, bank address, RAS#, CAS#, WE#, PAR_IN, TPD) share a common receiver power down register just dedicated for this group.
DXCCR	DXPDD	Powers down the output drivers for DQ, DM, and DQS/DQS# signal I/Os of all byte lanes. This is a convenient way of powering down the output drivers of all byte lane I/Os with just a single register write. In addition to this, each byte has a dedicated output driver power-down register control to allow only selected bytes to be powered down.
DXCCR	DXPDR	Powers down the input receivers for DQ, DM, and DQS/DQS# signal I/Os of all byte lanes. It also powers down the PDQSR cells of all bytes. This is a convenient way of powering down the input receivers of all byte lane I/Os with just a single register write. In addition to this, each byte has a dedicated input receiver power-down register control to allow only selected bytes to be powered down.
DSGCR	CKEPDD	Powers down the output drivers for CKE I/Os. Each rank CKE has a dedicated driver power down control register to allow finer control of CKE I/O driver power-down, especially that the CKE I/O driver of an SDRAM that is in self refresh is required to remain powered up.
DSGCR	ODTPDD	Powers down the output drivers for ODT I/Os. Each rank ODT has a dedicated driver power down control register to allow finer control of ODT I/O driver power-down, especially that the ODT I/O driver of an SDRAM that is in self refresh or power down mode may be required in certain DDR modes to remain powered up.
DSGCR	TPDPD	Powers down the output driver for the optional LPDDR TPD signal I/O.
DSGCR	NL2PD	Powers down the output driver and the input receiver on the I/O for non-LPDDR2 signals (ODT, RAS#, CAS#, WE#, and BA). This may be used when a chip that is designed for both LPDDR2 and other DDR modes is being used in LPDDR2 mode, in which case one may want to power down the unused I/Os. This power down control register is in addition to (ORed with) the individual ACIOCR power down control registers for these signals.
ZQnCR0	ZQPD	Powers down the PZQ cell. Each PZQ has a dedicated power down control register.
DXnDLLCR	DLLDIS	Disables (powers down) the byte lane DLL. Each byte lane has a dedicated DLL power down control register.
DXnGCR	DXPDD	Powers down the output drivers for DQ, DM, and DQS/DQS# signal I/Os of the byte lane. Each byte lane has a dedicated output driver power down control register, in conjunction with the global output driver power down control register DXCCR.DXPDD.

DXnGCR	DXPDR	Powers down the input receivers for DQ, DM, and DQS/DQS# signal I/Os of the byte lane. Each byte lane has a dedicated input receiver power down control register, in conjunction with the global input receiver power down control register DXCCR.DXPDR
DXnGCR	DQSRPD	Powers down the PDQSR cells of the byte lane. Each byte lane has a dedicated PDQSR power down control register, in conjunction with the global PDQSR power down control register DXCCR.DXPDR.
PGCR	PDDISDX	Selects whether the I/Os and DLL of a disabled byte should automatically be powered down by the PUB. A byte can be disabled by writing a '0' to the DXnGCR.DXEN register or by using the DFI data byte disable (dfi_data_byte_disable) signal.
DSGCR	LPIOPD	Specifies whether the PHY should respond to the controller-initiated DFI low power opportunity request and power down the I/Os of the PHY.
DSGCR	LPDLLPD	Specifies whether the PHY should respond to the controller-initiated DFI low power opportunity request and power down the DLL of the PHY if the requested wakeup time is greater than 2048 clock cycles

DFI-Initiated Power-Down

There are two ways how the controller can initiate PHY power down through the DFI interface. The first method is when the controller asserts the DFI data byte disable (dfi_data_byte_disable) signal during initialization when the DFI initialization start (dfi_init_start) signal is high. In this state, the PHY will power down the DLL and I/Os of the selected bytes if it is configured through DSGCR.BDISEN to respond to DFI data byte disable and if disabled bytes are configured through PGCR. PDDISDX to be powered down. The DFI data byte disable feature is normally used as a static configuration to disable bytes that are not being used.

The controller can also initiate PHY power down by using the DFI low power control interface. This is a dynamic low power request-acknowledge protocol that the controller may use to put the PHY into low power mode when it is not being used for a prolonged time. The PHY will acknowledge a low power request from the controller and power down I/Os and DLLs if it is configured to do so through DSGCR.LPIOPD and DSGCR.LPDLLPD. If the low power wakeup time requested by the controller is less than 2048 clock cycles, then only the I/Os will be powered down. Otherwise if the wakeup time is equal to or more than 2048 cycles, then the DLLs and the I/Os are all powered down. If the DLLs are powered down, then on low power wakeup the PUB will soft reset the DLLs and wait for them to lock before acknowledging the low power wakeup request to the controller.

13.8.9 Dynamic ODT for I/Os

By default the DFI turns on the ODT for the PHY I/Os for DQ/DQS# only when there is read data coming back. This is called dynamic ODT control and is used to reduce power consumed by the termination resistors. The DFI uses the timing of the DQS gating to accurately place the PHY I/O ODT enable signal around the read data. Typically, the DFI turns on the byte ODT enable signal 2 clocks before the pre-amble and turns it off one clock after the post-amble. This guarantees correct setup and hold on the I/Os.

The PHY ODT signal does not go through the ITMs and therefore has to fan out to the DQ/DQS from RTL logic in the PHYCTL. This may result in different timing on these signals depending on the routing. For this
reason various programmable features are provided on the ODT control signals to help mitigate some of the timing issues that may result from different

implementations. These are described in the DXnGCR register. In summary, both the starting position and the width of the enable signal can be adjusted relative to the default position and lengths.

Chapter 14 SMC(Static Memory Controller)

14.1 Overview

The SMC is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip peripheral. It consists of high-performance, area-optimized SRAM memory controllers with on-chip bus interfaces that conform to the AMBA Advanced extensible Interface (AXI) protocol.

The SRAM memory interface type is defined as supportingasynchronous SRAMand NOR flash.

The SMC provides the following features:

- Support asynchronous SRAM and Nor Flash
- Configurable SRAM memory data widths of 8-bit or 16-bit
- AXI data width of 32-bit
- Up to two chip selects, each is up to 16MBbytes
- Programmable cycle timings per chip select.
- Support shared and separated data/address bus
- Support for a remap signal

14.2 Block Diagram

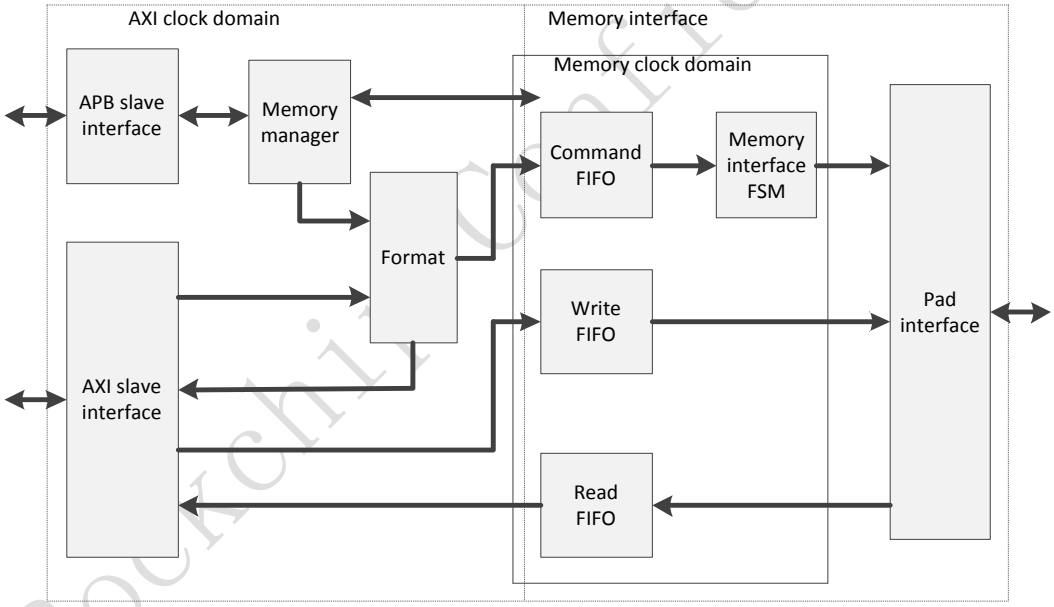


Fig. 14-1 SMC architecture diagram

14.3 Function Description

14.3.1 APB slave interface

The APB interface is a fully-compliant APB slave. The SMC has 4KB of memory allocated to it. The APB slave interface accesses the SMC registers to program the memory system configuration parameters and to provide status information.

The APB interface is clocked by the same clock as the AXI domain clock, aclk, but has a clock enable so that it can be slowed down to execute at an integer divisor of aclk.

14.3.2 Format

The format block receives memory accesses from the AXI slave interface and the memory manager. Requests from AR and AW channels are arbitrated on a round-robin basis. Requests from the manager have the highest priority. The format block also maps AXI transfers onto appropriate memory transfers and passes these to the memory interface through the command FIFO.

Hazard handling

The following types of hazard exist:

- Read after read (RAR)
- Write after write (WAW)
- Read after write (RAW)
- Write after read (WAR).

The AXI specification defines that RAW and WAR ordering is determined by the master, whereas RAR and WAW ordering is enforced by the slave. If an AXI master requires ordering between reads and writes to certain memory locations, it must wait for a write response before issuing a read from a location it has written to (RAW). It must also wait for read data before issuing a write to a location it has read from (WAR). The SMC ensures the ordering of read transfers from a single master is maintained (RAR), and additionally, that the ordering of write transfers from a single master is maintained (WAW).

RAR and WAW hazards only occur in configurations that have two memory interfaces. But we only use one memory interface, so we can ignore them.

SRAM memory accesses

A.Memory address shifting

To produce the address presented to the memory device, the AXI address is aligned to the memory width. This is done because the AXI address is a byte-aligned address, whereas the memory address is a memory-width-aligned address.

During initial configuration of a memory device, the memory mode register can be accessed with a sequence of transfers to specific addresses. You must take into consideration the shifting performance by the SMC when accessing memory mode registers.

B.Memory burst alignment

The SMC provides a programmable option for controlling the formatting of memory transfers with respect to memory burst boundaries, through the burst_align bit of the opmode registers.

When set, the burst_align bit causes memory bursts to be aligned to a memory burst boundary. This setting is intended for use with memories that use the concept of internal pages. This can be an asynchronous page mode memory, or a synchronous PSRAM. If an AXI burst crosses a memory burst boundary, the SMC partitions the AXI transfer into multiple memory bursts, terminating a memory transfer at the burst boundary. Ensure the page size is an integer multiple of the burst length, to avoid a memory burst crossing a page boundary.

When the burst_align bit is not set, the SMC ignores the memory burst boundary when mapping AXI commands onto memory commands. This setting is intended for use with devices such as NOR flash. These devices have no concept of pages.

C.Memory burst length

The SMC enables you to program the memory burst length on an individual chip basis, from length 1 to 32 beats, or a continuous burst. The length of

memory bursts are however automatically limited by the size of the read or write data FIFOs.

For read transfers, the maximum memory burst length on the memory interface is the depth of the read data FIFO. For writes, the maximum burst length is dependent on:

- the beat size of the AXI transfer, asize
- the memory data bus width, mw
- the depth of the write data FIFO depth, wfifo_depth.

The formula to determine the maximum memory write burst length is:

Memory write burst length = ((1<<asize) x wfifo_depth) / (1<<mw)

14.3.3 Memory manager

The memory manager tracks and controls the current state of the SMC aclk domain FSM. The block is responsible for Updating register values that are used in the mclk domain, and controlling direct commands issued to memory.

Chip configuration registers

The SMC provides a mechanism for synchronizing the switching of operating modes with that of the memory device.

The SMC_SET_CYCLES Register and SMC_SET_OPMODE Register act as holding registers for new operating parameters until the SMC detects the memory device has switched modes. This enables a memory device to be made to change its operating mode while still being accessed.

The manager register bank consists of all the timing parameters chip<x>_cycles, and access modes chip<x>_opmode. These are required for the SMC to correctly time any type of access to a supported memory type.

The APB registers SMC_SET_CYCLES and SMC_SET_OPMODE act as holding registers, the configuration registers within the manager are only updated if either:

- the Direct Command Register indicates only a register update is taking place
- the SMC_DIRECT_CMD Register indicates a mode register access either using the SMC_DIRECT_CMD Register or using the AXI interface and the command has completed

Direct commands

The SMC enables code to be executed from the memory while simultaneously, from the software perspective, moving the same chip to a different operating mode. This is achieved by synchronizing the update of the chip configuration registers from the holding registers with the dispatch of the memory configuration register write.

The SMC provides software mechanisms for simultaneously updating the controller and memory configuration registers.

For memories that require a sequence of read and write commands, for example, most NOR flash devices use the AXI interface, with the write data bus used to indicate when the last transfer has completed and when it is safe for the SMC to update the chip configuration registers.

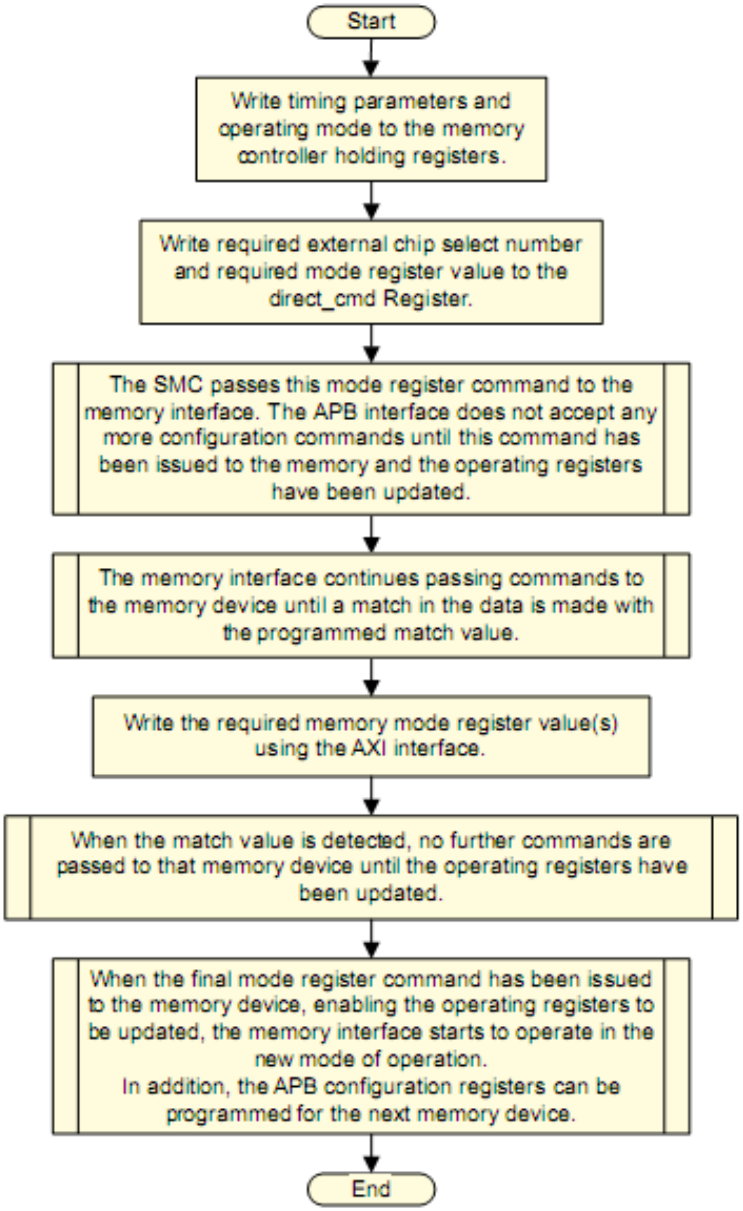


Fig. 14-2Software Mechanism of Direct Commands in SMC

14.3.4 Memory interface

The SMC supports SRAM memory interface type, it is composed of command, read data, and write data FIFOs plus a control FSM. The memory interface FSM is specific to SRAM.

The memory interface issues commands to the memory from the command FIFO, and controls the cycle timings of these commands. It only issues a new command after the previous command is complete and any turn-around times have been met. It only issues a read command when there is space for all the impending data in the read data FIFO.

The SMC does not perform WRAP transfers on the memory interface. For memory devices that only operate in WRAP mode, you must program the SMC_SET_OPMODE Register to align transfers to a memory burst boundary. If the SMC is programmed to perform transfers that cross a memory boundary, then you must program the memory device to operate in INCR mode.

14.3.5 Pad interface

The pad interface module provides a registered I/O interface for data and control signals. It also contains interrupt generation logic.

14.3.6 SRAM interface timing diagrams

All address, control, and write data outputs of the SMC are registered on the rising edge of mclkn, equivalent to the falling edge of mclk, for asynchronous accesses.

Read data output by the memory device is also registered on the rising edge of mclkn, equivalent to the falling edge of mclk, for asynchronous reads. For asynchronous accesses, the data is then pushed onto the read data FIFO to be returned by the AXI interface.

Asynchronous read

Following figure show a single asynchronous read transfer with an initial access time, tRC, of 3 cycles and an output enable assertion delay, tCEOE, of one cycle.

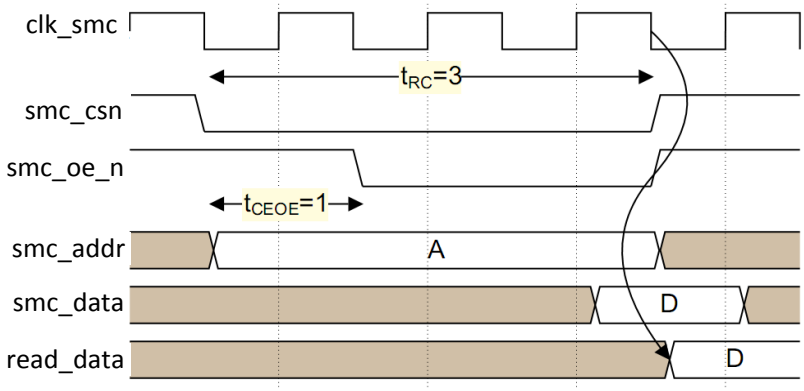


Fig. 14-3 SMC asynchronous read timing

Asynchronous read in multiplexed mode

Following figure show a single asynchronous read transfer in multiplexed SRAM mode, with tRC=7, and tCEOE=5.

In multiplexed mode, both address and data are output by the SMC on the data_out bus. Read data is accepted on the data_in bus. The address is still driven onto the address bus in multiplexed mode. This enables you to use the upper address bits for memories that require more address bits than data bits.

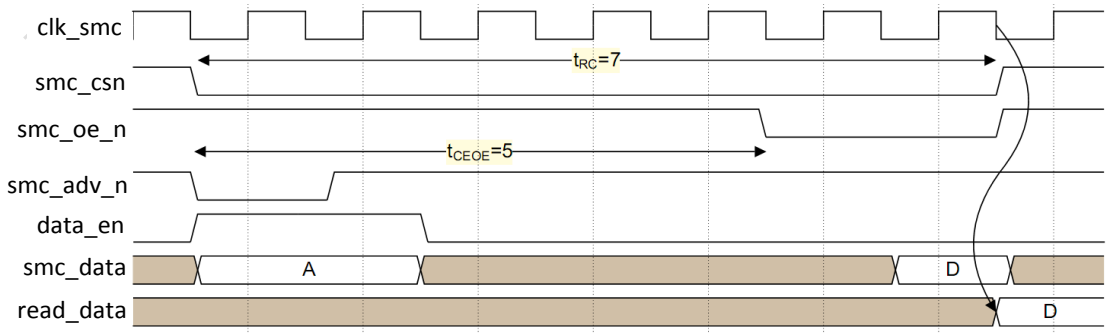


Fig. 14-4 SMC asynchronous read timing in multiplexed mode

Asynchronous write

Following figure show a single asynchronous write with a write cycle time tWC of four cycles and a we_n assertion duration, tWP, of two cycles.

The timing parameter tWP controls the deassertion of we_n. You can use it to vary the hold time of cs_n, addr and data. This differs from the read case where the timing parameter tCEOE controls the delay in the assertion of oe_n. Additionally, we_n is always asserted one cycle after cs_n to ensure the address bus is valid.

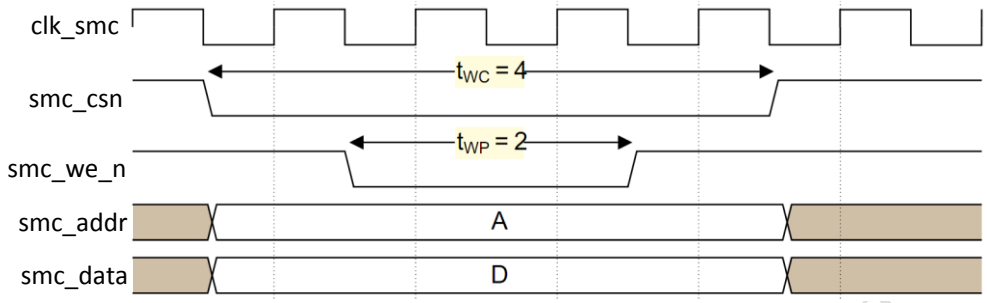


Fig. 14-5 SMC asynchronous write timing

Asynchronous write in multiplexed mode

Following figure show a single asynchronous write in multiplexed mode when the we_time bit is 0. tWC is seven cycles, tWP is four cycles, and the we_time bit programs the assertion of we_n to occur two clock cycles after cs_n goes LOW.

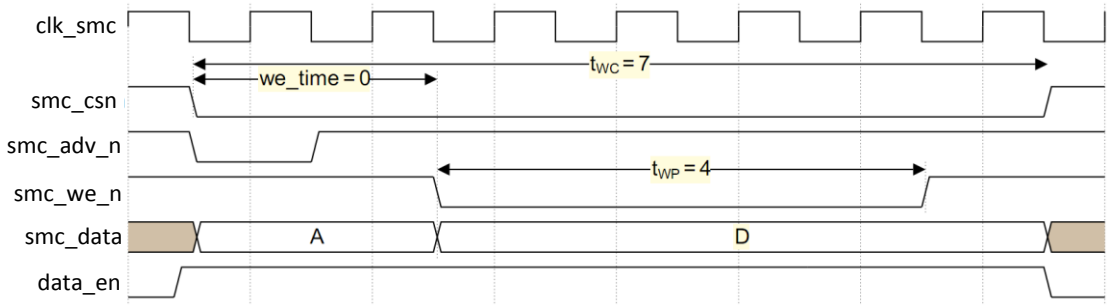


Fig. 14-6 SMC asynchronous write timing in multiplexed mode 1

Following figure show a single asynchronous write in multiplexed mode when the we_time bit is 1. tWC is seven cycles, tWP is four cycles, and the we_time bit programs the assertion of we_n to occur when cs_n goes LOW

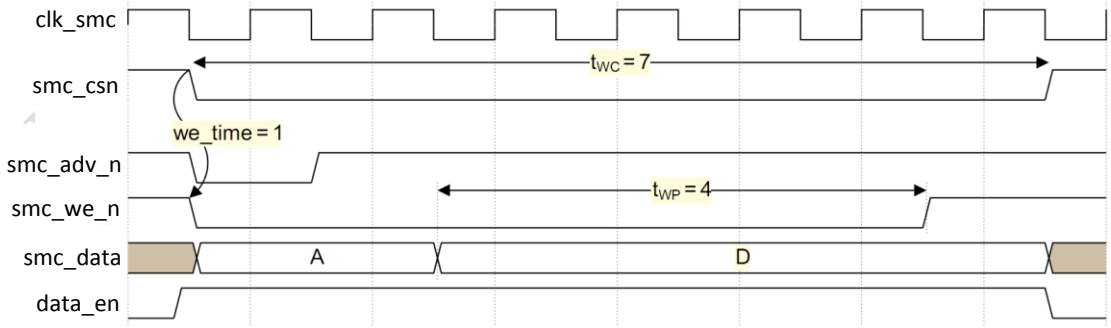


Fig. 14-7 SMC asynchronous write timing in multiplexed mode 2

Asynchronous page mode read

Following figure show a page read access, with an initial access time, tRC, of three cycles, an output enable assertion delay, tCEOE, of two cycles, and a page access time, tPC, of one cycle.

You enable Page mode in the SMC by setting the opmode Register for the relevant chip to asynchronous reads, and the burst length to the page size.

Multiplexed mode page accesses are not supported.

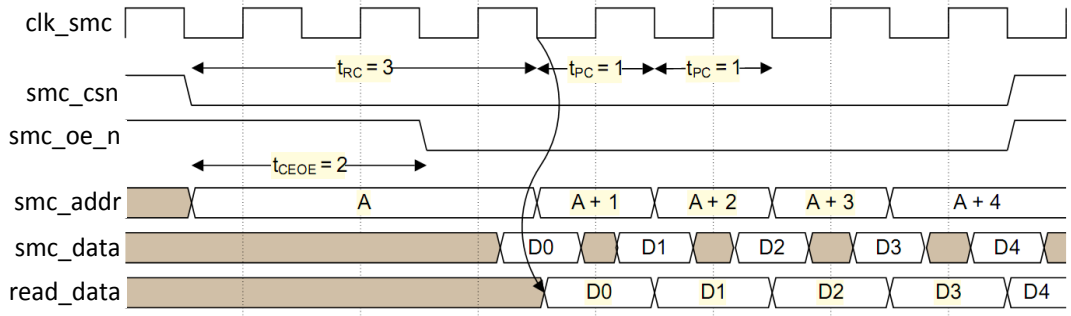


Fig. 14-8 SMC page read timing

14.4 Register Description

This section describes the control/status registers of the design.

14.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
SMC_MEMC_STATUS	0x0000	W	0x00000000	Memory Controller Status Register
SMC_MEMIF_CFG	0x0004	W	0x00000000	Memory Interface Configuration Register
SMC_MEMC_CFG_SET	0x0008	W	0x00000000	Set Configuration Register
SMC_SMC_MEMC_CFG_CLR	0x000c	W	0x00000000	Clear Configuration Register
SMC_DIRECT_CMD	0x0010	W	0x00000000	Direct Command Register
SMC_SET_CYCLES	0x0014	W	0x00000000	set_cycles Register
SMC_SET_OPMODE	0x0018	W	0x00000000	set_opmode Register
SMC_REFRESH_PERIOD0	0x0020	W	0x00000000	refresh_period_0 Register
SMC_SRAM_CYCLES0	0x0100	W	0x00000000	sram_cycles Register for CS0
SMC_OPMODE0	0x0104	W	0x00000000	opmode Register for CS0
SMC_SRAM_CYCLES1	0x0120	W	0x00000000	sram_cycles Register for CS1
SMC_OPMODE1	0x0124	W	0x00000000	opmode Register for CS1

Notes:

*Size: **B**– Byte (8 bits) access, **HW**– Half WORD (16 bits) access, **W**–WORD (32 bits) access*

14.4.2 Detail Register Description

SMC_MEMC_STATUS

Address: Operational Base + offset (0x0000)

Memory Controller Status Register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RO	0x0	raw_int_status0 Current raw interrupt status for interface 0
4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3	RO	0x0	int_status0 Current interrupt status for interface 0
2	RO	0x0	reserved
1	RO	0x0	int_en0 Status of memory interface 0 interrupt enable
0	RO	0x0	state Operating state of the SMC: 0 = SMC is in the ready state 1 = SMC is in the low-power state.

SMC_MEMIF_CFG

Address: Operational Base + offset (0x0004)

Memory Interface Configuration Register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17:16	RO	0x0	exclusive_monitors exclusive_monitor - Returns the number of exclusive access monitor resources that are implemented in the SMC. b00 = 0 monitors b01 = 1 monitor b10 = 2 monitors b11 = 4 monitors.
15:7	RO	0x0	reserved
6	RO	0x0	remap0 remap0 - Returns the value of the remap_0 input.
5:4	RO	0x0	memory_width0 memory_width0 - Returns the maximum width of the SMC memory data bus for interface 0: b00 = 8 bits b01 = 16 bits b10 = 32 bits b11 = reserved.
3:2	RO	0x0	memory_chips0 memory_chips0 - Returns the number of different chip selects that the memory interface 0 supports: b00 = 1 chip b01 = 2 chips b10 = 3 chips b11 = 4 chips.
1:0	RO	0x0	memory_type0 memory_type0 - Returns the memory interface 0 type: b00 = reserved b01 = SRAM non-multiplexed b10 = NAND b11 = SRAM multiplexed.

SMC_MEMC_CFG_SET
Address: Operational Base + offset (0x0008)
Set Configuration Register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	WO	0x0	low_power_req 0 = No effect 1 = Request the SMC to enter low-power state when it next becomes idle.
1	RO	0x0	reserved
0	WO	0x0	int_enable0 0 = No effect 1 = Interrupt enable, memory interface 0.

SMC_SMC_MEMC_CFG_CLR
Address: Operational Base + offset (0x000c)
Clear Configuration Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3	WO	0x0	int_clr_0 0 = No effect 1 = Clear SMC Interrupt 0 (as an alternative to an AXI read).
2	WO	0x0	low_power_exit 0 = No effect 1 = Request the SMC to exit low-power state.
1	RO	0x0	reserved
0	WO	0x0	int_disable0 0 = No effect 1 = Interrupt disable, memory interface 0.

SMC_DIRECT_CMD
Address: Operational Base + offset (0x0010)
Direct Command Register

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:23	WO	0x0	chip_select Selects chip configuration register bank to update, and enables chip mode register access depending on cmd_type. The encoding is: b000-b011 = Chip selects 1-4 on interface 0.
22:21	WO	0x0	cmd_type Selects the command type: b00 = UpdateRegs and AXI b01 = ModeReg b10 = UpdateRegs b11 = ModeReg and UpdateRegs

Bit	Attr	Reset Value	Description
20	WO	0x0	set_cre Maps to the configuration register enable signal, cre, when a ModeReg command is issued. The encoding is: 0 = cre is LOW 1 = cre is HIGH when ModeReg write occurs.
19:0	WO	0x00000	addr When cmd_type = UpdateRegs and AXI then: bits [15:0] are used to match wdata[15:0] bits [19:16] are reserved. Write as zero. When cmd_type = ModeReg or ModeReg and UpdateRegs, these bits map to the external memory address bits [19:0]. When cmd_type = UpdateRegs, these bits are reserved. Write as zero

SMC_SET_CYCLES
Address: Operational Base + offset (0x0014)
set_cycles Register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:20	WO	0x0	set_t6 Contains the value to be written towe_time bit of the sram_cycles Register.
19:17	WO	0x0	set_t5 Contains the value to be written tot_tr field of the sram_cycles Register.
16:14	WO	0x0	set_t4 Contains the value to be written tot_pc field of the sram_cycles Register.
13:11	WO	0x0	set_t3 Contains the value to be written to the t_wp field in sram_cycles Register.
10:8	WO	0x0	set_t2 Contains the value to be written tot_ceoe field of the sram_cycles Register.
7:4	WO	0x0	set_t1 Contains the value to be written to the t_wc field in sram_cycles Register.
3:0	WO	0x0	set_t0 Contains the value to be written to the t_rc field in sram_cycles Register.

SMC_SET_OPMODE
Address: Operational Base + offset (0x0018)
set_opmode Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:13	WO	0x0	set_burst_align Contains the value to be written to the specific SRAM chip opmode Register burst_align field. When you configure the SMC to perform synchronous transfersa, these bits control if memory bursts are split on memory burst boundaries: b000 = bursts can cross any address boundary b001 = burst split on memory burst boundary, that is, 32 beats for continuous b010 = burst split on 64 beat boundary b011 = burst split on 128 beat boundary b100 = burst split on 256 beat boundary b101-b111 = reserved.
12	WO	0x0	set_bls Contains the value to be written to the specific SRAM chip opmode Register byte lane strobe (bls) bit. This bit affects the assertion of the byte-lane strobe outputs. 0 = bls timing equals chip select timing. This is the default setting. 1 = bls timing equals we_n timing. This setting is used for eight memories that have no bls_n inputs. In this case, the bls_n output of the SMC is connected to the we_n memory input.
11	WO	0x0	set_adv Contains the value to be written to the specific SRAM chip opmode Register address valid (adv) bit. The memory uses the address advance signal adv_n when set.
10	WO	0x0	set_baa Contains the value to be written to the specific SRAM chip opmode Register burst address advance (baa) bit. The memory uses the baa_n signal when set.
9:7	WO	0x0	set_wr_bl Contains the value to be written to the specific SRAM chip opmode Register wr_bl field. Encodes the memory burst length: b000 = 1 beat b001 = 4 beats b010 = 8 beats b011 = 16 beats b100 = 32 beats b101 = continuous b110-b111 = reserved.
6	WO	0x0	set_wr_sync Contains the value to be written to the specific SRAM chip opmode Register wr_sync bit. The memory writes are synchronous when set.

Bit	Attr	Reset Value	Description
5:3	WO	0x0	set_rd_bl Contains the value to be written to the specific SRAM chip opmode Register rd_bl field. Encodes the memory burst length: b000 = 1 beat b001 = 4 beats b010 = 8 beats b011 = 16 beats b100 = 32 beats b101 = continuous b110-b111 = reserved.
2	WO	0x0	set_rd_sync Contains the value to be written to the specific SRAM chip opmode Register rd_sync bit. Memory in sync mode when set.
1:0	WO	0x0	set_mw Contains the value to be written to the specific chip opmode Register memory width (mw) field. Encodes the memory data bus width: b00 = 8 bits b01 = 16 bits b10 = 32 bits b11 = reserved. You can program this to the configured width, or half that width.

SMC_REFRESH_PERIOD0

Address: Operational Base + offset (0x0020)

refresh_period_0 Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	period Sets the number of consecutive memory bursts that are permitted, prior to the SMC deasserting chip select to enable the PSRAM to initiate a refresh cycle. The options are: b0000 = disables the insertion of idle cycles between consecutive bursts b0001 = an idle cycle occurs after each burst b0010 = an idle cycle occurs after 2 consecutive bursts b0011 = an idle cycle occurs after 3 consecutive bursts b0100 = an idle cycle occurs after 4 consecutive bursts . . . b1111 = an idle cycle occurs after 15 consecutive bursts.

SMC_SRAM_CYCLES0
Address: Operational Base + offset (0x0100)
sram_cycles Register for CS0

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	RO	0x0	we_time For asynchronous multiplexed transfers this bit controls when the SMC asserts we_n: 0 = SMC asserts we_n two mclk cycles after asserting cs_n. 1 = SMC asserts we_n and cs_n together
19:17	RO	0x0	t_tr Turnaround time for SRAM chip configurations. Minimum permitted value = 1.
16:14	RO	0x0	t_pc Page cycle time for SRAM chip configurations. Minimum permitted value = 1
13:11	RO	0x0	t_wp we_n assertion delay. Minimum permitted value = 1.
10:8	RO	0x0	t_ceoe oe_n assertion delay for SRAM chip configurations. Minimum permitted value = 1.
7:4	RO	0x0	t_wc Write cycle time. Minimum permitted value = 2.
3:0	RO	0x0	t_rc Read cycle time. Minimum permitted value = 2.

SMC_OPMODE0
Address: Operational Base + offset (0x0104)
opmode Register for CS0

Bit	Attr	Reset Value	Description
31:24	RO	0x00	address_match Returns the value of this tie-off. This is the comparison value for address bits [31:24] to determine the chip that is selected.
23:16	RO	0x00	address_mask Returns the value of this tie-off. This is the mask for address bits[31:24] to determine the chip that must be selected. A logic 1 indicates the bit is used for comparison.

Bit	Attr	Reset Value	Description
15:13	RO	0x0	burst_align When you configure the SMC to perform synchronous transfersa, these bits control if memory bursts are split on memory burst boundaries: b000 = bursts can cross any address boundary b001 = burst split on memory burst boundary, that is, 32 beats for continuous b010 = burst split on 64 beat boundary b011 = burst split on 128 beat boundary b100 = burst split on 256 beat boundary b101-b111 = reserved.
12	RO	0x0	bls This bit affects the assertion of the byte-lane strobe outputs: 0 = bls timing equals chip select timing. This is the default setting. 1 = bls timing equals we_n timing. This setting is used for 8-bit memories that have no bls inputs. In this case, the bls_n output of the SMC is connected to the we_n memory input.
11	RO	0x0	adv The memory uses the address advance signal, adv_n, when set.
10	RO	0x0	baa The memory uses the burst address advance signal, baa_n, when set.
9:7	RO	0x0	wr_bl Selects the memory burst length for writes: b000 = 1 beat b001 = 4 beats b010 = 8 beats b011 = 16 beats b100 = 32 beats b101 = continuous b110-b111 = reserved.
6	RO	0x0	wr_sync When set, the memory operates in write sync mode.
5:3	RO	0x0	rd_bl Selects the memory burst length for reads: b000 = 1 beat b001 = 4 beats b010 = 8 beats b011 = 16 beats b100 = 32 beats b101 = continuous b110-b111 = reserved.
2	RO	0x0	rd_sync When set, the memory operates in read sync mode.

Bit	Attr	Reset Value	Description
1:0	RO	0x0	mw Selects the SMC memory data bus width: b00 = 8 bits b01 = 16 bits b10 = 32 bits b11 = reserved.

SMC_SRAM_CYCLES1

Address: Operational Base + offset (0x0120)

sram_cycles Register for CS1

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	RO	0x0	we_time For asynchronous multiplexed transfers this bit controls when the SMC asserts we_n: 0 = SMC asserts we_n two mclk cycles after asserting cs_n. 1 = SMC asserts we_n and cs_n together
19:17	RO	0x0	t_tr Turnaround time for SRAM chip configurations. Minimum permitted value = 1.
16:14	RO	0x0	t_pc Page cycle time for SRAM chip configurations. Minimum permitted value = 1
13:11	RO	0x0	t_wp we_n assertion delay. Minimum permitted value = 1.
10:8	RO	0x0	t_ceoe oe_n assertion delay for SRAM chip configurations. Minimum permitted value = 1.
7:4	RO	0x0	t_wc Write cycle time. Minimum permitted value = 2.
3:0	RO	0x0	t_rc Read cycle time. Minimum permitted value = 2.

SMC_OPMODE1

Address: Operational Base + offset (0x0124)

opmode Register for CS1

Bit	Attr	Reset Value	Description
31:24	RO	0x00	address_match Returns the value of this tie-off. This is the comparison value for address bits [31:24] to determine the chip that is selected.

Bit	Attr	Reset Value	Description
23:16	RO	0x00	address_mask Returns the value of this tie-off. This is the mask for address bits[31:24] to determine the chip that must be selected. A logic 1 indicates the bit is used for comparison.
15:13	RO	0x0	burst_align When you configure the SMC to perform synchronous transfersa, these bits control if memory bursts are split on memory burst boundaries: b000 = bursts can cross any address boundary b001 = burst split on memory burst boundary, that is, 32 beats for continuous b010 = burst split on 64 beat boundary b011 = burst split on 128 beat boundary b100 = burst split on 256 beat boundary b101-b111 = reserved.
12	RO	0x0	bls This bit affects the assertion of the byte-lane strobe outputs: 0 = bls timing equals chip select timing. This is the default setting. 1 = bls timing equals we_n timing. This setting is used for 8-bit memories that have no bls inputs. In this case, the bls_n output of the SMC is connected to the we_n memory input.
11	RO	0x0	adv The memory uses the address advance signal, adv_n, when set.
10	RO	0x0	baa The memory uses the burst address advance signal, baa_n, when set.
9:7	RO	0x0	wr_bl Selects the memory burst length for writes: b000 = 1 beat b001 = 4 beats b010 = 8 beats b011 = 16 beats b100 = 32 beats b101 = continuous b110-b111 = reserved.
6	RO	0x0	wr_sync When set, the memory operates in write sync mode.
5:3	RO	0x0	rd_bl Selects the memory burst length for reads: b000 = 1 beat b001 = 4 beats b010 = 8 beats b011 = 16 beats b100 = 32 beats b101 = continuous b110-b111 = reserved.

Bit	Attr	Reset Value	Description
2	RO	0x0	rd_sync When set, the memory operates in read sync mode.
1:0	RO	0x0	mw Selects the SMC memory data bus width: b00 = 8 bits b01 = 16 bits b10 = 32 bits b11 = reserved.

*Notes: Attr: **RW**– Read/writable, **RO**– read only, **WO**–write only*

14.5 Timing Diagram

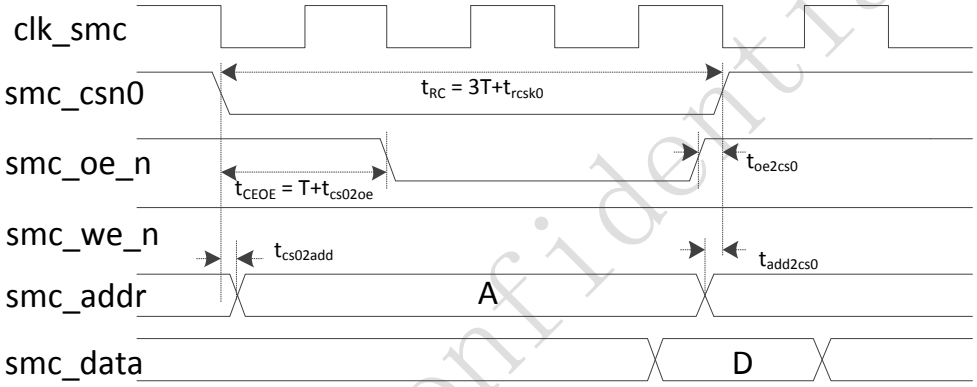


Fig. 14-9 SMC timing diagram of asynchronous read

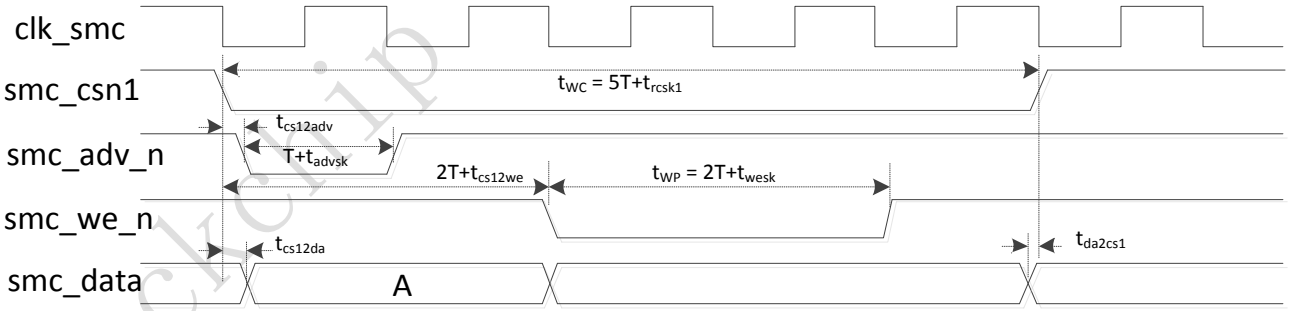


Fig. 14-10Asynchronous Write Timing Diagram In Multiplexed Mode

Table 14-1Meaning of The Parameter in Fig.14-9 and Fig.14-10

Parameter	Description	min	typ	max	unit
t_{rcsk0}	Rise and fall skew for smc_csn0	-273	-385	-544	ps
t_{rcsk1}	Rise and fall skew for smc_csn1	-121	-128	-102	ps
t_{wesk}	Rise and fall skew for smc_we_n	-213	-276	-403	ps
t_{cs02oe}	smc_csn0 valid to smc_oe_n valid skew	-58	-85	-148	ps
t_{cs12oe}	smc_csn1 valid to smc_oe_n valid skew	-1483	-2218	-3234	ps
t_{oe2cs0}	smc_oe_n invalid to smc_csn0 invalid skew	29	33	21	ps
t_{oe2cs1}	smc_oe_n invalid to smc_csn1 invalid skew	1606	2423	3549	ps
$t_{cs02add}$	smc_csn0 valid to smc_addr valid skew	-213	-303	-415	ps
$t_{cs12add}$	smc_csn1 valid to smc_addr valid skew	-1638	-2436	-3501	ps

$t_{add2cs0}$	smc_addr invalid to smc_csn0 invalid skew	-1658	-2515	-3426	ps
$t_{add2cs1}$	smc_addr invalid to smc_csn1 invalid skew	-81	-125	102	ps
t_{cs02we}	smc_csn0 valid to smc_we_n valid skew	-113	-178	-235	ps
t_{cs12we}	smc_csn1 valid to smc_we_n valid skew	-1538	-2311	-3321	ps
t_{advsk}	Rise and fall skew for smc_adv_n	-195	-243	-305	ps
$t_{cs02adv}$	smc_csn0 valid to smc_adv_n valid skew	89	111	135	ps
$t_{cs12adv}$	smc_csn1 valid to smc_adv_n valid skew	-1336	-2022	-2951	ps
t_{cs02da}	smc_csn0 valid to smc_data valid skew	-3	99	207	ps
t_{cs12da}	smc_csn1 valid to smc_data valid skew	-1422	-2034	-2879	ps
t_{da2cs0}	smc_data invalid to smc_csn0 invalid skew	-530	-826	-1283	ps
t_{da2cs1}	smc_data invalid to smc_csn1 invalid skew	1047	1564	2245	ps

14.6 Interface Description

SMC IOs are multiplexed with GPIO as following Table

Table 14-2 SMC interface mux description

ModulePin	Direction	Pad Name	IOMUX Setting
smc_oe_n	O	GPIO0_D[2]	GRF_GPIO0D_IOMUX[5:4] = 10
smc_bls_n1	O	GPIO2_C[2]	GRF_GPIO2C_IOMUX[5:4] = 10
smc_bls_n0	O	GPIO2_C[1]	GRF_GPIO2C_IOMUX[3:2] = 10
smc_we_n	O	GPIO0_D[1]	GRF_GPIO0D_IOMUX[3:2] = 10
smc_addr0	O	GPIO0_D[4]	GRF_GPIO0D_IOMUX[9:8] = 10
smc_addr1	O	GPIO0_D[5]	GRF_GPIO0D_IOMUX[11:10] = 10
smc_addr2	O	GPIO0_C[6]	GRF_GPIO0C_IOMUX[13:12] = 10
smc_addr3	O	GPIO0_C[7]	GRF_GPIO0C_IOMUX[15:14] = 10
smc_addr4	O	GPIO2_A[0]	GRF_GPIO2A_IOMUX[1:0] = 10
smc_addr5	O	GPIO2_A[1]	GRF_GPIO2A_IOMUX[3:2] = 10
smc_addr6	O	GPIO2_A[2]	GRF_GPIO2A_IOMUX[5:4] = 10
smc_addr7	O	GPIO2_A[3]	GRF_GPIO2A_IOMUX[7:6] = 10
smc_addr8	O	GPIO2_A[4]	GRF_GPIO2A_IOMUX[9:8] = 10
smc_addr9	O	GPIO2_A[5]	GRF_GPIO2A_IOMUX[11:10] = 10
smc_addr10	O	GPIO2_A[6]	GRF_GPIO2A_IOMUX[13:12] = 10
smc_addr11	O	GPIO2_A[7]	GRF_GPIO2A_IOMUX[15:14] = 10
smc_addr12	O	GPIO2_B[0]	GRF_GPIO2B_IOMUX[1:0] = 10
smc_addr13	O	GPIO2_B[1]	GRF_GPIO2B_IOMUX[3:2] = 10
smc_addr14	O	GPIO2_B[2]	GRF_GPIO2B_IOMUX[5:4] = 10
smc_addr15	O	GPIO2_B[3]	GRF_GPIO2B_IOMUX[7:6] = 10
smc_addr16	O	GPIO2_B[4]	GRF_GPIO2B_IOMUX[9:8] = 10
smc_addr17	O	GPIO2_B[5]	GRF_GPIO2B_IOMUX[11:10] = 10
smc_addr18	O	GPIO2_B[6]	GRF_GPIO2B_IOMUX[13:12] = 10
smc_addr19	O	GPIO2_B[7]	GRF_GPIO2B_IOMUX[15:14] = 10
smc_csn1	O	GPIO2_D[1]	GRF_GPIO2D_IOMUX[3:2] = 10
smc_csn0	O	GPIO0_D[0]	GRF_GPIO0D_IOMUX[1:0] = 10
smc_adv_n	O	GPIO0_D[3]	GRF_GPIO0D_IOMUX[7:6] = 10
smc_data0	I/O	GPIO4_C[0]	GRF_GPIO4C_IOMUX[1:0] = 01
smc_data1	I/O	GPIO4_C[1]	GRF_GPIO4C_IOMUX[3:2] = 01
smc_data2	I/O	GPIO4_C[2]	GRF_GPIO4C_IOMUX[5:4] = 01
smc_data3	I/O	GPIO4_C[3]	GRF_GPIO4C_IOMUX[7:6] = 01
smc_data4	I/O	GPIO4_C[4]	GRF_GPIO4C_IOMUX[9:8] = 01
smc_data5	I/O	GPIO4_C[5]	GRF_GPIO4C_IOMUX[11:10] = 01
smc_data6	I/O	GPIO4_C[6]	GRF_GPIO4C_IOMUX[13:12] = 01

smc_data7	I/O	GPIO4_C[7]	GRF_GPIO4C_IOMUX[15:14] = 01
smc_data8	I/O	GPIO4_D[0]	GRF_GPIO4D_IOMUX[1:0] = 01
smc_data9	I/O	GPIO4_D[1]	GRF_GPIO4D_IOMUX[3:2] = 01
smc_data10	I/O	GPIO4_D[2]	GRF_GPIO4D_IOMUX[5:4] = 01
smc_data11	I/O	GPIO4_D[3]	GRF_GPIO4D_IOMUX[7:6] = 01
smc_data12	I/O	GPIO4_D[4]	GRF_GPIO4D_IOMUX[9:8] = 01
smc_data13	I/O	GPIO4_D[5]	GRF_GPIO4D_IOMUX[11:10] = 01
smc_data14	I/O	GPIO4_D[6]	GRF_GPIO4D_IOMUX[13:12] = 01
smc_data15	I/O	GPIO4_D[7]	GRF_GPIO4D_IOMUX[15:14] = 01

14.7 Application Notes

14.7.1 multiplexed address/data mode

When SMC memory interface operates in multiplexed address/data mode, you first must config bit 6 of register GRF_SOC_CON1 in GRF(General Register Files) as 1.

In multiplexed address/data mode, smc_addr[15:0] are multiplexed with smc_data0 - smc_data15, so you don't need config GPIO mux of smc_addr[15:0], but smc_addr[19:16] need config GPIO mux. If SMC has be set to 8 bits data bus width, then smc_data8 - smc_data15 and addr[7:0] don't need config GPIO mux in multiplexed address/data mode and other data and address need.

14.7.2 Booting using the SRAM interface

The SMC enables the lowest SRAM chip select, normally chip 0, to be bootable To enable SRAM memory to be bootable, the SRAM interface does not require any special functionality, other than knowing the memory width of the memory concerned. This is indicated by a top-level tie-off. To enable the SMC to work with the slowest memories, the timing registers reset to the worst-case values. When the remap signal is HIGH, the memory with the bootable chip select is set by the sram_mw [1:0] tie-off signals.

Additionally, while the SMC input remap is HIGH, the bootable chip is aliased to base address 0x0.

You can config the remap and sram_mw via write register GRF_SOC_CON1 bit 3 and bit 5-4.

Chapter 15 NandC(Nand Flash Controller)

15.1 Overview

Nand Flash Controller (NandC) is used to control data transmission from host to flash device or from flash device to host. NandC is connected to AHB BUS through an AHB Master and an AHB Slave. The data transmission between host and external memory can be done through AHB Master Interface or AHB Slave Interface.

15.1.1 Features

- Software Interface Type
  - Support directly mode
  - Support LLP mode
- Flash Interface Type
  - Support Asynchronous Flash Interface with 8bits datawidth (“Asyn8x” for short)
  - Support Asynchronous Flash Interface with 16bits data width (“Asyn16x” for short)
  - Support ONFI Synchronous Flash Interface (“ONFI Syn” for short)
  - Support Toggle Flash Interface (“Toggle” for short)
  - Support 8 flash devices at most
- Flash Type
  - Support Managed NAND Flash(LBA) and Raw NAND Flash(NO-LBA)
  - Support SLC/MLC/TLC Flash
- Flash Interface Timing
  - Asyn8x: configurable timing, one byte per two host clocks at the fastest speed
  - Asyn16x: configurable timing, two bytes per two host clocks at the fastest speed
  - ONFI Syn: configurable timing, two bytes per two host clocksat the fastest speed
  - Toggle: configurable timing, two byte per two host clocks at the fastest speed
- Randomizer Ability
  - Supporttwo randomizer mode with different polynomial
  - Support two randomizer width, 8bit and 16bit parallel
- BCH/ECC Ability
  - 16bit/1KB BCH/ECC: support 16bitBCH/ECC, which can detect and correct up to 16 error bits in every 1K bytes data
  - 24bit/1KB BCH/ECC: support 24bitBCH/ECC, which can detect and correct up to 24 error bits in every 1K bytes data
  - 40bit/1KB BCH/ECC: support 40bitBCH/ECC, which can detect and correct up to 40 error bits in every 1K bytes data
  - 60bit/1KB BCH/ECC: support 60bitBCH/ECC, which can detect and correct up to 60 error bits in every 1K bytes data
  - 8bit/512B BCH/ECC: support 8bitBCH/ECC, which can detect and correct up to 8 error bits in every 512 bytes data
  - 12bit/512B BCH/ECC: support 12bitBCH/ECC, which can detect and correct up to 12 error bits in every 512 bytes data
  - 20bit/512B BCH/ECC: support 20bitBCH/ECC, which can detect and correct up to 20 error bits in every 512 bytes data
  - 30bit/512B BCH/ECC: support 30bitBCH/ECC, which can detect and

correct up to 30 error bits in every 512 bytes data
- 16bit/512B BCH/ECC: support 16bitBCH/ECC, which can detect and correct up to 16 error bits in every 512 bytes data
- 24bit/512B BCH/ECC: support 24bitBCH/ECC, which can detect and correct up to 24 error bits in every 512 bytes data
- 40bit/512B BCH/ECC: support 40bitBCH/ECC, which can detect and correct up to 40 error bits in every 512 bytes data
- 60bit/512B BCH/ECC: support 60bitBCH/ECC, which can detect and correct up to 60 error bits in every 512 bytes data
- Support auto correction for all" FF" code

- Transmission Ability
  - Support 16K bytes data transmission at a time at most
  - Support two transfer working modes: Bypass or DMA
  - Support two transfer codeword size for Managed NAND Flash: 1024 bytes/codeword or 512 bytes/codeword
- Internal Memory
  - 2 built-in srams, and the size is 1k bytes respectively
  - Can be accessed by other masters
  - Can be operated in pingpang mode by other masters

For detailed information about NandC (Nand Flash Controller), please refer to **RK30xx NandC.pdf**。

Chapter 16 eMMC Interface

16.1 Overview

The SDMMC Host Controller is designed to support Secure Digital memory (SD mem - version 3.00), Secure Digital I/O(SDIO-version 3.00), Multimedia Cards(MMC-version 4.41).The SDMMC support SD Card(1/4bit), SDIO, MMC(1/4/8bit).

Feature

- Supports AMBA AHB interface
- Supports DMA controller for data transfers
- Supports interrupt output
- Supports SD version3.0 except SPI mode
- Supports MMC version4.41 except SPI mode
- Supports SDIO version3.0
- Supports programmable baud rate.
- Provides individual clock control to selectively turn ON or OFF clock to a card
- Supports power management and power switch. Provides individual power control to selectively turn ON or OFF power to a card
- Support DDR in 4-bit mode

For detailed information about eMMC Controller, please refer to **RK30xx eMMC Controller.pdf**。

Chapter 17 SD/MMC Card Host Controller

17.1 Overview

The SDMMC Host Controller is designed to support Secure Digital memory (SD mem - version 3.00), Secure Digital I/O(SDIO-version 3.00), Multimedia Cards(MMC-version 4.41).The SDMMC support SD Card(1/4bit), SDIO, MMC(1/4bit).

Feature

- Supports AMBA AHB interface
- Supports DMA controller for data transfers
- Supports interrupt output
- Supports SD version3.0 except SPI mode
- Supports MMC version4.41 except SPI mode
- Supports SDIO version3.0
- Supports programmable baud rate.
- Provides individual clock control to selectively turn ON or OFF clock to a card
- Supports power management and power switch. Provides individual power control to selectively turn ON or OFF power to a card
- Support DDR in 4-bit mode

17.2 Block Diagram

The SD/MMC controller consists of the following main functional blocks, which are illustrated inFig. 17-1.

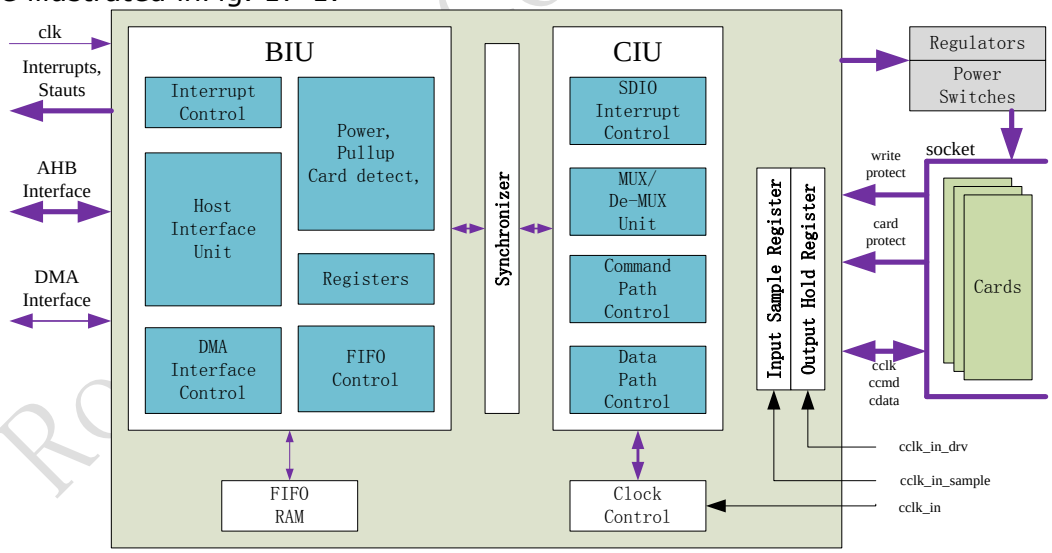


Fig. 17-1SD/MMC Controller Block Diagram

- Bus Interface Unit (BIU) – Provides AMBA AHB and DMA interfaces for register and dataread/writes.
- Card Interface Unit (CIU) – Takes care of the SD_MMC protocols and provides clockmanagement.

17.3 Function description

17.3.1 Bus Interface Unit

The Bus Interface Unit provides the following functions:

- Host interface
- DMA interface
- Interrupt control
- Register access
- FIFO access
- Power control and card detection
- 235x32bit external fifo

Host Interface Unit

The Host Interface Unit is an AHB slave interface, which provides theinterface between the SD/MMC host controller and the host bus. You can configure the host interface as either an AHB.

DMA Interface Unit

DMA signals interface the SD/MMC Host Controller to an external DMA controller to reduce the software overhead during FIFO data transfers. The DMA request/acknowledge handshake is used for only data transfers. The DMA interface provides a connection to the DMA Controller.

On seeing the DMA request, the DMA controller initiates accesses through the host interface to read or write into the data FIFO. The SD/MMC Host Controller has FIFO transmit/receive watermark registers that you can set, depending on system latency. The DMA interface asserts the request in the following cases:

- Read from a card when the data FIFO word count exceeds the Rx-Watermark level
- Write to a card when the FIFO word count is less than or equal to the Tx-Watermark level

When the DMA interface is enabled, you can use normal host read/writes to access the data FIFOs.

Register Unit

The register unit is part of the bus interface unit; it provides read and write access to the registers.

All registers reside in the Bus Interface Unit clock domain. When a command is sent to a card by setting the start_bit, which is bit[31] of the CMD register, all relevant registers needed for the CIU operation are transferred to the CIU block. During this time, the registers that are transferred from the BIU to the CIU should not be written. The software should wait for the hardware to clear the start bit before writing to these registers again. The register unit has a hardware locking feature to prevent illegal writes to registers. The lock is necessary in order to avoid metastability violations, both because the host and card clock domains are different and to prevent illegal software operations.

Once a command start is issued by setting the start_bit of the CMD register, the following registers cannot be reprogrammed until the command is accepted by the card interface unit:

- CMD – Command
- CMDARG – Command Argument
- BYTCNT – Byte Count
- BLKSIZ – Block Size
- CLKDIV – Clock Divider

- CLKENA – Clock Enable
- CLKSRC – Clock Source
- TMOUT – Timeout
- CTYPE – Card Type

The hardware resets the start_bit once the CIU accepts the command. If a host write to any of these registers is attempted during this locked time, then the write is ignored and the hardware lock error bit is set in the raw interrupt status register. Additionally, if the interrupt is enabled and not masked for a hardware lock error, then an interrupt is sent to the host.

When the Card Interface Unit is in an idle state, it typically takes the following number of clocks for the command handshake, where clk is the BIU clock and cclk_in is the CIU clock:

3 (clk) + 3 (cclk_in)

Once a command is accepted, you can send another command to the CIU-which has a one-deep command queue-under the following conditions:

- If the previous command was not a data transfer command, the new command is sent to the SD/MMC card once the previous command completes.
- If the previous command is a data transfer command and if wait_prvdata_complete (bit[13]) of the Command register is set for the new command, the new command is sent to the SD/MMC card only when the data transfer completes.
- If the wait_prvdata_complete is 0, then the new command is sent to the SD/MMC card as soon as the previous command is sent. Typically, you should use this only to stop or abort a previous data transfer or query the card status in the middle of a data transfer.

Interrupt Controller Unit

The interrupt controller unit generates an interrupt that depends on the controller raw interrupt status, the interrupt-mask register, and the global interrupt-enable register bit. Once an interrupt condition is detected, it sets the corresponding interrupt bit in the raw interrupt status register. The raw interrupt status bit stays on until the software clears the bit by writing a 1 to the interrupt bit; a 0 leaves the bit untouched.

The interrupt port, int, is an active-high, level-sensitive interrupt. The interrupt port is active only when any bit in the raw interrupt status register is active, the corresponding interrupt mask bit is 1, and the global interrupt enable bit is 1. The interrupt port is registered in order to avoid any combinational glitches.

The following bits are available as top-level ports for debug purposes:

- Interrupt mask bits (int_mask_n[31:0])
- Raw interrupt status bits (raw_ints[31:0])
- Interrupt enable bit (int_enable)

The int_enable is reset to 0 on power-on, and the interrupt mask bits are set to 32'h0, which masks all the interrupts.

Notes:Before enabling the interrupt, it is always recommended that you write 32'hffff_ffff to the raw interrupt status register in order to clear any pending unserviced interrupts. When clearing interrupts during normal operation, ensure that you clear only the interrupt bits that you serviced.

Table 17-1 Bits in Interrupt Status Register

Bits	Interrupt	Description
24	SDIO Interrupt	-

16	Card No busy Interrupts	If card exit busy status, the interrupt happened
15	End Bit Error (read) /Write no CRC (EBE)	Error in end-bit during read operation, or no data CRC or negative CRC received during write operation. *Notes: For MMC CMD19, there may be no CRC status returned by the card. Hence, EBE is set for CMD19. The application should not treat this as an error.*
14	Auto Command Done (ACD)	Stop/abort commands automatically sent by card unit and not initiated by host; similar to Command Done (CD) interrupt.
13	Start Bit Error (SBE)	Error in data start bit when data is read from a card. In 4-bit mode, if all data bits do not have start bit, then this error is set.
12	Hardware Locked write Error (HLE)	During hardware-lock period, write attempted to one of locked registers.
11	FIFO Underrun/ Overrun Error (FRUN)	Host tried to push data when FIFO was full, or host tried to read data when FIFO was empty. Typically this should not happen, except due to error in software. Card unit never pushes data into FIFO when FIFO is full, and pop data when FIFO is empty.
10	Data Starvation by Host Timeout (HTO)	To avoid data loss, card clock out (sdmmc_clkout) is stopped if FIFO is empty when writing to card, or FIFO is full when reading from card. Whenever card clock is stopped to avoid data loss, data-starvation timeout counter is started with data-timeout value. This interrupt is set if host does not fill data into FIFO during write to card, or does not read from FIFO during read from card before timeout period. Even after timeout, card clock stays in stopped state, with CIU state machines waiting. It is responsibility of host to push or pop data into FIFO upon interrupt, which automatically restarts sdmmc_clkout and card state machines. Even if host wants to send stop/abort command, it still needs to ensure it has to push or pop FIFO so that clock starts in order for stop/abort command to send on cmd signal along with data that is sent or received on data line.
9	Data Read Timeout (DRTO)	Data timeout occurred. Data Transfer Over (DTO) also set if data timeout occurs.
8	Response Timeout (RTO)	Response timeout occurred. Command Done (CD) also set if response timeout occurs. If command involves data transfer and when response times out, no data transfer is attempted by SD/MMC Host Controller.
7	Data CRC Error (DCRC)	Received Data CRC does not match with locally-generated CRC in CIU.
6	Response CRC Error	Response CRC does not match with

	(RCRC)	locally-generated CRC in CIU.
5	Receive FIFO Data Request (RXDR)	Interrupt set during read operation from card when FIFO level is greater than Receive-Threshold level.
4	Transmit FIFO Data Request (TXDR)	Interrupt set during write operation to card when FIFO level reaches less than or equal to Transmit-Threshold level.
3	Data Transfer Over (DTO)	Data transfer completed, even if there is Start Bit Error or CRC error. This bit is also set when "read data-timeout" occurs. *Notes:DTO bit is set at the end of the last data block, even if the device asserts MMC busy after the last data block.*
2	Command Done(CD)	Command sent to card and got response from card, even if Response Error or CRC error occurs. Also set when response timeout occurs
1	Response Error (RE)	Error in received response set if one of following occurs: ● Transmission bit != 0 ● Command index mismatch ● End-bit != 1
0	Card-Detect (CDT)	When card inserted or removed, this interrupt occurs. Software should read card-detect register (CDETECT, 0x50) to determine current card status.

FIFO Controller Unit

The FIFO controller interfaces the external FIFO to the host/DMA interface and the card controller unit. When FIFO overrun and underrun conditions occur, the card clock stops in order to avoid data loss.

The FIFO uses a two-clock synchronous read and synchronous write dual-port RAM. One of the ports is connected to the host clock, clk, and the second port is connected to the card clock, cclk_in.

Notes: The FIFO controller does not support simultaneous read/write access from the same port. For debugging purposes, the software may try to write into the FIFO and read back the data; results are indeterminate, since the design does not support read/write access from the same port.

Power Control and Card Detection Unit

The register unit has registers that control the power and MMC open-drain pullup. Power to each card can be selectively turned on or off.

The card detection unit looks for any changes in the card-detect signals for card insertion or card removal. It filters out the debounces associated with mechanical insertion or removal, and generates one interrupt to the host. You can program the debounce filter value.

On power-on, the controller should read in the card_detect port and store the value in the memory. Upon receiving a card-detect interrupt, it should again read the card_detect port and XOR with the previous card-detect status to find out which card has interrupted. If more than one card is simultaneously removed or inserted, there is only one card-detect interrupt; the XOR value indicates which cards have been disturbed. The memory should be updated with the new card-detect value.

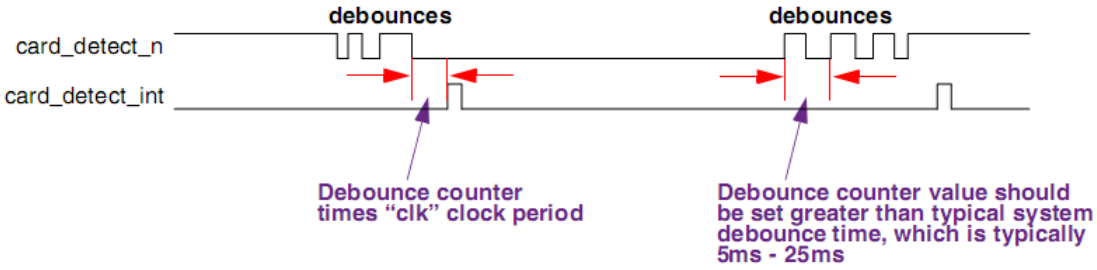


Fig. 17-2SD/MMC Card-Detect Signal

17.3.2 Card Interface Unit

The Card Interface Unit interfaces (CIU) with the Bus Interface Unit (BIU) and the SD/MMC cards or devices. The host writes command parameters to the SD/MMC Host Controller BIU control registers, and these parameters are then passed to the CIU. Depending on control register values, the CIU generates SD/MMC command and data traffic on a selected card bus according to SD/MMC protocol. The SD/MMC Host Controller accordingly controls the command and data path.

The following software restrictions should be met for proper CIU operation:

- Only one data transfer command can be issued at a time.
- During an open-ended card write operation, if the card clock is stopped because the FIFO is empty, the software must first fill the data into the FIFO and start the card clock. It can then issue only a stop/abort command to the card.
- During an SDIO card transfer, if the card function is suspended and the software wants to resume the suspended transfer, it must first reset the FIFO and start the resume command as if it were a new data transfer command.
- When issuing card reset commands (CMD0, CMD15 or CMD52_reset) while a card data transfer is in progress, the software must set the stop_abort_cmd bit in the Command register so that the SD/MMC Host Controller can stop the data transfer after issuing the card reset command.
- When the data end bit error is set in the RINTSTS register, the SD/MMC Host Controller does not guarantee SDIO interrupts. The software should ignore the SDIO interrupts and issue the stop/abort command to the card, so that the card stops sending the read data.
- If the card clock is stopped because the FIFO is full during a card read, the software should read at least two FIFO locations to start the card clock.

The CIU block consists of the following primary functional blocks:

- Command path
- Data path
- SDIO interrupt control
- Clock control
- Mux/demux unit

1. Command Path

The command path performs the following functions:

- Loads clock parameters
- Loads card command parameters
- Sends commands to card bus (ccmd_out line)
- Receives responses from card bus (ccmd_in line)
- Sends responses to BIU
- Drives the P-bit on command line

A new command is issued to the SD/MMC Host Controller by programming the BIU registers and setting the start_cmd bit in the Command register. The BIU asserts start_cmd, which indicates that a new command is issued to the SD/MMC Host Controller. The command path loads this new command (command, command argument, timeout) and sends an acknowledge to the BIU by asserting cmd_taken.

Once the new command is loaded, the command path state machine sends a command to the SD_MMC bus—including the internally generated CRC7—and receives a response, if any. The state machine then sends the received response and signals to the BIU that the command is done, and then waits for eight clocks before loading a new command.

Load Command Parameters

One of the following commands or responses is loaded in the command path:

- New command from BIU – When start_cmd is asserted, then the start_cmd bit is set in the Command register.
- Internally-generated auto-stop command – When the data path ends, the stop command request is loaded.
- IRQ response with RCA 0x000 – When the command path is waiting for an IRQ response from the MMC card and a "send irq response" request is signaled by the BIU, then the send_irq_response bit is set in the control register.

Loading a new command from the BIU in the command path depends on the following Command register bit settings:

- update_clock_registers_only – If this bit is set in the Command register, the command path updates only the clock enable, clock divider, and clock source registers. If this bit is not set, the command path loads the command, command argument, and timeout registers; it then starts processing the new command.
- wait_prvdata_complete – If this bit is set, the command path loads the new command under one of the following conditions:
  - Immediately, if the data path is free (that is, there is no data transfer in progress), or if an open-ended data transfer is in progress (byte_count = 0).
  - After completion of the current data transfer, if a predefined data transfer is in progress.

Send Command and Receive Response

Once a new command is loaded in the command path – update_clock_registers_only bit is unset – the command path state machine sends out a command on the SD_MMC bus; the command path state machine is illustrated in Fig.17-3.

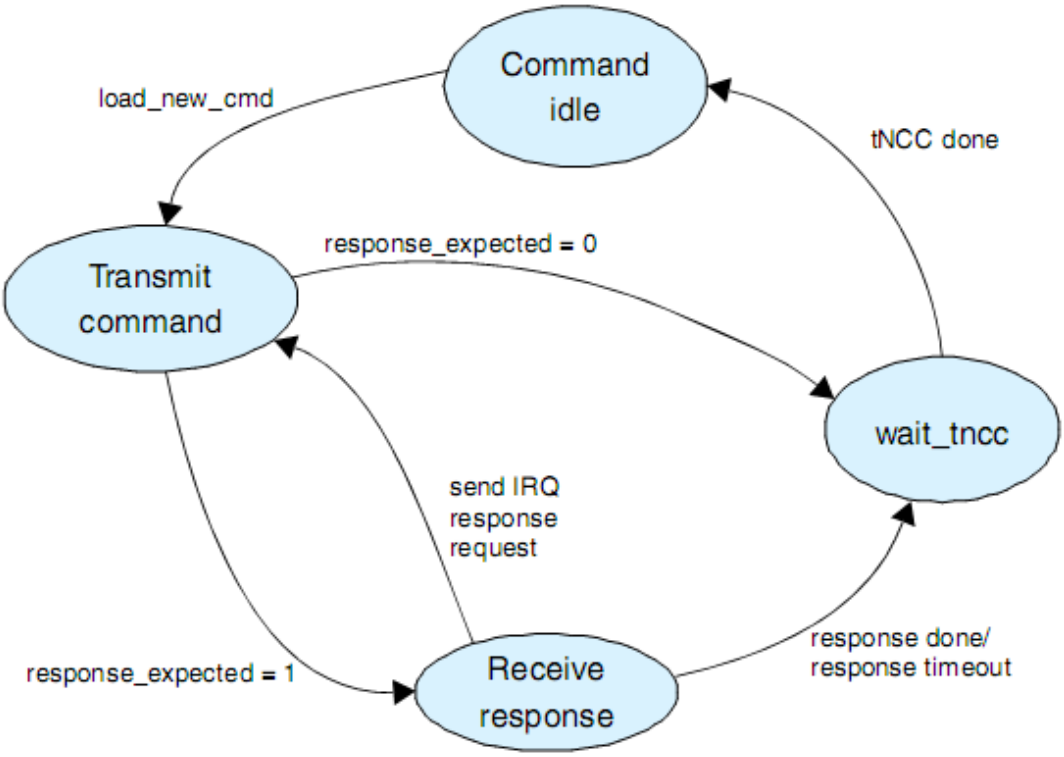


Fig. 17-3 SD/MMC Command Path State Machine

The command path state machine performs the following functions, according to Command register bit values:

A. send_initialization – Initialization sequence of 80 clocks is sent before sending the command.
B. response_expected – Response is expected for the command. After the command is sent out, the command path state machine receives a 48-bit or 136-bit response and sends it to the BIU. If the start bit of the card response is not received within the number of clocks programmed in the timeout register, then the response timeout and command done bit is set in the Raw Interrupt Status register as a signal to the BIU. If the response-expected bit is not set, the command path sends out a command and signals a response done to the BIU; that is, the command done bit is set in the Raw Interrupt Status register.
C. response_length – If this bit is set, a 136-bit response is received; if it is not set, a 48-bit response is received.
D. check_response_crc – If this bit is set, the command path compares CRC7 received in the response with the internally-generated CRC7. If the two do not match, the response CRC error is signaled to the BIU; that is, the response CRC error bit is set in the Raw Interrupt Status register.

Send Response to BIU

If the response_expected bit is set in the Command register, the received response is sent to the BIU. The Response0 register is updated for a short response, and the Response3, Response2, Response1, and Response0 registers are updated on a long response, after which the Command Done bit is set. If the response is for an auto_stop command sent by the CIU, the response is saved in the Response1 register, after which the Auto Command Done bit is set

Additionally, the command path checks for the following:

- Transmission bit = 0
- Command index matches command index of the sent command

- End bit = 1 in received card response

The command index is not checked for a 136-bit response or if the check_response_crc bit is unset. For a 136-bit response and reserved CRC 48-bit responses, the command index is reserved—that is, 111111.

Driving P-bit on CMD Line

The command path drives a P-bit = 1 on the CMD line between two commands if a response is not expected. If a response is expected, the P-bit is driven after the response is received and before the start of the next command; this is done by asserting both ccmd_out and ccmd_out_en. If the command expects the Command Completion Signal, then the P-bit is driven only after receiving the Command Completion Signal.During initialization, the software should set the ccmd_od_pullup_en bit, which indicates an open-drain mode, during which the controller drives only a 0 or high-impedance (Z) on the command bus; a hard 1 is never driven in open-drain mode.

2. Data Path

The data path block pops the data FIFO and transmits data on cdata_out during a write data transfer, or it receives data on cdata_in and pushes it into the FIFO during a read data transfer. The data path loads new data parameters—that is, data expected, read/write data transfer, stream/block transfer, block size, byte count, card type, timeout registers—whenever a data transfer command is not in progress.

If the data_expected bit is set in the Command register, the new command is a data transfer command and the data path starts one of the following:

- Transmit data if the read/write bit = 1
- Data receive if read/write bit = 0

Data Transmit

The data transmit state machine, illustrated in Fig.17-4, starts data transmission two clocks after a response for the data write command is received; this occurs even if the command path detects a response error or response CRC error. If a response is not received from the card because of a response timeout, data is not transmitted. Depending upon the value of the transfer_mode bit in the Command register, the data transmit state machine puts data on the card data bus in a stream or in block(s).

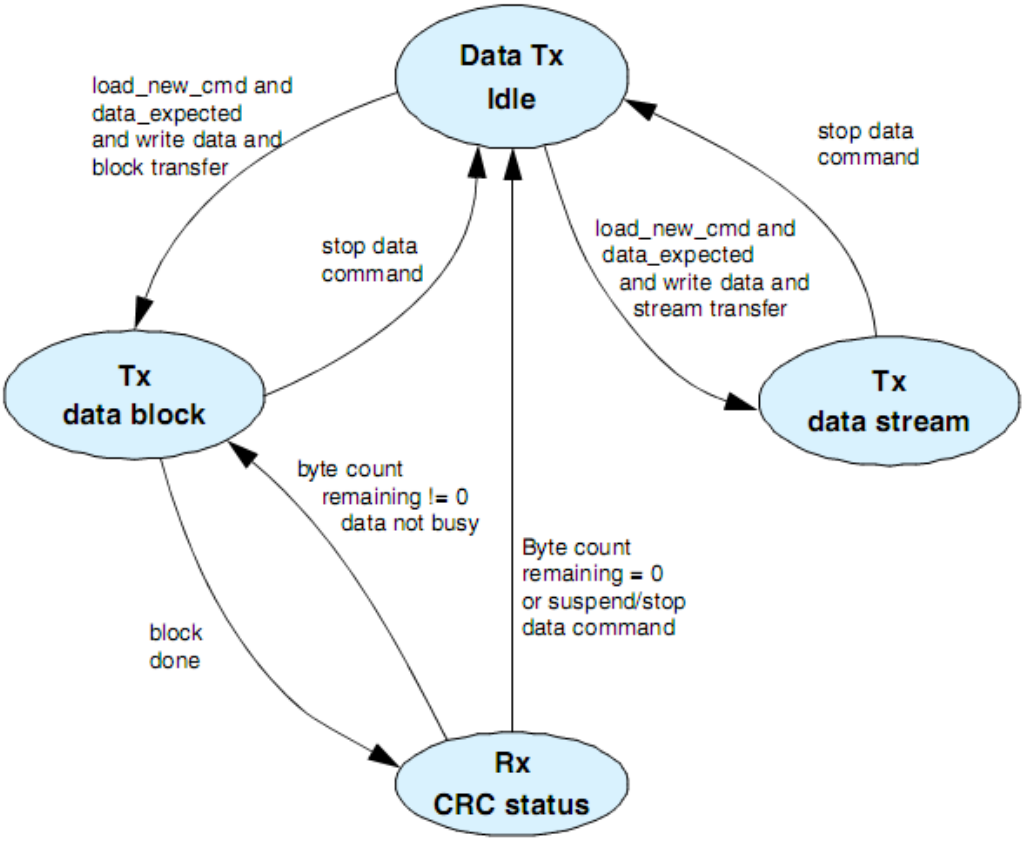


Fig. 17-4 SD/MMC Data Transmit State Machine

Stream Data Transmit

If the transfer_mode bit in the Command register is set to 1, it is a stream-write data transfer. The data path pops the FIFO from the BIU and transmits in a stream to the card data bus. If the FIFO becomes empty, the card clock is stopped and restarted once data is available in the FIFO.

If the byte_count register is programmed to 0, it is an open-ended stream-write data transfer. During this data transfer, the data path continuously transmits data in a stream until the host software issues a stop command. A stream data transfer is terminated when the end bit of the stop command and end bit of the data match over two clocks.

If the byte_count register is programmed with a non-zero value and the send_auto_stop bit is set in the Command register, the stop command is internally generated and loaded in the command path when the end bit of the stop command occurs after the last byte of the stream write transfer matches.

This data transfer can also terminate if the host issues a stop command before all the data bytes are transferred to the card bus.

Single Block Data

If the transfer_mode bit in the Command register is set to 0 and the byte_count register value is equal to the value of the block_size register, a single-block write-data transfer occurs. The data transmit state machine sends data in a single block, where the number of bytes equals the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set for a 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted for 1, 4, or 8 data lines, respectively.

After a single data block is transmitted, the data transmit state machine receives the CRC status from the card and signals a data transfer to the BIU; this happens when the data-transfer-over bit is set in the RINTSTS register.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register.

Additionally, if the start bit of the CRC status is not received by two clocks after the end of the data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the RINTSTS register.

Multiple Block Data

A multiple-block write-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value in the byte_count register is not equal to the value of the block_size register. The data transmit state machine sends data in blocks, where the number of bytes in a block equals the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to 1-bit, 4-bit, or 8-bit data transfer, the data is transmitted on 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and transmitted on 1, 4, or 8 data lines, respectively.

After one data block is transmitted, the data transmit state machine receives the CRC status from the card. If the remaining byte_count becomes 0, the data path signals to the BIU that the data transfer is done; this happens when the data-transfer-over bit is set in the RINTSTS register.

If the remaining data bytes are greater than 0, the data path state machine starts to transmit another data block.

If a negative CRC status is received from the card, the data path signals a data CRC error to the BIU by setting the data CRC error bit in the RINTSTS register, and continues further data transmission until all the bytes are transmitted.

Additionally, if the CRC status start bit is not received by two clocks after the end of a data block, a CRC status start bit error is signaled to the BIU by setting the write-no-CRC bit in the RINTSTS register; further data transfer is terminated.

If the send_auto_stop bit is set in the Command register, the stop command is internally generated during the transfer of the last data block, where no extra bytes are transferred to the card. The end bit of the stop command may not exactly match the end bit of the CRC status in the last data block.

If the block size is less than 4, 16, or 32 for card data widths of 1 bit, 4 bits, or 8 bits, respectively, the data transmit state machine terminates the data transfer when all the data is transferred, at which time the internally generated stop command is loaded in the command path.

If the byte_count is 0 – the block size must be greater than 0 – it is an open-ended block transfer. The data transmit state machine for this type of data transfer continues the block-write data transfer until the host software issues a stop or abort command.

Data Receive

The data-receive state machine, illustrated in Fig.17-5, receives data two clock cycles after the end bit of a data read command, even if the command path detects a response error or response CRC error. If a response is not received from the card because a response timeout occurs, the BIU does not receive a signal that the data transfer is complete; this happens if the command sent by the SD/MMC Host Controller is an illegal operation for the card, which keeps the card from starting a read data transfer.

If data is not received before the data timeout, the data path signals a data

timeout to the BIU and an end to the data transfer done. Based on the value of the transfer_mode bit in the Command register, the data-receive state machine gets data from the card data bus in a stream or block(s).

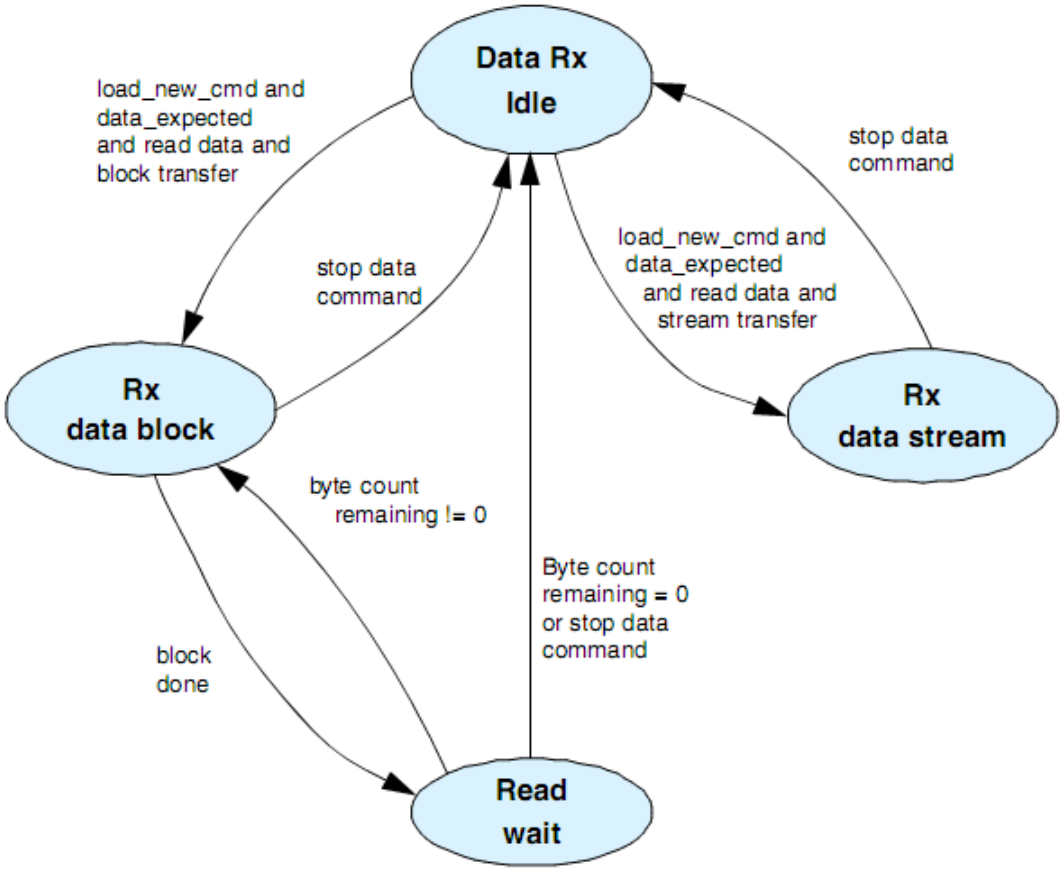


Fig. 17-5 SD/MMC Data Receive State Machine

Stream Data Read

A stream-read data transfer occurs if the transfer_mode bit in the Command register equals 1, at which time the data path receives data from the card and pushes it to the FIFO. If the FIFO becomes full, the card clock stops and restarts once the FIFO is no longer full.

An open-ended stream-read data transfer occurs if the byte_count register equals 0. During this type of data transfer, the data path continuously receives data in a stream until the host software issues a stop command. A stream data transfer terminates two clock cycles after the end bit of the stop command.

If the byte_count register contains a non-zero value and the send_auto_stop bit is set in the Command register, a stop command is internally generated and loaded into the command path, where the end bit of the stop command occurs after the last byte of the stream data transfer is received. This data transfer can terminate if the host issues a stop or abort command before all the data bytes are received from the card.

Single-Block Data Read

A single-block read-data transfer occurs if the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is equal to the value of the block_size register. When a start bit is received before the data times out, data bytes equal to the block size and CRC16 are received and checked with the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively. If there is a CRC16 mismatch, the data path signals a data CRC error to the BIU. If the received end bit is not 1, the BIU receives an end-bit error.

Multiple-Block Data Read

If the transfer_mode bit in the Command register is set to 0 and the value of the byte_count register is not equal to the value of the block_size register, it is a multiple-block read-data transfer. The data-receive state machine receives data in blocks, where the number of bytes in a block is equal to the block size, including the internally-generated CRC16.

If the CTYPE register bit for the selected card – indicated by the card_num value in the Command register – is set to a 1-bit, 4-bit, or 8-bit data transfer, data is received from 1, 4, or 8 data lines, respectively, and CRC16 is separately generated and checked for 1, 4, or 8 data lines, respectively.

After a data block is received, if the remaining byte_count becomes 0, the data path signals a data transfer to the BIU.

If the remaining data bytes are greater than 0, the data path state machine causes another data block to be received. If CRC16 of a received data block does not match the internally-generated CRC16, a data CRC error to the BIU and data reception continue further data transmission until all bytes are transmitted.

Additionally, if the end of a received data block is not 1, data on the data path signals terminate the bit error to the CIU and the data-receive state machine terminates data reception, waits for data timeout, and signals to the BIU that the data transfer is complete.

If the send_auto_stop bit is set in the Command register, the stop command is internally generated when the last data block is transferred, where no extra bytes are transferred from the card; the end bit of the stop command may not exactly match the end bit of the last data block.

If the requested block size for data transfers to cards is less than 4, 16, or 32 bytes for 1-bit, 4-bit, or 8-bit data transfer modes, respectively, the data-transmit state machine terminates the data transfer when all data is transferred, at which point the internally-generated stop command is loaded in the command path. Data received from the card after that are then ignored by the data path.

If the byte_count is 0—the block size must be greater than 0—it is an open-ended block transfer. For this type of data transfer, the data-receive state machine continues the block-read data transfer until the host software issues a stop or abort command.

Auto-Stop

The SD/MMC Host Controller internally generates a stop command and is loaded in the command path when the send_auto_stop bit is set in the Command register. The auto-stop command helps to send an exact number of data bytes using a stream read or write for the MMC, and a multiple-block read or write for SD memory transfer for SD cards

The software should set the send_auto_stop bit according to details listed in Table 17-2.

Table 17-2 Auto-Stop Generation

Card	Transfer type	Byte	send_auto_stop	Comments

type		Count	bit set	
MMC	Stream read	0	No	Open-ended stradm
MMC	Stream read	>0	Yes	Auto-stop after all bytes transfer
MMC	Stream write	0	No	Open-ended stradm
MMC	Stream write	>0	Yes	Auto-stop after all bytes transfer
MMC	Single-block read	>0	No	Byte count =0 is illegal
MMC	Single-block write	>0	No	Byte count =0 is illegal
MMC	Multiple-block read	0	No	Open-ended multiple block
MMC	Multiple-block read	>0	Yes①	Pre-defined multiple block
MMC	Multiple-block write	0	No	Open-ended multiple block
MMC	Multiple-block write	>0	Yes①	Pre-defined multiple block
SDMEM	Single-block read	>0	No	Byte count =0 is illegal
SDMEM	Single-block write	>0	No	Byte count =0 illegal
SDMEM	Multiple-block read	0	No	Open-ended multiple block
SDMEM	Multiple-block read	>0	Yes	Auto-stop after all bytes transfer
SDMEM	Multiple-block write	0	No	Open-ended multiple block
SDMEM	Multiple-block write	>0	Yes	Auto-stop after all bytes transfer
SDIO	Single-block read	>0	No	Byte count =0 is illegal
SDIO	Single-block write	>0	No	Byte count =0 illegal
SDIO	Multiple-block read	0	No	Open-ended multiple block
SDIO	Multiple-block read	>0	No	Pre-defined multiple block
SDIO	Multiple-block write	0	No	Open-ended multiple block
SDIO	Multiple-block write	>0	No	Pre-defined multiple block

①:*The condition under which the transfer mode is set to block transfer and byte_count is equal to block size is treated as a single-block data transfer command for both MMC and SD cards. If byte_count = n*block_size (n = 2,3, …), the condition is treated as a predefined multiple-block data transfer command. In the case of an MMC card, the host software can perform a predefined data transfer in two ways: 1) Issue the CMD23 command before issuing CMD18/CMD25 commands to the card – in this case, issue MD18/CMD25 commands without setting the send_auto_stop bit. 2) Issue CMD18/CMD25 commands without issuing CMD23 command to the card, with the send_auto_stop bit set. In this case, the multiple-block data transfer is terminated by an internally-generated auto-stop command after the programmed byte count.*

The following list conditions for the auto-stop command.

- Stream read for MMC card with byte count greater than 0 – The SD/MMC Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent out when the last byte of data is read from the card and no extra data byte is received. If the byte count is less than 6 (48 bits), a few extra data bytes are received from the card before the end bit of the stop command is sent.
- Stream write for MMC card with byte count greater than 0 - The SD/MMC Host Controller generates an internal stop command and loads it into the command path so that the end bit of the stop command is sent when the last byte of data is transmitted on the card bus and no extra data byte is transmitted. If the byte count is less than 6 (48 bits), the data path transmits the data last in order to meet the above condition.
- Multiple-block read memory for SD card with byte count greater than 0 – If the block size is less than 4 (single-bit data bus), 16 (4-bit data bus), or 32

(8-bit data bus), the auto-stop command is loaded in the command path after all the bytes are read. Otherwise, the top command is loaded in the command path so that the end bit of the stop command is sent after the last data block is received.

- Multiple-block write memory for SD card with byte count greater than 0 – If the block size is less than 3 (single-bit data bus), 12 (4-bit data bus), or 24 (8-bit data bus), the auto-stop command is loaded in the command path after all data blocks are transmitted. Otherwise, the stop command is loaded in the command path so that the end bit of the stop command is sent after the end bit of the CRC status is received.
- Precaution for host software during auto-stop – Whenever an auto-stop command is issued, the host software should not issue a new command to the SD/MMC Host Controller until the auto-stop is sent by the SD/MMC Host Controller and the data transfer is complete. If the host issues a new command during a data transfer with the auto-stop in progress, an auto-stop command may be sent after the new command is sent and its response is received; this can delay sending the stop command, which transfers extra data bytes. For a stream write, extra data bytes are erroneous data that can corrupt the card data. If the host wants to terminate the data transfer before the data transfer is complete, it can issue a stop or abort command, in which case the SD/MMC Host Controller does not generate an auto-stop command.

3. Non-Data Transfer Commands that Use Data Path

Some non-data transfer commands (non-read/write commands) also use the data path. Table 17-3 lists the commands and register programming requirements for them.

Table 17-3 Non-data Transfer Commands and Requirements

	CMD27	CMD30	CMD42	AACMD13	ACMD22	ACMD51
Command register programming						
Cmd_index	6'h1B	6'h1E	6'h2A	6'h0D	6'h16	6'h33
Response_expect	1	1	1	1	1	1
Response_length	0	0	0	0	0	0
Check_response_crc	1	1	1	1	1	1
Data_expected	1	1	1	1	1	1
Read/write	1	0	1	0	0	0
Transfer_mode	0	0	0	0	0	0
Send_auto_stop	0	0	0	0	0	0
Wait_prevdata_complete	0	0	0	0	0	0
Stop_abort_cmd	0	0	0	0	0	0
Command Argument register programming						
	Stuff bits	32-bit write protect data address	Stuff bits	Stuff bits	Stuff bits	Stuff bits

Block Size register programming						
	16	4	Num_bytes①	64	4	8
Byte Count register programming						
	16	4	Num_bytes①	64	4	8

①: *Num_bytes = No. of bytes specified as per the lock card data structure (Refer to the SD specification and the MMC specification)*

4. SDIO Interrupt Control

Interrupts for SD cards are reported to the BIU by asserting an interrupt signal for two clock cycles. SDIO cards signal an interrupt by asserting cdata_in low during the interrupt period; an interrupt period for the selected card is determined by the interrupt control state machine. An interrupt period is always valid for non-active or non-selected cards, and 1-bit data mode for the selected card. An interrupt period for a wide-bus active or selected card is valid for the following conditions:

- Card is idle
- Non-data transfer command in progress
- Third clock after end bit of data block between two data blocks
- From two clocks after end bit of last data until end bit of next data transfer command

Bear in mind that, in the following situations, the SD/MMC Host Controller does not sample the SDIO interrupt of the selected card when the card data width is 4 bits. Since the SDIO interrupt is level-triggered, it is sampled in a further interrupt period and the host does not lose any SDIO interrupt from the card.

A. Read/Write Resume – The CIU treats the resume command as a normal data transfer command. SDIO interrupts during the resume command are handled similarly to other data commands.
   According to the SDIO specification, for the normal data command the interrupt period ends after the command end bit of the data command; for the resume command, it ends after the response end bit. In the case of the resume command, the SD/MMC Host Controller stops the interrupt sampling period after the resume command end bit, instead of stopping after the response end bit of the resume command.
B. Suspend during read transfer – If the read data transfer is suspended by the host, the host sets the abort_read_data bit in the SD/MMC Host Controller to reset the data state machine. In the CIU, the SDIO interrupts are handled such that the interrupt sampling starts after the abort_read_data bit is set by the host. In this case the SD/MMC Host Controller does not sample SDIO interrupts between the period from response of the suspend command to setting the abort_read_data bit, and starts sampling after setting the abort_read_data bit.

5. Clock Control

The clock control block provides different clock frequencies required for SD_MMC cards. The cclk_in signal is the source clock (cclk_in >= card max operating frequency) for clock divider of the clock control block. This source clock (cclk_in) is used to generate different card clock frequencies

(sdmmc_clkout). The card clock can have different clock frequencies, since the SD card can be a low-speed SD card or a full-speed SD card. The SD/MMC Host Controller provides one clock signal (sdmmc_clkout).

The clock frequency of a card depends on the following clock control registers:

- Clock Divider register – Internal clock dividers are used to generate different clock frequencies required for card. The division factor for each clock divider can be programmed by writing to the Clock Divider register. The clock divider is an 8-bit value that provides a clock division factor from 1 to 510; a value of 0 represents a clock-divider bypass, a value of 1 represents a divide by 2, a value of 2 represents a divide by 4, and so on.
- Clock Control register – sdmmc_clkout can be enabled or disabled for each card under the following conditions:
  - clk_enable – sdmmc_clkout for a card is enabled if the clk_enable bit for a card in the Clock Control register is programmed (set to 1) or disabled (set to 0).
  - Low-power mode – Low-power mode of a card can be enabled by setting the low-power mode bit of the Clock Control register to 1. If low-power mode is enabled to save card power, the sdmmc_clkout is disabled when the card is idle for at least 8 card clock cycles. It is enabled when a new command is loaded and the command path goes to a non-idle state.

Additionally, sdmmc_clkout is disabled when an internal FIFO is full – card read (no more data can be received from card) – or when the FIFO is empty – card write (no data is available for transmission). This helps to avoid FIFO overrun and underrun conditions.It is used by the command and data path to qualify cclk_in for driving outputs and sampling inputs at the programmed clock frequency for the selected card, according to the Clock Divider and Clock Source register values.

Under the following conditions, the card clock is stopped or disabled, along with the active clk_en, for the selected card:

- Clock can be disabled by writing to Clock Enable register (clk_en bit = 1).
- If low-power mode is selected and card is idle, or not selected for 8 clocks.
- FIFO is full and data path cannot accept more data from the card and data transfer is incomplete –to avoid FIFO overrun.
- FIFO is empty and data path cannot transmit more data to the card and data transfer is incomplete – to avoid FIFO underrun.

6. Error Detection

- Response
  - Response timeout – Response expected with response start bit is not received within programmed number of clocks in timeout register.
  - Response CRC error – Response is expected and check response CRC requested; response CRC7 does not match with the internally-generated CRC7.
  - Response error – Response transmission bit is not 0, command index does not match with the command index of the send command, or response end bit is not 1.
- Data transmit
  - No CRC status – During a write data transfer, if the CRC status start bit is not received two clocks after the end bit of the data block is sent out, the data path does the following:
    - Signals no CRC status error to the BIU
    - Terminates further data transfer

- Signals data transfer done to the BIU
- Negative CRC – If the CRC status received after the write data block is negative (that is, not 010), a data CRC error is signaled to the BIU and further data transfer is continued.
- Data starvation due to empty FIFO – If the FIFO becomes empty during a write data transmission, or if the card clock is stopped and the FIFO remains empty for data timeout clocks, then a data-starvation error is signaled to the BIU and the data path continues to wait for data in the FIFO.

- Data receive
  - Data timeout – During a read-data transfer, if the data start bit is not received before the number of clocks that were programmed in the timeout register, the data path does the following:
    - Signals data-timeout error to the BIU
    - Terminates further data transfer
    - Signals data transfer done to BIU
  - Data start bit error – During a 4-bit or 8-bit read-data transfer, if the all-bit data line does not have a start bit, the data path signals a data start bit error to the BIU and waits for a data timeout, after which it signals that the data transfer is done.
  - Data CRC error – During a read-data-block transfer, if the CRC16 received does not match with the internally generated CRC16, the data path signals a data CRC error to the BIU and continues further data transfer.
  - Data end-bit error – During a read-data transfer, if the end bit of the received data is not 1, the data path signals an end-bit error to the BIU, terminates further data transfer, and signals to the BIU that the data transfer is done.
  - Data starvation due to FIFO full – During a read data transmission and when the FIFO becomes full, the card clock is stopped. If the FIFO remains full for data timeout clocks, a data starvation error is signaled to the BIU (Data Starvation by Host Timeout bit is set in RINTSTS Register) and the data path continues to wait for the FIFO to start to empty.

17.4 Register description

17.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SDMMC_CTRL	0x0000	W	0x00000000	Control register
SDMMC_PWREN	0x0004	W	0x00000000	Power-enable register
SDMMC_CLKDIV	0x0008	W	0x00000000	Clock-divider register
SDMMC_CLKENA	0x0010	W	0x00000000	Clock-enable register
SDMMC_TMOUT	0x0014	W	0xffffff40	Time-out register
SDMMC_CTYPE	0x0018	W	0x00000000	Card-type register
SDMMC_BLKSIZ	0x001c	W	0x00000200	Block-size register
SDMMC_BYTCNT	0x0020	W	0x00000200	Byte-count register
SDMMC_INTMASK	0x0024	W	0x00000000	Interrupt-mask register
SDMMC_CMDARG	0x0028	W	0x00000000	Command-argument register

Name	Offset	Size	Reset Value	Description
SDMMC_CMD	0x002c	W	0x00000000	Command register
SDMMC_RESP0	0x0030	W	0x00000000	Response-0 register
SDMMC_RESP1	0x0034	W	0x00000000	Response-1 register
SDMMC_RESP2	0x0038	W	0x00000000	Response-2 register
SDMMC_RESP3	0x003c	W	0x00000000	Response-3 register
SDMMC_MINTSTS	0x0040	W	0x00000000	Masked interrupt-status registe
SDMMC_RINTSTS	0x0044	W	0x00000000	Raw interrupt-status register
SDMMC_STATUS	0x0048	W	0x00000406	Status register
SDMMC_FIFOTH	0x004c	W	0x00000000	FIFO threshold register
SDMMC_CDETECT	0x0050	W	0x00000000	Card-detect register
SDMMC_WRTPRT	0x0054	W	0x00000000	Write-protect register
SDMMC_TCBCNT	0x005c	W	0x00000000	Transferred CIU card byte count
SDMMC_TBBCNT	0x0060	W	0x00000000	Transferred host/DMA to/from BIU-FIFO byte count
SDMMC_DEBNCE	0x0064	W	0x00ffffff	Card detect debounce register
SDMMC_USRID	0x0068	W	0x00000000	User ID register
SDMMC_VERID	0x006c	W	0x5342240a	Version ID register
SDMMC_UHS_REG	0x0074	W	0x00000000	UHS-1 register
SDMMC_RST_n	0x0078	W	0x00000001	Hardware reset register
SDMMC_CARDTHRCTL	0x0100	W	0x00000000	Card Read Threshold Enable
SDMMC_BACK_END_POWER	0x0104	W	0x00000000	Back-end Power
SDMMC_FIFO_BASE	0x0200	W	0x00000000	

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

17.4.2 Detail Register Description

SDMMC_CTRL

Address: Operational Base + offset (0x0000)

Control register

Bit	Attr	Reset Value	Description
31:9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	abort_read_data 0 –No change 1 –After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state machine resets to idle. Used in SDIO card suspend sequence.
7	RW	0x0	send_irq_response 0 –No change 1 –Send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40, and SDMMC Controller waits for interrupt response from MMC card(s). In meantime, if host wants SDMMC Controller to exit waiting for interrupt state, it can set this bit, at which time SDMMC Controller command state-machine sends CMD40 response on bus and returns to idle state.
6	RW	0x0	read_wait 0 –Clear read wait 1 –Assert read wait For sending read-wait to SDIO cards
5	RW	0x0	dma_enable 0 –Disable DMA transfer mode 1 –Enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside SDMMC Controller to prioritize simultaneous host/DMA access.
4	RW	0x0	int_enable Global interrupt enable/disable bit: 0 –Disable interrupts 1 –Enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	W1C	0x0	dma_reset 0 –No change 1 –Reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared after two AHB clocks.
1	W1C	0x0	fifo_reset 0 –No change 1 –Reset to data FIFO To reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation
0	W1C	0x0	controller_reset 0 –No change 1 –Reset SDMMC controller To reset controller, firmware should set bit to 1. This bit is auto-cleared after two AHB and two cclk_in clock cycles. This resets: * BIU/CIU interface * CIU and state machines * abort_read_data, send_irq_response, and read_wait bits of Control register * start_cmd bit of Command register Does not affect any registers or DMA interface, or FIFO or host interrupts

SDMMC_PWREN
Address: Operational Base + offset (0x0004)
Power-enable register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	power_enable Power on/off switch for the card. Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card. 0 –power off 1 –power on Bit values output to card_power_en port.

SDMMC_CLKDIV
Address: Operational Base + offset (0x0008)
Clock-divider register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	clk_divider0 Clock divider-0 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, value of "ff昬on.In MMC-Ver3.3-only mode, bits not implemented because only one clock divider is supported.

SDMMC_CLKENA
Address: Operational Base + offset (0x0010)
Clock-enable register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	cclk_low_power Low-power control for SD card clock and MMC card clock supported. 0 –Non-low-power mode 1 –Low-power mode; stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO cards, if interrupts must be detected, clock should not be stopped).
15:1	RO	0x0	reserved
0	RW	0x0	cclk_enable Clock-enable control for SD card clock and MMC card clock supported. 0 –Clock disabled 1 –Clock enabled

SDMMC_TMOUT
Address: Operational Base + offset (0x0014)
Time-out register

Bit	Attr	Reset Value	Description
31:8	RW	0xffffff	data_timeout Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. Value is in number of card output clocks –cclk_out of selected card.
7:0	RW	0x40	response_timeout Response timeout value. Value is in number of card output clocks –cclk_out.

SDMMC_CTYPE
Address: Operational Base + offset (0x0018)
Card-type register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	card_width_8 Indicates if card is 8-bit: 0 –Non 8-bit mode 1 –8-bit mode
15:1	RO	0x0	reserved
0	RW	0x0	card_width Indicates if card is 1-bit or 4-bit: 0 –1-bit mode 1 –4-bit mode

SDMMC_BLKSIZ
Address: Operational Base + offset (0x001c)
Block-size register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0200	block_size Block size

SDMMC_BYTCNT
Address: Operational Base + offset (0x0020)
Byte-count register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000200	byte_count Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.

SDMMC_INTMASK
Address: Operational Base + offset (0x0024)
Interrupt-mask register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RW	0x0	sdio_int_mask Mask SDIO interrupts When masked, SDIO interrupt detection for that card is disabled. A 0 masks an interrupt, and 1 enables an interrupt.
23:17	RO	0x0	reserved
16	RW	0x0	new_int_mask New Interrupt Mask 1: data no busy interrupt masked
15:0	RW	0x0000	int_mask Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. bit 15 –End-bit error (read)/Write no CRC (EBE) bit 14 –Auto command done (ACD) bit 13 –Start-bit error (SBE) bit 12 –Hardware locked write error (HLE) bit 11 –FIFO underrun/overrun error (FRUN) bit 10 –Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9 –Data read timeout (DRTO) bit 8 –Response timeout (RTO) bit 7 –Data CRC error (DCRC) bit 6 –Response CRC error (RCRC) bit 5 –Receive FIFO data request (RXDR) bit 4 –Transmit FIFO data request (TXDR) bit 3 –Data transfer over (DTO) bit 2 –Command done (CD) bit 1 –Response error (RE) bit 0 –Card detect (CD)

SDMMC_CMDARG
Address: Operational Base + offset (0x0028)
Command-argument register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cmd_arg Value indicates command argument to be passed to card.

SDMMC_CMD
Address: Operational Base + offset (0x002c)
Command register

Bit	Attr	Reset Value	Description
31	RW	0x0	start_cmd Start command. Once command is taken by CIU, bit is cleared. When bit is set, host should not attempt to write to any command registers. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC cards, Command Done bit is set in raw interrupt register.
30	RO	0x0	reserved
29	RW	0x0	use_hold_reg Use Hold Register 0 - CMD and DATA sent to card bypassing HOLD Register 1 - CMD and DATA sent to card through the HOLD Register Note: a. Set to 1'b1 for SDR12 and SDR25 (with non-zero phase-shifted cclk_in_drv); zero phase shift is not allowed in these modes. b. Set to 1'b0 for SDR50, and DDR50 (with zero phase- shifted cclk_in_drv) c. Set to 1'b1 for SDR50, and DDR50 (with non-zero phase-shifted cclk_in_drv)
28	RW	0x0	volt_switch Voltage switch bit 0 - No voltage switching 1 - Voltage switching enabled; must be set for CMD11 only
27	RW	0x0	boot_mode Boot Mode 0 - Mandatory Boot operation 1 - Alternate Boot operation
26	RW	0x0	disable_boot Disable Boot. When software sets this bit along with start_cmd, CIU terminates the boot operation. Do NOT set disable_boot and enable_boot together.

Bit	Attr	Reset Value	Description
25	RW	0x0	expect_boot_ack Expect Boot Acknowledge. When Software sets this bit along with enable_boot, CIU expects a boot acknowledge start pattern of 0-1-0 from the selected card.
24	RW	0x0	enable_boot Enable Boot—this bit should be set only for mandatory boot mode.When Software sets this bit along with start_cmd, CIU starts the boot sequence for the corresponding card by asserting the CMD line low. Do NOT set disable_boot and enable_boot together.
23:22	RO	0x0	reserved
21	RW	0x0	update_clock_registers_only 0 –Normal command sequence 1 –Do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_registers_only = 0, following control registers are transferred from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card. When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC cards.
20:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15	RW	0x0	send_initialization 0 –Do not send initialization sequence (80 clocks of 1) before sending this command 1 –Send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card. This bit should not be set for either of the boot modes (alternate or mandatory).
14	RW	0x0	stop_abort_cmd 0 –Neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1 –Stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should be set so that command/data state-machines of CIU can return correctly to idle state. This is also applicable for Boot mode transfers. To Abort boot mode, this bit should be set along with CMD[26] = disable_boot.
13	RW	0x0	wait_prvdata_complete 0 –Send command at once, even if previous data transfer has not completed 1 –Wait for previous data transfer completion before sending command The wait_prvdata_complete = 0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.

Bit	Attr	Reset Value	Description
12	RW	0x0	send_auto_stop 0 –No stop command sent at end of data transfer 1 –Send stop command at end of data transfer When set, SDMMC Controller sends stop command to SD_MMC cards at end of data transfer. * when send_auto_stop bit should be set, since some data transfers do not need explicit stop commands * open-ended transfers that software should explicitly send to stop command Additionally, when "resume"is sent to resume –suspended memory access of SD-Combo card –bit should be set correctly if suspended data transfer needs send_auto_stop. Don't care if no data expected from card.
11	RW	0x0	transfer_mode 0 –Block data transfer command 1 –Stream data transfer command Don't care if no data expected.
10	RW	0x0	wr 0 – Read from card 1 – Write to card Don't care if no data expected from card.
9	RW	0x0	data_expected 0 –No data transfer expected (read/write) 1 –Data transfer expected (read/write)
8	RW	0x0	check_response_crc 0 –Do not check response CRC 1 –Check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller
7	RW	0x0	response_length 0 –Short response expected from card 1 –Long response expected from card
6	RW	0x0	response_expect 0 –No response expected from card 1 –Response expected from card
5:0	RW	0x00	cmd_index Command index

SDMMC_RESP0
Address: Operational Base + offset (0x0030)
Response-0 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response0 Bit[31:0] of response

SDMMC_RESP1
Address: Operational Base + offset (0x0034)
Response-1 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response Register represents bit[63:32] of long response. When CIU sends auto-stop command, then response is saved in register. Response for previous command sent by host is still preserved in Response 0 register. Additional auto-stop issued only for data transfer commands, and response type is always "short"for them.

SDMMC_RESP2
Address: Operational Base + offset (0x0038)
Response-2 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response2 Bit[95:64] of long response

SDMMC_RESP3
Address: Operational Base + offset (0x003c)
Response-3 register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	response3 Bit[127:96] of long response

SDMMC_MINTSTS
Address: Operational Base + offset (0x0040)
Masked interrupt-status registe

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RO	0x0	sdio_interrupt Interrupt from SDIO card; SDIO interrupt for card enabled only if corresponding sdio_int_mask bit is set in Interrupt mask register (mask bit 1 enables interrupt; 0 masks interrupt). 0 –No SDIO interrupt from card 1 –SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	new_int_status New Interrupt Status 0: data_no busy int
15:0	RO	0x0000	int_status Interrupt enabled only if corresponding bit in interrupt mask register is set. bit 15 –End-bit error (read)/write no CRC (EBE) bit 14 –Auto command done (ACD) bit 13 –Start-bit error (SBE) bit 12 –Hardware locked write error (HLE) bit 11 –FIFO underrun/overrun error (FRUN) bit 10 –Data starvation by host timeout (HTO)/Volt_switch_int bit 9 –Data read timeout (DRTO) bit 8 –Response timeout (RTO) bit 7 –Data CRC error (DCRC) bit 6 –Response CRC error (RCRC) bit 5 –Receive FIFO data request (RXDR) bit 4 –Transmit FIFO data request (TXDR) bit 3 –Data transfer over (DTO) bit 2 –Command done (CD) bit 1 –Response error (RE) bit 0 –Card detect (CD)

SDMMC_RINTSTS

Address: Operational Base + offset (0x0044)

Raw interrupt-status register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24	RO	0x0	sdio_interrupt Interrupt from SDIO card; Writes to these bits clear them. Value of 1 clears bit and 0 leaves bit intact. 0 –No SDIO interrupt from card 1 –SDIO interrupt from card
23:17	RO	0x0	reserved
16	RW	0x0	new_int_status New Interrupt Status 0: data no busy int
15:0	RO	0x0000	int_status Writes to bits clear status bit. Value of 1 clears status bit, and value of 0 leaves bit intact. Bits are logged regardless of interrupt mask status. bit 15 –End-bit error (read)/write no CRC (EBE) bit 14 –Auto command done (ACD) bit 13 –Start-bit error (SBE) bit 12 –Hardware locked write error (HLE) bit 11 –FIFO underrun/overrun error (FRUN) bit 10 –Data starvation-by-host timeout (HTO) /Volt_switch_int bit 9 –Data read timeout (DRTO)/Boot Data Start (BDS) bit 8 –Response timeout (RTO)/Boot Ack Received (BAR) bit 7 –Data CRC error (DCRC) bit 6 –Response CRC error (RCRC) bit 5 –Receive FIFO data request (RXDR) bit 4 –Transmit FIFO data request (TXDR) bit 3 –Data transfer over (DTO) bit 2 –Command done (CD) bit 1 –Response error (RE) bit 0 –Card detect (CD)

SDMMC_STATUS

Address: Operational Base + offset (0x0048)

Status register

Bit	Attr	Reset Value	Description
31	RO	0x0	dma_req DMA request signal state

Bit	Attr	Reset Value	Description
30	RO	0x0	dma_ack DMA acknowledge signal state
29:17	RO	0x0000	fifo_count FIFO count –Number of filled locations in FIFO
16:11	RO	0x00	response_index Index of previous response, including any auto-stop sent by core
10	RO	0x1	data_state_mc_busy Data transmit or receive state-machine is busy
9	RO	0x0	data_busy Inverted version of raw selected card_data[0] 0 –card data not busy 1 –card data busy default value is 1 or 0 depending on cdata_in
8	RO	0x0	data_3_status Raw selected card_data[3]; checks whether card is present 0 –card not present 1 –card present default value is 1 or 0 depending on cdata_in

Bit	Attr	Reset Value	Description
7:4	RO	0x0	command_fsm_states Command FSM states: 0 –Idle 1 –Send init sequence 2 –Tx cmd start bit 3 –Tx cmd tx bit 4 –Tx cmd index + arg 5 –Tx cmd crc7 6 –Tx cmd end bit 7 –Rx resp start bit 8 –Rx resp IRQ response 9 –Rx resp tx bit 10 –Rx resp cmd idx 11 –Rx resp data 12 –Rx resp crc7 13 –Rx resp end bit 14 –Cmd path wait NCC 15 –Wait; CMD-to-response turnaround NOTE: The command FSM state is represented using 19 bits. The STATUS Register(7:4) has 4 bits to represent the command FSM states. Using these 4 bits, only 16 states can be represented. Thus three states cannot be represented in the STATUS(7:4) register. The three states that are not represented in the STATUS Register(7:4) are: * Bit 16 –Wait for CCS * Bit 17 –Send CCSD * Bit 18 –Boot Mode Due to this, while command FSM is in "Wait for CCS state"or "Send CCSD"or "Boot Mode", the Status register indicates status as 0 for the bit field 7:4.
3	RO	0x0	fifo_full FIFO is full status
2	RO	0x1	fifo_empty FIFO is empty status
1	RO	0x1	fifo_tx_watermark FIFO reached Transmit watermark level; not qualified with data transfer

Bit	Attr	Reset Value	Description
0	RO	0x0	fifo_rx_watermark FIFO reached Receive watermark level; not qualified with data transfer

SDMMC_FIFOTH
Address: Operational Base + offset (0x004c)
FIFO threshold register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:28	RW	0x0	DMA_Mutiple_Transaction_Size Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE. 000 –1 transfers 001 –4 010 –8 011 –16 100 –32 101 –64 110 –128 111 –256 The units for transfers is the H_DATA_WIDTH parameter. A single transfer would be signalled based on this value. Value should be sub-multiple of (RX_WMark + 1)* (F_DATA_WIDTH/H_DATA_WIDTH) and (FIFO_DEPTH - TX_WMark)* (F_DATA_WIDTH/ H_DATA_WIDTH) For example, if FIFO_DEPTH = 16, FDATA_WIDTH == H_DATA_WIDTH Allowed combinations for MSize and TX_WMark are: MSize = 1, TX_WMARK = 1-15 MSize = 4, TX_WMark = 8 MSize = 4, TX_WMark = 4 MSize = 4, TX_WMark = 12 MSize = 8, TX_WMark = 8 MSize = 8, TX_WMark = 4 Allowed combinations for MSize and RX_WMark are: MSize = 1, RX_WMARK = 0-14 MSize = 4, RX_WMark = 3 MSize = 4, RX_WMark = 7 MSize = 4, RX_WMark = 11 MSize = 8, RX_WMark = 7 Recommended: MSize = 8, TX_WMark = 8, RX_WMark = 7

Bit	Attr	Reset Value	Description
27:16	RW	0x000	RX_WMark FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits –1 bit less than FIFO-count of status register, which is 13 bits. Limitation: RX_WMark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1) NOTE: In DMA mode during CCS time-out, the DMA does not generate the request at the end of packet, even if remaining bytes are less than threshold. In this case, there will be some data left in the FIFO. It is the responsibility of the application to reset the FIFO after the CCS timeout.
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x000	TX_WMark FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits –1 bit less than FIFO-count of status register, which is 13 bits. Limitation: TX_WMark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to FIFO_DEPTH/2)

SDMMC_CDETECT
Address: Operational Base + offset (0x0050)
Card-detect register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	card_detect_n Value on card_detect_n input ports; read-only bits. 0 represents presence of card.

SDMMC_WRTPRT
Address: Operational Base + offset (0x0054)
Write-protect register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	write_protect Value on card_write_prt input port. 1 represents write protection.

SDMMC_TCBCNT
Address: Operational Base + offset (0x005c)
Transferred CIU card byte count

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_card_byte_count Number of bytes transferred by CIU unit to card. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied. Both TCBCNT and TBBCNT share same coherency register. When AREA_OPTIMIZED parameter is 1, register should be read only after data transfer completes; during data transfer, register returns 0.

SDMMC_TBBCNT
Address: Operational Base + offset (0x0060)
Transferred host/DMA to/from BIU-FIFO byte count

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	trans_fifo_byte_count Number of bytes transferred between Host/DMA memory and BIU FIFO. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied. Both TCBCNT and TBBCNT share same coherency register.

SDMMC_DEBNCE
Address: Operational Base + offset (0x0064)
Card detect debounce register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0xffffff	debounce_count Number of host clocks (clk) used by debounce filter logic; typical debounce time is 5-25 ms.

SDMMC_USRID
Address: Operational Base + offset (0x0068)
User ID register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	USRID User identification register; value set by user. Default reset value can be picked by user while configuring core before synthesis. Can also be used as scratch pad register by user. the default value is determined by Configuration Value.

SDMMC_VERID
Address: Operational Base + offset (0x006c)
Version ID register

Bit	Attr	Reset Value	Description
31:0	RO	0x5342240a	VERID Version identification register; register value is hard-wired. Can be read by firmware to support different versions of core.

SDMMC_UHS_REG
Address: Operational Base + offset (0x0074)
UHS-1 register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	DDR_REG DDR mode. Determines the voltage fed to the buffers by an external voltage regulator. 0 –Non-DDR mode 1 –DDR mode UHS_REG [16] should be set for card

Bit	Attr	Reset Value	Description
15:1	RO	0x0	reserved
0	RW	0x0	VOLT_REG High Voltage mode. Determines the voltage fed to the buffers by an external voltage regulator. 0 –Buffers supplied with 3.3V Vdd 1 –Buffers supplied with 1.8V Vdd These bits function as the output of the host controller and are fed to an external voltage regulator. The voltage regulator must switch the voltage of the buffers of a particular card to either 3.3V or 1.8V, depending on the value programmed in the register. VOLT_REG[0] should be set to 1'b1 for card in order to make it operate for 1.8V.

SDMMC_RST_n
Address: Operational Base + offset (0x0078)
Hardware reset register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	CARD_RESET Hardware reset. 1 –Active mode 0 –Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized. CARD_RESET[0] should be set to 1'b1 to reset card

SDMMC_CARDTHRCTL
Address: Operational Base + offset (0x0100)
Card Read Threshold Enable

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	CardRdThreshold Card Read Threshold size
15:1	RO	0x0	reserved

Bit	Attr	Reset Value	Description
0	RW	0x0	CardRdThrEn Card Read Threshold Enable 1'b0 - Card Read Threshold disabled 1'b1 - Card Read Threshold enabled. Host Controller initiates Read Transfer only if CardRdThreshold amount of space is available in receive FIFO.

SDMMC_BACK_END_POWER
Address: Operational Base + offset (0x0104)
Back-end Power

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	Back_End_Power Back end power 1'b0 –Off; Reset 1'b1 –Back-end Power supplied to card application

SDMMC_FIFO_BASE
Address: Operational Base + offset (0x0200)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	fifo_base_addr fifo base addr

17.5 Timing Diagram

17.6 Interface description

17.6.1 Card-Detect and Write-Protect Mechanism

Figure 17-6 illustrates how the SD/MMC Host Controller card detection and write-protect signals are connected. Most of the SD_MMC sockets have card-detect pins. When no card is present, card_detect_n is 1 due to the pull-up. When the SD_MMC card is inserted, the card-detect pin is shorted to ground, which makes card_detect_n go to 0. Similarly in SD cards, when the write-protect switch is toward the left, it shorts the write_protect port to ground.

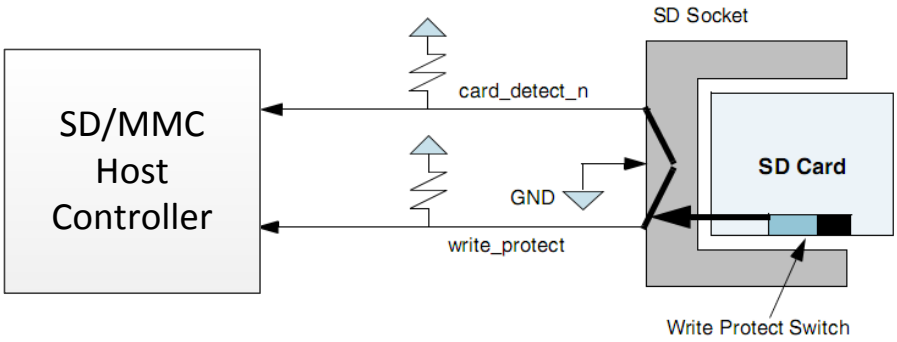


Fig. 17-6 Card-Detect and Write-Protect

17.6.2 SD/MMC Controller Termination Requirement

Fig.17-7 illustrates the SD/MMC Host Controller termination requirements, which is required to pull up ccmd and cdata lines on the SD_MMC bus. The recommended specification for pull-up on the ccmd line (Rcmd) is 4.7K - 100K for MMC, and 10K - 100K for an SD. The recommended pull-up on the cdata line (Rdat) is 50K - 100K.

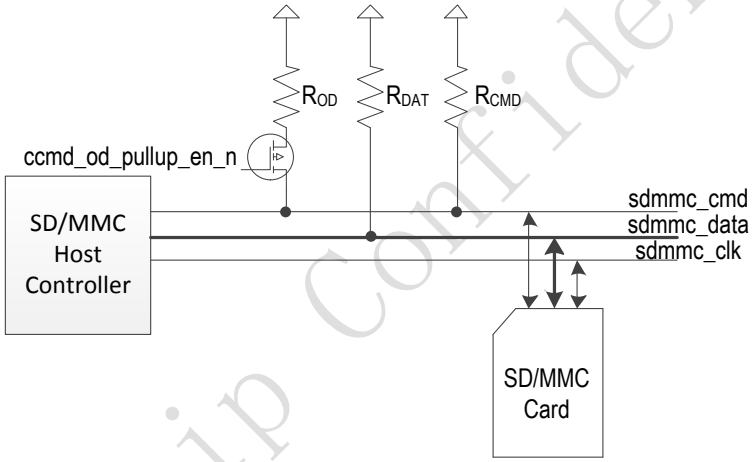


Fig. 17-7 SD/MMC Termination

Rcmd and Rod Calculation

The SD and MMC card enumeration happens at a very low frequency – 100-400KHz. Since the MMC bus is a shared bus between multiple cards, during enumeration open-drive mode is used to avoid bus conflict. Cards that drive 0 win over cards that drive "z." The pull-up in the command line pulls the bus to 1 when all cards drive "z." MMC interrupt mode also uses the pull-up. During normal data transfer, the hose chooses only one card and the card driver switches to push-pull mode.

For example, if enumeration is done at 400KHz and the total bus capacitance is 200 pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz
R = 1/(2.2 * C * 100KHz)
= 1/(2.2 x 200 x 10**-12 x 400 x 10**3)
= 1/(17.6 x 10**-5)
= 5.68K

The Rod and Rcmd should be adjusted in such a way that the effective pull-up is at the maximum 5.68K during enumeration. If there are only a few cards in the bus, a fixed Rcmd resister is sufficient and there is no need for an additional

Rod pull-up during enumeration. You should also ensure the effective pull-up will not violate the Iol rating of the drivers.

In SD mode, since each card has a separate bus, the capacitance is less, typically in the order of 20-30pf (host capacitance + card capacitance + trace + socket capacitance). For example, if enumeration is done at 400KHz and the total bus capacitance is 20pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz
R = 1/(2.2 * C * 100KHz)
= 1/(2.2 x 20 x 10**-12 x 400 x 10**3)
= 1/(1.76 x 10**-5)
= 56.8K

Therefore, a fixed 56.8K permanent Rcmd is sufficient in SD mode to enumerate the cards.

The driver of the SD/MMC Host Controller on the "command" port needs to be only a push-pull driver. During enumeration, the SD/MMC Host Controller emulates an open-drain driver by driving only a 0 or a "z" by controlling the ccmd_out and ccmd_out_en signals.

17.6.3 SD/MMC Controller IOMUX

The SDMMC Host Controller share the pin with GPIO. In default, the pins are used for GPIO, if user want to work in sdmmc function, the user must configure the GRF registers as following table:

Table 17-4 SDMMC IOMUX Settings

Module Pin	Direction	Pad Name	IOMUX Setting
sdmmc_clkout	O	GPIO3_B[0]	GPIO3B_IOMUX[0]=0x1& GPIO3B_IOMUX[16]=0x1
sdmmc_cmd	I/O	GPIO3_B[1]	GPIO3B_IOMUX[2]=0x1& GPIO3B_IOMUX[18]=0x1
sdmmc_data0	I/O	GPIO3_B[2]	GPIO3B_IOMUX[4]=0x1& GPIO3B_IOMUX[20]=0x1
sdmmc_data1	I/O	GPIO3_B[3]	GPIO3B_IOMUX[6]=0x1& GPIO3B_IOMUX[22]=0x1
sdmmc_data2	I/O	GPIO3_B[4]	GPIO3B_IOMUX[8]=0x1& GPIO3B_IOMUX[24]=0x1
sdmmc_data3	I/O	GPIO3_B[5]	GPIO3B_IOMUX[10]=0x1& GPIO3B_IOMUX[26]=0x1
sdmmc_detect_n	I	GPIO3_B[6]	GPIO3B_IOMUX[12]=0x1& GPIO3B_IOMUX[28]=0x1
sdmmc_write_prt	I	GPIO3_B[7]	GPIO3B_IOMUX[14]=0x1& GPIO3B_IOMUX[30]=0x1
sdmmc_rstn_out	O	GPIO3_A[6]	GPIO3A_IOMUX[12]=0x1& GPIO3A_IOMUX[28]=0x1
sdmmc_pwr_en	O	GPIO3_A[7]	GPIO3B_IOMUX[14]=0x1& GPIO3B_IOMUX[30]=0x1

*Notes: Direction: **I**– Input, **O**– Output, **I/O**– Input/Output*

17.7 Application Notes

17.7.1 Software/Hardware Restriction

Before issuing a new data transfer command, the software should ensure that the card is not busy due to any previous data transfer command. Before changing the card clock frequency, the software must ensure that there are no data or command transfers in progress.

If the card is enumerated in SDR50, or DDR50 mode, then the application must program the use_hold_reg bit[29] in the CMD register to 1'b0 (phase shift of cclk_in_drv = 0) or 1'b1 (phase shift of cclk_in_drv > 0). If the card is enumerated in SDR12 or SDR25 mode, the application must program the use_hold_reg bit[29] in the CMD register to 1'b1.

This programming should be done for all data transfer commands and non-data commands that are sent tothe card. When the use_hold_reg bit is programmed to 1'b0, the SD/MMC Controller bypasses the Hold Registers in the transmit path. The value of this bit should not be changed when a Command or Data Transfer is in progress. For more details on using use_hold_reg and the implementation requirements for meeting the Card input hold time, refer to "Recommended Usage" and Table 17-5.

Table 17-5 Recommended Usage of use_hold_reg

No.	Speed Mode	use_hold_reg	cclk_in	clk_in_drv	clk_divider
1	SDR50	1'b1	200	200	1
2	DDR50	1'b1	50	50	0
3	SDR25	1'b1	50	50	0
4	SDR12	1'b1	50	50	1

To avoid glitches in the card clock outputs (sdmmc_clkout), the software should use the following steps when changing the card clock frequency:

1. Before disable the clocks, ensure that the card is not busy due to any previous data command. To determine this, check for 0 in bit9 of STATUS register.
2. Update the Clock Enable register to disable all clocks. To ensure completion of any previous command before this update, send a command to the CIU to update the clock registers by setting:
   - start_cmd bit
   - "update clock registers only" bits
   - "wait_previous data complete" bit

   Wait for the CIU to take the command by polling for 0 on the start_cmd bit.
3. Set the start_cmd bit to update the Clock Divider and/or Clock Source registers, and send a command to the CIU in order to update the clock registers; wait for the CIU to take the command.
4. Set start_cmd to update the Clock Enable register in order to enable the required clocks and send a command to the CIU to update the clock registers; wait for the CIU to take the command.

In non-DMA mode, while reading from a card, the Data Transfer Over (RINTSTS[3]) interrupt occurs as soon as the data transfer from the card is over. There still could be some data left in the FIFO, and the RX_WMark interrupt may or may not occur, depending on the remaining bytes in the FIFO. Software should read any remaining bytes upon seeing the Data Transfer Over (DTO) interrupt. While using the external DMA interface for reading from a card, the DTO interrupt occurs only after all the data is flushed to memory by the DMA

interface unit.

While writing to a card in external DMA mode, if an undefined-length transfer is selected by setting the Byte Count Register to 0, the DMA logic will likely request more data than it will send to the card, since it has no way of knowing at which point the software will stop the transfer. The DMA request stops as soon as the DTO is set by the CIU.

If the software issues a controller_reset command by setting control register bit[0] to 1, all the CIU state machines are reset; the FIFO is not cleared. The DMA sends all remaining bytes to the host. In addition to a card-reset, if a FIFO reset is also issued, then:

- Any pending DMA transfer on the bus completes correctly
- DMA data read is ignored
- Write data is unknown(x)

Additionally, if dma_reset is also issued, any pending DMA transfer is abruptly terminated. When the DW-DMA is used, the DMA controller channel should also be reset and reprogrammed.

If any of the previous data commands do not properly terminate, then the software should issue the FIFO reset in order to remove any residual data, if any, in the FIFO. After asserting the FIFO reset, you should wait until this bit is cleared.

One data-transfer requirement between the FIFO and host is that the number of transfers should be a multiple of the FIFO data width (32bits). For example, you want to write only 15 bytes to an SDMMC card (BYTCNT), the host should write 16 bytes to the FIFO or program the DMA to do 16-byte transfers. The software can still program the Byte Count register to only 15, at which point only 15 bytes will be transferred to the card. Similarly, when 15 bytes are read from a card, the host should still read all 16 bytes from the FIFO.

It is recommended that you not change the FIFO threshold register in the middle of data transfers.

17.7.2 Programming Sequence

Initialization

Fig. 17-8 illustrates the initialization flow.

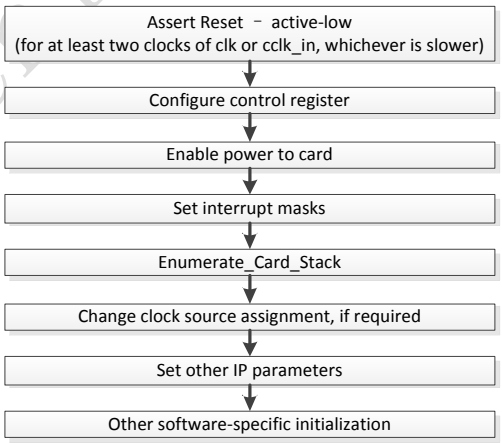


Fig. 17-8 Initialization Sequence

Once the power and clocks are stable, reset_n should be asserted(active-low) for at least two clocks of clk or cclk_in, whichever is slower. The reset initializes the registers, ports, FIFO-pointers, DMA interface controls, and state-machines in the design. After power-on reset, the software should do the following:

1. Configure control register – For MMC mode, enable the open-drain pullup by setting enable_OD_pullup(bit24) in the control register.
2. Enable power to cards – Before enabling the power, confirm that the voltage

setting to the voltage regulators is correct. Enable power to the connected cards by setting the corresponding bit to 1 in the Power Enable register. Wait for the power ramp-up time.

3. Set masks for interrupts by clearing appropriate bits in the Interrupt Mask register. Set the global int_enable bit of the Control register. It is recommended that you write 0xffff_ffff to the Raw Interrupt register in order to clear any pending interrupts before setting the int_enable bit.
4. Enumerate card stack – Each card is enumerated according to card type; for details, refer to "Enumerated Card Stack". For enumeration, you should restrict the clock frequency to 400KHz.
5. Changing clock source assignment – set the card frequency using the clock-divider and clock-source registers; for details, refer to "Clock Programming". MMC cards operate at a maximum of 20MHz (at maximum of 52MHz in high-speed mode). SD mode operates at a maximum of 25MHz (at maximum of 50MHz in high-speed mode).
6. Set other parameters, which normally do not need to be changed with every command, with a typical value such as timeout values in sdmmc_clkout according to SDMMC specifications.
   - ResponseTimeOut = 0x64
   - DataTimeOut = highest of one of the following:
     (10*((TAAC*Fop)+(100*NSAC))
     Host FIFO read/write latency from FIFO empty/full
   - Set the debounce value to 25ms(default:0x0fffff) in host clock cycle units in the DEBNCE register.
   - FIFO threshold value in bytes in the FIFOTH register. Typically, the threshold value can be set to half the FIFO depth; that is:
     RX_WMark=15;
     TX_WMark=16

Enumerated Card Stack

The card stack does the following:

- Enumerates all connected cards
- Sets the RCA for the connected cards
- Reads card-specific information
- Stores card-specific information locally

Enumeration depends on the operating mode of the SDMMC Host Controller; the card type is first identified and the appropriate card enumeration routine is called.

1. Check if the card is connected.
2. Clear the card type register to set the card width as a single bit. For the given card number, clear the corresponding bits in the card_type register. Clear the register bit for a 1-bit, 4-bit, or 8-bit bus width. For example, for card number=1, clear bit 0 and bit 16 of the card_type register.
3. Set clock frequency to Fod=400KHz, maximum – Program clock divider0 (bits 0-7 in the CLKDIV register) value to one-half of the cclk_in frequency divided by 400KHz. For example, if cclk_in is 20MHz, then the value is 20,000/(2*400)=25.
4. Identify the card type; that is, SD, MMC, or SDIO.
   a. Send CMD5 first. If a response is received, then the card is SDIO
   b. If not, send CMD8 with the following Argument
      Bit[31:12] = 20'h0 //reserved bits
      Bit[11:8] = 4'b0001 //VHS value
      Bit[7:0] = 8'b10101010 //Preferred Check Pattern by SD2.0
   c. If Response is received the card supports High Capacity SD2.0 then send

ACMD41 with the following Argument
Bit[31] = 1'b0; //Reserved bits
Bit[30] = 1'b1; //High Capacity Status
Bit[29:24] = 6'h0; //Reserved bits
Bit[23:0] = Supported Voltage Range

d. If Response is received for ACMD41 then the card is SD. Otherwise the card is MMC.
e. If response is not received for initial CMD8 then card does not support High Capacity SD2.0, then issue CMD0 followed by ACMD41 with the following Argument
Bit[31] = 1'b0; //Reserved bits
Bit[30] = 1'b0; //High Capacity Status
Bit[29:24] = 6'h0; //Reserved bits
Bit[23:0] = Supported Voltage Range

5. Enumerate the card according to the card type.
6. Use a clock source with a frequency = Fod (that is, 400KHz) and use the following enumeration command sequence:
   - SD card – Send CMD0, CMD8, ACMD41, CMD2, CMD3.
   - SDIO – Send CMD5, CMD3.
   - MMC – Send CMD0, CMD1, CMD2, CMD3.

Power Control

You can implement power control using the following registers, along with external circuitry:

- Control register bits card_voltage_a and card_voltage_b – Status of these bits is reflected at the IO pins. The bits can be used to generate or control the supply voltage that the memory cards require.
- Power enable register – Control power to individual cards.

Programming these two register depends on the implemented external circuitry. While turning on or off the power enable, you should confirm that power supply settings are correct. Power to all cards usually should be disable while switching off the power.

Clock Programming

The SDMMC controller supports four clock sources, each of which can be programmed with a different frequency; software can select the clock source for each card. The clock to an individual card can be enabled or disabled. Registers that support this are:

- CLKDIV – Programs individual clock source frequency.
- CLKSRC – Assign clock source for each card.
- CLKENA – Enables or disables clock for individual card and enables low-power mode, which automatically stops the clock to a card when the card is idle for more than 8 clocks.

The SDMMC Controller loads each of these registers only when the start_cmd bit and the Update_clk_regs_only bit in the CMD register are set. When a command is successfully loaded, the SDMMC Controller clears this bit, unless the SDMMC Controller already has another command in the queue, at which point it gives an HLE(Hardware Locked Error).

Software should look for the start_cmd and the Update_clk_regs_only bits, and should also set the wait_prvdata_complete bit to ensure that clock parameters do not change during data transfer. Note that even though start_cmd is set for updating clock registers, the SDMMC Controller does not raise a command_done signal upon command completion.

The following shows how to program these registers:

1. Confirm that no card is engaged in any transaction; if there is a transaction, wait until it finishes.
2. Stop all clocks by writing xxxx0000 to the CLKENA register. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
3. Program the CLKDIV and CLKSRC registers, as required. Set the start_cmd, Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmd is cleared or an HLE is set; in case of an HLE, repeat the command.
4. Re-enable all clocks by programming the CLKENA register. Set the start_cmd,Update_clk_regs_only, and wait_prvdata_complete bits in the CMD register. Wait until start_cmdis cleared or an HLE is set; in case of an HLE, repeat the command.

No-Data Command With or Without Response Sequence

To send any non-data command, the software needs to program the CMD register @0x2C and theCMDARG register @0x28 with appropriate parameters. Using these two registers, the SD/MMC controller forms the command and sends it to the command bus. The SD/MMC controller reflects the errors in the command response through the error bits of theRINTSTS register.

When a response is received – either erroneous or valid – the SD/MMC controller sets thecommand_done bit in the RINTSTS register. A short response is copied in Response Register0, while along response is copied to all four response registers @0x30, 0x34, 0x38, and 0x3C. The Response3register bit 31 represents the MSB, and the Response0 register bit 0 represents the LSB of a longresponse.

For basic commands or non-data commands, follow these steps:

1. Program the Command register @0x28 with the appropriate command argument parameter.
2. Program the Command register @0x2C with the settings in Table 17-6.

Table 17-6 Command Settings for No-Data Command

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on CMD register
Update_clk_regs_only	0	No clock parameters update command
data_expected	0	No data command
card number	0	Actual card number(one controller only connect one card, the num is No.0)
cmd_index	command-index	-
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
wait_prvdata_complete	1	Before sending command on command line,

		host should wait for completion of any data command in process, if any (recommended to always set this bit, unless the current command is to query status or stop data transfer when transfer is in progress)
check_response_crc	1	If host should crosscheck CRC of response received

3. Wait for command acceptance by host. The following happens when the command is loaded intothe SD/MMC controller:
   - SD/MMC controller accepts the command for execution and clears the start_cmd bit in theCMD register, unless one command is in process, at which point the SD/MMC controller can load and keep the second command in the buffer.
   - If the SD/MMC controller is unable to load the command – that is, a command is already inprogress, a second command is in the buffer, and a third command is attempted – then itgenerates an HLE (hardware-locked error).
4. Check if there is an HLE.
5. Wait for command execution to complete. After receiving either a response from a card orresponse timeout, the SD/MMC controller sets the command_done bit in the RINTSTS register.Software can either poll for this bit or respond to a generated interrupt.
6. Check if response_timeout error, response_CRC error, or response error is set. This can be doneeither by responding to an interrupt raised by these errors or by polling bits 1, 6, and 8 from theRINTSTS register @0x44. If no response error is received, then the response is valid. If required,thesoftware can copy the response from the response registers @0x30-0x3C.

Software should not modify clock parameters while a command is being executed.

Data Transfer Commands

Data transfer commands transfer data between the memory card and the SD/MMC controller. To send a data command, the SD/MMC controller needs a command argument, total data size, and block size. Software can receive or send data through the FIFO.

Before a data transfer command, software should confirm that the card is not busy and is in a transfer state, which can be done using the CMD13 and CMD7 commands, respectively.

For the data transfer commands, it is important that the same bus width that is programmed in the card should be set in the card type register @0x18.
The SD/MMC controller generates an interrupt for different conditions during data transfer, which are reflected in the RINTSTS register @0x44 as:

1. Data_Transfer_Over (bit 3) – When data transfer is over or terminated. If there is a response timeout error, then the SD/MMC Host Controller does not attempt any data transfer and the "Data Transfer Over" bit is never set.
2. Transmit_FIFO_Data_request (bit 4) – FIFO threshold for transmitting data was reached; software is expected to write data, if available, in FIFO.

3. Receive_FIFO_Data_request (bit 5) – FIFO threshold for receiving data was reached; software is expected to read data from FIFO.
4. Data starvation by Host timeout (bit 10) – FIFO is empty during transmission or is full during reception. Unless software writes data for empty condition or reads data for full condition, the SD/MMC controller cannot continue with data transfer. The clock to the card has been stopped.
5. Data read timeout error (bit 9) – Card has not sent data within the timeout period.
6. Data CRC error (bit 7) – CRC error occurred during data reception.
7. Start bit error (bit 13) – Start bit was not received during data reception.
8. End bit error (bit 15) – End bit was not received during data reception or for a write operation; a CRC error is indicated by the card.

Conditions 6, 7, and 8 indicate that the received data may have errors. If there was a response timeout, then no data transfer occurred.

Single-Block or Multiple-Block Read

Steps involved in a single-block or multiple-block read are:

1. Write the data size in bytes in the BYTCNT register @0x20.
2. Write the block size in bytes in the BLKSIZ register @0x1C. The SD/MMC controller expects data from the card in blocks of size BLKSIZ each.
3. Program the CMDARG register @0x28 with the data address of the beginning of a data read.
   Program the Command register with the parameters listed in Table 17-7. For SD and MMC cards, use CMD17 for a single-block read and CMD18 for a multiple-block read. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 17-7 Command Setting for Single-Block or Multiple-Block Read

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on CMD register
Update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No.0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0 or 1	Set according to Table xx
transfer_mode	0	Block transfer
read_write	0	Read from card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-
wait_prvdata_complete	1	0- Sends command immediately 1- Sends command after previous data transfer ends

check_response_crc	1	0- SD/MMC controller should not check response CRC 1- SD/MMC controller should check response CRC

After writing to the CMD register, the SD/MMC controller starts executing the command; when the command is sent to the bus, the command_done interrupt is generated.

4. Software should look for data error interrupts; that is, bits 7, 9, 13, and 15 of the RINTSTS register. If required, software can terminate the data transfer by sending a STOP command.
5. Software should look for Receive_FIFO_Data_request and/or data starvation by host timeout conditions. In both cases, the software should read data from the FIFO and make space in the FIFO for receiving more data.
6. When a Data_Transfer_Over interrupt is received, the software should read the remaining data from the FIFO.

Single-Block or Multiple-Block Write

Steps involved in a single-block or multiple-block write are:

1. Write the data size in bytes in the BYTCNT register @0x20.
2. Write the block size in bytes in the BLKSIZ register @0x1C; the SD/MMC controller sends data in blocks of size BLKSIZ each.
3. Program CMDARG register @0x28 with the data address to which data should be written.
4. Write data in the FIFO; it is usually best to start filling data the full depth of the FIFO.
5. Program the Command register with the parameters listed in Table 17-8. For SD and MMC cards, use CMD24 for a single-block write and CMD25 for a multiple-block write. For SDIO cards, use CMD53 for both single-block and multiple-block transfers.

Table 17-8 Command Settings for Single-Block or Multiple-Block Write

Parameter	Value	Description
Default		
start_cmd	1	-
use_hold_reg	1/0	Choose value based on speed mode being used;ref to "use_hold_reg" on CMD register
Update_clk_regs_only	0	No clock parameters update command
card number	0	Actual card number(one controller only connect one card, the num is No.0)
send_initialization	0	Can be 1, but only for card reset commands, such as CMD0
stop_abort_cmd	0	Can be 1 for commands to stop data transfer, such as CMD12
send_auto_stop	0 or 1	Set according to Table xx
transfer_mode	0	Block transfer
read_write	1	Write to card
data_expected	1	Data command
response_length	0	Can be 1 for R2(long) response
response_expect	1	Can be 0 for commands with no response; for example, CMD0, CMD4, CMD15, and so on
User-selectable		
cmd_index	command-index	-

wait_prvdata_complete	1	0- Sends command immediately 1- Sends command after previous data transfer ends
check_response_crc	1	0- SD/MMC controller should not check response CRC 1- SD/MMC controller should check response CRC

After writing to the CMD register, SD/MMC controller starts executing a command; when the command is sent to the bus, a command_done interrupt is generated.

6. Software should look for data error interrupts; that is, for bits 7, 9, and 15 of the RINTSTS register. If required, software can terminate the data transfer by sending the STOP command.
7. Software should look for Transmit_FIFO_Data_request and/or timeout conditions from data starvation by the host. In both cases, the software should write data into the FIFO.
8. When a Data_Transfer_Over interrupt is received, the data command is over. For an open-ended block transfer, if the byte count is 0, the software must send the STOP command. If the byte count is not 0, then upon completion of a transfer of a given number of bytes, the SD/MMC Host Controller should send the STOP command, if necessary. Completion of the AUTO-STOP command is reflected by the Auto_command_done interrupt – bit 14 of the RINTSTS register. A response to AUTO_STOP is stored in RESP1 @0x34.

Stream Read

A stream read is like the block read mentioned in "Single-Block or Multiple-Block Read",except for the following bits in the Command register:

transfer_mode = 1; //Stream transfer

cmd_index = CMD20;

A stream transfer is allowed for only a single-bit bus width.

Stream Write

A stream write is exactly like the block write mentioned in "Single-Block or Multiple-Block Write", except for the following bits in the Command register:

transfer_mode = 1;//Stream transfer

cmd_index = CMD11;

In a stream transfer, if the byte count is 0, then the software must send the STOP command. If the bytecount is not 0, then when a given number of bytes completes a transfer, the SD/MMC controller sends the STOP command. Completion of this AUTO_STOP command is reflected by theAuto_command_done interrupt. A response to an AUTO_STOP is stored in the RESP1 register@0x34.

A stream transfer is allowed for only a single-bit bus width.

Sending Stop or Abort in Middle of Transfer

The STOP command can terminate a data transfer between a memory card and theSD/MMC controller, while the ABORT command can terminate an I/O data transfer for only the SDIO_IOONLY and SDIO_COMBO cards.

- Send STOP command – Can be sent on the command line while a data

transfer is in progress; this command can be sent at any time during a data transfer. For information on sending this command, refer to "No-Data Command With or Without Response Sequence".You can also use an additional setting for this command in order to set the Command register bits (5-0) to CMD12 and set bit 14 (stop_abort_cmd) to 1. If stop_abort_cmd is not set to 1, the SD/MMC controller does not know that the user stopped a data transfer. Reset bit 13 of the Command register (wait_prvdata_complete) to 0 in order to make the SD/MMC controller sendthe command at once, even though there is a data transfer in progress.

- Send ABORT command – Can be used with only an SDIO_IOONLY or SDIO_COMBO card. To abort the function that is transferring data, program the function number in ASx bits (CCCR register of card, address 0x06, bits (0-2) using CMD52.
  This is a non-data command. For information on sending this command, refer to "No-Data Command With or Without Response Sequence".
  The command format for CMD52 is illustrated in Fig. 17-9:

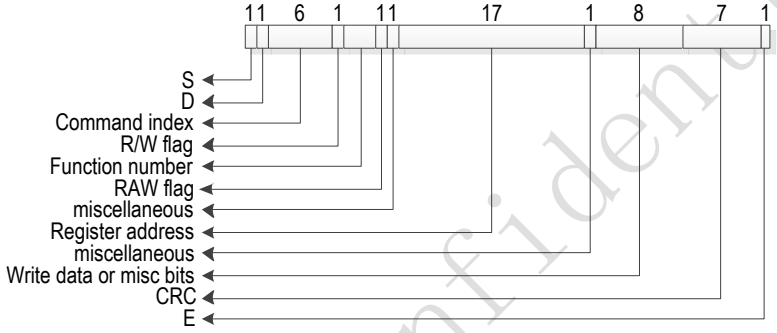


Fig. 17-9 Command format for CMD52

a. Program the CMDARG register @0x28 with the appropriate command argument parameterslisted in Table 17-9.

Table 17-9 Parameters for CMDARG Registers

CMDARG Bits	Contents	Value
31	R/W flag	1
30-28	Function Number	0, for CCCR access
27	RAW flag	1, if needed to read after write
26	Don't care	-
25-9	Register address	0x06
8	Don't care	-
7-0	Write Data	Function number to be aborted

b. Program the Command register using the command index as CMD52. Similar to the STOP command described, set bit 14 of the Command register (stop_abort_cmd) to 1, which must be done in order to inform the SD/MMC controller that the user aborted the data transfer. Reset bit 13 (wait_prvdata_complete) of the Command register to 0 in order to make the SD/MMC controller send the command at once, even though a data transfer is in progress.
c. Wait for command_transfer_over.
d. Check response (R5) for errors.

Suspend or Resume Sequence

In an SDIO card, the data transfer between an I/O function and the SD/MMC controller can betemporarily halted using the SUSPEND command; this may be

required in order to perform ahigh-priority data transfer with another function. When desired, the data transfer can be resumed usingthe RESUME command.

The following functions can be implemented by programming the appropriate bits in the CCCRregister (Function 0) of the SDIO card. To read from or write to the CCCR register, use the CMD52command.

1. SUSPEND data transfer – Non-data command.
   a. Check if the SDIO card supports the SUSPEND/RESUME protocol; this can be done through the SBS bit in the CCCR register @0x08 of the card.
   b. Check if the data transfer for the required function number is in process; the function number that is currently active is reflected in bits 0-3 of the CCCR register @0x0D. Note that if the BS bit (address 0xc::bit 0) is 1, then only the function number given by the FSx bits is valid.
   c. To suspend the transfer, set BR (bit 2) of the CCCR register @0x0C.
   d. Poll for clear status of bits BR (bit 1) and BS (bit 0) of the CCCR @0x0C. The BS (Bus Status) bit is 1 when the currently-selected function is using the data bus; the BR (Bus Release) bit remains 1 until the bus release is complete. When the BR and BS bits are 0, the data transfer from the selected function has been suspended.
   e. During a read-data transfer, the SD/MMC controller can be waiting for the data from the card. If the data transfer is a read from a card, then the SD/MMC controller must be informed after the successful completion of the SUSPEND command. The SD/MMC controller then resets the data state machine and comes out of the wait state. To accomplish this, set abort_read_data (bit 8) in the Control register.
   f. Wait for data completion. Get pending bytes to transfer by reading the TCBCNT register @0x5C.
2. RESUME data transfer – This is a data command.
   a. Check that the card is not in a transfer state, which confirms that the bus is free for data transfer.
   b. If the card is in a disconnect state, select it using CMD7. The card status can be retrieved in response to CMD52/CMD53 commands.
   c. Check that a function to be resumed is ready for data transfer; this can be confirmed by reading the RFx flag in CCCR @0x0F. If RF = 1, then the function is ready for data transfer.
   d. To resume transfer, use CMD52 to write the function number at FSx bits (0-3) in the CCCR register @0x0D. Form the command argument for CMD52 and write it in CMDARG @0x28; bit values are listed in Table 17-10.

Table 17-10 CMDARG Bit Values

CMDARG Bits	Contents	Value
31	R/W flag	1
30-28	Function Number	0, for CCCR access
27	RAW flag	1, read after write
26	Don’t care	-
25-9	Register address	0x0D
8	Don’t care	-
7-0	Write Data	Function number to be resumed

   e. Write the block size in the BLKSIZ register @0x1C; data will be transferred in units of this block size.
   f. Write the byte count in the BYTCNT register @0x20. This is the total size of the data; that is, the remaining bytes to be transferred. It is the responsibility of the software to handle the data.

g. Program Command registers; similar to a block transfer. For details, refer to "Single-Block or Multiple-Block Read" and "Single-Block or Multiple-Block Write".
h. When the Command register is programmed, the command is sent and the function resumes data transfer. Read the DF flag (Resume Data Flag). If it is 1, then the function has data for the transfer and will begin a data transfer as soon as the function or memory is resumed. If it is 0, then the function has no data for the transfer.
i. If the DF flag is 0, then in case of a read, the SD/MMC Host Controller waits for data. After the data timeout period, it gives a data timeout error.

Read Wait Sequence

Read_wait is used with only the SDIO card and can temporarily stall the data transfer- either fromfunction or memory—and allow the host to send commands to any function within the SDIO device.The host can stall this transfer for as long as required. The SD/MMC Host Controller provides the facilityto signal this stall transfer to the card. The steps for doing this are:

1. Check if the card supports the read_wait facility; read SRW (bit 2) of the CCCR register @0x08. Ifthis bit is 1, then all functions in the card support the read_wait facility. Use CMD52 to read thisbit.
2. If the card supports the read_wait signal, then assert it by setting the read_wait (bit 6) in the CTRLregister @0x00.
3. Clear the read_wait bit in the CTRL register.

Controller/DMA/FIFO Reset Usage

Communication with the card involves the following:

- Controller – Controls all functions of the SD/MMC controller.
- FIFO – Holds data to be sent or received.
- DMA – If DMA transfer mode is enabled, then transfers data between system memory and the FIFO.
- Controller reset – Resets the controller by setting the controller_reset bit (bit 0) in the CTRL register; this resets the CIU and state machines, and also resets the BIU-to-CIU interface. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- FIFO reset – Resets the FIFO by setting the fifo_reset bit (bit 1) in the CTRL register; this resets the FIFO pointers and counters of the FIFO. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.
- DMA reset – Resets the internal DMA controller logic by setting the dma_reset bit (bit 2) in the CTRL register, which abruptly terminates any DMA transfer in process. Since this reset bit is self-clearing, after issuing the reset, wait until this bit is cleared.

The following are recommended methods for issuing reset commands:

- Non-DMA transfer mode – Simultaneously sets controller_reset and fifo_reset; clears the RAWINTS register @0x44 using another write in order to clear any resultant interrupt.
- DMA mode – Sets controller_reset and fifo_reset; waits until dma_req goes inactive (the Status register indicates the value of this signal). Resets the FIFO again. Clears the interrupts by clearing the RAWINTS register @0x44 using another write in order to clear any resultant interrupt. You also need to reset and reprogram the channel(s) of the DMA controller that are

interfaced to the SD/MMC Host Controller.
In external DMA transfer mode, even when the FIFO pointers are reset, if there is a DMA transfer in progress, it could push or pop data to or from the FIFO; the DMA itself completes correctly. In order to clear the FIFO, the software should issue an additional FIFO reset and clear any FIFO underrun or overrun errors in the RAWINTS register caused by the DMA transfers after the FIFO was reset.

Error Handling

The SD/MMC controller implements error checking; errors are reflected in the RAWINTS register@0x44 and can be communicated to the software through an interrupt, or the software can poll forthese bits. Upon power-on, interrupts are disabled (int_enable in the CTRL register is 0), and all theinterrupts are masked (bits 0-31 of the INTMASK register; default is 0).
Error handling:

- Response and data timeout errors – For response timeout, software can retry the command. Fordata timeout, the SD/MMC controller has not received the data start bit – either for the firstblock or the intermediate block – within the timeout period, so software can either retry the wholedata transfer again or retry from a specified block onwards. By reading the contents of theTCBCNT later, the software can decide how many bytes remain to be copied.
- Response errors – Set when an error is received during response reception. In this case, theresponse that copied in the response registers is invalid. Software can retry the command.
- Data errors – Set when error in data reception are observed; for example, data CRC, start bit notfound, end bit not found, and so on. These errors could be set for any block-first block,intermediate block, or last block. On receipt of an error, the software can issue a STOP or ABORTcommand and retry the command for either whole data or partial data.
- Hardware locked error – Set when the SD/MMC controller cannot load a command issued bysoftware. When software sets the start_cmd bit in the CMD register, the SD/MMC controller tries to load the command. If the command buffer is already filled with a command, this error israised. The software then has to reload the command.
- FIFO underrun/overrun error – If the FIFO is full and software tries to write data in the FIFO, thenan overrun error is set. Conversely, if the FIFO is empty and the software tries to read data fromthe FIFO, an underrun error is set. Before reading or writing data in the FIFO, the software shouldread the fifo_empty or fifo_full bits in the Status register.
- Data starvation by host timeout – Raised when the SD/MMC controller is waiting for softwareintervention to transfer the data to or from the FIFO, but the software does not transfer within thestipulated timeout period. Under this condition and when a read transfer is in process, the softwareshould read data from the FIFO and create space for further data reception. When a transmitoperation is in process, the software should fill data in the FIFO in

order to start transferring datato the card.

- CRC Error on Command – If a CRC error is detected for a command, the CE-ATA device does notsend a response, and a response timeout is expected from the SD/MMC controller. The ATAlayer is notified that an MMC transport layer error occurred.

Notes: During a multiple-block data transfer, if a negative CRC status is received from thedevice, the data path signals a data CRC error to the BIU by setting the data CRC error bitin the RINTSTS register. It then continues further data transmission until all the bytes aretransmitted.

17.7.3 Programming SD/MMC Controller for Boot Operation

Boot Operation

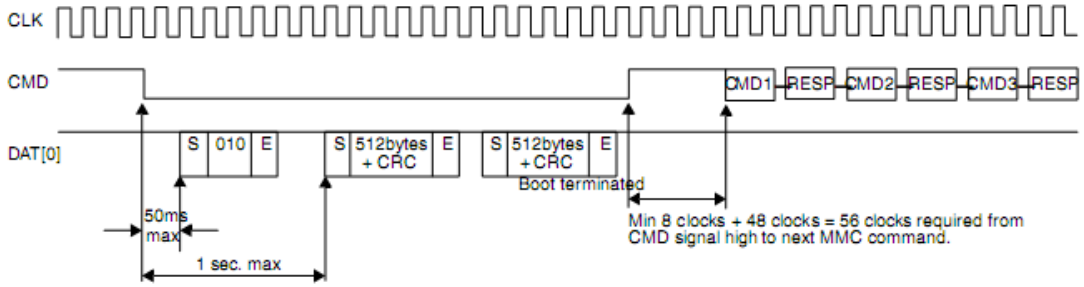


Fig. 17-10 Boot Operation

Fig. 17-10 illustrates timing for Boot operation.

Once the power and clocks are stable, reset_n should be asserted (active-low) for at least two clocks of clk or cclk_in, whichever is slower. The reset initializes the following:

- Registers
- Ports
- FIFO-pointers
- DMA interface controls
- State-machines in the design

After power-on reset, the software should perform the appropriate steps described in the following sections for the respective types of cards.

Following are the steps that the software driver must follow when working with eMMC cards for Boot operation.

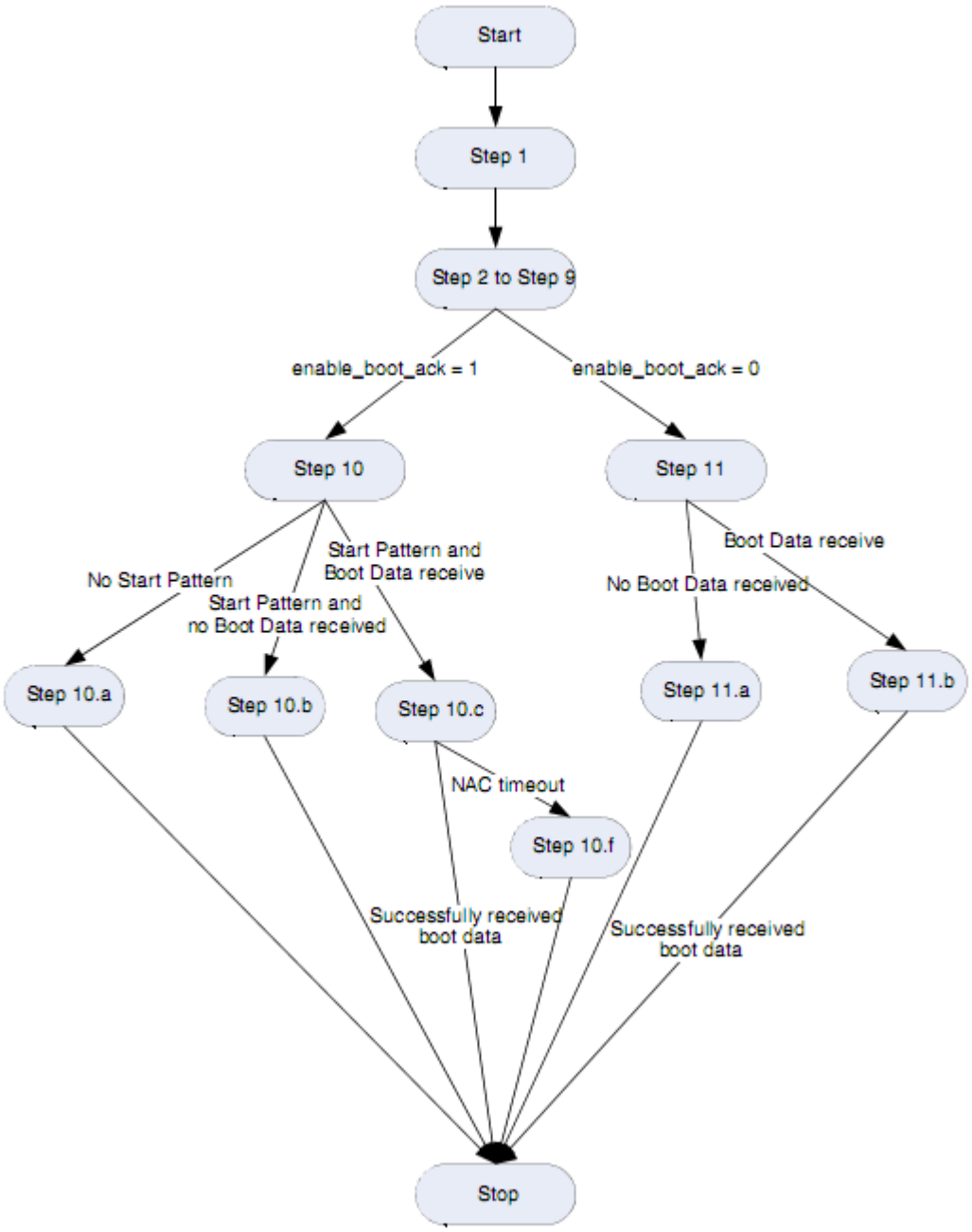


Fig. 17-11 SD/MMC Controller Flow for Boot Operation

1. The software driver is aware:
   - That the card supports boot operation—BOOT_PARTITION_ENABLE bit set in the card.
   - Of the BOOT_SIZE_MULT value in the card and the data bus width to use during boot operation—Extend CSD register byte[177] bit[0:1].
2. Set the following:
   - Masks for interrupts by clearing appropriate bits in the Interrupt Mask register @0x024.
   - Global int_enable bit of the Control register @0x00.

It is recommended that you write 0xffff_ffff to the Raw Interrupt register @0x044 and IDSTS @0x8C in order to clear any pending interrupts before

setting the int_enable bit.

For Internal DMAC mode, the software driver needs to unmask all the relevant fields in the IDINTEN register.

3. Configure control register (CTRL):
   - int_enable = 1'b1
   - Other fields should be 1'b0.
4. Change clock source assignment – Set the card frequency to 400 KHz using the clock-divider and clock-source registers; for details, refer to "Clock Programming".
5. Set DataTimeOut = (10 * ((TAAC * Fop) + (100 * NSAC)); this is NAC.
6. Program the BLKSIZ register with 0x200 (512 bytes).
7. Program the BYTCNT register with multiples of 128K bytes, as indicated by the BOOT_SIZE_MULT value in the card.
8. Program the Rx FIFO threshold value in bytes in the FIFOTH register @0x04C.Typically, the threshold value can be set to half the FIFO depth; that is, RX_WMark = (FIFO_DEPTH/2) - 1.
9. Program the CMD register with the following fields:
   - start_cmd = 1'b1
   - enable_boot = 1'b1
   - enable_boot_ack – depends on whether a start-acknowledge pattern is expected from the card
   - Card_number = appropriate_card_number; obtained by referring to CDETECT register
   - Data_expected = 1'b1
   - Remainder of CMD register fields = 1'b0
10. If enable_boot_ack = 1'b1, the software driver should start a timer after step #9; the terminal value is 50ms.
    - Before this timer elapses, the BAR interrupt should be received from the SD/MMC Controller. If this does not occur, the software driver must program the CMD register with the following fields:
      - start_cmd = 1'b1
      - disable_boot = 1'b1
      - All other fields = 0

      The SD/MMC Controller generates a Command Done (CD) interrupt after de-asserting the CMD line of the card.
    - If the BAR interrupt is received, the software driver should clear this interrupt by writing a 1 to it. The software driver should then start another timer with a terminal value of 1 - 0.05 = 0.95 seconds. Before this timer elapses, the BDS interrupt should be received from the SD/MMC Controller. If this does not occur, the software driver must program the CMD register with the following fields:
      - start_cmd = 1'b1
      - disable_boot = 1'b1
      - All other fields = 0

The SD/MMC Controller generates a Command Done (CD) interrupt after de-asserting the CMD line of the card.

- If the BDS interrupt is received, it indicates that the boot data is being received from the card. The software driver can then initiate a data read from the SD/MMC Controller based on the RXDR interrupt bit in the RINTSTS register.At the end of a successful boot data transfer from the card, the following interrupts are generated:
  - Command Done (CD) in RINTSTS register
  - Data Transfer Over (DTO) in RINTSTS register
- If an Error occurs in Boot Ack pattern (010) or an end bit Error occurs:
  - RTL automatically aborts boot by pulling CMD line high
  - RTL generates Command done interrupt
  - RTL does not generate BAR interrupt
  - Application aborts boot transfer
- If between data block transfers NAC is violated, DRTO (Data Read Timeout) is asserted.Apart from this, if there are errors associated with Start/End bits, SBE/EBE interrupts are also generated.

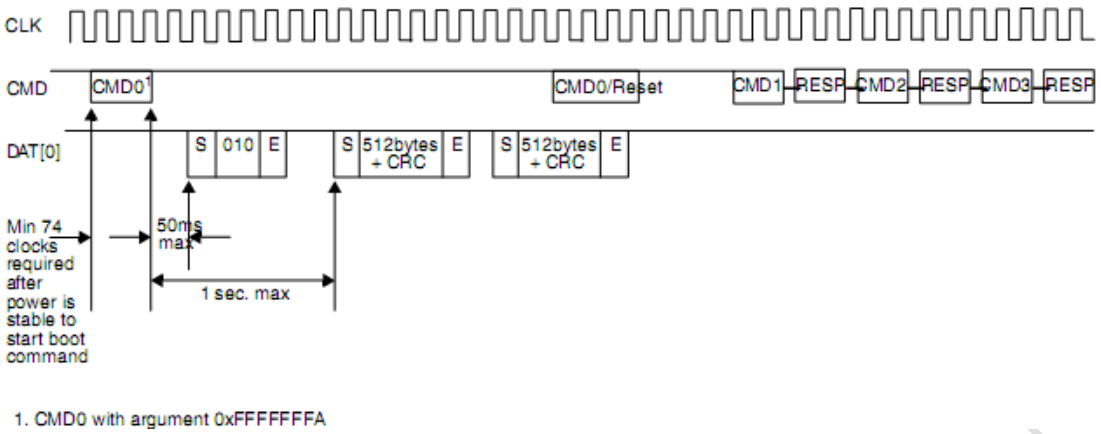
11. If enable_boot_ack = 1'b0, the software driver should start a timer after the step #9 where the terminal value is 1 second.
    - Before this timer elapses, a BDS interrupt should be received from the SD/MMC Controller. If this does not occur, the software driver must program the CMD register with the following fields:
      - start_cmd = 1'b1
      - disable_boot = 1'b1
      - All other fields = 0

    The SD/MMC Controller generates a Command Done (CD) interrupt after de-asserting the CMD line of the card.
    - If a BDS interrupt is received, it indicates that the boot data is being received from the card. The software driver can then initiate a data read from the SD/MMC Controller based on the RXDR interrupt bit in the RINTSTS register.

    At the end of a successful boot data transfer from card, the following interrupts are generated.
      - Command Done (CD) in RINTSTS.
      - Data Transfer Over (DTO) in RINTSTS.

Alternative Boot Operation

The Alternative Boot Operation differs from the Boot Operation in that CMD0 is used to boot the card rather than holding down the CMD-line of the card. The Alternative Boot Operation can be done only if bit 0 in the extended CSD byte[228] (BOOT_INFO) is set to 1.



1. CMD0 with argument 0xFFFFFFFA

Fig. 17-12 Alternative Boot Opertation

Following are the steps that the software driver must follow when working with eMMC for the Alternative Boot operation.

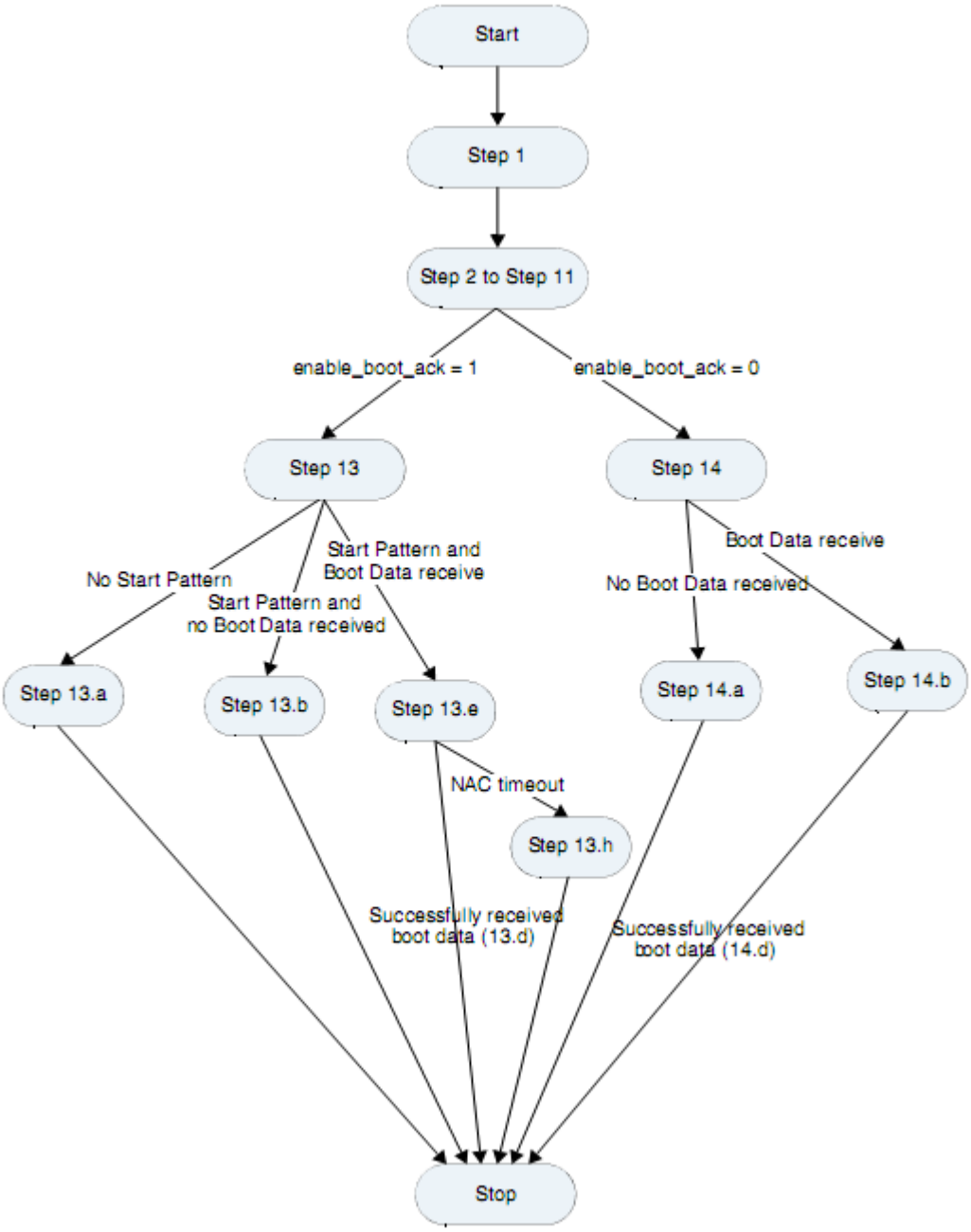


Fig. 17-13 Host Controller Flow for Alternative Boot Mode

1. The software driver is aware:
   - That the card supports the Alternative Boot operation—BOOT_INFO bit is set in the card.
   - Of the BOOT_SIZE_MULT value in the card and the data bus width to use during the boot operation—Extend CSD register byte[177] bit[0:1]..
2. Set the following:
   - Masks for interrupts by clearing appropriate bits in the Interrupt Mask register @0x024
   - Global int_enable bit of the Control register @0x00

   It is recommended that you write 0xffff_ffff to the Raw Interrupt register @0x044 and IDSTS @0x8C in order to clear any pending interrupts

before setting the int_enable bit.
For Internal DMAC mode, software driver needs to unmask all the relevant fields in IDINTEN register.

3. Configure control register (CTRL):
   - enable_OD_pullup = 1'b0
   - int_enable = 1'b1
   - Other fields should be 1'b0
4. Changing clock source assignment – Set the card frequency to 400 KHz using the clock-divider and clock-source registers; for details, refer to "Clock Programming".Ensure that the card clock—cclk_out—is running.
5. Wait for a time that ensures that at least 74 card clock cycles have occurred on the card interface.
6. Set DataTimeOut = (10 * ((TAAC * Fop) + (100 * NSAC)); this is NAC.
7. Program the BLKSIZ register with 0x200—512 bytes.
8. Program the BYTCNT register with multiples of 128K bytes, as indicated by the BOOT_SIZE_MULT value in the card.
9. Program the Rx FIFO threshold value in bytes in the FIFOTH register @0x04C.Typically, the threshold value can be set to half the FIFO depth; that is, RX_WMark = (FIFO_DEPTH/2) - 1.
10. Program CMDARG = 0xFFFFFFFA.
11. Program the CMD register with the following fields.
    - start_cmd = 1'b1
    - boot_mode = 1'b1
    - enable_boot_ack – depends on whether a start-acknowledge pattern is expected from the card.
    - Card_number – appropriate_card_number, obtained by referring to CDETECT register
    - Data_expected = 1'b1
    - Cmd_index = 0
    - Remainder of CMD register fields = 1'b0
12. The software driver should wait for the Command Done (CD) interrupt.
13. If enable_boot_ack = 1'b1 in step 11, the software driver should start a timer after the above step with a terminal value of 50ms.
    - Before this timer elapses, the BAR interrupt should be received fromthe SD/MMC Controller. If this does not occur, the software driver needs to infer that the start-pattern has not been received and should discontinue the boot process and start with normal enumeration.
    - If the BAR interrupt is received, the software driver should clear this interrupt by writing a 1 to it. The software driver should then start another timer with a terminal value of 1 - 0.05 = 0.95 seconds. Before this timer elapses, the BDS interrupt should be received from the SD/MMC Controller. If this does not occur, the software driver should discontinue the boot process and start with normal enumeration.
    - If the BDS interrupt is received, it indicates that the boot data is being received from the card. In non-IDMAC mode, the software driver can then initiate a data read from the SD/MMC Controller based on the RXDR interrupt bit in the RINTSTS register.
    - It is the responsibility of the software driver to terminate the boot operation by programming the SD/MMC Controller to send a CMD0 by programming the registers CMDARG = 0 and CMD = {start_cmd = 1, card number = appropriate_card_number, cmd_index = 0, all_other_fields = 0}.

- At the end of a successful boot data transfer from the card, the following interrupts are:
  - Command Done (CD) in RINTSTS
  - Data Transfer Over (DTO) in RINTSTS
  - Receive Interrupt (RI) in IDSTS in IDMAC mode only
- If an Error occurs in Boot Ack pattern (010) or an end bit Error occurs:
  - RTL does not generate BAR interrupt
  - RTL detects Boot Data Start and generates BDS interrupt
  - RTL continues to receive Boot Data
  - Application must abort boot after receiving BDS interrupt
- If between data block transfers NAC is violated, DRTO (Data Read Timeout) is asserted.Apart from this, if there are errors associated with Start/End bits, SBE/EBE interrupts are also generated.

14. If enable_boot_ack = 1'b0 in step 11, the software driver should start a timer after step #11with a terminal value of 1 second.
    - Before this timer elapses, the BDS interrupt should be received from the SD/MMC Controller. If this does not occur, the software driver should discontinue the boot process and start with normal enumeration.
    - If the BDS interrupt is received, it indicates that the boot data is being received from the card. In non-IDMAC mode, the software driver can then initiate a data read from the SD/MMC Controller based on the RXDR (in RINTSTS) interrupt.
    - It is the responsibility of the software driver to terminate the boot operation by programming the SD/MMC Controller to send a CMD0 by programming the registers CMDARG = 0 and CMD = {start_cmd=1,card number = appropriate card number, cmd_index =0, rest of the fields = 0}.
    - At the end of a successful boot data transfer from card, the following interrupts are generated.
      - Command Done (CD) in RINTSTS.
      - Data Transfer Over (DTO) in RINTSTS.
      - Receive Interrupt (RI) in IDSTS in IDMAC mode only.

17.7.4 Voltage Switching and DDR Operations

Voltage Switch Operation

The Voltage Switch operation must be performed in SD mode only.

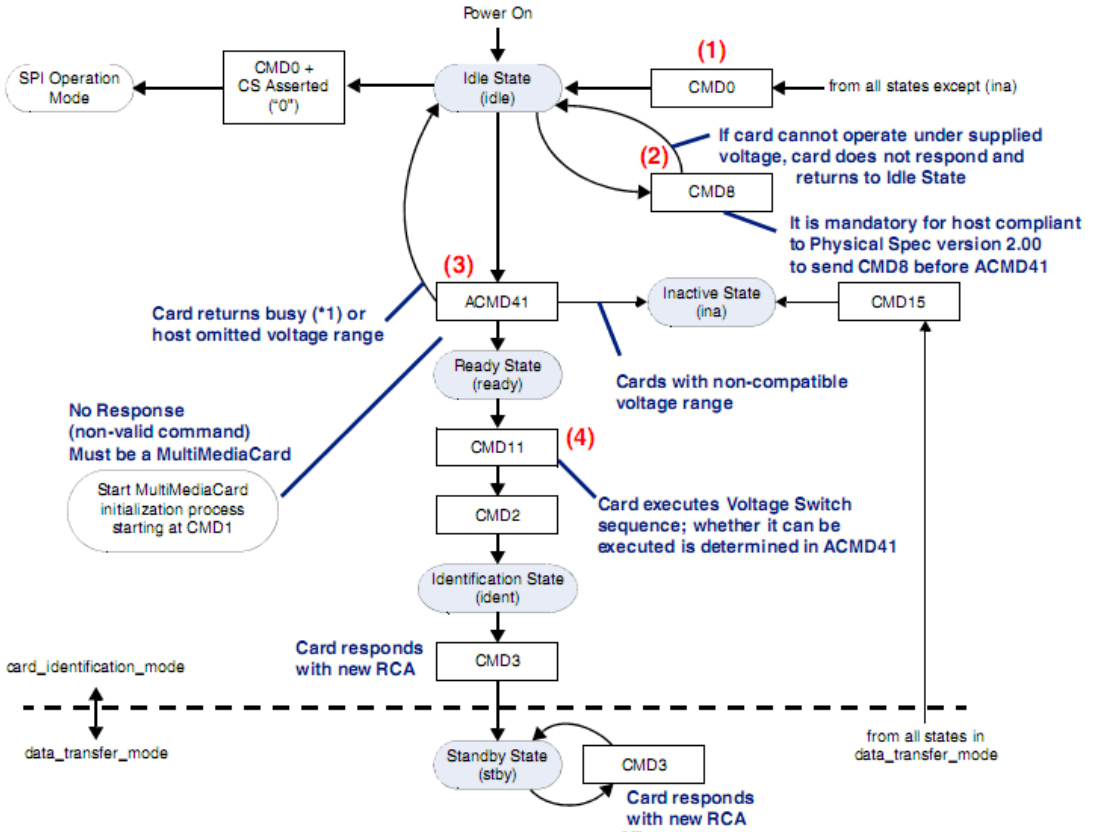


Fig. 17-14 Voltage Switching Command Flow Diagram

The following outlines the steps for the voltage switch programming sequence

1. Software Driver starts CMD0, which selects the bus mode as SD.
2. After the bus is in SD card mode, CMD8 is started in order to verify if the card is compatible with the SD Memory Card Specification, Version 2.00. CMD8 determines if the card is capable of working within the host supply voltage specified in the VHS (19:16) field of the CMD; the card supports the current host voltage if a response to CMD8 is received.
3. ACMD 41 is started.The response to this command informs the software if the card supports voltage switching; bits 38, 36, and 32 are checked by the card argument of ACMD41; refer to Figure 17-15.

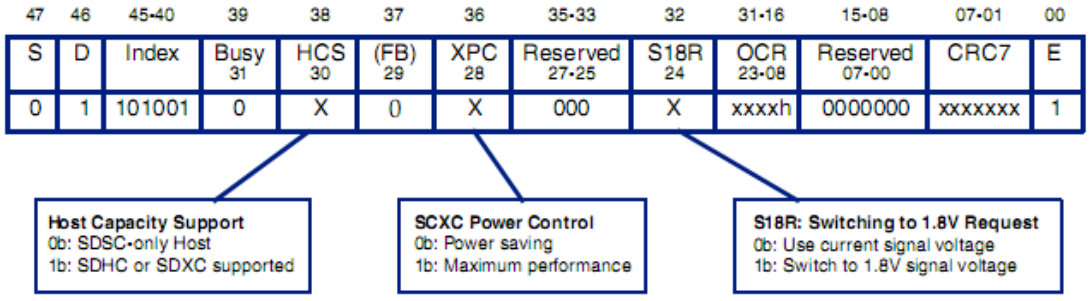


Fig. 17-15 ACMD41 Argument

- Bit 30 informs the card if host supports SDHC/SDXC or not; this bit should be set to 1'b1.
- Bit 28 can be either 1 or 0.
- Bit 24 should be set to 1'b1, indicating that the host is capable of voltage switching; refer to Figure 17-16.

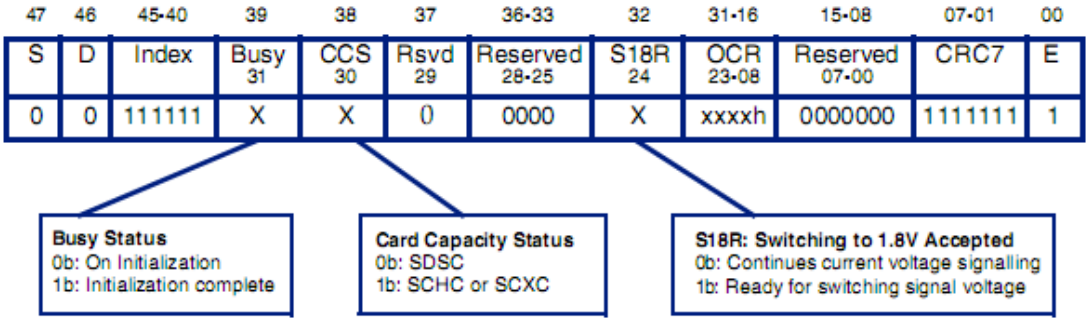


Fig. 17-16 ACMD41 Response(R3)

- Bit 30 – If set to 1'b1, card supports SDHC/SDXC; if set to 1'b0, card supports only SDSC
- Bit 24 – If set to 1'b1, card supports voltage switching and is ready for the switch
- Bit 31 – If set to 1'b1, initialization is over; if set to 1'b0, means initialization in process

4. If the card supports voltage switching, then the software must perform the steps discussed for either the "Voltage Switch Normal Scenario" or the "Voltage Switch Error Scenario".

Voltage Switch Normal Scenario

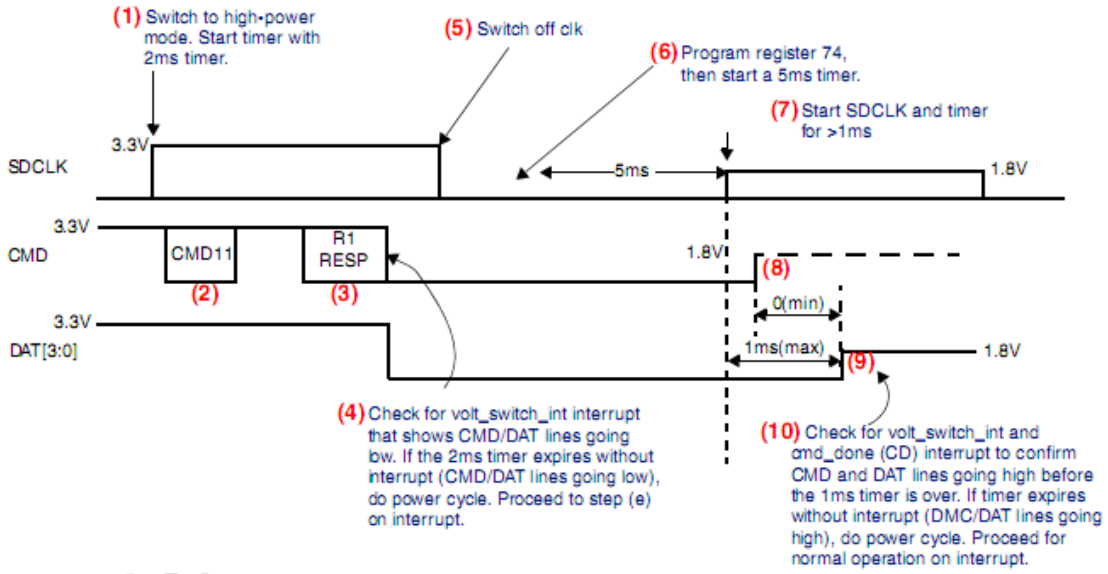


Fig. 17-17 Voltage Switch Normal Scenario

1. The host programs CLKENA—cclk_low_power register—with zero (0) for the corresponding card, which makes the host controller move to high-power mode. The application should start a timer with a recommended value of 2ms; this value of 2 ms is determined as below:Total clk required for CMD11 = 48 clks
   Total clk required for RESP R1 = 48 clks
   Maximum clk delay between MCD11 end to start of RESP1 = 60 clks
   Total = 48+48 + 60 = 160
   Minimum frequency during enumeration is 100KHz; that is, 10us
   Total time = 160 * 10us = 1600us = 1.6ms ~ 2ms
2. The host issues CMD11 to start the voltage switch sequence. Set bit 28 to 1'b1 in CMD when setting CMD11; for more information on setting bits, refer to "Boot Operation".

3. The card returns R1 response; the host controller does not generate cmd_done interrupt on receiving R1 response.
4. The card drives CMD and DAT [3:0] to low immediately after the response. The host controller generates interrupt (VOLT_SWITCH_INT) once the CMD or DAT [3:0] line goes low. The application should wait for this interrupt. If the 2ms timer expires without an interrupt (CMD/DAT lines going low), do a power cycle.
   Note: Before doing a power cycle, switch off the card clock by programming CLKENA register
   Proceed to step (5) on getting an interrupt (VOLT_SWITCH_INT).
   Note: This interrupt must be cleared once this interrupt is received. Additionally, this interrupt should not be masked during the voltage switch sequence.
   If the timer expires without interrupt (CMD/DAT lines going low), perform a power cycle. Proceed to step (5) on interrupt.
5. Program the CLKENA, cclk_enable register, with 0 for the corresponding card; the host stops supplying SDCLK.
6. Program VOLT_REG to the required values for the corresponding card. The application must program the newly-defined VOLT_REG register to assign 1 for the bit corresponding to the card number. The application should start a timer > 5ms.
7. After the 5ms timer expires, the host voltage regulator is stable. Program CLKENA, cclk_enable register, with 1 for the corresponding card; the host starts providing SDCLK at 1.8V; this can be at zero time after VOLT_REG has been programmed. When the CLKENA register is programmed, the application should start another timer > 1ms.
8. By detecting SDCLK, the card drives CMD to high at 1.8V for at least one clock and then stops driving (tri-state); CMD is triggered by the rising edge of SDCLK (SDR timing).
9. If switching to 1.8V signaling is completed successfully, the card drives DAT [3:0] to high at 1.8V for at least one clock and then stops driving (tri-state); DAT [3:0] is triggered by the rising edge of SDCLK (SDR timing). DAT[3:0] must be high within 1ms from the start of SDCLK.
10. The host controller generates a voltage switch interrupt(VOLT_SWITCH_INT) and a command done (CD) interrupt once the CMD and DAT[3:0] lines go high. The application should wait for this interrupt to confirm CMD and DAT lines going high before the 1ms timer is done.
    If the timer expires without the voltage switch interrupt (VOLT_SWITCH_INT), a power cycle should be performed. Program the CLKENA register to stop the clock for the corresponding card number. Wait for the cmd_done (CD) interrupt. Proceed for normal operation on interrupt. After the sequence is completed, the host and the card start communication in SDR12 timing.

Voltage Switch Error Scenario

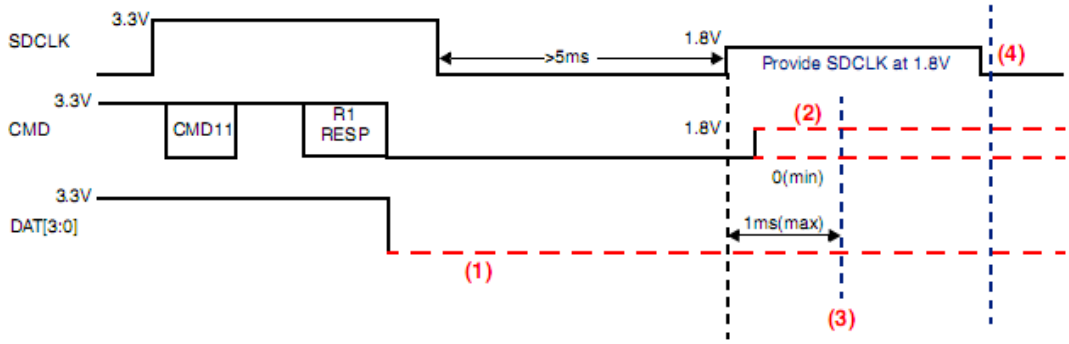


Fig. 17-18 Voltage Switch Error Scenario

1. If the interrupt (VOLT_SWITCH_INT) does not come, then the 2 ms timer should time out and a power cycle should be initiated.
   Note: Before performing a power cycle, switch off the card clock by programming CLKENA register; no cmd_done (CD) interrupt is generated.
   Additionally, if the card detects a voltage error at any point in between steps (5) and (7) in Figure 17-17, the card keeps driving DAT[3:0] to low until card power off.
2. CMD can be low or tri-state.
3. The host controller generates a voltage switch interrupt once the CMD and DAT[3:0] lines go high. The application should check for an interrupt to confirm CMD and DAT lines going high before the 1 ms timer is done. If the 1 ms timer expires without interrupt (VOLT_SWITCH_INT) and cmd_done (CD), a power cycle should be performed. Program the CLKENA register to stop SDCLK of the corresponding card. Wait for the cmd_done interrupt. Proceed for normal operation on interrupt.
4. If DAT[3:0] is low, the host drives SDCLK to low and then stops supplying the card power.
   Note: The card checks voltages of its own regulator output and host signals to ensure they are less than 2.5V. Errors are indicated by (1) and (2) in Figure 7-18.
   - If voltage switching is accepted by the card, the default speed is SDR12.
   - Command Done is given:
     - If voltage switching is properly done, CMD and DAT line goes high.
     - If switching is not complete, the 1ms timer expires, and the card clk is switched off.

   Note: No other CMD should be driven before the voltage switching operation is completed and Command Done is received.
   - The application should use CMD6 to check and select the particular function; the function appropriate-speed should be selected. After the function switches, the application should program the correct value in the CLKDIV register, depending on the function chosen. Additionally, if Function 0x4 of the Access mode is chosen—that is, DDR50, then the application should also program 1'b1 in DDR_REG for the card number that has been selected for DDR50 mode.

DDR Operation

DDR programming should be done only after the voltage switch operation has completed.The following outlines the steps for the DDR programming sequence:

1. Once the voltage switch operation is complete, the user must program VOLT_REG to the required values for the corresponding card.
   - To start a card to work in DDR mode, the application must program a

bit of the newly defined VOLT_REG[31:16] register with a value of 1'b1.
- The bit that the user programs depends on which card is to be accessed in DDR mode.

2. To move back to SDR mode, a power cycle should be run on the card—putting the card in SDR12 mode—and only then should VOLT_REG[31:16] be set back to 1'b0 for the appropriate card.

Reset Command/Moving from DDR50 to SDR12

To reset the mode of operation from DDR50 to SDR12, the following sequence of operations has to be done by the application:

1. Issue CMD0.
   When CMD0 is received, the card changes from DDR50 to SDR12.
2. Program the CLKDIV register with an appropriate value.
3. Set DDR_REGto 0.

Note: The VOLT_REG register should not be programmed to 0 while switching from DDR50 to SDR12, since the card is still operating in 1.8V mode after receiving CMD0.

17.7.5 H/W Reset Operation

When the RST_n signal goes low, the card enters a pre-idle state from any state other than the inactive state.

H/W Reset Programming Sequence

The following outlines the steps for the H/W reset programming sequence:

1. Program CMD12 to end any transfer in process.
2. Wait for DTO, even if no response is sent back by the card.
3. Set the following resets:
   - DMA reset– CTRL[2]
   - FIFO reset – CTRL[1] bits

Note: The above steps are required only if a transfer is in process.

4. Program the CARD_RESET register with a value of 0; this can be done at any time when the card is connected to the controller. This programming asserts the RST_n signal and resets the card.
5. Wait for minimum of 1 µs or cclk_in period, which ever is greater
6. After a minimum of 1 µs, the application should program a value of 0 into the CARD_RESET register. This de-asserts the RST_n signal and takes the card out of reset.
7. The application can program a new CMD only after a minimum of 200 µs after the de-assertion of the RST_n signal, as per the MMC 4.41 standard.
   Note: For backward compatibility, the RST_n signal is temporarily disabled in the card by default. The host may need to set the signal as either permanently enabled or permanently disabled before it uses the card.

Chapter 18 Embedded SRAM

18.1 Overview

The Embedded SRAM is the AXI slave device, which support read and write access to provide system fast access data storage.

18.1.1 Features supported

- Provide 64KB access space
- Support security and non-security access
- Security or non-security space is software programmable
- Security space is 0KB,4KB,8KB,12K ,16K, 20K, 24K, 28K, 32K, 36K, 40K, 44K, 48K, 52K, 56K, 60K, 64K (the whole memory space)
- Support 64bit AXI bus

18.1.2 Features not supported

- Don't support AXI lock transaction
- Don't support AXI exclusive transaction
- Don't support AXI cache function
- Don't support AXI protection function

18.2 Block Diagram

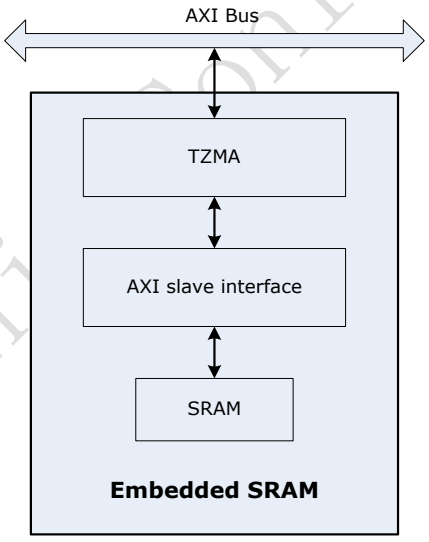


Fig. 18-1 Embedded SRAM block diagram

18.3 Function Description

18.3.1 TZMA

Please refer to 5.3.3 for TZMA functional description.

18.3.2 AXI slave interface

The AXI slave interface is bridge which translate AXI bus access to SRAM interface.

18.3.3 Embedded SRAM access path

The Embedded SRAM can only be accessed by Cortex-A9 and DMAC0.

Chapter 19 GPU (Graphics Process Unit)

19.1 Overview

The gpu is a hardware accelerator for 2D and 3D graphics systems. Its triangle rate can be 30 Mtris/s , pixel rate can be 1Gpix/s.
The GPU supports the following graphics standards:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1

The GPU consists of :

- Four Pixel Processors (PPs)
- a geometry Processor (GP)
- a Level2 Cache controller (L2)
- a Memory Management Unit (MMU) for each GP and PP included in the GPU

The GPU contain a 64-bit APB bus and an 128-bit AXI bus. CPU config GPU through APB bus, GPU read and write data through AXI bus.

For detailed information about GPU(graphics process unit), please refer to **RK30xx GPU.pdf**。

Chapter 20 VCODEC (Video encoder and decoder Unit)

20.1 Overview

VCODEC is composed of video decoder and video encoder. VCODEC is connected to VCODEC_AHB bus through an AHB slave and VCODEC_AXI through an AXI master. The register setting is configured through the AHB slave interface and the stream data is read and written through the AXI master interface.

Video decoder and video encoder share many internal memories and they also share the bus master and slave interfaces. So it prevents video decoder and video encoder from working simultaneously. Encoding and decoding now have to time-share the memory resource on a frame by frame basis.

For detailed information about VCODEC, please refer to **RK30xx VCODEC.pdf**。

Chapter 21 IPP (Image Post Processor)

21.1 Overview

Image Post Processing (IPP) is used doing image scaler, deinterlace and rotation. Each processing can be done independently or combined with others.

21.1.1 Features

- **Input data format**
  - RGB888: 16x16 to 8191x8191
  - RGB565:16x16 to 8191x8191
  - YUV422/YUV420: 16x16 to 8190x8190
  - YUV444:16x16 to 8190x8190

- **Pre Scaler**
  - Down-scaling
  - Integral scaling ratio, from 1/8 to 1/2
  - Linear filter
  - Deinterlace

- **Post Scaler**
  - Down-scaling and up-scaling
  - Arbitrary non-integer scaling ratio, from 1/2 to 4
  - 4-tap vertical, 2-tap horizontal filter
  - The max output image width of post scaler is 4096

- **Rotation**
  - 90-degree, 180-degree,270-degree ratation
  - x-mirror, y-mirror

For detailed information about IPP(Image Post Processor), please refer to **RK30xx IPP.pdf**。

Chapter 22 LCDC

22.1 Overview

LCD Controller is the display interface from memory frame buffer to display device(LCD panel or TV set). LCDC is connected to LCDC_AHB bus through an AHB slave and DISP_AXI bus through an AXI master. The register setting is configured through the AHB slave interface and the display frame data is read through the AXI master interface.

There are two symmetrical LCDCs in SOC for dual panel display application.

22.1.1 Features

- **Display interface**
  - Parallel RGB LCD Interface: 24-bit(RGB888), 18-bit(RGB666), 15-bit(RGB565)
  - Serial RGB LCD Interface: 3x8-bit(RGB delta support), 3x8-bit + dummy, 16-bit + 8-bit
  - MCU LCD interface: i-8080(up to 24-bit RGB), Hold/Auto/Bypass modes
  - TV Interface: ITU-R BT.656(8-bit, 480i/576i/1080i)
- **Display process**
  - Background layer: programmable 24-bit color
  - Win0 layer:
    - RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, AYCbCr
    - Maximum resolution is 1920x1080, support virtual display
    - 1/8 to 8 scaling-down and scaling-up engine
    - 256 level alpha blending
    - Transparency color key
    - 3D display support
  - Win1 layer:
    - RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, AYCbCr
    - Maximum resolution is 1920x1080, support virtual display
    - 1/8 to 8 scaling-down and scaling-up engine
    - 256 level alpha blending
    - Transparency color key
  - Win2 layer:
    - 1/2/4/8bpp, RGB888, ARGB888, RGB565
    - Support virtual display
    - 256 level alpha blending
    - Transparency color key
  - Hardware cursor:
    - 2bpp
    - Two size mode: 32x32 and 64x64
    - 3-color and transparentmode
    - 2-color + transparency + tran_invert mode
    - 16 level alpha blending

◆ **Others**

- 3 x 256 x 8 bits display LUTs
- Win0 layer and Win1 layer overlay exchangeable
- De-flicker support for interlace output
- YCbCr2RGB(rec601-mpeg/ rec601-jpeg/rec709) and RGB2YCbCr modules
- Replication(16-bit to 24-bit) and Dithering(24-bit to 16-bit/18-bit)
- Blank and black display
- Standby mode
- Auto dynamic power control

22.2 Block Diagram

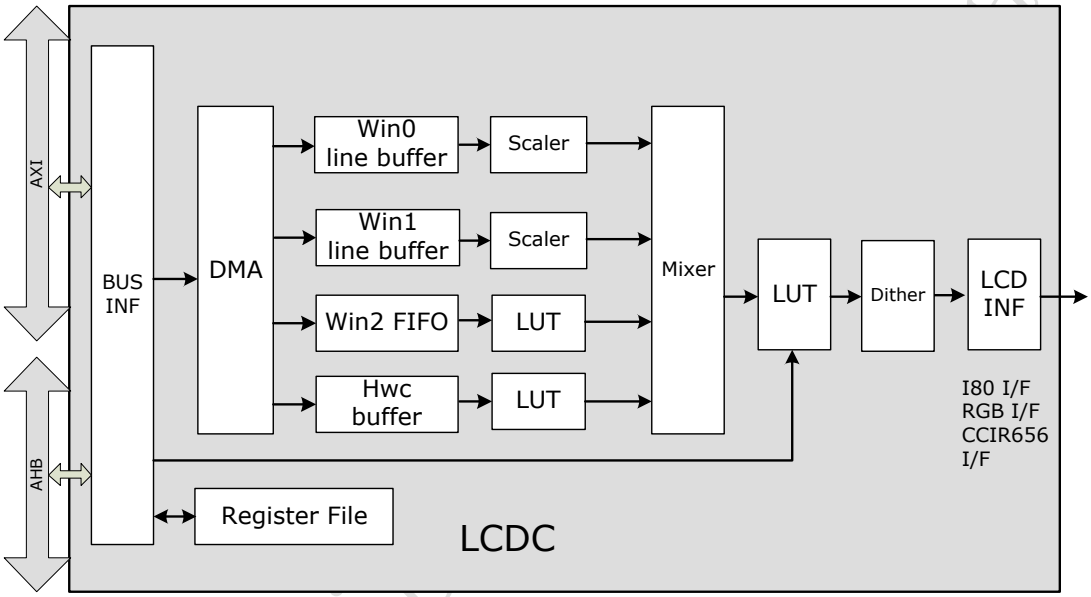


Fig. 22-1LCDC Block Diagram

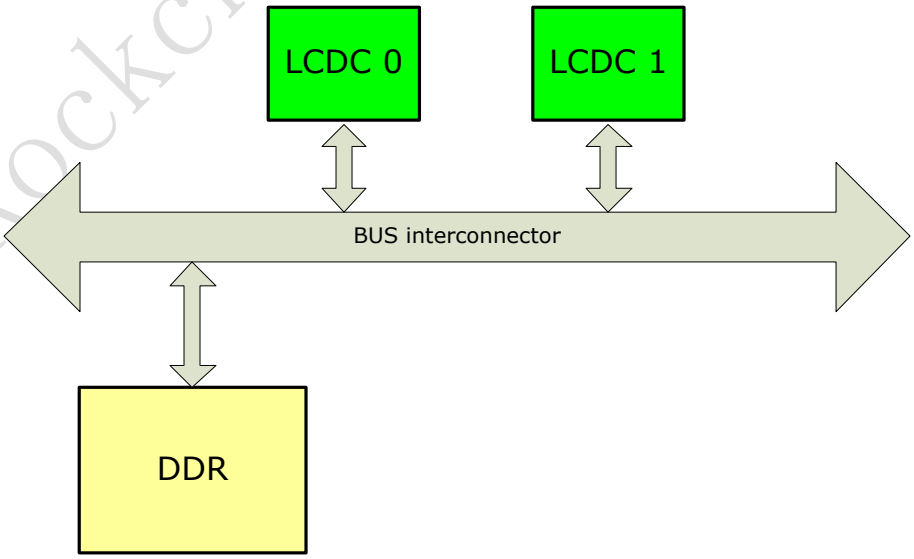


Fig. 22-2LCDC Dual LCDCs in SOC

22.3 Function Description

22.3.1 Data Format

LCDC master read the frame data from the frame buffer in the system memory (SDR or DDR). There are total 10 formats supported in three layers.

- Win0: RGB888,ARGB888, RGB565, YCbCr422, YCbCr420, YCbCr444
- Win1: RGB888, ARGB888, RGB565, YCbCr422, YCbCr420, YCbCr444
- Win2: 1bpp, 2bpp, 4bpp, 8bpp, RGB888, ARGB888, RGB565
- Hwc: 2bpp

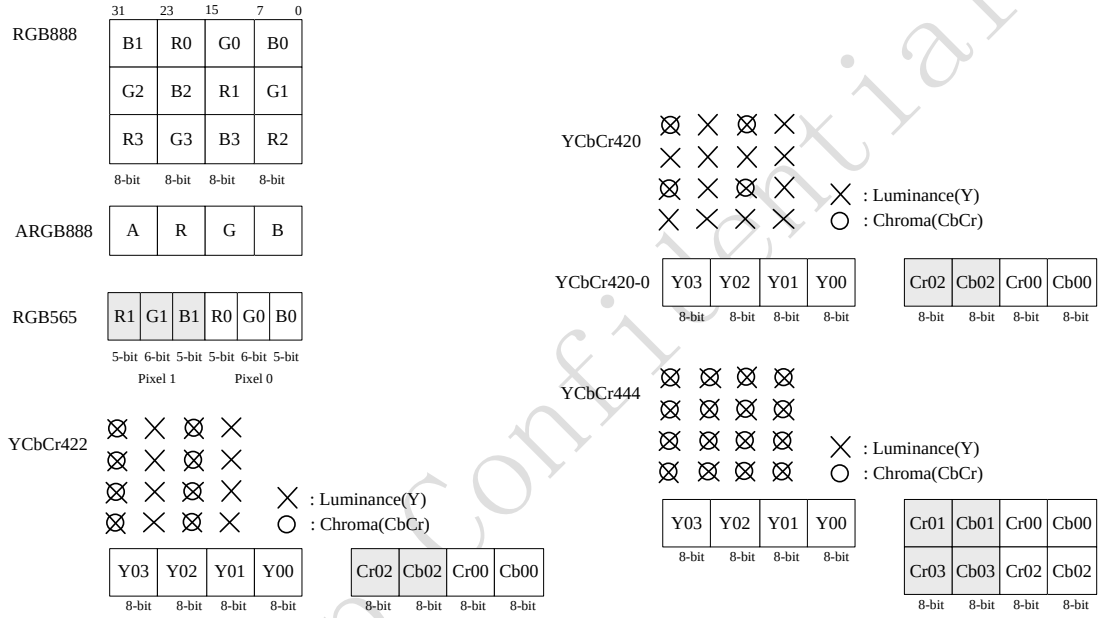
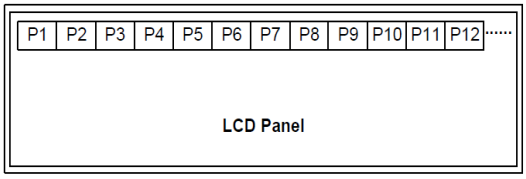


Fig. 22-3LCDC Frame Buffer Data Format

	D[31:24]	D[23:16]	D[15:8]	D[7:0]
000H	P1	P2	P3	P4
004H	P5	P6	P7	P8
008H	P9	P10	P11	P12
...				



	D[31:28]	D[27:24]	D[23:20]	D[19:16]	D[15:12]	D[11:8]	D[7:4]	D[3:0]
000H	P1	P2	P3	P4	P5	P6	P7	P8
004H	P9	P10	P11	P12	P13	P14	P15	P16
008H	P17	P18	P19	P20	P21	P22	P23	P24
...								

Fig. 22-4LCDC Win2 Palette (8bpp/4bpp)

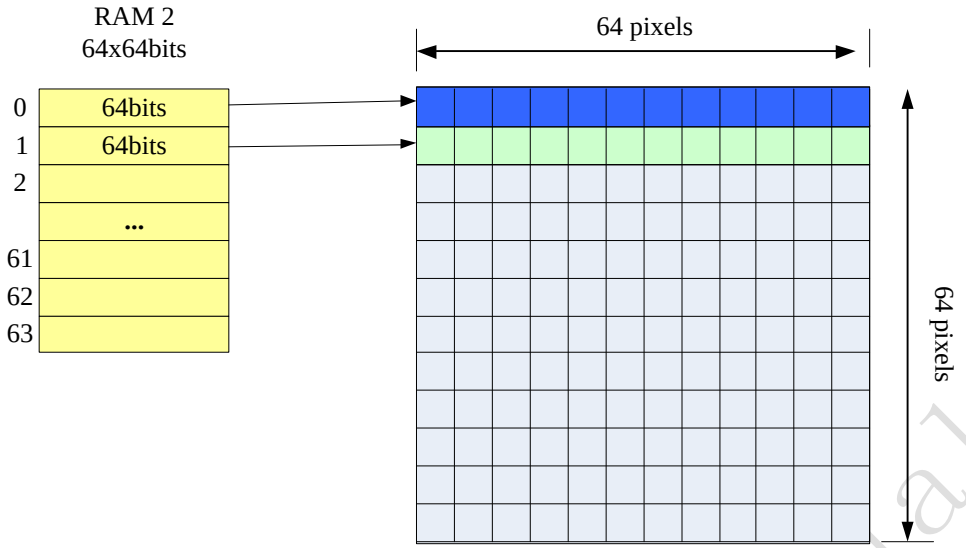


Fig. 22-5LCDC Hwc Data Format

Table 22-1Hwc 3-color Transparency Mode

Data[1:0]	Display Pixel color
00	Cusor color 0
01	Cusor color 1
10	Cusor color 2
11	Transparent

Table 22-2Hwc 2-color Transparency Mode

Data[1:0]	Display Pixel color
00	Cusor color 0
01	Cusor color 1
10	Transparent
11	Inverted-Transparent

Data SWAP function

There are several data-swap modes for flexible application. The register is LCDC_SYS_CTRL1[29:19].

All the data swap types are in the following table.

Table 22-3LCDC Data Swap of Win0, Win1 and Win2

Data-swap	*RB swap*	*Alpha swap*	*Y-M8 swap*	*CbCr swap*	Big-endian/Little-endian
Win0	yes	yes	yes	yes	No
Win1	yes	yes	yes	yes	No
Win2	yes	yes	No	No	yes

22.3.2 Virtual display

Virtual display is supported in Win0, Win1 and Win2. The active image is part of the virtual (original) image in frame buffer memory. The virtual width is indicated by setting WIN0/WIN1/WIN2_VIR_STRIDE for different data format.

The virtual stride should bemultiples of word (32-bit). That means dummy bytes in the end of virtual line if the real pixels are not 32-bit aligned.

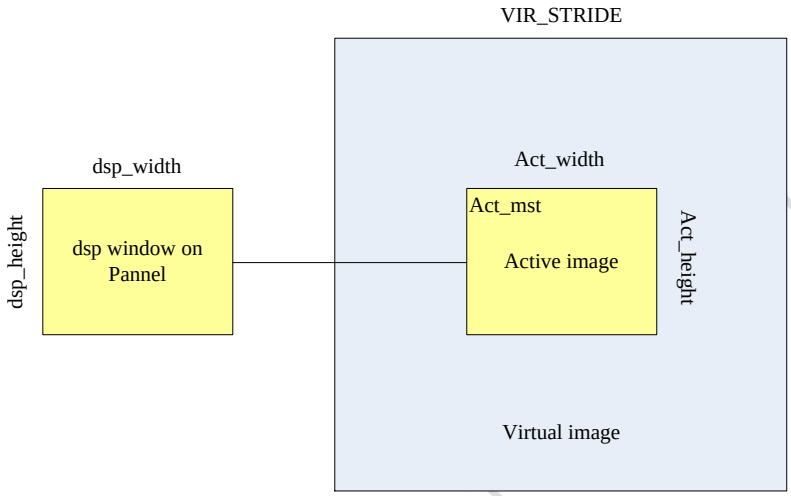


Fig. 22-6LCDC Virtual Display Mode

22.3.3 Scaling

The scaling operation is the imageresizing processof data transfer from the frame buffer memory toLCD panel or TV set.

The scaling units in layer Graphic (Win1) and layer Video (Win0) are independent. So Win0 and Win1's scaling can be enabled simultaneously.

Horizontal and vertical scaling factor should be set according the window scaling ratio.

1. Scaling factor

Because the chroma data may have different sampling rate with Luma data in the memory format of YCbCr422/YCbCr420. The scaling factor of Win0 has two couples of factor registers:

LCDC_WIN0_SCL_FACTOR_Y/LCDC_WIN0_SCL_FACTOR_CBR

Software calculates the scaling factor value using thefollowing equations:

$$y\_rgb\_vertical\_factor = \left(\frac{LCDC\_WIN0\_ACT\_INFO[31:16]}{LCDC\_WIN0\_DSP\_INFO[31:16]}\right) \times 2^{12}$$

$$y\_rgb\_horizontal\_factor = (\frac{LCDC\_WIN0\_ACT\_INFO[15:0]}{LCDC\_WIN0\_DSP\_INFO[15:0]}) \times 2^{12}$$

$$yuv422\_yuv444\_Cbr\_vertical\_factor = (\frac{LCDC\_WIN0\_ACT\_INFO[31:16]}{LCDC\_WIN0\_DSP\_INFO[31:16]}) \times 2^{12}$$

$$yuv420\_Cbr\_vertical\_factor = (\frac{LCDC\_WIN0\_ACT\_INFO[31:16]/2}{LCDC\_WIN0\_DSP\_INFO[31:16]}) \times 2^{12}$$

$$yuv444\_Cbr\_horizontal\_factor = (\frac{LCDC\_WIN0\_ACT\_INFO[15:0]}{LCDC\_WIN0\_DSP\_INFO[15:0]}) \times 2^{12}$$

$$yuv422\_yuv420\_Cbr\_horizontal\_factor = (\frac{LCDC\_WIN0\_ACT\_INFO[15:0]/2}{LCDC\_WIN0\_DSP\_INFO[15:0]}) \times 2^{12}$$

2. Scaling start point offset

The x and y start point of the generated pixels can be adjusted, the offset value is in the range of 0 to 0.99.

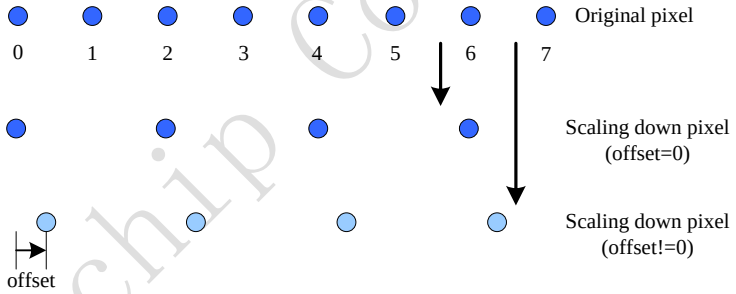


Fig. 22-7LCDC Scaling Down Offset

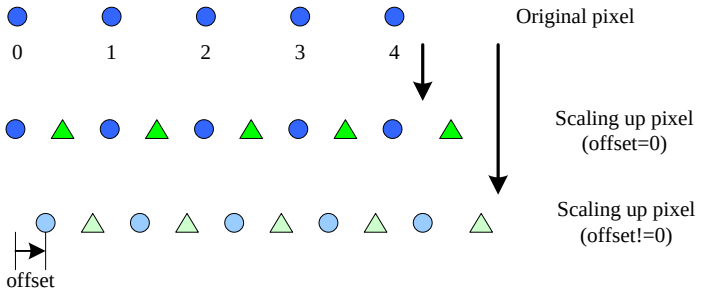


Fig. 22-8LCDC Scaling Up Offset

Table 22-4LCDC Scaling Start Point Offset Registers

scaling down/up start point offset	*Offset variable*	**Register**
Win0 YRGB vertical scaling offset	Win0_YRGB_vscl_offset	Win0_SCL_OFFSET [32:24]
Win0 YRGB horizontal scaling offset	Win0_YRGB_hscl_offset	Win0_SCL_OFFSET [23:16]
Win0 Cbr vertical scaling offset	Win0_CBR_vscl_offset	Win0_SCL_OFFSET [15:8]
Win0 Cbr horizontal scaling offset	Win0_CBR_hscl_offset	Win0_SCL_OFFSET [7:0]

3. De-flicker (Interlace vertical filtering)

It is necessary to display a non-interlaced video signal on an interlaced display (such as TV set). Thus some from of"non-interlaced-to-interlaced conversion" may be required.

The easiest approach is to throw away every other active scan line in each non-interlaced frame. Although the cost is minimal, there are problems with this approach. If there is a sharp vertical transition of color or intensity. It will flicker at one-half the refresh rate.

A better solution is to use two lines of non-interlaced data to generation one line of interlace data. Fast vertical transitions are smoothed out over several interlace lines.

The vertical filtering of two non-interlaced lines can be done by enabling the vertical scaling offset dynamic change in different field (even/odd). The dynamic change value of scaling offset is half of the scaling factor. You should enable the scaling down vertical offset in scaling down mode; enable the scaling up vertical offset in scaling up mode, or one of it in no-scaling mode.

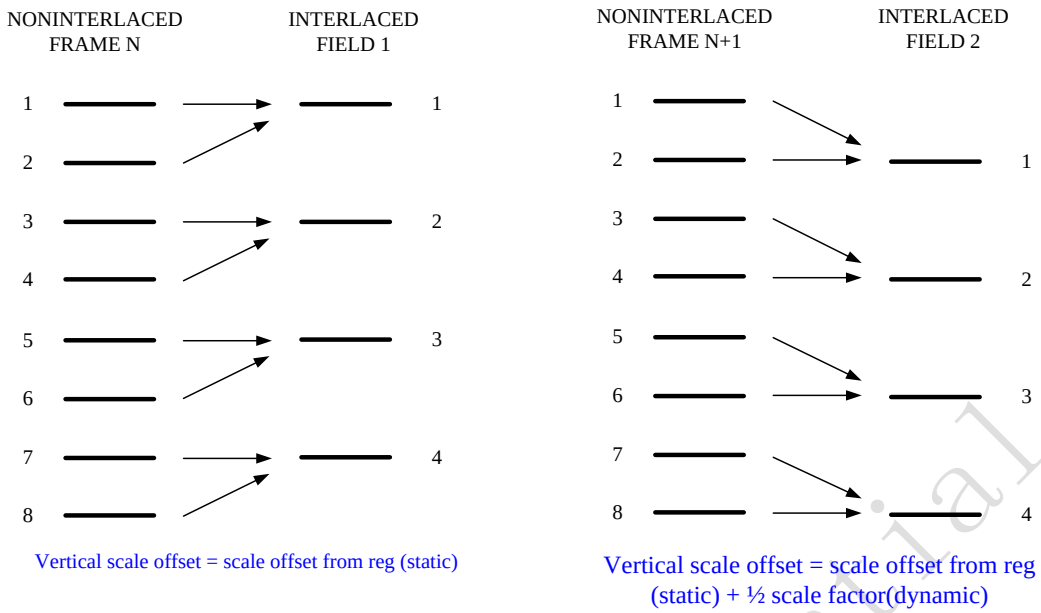


Fig. 22-9LCDC Interlace Vertical Filtering

22.3.4 3D-display

3D-display is supported in Win0 layer. There are two modes: MIX mode and Interleave mode.

3D-display merges two source images into one screen. Both images have same data format, source information (win0_vir_stride, win0_act_width, win0_act_height) and display information (dsp_win0_width, dsp_win0_height, dsp_win0_st). The second image's start address should be set in 3D-display mode.

Interlace display is not support in win0 3D-display mode.

1. MIX mode

There are three types of MIX 3D-display: R-GB color mix, G-RB color mix and B-RG color mix.

The scaling for two images is done individually if necessary. The scaling factor is same for two images.

For RGB format, the factor scaling setting is the same with win0 non-3D mode. For YUV format, image reading data would be duplicated to YUV444 format before RGB color mix, so the cbr scaling factor setting should be the same with yrgb scaling factor.

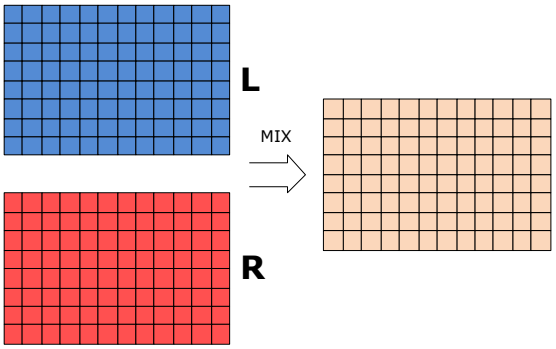


Fig. 22-10LCDC Mix 3D Display

2. Interlace mode

There are two types of interlace 3D-display: H-interlace and V-interlace.

The scaling for two images is done individually if necessary. The scaling factor is same for two images.

In H-interlace mode, the display width of each image is half of the final display width. So the display width value for horizontal scaling factor calculation should be halved.

In V-interlace mode, the display height of each image is half of the final display height. So the display height value for vertical scaling factor calculation should be halved.

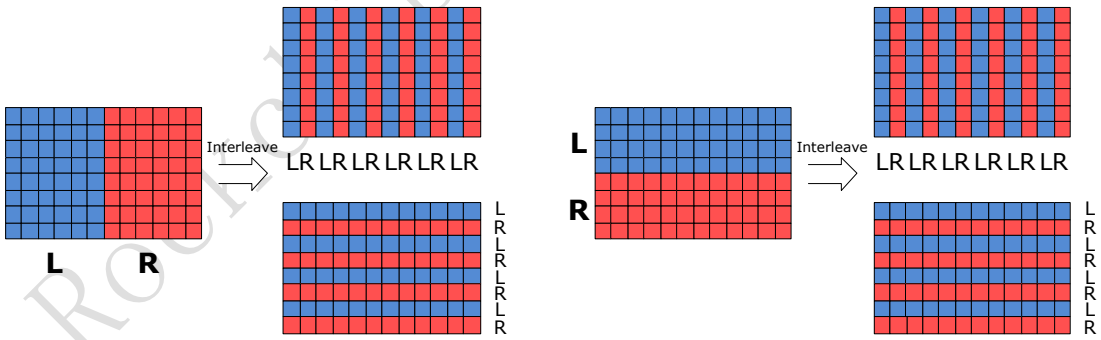


Fig. 22-11LCDC Interleave 3D Display

22.3.5 Overlay

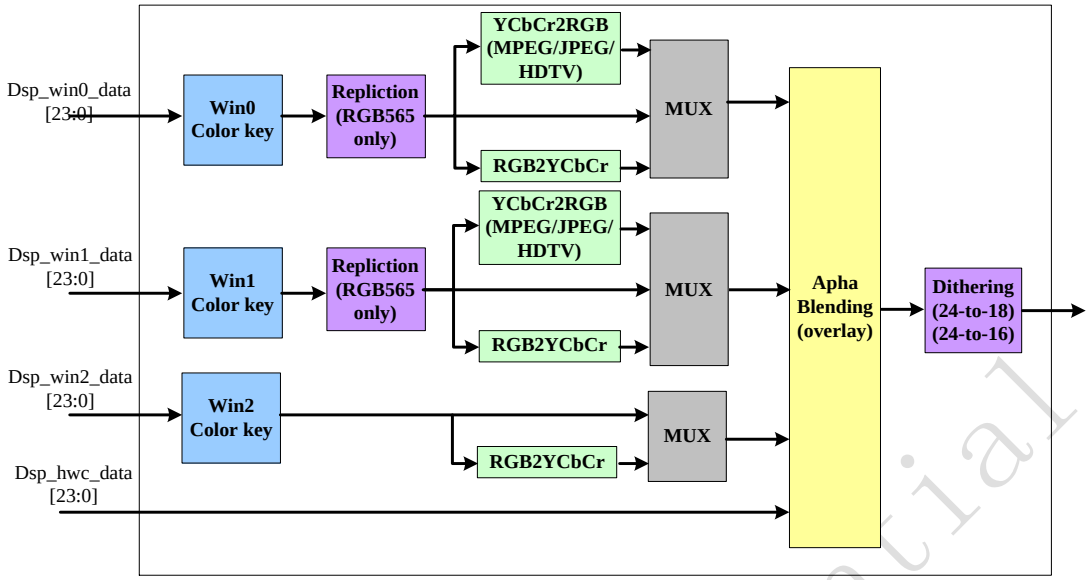


Fig. 22-12LCDC Overlay Block Diagram

1. Overlay display

There are totally 5 layers for overlay display: Background, Win0, Win1, Win2 and Hwc.

The background is a programmable solid color layer, which is always the bottom of the display screen.

Hwc is a 32x32 or 64x64 3-LUT-colors layer, which is on the top layer of the display screen.

The two middle layers are Win0 and Win1. Win1 is on the top of Win0 in default setting, setting LCDC_DSP_CTRL_REG0[8] to '1' can let Win0 be on the top of Win1.

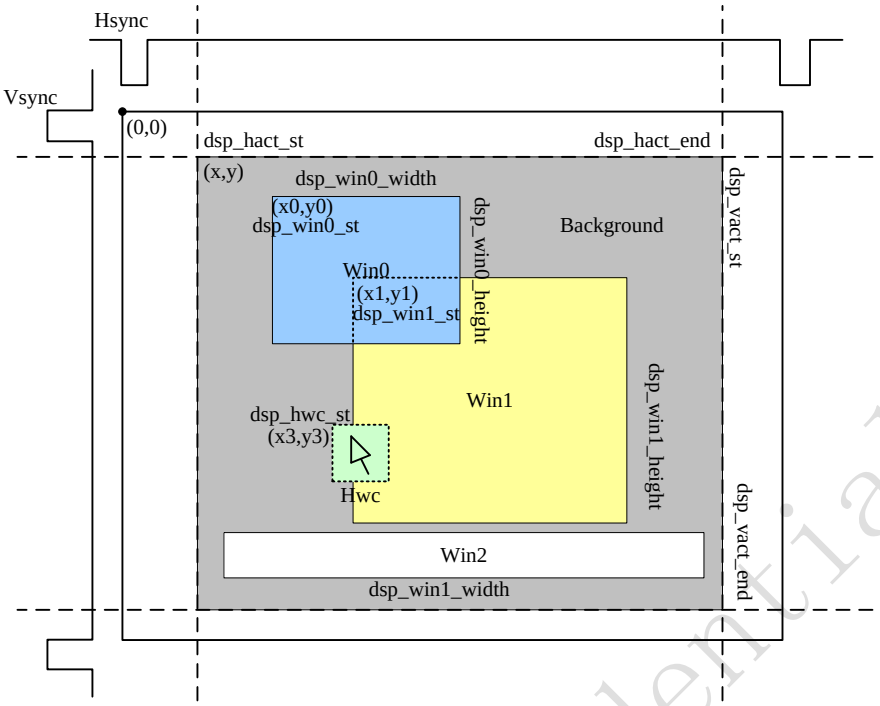


Fig. 22-13LCDC Overlay Display

2. Transparency color key

There are three transparency color keys Win0/Win1/Win2 layer.The two transparency color keycan be active at the same time.

The pixel color value is compared to the transparency color key before final display. The transparency color key value defines the pixel data considered asthe transparent pixel. The pixel values with the source color key value are pixels not visibleon the screen, and the under layer pixel values or solid background color are visible.

Transparency color key is done after the scaling module and before the YCbCr2RGB color space converter. So transparency color key can only be used in non-scaling mode.

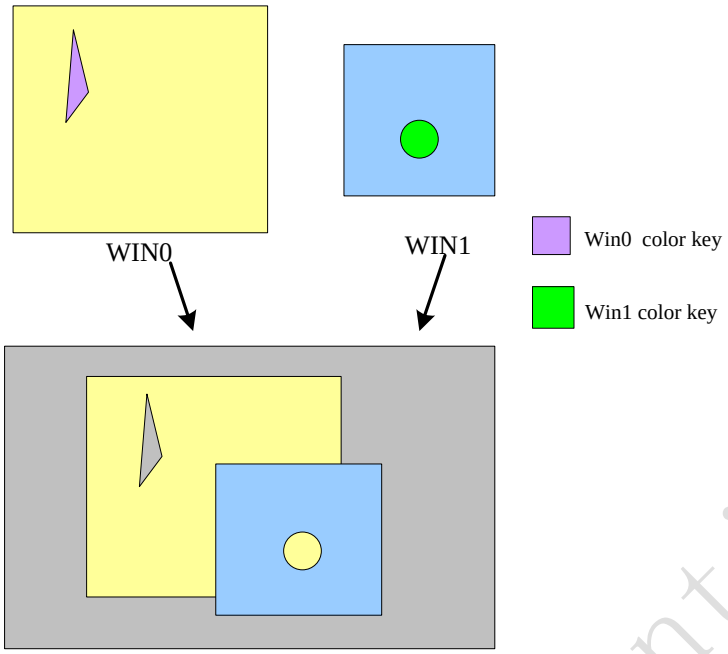


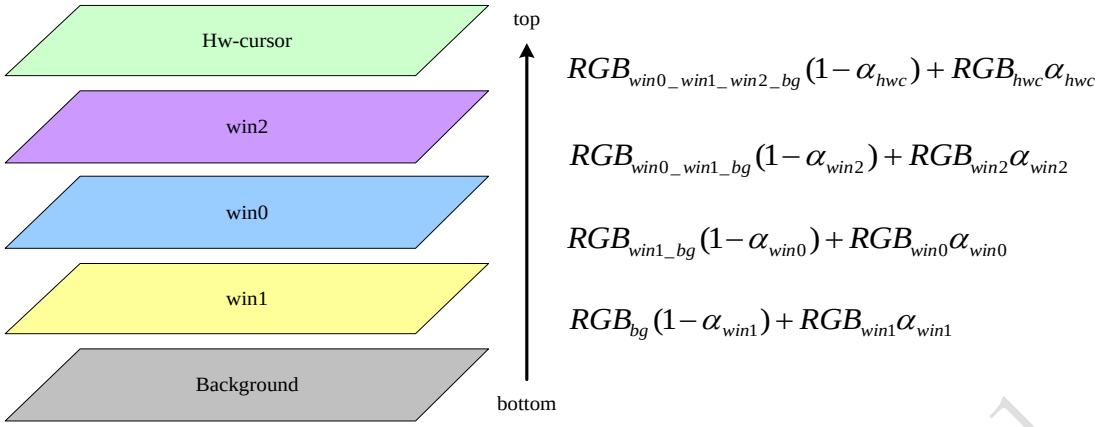
Fig. 22-14LCDC Transparency Color Key

3. Alpha Blending

There are four alpha values for blending between five overlay layers: alpha_win0[7:0], alpha_win1[7:0], alpha_win2[7:0], alpha_hwc[3:0].

Two blending modes are supported. One is per-pixel (ARGB) mode; the other is user-specified mode. In ARGB mode, the alpha value is in the ARGB data (Win0 Win1 and Win2 normal mode only). In user-specified mode, the alpha value comes from the register (LCDC_ALPHA_CTRL[31: 4]).

In HWC layer, if the data of the hwc pixel is 2'b00, then this pixel is transparent (alpha = 0), regardless the alpha setting of alpha value.



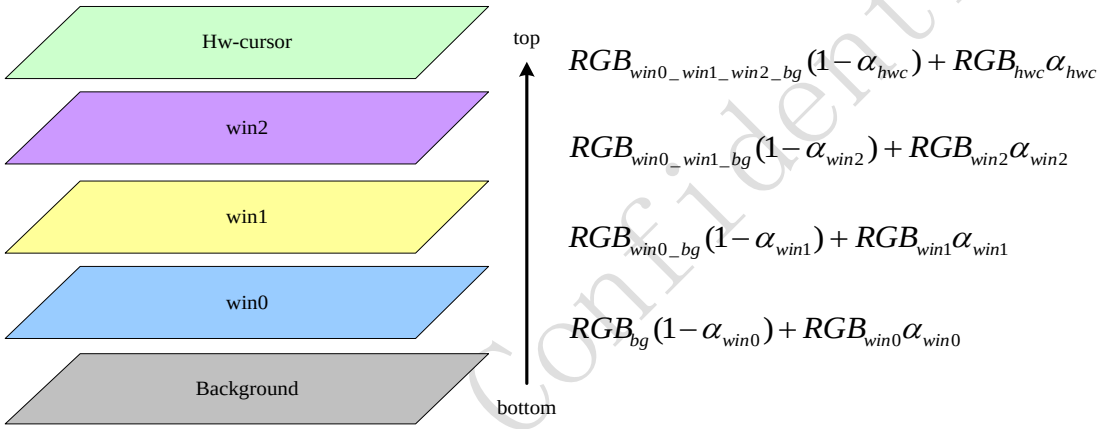


Fig. 22-15LCDC Alpha blending

4. Replication and Dithering

If the interface data bus is wider than the pixelformat size, by programming the pixel components replication active/inactive, the MSB is replicated tothe LSB of the interface data bus or the LSB is filled with 0s.

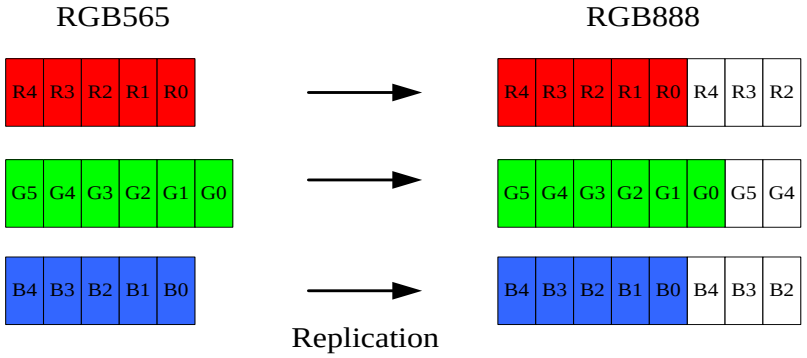


Fig. 22-16LCDC Replicaiton

Dithering is an intentionally applied form of noise, used to randomize quantization error, thereby preventing large-scaling patterns such as "banding".

The pixel values are used by Dithering logic todisplay the data in a lower color depth on the LCD panel. The Dithering algorithm isbased on the (x,y) pixel position and the value of removed bits. The picture qualityis improved when enabling the Dithering logic. When Dithering is notenabled, the MSBs of the pixel color components are output on the interface data bus if theinterface data bus is smaller than the pixel format size.

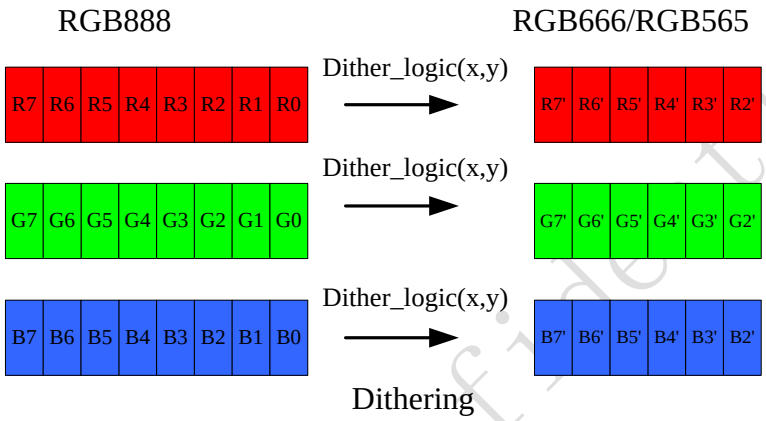


Fig. 22-17LCDC Dithering

22.4 Register Description

22.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
LCDC_SYS_CTRL0	0x0000	W	0x00000000	LCDC system control register0
LCDC_SYS_CTRL1	0x0004	W	0x00000000	LCDC system control register1
LCDC_DSP_CTRL0	0x0008	W	0x00000000	Display control register0
LCDC_DSP_CTRL1	0x000c	W	0x00000000	Display control register1
LCDC_INT_STATUS	0x0010	W	0x00000000	Interrupt status register
LCDC_MCU_CTRL	0x0014	W	0x00711c08	MCU mode control register
LCDC_BLEND_CTRL	0x0018	W	0x00000000	Blending control register
LCDC_WIN0_COLOR_KEY	0x001c	W	0x00000000	Win0 color key register
LCDC_WIN1_COLOR_KEY	0x0020	W	0x00000000	Win1 color key register

Name	Offset	Size	Reset Value	Description
LCDC_WIN2_COLOR_KEY	0x0024	W	0x00000000	Win2 color key register
LCDC_WIN0_YRGB_MST0	0x0028	W	0x00000000	Win0 YRGB memory start address 0
LCDC_WIN0_CBR_MST0	0x002c	W	0x00000000	Win0 Cbr memory start address 0
LCDC_WIN0_YRGB_MST1	0x0030	W	0x00000000	Win0 YRGB memory start address 1
LCDC_WIN0_CBR_MST1	0x0034	W	0x00000000	Win0 Cbr memory start address 1
LCDC_WIN0_VIR	0x0038	W	0x00000140	Win0 virtual display width
LCDC_WIN0_ACT_INFO	0x003c	W	0x00ef013f	Win0 active window width/height
LCDC_WIN0_DSP_INFO	0x0040	W	0x00ef013f	Win0 display width/height on panel
LCDC_WIN0_DSP_ST	0x0044	W	0x00000000	Win0 display start point on panel
LCDC_WIN0_SCL_FACTOR_YRGB	0x0048	W	0x10001000	Win0 YRGB scaling factor
LCDC_WIN0_SCL_FACTOR_CBR	0x004c	W	0x10001000	Win0 Cbr scaling factor
LCDC_WIN0_SCL_OFFSET	0x0050	W	0x00000000	Win0 scaling start point offset
LCDC_WIN1_YRGB_MST	0x0054	W	0x00000000	Win1 YRGB memory start address
LCDC_WIN1_CBR_MST	0x0058	W	0x00000000	Win1 Cbr memory start address
LCDC_WIN1_VIR	0x005c	W	0x00000140	Win1 virtual display width
LCDC_WIN1_ACT_INFO	0x0060	W	0x00ef013f	Win1 active window width/height
LCDC_WIN1_DSP_INFO	0x0064	W	0x00ef013f	Win1 display width/height on panel
LCDC_WIN1_DSP_ST	0x0068	W	0x00000000	Win1 display start point on panel
LCDC_WIN1_SCL_FACTOR_YRGB	0x006c	W	0x10001000	Win1 YRGB scaling factor
LCDC_WIN1_SCL_FACTOR_CBR	0x0070	W	0x10001000	Win1 Cbr scaling factor

Name	Offset	Size	Reset Value	Description
LCDC_WIN1_SCL_OFFSET	0x0074	W	0x00000000	Win1 scaling start point offset
LCDC_WIN2_MST	0x0078	W	0x00000000	Win2 memory start address
LCDC_WIN2_VIR	0x007c	W	0x00000140	Win2 virtual display width
LCDC_WIN2_DSP_INFO	0x0080	W	0x00ef013f	Win2 display width/height on panel
LCDC_WIN2_DSP_ST	0x0084	W	0x00000000	Win2 display start point on panel
LCDC_HWC_MST	0x0088	W	0x00000000	Hwc memory start address
LCDC_HWC_DSP_ST	0x008c	W	0x00000000	Hwc display start point on panel
LCDC_HWC_COLOR_LUT0	0x0090	W	0x00000000	Hwc LUT color 0
LCDC_HWC_COLOR_LUT1	0x0094	W	0x00000000	Hwc LUT color 1
LCDC_HWC_COLOR_LUT2	0x0098	W	0x00000000	Hwc LUT color 2
LCDC_DSP_HTOTAL_HS_END	0x009c	W	0x014a000a	Panel scanning horizontal width and hsync pulse end
LCDC_DSP_HACT_ST_END	0x00a0	W	0x014a000a	Panel active horizontal scanning start point and end point
LCDC_DSP_VTOTAL_VS_END	0x00a4	W	0x00fa000a	Panel scanning vertical height and vsync pulse end point
LCDC_DSP_VACT_ST_END	0x00a8	W	0x00fa000a	Panel active vertical scanning start point and end point
LCDC_DSP_VS_ST_END_F1	0x00ac	W	0x00000000	Vsync start point and end point of filed1
LCDC_DSP_VACT_ST_END_F1	0x00b0	W	0x00000000	Vertical active start point and end point of filed1
LCDC_REG_LOAD_EN	0x00c0	W	0x00000000	Register config done flag
LCDC_MCU_BYPASS_WPORT	0x0100	W	0x00000000	LCDC MCU bypass mode data write port
LCDC_MCU_BYPASS_RPORT	0x0200	W	0x00000000	LCDC MCU bypass mode data read port

Name	Offset	Size	Reset Value	Description
LCDC_Win2_LUT_ADDR	0x0400	W	0x00000000	LCDC win2 lut bus access address map
LCDC_DSP_LUT_ADDR	0x0800	W	0x00000000	LCDC dsp lut bus access address map

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

22.4.2 Detail Register Description

LCDC_SYS_CTRL0
Address: Operational Base + offset (0x0000)
LCDC system control register0

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2	RW	0x0	hwc_load_en hardware cursor data reload enable 0 : disable 1 : enable *Setting this bit would reload the Hwc data. It would be auto cleared after reload finish.
1	RW	0x0	lcdc_standby_mode LCDC standby mode Writing "1" to turn LCDC into standby mode, all the layer would disable and the data transfer from frame buffer memory would stop at the end of current frame. The output would be blank. When writing "0" to this bit, standby mode would disable and the LCDC go back to work immediately. 0: disable 1: enable * Black display is recommended before setting standby mode enable.
0	RW	0x0	lcdc_dma_stop_mode LCDC DMA stop mode 0 : disable 1 : enable * If DMA is working, the stop mode would not be active until current bus transfer is finished.

LCDC_SYS_CTRL1
Address: Operational Base + offset (0x0004)
LCDC system control register1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	RW	0x0	dsp_lut_en Display LUT ram enable 0: disable 1: enable *This bit should be '0' when CPU updates the LUT, and should be '1' when Display LUT mode enable.
30	RW	0x0	win2_lut_ram_en Win2 LUT ram enable 0: disable 1: enable *This bit should be '0' when CPU updates the LUT, and should be '1' when Win2 LUT mode enable.
29	RW	0x0	win2_endian_sel Win2 8pp palette data Big-endian/ Little-endian select 0: Big-endian 1: Little-endian
28	RW	0x0	win2_alpha_swap Win2 RGB alpha swap 0: ARGB 1: RGBA
27	RW	0x0	win2_rb_swap Win2 RGB Red and Blue swap 0: RGB 1: BGR
26	RW	0x0	win1_uv_swap Win1 CbCr swap 0: CrCb 1: CbCr
25	RW	0x0	win1_y8_swap Win1 Y middle 8-bit swap 0: Y3Y2Y1Y0 1: Y3Y1Y2Y0
24	RW	0x0	win1_alpha_swap Win1 RGB alpha swap 0: ARGB 1: RGBA
23	RW	0x0	win1_rb_swap Win1 RGB Red and Blue swap 0: RGB 1: BGR
22	RW	0x0	win0_uv_swap Win0 CbCr swap 0: CrCb 1: CbCr
21	RW	0x0	win0_y8_swap Win0 Y middle 8-bit swap 0: Y3Y2Y1Y0 1: Y3Y1Y2Y0

Bit	Attr	Reset Value	Description
20	RW	0x0	win0_alpha_swap Win0 RGB alpha swap 0: ARGB 1: RGBA
19	RW	0x0	win0_rb_swap Win0 RGB Red and Blue swap 0: RGB 1: BGR
18:16	RW	0x0	win0_3d_mode Win0 3D mode 000: MIX mode，R-GB 001: MIX mode，G-RB 010: MIX mode，B-RG 100: Interleave mode, horizontal 101: Interleave mode, Vertical
15	RW	0x0	win0_3d_en Win0 3D mode en 0: Win0 3D-display mode disable 1: Win0 3D-display mode enable
14	RW	0x0	hwc_size Hwc size select 0: 32x32 1: 64x64
13	RW	0x0	hwc_mode Hwc color mode 0: normal color mode 1: reversed color mode
12:10	RW	0x0	win2_fmt Win2 source Format 3'b000 : ARGB888 3'b001 : RGB888 3'b010 : RGB565 3'b100: 8bpp 3'b101: 4bpp 3'b110: 2bpp 3'b111: 1bpp
9:7	RW	0x0	win1_fmt Win1 source data Format 3'b000 : ARGB888 3'b001 : RGB888 3'b010 : RGB565 3'b100 : YCbCr420 3'b101 : YCbCr422 3'b110 : YCbCr444
6:4	RW	0x0	win0_fmt Win0 source data Format 3'b000 : ARGB888 3'b001 : RGB888 3'b010 : RGB565 3'b100 : YCbCr420 3'b101 : YCbCr422 3'b110 : YCbCr444

Bit	Attr	Reset Value	Description
3	RW	0x0	hwc_en Hwc enable bit 0: Hwc layer disable 1: Hwc layer enable
2	RW	0x0	win2_en Win2 enable bit 0: Win2 layer disable 1: Win2 layer enable
1	RW	0x0	win1_en Win1 enable bit 0: Win1 layer disable 1: Win1 layerenable
0	RW	0x0	win0_en Win0 enable bit 0: Win0 layer disable 1: Win0 layer enable

LCDC_DSP_CTRL0
Address: Operational Base + offset (0x0008)
Display control register0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	dsp_ccir565_avg Cb-Cr filter in CCIR656 mode 0: drop mode 1: average mode
28	RW	0x0	yuv_clip YCrCb clip 0: disable, YCbCr no clip 1: enable, YCbCr clip before YCbCr2RGB *Y clip: 16~235, CBCR clip: 16~239
27:26	RW	0x0	win1_csc_mode Win1 YUV2RGB Color space conversion: 2'b00: mpeg 2'b01: jpeg 2'b10: hd 2'b11: Bypass
25:24	RW	0x0	win0_csc_mode Win0 YUV2RGB Color space conversion: 2'b00: mpeg 2'b01: jpeg 2'b10: hd 2'b11: Bypass
23	RW	0x0	win2_alpha_mode Win2 alpha mode 0: user-defined alpha 1: per-pixel alpha

Bit	Attr	Reset Value	Description
22	RW	0x0	win1_alpha_mode Win1 alpha mode 0: user-defined alpha 1: per-pixel alpha
21	RW	0x0	win0_alpha_mode Win0 alpha mode 0: user-defined alpha 1: per-pixel alpha
20	RW	0x0	win1_cbr_deflick Win1 Cbr deflick mode 0: disable 1: enable
19	RW	0x0	win1_yrgb_deflick Win1 YRGB deflick mode 0: disable 1: enable
18	RW	0x0	win0_cbr_deflick Win0 Cbr deflick mode 0: disable 1: enable
17	RW	0x0	win0_yrgb_deflick Win0 YRGB deflick mode 0: disable 1: enable
16	RW	0x0	win2_interlace_read Win2 interlace read mode 0: disable 1: enable
15	RW	0x0	win1_interlace_read Win1 interlace read mode 0: disable 1: enable
14	RW	0x0	win0_interlace_read Win0 interlace read mode 0: disable 1: enable
13	RW	0x0	interlace_field_pol Interlace field polarity 0: normal 1: invert
12	RW	0x0	dsp_interlace Interlace display enable 0: disable 1: enable *This mode is related to the ITU-R656 output, the display timing of odd field must be set correctly. (lcdc_dsp_vs_st_end_f1/Lcdc_dsp_vCt_end_f1)
11	RW	0x0	dither_down Dither-down enable 0: disable 1: enable

Bit	Attr	Reset Value	Description
10	RW	0x0	dither_down_mode Dither-down mode 0: RGB888 to RGB565 1: RGB888 to RGB666
9	RW	0x0	dither_up dither up RGB565 to RGB888 enable 0: disable 1: enable
8	RW	0x0	dsp_win0_top Win0 and Win1 position swap 0: win1 on the top of win0 1: win0 on the top of win1
7	RW	0x0	dclk_inv_en DCLK invert enable 0: normal 1: invert
6	RW	0x0	den_polarity DEN polarity 0: positive 1: negative
5	RW	0x0	vs_polarity VSYNC polarity 0: negative 1: positive
4	RW	0x0	hs_polarity HSYNC polarity 0: negative 1: positive
3:0	RW	0x0	dsp_out_mode Display output format 4'b0000: Parallel 24-bit RGB888 output R[7:0],G[7:0],B[7:0] 4'b0001: Parallel 16-bit RGB666 output 6'b0,R[5:0],G[5:0],B[5:0] 4'b0010: Parallel 15-bit RGB565 output 8'b0,R[4:0],G[5:0],B[4:0] 4'b0100: Serial 2x16-bit RGB888x 8'b0,G[7:0],B[7:0] + 16'b0,R[7:0] 4'b0110: ITU-656 output 16'b0,pixel_data[7:0] 4'b1000: Serial 3x8-bit RGB888 16'b0, B[7:0]+16'b0,G[7:0]+16'b0,R[7:0] 4'b1100: Serial 3x8-bit RGB888 + dummy 16'b0, B[7:0]+16'b0,G[7:0]+16'b0,R[7:0] + dummy Others: Reserved.

LCDC_DSP_CTRL1

Address: Operational Base + offset (0x000c)

Display control register1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30	RW	0x0	dsp_dummy_swap Display dummy swap enable 0: B+G+R+dummy 1: dummy+B+G+R
29	RW	0x0	dsp_delta_swap Display delta swap enable 0: disable 1: enable *See detail description in Delta display charpter.
28	RW	0x0	dsp_rg_swap Display output red and green swap enable 0: RGB 1: GRB
27	RW	0x0	dsp_rb_swap Display output red and blue swap enable 0: RGB 1: BGR
26	RW	0x0	dsp_bg_swap Display output blue and green swap enable 0: RGB 1: RBG
25	RW	0x0	dsp_black_en Black display mode When this bit enable, the pixel data output is all black (0x000000)
24	RW	0x0	dsp_blank_en Blank display mode When this bit enable, the hsync/vsync/den output is blank
23:16	RW	0x00	dsp_bg_red Background Red color
15:8	RW	0x00	dsp_bg_green Background Green color
7:0	RW	0x00	dsp_bg_blue Background Blue color

LCDC_INT_STATUS

Address: Operational Base + offset (0x0010)

Interrupt status register

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:12	RW	0x000	dsp_line_flag_num Line number of the Line flag interrupt The display line number when the flag interrupt occur, the range is (0~ DSP_VTOTAL-1).
11	W1C	0x0	bus_err_intr_en Bus error Interrupt clear (Auto clear)
10	W1C	0x0	line_flag_intr_en Line flag Interrupt clear (Auto clear)

Bit	Attr	Reset Value	Description
9	W1C	0x0	fs_intr_en Frame start interrupt clear (Auto clear)
8	W1C	0x0	hs_intr_en Horizontal start interrupt clear (Auto clear)
7	WO	0x0	bus_error_intr_clr Bus error interrupt enable 0: disable 1: enable
6	RW	0x0	line_flag_intr_clr Line flag Interrupt enable 0: disable 1: enable
5	RW	0x0	fs_intr_clr Frame start interrupt enable 0: disable 1: enable
4	RW	0x0	hst_intr_clr Horizontal start interrupt enable 0: disable 1: enable
3	RO	0x0	bus_error_intr Bus error Interrupt status
2	RO	0x0	line_flag_intr Line flag Interrupt status
1	RO	0x0	fs_intr Frame start interrupt status
0	RO	0x0	hst_intr Horizontal start interrupt status

LCDC_MCU_CTRL
Address: Operational Base + offset (0x0014)
MCU mode control register

Bit	Attr	Reset Value	Description
31	RW	0x0	mcu_type MCU LCD output SELECT
30	RW	0x0	mcu_bypass MCU LCD BYPASS MODE Select
29	RW	0x0	mcu_rs MCU LCD RS Select
28	W1C	0x0	mcu_frm_st mcu frame start Write"1" :MCU HOLD Mode Frame Start Read: MCU HOLD status
27	RW	0x0	mcu_hold_mode MCU HOLD Mode Select
26	RW	0x0	mcu_clk_sel MCU_CLK_SEL for MCU bypass 1：MCU BYPASS sync with DCLK 0：MCU BYPASS sync with HCLK
25:20	RW	0x07	mcu_rw_pend MCU_RW signal end point (0-63]

Bit	Attr	Reset Value	Description
19:16	RW	0x1	mcu_rw_pst MCU_RW signal start point (0-15)
15:10	RW	0x07	mcu_cs_pend MCU_CS signal end point (0-63)
9:6	RW	0x0	mcu_cs_pst MCU_CS signal start point (0-15)
5:0	RW	0x08	mcu_pix_total MCU LCD Interface writing period (0-63)

LCDC_BLEND_CTRL
Address: Operational Base + offset (0x0018)
Blending control register

Bit	Attr	Reset Value	Description
31:28	RW	0x0	hwc_alpha_value HWC alpha blending value
27:20	RW	0x00	win2_alpha_value Win2 alpha blending value
19:12	RW	0x00	win1_alpha_value Win1 alpha blending value
11:4	RW	0x00	win0_alpha_value Win0 alpha blending value
3	RW	0x0	hwc_alpha_en HWC alpha blending enable 0: disable; 1: enable;
2	RW	0x0	win2_alpha_en Win2 alpha blending enable 0: disable; 1: enable;
1	RW	0x0	win1_alpha_en Win1 alpha blending enable 0: disable; 1: enable;
0	RW	0x0	win0_alpha_en Win0 alpha blending enable 0: disable; 1: enable;

LCDC_WIN0_COLOR_KEY
Address: Operational Base + offset (0x001c)
Win0 color key register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	win0_key_en Win0 transparency color key enable 0: disable; 1: enable;

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	win0_key_color Win0 key color Win0 key color red[7:0],green[7:0],blue[7:0]

LCDC_WIN1_COLOR_KEY
Address: Operational Base + offset (0x0020)
Win1 color key register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	win1_key_en Win1 transparency color key enable 0: disable; 1: enable;
23:0	RW	0x000000	win1_key_color Win1 key color Win1 key color red[7:0],green[7:0],blue[7:0]

LCDC_WIN2_COLOR_KEY
Address: Operational Base + offset (0x0024)
Win2 color key register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	win2_key_en Win2 transparency color key enable 0: disable; 1: enable;
23:0	RW	0x000000	win2_key_color Win2 key color red Win2 key color red[7:0],green[7:0],blue[7:0]

LCDC_WIN0_YRGB_MST0
Address: Operational Base + offset (0x0028)
Win0 YRGB memory start address 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_yrgb0_mst win0 YRGB frame buffer memory start address 0

LCDC_WIN0_CBR_MST0
Address: Operational Base + offset (0x002c)
Win0 Cbr memory start address 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_cbr0_mst win0 CBR frame buffer memory start address 0

LCDC_WIN0_YRGB_MST1
Address: Operational Base + offset (0x0030)
Win0 YRGB memory start address 1

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_yrgb1_mst win0 YRGB frame buffer memory start address 1

LCDC_WIN0_CBR_MST1
Address: Operational Base + offset (0x0034)
Win0 Cbr memory start address 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win0_cbr1_mst win0 CBR frame buffer memory start address 1

LCDC_WIN0_VIR
Address: Operational Base + offset (0x0038)
Win0 virtual display width

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0140	win0_vir_stride Win0 Virtual Display stride Number of words of Win0 Virtual width ARGB888 : win0_vir_width RGB888 : (win0_vir_width*3/4) + (win0_vir_width%3) RGB565 : ceil(win0_vir_width/2) YUV : ceil(win0_vir_width/4)

LCDC_WIN0_ACT_INFO
Address: Operational Base + offset (0x003c)
Win0 active window width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x00ef	win0_act_height Win0 active(original) window height win_act_height = (win0 vertical size -1)
15:13	RO	0x0	reserved
12:0	RW	0x013f	win0_act_width Win0 active(original) window width win_act_width = (win0 horizontial size -1)

LCDC_WIN0_DSP_INFO
Address: Operational Base + offset (0x0040)
Win0 display width/height on panel

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win0_height Win0 display window height win0_dsp_height = (win0 vertical size -1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win0_width Win0 display window width win0_dsp_width = (win0 horizontial size -1)

LCDC_WIN0_DSP_ST
Address: Operational Base + offset (0x0044)
Win0 display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_win0_yst Win0 vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_win0_xst Win0 horizontal start point(x) of the Panel scanning

LCDC_WIN0_SCL_FACTOR_YRGB
Address: Operational Base + offset (0x0048)
Win0 YRGB scaling factor

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win0_vs_factor_yrgb Win0 YRGB vertical scaling factor
15:0	RW	0x1000	win0_hs_factor_yrgb Win0 YRGB horizontal scaling factor

LCDC_WIN0_SCL_FACTOR_CBR
Address: Operational Base + offset (0x004c)
Win0 Cbr scaling factor

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win0_vs_factor_cbr Win0 CBR vertical scaling factor
15:0	RW	0x1000	win0_hs_factor_cbr Win0 CBR horizontal scaling factor

LCDC_WIN0_SCL_OFFSET
Address: Operational Base + offset (0x0050)
Win0 scaling start point offset

Bit	Attr	Reset Value	Description
31:24	RW	0x00	win0_vs_offset_cbr Cbr Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
23:16	RW	0x00	win0_vs_offset_yrgb Y Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
15:8	RW	0x00	win0_hs_offset_cbr Cbr Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99
7:0	RW	0x00	win0_hs_offset_yrgb Y Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99

LCDC_WIN1_YRGB_MST
Address: Operational Base + offset (0x0054)
Win1 YRGB memory start address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win1_yrgb_mst Win1 YRGB frame buffer memory start address

LCDC_WIN1_CBR_MST
Address: Operational Base + offset (0x0058)
Win1 Cbr memory start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win1_cbr_mst Win1 CBR frame buffer memory start address

LCDC_WIN1_VIR
Address: Operational Base + offset (0x005c)
Win1 virtual display width

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0140	win1_vir_stride Win1 Virtual Display stride Number of words of Win1 Virtual width ARGB888 : win1_vir_width RGB888 : (win1_vir_width*3/4) + (win1_vir_width%3) RGB565 : ceil(win1_vir_width/2) YUV : ceil(win1_vir_width/4)

LCDC_WIN1_ACT_INFO
Address: Operational Base + offset (0x0060)
Win1 active window width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x00ef	win1_act_height Win1 active(original) window height win1_act_height = (win1 vertical size -1)
15:13	RO	0x0	reserved
12:0	RW	0x013f	win1_act_width Win1 active(original) window width win1_act_width = (win1 horizontial size -1)

LCDC_WIN1_DSP_INFO
Address: Operational Base + offset (0x0064)
Win1 display width/height on panel

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win1_height Win1 display window height win1_dsp_height = (win1 dsp vertical size -1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win1_width Win1 display window width win1_dsp_width = (win1 dsp horizontial size -1)

LCDC_WIN1_DSP_ST
Address: Operational Base + offset (0x0068)
Win1 display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_win1_yst Win1 vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_win1_xst Win1 horizontal start point(x) of the Panel scanning

LCDC_WIN1_SCL_FACTOR_YRGB
Address: Operational Base + offset (0x006c)
Win1 YRGB scaling factor

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win1_vs_factor_yrgb Win1 YRGB vertical scaling factor
15:0	RW	0x1000	win1_hs_factor_yrgb Win1 YRGB horizontal scaling factor

LCDC_WIN1_SCL_FACTOR_CBR
Address: Operational Base + offset (0x0070)
Win1 Cbr scaling factor

Bit	Attr	Reset Value	Description
31:16	RW	0x1000	win1_vs_factor_cbr Win1 CBR vertical scaling factor
15:0	RW	0x1000	win1_hs_factor_cbr Win1 CBR horizontal scaling factor

LCDC_WIN1_SCL_OFFSET
Address: Operational Base + offset (0x0074)
Win1 scaling start point offset

Bit	Attr	Reset Value	Description
31:24	RW	0x00	wn1_vs_offset_cbr Cbr Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
23:16	RW	0x00	wn1_vs_offset_yrgb Y Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
15:8	RW	0x00	wn1_hs_offset_cbr Cbr Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99
7:0	RW	0x00	wn1_hs_offset_yrgb Y Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99

LCDC_WIN2_MST
Address: Operational Base + offset (0x0078)
Win2 memory start address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	win2_mst Win2 frame buffer memory start address *must be alianed to 8byte address

LCDC_WIN2_VIR
Address: Operational Base + offset (0x007c)
Win2 virtual display width

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0140	win2_vir_stride Win2 Virtual Display stride Number of words of Win2 Virtual width ARGB888 : win2_vir_width RGB888 : (win2_vir_width*3/4) + (win2_vir_width%3) RGB565 : ceil(win2_vir_width/2) 8BPP : ceil(win0_vir_width/4) 4BPP : ceil(win0_vir_width/8) 2BPP : ceil(win0_vir_width/16) 1BPP : ceil(win0_vir_width/32)

LCDC_WIN2_DSP_INFO
Address: Operational Base + offset (0x0080)
Win2 display width/height on panel

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x0ef	dsp_win2_height Win2 display window height win2_dsp_height = (win2 dsp vertical size -1)
15:11	RO	0x0	reserved
10:0	RW	0x13f	dsp_win2_width Win2 display window width win2_dsp_width = (win2 dsp horizontial size -1)

LCDC_WIN2_DSP_ST
Address: Operational Base + offset (0x0084)
Win2 display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_win2_yst Win2 vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_win2_xst Win2 horizontal start point(x) of the Panel scanning

LCDC_HWC_MST
Address: Operational Base + offset (0x0088)
Hwc memory start address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	hwc_mst HWC data memory start address

LCDC_HWC_DSP_ST

Address: Operational Base + offset (0x008c)

Hwc display start point on panel

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_hwc_yst HWC vertical start point(y) of the Panel scanning
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_hwc_xst HWC horizontal start point(x) of the Panel scanning

LCDC_HWC_COLOR_LUT0

Address: Operational Base + offset (0x0090)

Hwc LUT color 0

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	hwc_color0 Hardware cursor color [23:16] : Hardware cursor color red for 2'b00 [15:8] : Hardware cursor color green for 2'b00 [7:0] : Hardware cursor color blue for 2'b00

LCDC_HWC_COLOR_LUT1

Address: Operational Base + offset (0x0094)

Hwc LUT color 1

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	hwc_color1 Hardware cursor color [23:16] : Hardware cursor color red for 2'b01 [15:8] : Hardware cursor color green for 2'b01 [7:0] : Hardware cursor color blue for 2'b01

LCDC_HWC_COLOR_LUT2

Address: Operational Base + offset (0x0098)

Hwc LUT color 2

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	hwc_color2 Hardware cursor color [23:16] : Hardware cursor color red for 2'b10 [15:8] : Hardware cursor color green for 2'b10 [7:0] : Hardware cursor color blue for 2'b10

LCDC_DSP_HTOTAL_HS_END
Address: Operational Base + offset (0x009c)
Panel scanning horizontal width and hsync pulse end

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x14a	dsp_hs_end Panel display scanning horizontal period
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_htotal Panel display scanning hsync pulse width

LCDC_DSP_HACT_ST_END
Address: Operational Base + offset (0x00a0)
Panel active horizontal scanning start point and end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x14a	dsp_hact_end Panel display scanning horizontal active start point
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_hact_st Panel display scanning horizontal active end point

LCDC_DSP_VTOTAL_VS_END
Address: Operational Base + offset (0x00a4)
Panel scanning vertical height and vsync pulse end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x0fa	dsp_vtotal Panel display scanning vertical period
15:12	RO	0x0	reserved
11:0	RW	0x00a	dsp_vs_end Panel display scanning vsync pulse width

LCDC_DSP_VACT_ST_END
Address: Operational Base + offset (0x00a8)
Panel active vertical scanning start point and end point

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x0fa	dsp_vact_end Panel display scanning vertical active start point
15:12	RO	0x0	reserved

Bit	Attr	Reset Value	Description
11:0	RW	0x00a	dsp_vact_st Panel display scanning vertical active end point

LCDC_DSP_VS_ST_END_F1
Address: Operational Base + offset (0x00ac)
Vsync start point and end point of filed1

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_vs_end_f1 interlace display vs end point Panel display scanning vertical vsync end point of 2nd field(interlace display mode)
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_vs_st_f1 interlace display vsync start point Panel display scanning vertical vsync start point of 2nd field (interlace display mode)

LCDC_DSP_VACT_ST_END_F1
Address: Operational Base + offset (0x00b0)
Vertical active start point and end point of filed1

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:16	RW	0x000	dsp_vact_end_f1 interlace display vertical active end point Panel display scanning vertical active end point of 2nd field (interlace display mode)
15:12	RO	0x0	reserved
11:0	RW	0x000	dsp_vact_st_f1 interlace display vertical active start point Panel display scanning vertical active start point of 2nd field (interlace display mode)

LCDC_REG_LOAD_EN
Address: Operational Base + offset (0x00c0)
Register config done flag

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	reg_load_en LCDC register config done In the first setting of the register, the new value was saved into the mirror register. When all the register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.

LCDC_MCU_BYPASS_WPORT
Address: Operational Base + offset (0x0100)
LCDC MCU bypass mode data write port

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	WO	0x000000	mcu_bypass_wport LCDC MCU bypass mode data write port When MCU is in BYPASS Mode, BYPASS data is written through this Port.

LCDC_MCU_BYPASS_RPORT
Address: Operational Base + offset (0x0200)
LCDC MCU bypass mode data read port

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RO	0x000000	mcu_bypass_rport LCDC MCU bypass mode data read port When MCU is in BYPASS Mode, BYPASS data is read through this Port (addr fixed).

LCDC_Win2_LUT_ADDR
Address: Operational Base + offset (0x0400)
LCDC win2 lut bus access address map

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:0	RW	0x0000000	win2_lut_addr LCDC win2 lut bus access address map Access entry for win2 LUT memory (word size only).

LCDC_DSP_LUT_ADDR
Address: Operational Base + offset (0x0800)
LCDC dsp lut bus access address map

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x000000	dsp_lut_addr LCDC dsp lut bus access address map Access entry for DSP LUT memory (word size only)

22.5 Timing Diagram

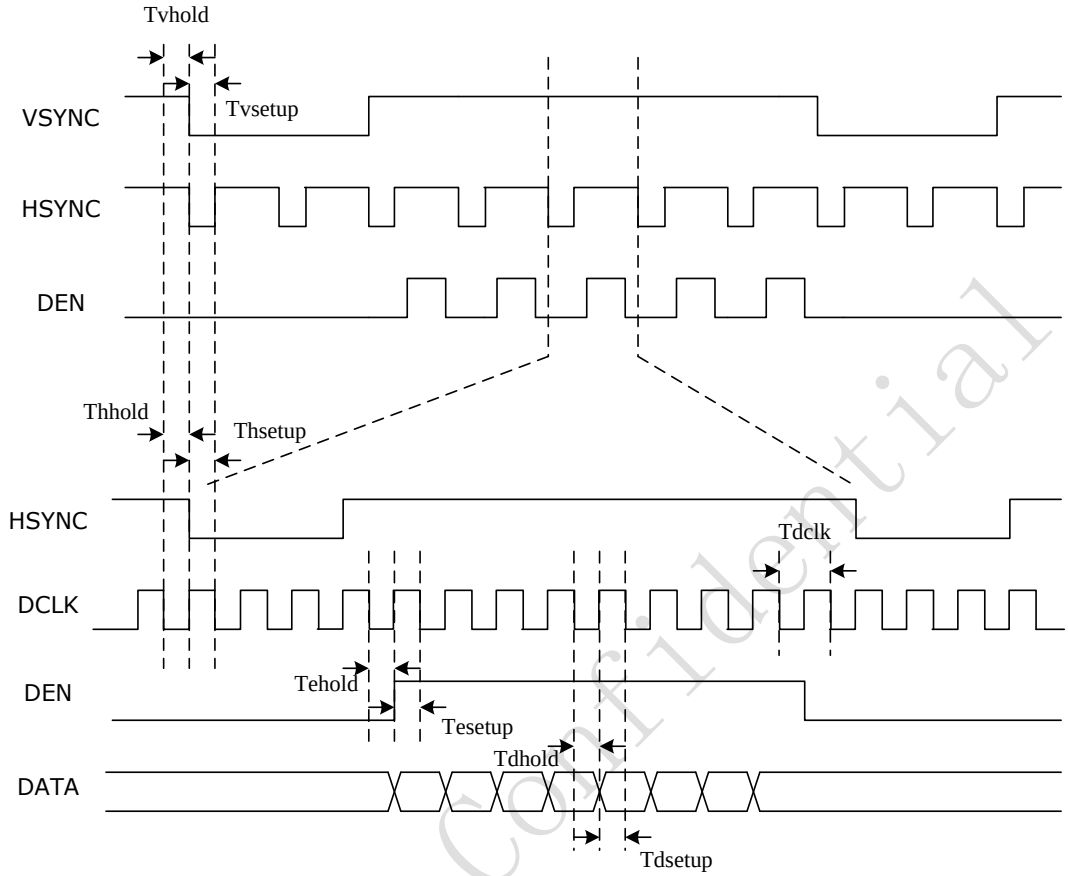


Fig. 22-18 LCDC RGB interface timing

Table 22-5 LCDC0 RGB interface signal timing constant

(VDD_core =0.99V to 1.1V, VDD_IO=3.0V to 3.6V , TA = -40℃ ot 125℃)

Item	Symbol	Min	Typ	Max	Unit
Display clock period	Tdclk	6.64	-	-	ns
VSYNC setup to DCLK falling edge	Tvsetup	2.377	2.454	2.628	ns
VSYNC hold from DCLK falling edge	Tvhold	2.531	2.686	2.905	ns
HSYNC setup to DCLK falling edge	Thsetup	2.257	2.378	2.586	ns
HSYNC hold from DCLK falling edge	Thhold	2.648	2.754	2.943	ns
DEN setup to DCLK falling edge	Tesetup	2.291	2.436	2.637	ns
DEN hold from DCLK falling edge	Tehold	2.514	2.672	2.878	ns
DATA setup to DCLK falling edge	Tdsetup	2.433	2.519	2.711	ns
DATA hold from DCLK falling edge	Tdhold	2.248	2.494	2.750	ns

Table 22-6 LCDC1 RGB interface signal timing constant

(VDD_core =0.99V to 1.1V, VDD_IO=3.0V to 3.6V , TA = -40℃ ot 125℃)

Item	Symbol	Min	Typ	Max	Unit
Display clock period	Tdclk	6.64	-	-	ns
VSYNC setup to DCLK falling edge	Tvsetup	2.398	2.497	2.689	ns
VSYNC hold from DCLK falling edge	Tvhold	2.304	2.523	2,787	ns
HSYNC setup to DCLK falling edge	Thsetup	2.220	2.367	2.595	ns
HSYNC hold from DCLK falling edge	Thhold	2.444	2.625	2.866	ns
DEN setup to DCLK falling edge	Tesetup	2.216	2.348	2.571	ns
DEN hold from DCLK falling edge	Tehold	2.448	2.636	2.885	ns
DATA setup to DCLK falling edge	Tdsetup	2.058	2.314	2.591	ns
DATA hold from DCLK falling edge	Tdhold	2.368	2.567	2.801	ns

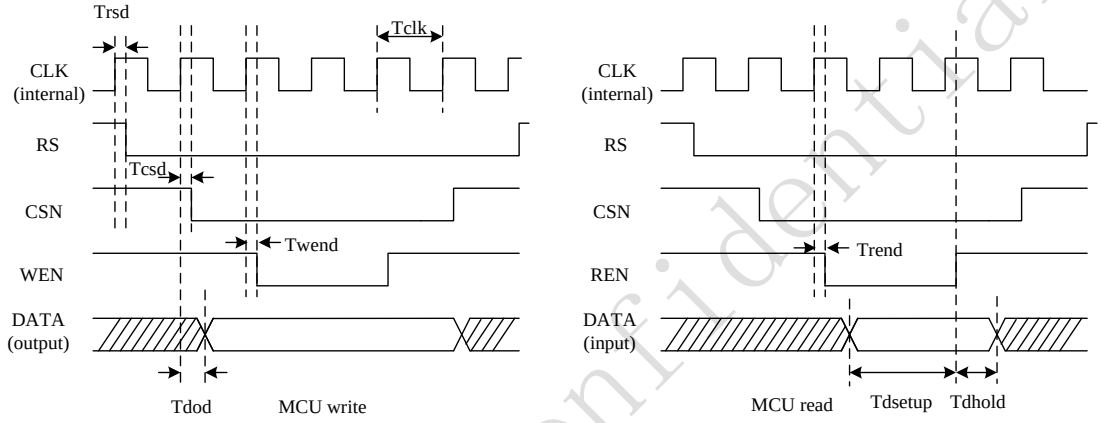


Fig. 22-19 LCDC MCU interface (i80)timing

Table 22-7 LCDC0 RGB interface signal timing constant

(VDD_core =0.99V to 1.1V, VDD_IO=3.0V to 3.6V , TA = -40℃ ot 125℃)

Item	Symbol	Min	Typ	Max	Unit
Internal clock period	Tclk	6.64	-	-	ns
RS delay from CLK rising edge	Trsd	4.407	6.605	9.801	ns
CSN delay from CLK rising edge	Tcsd	5.331	8.070	12.098	ns
WEN delay from CLK rising edge	Twend	5.383	8.166	12.269	ns
REN delay from CLK rising edge	Trend	5.410	8.214	12.387	ns
D_out delay from CLK rising edge	Tdod	6.988	8.214	16.135	ns
D_in setupto REN rising edge	Tdsetup	8.442	12.55	18.234	ns
D_in hold from REN rising edge	Tdhold	-13.1	-8.57	-6.173	ns

Table 22-8 LCDC1 RGB interface signal timing constant

(VDD_core =0.99V to 1.1V, VDD_IO=3.0V to 3.6V , TA = -40℃ ot 125℃)

Item	Symbol	Min	Typ	Max	Unit
Internal clock period	Tclk	6.64	-	-	ns
RS delay from CLK rising edge	Trsd	4.377	6.547	9.775	ns
CSN delay from CLK rising edge	Tcsd	5.218	7,953	12.008	ns
WEN delay from CLK rising edge	Twend	5.302	8.073	12.172	ns
REN delay from CLK rising edge	Trend	5.385	8.168	12.301	ns
D_out delay from CLK rising edge	Tdod	6.799	10.40	15.547	ns
D_in setupto REN rising edge	Tdsetup	7.118	11.41	17.827	ns
D_in hold from REN rising edge	Tdhold	-11.9	-7.03	-4.054	ns

22.6 Interface Description

22.6.1 LCDC0 Ouputs

LCDC0 outputs are connected to two places, one is HDMI TX video input (), and the other is LCDC0 IOs. There is no IOMUX for LCDC0 IOs.

GRF_SOC_CON0[14] is the select bit for HDMI TX input.

GRF_SOC_CON0[14] = 1'b0: HDMI TX input from LCDC0;

GRF_SOC_CON0[14] = 1'b1: HDMI TX input from LCDC1;

22.6.2 LCDC1 Outputs

LCDC1 outputs are connected to two places, one is HDMI TX video input, the other is IOs. The IOMUX for LCDC1 IOs see following table.

Table 22-9LCDC1 IOMUX

Module Pin	Direction	Pad Name	IOMUX Setting
LCDC_DCLK	O	IO_GPIO2_D[0]	GRF_GPIO2D_IOMUX[1:0]=2'b01
LCDC_DEN	O	IO_GPIO2_D[1]	GRF_GPIO2D_IOMUX[3:2]=2'b01
LCDC_HSYNC	O	IO_GPIO2_D[2]	GRF_GPIO2D_IOMUX[5:4]=2'b01
LCDC_VSYNC	O	IO_GPIO2_D[3]	GRF_GPIO2D_IOMUX[7:6]=2'b01
LCDC_DATA[23:0]	I/O	LCDC_DATA[23:0]	GRF_GPIO2A_IOMUX[15:0] =16'h5555 GRF_GPIO2B_IOMUX[15:0] =16'h5555 GRF_GPIO2C_IOMUX[15:0] =16'h5555

Notes: I=input, O=output, I/O=input/output, bidirectional

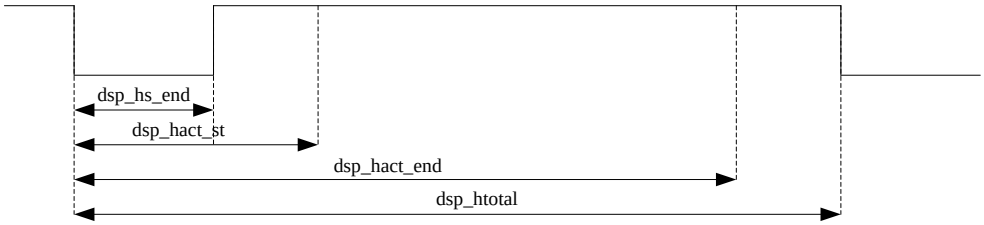
22.6.3 Pin Definition

Table 22-10 LCDC output pin definition

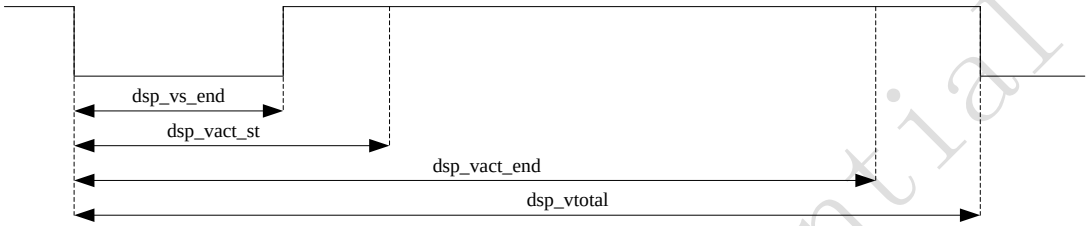
Pin	RGB 24-bit mode	RGB 18-bit mode	RGB 16-bit mode	MCU mode	656 mode
LCDC_DCLK	DCLK	DCLK	DCLK	RS	DCLK
LCDC_VSYNC	VSYNC	VSYNC	VSYNC	CSN	-
LCDC_HSYNC	HSYNC	HSYNC	HSYNC	WEN	-
LCDC_DEN	DEN	DEN	DEN	REN	-
LCDC_DATA [23:0]	DATA[23:0] = {R8,G8,B8}	DATA[17:0] = {R6,G6,B6}	DATA[15:0] = {R5,G6,B5}	DATA[23:0] DATA[17:0] DATA[15:0]	DATA[7:0]

**NOTE: In MCU mode, LCD_DCLK is used as RS signal of i80 interface, others, LCD_DCLK is used as lcdc output data clock, which can be inverted according the register setting.*

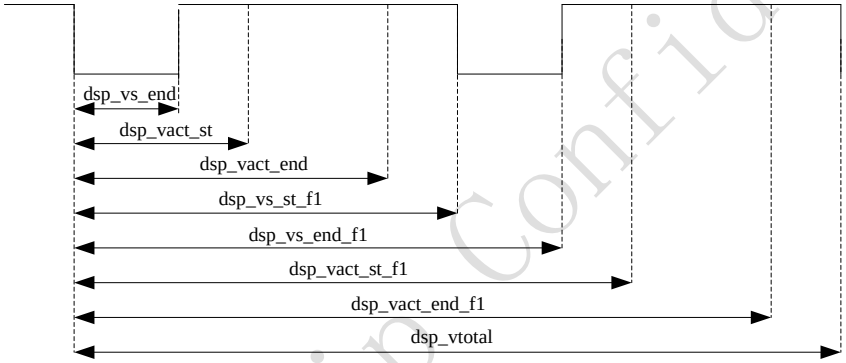
22.6.4 RGB Interface



Horizontal timing



Vertical timing(Progressive mode)



Vertical timing(Interlace mode)

Fig. 22-20 LCDC RGB interface timing setting

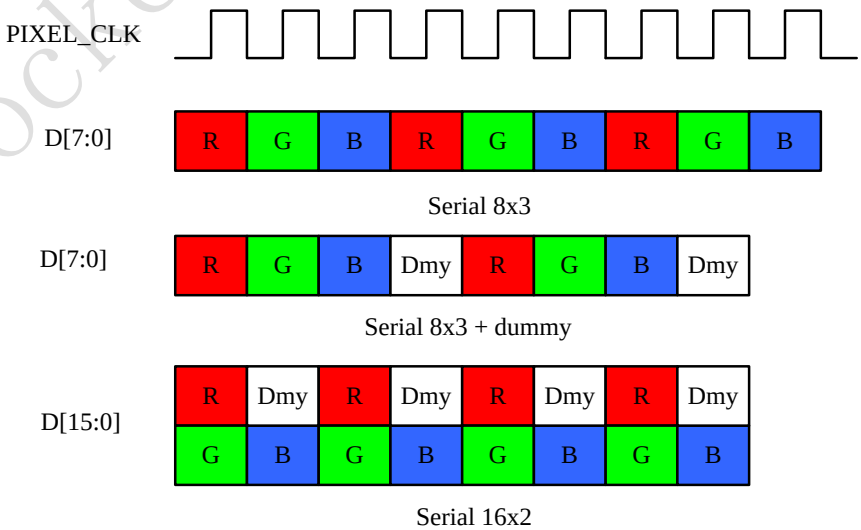


Fig. 22-21 LCDC Serail RGB LCD interface

22.6.5 MCU Interface (i80)

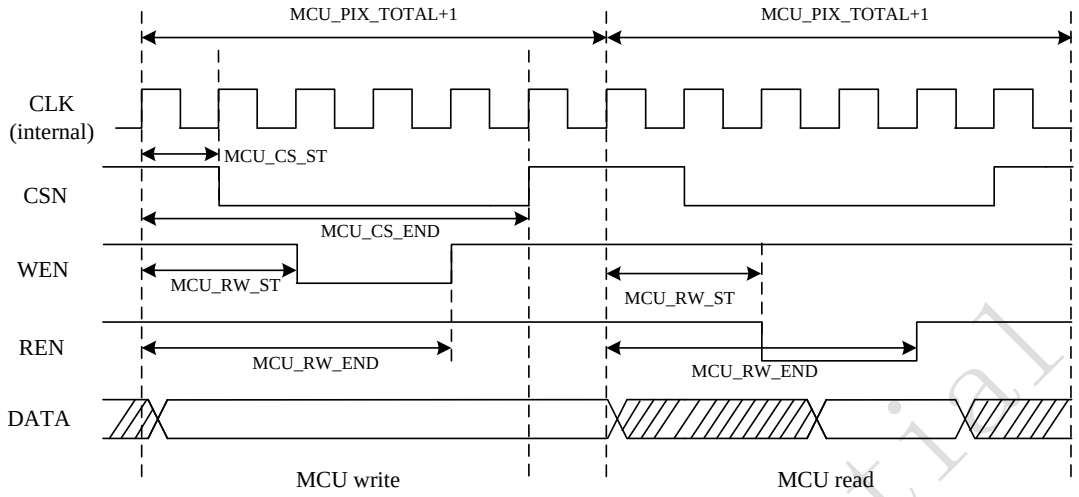


Fig. 22-22 LCDC MCU interface timing setting

22.6.6 RGB Delta Interface

RGB delta LCD handles serial 8bit data. In the case of RGB 8bit serial, there are four scanning modes for the RGB delta data.

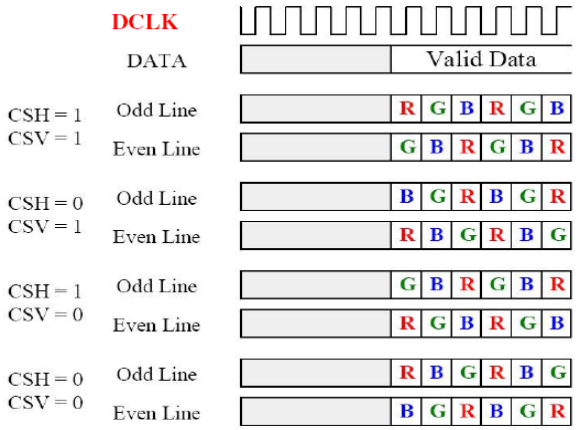


Fig. 22-23 LCDC RGB delta LCD interface

There are four setting modes for the four scanning modes of RGB delta LCD in LCD controller.

Table 22-11 LCDC delta and swap setting for RGB delta LCD

RGB delta mode	Delta_en	Dsp_rg_swap	Dsp_rb_swap	*Dsp_bg_swap*
CSH=1,CSV=1	1	0	1	0
CSH=0,CSV=1	1	0	0	0
CSH=1,CSV=0	1	0	0	1
CSH=0,CSV=0	1	0	1	1

Chapter 23 RGA

23.1 Overview

RGAis a separate 2D raster graphic acceleration unit. It accelerates 2D graphics operations, such aspoint/line drawing, image scaling, rotation, BitBLT, alpha blending and image blur/sharpness.

23.1.1 Features

- **Data format**
  - Input data: ARGB/RGB888/RGB565/YUV420/YUV422
  - Output data: ARGB/RGB888/RGB565 (YUV420/YUV422 for blur/sharpness)
  - Pixel Format conversion, BT.601/BT.709
  - Dither operation
  - Max resolution: 8192x8192 source image, 2048x2048 frame buffer
- **Scaling**
  - Down-scaling and up-scaling
  - Three sampling modes: Nearest sampling (Stretched BitBLT), Bi-linear filter or Bi-cubic filter
  - Arbitrary non-integer scaling ratio, from 1/2 to 8
  - Average filter pre-scaling (2's Down-scalingbypass path, not avaliable with other 2D operation)
- **Rotation**
  - Arbitrary rotation, minimum 1 degree step
  - No per-pixel alpha in arbitrary ratation (without 90, 180, 270)
  - x-mirror, y-mirror
- **BitBLT**
  - Block transfer
  - Color palette (with transparency mode)/Color fill
  - Transparency mode (color keying/stencil test, specified value/range)
- **Alpha Blending**
  - Per-pixel/user-specified alpha blending (Porter-duff alpha support)
  - Fading
  - Anti-aliasing (for rotation)
- **Raster operation**
  - ROP2/ROP3/ROP4
  - NoROP in arbitrary rotation (except90/180/270 degree)
- **Line/Point drawing**
  - Bresonham algorithm, Specified width
  - Anti-aliasing
- **Blur/sharpness**
  - Bypass post processingpath (not avaliable with other 2D operation)
  - Tile_based

For detailed information about RGA,please refer to **RK30xx RGA.pdf**。

Chapter 24 HDMI TX

24.1 Overview

HDMI TX is fully compliant with HDMI 1.4a specification. It offers a simple implementation for consumer electronics like DVD/player/recorder and camcorder. HDMI TX consists of one HMDI transmitter controller and one HMDI transmitter PHY.

24.1.1 Features

- HDMI version 1.4a, HDCP revision 1.4 and DVI version 1.0 compliant transmitter
- Supports DTV from 480i to 1080i/p HD resolution, and PC from VGA to UXGA
- Supports 3D and 2k x 4k video resolution output
- Programmable 2-way color space converter
- Compliant with EIA/CEA-861D
- xvYCC Enhanced Colorimetry
- Gamut Metadata transmission
- Supports RGB, YCbCr digital video input format includes ITU.656
- Supports standard SPDIF for stereo or compressed audio up to 192KHz
  - Support PCM, Dolby digital, DTS digital audio transmission through 4bits I2S up to 8 channel
  - IEC60958 or IEC61937 compatible
- Master I2C interface for DDC connection
- Configuration registers programmable via parallel interface
- Wide range channel speed up to 2.2Gbps
- Programmable PLL characteristics, channel delay, and transmitter pre-emphasis rate
- Small ISI jitter by full differential data path

24.2 Block Diagram

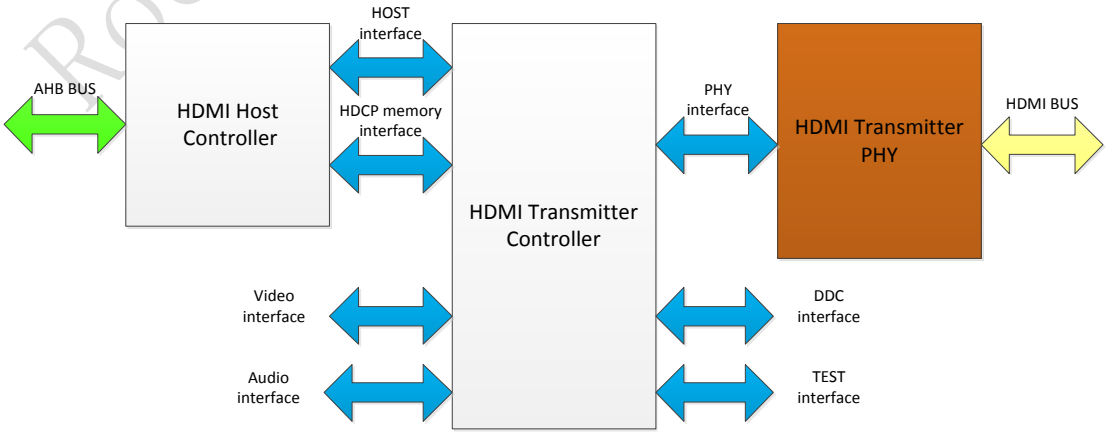


Fig. 24-1HDMI TX Block Diagram

24.3 Function Description

24.3.1 Video Data Processing

The following diagram shows video data processing path.

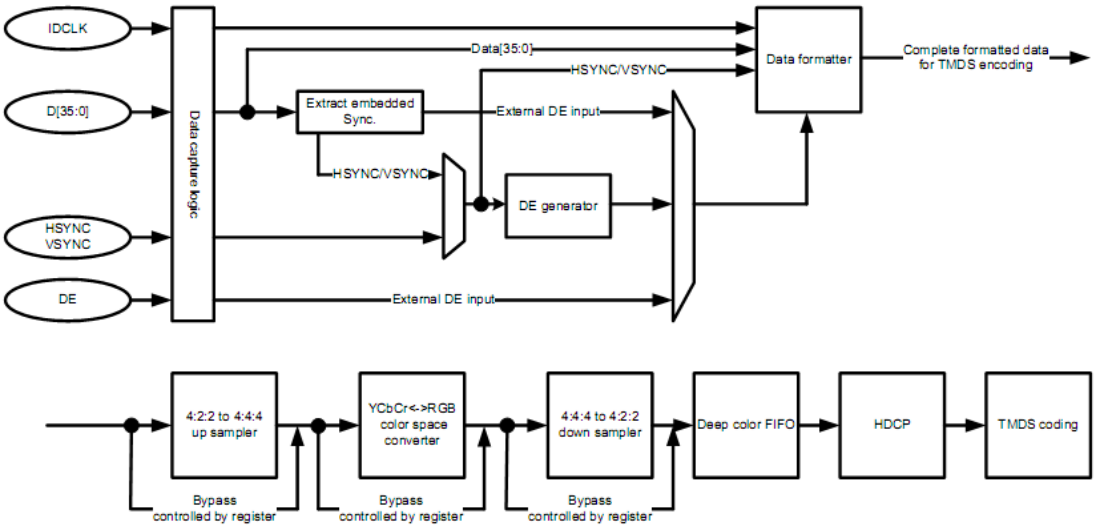


Fig. 24-2HDMI TXVideo Data Processing

23.3.1.1 Video Data Capture Logic

Video Data Capture Logic takes in uncompressed digital video through an interface from 8 to 24 bits in width. The interface has three (3) 8-bit data channels which can be configured for a number of different video formats (VID). It provides a direct connection to major MPEG decoders. The interface is configured by registers to set the bus width and format (8/10/12/16/20/24-bit) and rising/falling edge latching. The appropriate registers must be configured to describe which format of video is being input. This information is passed over the HDMI link in the CEA-861D Active Video Information (AVI) packets. It also supports dual-edge clocking using 8-18 data pins.

Table 24-1HDMI TX Supported Input Video Formats

Input ID	Color Space	Pixel Encoding	Sync	Clock Rate	Bit Width per Color	Pin Count
1	RGB	4:4:4	Separate	1x	8	24
2	RGB	4:4:4	Separate	DDR	8	12
3	YCbCr	4:4:4	Separate	1x	8	24
4	YCbCr	4:4:4	Separate	DDR	8	12
5	YCbCr	4:2:2	Separate	1x	12	24
6	YCbCr	4:2:2	Separate	1x	10	20
7	YCbCr	4:2:2	Separate	1x	8	16
8	YCbCr	4:2:2	Separate	DDR	12	12
9	YCbCr	4:2:2	Separate	DDR	10	10
10	YCbCr	4:2:2	Separate	DDR	8	8
11	YCbCr	4:2:2	Separate	2x	12	12
12	YCbCr	4:2:2	Separate	2x	10	10
13	YCbCr	4:2:2	Separate	2x	8	8
14	YCbCr	4:2:2	Embedded	1x	12	24
15	YCbCr	4:2:2	Embedded	1x	10	20
16	YCbCr	4:2:2	Embedded	1x	8	16
17	YCbCr	4:2:2	Embedded	2x	12	12
18	YCbCr	4:2:2	Embedded	2x	10	10
19	YCbCr	4:2:2	Embedded	2x	8	8
20	RGB	4:4:4	Separate	1x	12	36
21	RGB	4:4:4	Separate	1x	10	30
22	RGB	4:4:4	Separate	DDR	12	18
23	RGB	4:4:4	Separate	DDR	10	18
24	YCbCr	4:4:4	Separate	1x	12	36
25	YCbCr	4:4:4	Separate	1x	10	30
26	YCbCr	4:4:4	Separate	DDR	12	18
27	YCbCr	4:4:4	Separate	DDR	10	18
28	YCbCr	4:2:2	Separate	1x	12	24
29	YCbCr	4:2:2	Separate	1x	10	20
30	YCbCr	4:2:2	Separate	ITU656	12	12
31	YCbCr	4:2:2	Separate	ITU656	10	10
32	YCbCr	4:2:2	Separate	ITU656	8	8
33	YCbCr	4:2:2	Embedded	ITU656	12	12
34	YCbCr	4:2:2	Embedded	ITU656	10	10
35	YCbCr	4:2:2	Embedded	ITU656	8	8

1. Video input data pin assignment

The below tables show the data assignment for each input video modes. For double data rate modes, (2, 4, 8, 9, and 10), timing should be arranged so that the data on the first row (indicated with letter "R") are latched with rising edge of the clock and the data on the second row (indicated with letter "F") are latched with falling edge of the clock.Note that video data input ports colored by "gray" shall be fixed by "0".

Table 24-2HDMI TX Video Data Assignment (1-19)

Input ID		data[23:0] {[R8[7:0], G8[7:0], B8[7:0]})																							
		23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	R	R[7:0]								G[7:0]								B[7:0]							
2	F													G[3:0]				B[7:0]							
	R													R[7:0]								G[7:4]			
3	R	Cr[7:0]								Y[7:0]								Cb[7:0]							
4	F													Y[3:0]				Cb[7:0]							
	R													Cr[7:0]								Y[7:4]			
5	P0	Cb0[11:4]								Y0[11:4]								Cb0[3:0]				Y0[3:0]			
	P1	Cr0[11:4]								Y1[11:4]								Cr0[3:0]				Y1[3:0]			
6	P0	Cb0[9:2]								Y0[9:2]								Cb0[1:0]				Y0[1:0]			
	P1	Cr0[9:2]								Y1[9:2]								Cr0[1:0]				Y1[1:0]			
7	P0	Cb0[7:0]								Y0[7:0]															
	P1	Cr0[7:0]								Y1[7:0]															
8	P0-F													Y0[7:4]				Cb0[3:0]				Y0[3:0]			
	P0-R													Cb0[11:4]								Y0[11:8]			
	P1-F													Y1[7:4]				Cr0[3:0]				Y1[3:0]			
	P1-R													Cr0[11:4]								Y1[11:8]			
9	P0-F													Y0[5:4]		Cb0[3:0]					Y0[3:0]				
	P0-R													Cb0[9:4]							Y0[9:6]				
	P1-F													Y1[5:4]		Cr0[3:0]					Y1[3:0]				
	P1-R													Cr0[9:4]							Y1[9:6]				
10	P0-F													Cb0[3:0]				Y0[3:0]							
	P0-R													Cb0[7:4]				Y0[7:4]							
	P1-F													Cr0[3:0]				Y1[3:0]							
	P1-R													Cr0[7:4]				Y1[7:4]							
11	P0-R									Cb0[11:4]												Cb0[3:0]			
	P0-R									Y0[11:4]												Y0[3:0]			
	P1-R									Cr0[11:4]												Cr0[3:0]			
	P1-R									Y1[11:4]												Y1[3:0]			
12	P0-R									Cb0[9:2]												Cb0[1:0]			
	P0-R									Y0[9:2]												Y0[1:0]			
	P1-R									Cr0[9:2]												Cr0[1:0]			
	P1-R									Y1[9:2]												Y1[1:0]			
13	P0-R									Cb0[7:0]															
	P0-R									Y0[7:0]															
	P1-R									Cr0[7:0]															
	P1-R									Y1[7:0]															
14	P0	Cb0[11:4]								Y0[11:4]								Cb0[3:0]				Y0[3:0]			
	P1	Cr0[11:4]								Y1[11:4]								Cr0[3:0]				Y1[3:0]			
15	P0	Cb0[9:2]								Y0[9:2]								Cb0[1:0]				Y0[1:0]			
	P1	Cr0[9:2]								Y1[9:2]								Cr0[1:0]				Y1[1:0]			
16	P0	Cb0[7:0]								Y0[7:0]															
	P1	Cr0[7:0]								Y1[7:0]															
17	P0-R									Cb0[11:4]												Cb0[3:0]			
	P0-R									Y0[11:4]												Y0[3:0]			
	P1-R									Cr0[11:4]												Cr0[3:0]			
	P1-R									Y1[11:4]												Y1[3:0]			
18	P0-R									Cb0[9:2]												Cb0[1:0]			
	P0-R									Y0[9:2]												Y0[1:0]			
	P1-R									Cr0[9:2]												Cr0[1:0]			
	P1-R									Y1[9:2]												Y1[1:0]			
19	P0-R									Cb0[7:0]															
	P0-R									Y0[7:0]															
	P1-R									Cr0[7:0]															
	P1-R									Y1[7:0]															

Table 24-3HDMI TX Video Data Assignment (20-29)

Input ID		data[35:0] {[DCR[3:0], DCG[3:0], DCB[3:0], R8[7:0], G8[7:0], B8[7:0]})																							
		35~32				31~28				27~24				23~16				15~8						7~0	
20	R	R[3:0]				G[3:0]				B[3:0]				R[11:4]				G[11:4]						B[11:4]	
21	R	R[1:0]				G[1:0]				B[1:0]				R[9:2]				G[9:2]						B[9:2]	
22	F														G[1:0]			B[3:0]				G[7:4]		B[11:4]	
	R															R[3:0]			G[3:2]			R[11:4]			G[11:8]
23	F																B[1:0]					G[5:2]		B[9:2]	
	R														R[1:0]				G[1:0]			R[9:2]			G[9:6]
24	R	Cr[3:0]				Y[3:0]				Cb[3:0]				Cr[11:4]				Y[11:4]						Cb[11:4]	
25	R	Cr[1:0]				Y[1:0]				Cb[1:0]				Cr[9:2]				Y[9:2]						Cb[9:2]	
26	F														Y[1:0]			Cb[3:0]				Y[7:4]		Cb[11:4]	
	R															Cr[3:0]			Y[3:2]			Cr[11:4]			Y[11:8]
27	F																Cb[1:0]					Y[5:2]		Cb[9:2]	
	R														Cr[1:0]				Y[1:0]			Cr[9:2]			Y[9:6]
28	P0					Y0[3:0]				Cb[3:0]								Y0[11:4]						Cb[11:4]	
	P1					Y1[3:0]				Cr[3:0]								Y1[11:4]						Cr[11:4]	
29	P0					Y0[1:0]				Cb[1:0]								Y0[9:2]						Cb[9:2]	
	P1					Y1[1:0]				Cr[1:0]								Y1[9:2]						Cr[9:2]	

Table 24-4HDMI TX Video Data Assignment (30-35)

Input ID		data[23:0] ({Rs[7:0], Gs[7:0], Bs[7:0]})																							
		23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
30	P0-R									Cb0[11:4]												Cb0[3:0]			
	P0-R									Y0[11:4]												Y0[3:0]			
	P1-R									Cr0[11:4]												Cr0[3:0]			
	P1-R									Y1[11:4]												Y1[3:0]			
31	P0-R									Cb0[9:2]												Cb0[1:0]			
	P0-R									Y0[9:2]												Y0[1:0]			
	P1-R									Cr0[9:2]												Cr0[1:0]			
	P1-R									Y1[9:2]												Y1[1:0]			
32	P0-R									Cb0[7:0]															
	P0-R									Y0[7:0]															
	P1-R									Cr0[7:0]															
	P1-R									Y1[7:0]															
33	P0-R									Cb0[11:4]												Cb0[3:0]			
	P0-R									Y0[11:4]												Y0[3:0]			
	P1-R									Cr0[11:4]												Cr0[3:0]			
	P1-R									Y1[11:4]												Y1[3:0]			
34	P0-R									Cb0[9:2]												Cb0[1:0]			
	P0-R									Y0[9:2]												Y0[1:0]			
	P1-R									Cr0[9:2]												Cr0[1:0]			
	P1-R									Y1[9:2]												Y1[1:0]			
35	P0-R									Cb0[7:0]															
	P0-R									Y0[7:0]															
	P1-R									Cr0[7:0]															
	P1-R									Y1[7:0]															

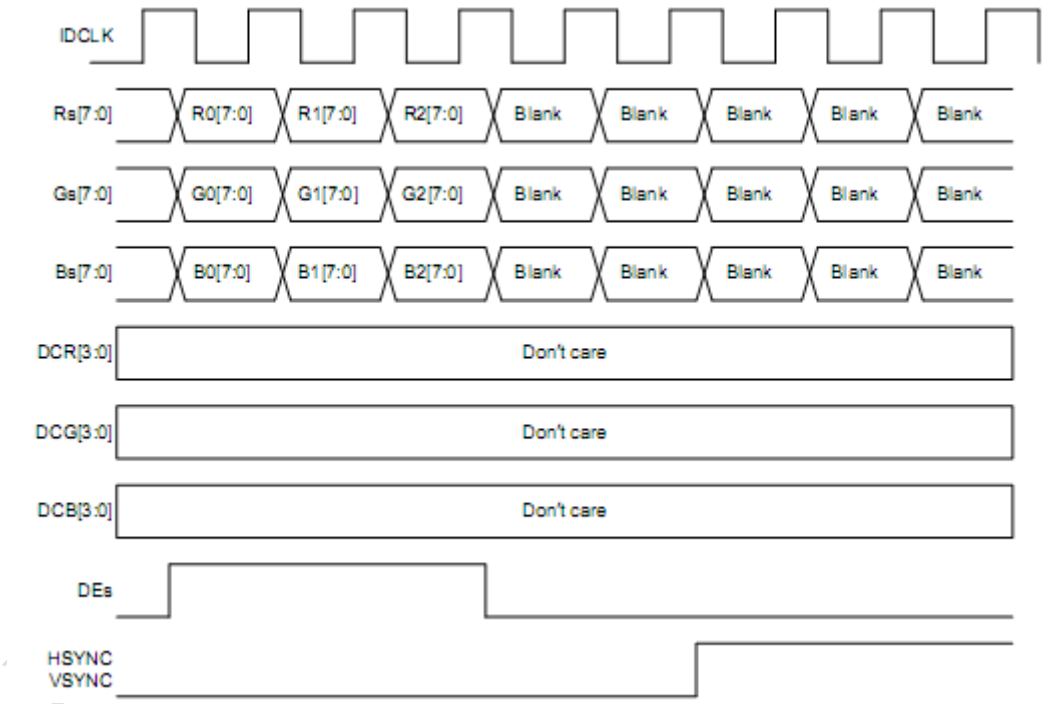


Fig. 24-3HDMI TX Video Input Timing ID.1

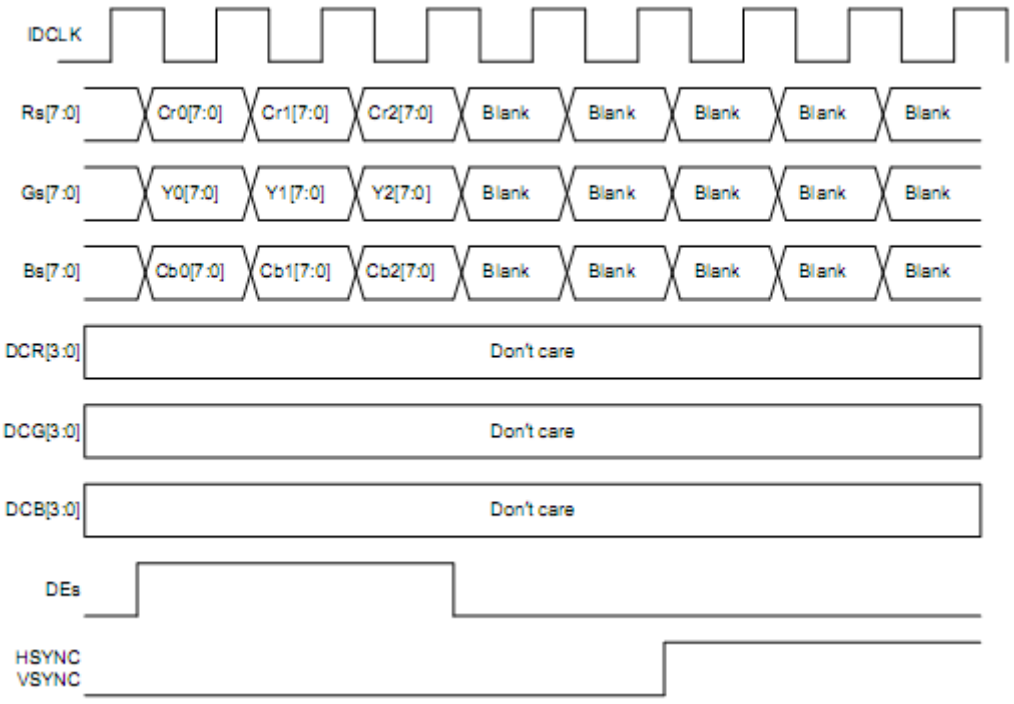


Fig. 24-4HDMI TX Video Input Timing ID.3

2. Video control register settings

(1) Audio/Video Setting.1 (#54h) and Video Setting.3 (#350h) register
It controls Data capture logics, 422 to 444 pixel encoding converter and Extract embedded Sync.

(2) Video Setting.1 (#58h)
It controls Data capture logics, YCbCr to RGB color space converter, Extract embedded Sync.

(3) Deep Color Modes (#5Ch)
It controls Deep color FIFO.

(4) Color space conversion configuration.1(#34Ch)
It controls Data capture logics and YCbCr to RGB color space converter.

Table 24-5HDMI TX control register settings for each video input ID.

ID	Input Format	Output Format	15h	16h	17h	D3h	D4h
1	RGB 4:4:4, Separate SYNC, 1x clk, 8 bit	RGB 4:4:4	00h	34h	20h	81h	00h
		YCbCr 4:4:4	00h	74h	20h	*1	00h
		YCbCr 4:2:2	00h	B4h	20h	*1	00h
2	RGB 4:4:4, Separate SYNC, DDR, 8 bit	RGB 4:4:4	0Ah	34h	20h	81h	00h
		YCbCr 4:4:4	0Ah	74h	20h	*1	00h
		YCbCr 4:2:2	0Ah	B4h	20h	*1	00h
3	YCbCr 4:4:4, Separate SYNC, 1x clk, 8 bit	YCbCr 4:4:4	00h	75h	20h	81h	00h
		RGB 4:4:4	00h	35h	20h	81h	00h
		YCbCr 4:2:2	00h	B5h	20h	81h	00h
4	YCbCr 4:4:4, Separate SYNC, DDR, 8 bit	YCbCr 4:4:4	0Ah	75h	20h	81h	00h
		RGB 4:4:4	0Ah	35h	20h	81h	00h
		YCbCr 4:2:2	0Ah	B5h	20h	81h	00h

*1 : Color space conversion from RGB to YCbCr is not automatically. To convert the colors, set appropriate coefficients into 18h~2Ch and set bit0 of 3Bh to turn on color conversion.

When the manual CSC is performed, auto bit (bit7 of D3h) must be cleared.
*2 : ITU656 stream is taken from any of the 3 channels. The value is specified in bit 3:2 in 16h register. The default (and also this example) uses ch1 (D8-D15).

23.3.1.2 Setting Video ID (VID) in CEA-861D

There are two methods to specify VID.

1. Using external video parameter setting registers.

HDMI TX supports any type of video format by setting appropriate values to External video parameter

Registers as below.

Address C0h External video parameter settings

Address C4h External horizontal total (LSB)

Address C8h External horizontal total (MSB)

Address CCh External horizontal blank (LSB)

Address D0h External horizontal blank (MSB)

Address D4h External horizontal delay (LSB)

Address D8h External horizontal delay (MSB)

Address DCh External horizontal duration (LSB)

Address E0h External horizontal duration (MSB)

Address E4h External vertical total (LSB)

Address E8h External vertical total (MSB)

Address F4h External vertical blank

Address F8h External vertical delay

Address FCh External vertical duration

In order to activate these register values, bit.0 in External video parameter settings register shall be set. Refer to diagram below for each parameters correspondence at video format.

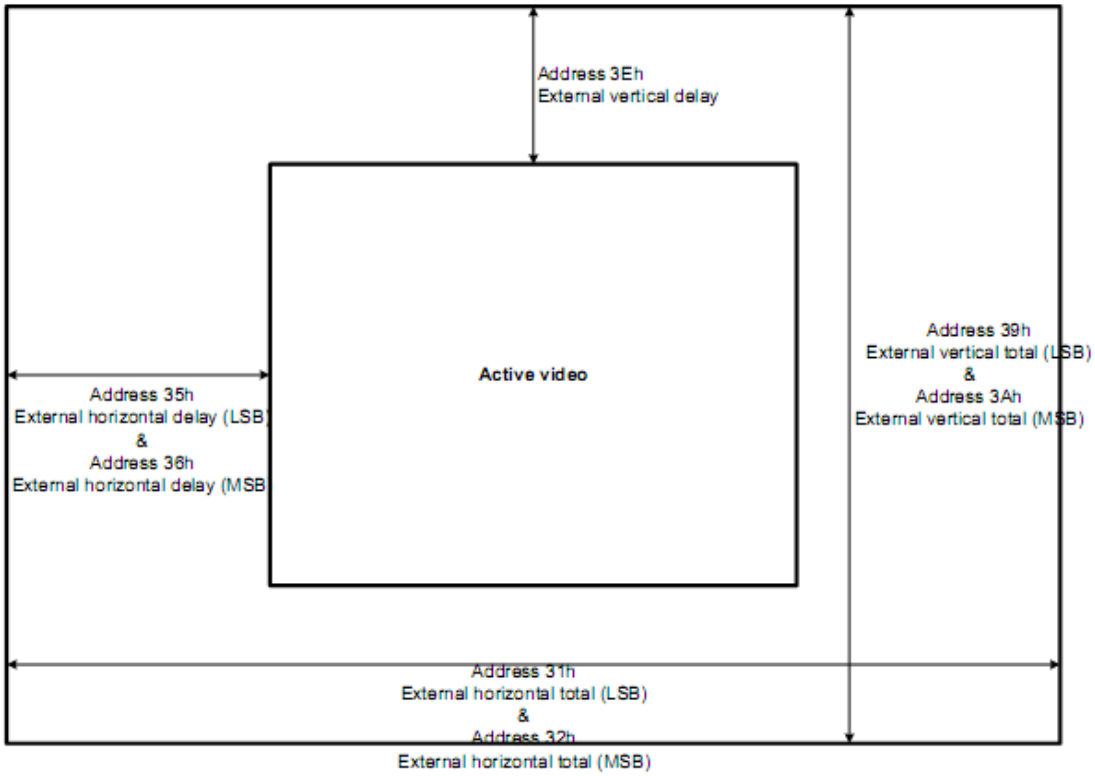


Fig. 24-5HDMI TXExternal Video Parameter Setting

External video parameter setting value

Example for VID2

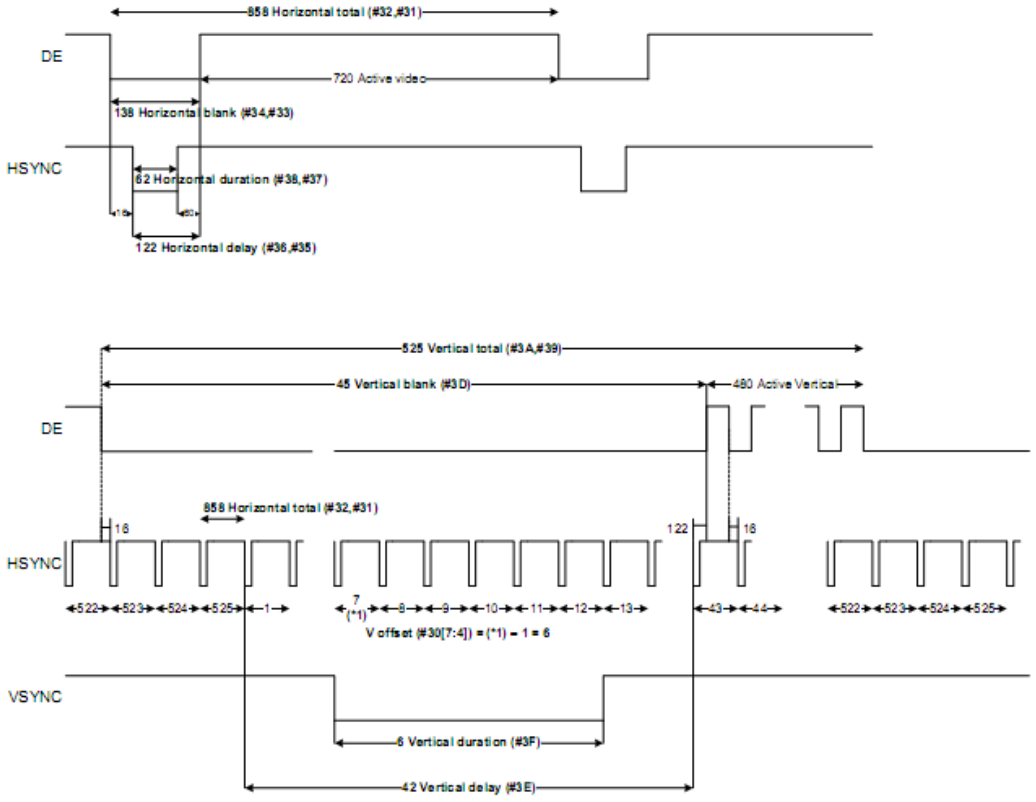


Fig. 24-6HDMI TXExternal Video SettingParameters Diagram

Table 24-6HDMI TX External Video Setting Example

The following is an example of external video parameter registers.

Addr.	Register Name	VID7	VID3	VID10	VID14 VID15	VID18	VID22	VID25 VID26	VID29 VID30	VID35 VID36	VID37 VID38
#30h	External video parameter settings	33h	61h	33h	61h	01h	03h	03h	01h	61h	01h
#31h	External horizontal total (LSB)	B4h	5Ah	68h	B4h	60h	C0h	80h	C0h	68h	80h
#32h	External horizontal total (MSB)	06h	03h	0Dh	06h	03h	06h	0Dh	06h	0Dh	0Dh
#33h	External horizontal blank (LSB)	14h	8Ah	28h	14h	90h	20h	40h	20h	28h	40h
#34h	External horizontal blank (MSB)	01h	00h	02h	01h	00h	01h	02h	01h	02h	02h
#35h	External horizontal delay (LSB)	EEh	7Ah	DCh	F4h	84h	08h	10h	08h	E8h	10h
#36h	External horizontal delay (MSB)	00h	00h	01h	00h	00h	01h	02h	01h	01h	02h
#37h	External horizontal duration (LSB)	7Ch	3Eh	F8h	7Ch	40h	7Eh	FCh	80h	F8h	00h
#38h	External horizontal duration (MSB)	00h	00h	00h	00h	00h	00h	00h	00h	00h	01h
#39h	External vertical total (LSB)	0Dh	0Dh	0Dh	0Dh	71h	71h	71h	71h	0Dh	71h
#3Ah	External vertical total (MSB)	02h	02h	02h	02h	02h	02h	02h	02h	02h	02h
#3Dh	External vertical blank	16h	2Dh	16h	2Dh	31h	18h	18h	31h	2Dh	31h
#3Eh	External vertical delay	15h	2Ah	15h	2Ah	2Ch	16h	16h	2Ch	2Ah	2Ch
#3Fh	External vertical duration	03h	06h	03h	06h	05h	03h	03h	05h	06h	05h

Additionally, appropriate data shall be set into AVI InfoFrame data buffer including VID.

(1) Special video format

The following special video format can also be supported by using external video parameter settings. Note that "virtual VID" is just for simulation purpose and not compliant with CEA861D.

Table 24-7HDMI TX Special video format

method		Vertial VID	H total	V total	H active	V active	H blank	V blank	V Freq (Hz)	Pixel Freq(MHz)
480i60Hz	Field alt.	Code106	858	1050	720	240 x4	138	22	29.97	27
576i50Hz		Code121	864	1250	720	288 x4	144	24	25	27
1080i60Hz		Code105	2200	2250	1920	540 x4	280	22	30(29.97)	148.5(148.351)
1080i50Hz		Code139	2304	2500	1920	540 x4	384	85	25	144
1080i50Hz		Code120	2640	2250	1920	540 x4	720	22	25	148.5
480p60Hz	Frame alt	Code102	858	1050	720	480 x2	138	45	59.94	54
576p50Hz		Code117	864	1250	720	576 x2	144	49	50	54
720p60Hz		Code104	1650	1500	1280	720 x2	370	30	60(59.94)	148.5(148.351)
720p50Hz		Code119	1980	1500	1280	720 x2	700	30	59	148.5
720p24Hz		Code160	3300	1500	1280	720 x2	2020	30	24(23.976)	118.8(118.681)
1080p24Hz		Code132	2750	2250	1920	1080 x2	830	45	24(23.976)	148.5(148.351)

The following is table of external video parameter registers for special video formats.

Table 24-8HDMI TX External Video ParameterSetting for Special video format

Addr.	Register Name	Code 106	Code 121	Code 105	Code 139	Code 120	Code 102	Code 117	Code 104	Code 119	Code 160	Code 132
#30h	External video parameter settings	01h	01h	0Dh	05h	0Dh	01h	01h	0Dh	0Dh	0Dh	0Dh
#31h	External horizontal total (LSB)	5Ah	60h	98h	00h	50h	5Ah	60h	72h	BCh	E4h	BEh
#32h	External horizontal total (MSB)	03h	03h	08h	09h	0Ah	03h	03h	06h	07h	0Ch	0Ah
#33h	External horizontal blank (LSB)	8Ah	90h	18h	80h	D0h	8Ah	90h	72h	BCh	E4h	3Eh
#34h	External horizontal blank (MSB)	00h	00h	01h	01h	02h	00h	00h	01h	02h	07h	03h
#35h	External horizontal delay (LSB)	77h	84h	C0h	60h	C0h	7Ah	84h	04h	04h	04h	C0h
#36h	External horizontal delay (MSB)	00h	00h	00h	01h	00h	00h	00h	01h	01h	01h	00h
#37h	External horizontal duration (LSB)	3Eh	3Fh	2Ch	A8h	2Ch	3Eh	40h	28h	28h	28h	2Ch
#38h	External horizontal duration (MSB)	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h	00h
#39h	External vertical total (LSB)	1Ah	E2h	CAh	C4h	CAh	1Ah	E2h	DCh	DCh	DCh	CAh
#3Ah	External vertical total (MSB)	04h	04h	08h	09h	08h	04h	04h	05h	05h	05h	08h
#3Dh	External vertical blank	16h	18h	16h	55h	16h	2Dh	31h	1Eh	1Eh	1Eh	2Dh
#3Eh	External vertical delay	12h	16h	14h	55h	14h	24h	2Ch	19h	19h	1Eh	29h
#3Fh	External vertical duration	03h	03h	05h	05h	05h	06h	05h	05h	05h	05h	05h

(2) Vertical blanking insertion

Maximun 3 of vertical blanking duration can be inserted at any position during vertical active video line. It works with external DE input mode, i.e. bit.0 of #54h shall be set.

The start vertical line number and blanking duration shall be specified by #3CCh - #3ECh registers. Refer to 1.4 for detail. External DE input shall be toggled by exact timing according to register settings.

(3) Support variable duration of HSYNC/VSYNC input

In order to allow variable duration of HSYNC/VSYNC input signaling, Video input option register (#3F0h) settings are required. Refer to 1.4 for detail.

2. Using pre-programmed VID

HDMI TX has pre-programmed VID settings internally for basic video formats as below.

Table 24-9HDMI TX External Video ParameterSetting for Special video format

VID	Horizontal total	Vertical total	Progressive/Interlace	Pixel clock (MHz)	V freq. (Hz)
1	640	480	P	25.20/25.175	60/59.94
2	720	480	P	27.027/27.00	60/59.94
4	1280	720	P	74.25/74.176	60/59.94
5	1920	1080	I	74.25/74.176	60/59.94
6	1440	480	I	27.027/27.00	60/59.94
16	1920	1080	P	148.5/148.352	60/59.94
17	720	576	P	27.00	50
19	1280	720	P	74.25	50
20	1920	1080	I	74.25	50
21	1440	576	I	27.00	50
31	1920	1080	P	148.5	50
32	1920	1080	P	74.25	24

Host controller shall set just appropriate VID to AVI InfoFrame data buffer for PB4, i.e. #19Ch(PB4) with index register #17Ch=06h(AVI InfoFrame) for pre-programmed VID. Also, bit.0 in External video parameter settings register (#C0h) shall be reset. Other external video setting register value shall be ignored.

23.3.1.3 Embedded Sync. Extraction

DE, HSYNC and VSYNC signals may be extracted from the start of active video (SAV) and end of active video (EAV) codes within the 656 video stream. Refer to a diagram as below for sync. extraction parameters.

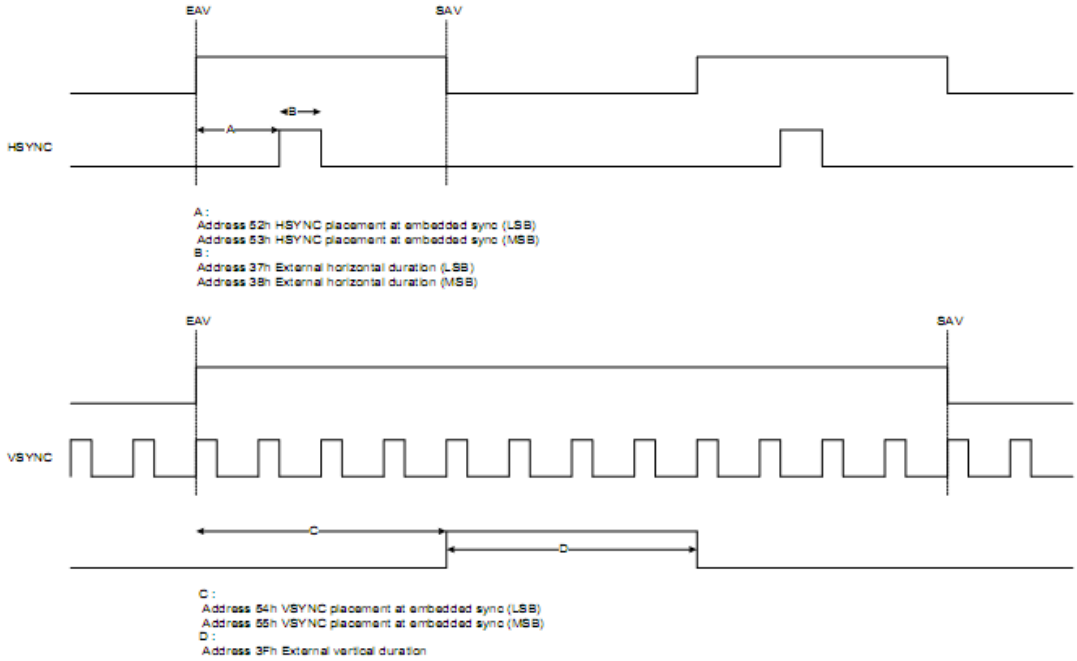


Fig. 24-7HDMI TXEmbedded Sync. Extraction Parameters Diagram

23.3.1.4 Data Enable (DE) Generator

HDMI TX has DE signal generator by incoming HSYNCs, VSYNCs and idclk. External DE is optional and selected by appropriate register settings. This feature is particularly useful when interfacing to MPEG decoders that do not provide a specific DE output signal.

23.3.1.5 Color Space Converter

Color space converter (CSC) is available to interface to several MPEG decoders like with YCbCr-only outputs, and to provide full DVI 1.0 backwards compatibility.

1. Input/Output format

   Color space : RGB or YCbCr

   Pixel encoding : 4:4:4

   Bit width : 8,10,12bit

2. Register settings

Refer to Register Settings for Each Video Formats table for detail.

3. Register Setting for coefficients of Color Space Conversion

Color space conversion is supported via 3x3 matrix calculations. Predefined

coefficients are prepared for convenience, but users can modify the coefficients via register access. The predefined values are set as recommended in ITU-R BT 601.5 for SDTV and ITU-R BT 709.5 for HDTV and are only available for color conversion from YCbCr to RGB.

The predefined coefficients are loaded based on the following equations.

480p, 480i, 576p, 576i, 240p, and 288p - limited range (a)

R : Y + 1.403(Cr-128)

G : Y - 0.714(Cr-128) - 0.344(Cb-128)

B : Y + 1.773(Cb-128)

VGA - full range (b)

R : 1.164(Y-16) + 1.596(Cr-128)

G : 1.164(Y-16) - 0.813(Cr-128) - 0.391(Cb-128)

B : 1.164(Y-16) + 2.018(Cb-128)

1080i, 1080p, and 720p - limited range 60Hz (c)

R : Y + 1.5748(Cr-128)

G : Y - 0.4681(Cr-128) - 0.1873(Cb-128)

B : Y + 1.8556(Cb-128)

1080i, 1080p, and 720p - limited range 50Hz (d)

R : Y + 1.403(Cr-128)

G : Y - 0.714(Cr-128) - 0.344(Cb-128)

B : Y + 1.773(Cb-128)

The equation to calculate CSC is defined as follows:

R : (C0*Y + C1*Cr + C2*Cb) / 1024 + C3

G : (C4*Y + C5*Cr + C6*Cb) / 1024 + C7

B : (C8*Y + C9*Cr + C10*Cb) / 1024 + C11

The value C0 is {csc_coeff_hi[3:0], csc_coeff_lo[7:0]} of 60h and 64h registers. The value C1 is {csc_coeff_hi[3:0], csc_coeff_lo[7:0]} of 68h and 6Ch registers and so on. The value C3 is {csc_coeff_hi[0], csc_coeff_lo[7:0]} of 78h and 7Ch registers.

To use custom coefficient values, first clear csc_auto bit in 34Ch register. Then write values into csc_coeff registers. To enable CSC, set csc_en bit in ECh register.

4. CSC Coefficient Register Values

When auto CSC bit is set (it is set by default), CSC coefficients registers are loaded automatically to YCbCr to RGB conversion value. The values depend on the current video format in AVI InfoFrame. There are 4 types of CSC coefficients as described in the previous section. When AVI InfoFrame is not

set, the default format, SDTV limited range (a), is loaded. As soon as a valid AVI InfoFrame is written, the register values get overwritten by appropriate coefficients. The table below shows the pre-defined register values to be loaded into CSC coefficient registers.

Table 24-10HDMI TX CSC coeffcients Values

CSC Register Addr	(a) SDTV limited range	(b) SDTV full range	(c) HDTV 60Hz	(d) HDTV 50Hz
18h	04	04	04	04
19h	00	A8	00	00
1Ah	05	06	06	05
1Bh	9D	62	4D	9D
1Ch	00	00	00	00
1Dh	00	00	00	00
1Eh	02	02	02	02
1Fh	B4	DF	CA	B4
20h	04	04	04	04
21h	00	A8	00	00
22h	12	13	11	12
23h	DB	41	DF	DB
24h	11	11	10	11
25h	60	90	C0	60
26h	00	00	00	00
27h	88	87	54	88
28h	04	04	04	04
29h	00	A8	00	00
2Ah	00	00	00	00
2Bh	00	00	00	00
2Ch	07	08	07	07
2Dh	18	12	6C	18
2Eh	02	03	02	02
2Fh	E3	15	EE	E3

23.3.1.6 Pixel Encoding Converter

HDMI TX has two kind of pixel encoding converter, up sampler and down sampler.

1. Input/Output format

   Input : 4:2:2 to up sampler or 4:4:4 to down sampler

   Output : 4:4:4 from up sampler or 4:2:2 from down sampler

2. Register settings

Refer to Register Settings for Each Video Formats table for detail.

23.3.1.7 HDCP Authentication

HDMI TXhas a capability for HDCP authentication by hardware. The following is state diagram of HDCP authentication.

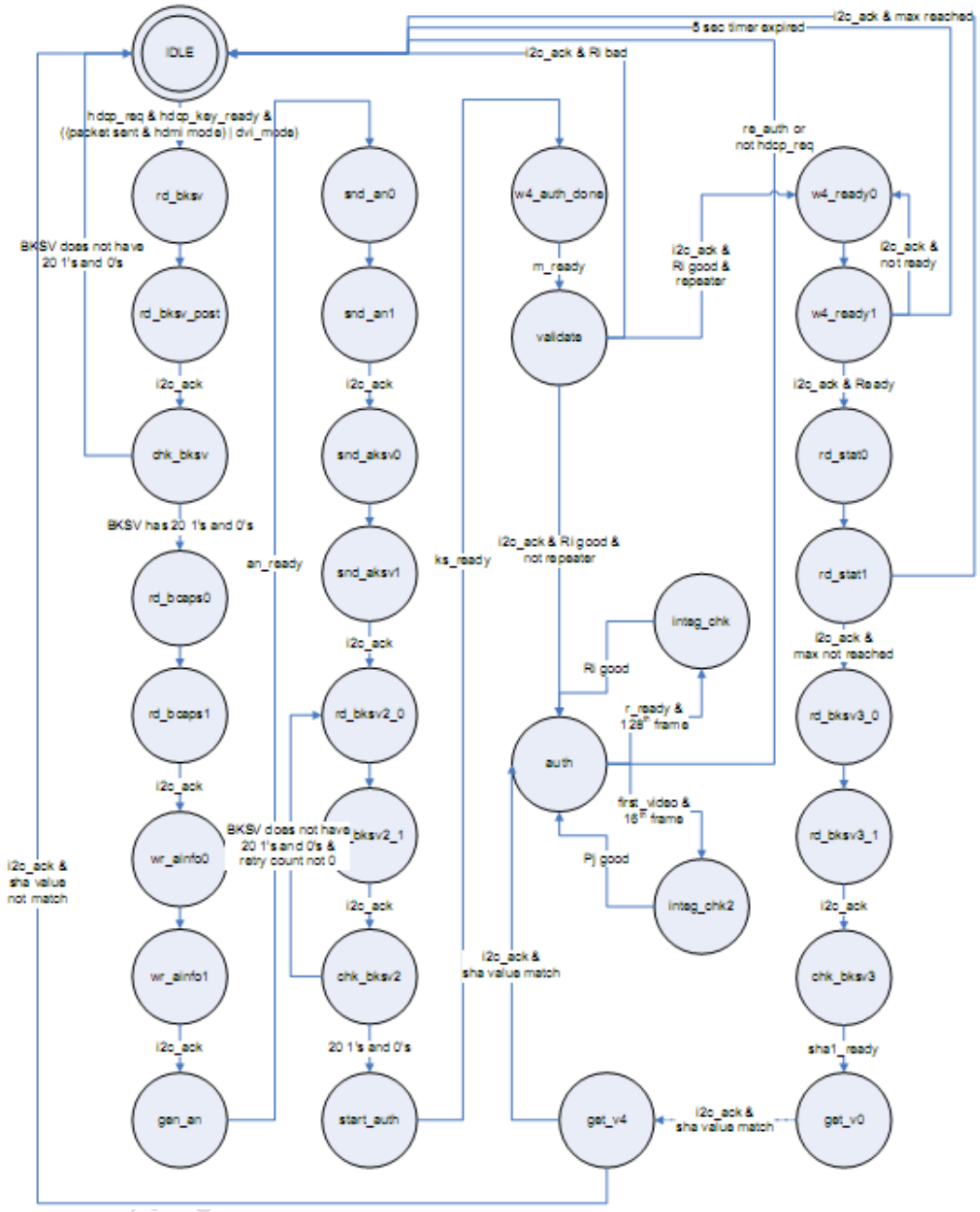


Fig. 24-8 HDMI TX HDCP Authentication State Diagram

- Red task will be done by software that is initial command to start HDCP authentication.

- After green line that is error case or authentication done state, software gets interrupt from IP core.

- Violet states will be repeated during normal HDCP transmission.

23.3.1.8 HDCP Cipher Encryption

HDCP Cipher Encryption Block is compliant with HDCP 1.4 specification. Upon request via configuration register,

HDCP block can authenticate a connected device(s) and send encrypted data over HDMI. This block has following features.

Authenticate up to 2 receivers and repeaters with maximum cascade of 7.

Advanced Cipher Mode support.

Enhanced Link Verification support.

DVI Mode support.

Requires system software to check for Bksv revocation list.

23.3.1.9 HDCP key

HDCP key shall be stored in external memory. HDMI TX provides an interface to this memory and read key at beginning of HDCP authentication. Purchasing HDCP keys, programming and security are customer's responsibility.

23.3.1.10 TMDS encoder

TMDS encoder performs 8-to-10-bit TMDS encoding on the audio/video/aux data received from the HDCP XOR mask. This data is output onto three TMDS differential data lines along with a TMDS differential clock. A resister tied to the REXT pin is used to control the TMDS swing amplitude.

24.3.2 Audio Data Processing

HDMI TX supports digital audio over either SPDIF and four(4) I2S inputs.

23.3.2.1 SPDIF

SPDIF stream can carry 2-channel uncompressed PCM data (IEC 60958) or a compressed bit stream for multi-channel (IEC 61937) formats. The audio data capture logic forms the audio data into packets in accordance with the HDMI specification. SPDIF input supports audio sampling rates from 32 to 192kHz. The following shows the SPDIF audio formats that are supported for each of the video formats.

Table 24-11HDMI TX Supported SPDIF Sampling frequency at each video format

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
720x480p/720x576p	Yes	Yes	Yes	Yes	Yes	No	No
1440x480i/1440x576i	Yes	Yes	Yes	Yes	Yes	No	No
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

23.3.2.2 I2S

Four(4) I2S inputs also allow transmission of DVD-Audio and decoded Dolby Digital to A/V Receivers and high-end displays. The interface supports from 2-channel to 8-channel audio up to 192 kHz. The I2S pins must also be coherent with MCLK. The appropriate registers must be configured to describe the format of audio being input. This information is passed over the HDMI link in the CEA-861D Audio Info (AI) packets. Table shows the I2S 8 channel audio formats that are supported for each of the video formats.

Table 24-12HDMI TX Supported I2S 2Ch Audio Sampling frequency at each video format

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
720x480p/720x576p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1440x480i/1440x576i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 24-13HDMI TX Supported I2S 8Ch Audio Sampling frequency at each video format

Video Format	32kHz	44.1kHz	48kHz	88.2kHz	96kHz	176.4kHz	192kHz
720x480p/720x576p	Yes	Yes	Yes	No	No	No	No
1440x480i/1440x576i	Yes	Yes	Yes	Yes	No	No	No
720p	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080i	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1080p	Yes	Yes	Yes	Yes	Yes	Yes	Yes

1. Variable word length

HDMI TX supports variable word length, 16bits to 32bits for I2S audio inputs. Note that 32bits word lengthis not supported at Right-justified mode. It shall be controlled by bit.3-0 of Source number/Word length register (#54h).

2. Interface modes

There are three (3) I2S input modes supported. These modes shall be controlled by bit.1-0 of I2S audio setting register (#30h).

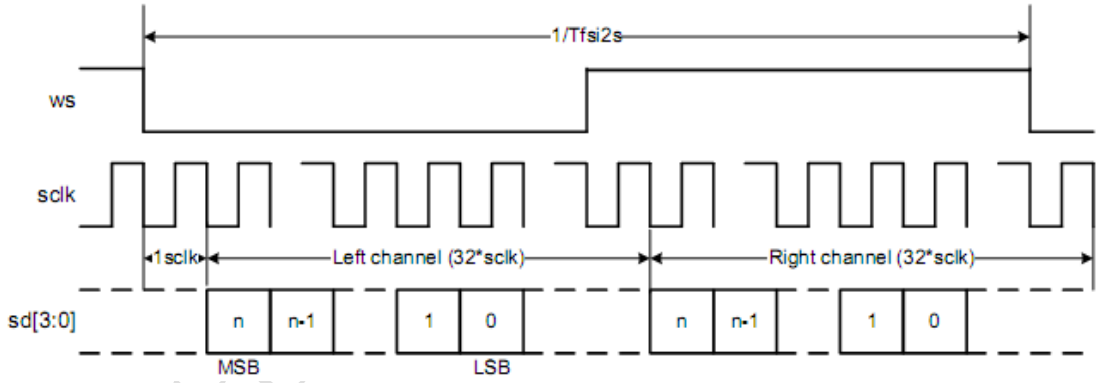


Fig. 24-9HDMI TX I2S input timing at Standard I2S mode

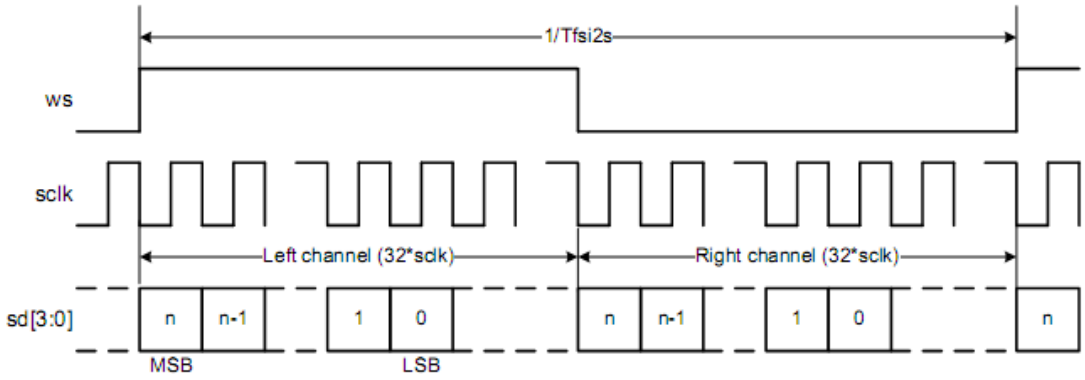


Fig. 24-10HDMI TX I2S input timing at Left justified mode

23.3.2.3 High Bit Rate Audio (HBR)

SPDIF stream can carry 2-channel uncompressed PCM data (IEC 60958) or a compressed bit stream for multi-channel (IEC 61937) formats. The audio data capture logic forms the audio data into packets in accordance with the HDMI specification. SPDIF input supports audio sampling rates from 32 to 192kHz. The following shows the SPDIF audio formats that are supported for each of the video formats.

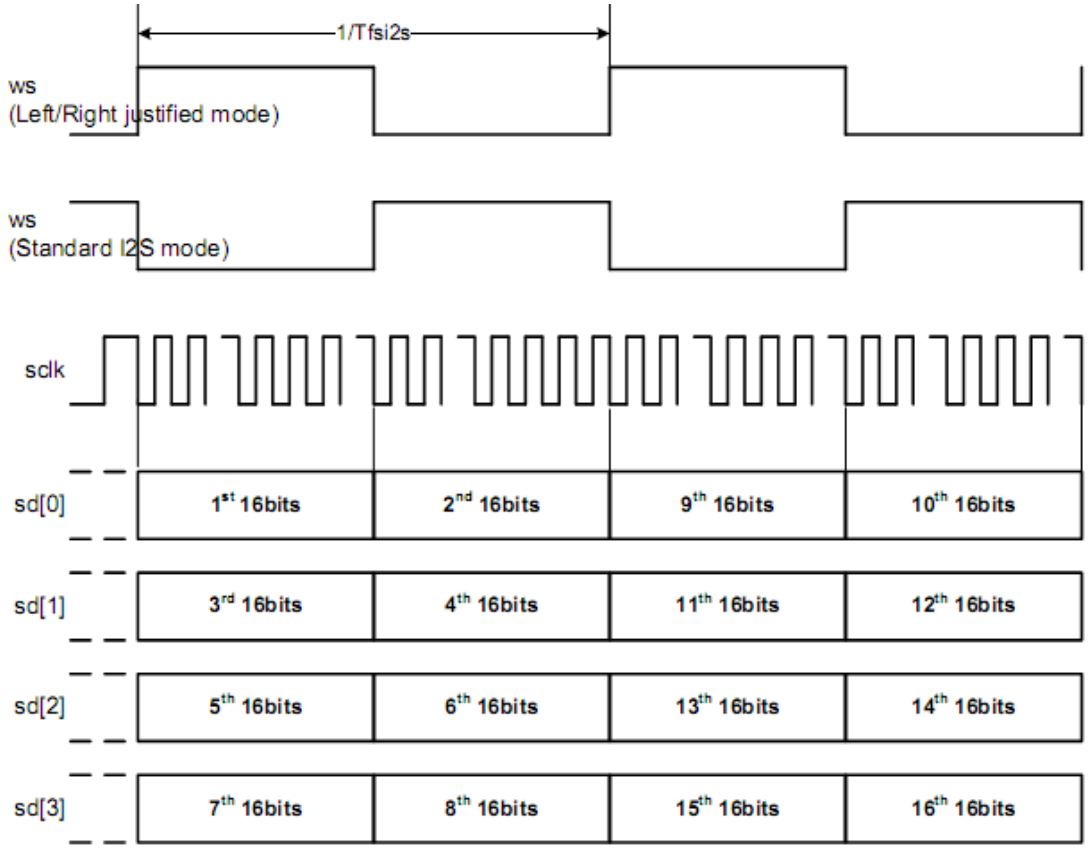


Fig. 24-11 HDMI TX HBR data stream input ordering

24.3.3 Controller Interface

23.3.3.1 Interrupt Signals

intr_n pin outputs a signal to interrupt the microcontroller based on the following conditions.

- Monitor Detect (either from the HPD input level, or from the Receiver Sense)
- VSYNC (It helps for host controller to synchronize with vertical timing interval)
- Error conditions like Audio FIFO overflow, HDCP authentication failure etc.

Its active level can be configured by int_omode signals.

The following shows the sheme of interrupt signaling.

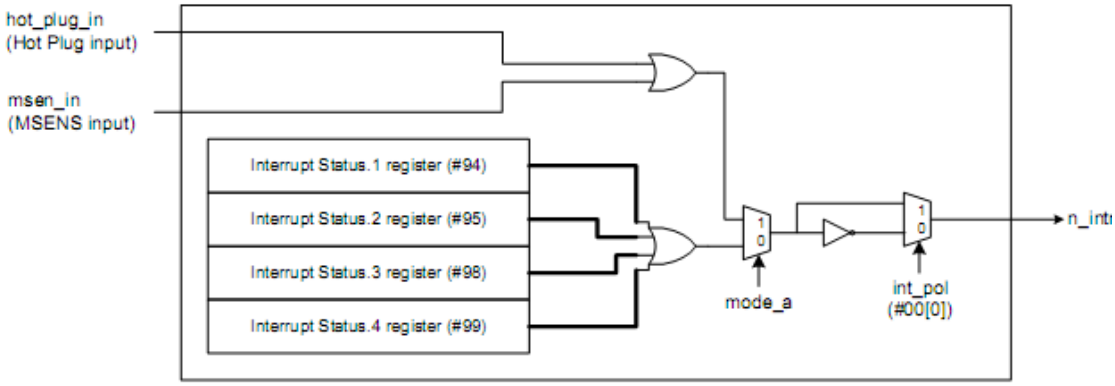


Fig. 24-12 HDMI TXInterrupt Signaling Block Diagram

The source of interrupt signaling is varied depending on power save mode.

At power save mode_a, the source of interrupt signaling is hot plug or msens input from external of HDMI TX which are asynchronous inputs. Interrupt status register shall not get set by them because clock for registers (i.e. DDC_CK) is not active in power save mode_a.

Host controller shall move to power save mode_b once it gets interrupted in power save mode_a. In power save mode_b/d/e, the source of inpterrupt signaling shall be switched to wired-or signal of interrupt status register outputs. The interrupt status registers shall get set by hot plug and msens input in mode_b/d/e because clock for registers (i.e. DDC_CK) now becomes active in this power save mode. Refer to HOT PLUG DETECTION SEQUENCES for detail.

23.3.3.2 DDC Master I2C Interface

HDMI TX has I2C Master Interface for DDC transactions. It enables for host controller to read EDID, HDCP authentication by issuing simple register access. The I2C bus speed is limited by DDC specification. DDC bus access frequency can be controlled by register #204h and #208h. Refer to register detail descriptions.

1. Access frequency

DDC bus access frequency can be programmed by registers. Refer to register detail descriptions for details.

2. Access override

DDC bus access can be overrided by bus control registers. Refer to register detail descriptions for details.

3. SOC integration example

The following diagram shows an example of SOC integration related with DDC bus.

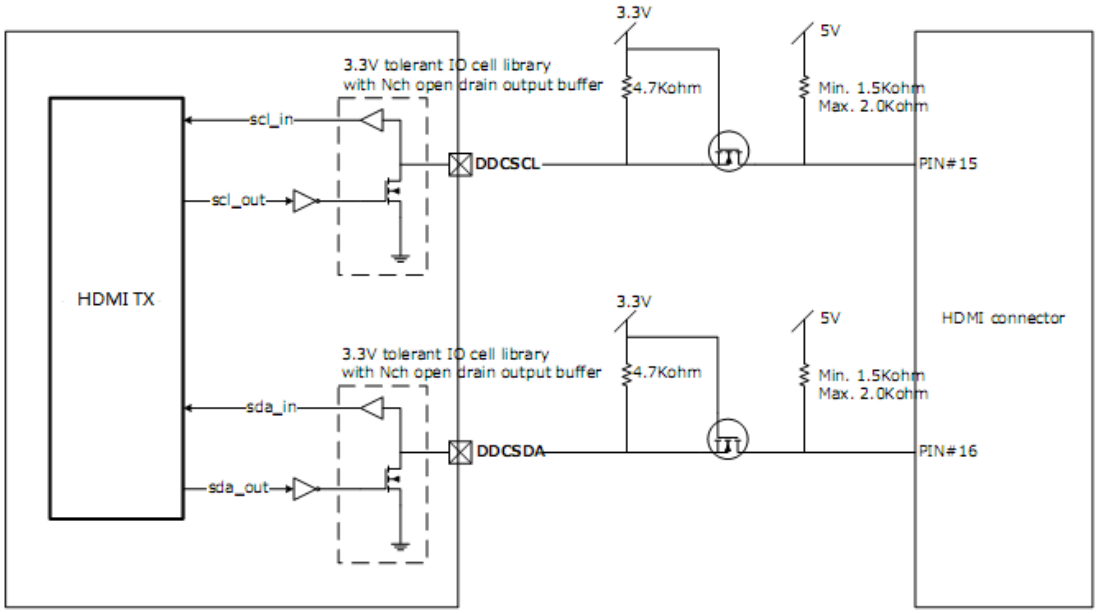


Fig. 24-13 HDMI TXintegration example diagram related with DDC bus

24.3.4 HDCP Key Memory Interface

HDCP in the external memory must be stored in the following manner. The table below shows a HDCP key as provided by DCP LLC.

- The first 4 bytes of the data "01 00 00 00" indicates that this file contains keys for TX side. The next 5 bytes of data "11 22 33 44 55" is the KSV example of 1st key.
- The next 3 bytes of data "00 00 00" are inserted as a place holder.
- The next 280 bytes of data starting at address 00Ch and ending at 123h is the private key example.
- The next 20 bytes of data starting at 124h and ending at 137h is hash value example of the key and this is the end of the 1st key.
- The next 5 bytes starting at 138h and ending at 13Ch is the next KSV and continues until the end of file.

Table 24-14HDMI TX HDCP key map example as provided by DCP LLC

addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000h	01	00	00	00	11	22	33	44	55	00	00	00	01	02	03	04
010h	05	06	07	08	09	0a	0b	0c	0d	0e	0f	10	11	12	13	14
020h	15	16	17	18	19	1a	1b	1c	1d	1e	1f	20	21	22	23	24
030h	25	26	27	28	29	2a	2b	2c	2d	2e	2f	30	31	32	33	34
040h	35	36	37	38	39	3a	3b	3c	3d	3e	3f	40	41	42	43	44
050h	45	46	47	48	49	4a	4b	4c	4d	4e	4f	50	51	52	53	54
060h	55	56	57	58	59	5a	5b	5c	5d	5e	5f	60	61	62	63	64
070h	65	66	67	68	69	6a	6b	6c	6d	6e	6f	70	71	72	73	74
080h	75	76	77	78	79	7a	7b	7c	7d	7e	7f	80	81	82	83	84
090h	85	86	87	88	89	8a	8b	8c	8d	8e	8f	90	91	92	93	94
0A0h	95	96	97	98	99	9a	9b	9c	9d	9e	9f	a0	a1	a2	a3	a4
0B0h	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af	b0	b1	b2	b3	b4
0C0h	b5	b6	b7	b8	b9	ba	bb	bc	bd	be	bf	c0	c1	c2	c3	c4
0D0h	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf	d0	d1	d2	d3	d4
0E0h	d5	d6	d7	d8	d9	da	db	dc	dd	de	df	e0	e1	e2	e3	e4
0F0h	e5	e6	e7	e8	e9	ea	eb	ec	ed	ee	ef	f0	f1	f2	f3	f4
100h	f5	f6	f7	f8	f9	fa	fb	fc	fd	fe	ff	00	01	02	03	04
110h	05	06	07	08	09	0a	0b	0c	0d	0e	0f	10	11	12	13	14
120h	15	16	17	18	00	01	02	03	04	05	06	07	08	09	0a	0b
130h	0c	0d	0e	0f	10	11	12	13	aa	bb	cc	dd	ee	00	00	00

24.3.5 Power Save Mode

HDMI TX supports multiple power save modes that realize most efficient power control. Table shows the summary of power save modes. Note that 1-10ms waiting time between each power save mode transition is required for stable PHY functions. The source signal of interrupt is varied depending on each power save mode. Refer to 1.3.3.1 Interrupt signaling for details.

Table 24-15HDMI TX Power save mode summary

power save mode	idclk input	Clock tree	TMDS driver	Descriptions	Active modules
mode_a	Not required	OFF	OFF	Minimum power consumption Only power save mode register can be accessed.	slishdmi13t_ps_mode_reg
mode_b	Required	OFF	OFF	DDC I2C active After PHY parameter settings, DDC_CK equals to idclk input.	slishdmi13t_ps_mode_reg slishdmi13t_regs slishdmi13t_i2c_edid slishdmi13t_hot_plug Note that clock tree for other modules is inactive
mode_d	Required	ON	OFF	Internal clock tree active All logics except for control registers are under synchronous reset. Any unstable input signals shall not be propagated into logics.	Same as mode_b Note that clock tree for all modules is active
mode_e	Required	ON	ON	All functions are active It is prohibited for host controller to change some register settings under mode_e. It might cause functions unstable. Host controller shall get back to mode_d when it is required to change these register settings. Refer to 9.3 Register access rules chapter.	All modules

24.3.6 Clock Distribution

23.3.6.1 Input Clocks form Controller

- idclk (sc_clk)
  MAX. Freq. = 148.5MHz at normal video input mode.
  It shall be provided by host controller. All video input data shall be synchronized with it. It is fed to all of HDMI TX logics at SCAN test mode.
- cpu_clk

It shall be provided by host controller. It is required to access Power save mode register (Address.00h).

- hdcp_mem_clk
  It shall be provided by host controller. It is required to load HDCP key into HDMI TX.
- mclk
  MAX. Freq. = 73.728MHz
  It shall be provided by audio source devices and shall be coherence with audio data input.
- sclk
  MAX. Freq. = 25MHz
  It shall be provided by audio source devices and synchronized with sd[3:0] or dsd_l[3:0]/dsd_r[3:0] audio input data.

23.3.6.2 Clocks from PHY to Controller

- IDCKX2
  MAX. Freq. = 148.5MHz
  Its frequency is double of idclk. It is fed for audio sampling modules. It is available at mode_d/mode_e in power save mode.
- TMDS_CK
  MAX. Freq. = 222.75MHz at deep color 12bit mode, 0r 148.5MHz at 8bit color depth
  It is identical to TXC/TXC_N on TMDS bus. Video input signals with idclk input are synchronized with TMDS_CK by deep_color.v module. It is available at mode_d/mode_e in power save mode.
- DDC_CK
  MAX. Freq. = 148.5MHz
  Its frequency depends on power save mode. Basically, DDC_CK equals to idclk input once PHY parameters have been set.
- LB_CK
  MAX. Freq. = 222.75MHz at deep color 12bit mode, 0r 148.5MHz at 8bit color depth
  It is for loopback test mode. lb_data[2:0] are synchronized with it.
- FBCK_A
  MAX. Freq. = 148.5MHz
  It is for PLL frequency test. It is generated by PLL_A feedback clock.
- FBCK_B
  MAX. Freq. = 222.75MHz at deep color 12bit mode, 0r 148.5MHz at 8bit color depth
  It is for PLL frequency test. It is generated by PLL_B feedback clock.

24.4 Register Description

24.4.1 Register Summary

24.4.2 Detail Register Description

Refer to 《RK30xx HDMI TX Register Description》

24.5 Interface Description

24.5.1 Video Input Source

In RK30xx, the HDMI TX video source comes from LCDC0 or LCDC1.

GRF_SOC_CON0[14] is the select bit for HDMI TX input.

GRF_SOC_CON0[14] = 1'b0: HDMI TX input from LCDC0;

GRF_SOC_CON0[14] = 1'b1: HDMI TX input from LCDC1;

24.5.2 Audio Input Source

In RK30xx, the HDMI TX audio source comes from I2S_8CH.

24.6 Programming Guide

24.6.1 Main Sequence

The following diagram shows video data processing path.

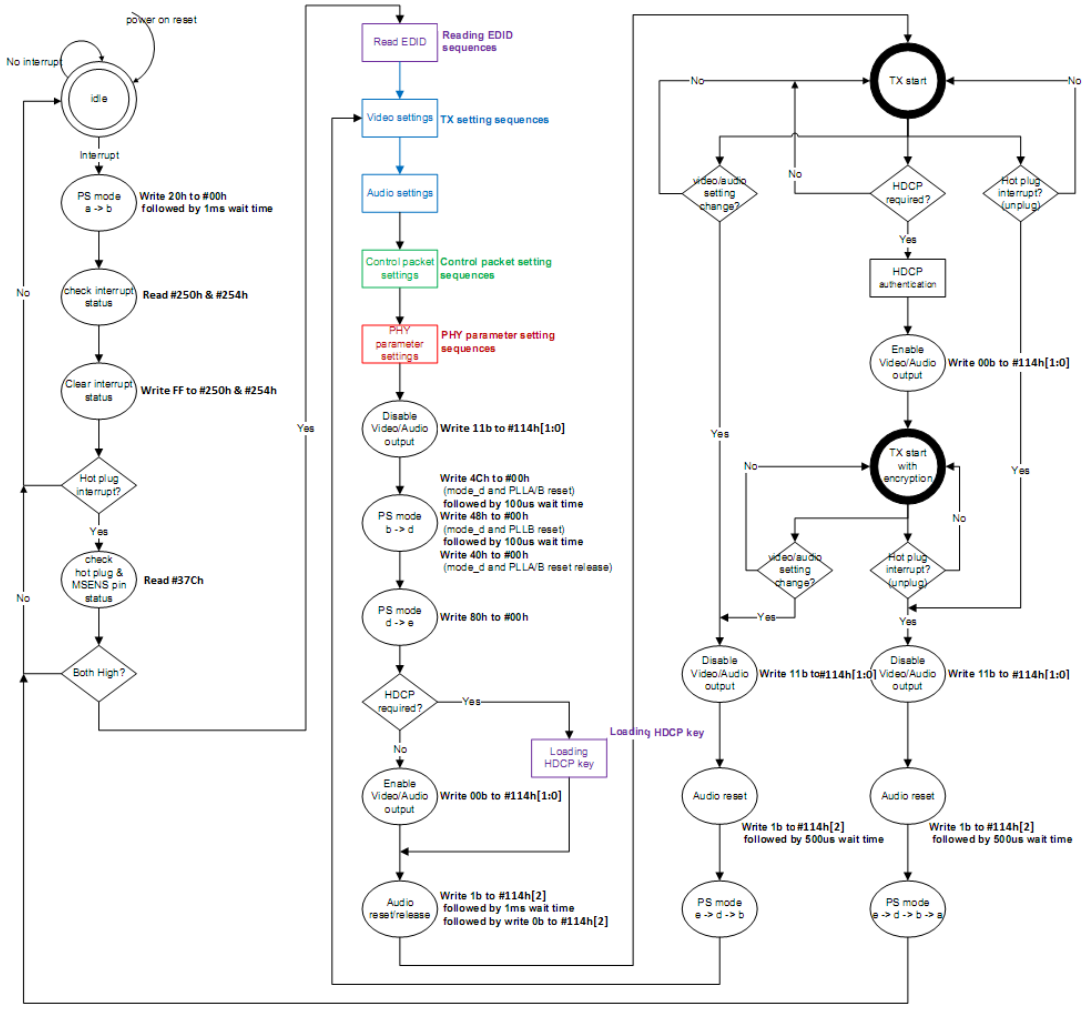


Fig. 24-14HDMI TX Software Main Sequence Diagram

In this diagram, "PS mode" means Power save mode and "#xxh" means register address xxh. Software works based on interrupt signaling from HDMI TX. After power on reset, host controller waits interrupt which indicates normally hot plug in and MSENS in. Once it is detected, host controller shall change power save mode of HDMI TX to mode_b in order to access all of registers. Then, it starts to read EDID from receiver. Based on EDID information, host controller shall set video/audio parameters as well as PHY parameters. Then it will start TMDS transmitting by change power save mode, mode_b->mode_d-> mode_e. Note that at entering into mode_d, host controller shall clear bit.2 in #00h, then clear bit.3 in #00h to release reset of PLLA and PLLB. If HDCP encryption is required, host controller shall start HDCP authentications. The followings are more detail of each sub sequences.

Note that idclk clock input shall be stable at PHY parameter settings in mode_b. Also, Video/Audio input shall be stable before entering PS mode_d. And After setting Video/Audio/PHY parameter and entering into mode_e, host controller shall set and clear bit2 in #114, for Audio reset.

When HDCP is required, it is recommended that host controller disable Video/Audio output before entering to mode_e by setting bit.0-1 of #114 register that makes black video output and no audio output. Then, host controller resets these bits after HDCP authentication has been completed and encryption has started. These sequences prevent none encrypted Video/Audio are displayed until HDCP authentication has been completed.

When hot plug interrupt by unplug is detected in "TX start"/"TX start with encryption" state, it is required to write 11b to #114[1:0] to disable video/audio output first. Also, required to write 1b to #114[2] to do audio reset. Then PS mode shall move to mode_d followed by mode_b, mode_a sequentially. Afer that, the state will return to "idle".

When video/audio settings need to be changed in "TX start"/"TX start with encryption" state, it is also required to write 11b to #114[1:0] to disable video/audio output first. Also, required to write 1b to #114[2] to do audio reset. Then PS mode shall move to mode_d followed by mode_b sequentially. Afer that, the state will return to "video settings".

Note that audio reset sequence (i.e. write 1b to #114[2]) is required just before getting out of power save mode_e.

24.6.2 Hot Plug Detection Sequences

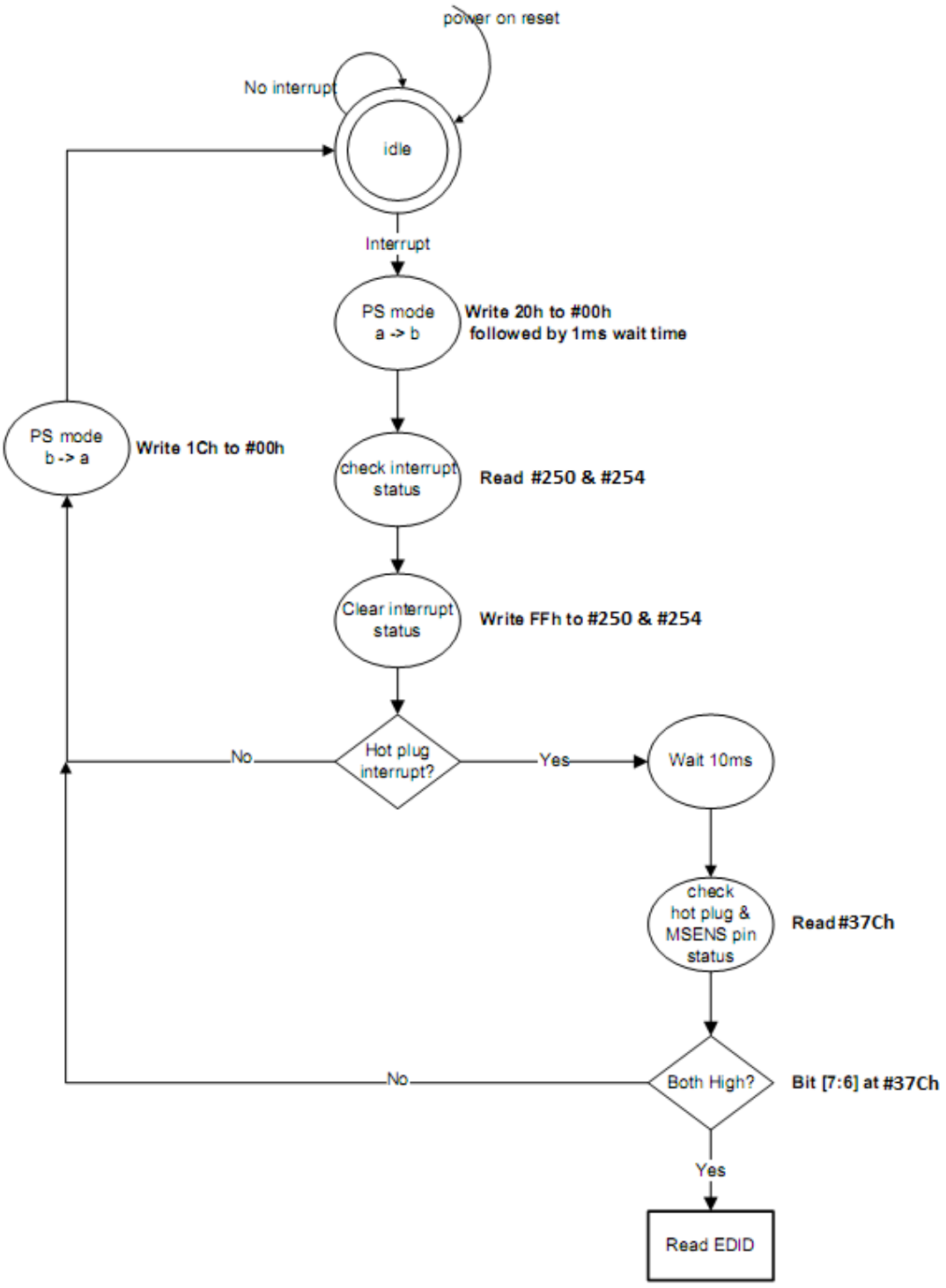


Fig. 24-15HDMI TX Hot Plug Sequence Diagram

When the host controller detects an interrupt at idle state (i.e. power save mode_a), it shall change the power save mode to mode_b and read interrupt status registers. If not hot plug interrupt, it shall change power save mode back to mode_a and wait for the next interrupt. When hot plug interrupt is detected, wait for 10 ms in order to give sufficient time to get MSENS condition, then read

DFh register for hot plug and MSENS status (bit 7 and 6). If those two bits are high, move to the Reading EDID sequence.

Note that the source of interrupt signaling is varied depending on power save mode. Refer to 1.3.3.1 Interrupt signaling for detail.

24.6.3 Reading EDID Sequence

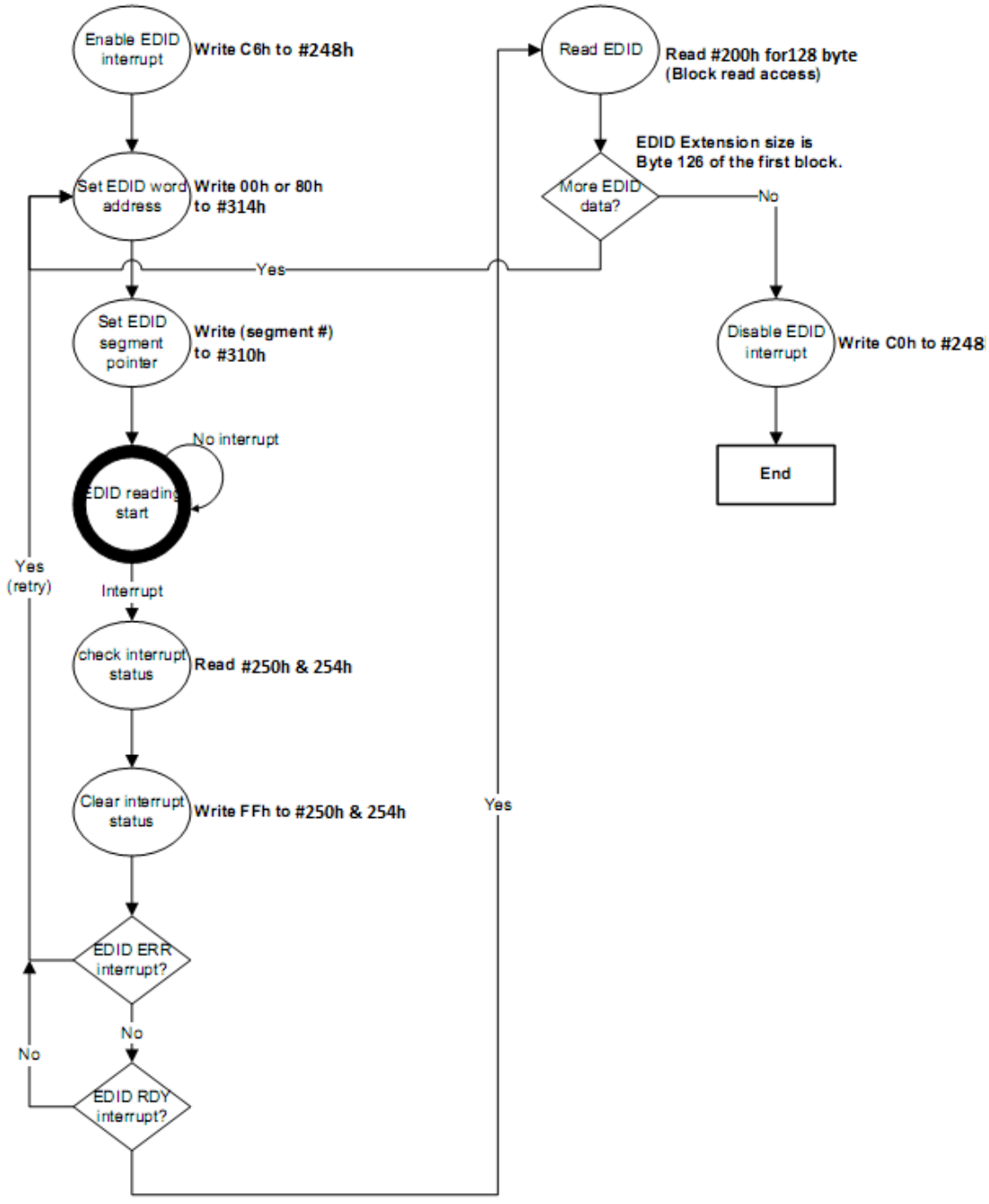


Fig. 24-16HDMI TX Software EDID Read Sequence Diagram

EDID reading sequence needs to be repeated several times in order to read all EDID blocks from the connected device. After setting EDID interrupt enable, set EDID word address and segment pointer to select which EDID block to read. Writing segment pointer at register C4h starts the EDID reading. When EDID is read correctly, EDID_RDY interrupt is generated. Otherwise, EDID_ERR interrupt is generated and host shall retry EDID reading by setting the same word address and segment pointer. If successful, move to the next EDID block until it reads all blocks, and then disable the EDID interrupt.

24.6.4 TX Setting Sequence

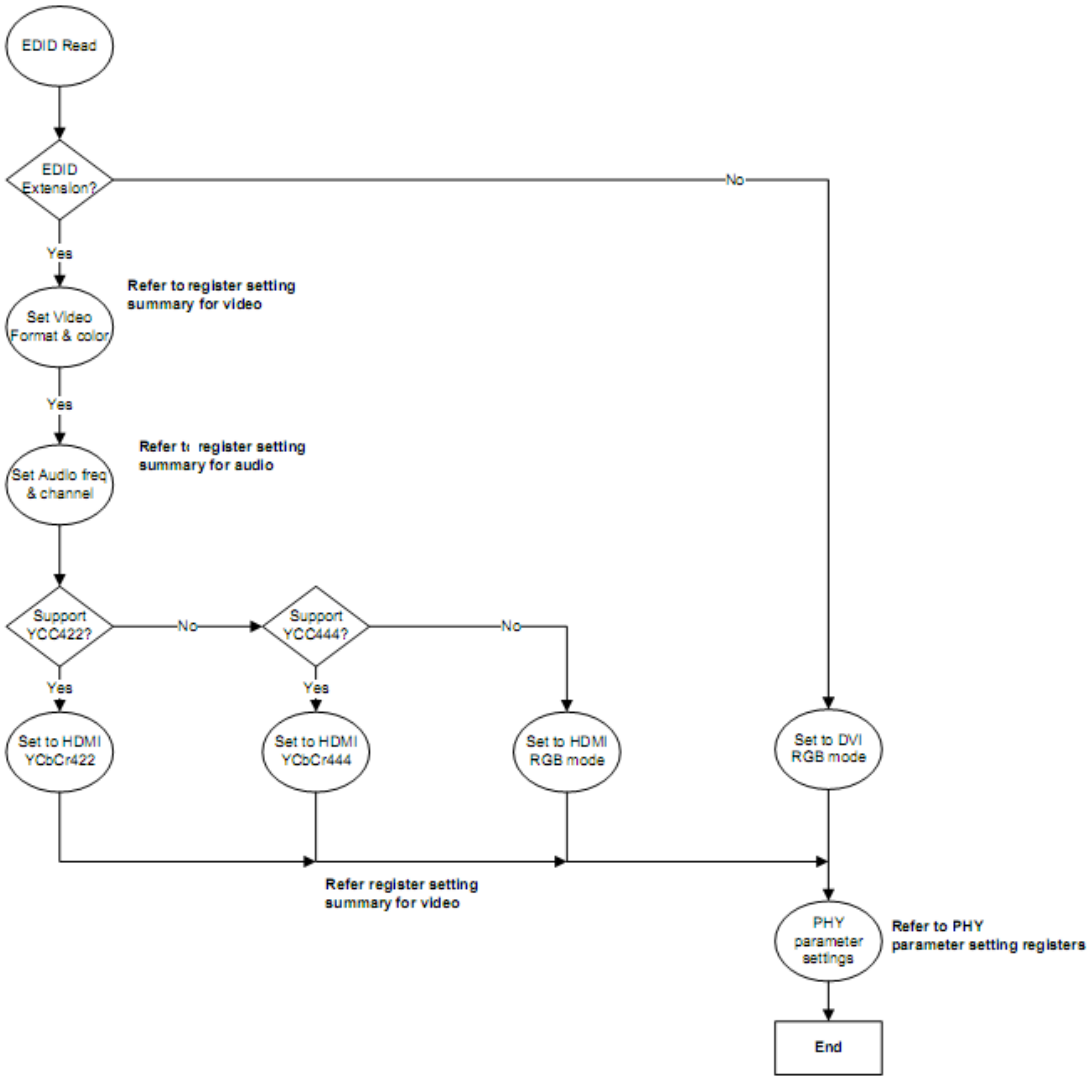


Fig. 24-17HDMI TX Software TX SettingSequence Diagram

23.6.4.1 Register Setting Summary for Video

The following registers shall be set to enable video input. Refer to each

register description for its detail function.

Table 24-16HDMI TX Register Setting Summary for Video

Register address	Bit	Setting value explanation	Example settings		
			480p	720p	1080i
#54h	[3:1] [0]	Input video format Internal/External DE select	3'b000 1'b0	3'b000 1'b0	3'b000 1'b0
#58h	[7:6] [5:4] [3:2] [0]	Video output format Input video data width EAV/SAV location if embedded sync video input selected Video input color space	2'b00 2'b11 3'b010 1'b0	2'b00 2'b11 3'b010 1'b0	2'b00 2'b11 3'b010 1'b0
#5Ch	[7:6]	Deep color mode	2'b00	2'b00	2'b00
#60 - #BC	[7:0]	Color space conversion parameters if required	N.A.	N.A.	N.A.
#C0 - #E8 #F4 - #FC	[7:0]	External video parameter settings if it is enable by #30[0], i.e. non-preprogrammed VID used.	#30[0]=1'b0 Other bits are N.A.		
#ECh	[0]	CSC enable if required	1'b0	1'b0	1'b0
#114h	[7:6]	AV mute control if required	2'b00	2'b00	2'b00
#148 - #154	[7:0]	SYNC placement it ITU656 video input mode selected	N.A.	N.A.	N.A.
#17C - #1F8	[7:0]	Appropriate AVI InfoFrame data	Refer to AVI InfoFrame descriptions in 861D spec		
#34C	[7:0]	CSC configuration if required	N.A.	N.A.	N.A.
#350	[7:4]	ITU656 control if required	N.A.	N.A.	N.A.

Note that example setting conditions are as below.

480p : 720x480p@59.94Hz(VID=2), Pre-programmed VID used, 8bit color depth, RGB input/output, CSC diable, I2S audio, fs=48KHz, 2Ch

720p : 1280x720p@60Hz(VID=4), Pre-programmed VID used, 8bit color depth, RGB input/output, CSC diable, I2S audio, fs=48KHz, 2Ch

1080i : 1920x1080i@60Hz(VID=5), Pre-programmed VID used, 8bit color depth, RGB input/output, CSC diable, I2S audio, fs=48KHz, 6Ch

23.6.4.2 Register Setting Summary for Audio

The following registers shall be set to enable audio input. Refer to each register description for its detail function.

Table 24-17HDMI TX Register Setting Summary for Audio

Register address	Bit	Setting value explanation	Affect for audio source	Example settings		
				480p	720p	1080i
#04	[7:4] [3:0]	L/R data swap control if required appropriate N[19:16]	All	0h 0h	0h 0h	0h 0h
#08	[7:0]	appropriate N[15:8]	All	18h	18h	18h
#0C	[7:0]	appropriate N[7:0]	All	00h	00h	00h
#1C	[3:0]	static CTS[19:16] if external CTS is selected by bit.7 of #0A register	All	N.A.	N.A.	N.A.
#20	[7:0]	static CTS[15:8] if external CTS is selected by bit.7 of #0A register	All	N.A.	N.A.	N.A.
#24	[7:0]	static CTS[7:0] if external CTS is selected by bit.7 of #0A register	All	N.A.	N.A.	N.A.
#28	[7]	set "1" if external CTS shall be used.	All	1'b0	1'b0	1'b0
	[6:5]	set required down sampling rate if required	I2S/SPDIF(LPCM)	N.A.	N.A.	N.A.
	[4:3]	set required audio source	All	2'b00	2'b00	2'b00
	[2]	set "1" if MCLK shall be used for CTS calculation	All	1'b0	1'b0	1'b0
	[1:0]	set input MCLK ratio if bit.2 is set by "1"	All	N.A.	N.A.	N.A.
#2C	[3:0]	set sending channel number for channel status	I2S/HBR/ SPDIF (LPCM down sampling)	0h	0h	0h
#30	[5:2]	set valid I2S source input pin	I2S/HBR	1h	1h	7h
	[1:0]	set input I2S source mode	I2S/HBR	2'b00	2'b00	2'b00
#34	[7:4]	set valid DSD source input pin	DSD	0h	0h	0h
#40	[7:0]	set appropriate I2S input pin swap settings if required	I2S/HBR	0h	0h	0h
#44	[7]	set validity bit for channel status	I2S/HBR/ SPDIF (LPCM down sampling)	1'b0	1'b0	1'b0
	[3:0]	set original sample frequency for channel status		0h	0h	0h
#48	[7]	set value for channel status bit.1		1'b0	1'b0	1'b0
	[6]	set value for channel status bit.0		1'b0	1'b0	1'b0
	[5]	set copyright bit for channel status		1'b0	1'b0	1'b0
	[4:2]	set additional information for channel status		0h	0h	0h
	[1:0]	set clock accuracy for channel status		2'b00	2'b00	2'b00
#4C	[7:0]	set category code for channel status		00h	00h	00h
#50	[7:4]	set source number for channel status		0h	0h	0h
	[3:0]	set word length for channel status		0h	0h	0h
#54	[7:4]	set sample frequency for channel status		2h	2h	2h
#17C - #1F8	[7:0]	Appropriate Audio InfoFrame data	All	Refer to Audio InfoFrame descriptions in 861D spec		

Note that example setting conditions are as below.

480p : 720x480p@59.94Hz(VID=2), Pre-programmed VID used, 8bit color depth, RGB input/output, CSC diable, I2S audio, fs=48KHz, 2Ch

720p : 1280x720p@60Hz(VID=4), Pre-programmed VID used, 8bit color depth, RGB input/output, CSC diable, I2S audio, fs=48KHz, 2Ch

1080i : 1920x1080i@60Hz(VID=5), Pre-programmed VID used, 8bit color depth, RGB input/output, CSC diable, I2S audio, fs=48KHz, 6Ch

23.6.4.3 PHY parameter settings

Host controller shall set appropriate PHY parameters into the following registers in power save mode_b based on required pixel clock frequency and color depth. Refer to 1.4 for detail phy parameter registers. Note that PHY parameter setting values shall be varied depending on process technology.

Table 24-18HDMI TX PHY Parameter Setting

- **PHY parameters for TSMC65G/55G process**

HDMI Color depth	Example Video Resolution	IDCLK Freq [MHz]	TMDS_CK Freq [MHz]	Register address									
				#5Ch	#158	#15C	#160	#164	#168	#16C	#170	#174	#178
8bit	480p, 480i, 576p	27.00	27.00	22	11	00	00	44	32	4B	0E	70	00
	1080i, 720p	74.25	74.25	22	19	00	00	44	32	48	0E	70	00
	1080p	148.50	148.50	22	1D	00	00	4C	1E	47	0E	70	00
10bit	480p, 480i, 576p	27.00	33.75	62	15	00	00	48	32	46	0E	70	00
	1080i, 720p	74.25	92.81	62	19	00	00	44	32	48	0E	70	00
	1080p	148.50	185.63	62	1D	00	00	4C	1E	48	0E	70	00
12bit	480p, 480i, 576p	27.00	40.50	A2	15	00	00	48	32	48	0E	70	00
	1080i, 720p	74.25	111.38	A2	19	00	00	44	32	4B	0E	70	00
	1080p	148.50	222.75	A2	1D	04	00	4C	1E	7F	0E	72	00

The following indicates PHY parameter setting sequences.

(1) Write 20h to register #00. (i.e. power save mode_b).

If the current power save mode is mode_b, it can be omitted. At this point, a stable idclk input is required.

(2) Write appropriate parameter value to register #158 according to output video resolution.

(3) Write 2Ch to register #00 (i.e. PLLA/B reset) and wait 100us.

(4) Write 20h to register #00 (i.e. PLLA/B reset release) and wait 1ms for PLL lock.

(5) Repeat above (2) - (4) for register #160.

(6) Repeat above (2) - (4) for register #164.

(7) Repeat above (2) - (4) for register #168.

(8) Repeat above (2) - (4) for register #16C.

(9) Repeat above (2) - (4) for register #170.

(10) Repeat above (2) - (4) for register #174.

(11) Repeat above (2) - (4) for register #178.

(12) Write appropriate parameter value to register #158 according to output video resolution.

(13) Change idclk input frequency if changing output video resolution.

(14) Write 2Ch to register #00 (i.e. PLLA/B reset) and wait 100us.

(15) Write 20h to register #00 (i.e. PLLA/B reset release) and wait 1ms for PLL lock.

(16) Write 4Ch to register #00. (i.e. power save mode_d with PLLA/B reset)

and wait 100us.

(17) Write 48h to register #00. (i.e. power save mode_d with PLLB reset) and wait 100us.

(18) Write 40h to register #00. (i.e. power save mode_d) and wait 100us.

24.6.5 Control Packet

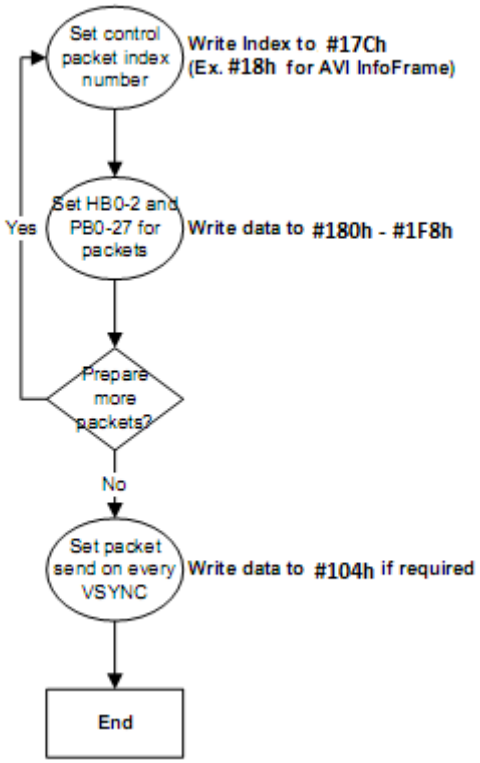


Fig. 24-18HDMI TX Software Control Packet Sequence Diagram

There are two controls to send control packets.

(1) Controll packets shall be sent once at every vertical sync. Host controller shall program it to #104h in setting sequences as above.

(2) Control packet can be sent once manually. Host controller can set bits in #100h to send a packet manually at any timing in power save mode_e if packet data is prepared.

24.6.6 Loading HDCP Key

For preparing HDCP authentication, HDCP key shall be loaded from external HDCP memory.

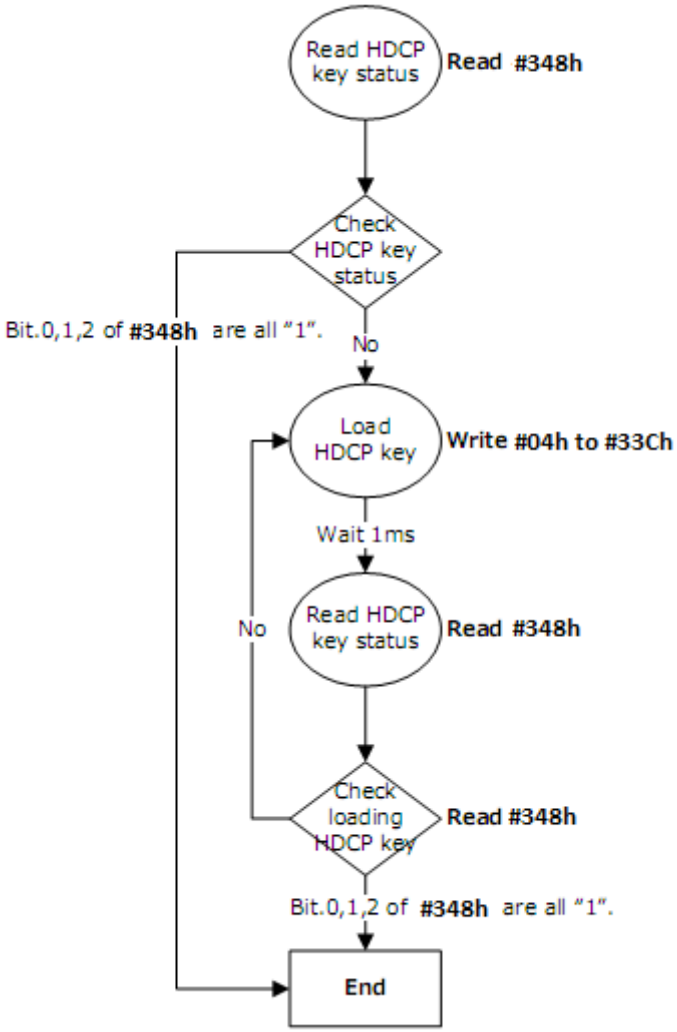


Fig. 24-19HDMI TX Loading HDCP Key Sequence Diagram

24.6.7 Hardware HDCP Authentication

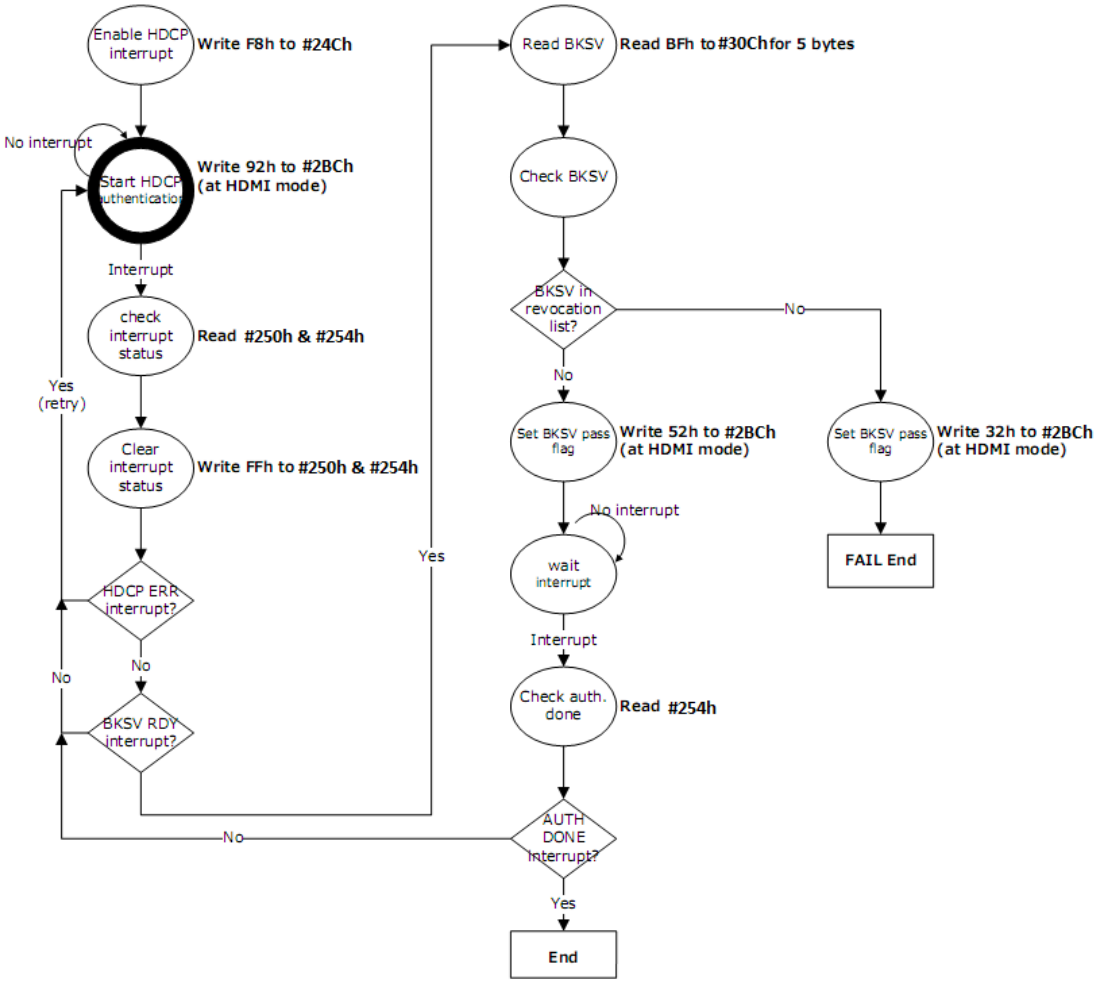


Fig. 24-20HDMI TX Hardware HDCP Authentication Sequence Diagram

24.6.8 Software HDCP Authentication

23.6.8.1 Enabling Software HDCP Authentication

To enable software authentication, set en_soft_auth (268h bit7) bit. HW will generate sf_mode_ready interrupt (260h bit7) to notify software that it is ready to start software authentication.

23.6.8.2 First Part of Authentication

Software sequences the authentication process by reading from and writing to receiver to HDCP port via DDC accesses. DDC accesses are performed by accessing I2C registers (278h-2B8h).

For example, to read Ri via DDC, set i2c_length(278h) to 2, i2c_offset(280h)

to 8, and i2c_read(284h bit0) to 1. A successful completion is notified by i2c_ack interrupt (264h bit 7) and a failure generates i2c_err_ack interrupt (264h bit 6). When the interrupt is generated, Ri is available from i2c_rd_buf registers (288h-28Ch).

For another example, to write Aksv via DDC, set i2c_length(278h) to 5, i2c_offset(280h) to 10h, load Aksv contents to i2c_wr_buf(29Ch-2ACh), and set i2c_write(284H bit2) to 1. A successful completion is notified by i2c_ack interrupt (264h bit 7) and a failure generates i2c_err_ack interrupt (264h bit 6).

In the first part of authentication, software needs to read Bksv and Bcaps from DDC. After obtaining these values, they need to be written back into registers, 38Ch-39Ch and 380h respectively. Also, depending on Bcaps value, set_repeater bit in 268h may be set.

To send An, a random number must be generated using cipher by setting prep_an bit in 268h register.

To generate the first Ri, set start_auth bit in 268h after loading Bksv, Bcaps, set_repeater bit, and generating An. If start_auth bit is set before any of these values are set, a wrong Ri will be generated and the authentication will result in failure.

According to HDCP specification, Ri must not be read within 100ms of writing Aksv. Therefore, the S/W sequencer shall implement a timer to wait for 100ms before reading Ri via DDC.

If for some reason, the S/W sequencer determines that the authentication has failed, S/W sequencer shall clear all bits in Software HDCP Control (268h) register including en_soft_auth bit (bit7), return to the idle state, and retry the authentication.

23.6.8.3 Second Part of Authentication

A 5 second timer shall be implemented by S/W sequencer to time out from waiting for READY bit in Bcaps register. S/W sequencer shall retry reading Bcaps until either the READY bit is set or the 5 seconds has passed.

What software does for the second part of authentication is very similar to the first part. When software reads Bstatus, the value needs to be written into the register (384h-388h). Also, num_devices (370h) needs to be written as well.

After reading Bstatus, software reads KSV List via DDC. Because the size of KSV List is much larger than i2c_rd_buf can take, set i2c_no_read bit as well as i2c_read bit in A1h register. When the DDC access is complete,KSV List is ready for block read out via 200h register. Software needs to set calc_sha1 bit in 268h to enable hash calculation. Hash values will be loaded into SHA0~SHA4 registers when they are ready (sha1_ready interrupt in 260h).

Use sha1_index (360h) to read out values from the registers.

23.6.8.4 Third Part of Authentication

There are 2 ways to perform the 3rd part of authentication. The simplest way is to use the frame counter implemented in the hardware. The frame counter is a counter that keeps track of when to update Ri and Pj values for link integrity checks.

1. Hardware Frame Counter

The hardware maintains a frame counter and is displayed at 270h, although, software may not need to read this value at all. In this mode, software must enable ri_save_ready and pj_save_ready interrupts. The software may disable ri_ready, pj_ready, enc_en, enc_dis_avmute, enc_dis_no_avmute, fr_cnt_update interrupts as they are not used in this mode.

When set_auth bit in 268h is set, the hardware generates ri_save_ready interrupt when Ri gets updated at 128th frame. The hardware also generates pj_save_ready interrupt when Pj gets updated at 16th frame. When ri_save_ready interrupt is generated, software shall read Ri_save register and Ri value from the receiver and compare them. If they do not match, the software shall clear set_auth and en_soft_auth bits and re-authentication from the beginning. When pj_save_ready interrupt is generated, software shall read Pj_save register and Pj value from the receiver and compare them.

ri_save_ready interrupt is generated at every 128th frame regardless of RiFrameCount (2C8h) register value and Ri_save (364h-368h) remains unchanged for 128 frames. The S/W sequencer shall check for Ri values within the 128 frame window. For 60Hz systems, 128 frames give about 2 seconds for progressive video formats and 1 second for interlaced video formats.

Similarly, pj_save_ready interrupt is generated at every 16th frame. The S/W sequencer shall check for Pj values within the 16 frame window.

2. Software Frame Counter

In this mode, software must enable ri_ready, pj_ready, enc_en, enc_dis_avmute, and enc_dis_no_avmute interrupts. The software may disable ri_save_ready, pj_save_ready, and fr_cnt_update interrupts.

In this mode, software keeps track of frame counter by counting number of enc_en, enc_dis_avmute, and enc_dis_no_avmute interrupts.

When set_auth bit in 268h is set, the hardware generates ri_ready interrupt when Ri gets updated every frame.

Software keeps the 128th Ri for the next 128 frames for comparison with the receiver. Also, the software keeps the 16th Pj for the next 16 frames for

comparison.

In this mode, the S/W HDCP sequencer's behavior is more flexible. S/W sequencer must save and keep Ri values for 128 frames, but it may check for Ri at any frame.

Chapter 25 Camera Interface

25.1 Overview

The Camera interface, receives the data from Camera or CCIR656 encoder, and transfers the data into system main memory by AXI bus.

The features of camera interface are as follow:

- Support YCbCr422 input
- Support Raw8/10/12 bit input
- Support CCIR656(PAL/NTSC) input
- Support JPEG input
- Support YCbCr422/420 output
- Support UYVY/VYUY/YUYV/YVYU configurable
- Support up to 8192x8192 resolution source
- Support picture in picture
- Support arbitrary size window crop
- Support scale range from 1/8 to 8, and destination width up to 1920
- Support SCM with configurable statistics cycles
- Support white balance
- Support error/terminate interrupt and combined interrupt output
- Support clk/vsync/href polarity configurable
- Support one frame stop/ping-pong/line loop mode

25.2 Block Diagram

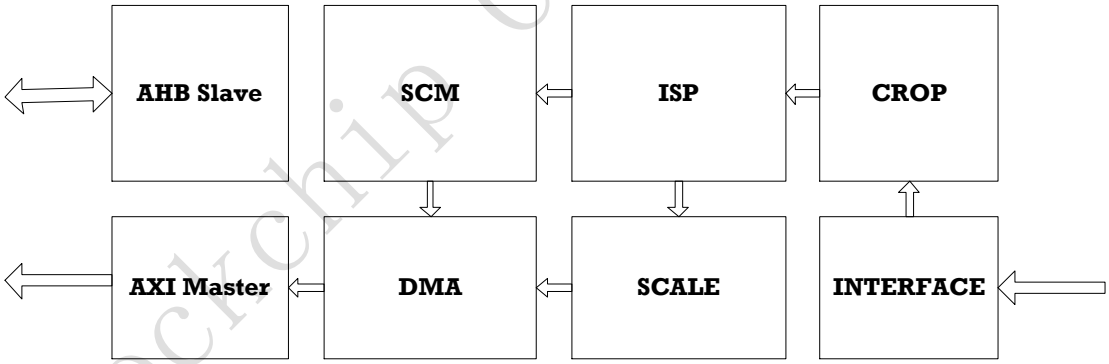


Fig. 25-1CIF block diagram

The CIF comprises with:

- AHB Slave
  Host configure the registers via the AHB Slave
- AXI Master
  Transmit the data to chip memory via the AXI Master
- INTERFACE
  Translate the input video data into the requisite data format
- CROP
  Bypass or crop the source video data to a smaller size destination
- ISP
  Modify the pixel value for more good visual effect
- SCM

Collect the statistics of the pixel value

- SCALE
  Scale up or down the input video data
- DMA
  Control the operation of AXI Master

25.3 Function description

This chapter is used to illustrate the operational behavior of how CIF works. CIF receive the sensor, ccir656 signal from external devices and translate it into YUV422/420 data, separate the data to Y and UV data, then store them to different memory via AXI bus separately.

Input data format

The CIF module support the 8bit YUV422 and CCIR656, 10/12-bit raw data input.

1. Support Vsync high active or low active
   - Vsync Low active as below:

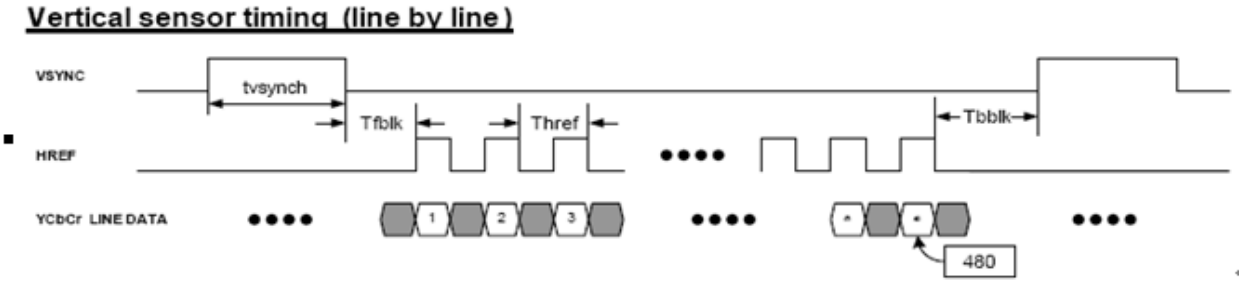


Fig. 25-2 Timing diagram for CIF when vsync low active

   - Vsync High active :

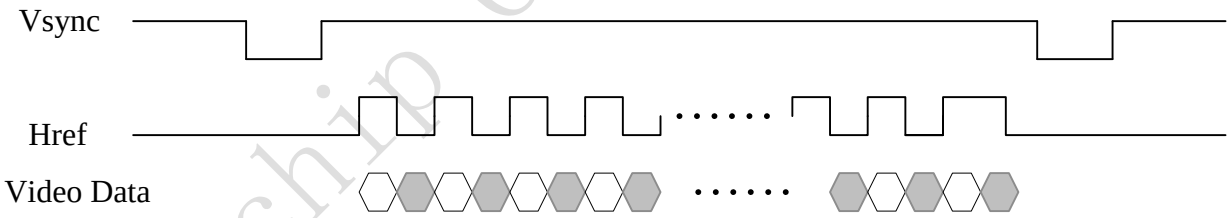


Fig. 25-3Timing diagram for CIF when vsync high active

2. Support href high active or low active
   - Href high active:

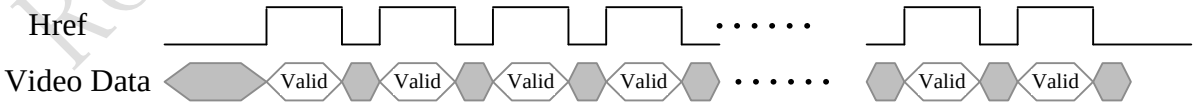


Fig. 25-4Timing diagram for CIF when href high active

   - Href Low active

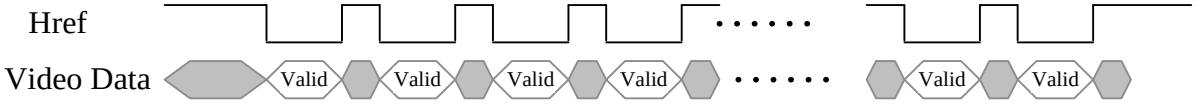


Fig. 25-5Timing diagram for CIF when href low active

- Y first

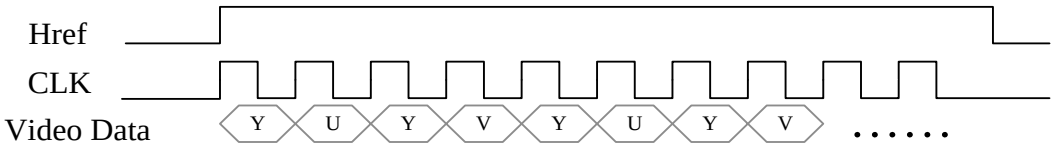


Fig. 25-6Timing diagram for CIF when Y data first

- U first

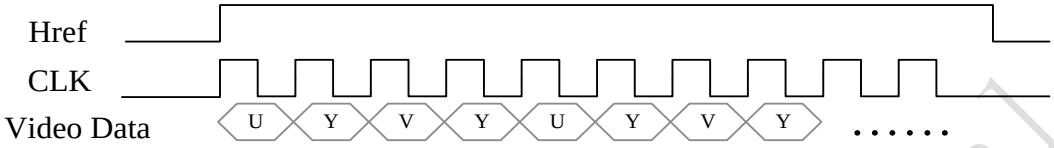


Fig. 25-7Timing diagram for CIF when U data first

3. Support CCIR656 (NTSC and PAL)

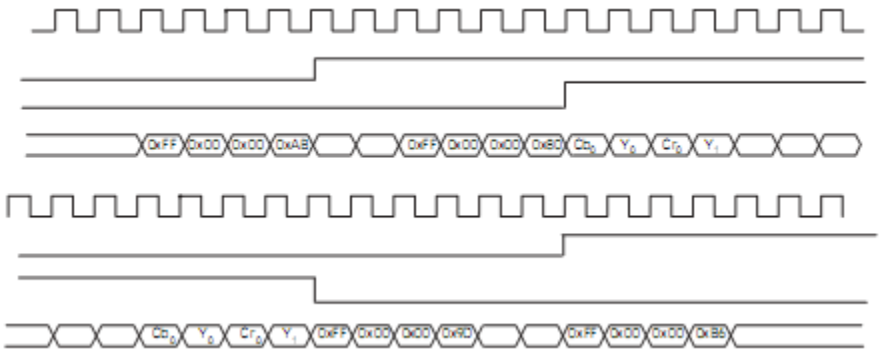


Fig. 25-8CCIR656 timing

4. Support Raw data(10/12-bit) or JPEG

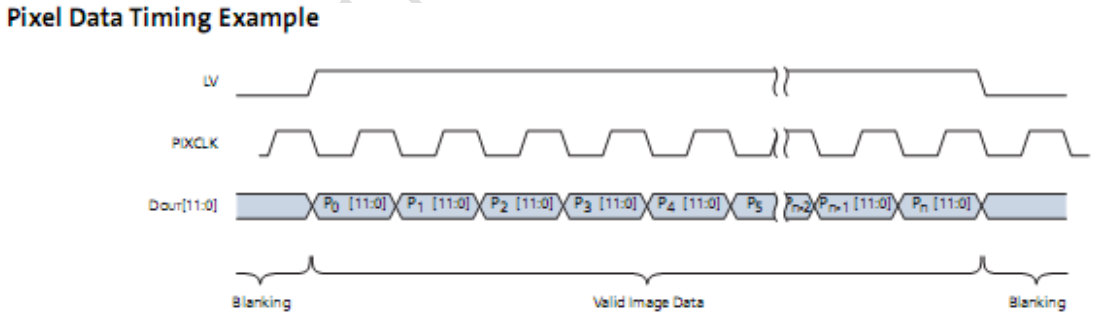


Fig. 25-9Raw Data or JPEG Timing

CIF module can work in three modes: one frame stop mode, ping-pong mode and Line Loop mode.

One frame stop mode

In this mode, configure the parameter WORK_MODE to one frame stop mode. After one frame captured, CIF will automatic stop. After capturing, the image Y, UV data will be stored at main memory location defined by CIF_FRM0_ADDR_Y, FRM0_ADDR_UV separately.

Ping-Pong mode

After one frame(F1) captured, CIF will start to capture the next frame(F2) automatically, and host must assign new address pointer of frame1 and clear the frame1 status, thus CIF will capture the third frame automatically(by new F1 address) without any stop and so on for the following frames. But if host did not update the frame buffer address, the CIF will cover the pre-frame data stored in the memory with the following frame data.

Line Loop mode

In this mode, LINE_LOOP_NUM can be set to decide when to switch the ping-pang buffer address at the middle of a frame.For example, if this parameter is equal to ten,the first ten lines is written to FRM0_ADDR_Y and FRM0_ADDR_UV. And the second ten lines are written to FRM1_ADDR_Y and FRM1_ADDR_UV.The third ten lines, the fourth ten lines, it runs like this in circulations.

Assume that there are 384 lines in the frame, the FRAME_END_SWITCH can be set to switch the frame buffer address at the end of the frame although less than ten lines written in the frame buffer.But if the FRAME_END_SWITCH bit is not set, the address switching operation will occur after the first six lines in the next frame are written in the current frame buffer.

The current line number of the frame can be writtten to LIN_NUM_ADDR if ISSUE_LINE_NUM_MODE is enabled.

Storage

Difference between the YUV mode and raw mode is that in the YUV mode or ccir656 mode, data will be storage in the Y data buffer and UV data buffer; but in the raw or jpeg mode, RGB data will be storage in the same buffer. In addition, in the yuv mode, the width of Y, U or V data is a byte in memory; in Raw or JPEGE mode, the width is a halfword no matter the data source is 8 bit, 10 bit or 12 bit.

CROP

The parameter START_Y and START_X defines the coordinate of crop start point.And the frame size after cropping is following the value of SET_WIDTH and SET_HEIGHT.

ISP

The ISP LUT RAM address offset is range from 0x1000 to 0x13FC and bits [1:0] must be 2'b0. There are 256 RAM entries and [23:16], [15:8], [7:0] bits of data in each entry correspond to Y, U, V separately.

The function of ISP operation is mapping the input YUV value to output YUV value based on the ISP LUT RAM. The input YUV value is the address of LUT RAM, and the ouput value is the data in the input address. After reset, there are invalid values in LUT RAM. It is recommend to initialize the LUT RAM before doing ISP operation.

The configuration flow is the following.Firstly, set the ISP_INIT_LD bit and initialize the ISP LUT RAM for the pixel value mapping. The ISP_EN can be set only if the initializing operation is ready.

SCM

Just like the ISP operation flow, the first task of SCM operation is writing 1 to SCM_INIT_LD bit. Next, it is recommended to initialize the SCM RAM to all zero.After that, SCM_EN can be enabled to start to collect the statistics of the pixel value.

The SCM frequency is also configurable.The frequency of pixel is controlled by

PIX_SAMPLE_FREQ.And the frequency of frame is controlled by FRM_SAMPLE_FREQ.

The SCM result can be read by AXI Master or AHB Slave.In AHB Slave mode,only one frame can be collected and the result will be absolutely right.But if the AXI Master mode is selected, SCM error maybe occur when the SCM information of previous frame is not reading completely but current frame is coming.

The SCM RAM address is word address and the offset is the following:

Y: 0x400~0x7FC
U: 0x800~0xBFC
V: 0xC00~0xFFC

WBC

When all of the following conditions is true, the pixels will be counted in the resiter WBC_CNT.

MAX_Y < y value < MIN_Y
MAX_U < u value < MIN_U
MAX_V < v value < MIN_V

SCALE

The scale source size is SET_WIDTH and SET_HEIGHT, the scale destination size is SCL_WIDTH and SCL_HEIGHT.And the scale factor must be set correctly ,or scale error may be occurred.

SCL_HOR_FCT = ((src_width-1)/(dst_width -1)) * 2^{12}
SCL_VER_FCT =((src_height-1)/(dst_height -1)) * 2^{12}

25.4 Register description

25.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
CIF_CIF_CTRL	0x0000	W	0x00007000	CIF control
CIF_CIF_INTEN	0x0004	W	0x00000000	CIF interrupt enable
CIF_CIF_INTSTAT	0x0008	W	0x00000000	CIF interrupt status
CIF_CIF_FOR	0x000c	W	0x00000000	CIF format
CIF_CIF_LINE_NUM_ADDR	0x0010	W	0x00000000	CIF line number address
CIF_CIF_FRM0_ADDR_Y	0x0014	W	0x00000000	CIF frame0 y address
CIF_CIF_FRM0_ADDR_UV	0x0018	W	0x00000000	CIF frame0 uv address
CIF_CIF_FRM1_ADDR_Y	0x001c	W	0x00000000	CIF frame1 y address
CIF_CIF_FRM1_ADDR_UV	0x0020	W	0x00000000	CIF frame1 uv address
CIF_CIF_VIR_LINE_WIDTH	0x0024	W	0x00000000	CIF virtual line width
CIF_CIF_SET_SIZE	0x0028	W	0x01e002d0	CIF frame set size
CIF_CIF_SCM_ADDR_Y	0x002c	W	0x00000000	CIF scm y data address
CIF_CIF_SCM_ADDR_U	0x0030	W	0x00000000	CIF scm u data address
CIF_CIF_SCM_ADDR_V	0x0034	W	0x00000000	CIF scm v data address
CIF_CIF_WB_UP_FILTER	0x0038	W	0x00000000	CIF white balance up filter
CIF_CIF_WB_LOW_FILTER	0x003c	W	0x00000000	CIF white balance low filter

Name	Offset	Size	Reset Value	Description
CIF_CIF_WBC_CNT	0x0040	W	0x00000000	CIF white balance count
CIF_CIF_CROP	0x0044	W	0x00000000	CIF crop start point
CIF_CIF_SCL_CTRL	0x0048	W	0x00000000	CIF scale control
CIF_CIF_SCL_DST	0x004c	W	0x00000000	CIF scale destination frame size
CIF_CIF_SCL_FCT	0x0050	W	0x20002000	CIF scale factor
CIF_CIF_SCL_VALID_NUM	0x0054	W	0x00000000	CIF scale valid number
CIF_CIF_LINE_LOOP_CTRL	0x0058	W	0x00000000	CIF line loop control
CIF_CIF_FRAME_STATUS	0x0060	W	0x00000000	CIF frame status
CIF_CIF_CUR_DST	0x0064	W	0x00000000	CIF current destination address
CIF_CIF_LAST_LINE	0x0068	W	0x00000000	CIF last frame line number
CIF_CIF_LAST_PIX	0x006c	W	0x00000000	CIF last line pixel number

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

25.4.2 Detail Register Description

CIF_CIF_CTRL
Address: Operational Base + offset (0x0000)
CIF control

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RW	0x0	SCM_INIT_LD scm sram initial load 0-disable 1-load enable
16	RO	0x0	reserved
15:12	RW	0x7	AXI_BURST_TYPE axi master burst type 0-15 : burst1~16
11	RW	0x0	WBC_EN white balance collect 0-disable 1-enable
10	RW	0x0	ISP_EN isp enable 0-disable 1-enable

Bit	Attr	Reset Value	Description
9	RW	0x0	ISP_INIT_LD isp lut initial enable 0-disable 1-load enable
8	RW	0x0	SCM_RD_MODE scm read mode 0-AXI master 1-AHB slave Note: when in slave mode, only collect one frame.
7:6	RW	0x0	PIX_SAMPLE_FRQ pixel scm sample frequency 00 - none 01 - every 1 uv/2 y 10 - every 2 uv/4 y 11 - every 4 uv/8 y
5:4	RW	0x0	FRM_SAMPLE_FRQ frame SCM sample frequency 000 - every 1 frame 001 - every 2 frame 010 - every 3 frame 011 - every 4 frame
3	RW	0x0	SCM_EN SCM enable 0-SCM disable 1-SCM enable
2:1	RW	0x0	WORK_MODE Working Mode 00-one frame stop mode 01-ping-pong mode 02-line loop mode 03-reserved
0	RW	0x0	CAP_EN capture enable 0-disable 1-enable

CIF_CIF_INTEN
Address: Operational Base + offset (0x0004)
CIF interrupt enable

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7	RW	0x0	SCL_ERR_EN scale error 0-disable 1-enable
6	RW	0x0	BUS_ERR_EN bus error axi master or ahb slave response error 0-disable 1-enable
5	RW	0x0	SCM_ERR_EN scm error scm start when the last scm data have not be send 0-disable 1-enable
4	RW	0x0	LINE_BUF_OVER_EN line buffer overflow 0-disable 1-enable
3	RW	0x0	PIX_ERR_EN pixel err interrupt enable the pixel number of last line not equal to the set height 0-disable 1-enable
2	RW	0x0	LINE_ERR_EN line err interrupt enable the line number of last frame not equal to the set height 0-disable 1-enable
1	RW	0x0	LINE_END_EN line end interrupt enable 0-disable 1-enable
0	RW	0x0	FRAME_END_EN frame end interrupt enable after dma transfer the frame data 0-disable 1-enable

CIF_CIF_INTSTAT

Address: Operational Base + offset (0x0008)
CIF interrupt status

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	W1C	0x0	SCL_ERR scale error 0-no interrupt 1-interrupt
6	W1C	0x0	BUS_ERR bus error axi master or ahb slave response error 0-no interrupt 1-interrupt
5	W1C	0x0	SCM_ERR scm error scm start when the last scm data have not be send 0-no interrupt 1-interrupt
4	W1C	0x0	LINE_BUF_OVER line buffer overflow 0-no interrupt 1-interrupt
3	W1C	0x0	PIX_ERR pixel err interrupt the pixel number of last line not equal to the set height 0-no interrupt 1-interrupt
2	W1C	0x0	LINE_ERR line err interrupt the line number of last frame not equal to the set height 0-no interrupt 1-interrupt
1	W1C	0x0	LINE_END line end interrupt enable 0-no interrupt 1-interrupt
0	W1C	0x0	FRAME_END frame end interrupt after dma transfer the frame data 0-no interrupt 1-interrupt

CIF_CIF_FOR
Address: Operational Base + offset (0x000c)
CIF format

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RW	0x0	UV_STORE_ORDER UV storage order 0 - UVUV 1 - VUVU
18	RW	0x0	RAW_END raw data endian 0 - little end 1 - big end
17	RW	0x0	OUT_420_ORDER output 420 order 00 - UV in the even line 01 - UV in the odd line Note: The first line is even line(line 0).
16	RW	0x0	OUTPUT_420 output 420 or 422 0 - output is 422 1 - output is 420
15	RO	0x0	reserved
14:13	RW	0x0	MIPI_MODE mipi mode 00 - bypass 01 - rgb 10 - yuv 11- reserved
12:11	RW	0x0	RAW_WIDTH raw data width 00 - raw8 01 - raw10 10 - raw12 11 - reserved
10	RW	0x0	JPEG_MODE JPEG mode 0 - other mode 1 - mode1
9	RW	0x0	FIELD_ORDER ccir input order 0-odd field first 1-even field first

Bit	Attr	Reset Value	Description
8	RW	0x0	IN_420_ORDER 420 input order 00 - UV in the even line 01 - UV in the odd line Note: The first line is even line(line 0).
7	RW	0x0	INPUT_420 input 420 or 422 0 - 422 1 - 420
6:5	RW	0x0	YUV_IN_ORDER YUV input order 00 - UYVY 01 - YVYU 10 - VYUY 11 - YUYV
4:2	RW	0x0	INPUT_MODE input mode 000 - YUV 010 - PAL 011 - NTSC 100 - RAW 101 - JPEG 110 - MIPI Other - invalid
1	RW	0x0	HREF_POL href input polarity 0-high active 1-low active
0	RW	0x0	VSYNC_POL vsync input polarity 0-;ow active 1-high active

CIF_CIF_LINE_NUM_ADDR
Address: Operational Base + offset (0x0010)
CIF line number address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	LIN_NUM_ADDR line number address

CIF_CIF_FRM0_ADDR_Y
Address: Operational Base + offset (0x0014)
CIF frame0 y address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM0_ADDR_Y frame0 y address

CIF_CIF_FRM0_ADDR_UV
Address: Operational Base + offset (0x0018)
CIF frame0 uv address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM0_ADDR_UV frame0 uv address

CIF_CIF_FRM1_ADDR_Y
Address: Operational Base + offset (0x001c)
CIF frame1 y address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM1_ADDR_Y frame1 y address

CIF_CIF_FRM1_ADDR_UV
Address: Operational Base + offset (0x0020)
CIF frame1 uv address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM1_ADDR_UV frame1 uv address

CIF_CIF_VIR_LINE_WIDTH
Address: Operational Base + offset (0x0024)
CIF virtual line width

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RW	0x0000	VIR_LINE_WIDTH virtual line width

CIF_CIF_SET_SIZE
Address: Operational Base + offset (0x0028)
CIF frame set size

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x01e0	SET_HEIGHT set height
15:13	RO	0x0	reserved
12:0	RW	0x02d0	SET_WIDTH set width

CIF_CIF_SCM_ADDR_Y
Address: Operational Base + offset (0x002c)
CIF scm y data address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM_ADDR_Y frame y address

CIF_CIF_SCM_ADDR_U
Address: Operational Base + offset (0x0030)
CIF scm u data address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM_ADDR_U frame u address

CIF_CIF_SCM_ADDR_V
Address: Operational Base + offset (0x0034)
CIF scm v data address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	FRM_ADDR_V frame v address

CIF_CIF_WB_UP_FILTER
Address: Operational Base + offset (0x0038)
CIF white balance up filter

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	MAX_Y max y value
15:8	RW	0x00	MAX_U max u value
7:0	RW	0x00	MAX_V max v value

CIF_CIF_WB_LOW_FILTER
Address: Operational Base + offset (0x003c)
CIF white balance low filter

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	MIN_Y min y value
15:8	RW	0x00	MIN_U min u value

Bit	Attr	Reset Value	Description
7:0	RW	0x00	MIN_V min v value

CIF_CIF_WBC_CNT
Address: Operational Base + offset (0x0040)
CIF white balance count

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RO	0x000000	WBC_CNT white balance count

CIF_CIF_CROP
Address: Operational Base + offset (0x0044)
CIF crop start point

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RW	0x0000	START_Y start y point
15:13	RO	0x0	reserved
12:0	RW	0x0000	START_X start x point

CIF_CIF_SCL_CTRL
Address: Operational Base + offset (0x0048)
CIF scale control

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	B32_BP 32bit bypass 0-no bypass 1-bypass
5	RW	0x0	RAW_16B_BP raw 16 bit bypass 0-no bypass 1-bypass
4	RW	0x0	YUV_16B_BP YUV 16 bit bypass 0-no bypass 1-bypass
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	SCL_UP_EN scale up enable 0:disable 1:enable
0	RW	0x0	SCL_DOWN_EN scale down control 0:disable 1:enable

CIF_CIF_SCL_DST
Address: Operational Base + offset (0x004c)
CIF scale destination frame size

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:16	RW	0x000	SCL_HEIGHT scale height
15:11	RO	0x0	reserved
10:0	RW	0x000	SCL_WIDTH scale width

CIF_CIF_SCL_FCT
Address: Operational Base + offset (0x0050)
CIF scale factor

Bit	Attr	Reset Value	Description
31:16	RW	0x2000	SCL_VER_FCT vertical scale factor vrt_scl_fct =((src_height-1)/(dst_height -1)) * 2^{13}
15:0	RW	0x2000	SCL_HOR_FCT horizontal scale factor hrz_scl_fct = ((src_width-1)/(dst_width -1)) * 2^{13}

CIF_CIF_SCL_VALID_NUM
Address: Operational Base + offset (0x0054)
CIF scale valid number

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:8	RW	0x00	SCL_VALID_NUM_UV scale valid number in Y FIFO write 0 clear
7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RO	0x00	SCL_VALID_NUM_Y scale valid number in Y FIFO write 0 clear

CIF_CIF_LINE_LOOP_CTRL
Address: Operational Base + offset (0x0058)
CIF line loop control

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x0	ISSUE_LINE_NUM_MODE issue line num mode 0-diable 1-enable
4	RW	0x0	FRAME_END_SWITCH frame end switch enable 0-disable 1-enable
3:0	RW	0x0	LINE_LOOP_NUM line loop num 1~16

CIF_CIF_FRAME_STATUS
Address: Operational Base + offset (0x0060)
CIF frame status

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	FRAME_NUM complete frame number write 0 to clear
15:2	RO	0x0	reserved
1	RO	0x0	F1_STS frame 0 status 0- frame 1 not ready 1- frame 1 ready write 0 clear
0	RO	0x0	F0_STS frame 0 status 0- frame 0 not ready 1- frame 0 ready write 0 clear

CIF_CIF_CUR_DST
Address: Operational Base + offset (0x0064)

CIF current destination address

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	CUR_DST current destination address May be not the current, because the clock synchronization.

CIF_CIF_LAST_LINE
Address: Operational Base + offset (0x0068)
CIF last frame line number

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13:0	RO	0x0000	LAST_LINE_NUM line number of last frame

CIF_CIF_LAST_PIX
Address: Operational Base + offset (0x006c)
CIF last line pixel number

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RO	0x0000	LAST_PIX_NUM pixel number of last line

25.5 Timing Diagram

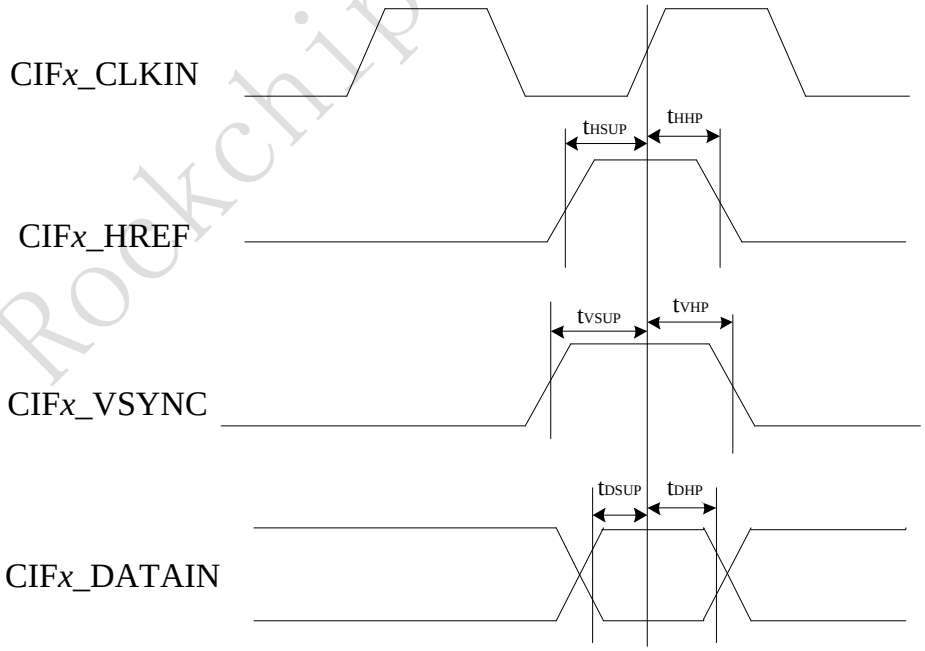


Table 25-1CIF0 Timing

Symbol	Parameter	best	Typ.	worst	unit

		case	case	case	
t_{HSUP}	Input HREF setup time to CIF_CLKIN rising edge	0.24	0.23	0.12	ns
t_{HHP}	Input HREF hold time to CIF_CLKIN rising edge	0	0	0	ns
t_{VSUP}	Input VSYNC setup time to CIF_CLKIN rising edge	0.17	0.27	0.12	ns
t_{VHP}	Input VSYNC hold time to CIF_CLKIN rising edge	0	0	0	ns
t_{DSUP}	Input DATAIN setup time to CIF_CLKIN rising edge	0.661	0.975	1.34	ns
t_{DHP}	Input DATAIN hold time to CIF_CLKIN rising edge	0	0	0	ns

x=0

Table 25-2CIF1 Timing

Symbol	Parameter	best case	Typ. case	worst case	unit
t_{HSUP}	Input HREF setup time to CIF_CLKIN rising edge	0.26	0.34	0.02	ns
t_{HHP}	Input HREF hold time to CIF_CLKIN rising edge	0	0	0	ns
t_{VSUP}	Input VSYNC setup time to CIF_CLKIN rising edge	0.28	0.10	0	ns
t_{VHP}	Input VSYNC hold time to CIF_CLKIN rising edge	0	0	0.07	ns
t_{DSUP}	Input DATAIN setup time to CIF_CLKIN rising edge	0.90	1.29	1.50	ns
t_{DHP}	Input DATAIN hold time to CIF_CLKIN rising edge	0	0	0	ns

x=1

25.6 Interface description

Module Pin	Direction	Pad Name	IOMUX Setting
CIF0			
cif0_clkout	O	GPIO1_B[3]	GRF_GPIO1B_IOMUX[6]=1'b1
cif0_clkin	I	CIF0_CLKIN	N/A
cif0_href	I	CIF0_HREF	N/A
cif0_vsync	I	CIF0_VSYNC	N/A
cif0_data0	I	GPIO1_B[4]	GRF_GPIO1B_IOMUX[8]=1'b1
cif0_data1	I	GPIO1_B[5]	GRF_GPIO1B_IOMUX[10]=1'b1
cif0_data2	I	CIF0_DATAIN[2]	N/A
cif0_data3	I	CIF0_DATAIN[3]	N/A
cif0_data4	I	CIF0_DATAIN[4]	N/A
cif0_data5	I	CIF0_DATAIN[5]	N/A
cif0_data6	I	CIF0_DATAIN[6]	N/A
cif0_data7	I	CIF0_DATAIN[7]	N/A
cif0_data8	I	CIF0_DATAIN[8]	N/A
cif0_data9	I	CIF0_DATAIN[9]	N/A
cif0_data10	I	GPIO1_B[6]	GRF_GPIO1B_IOMUX[12]=1'b1
cif0_data11	I	GPIO1_B[7]	GRF_GPIO1B_IOMUX[14]=1'b1
CIF1			

cif1_clkout	O	GPIO1_D[7]	GRF_GPIO1D_IOMUX[14]=1'b1
cif1_clkin	I	GPIO1_D[2]	GRF_GPIO1D_IOMUX[4]=1'b1
cif1_href	I	GPIO1_D[1]	GRF_GPIO1D_IOMUX[2]=1'b1
cif1_vsync	I	GPIO1_D[0]	GRF_GPIO1D_IOMUX[0]=1'b1
cif1_data0	I	GPIO1_D[3]	GRF_GPIO1D_IOMUX[6]=1'b1
cif1_data1	I	GPIO1_D[4]	GRF_GPIO1D_IOMUX[8]=1'b1
cif1_data2	I	GPIO1_C[0]	GRF_GPIO1C_IOMUX[1:0]=2'b1
cif1_data3	I	GPIO1_C[1]	GRF_GPIO1C_IOMUX[3:2]=2'b1
cif1_data4	I	GPIO1_C[2]	GRF_GPIO1C_IOMUX[5:4]=2'b1
cif1_data5	I	GPIO1_C[3]	GRF_GPIO1C_IOMUX[7:6]=2'b1
cif1_data6	I	GPIO1_C[4]	GRF_GPIO1C_IOMUX[9:8]=2'b1
cif1_data7	I	GPIO1_C[5]	GRF_GPIO1C_IOMUX[11:10]=2'b1
cif1_data8	I	GPIO1_C[6]	GRF_GPIO1C_IOMUX[13:12]=2'b1
cif1_data9	I	GPIO1_C[7]	GRF_GPIO1C_IOMUX[15:14]=2'b1
cif1_data10	I	GPIO1_D[5]	GRF_GPIO1D_IOMUX[10]=1'b1
cif1_data11	I	GPIO1_D[6]	GRF_GPIO1D_IOMUX[12]=1'b1

The valid data bits of different widths are the following:
8 bit: data2~data9;
10bit: data0~data9;
12bit: data0~data11;

25.7 Application Notes

There are two operations strongly depend on the configuration order: SCM and ISP.

The SCM configuration flow is:

a. Write 1 to SCM_INIT_LD bit to make sure the SCM RAM can be written;
b. Initialize all zero to the three 256x23 bit SCM RAM for Y/U/V statistics collection;
c. Set SCM_EN to announce SCM configuration is ready.

The ISP configuration is is very similar with SCM:

a. Write 1 to ISP_INIT_LD bit bit to make sure the ISP RAM can be written;
b. Initialize the ISP lut RAM for pixel value mapping.
c. Set ISP_EN to announce ISP configuration is ready.

The biggest configuration requirement of all operations is the CAP_EN bit must be set after all the mode selection is ready.The configuration order of the input/output data format, YUV order, the address ,frame size/width , AXI burst length and other options do not need to care.

There are many debug registers to make it easy to read the internal operation information of CIF. The valid pixel number of scale result in FIFO can be known by readCIF_CIF_SCL_VALID_NUM.The line number of last frame and the pixel number of last line can be also known by read the CIF_CIF_LAST_LINE and CIF_CIF_LAST_PIX.

Chapter 26 USB Host2.0

26.1 Overview

USB HOST2.0 supports host functions and is fully compliant with USB2.0 specification, andsupport high-speed(480Mbps),full-speed(12Mbps),low-speed(1.5Mbps) transfer. USB HOST 2.0supports high-speed(480Mbps), full-speed(12Mbps), low-speed (1.5Mbps) transfer. It is optimized for point-to-point applications (no hub, direct connection to device).

26.1.1 Features

- Compliant with the USB2.0 Specification
- Operates in host mode
- Operates in High-Speed and Full-Speed mode
- Support 16 channels in host mode
- Built-in one 840x35 bits FIFO
- Internal DMA with scatter/gather function

26.2 Block Diagram

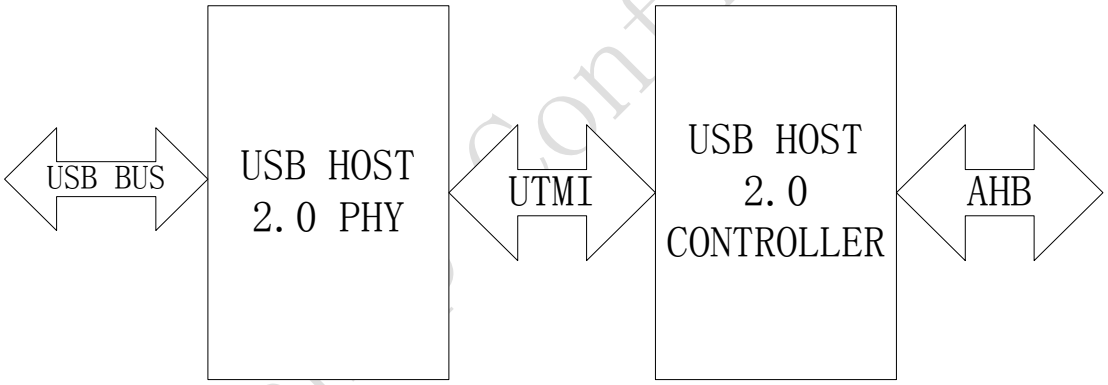


Fig. 26-1USB HOST 2.0 Architecture

Fig.26-1 shows the architecture of USB HOST 2.0. It is broken up into two separate units: USB HOST 2.0 controller and USB HOST 2.0 PHY. The two units are interconnected with UTMI interface.

26.2.1 USB HOST 2.0 Controller Function

The USB HOST 2.0 Controller controlsSIE(Serial Interface Engine) Logic , the Channel logic and the internal DMA logic.

The SIElogic contains the USB PID and address recognition logic, and other sequencingand state machine logic to handle USB packets and transactions.

The Channel Logic contains the channel tasks schedule, FIFOs and FIFO control, etc.

The internal DMA logic controls data trasaction between system memory and USB FIFOs.

26.2.2 USB HOST 2.0 PHY Function

The USB HOST 2.0 PHY handles the low level USB protocol and signaling. This

includes features such as; dataserialization and deserialization, bit stuffing and clock recovery and synchronization. The primary focus ofthis block is to shift the clock domain of the data from the USB 2.0 rate to the frequency of UTMI clock which is 30MHz.

26.3 USB Host2.0 Controller

USB HOST Controller is a usb host controller, which supports both high-speed(480Mbps), full-speed(12Mbps) and is fully compliant with USB2.0 specification. This controller will support UTMI+ Level 3 PHY interface. It connects to the industry-standard AMBA AHB for communication with the application and system memory. And it is optimized for portable electronic device and point-to-point applications (no hub, direct connection to device) .

Fig.26-2 shows the architecture of USB HOST 2.0 Controller.

The host uses one transmit FIFO for all non-periodic OUT transactions and one transmit FIFO for all periodic OUT transactions.These transmit FIFOs are used as transmit buffers to hold the data (payload of the transmit packet) to be transmitted over USB. The host pipes the USB transactions through Request queues (one for periodic and one for non-periodic). Each entry in the Request queue holds the IN or OUT channel number along with other information to perform a transaction on the USB. The order in which the requests are written into the queue determines the sequence of transactions on the USB. The host processes the periodic Request queue first, followed by the non-periodic Request queue, at the beginning of each (micro)frame. The host uses one receive FIFO for all periodic and non-periodic transactions. The FIFO is used as a receive buffer to hold the received data (payload of the received packet) from the USB until it is transferred to the system memory. The status of each packet received also goes into the FIFO. The status entry holds the IN channel number along with other information, such as received byte count and validity status, to perform a transaction on the AHB.

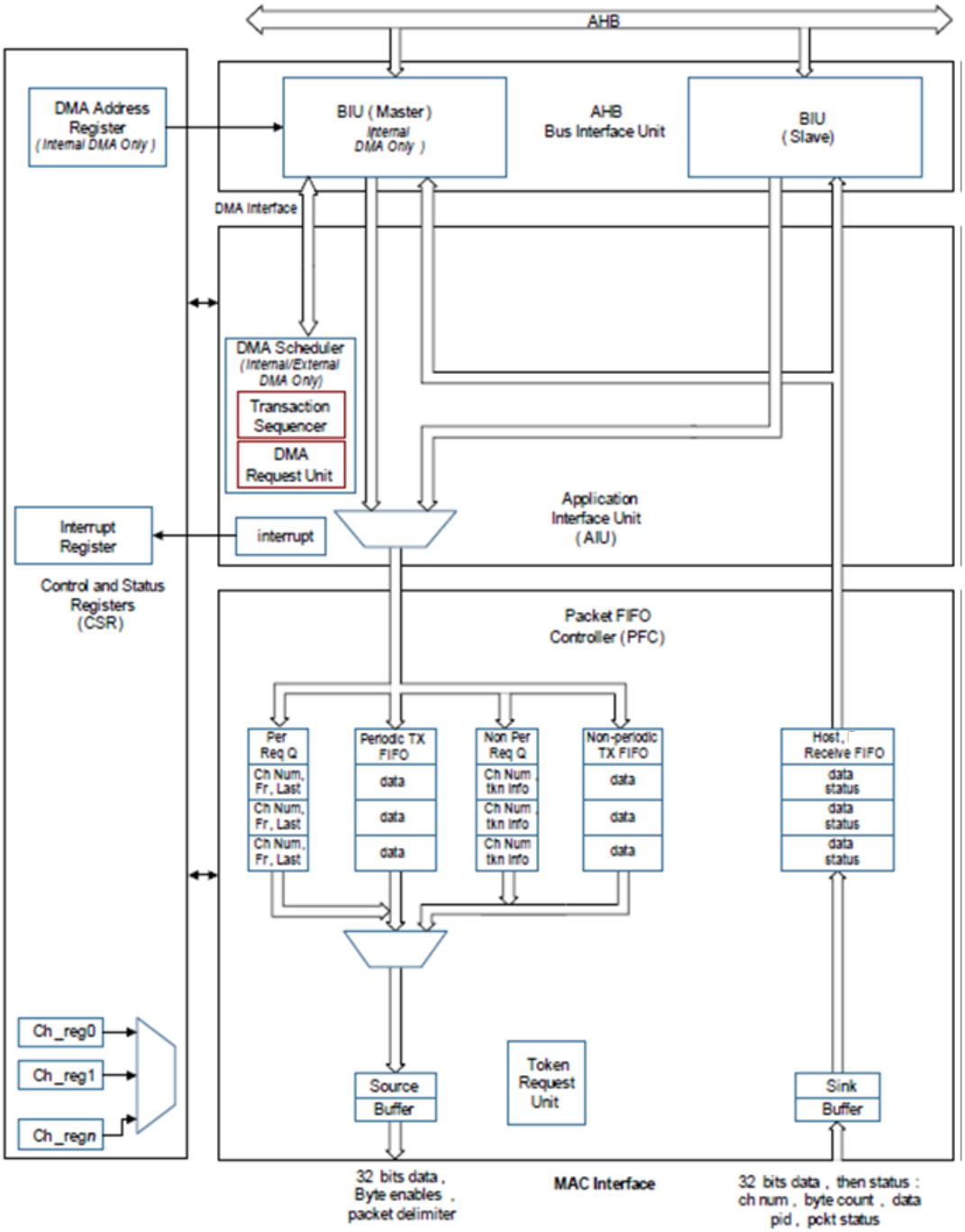


Fig. 26-2 USB HOST 2.0 Controller Architecture

26.4 USB Host2.0 PHY

The USB HOST 2.0 PHY connects a host controller to a USB system. It is a complete mixed-signal IP designed to implement USB connectivity in a System-on-Chip(SOC) design targeted to a specific fabrication process using core and 2.5-V thick-oxide devices. The USB 2.0 PHY supports the USB2.0 480-Mbps protocol and data rate, and is backward compatible with the USB 1.1 1.5-Mbps and 12-Mbps protocol and data rates.

26.4.1 Block Diagram

Fig.27-13 shows the USB HOST 2.0 PHY functional block diagram for a one-port macro.

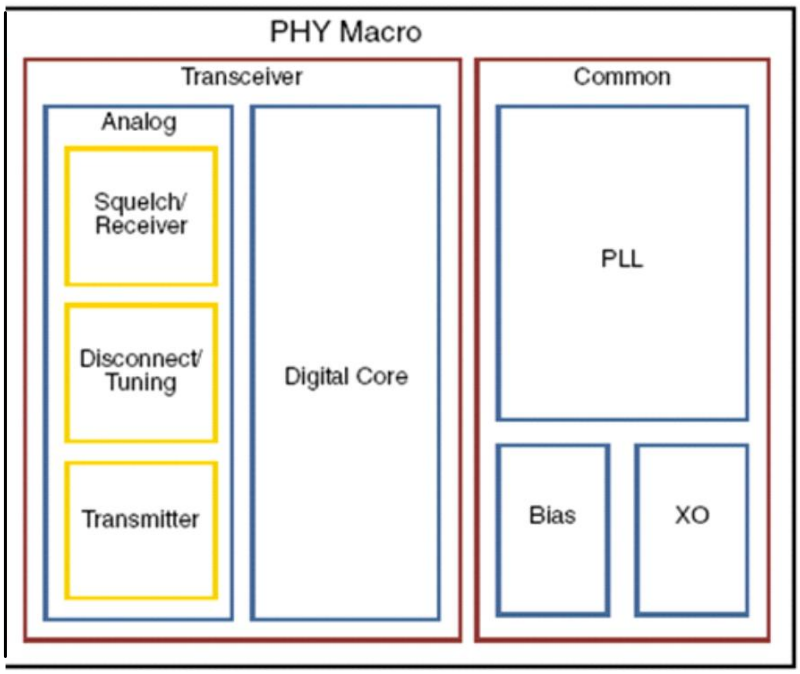


Fig. 26-3 USB HOST 2.0 PHY Architecture

The USB HOST 2.0 PHY consists of three basic components: the Common block and Transceiver block.

- Common block: This block contains design components that can be reused for multiple transceivers.
- Transceiver block: This block contains the bulk of USB HOST 2.0 PHY circuitry for data processingand transfers.

26.5 Register description

26.5.1 Registers Summary

Name	Offset	Size	Reset Value	Description
HOST20_GOTGCTL	0x0000	W	0x00000000	Control and Status Register
HOST20_GOTGINT	0x0004	W	0x00000000	Interrupt Register
HOST20_GAHBCFG	0x0008	W	0x00000000	AHB Configuration Register
HOST20_GUSBCFG	0x000c	W	0x00001400	USB Configuration Register
HOST20_GRSTCTL	0x0010	W	0x80000000	Reset Register
HOST20_GINTSTS	0x0014	W	0x00000000	Interrupt Register
HOST20_GINTMSK	0x0018	W	0x00000000	Interrupt Mask Register

Name	Offset	Size	Reset Value	Description
HOST20_GRXSTSR	0x001c	W	0x00000000	Receive Status Debug Read
HOST20_GRXSTSP	0x0020	W	0x00000000	Receive Status Debug Pop
HOST20_GRXFSIZ	0x0024	W	0x00000000	Receive FIFO Size Register
HOST20_GNPTXFSIZ	0x0028	W	0x00000000	Non-Periodic Transmit FIFO Size Register
HOST20_GNPTXSTS	0x002c	W	0x00000000	Non-Periodic Transmit FIFO/Queue Status Register
HOST20_GI2CCTL	0x0030	W	0x11000000	I2C Address Register
HOST20_GPVNDCTL	0x0034	W	0x00000000	PHY Vendor Control Register
HOST20_GGPIO	0x0038	W	0x00000000	General Purpost Input/Output Register
HOST20_GUID	0x003c	W	0x00000000	User ID Register
HOST20_GSNPSID	0x0040	W	0x00004f54	Vendor ID Register
HOST20_GHWCFG1	0x0044	W	0x00000000	User HW Config1 Register
HOST20_GHWCFG2	0x0048	W	0x00000000	User HW Config2 Register
HOST20_GHWCFG3	0x004c	W	0x00000000	User HW Config3 Register
HOST20_GHWCFG4	0x0050	W	0x00000000	User HW Config4 Register
HOST20_GLPMCFG	0x0054	W	0x00000000	Core LPM Configuration Register
HOST20_GPWRDN	0x0058	W	0x00000000	Global Power Down Register
HOST20_GDFIFOCFG	0x005c	W	0x00000000	Global DFIFO Software Config Register
HOST20_GADPCTL	0x0060	W	0x00000000	ADP Timer,Control and Status Register
HOST20_HPTXFSIZ	0x0100	W	0x00000000	Host Periodic Transmit FIFO Size Register
HOST20_DIEPTXFn	0x0104	W	0x00000000	Device Periodic Transmit FIFO-1 Size Register
HOST20_HCFG	0x0400	W	0x00000000	Host Configuration Register

Name	Offset	Size	Reset Value	Description
HOST20_HFIR	0x0404	W	0x00000000	Host Frame Interval Register
HOST20_HFNUM	0x0408	W	0x0000ffff	Host Frame Number/Frame Time Remaining Register
HOST20_HPTXSTS	0x0410	W	0x00000000	Host Periodic Transmit FIFO/Queue Status Register
HOST20_HAINT	0x0414	W	0x00000000	Host All Channels Interrupt Reigster
HOST20_HAINTMSK	0x0418	W	0x00000000	Host All Channels Interrupt Mask Register
HOST20_HPRT	0x0440	W	0x00000000	Host Port Control and Status Register
HOST20_HCCHARn	0x0500	W	0x00000000	Host Channel-n Characteristics Register
HOST20_HCSPLTn	0x0504	W	0x00000000	Host Channel-n Split Control Register
HOST20_HCINTn	0x0508	W	0x00000000	Host Channel-n Interrupt Register
HOST20_HCINTMSKn	0x050c	W	0x00000000	Host Channel-n Interrupt Mask Register
HOST20_HCTSIZn	0x0510	W	0x00000000	Host Channel-n Transfer Size Register
HOST20_HCDMAn	0x0514	W	0x00000000	Host Channel-n DMA Address Register
HOST20_HCDMABn	0x051c	W	0x00000000	Host Channel-n DMA Buffer Address Register
HOST20_DCFG	0x0800	W	0x08200000	Device Cconfiguration Register
HOST20_DCTL	0x0804	W	0x00002000	Device Control Register
HOST20_DSTS	0x0808	W	0x00000000	Device Status Register
HOST20_DIEPMSK	0x0810	W	0x00000000	Device IN Endpoint common interrupt mask register
HOST20_DOEPMSK	0x0814	W	0x00000000	Device OUT Endpoint common interrupt mask register
HOST20_DAINT	0x0818	W	0x00000000	Device All Endpoints interrupt register

Name	Offset	Size	Reset Value	Description
HOST20_DAINTMSK	0x081c	W	0x00000000	Device All Endpoint interrupt mask register
HOST20_DTKNQR1	0x0820	W	0x00000000	Device IN token sequence learning queue read register1
HOST20_DTKNQR2	0x0824	W	0x00000000	Device IN token sequence learning queue read register2
HOST20_DVBUSDIS	0x0828	W	0x00000b8f	Device VBUS discharge time register
HOST20_DVBUSPULSE	0x082c	W	0x00000000	Device VBUS Pulsing Timer Register
HOST20_DTHRCTL	0x0830	W	0x08100020	Device Threshold Control Register
HOST20_DIEPEMPMSK	0x0834	W	0x00000000	Device IN endpoint FIFO empty interrupt mask register
HOST20_DEACHINT	0x0838	W	0x00000000	Device each endpoint interrupt register
HOST20_DEACHINTMSK	0x083c	W	0x00000000	Device each endpoint interrupt register mask
HOST20_DIEPEACHMSKn	0x0840	W	0x00000000	Device each IN endpoint -n interrupt Register
HOST20_DOEPEACHMSKn	0x0880	W	0x00000000	Device each out endpoint-n interrupt register
HOST20_DIEPCTL0	0x0900	W	0x00008000	Device control IN endpoint 0 control register
HOST20_DIEPINTn	0x0908	W	0x00000000	Device Endpoint-n Interrupt Register
HOST20_DIEPTSIZn	0x0910	W	0x00000000	Device endpoint n transfer size register
HOST20_DIEPDMAn	0x0914	W	0x00000000	Device endpoint-n DMA address register
HOST20_DTXFSTSn	0x0918	W	0x00000000	Device IN endpoint transmit FIFO status register
HOST20_DIEPDMABn	0x091c	W	0x00000000	Device endpoint-n DMA buffer address register

Name	Offset	Size	Reset Value	Description
HOST20_DIEPCTLn	0x0920	W	0x00000000	Device endpoint-n control register
HOST20_DOEPCTL0	0x0b00	W	0x00000000	Device control OUT endpoint 0 control register
HOST20_DOEPINTn	0x0b08	W	0x00000000	Device endpoint-n control register
HOST20_DOEPTSIZn	0x0b10	W	0x00000000	Device endpoint n transfer size register
HOST20_DOEPDMAn	0x0b14	W	0x00000000	Device Endpoint-n DMA Address Register
HOST20_DOEPDMABn	0x0b1c	W	0x00000000	Device endpoint-n DMA buffer address register
HOST20_DOEPCTLn	0x0b20	W	0x00000000	Device endpoint-n control register
HOST20_PCGCR	0x0b24	W	0x200b8000	Power and clock gating control register
HOST20_EPBUF0	0x1000	W	0x00000000	Device endpoint 0 / host out channel 0 address
HOST20_EPBUF1	0x2000	W	0x00000000	Device endpoint 1 / host out channel 1 address
HOST20_EPBUF2	0x3000	W	0x00000000	Device endpoint 2 / host out channel 2 address
HOST20_EPBUF3	0x4000	W	0x00000000	Device endpoint 3 / host out channel 3 address
HOST20_EPBUF4	0x5000	W	0x00000000	Device endpoint 4 / host out channel 4 address
HOST20_EPBUF5	0x6000	W	0x00000000	Device endpoint 5 / host out channel 5 address
HOST20_EPBUF6	0x7000	W	0x00000000	Device endpoint 6 / host out channel 6 address
HOST20_EPBUF7	0x8000	W	0x00000000	Device endpoint 7 / host out channel 7 address
HOST20_EPBUF8	0x9000	W	0x00000000	Device endpoint 8 / host out channel 8 address
HOST20_EPBUF9	0xa000	W	0x00000000	Device endpoint 9 / host out channel 9 address
HOST20_EPBUF10	0x9000	W	0x00000000	Device endpoint 10 / host out channel 10 address

Name	Offset	Size	Reset Value	Description
HOST20_EPBUF11	0xa000	W	0x00000000	Device endpoint 11 / host out channel 11 address
HOST20_EPBUF12	0x9000	W	0x00000000	Device endpoint 12 / host out channel 12 address
HOST20_EPBUF13	0xa000	W	0x00000000	Device endpoint 13 / host out channel 13 address
HOST20_EPBUF14	0x9000	W	0x00000000	Device endpoint 14 / host out channel 14 address
HOST20_EPBUF15	0xa000	W	0x00000000	Device endpoint 15 / host out channel 15 address

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

26.5.2 Registers Description

Refer to 27.5.2

26.6 Interface description

Table 26-1 USB HOST 2.0 Interface Description

Module Pin	Direction	Pad Name	pinmux
HOST_VSSAC	AG	HOST_VSSAC	-
HOST_DVSS	DG	HOST_DVSS	-
HOST_DVDD	DP	HOST_DVDD	-
HOST_VDD25	AP	HOST_VDD25	-
HOST_DM	A	HOST_DM	-
HOST_RKELVIN	A	HOST_RKELVIN	-
HOST_DP	A	HOST_DP	-
HOST_VSSA	AG	HOST_VSSA	-
HOST_VBUS	A	HOST_VBUS	-
HOST_VDD33	AP	HOST_VDD33	-
host_drv_vbus	O	GPIO0_A[6]	GRF_GPIO0A_IOMUX [12]=1

***Note**: **A**—Analog pad ; **AP**—Analog power; **AG**—Analog ground ;**DP**—Digital power ;**DG**–Digital ground;*

Chapter 27 USB OTG2.0

27.1 Overview

USB OTG 2.0 is a Dual-Role Device controller, which supports both device andhost functions and is fully compliant with OTG Supplement to USB2.0 specification, andsupport high-speed(480Mbps),full-speed(12Mbps),low-speed(1.5Mbps) transfer.

USB OTG 2.0 isoptimized for portable electronic devices, point-to-point applications(no hub, directcoernnection to device) and multi-point applications to devices.USB OTG 2.0 interface supports both device and host functions and is fully compliant with OTG Supplement to USB2.0 specification, and support high-speed(480Mbps), full-speed(12Mbps) , low-speed (1.5Mbps) transfer. It is optimized for portable electronic device , point-to-point applications (no hub, direct connection to device) and multi-point applications to devices.

27.1.1 Features

- Compliant with the OTG Supplement to the USB2.0 Specification
- Operates in High-Speed and Full-Speed mode
- Support Session Request Protocol(SRP) and Host Negotiation Protocol(HNP)
- Support 9 channels in host mode
- 9 Device mode endpoints in addition to control endpoint 0, 4 in,3 out and 2 IN/OUT
- Built-in one 1024x35 bits FIFO
- Internal DMA with scatter/gather function
- Supports packet-based , synamic FIFO memory allocation for endpoints for flexible, efficient use of RAM
- Provides support to change an endpoint's FIFO memory size during transfers

27.2 Block Diagram

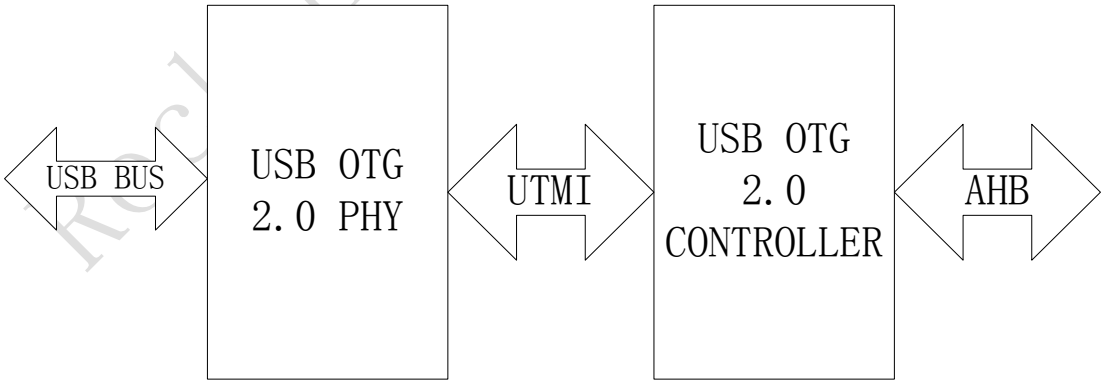


Fig. 27-1USB OTG 2.0 Architecture

Fig.27-1 shows the architecture of USB OTG 2.0. It is broken up into two separate units: USB OTG 2.0 controller and USB OTG 2.0 PHY. The two units are interconnected with UTMI interface.

27.2.1 USB OTG 2.0 Controller Function

The USB OTG 2.0 Controller controlsSIE(Serial Interface Engine) Logic , the

Endpointlogic , the Channel logic and the internal DMA logic.

The SIElogic contains the USB PID and address recognition logic, and other sequencingand state machine logic to handle USB packets and transactions. Generally the SIE Logicis required for any USB implementation while the number and types of endpoints will vary as function ofapplication and performance requirements.

The Endpoint Logic contains the endpointspecific logic: endpoint number recognition, FIFOs and FIFO control, etc.

The Channel Logic contains the channel tasks schedule, FIFOs and FIFO control, etc.

The internal DMA logic controls data trasaction between system memory and USB FIFOs.

27.2.2 USB OTG 2.0 PHY Function

The USB OTG 2.0 PHY handles the low level USB protocol and signaling. This includes features such as; dataserialization and deserialization, bit stuffing and clock recovery and synchronization. The primary focus ofthis block is to shift the clock domain of the data from the USB 2.0 rate to the frequency of UTMI clock which is 30MHz.

27.2.3 UTMI Interface

- Trasmit

Transmit must be asserted to enable any transmissions.

1) The USB OTG2.0 CONTROLLER asserts TXValid to begin a transmission and negates TXValid to end a transmission.After the USB OTG2.0 CONTROLLER asserts TXValid it can assume that the transmission has started when it detectsTXReady asserted.
2) The USB OTG2.0 CONTROLLER assumes that the USB OTG2.0 PHYhas consumed a data byte if TXReady and TXValid are asserted.
3) The USB OTG2.0 CONTROLLER must have valid packet information (PID) asserted on the DataIn bus coincident with theassertion of TXValid. Depending on the USB OTG2.0 PHYimplementation, TXReady may be asserted by theTransmit State Machine as soon as one CLK after the assertion of TXValid.
4) TXValid and TXReady are sampled on the rising edge of CLK.
5) The Transmit State Machine does NOT automatically generate Packet ID's (PIDs) or CRC. Whentransmitting, the USB OTG2.0 CONTROLLER is always expected to present a PID as the first byte of the data stream and ifappropriate , CRC as the last bytes of the data stream.
6) The USB OTG2.0 CONTROLLER must use LineState to verify a Bus Idle condition before asserting TXValid in the TX Waitstate.
7) The state of TXReady in the TX Wait and Send SYNC states is undefined. An MTU implementationmay prepare for the next transmission immediately after the Send EOP state and assert TXReady inthe TX Wait state. An MTU implementation may also assert TXReady in the Send SYNC state. Thefirst assertion of TXReady is Macrocell implementation dependent. The USB OTG2.0 CONTROLLER must prepare DataIn forthe first byte to be transmitted before asserting TXValid.

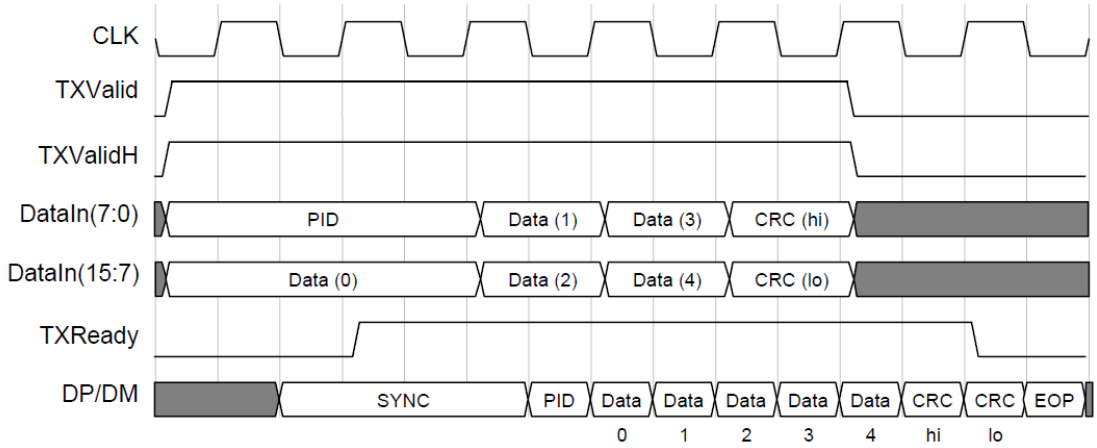


Fig. 27-2 UTMI interface –Transmit timing for a data packet

- Receive

1) RXActive and RXValid are sampled on the rising edge of CLK.
2) In the RX Wait state the receiver is always looking for SYNC.
3) The USB OTG 2.0 PHY asserts RXActive when SYNC is detected (Strip SYNC state).
4) The USB OTG 2.0 PHY negates RXActive when an EOP is detected (Strip EOP state).
5) When RxActive is asserted, RXValid will be asserted if the RX Holding Register is full.
6) RXValid will be negated if the RX Holding Register was not loaded during the previous byte time.
7) This will occur if 8 stuffed bits have been accumulated.
8) The USB OTG2.0 Controller must be ready to consume a data byte if RXActive and RXValid are asserted (RX Data state).
9) In FS mode, if a bit stuff error is detected then the Receive State Machine will negate RXActive and
10) RXValid, and return to the RXWait state.

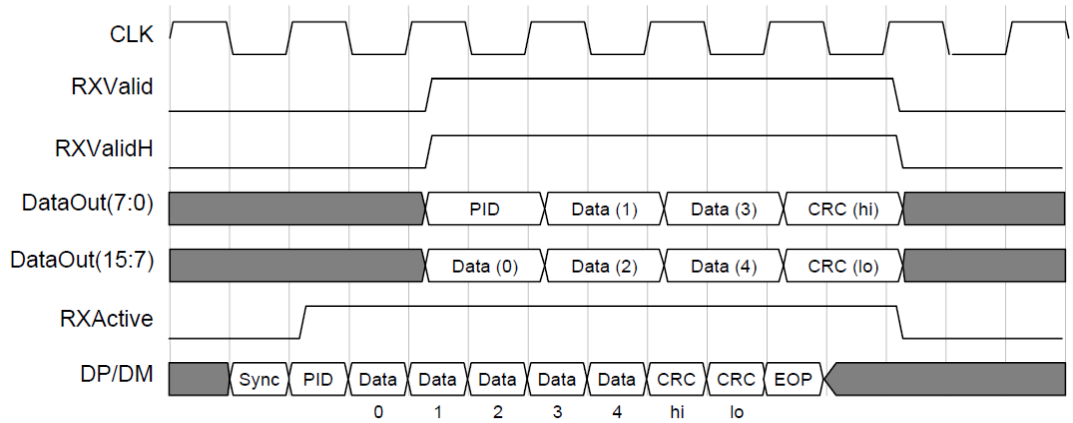


Fig. 27-3 UTMI interface – Receive timing for a data packet

27.3 USB OTG2.0 Controller

USB OTG Controller is a Dual-Role Device controller, which supports both device and host functions and is fully compliant with OTG Supplement to USB2.0 specification, and support high-speed(480Mbps), full-speed(12Mbps) , low-speed (1.5Mbps) transfer. This controller will support UTMI+ Level 3 PHY interface. It connects to the industry-standard AMBA AHB for communication with the application and system memory. And it is optimized for portable

electronic device , point-to-point applications (no hub, direct connection to device) and multi-point applications to devices.

Fig.27-2 shows the main components and flow of the USB OTG 2.0 controller system.

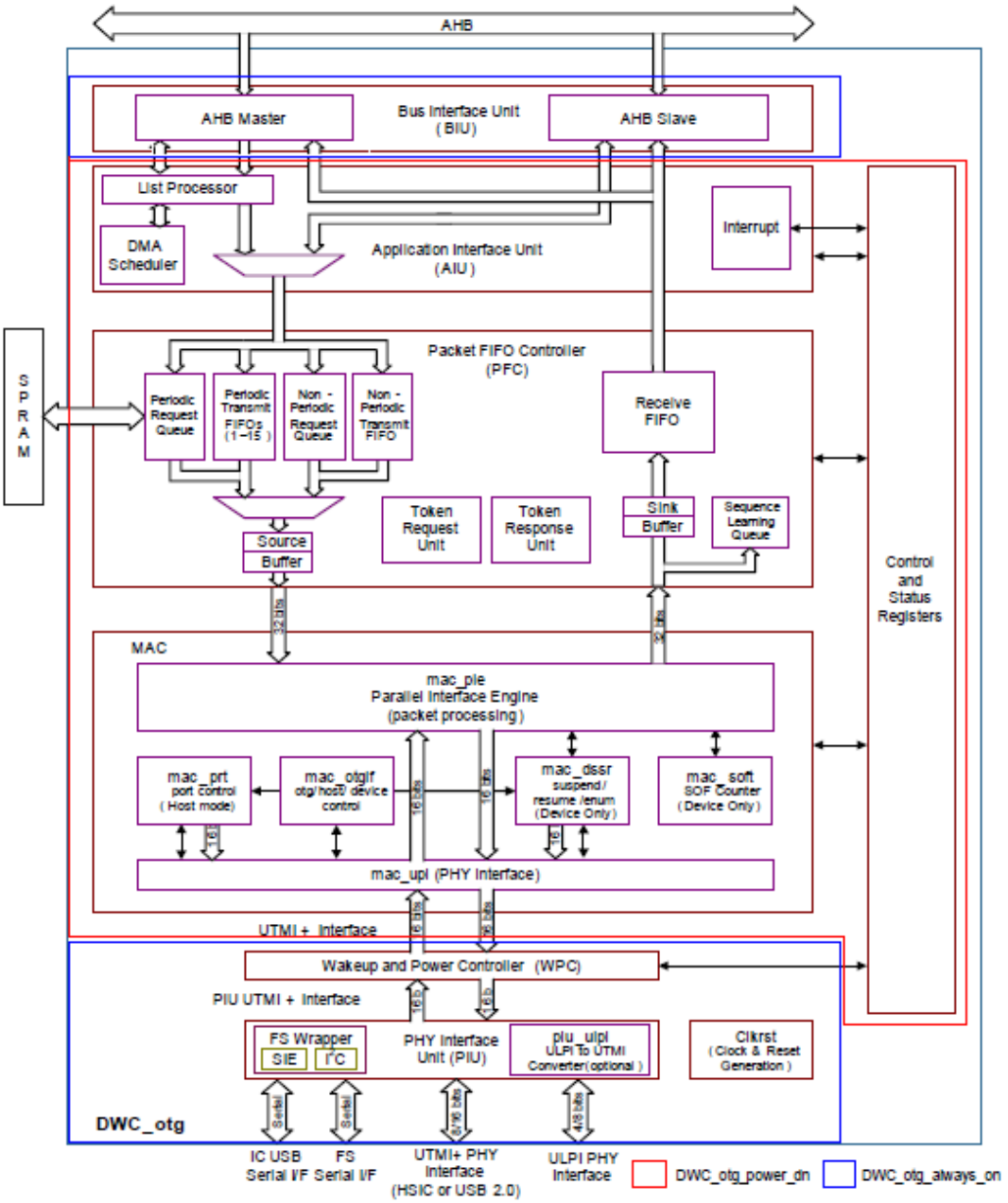


Fig. 27-4 USB OTG2.0 Controller Architecture

27.3.1 Host Architercture

The host uses one transmit FIFO for all non-periodic OUT transactions and one transmit FIFO for all periodic OUT transactions (periodic FIFOs 2 to n are only used in Device mode, where n is number of periodic IN endpoints in Device mode). These transmit FIFOs are used as transmit buffers to hold the data (payload of the transmit packet) to be transmitted over USB. The host pipes the USB transactions through Request queues (one for periodic and one for non-periodic). Each entry in the Request queue holds the IN or OUT channel number along with other information to perform a transaction on the USB. The order in which the requests are written into the queue determines the sequence

of transactions on the USB. The host processes the periodic Request queue first, followed by the non-periodic Request queue, at the beginning of each (micro)frame. The host uses one receive FIFO for all periodic and non-periodic transactions. The FIFO is used as a receive buffer to hold the received data (payload of the received packet) from the USB until it is transferred to the system memory. The status of each packet received also goes into the FIFO. The status entry holds the IN channel number along with other information, such as received byte count and validity status, to perform a transaction on the AHB. Fig27-2 shows the bus interface architecture of the USB OTG 2.0Controller in Host mode.

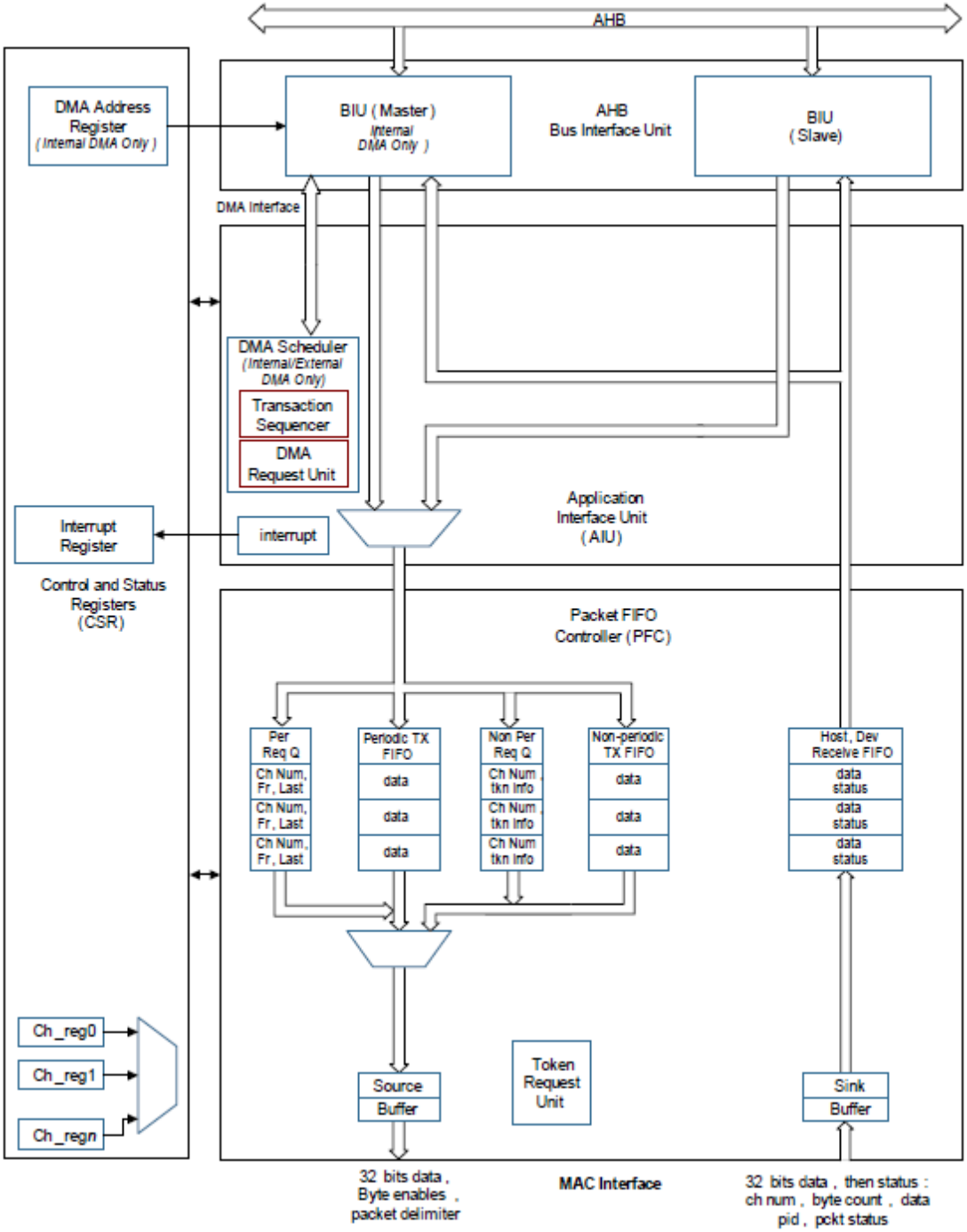


Fig. 27-5 USB OTG2.0 Controller – Host Architecture

27.3.2 Device Architercture

- Dedicated Transmit FIFO Operation

  When dedicated transmit FIFO architecture is used (OTG_EN_DED_TX_FIFO = 1), the core uses individual transmit FIFOs for each IN endpoint. There are no Request queues associated with any of the FIFOs. There is no need for the application to predict the order in which the USB host is going to access the non-periodic endpoints. In Dedicated Transmit FIFO Operation, the core also supports thresholding in the transmit and receive directions when DMA mode is selected. For Transmit mode there are separate controls to enable thresholding for isochRnous and non-isochRnous transfers.When thresholding is enabled, the core can be configured to have less-than-one-packet-sized FIFO. The core internally handles underrun condition during transmit and corrupts the packet (inverts the CRC) on the USB. During receive with thresholding, when a packet ends up in a FIFO overflow condition, the core NAKs the OUT packet and internally rewinds the pointers. When thresholding is enabled, vendor recommends that you have a FIFO size two times the threshold value. If packet transmission results in underrun condition—(eventually resulting in packet corruption on the USB—often the host can time out the endpoint after three consecutive errors.

- Single Receive FIFO

  The OTG device uses a single receive FIFO to receive the data for all the OUT endpoints. The receive FIFO holds the status of the received data packet, such as byte count, data PID and the validity of the received data. The DMA or the application reads the data out of the receive FIFO as it is received.

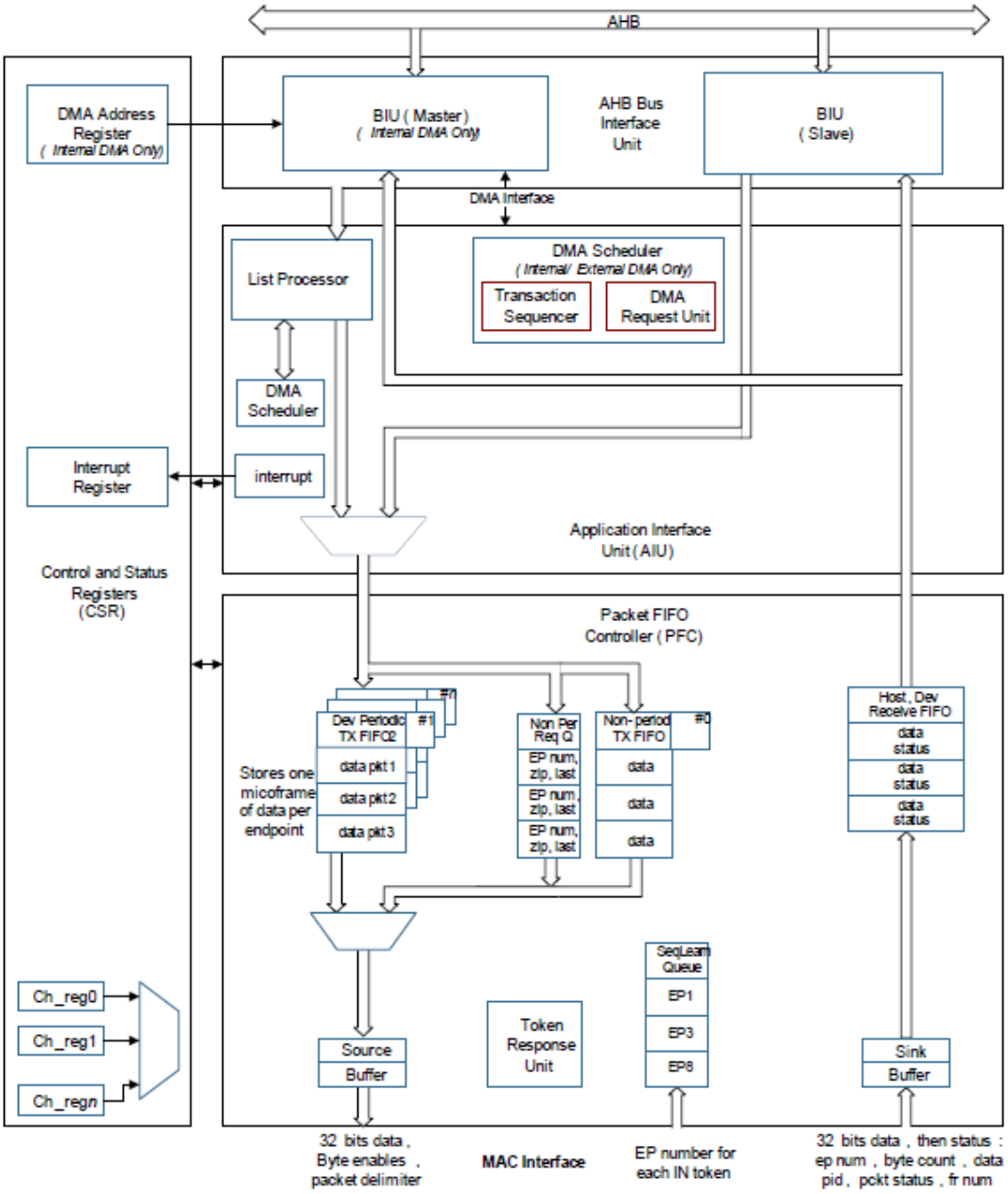


Fig. 27-6 USB OTG2.0 Controller – Host Architecture

27.3.3 Internal DMA Mode

Internal DMA mode is typically selected when the CPU bandwidth to process the USB transfer is limited and you Wuld like an internal DMA controller to take care of the data transfers between the system memory and the USB OTG 2.0 Controller. The driver sets up the transfer and the USB OTG 2.0 Controller interrupts the processor only on transfer completion or an error condition

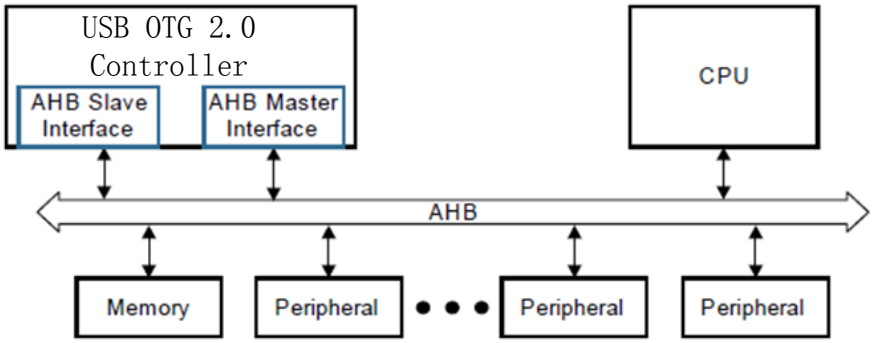


Fig. 27-7 USB OTG2.0 Controller – Internal DMA mode

27.3.4 FIFO Mapping

- Fig.27-8 shows FIFO mapping in Host mode.

When the device is operating in Internal DMA mode, the last locations of the SPRAM are used to store the DMAADDR values for each channel.

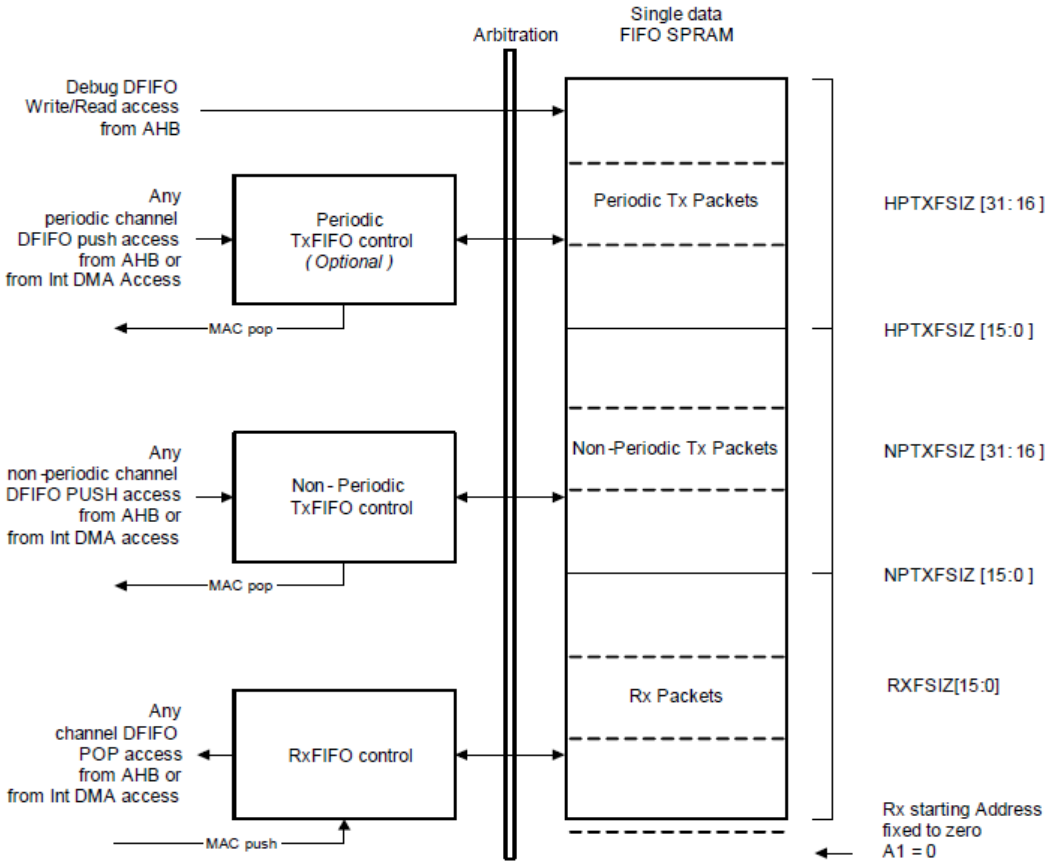


Fig. 27-8 USB OTG 2.0 Controller host mode FIFO address mapping

- Fig.27-9shows FIFO mapping in Device mode.

When the device is operating in non Descriptor Internal DMA mode, the last locations of the SPRAM are used to store the DMAADDR values for each channel. When the device is operating in Descriptor mode, then the last locations of the SPRAM store the Base Descriptor address, Current Descriptor address, Current Buffer address, and status quadlet information for each endpoint direction.

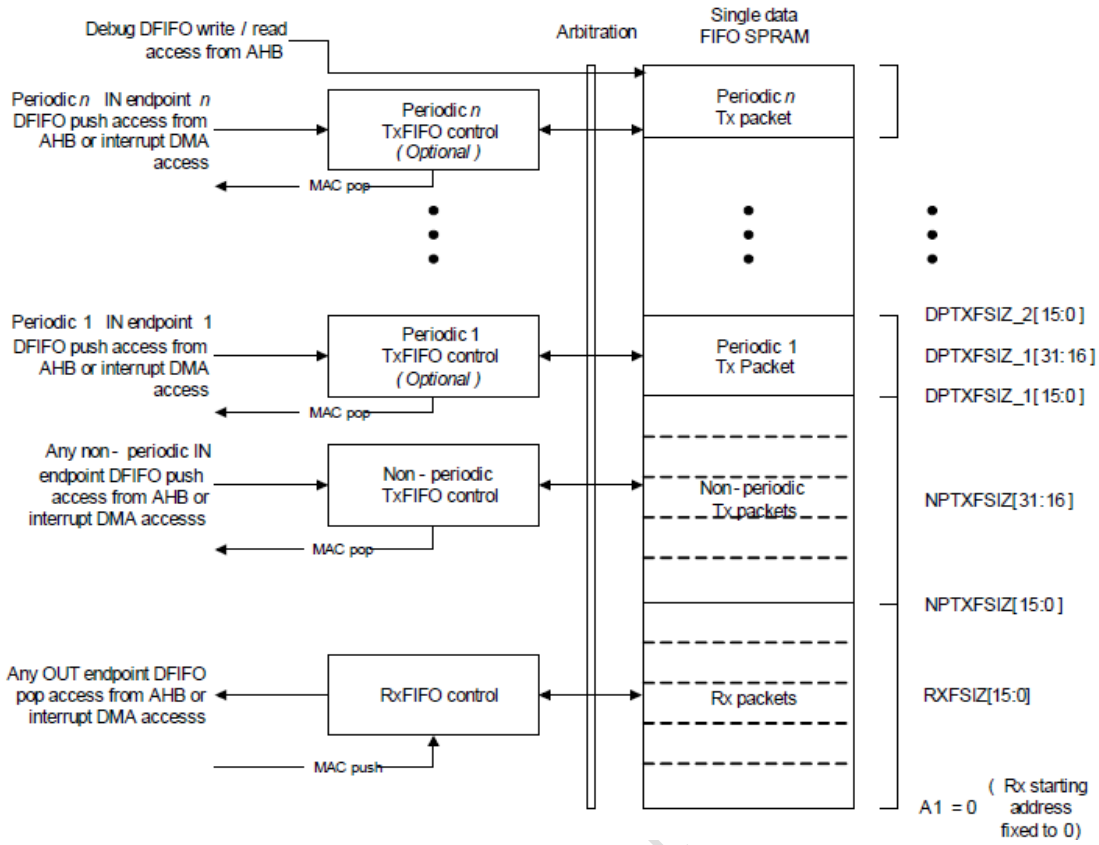


Fig. 27-9 USB OTG 2.0 Controller device mode FIFO address mapping

27.3.5 Sub-Module Architecture

1.DMA Scheduler (DSCH)

This block is used only in DMA mode. It controls the transfer of data packets between the system memory and the USB OTG 2.0 Controller for both Internal and External DMA. The following functions constitute the DMA scheduler:

- Arbiter

  This logic provides the sequence in which the channels/endpoints are to be processed in DMA mode. In Host mode, the arbiter provides Rund-Rbin arbitration among periodic and non-periodic channels. Periodic channels are processed with higher priority. In Device mode with Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0), the arbiter provides a Rund-Rbin arbitration only among periodic endpoints. Arbitration for non-periodic endpoints is based on Next EP Number1 link register values. Periodic endpoints are processed with higher priority. In Device mode, during a Dedicated Transmit FIFO operation (OTG_EN_DED_TX_FIFO = 1), and when threshold is enabled, the priority is as follows:

  a) Any transmit endpoint which is active on the USB.
  b) Any receive data in a receive FIFO.
  c) Rund-Rbin arbitration on periodic transmit endpoints.
  d) Rund-Rbin arbitration on non-periodic transmit endpoints.

  In device mode when dedicated FIFO mode is used and when thresholding is not enabled, Rund Rbin arbitration is used for periodic and non-periodic IN endpoints with priority given to periodic IN endpoints.
- DMA Request State Machine The state machine is responsible for the following:

- Requesting the External/Internal DMA for data fetch (from system memory to transmit FIFO, one maximum packet size or last packet size at a time)
- Writing the OUT request token into the request queue at the end of data fetch, in host mode and in Device mode when OTG_EN_DED_TX_FIFO = 0.
- Writing the IN request token into the Request Queue in Host mode
- Requesting the External/Internal DMA for data update (from receive FIFO to system memory, one maximum packet size or last packet size at a time)
- Writing the Request Queue for ping, complete split, zeR-length packet, or disable channel requests for the host.

2.Packet FIFO Controller (PFC)

Fig.27-10 represents the Packet FIFO Controller in Shared FIFO operation.Several FIFOs are used in Device and Host modes to store data inside the core before transmitting it on either the AHB or the USB.

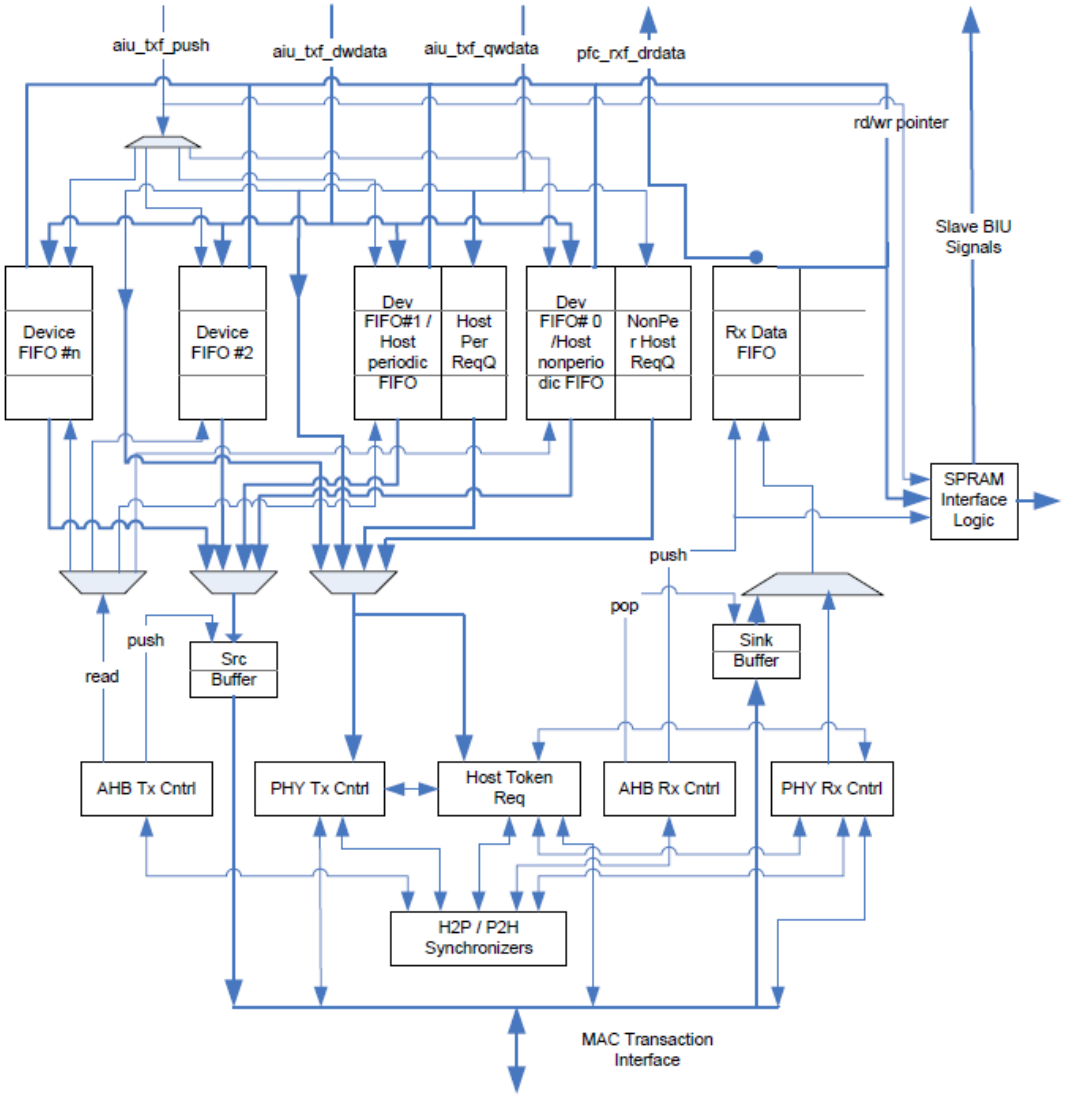


Fig. 27-10 USB OTG 2.0 Controller Packet FIFO controller

- PFC-to-SPRAM interface

Fig.27-11 shows how to connect the USB OTG 2.0 Controller Data FIFO

interface to an industry-standard, single-port synchRnous SRAM. Address, write data, and control outputs are driven late by the USB OTG 2.0 Controller, but in time to meet the SRAM setup requirements. Input read data is expected late from the SSRAM and registered inside the core before being used.

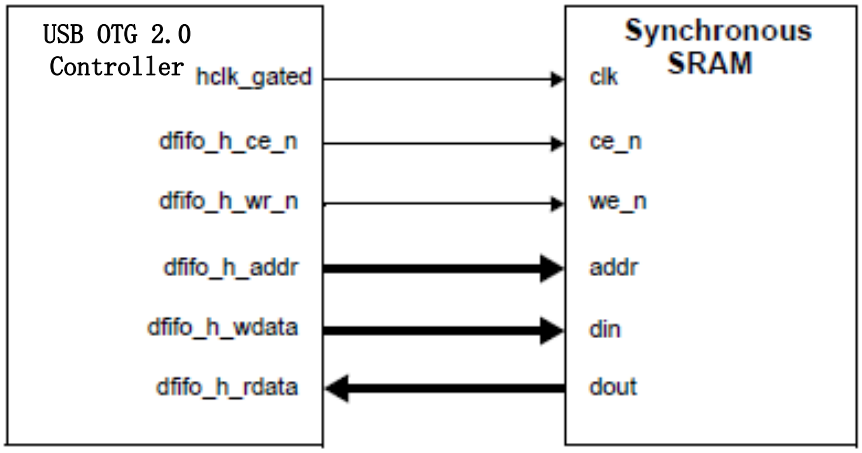


Fig. 27-11 DFIFO single-port synchRnous SRAM interface

3.Media Access Controller

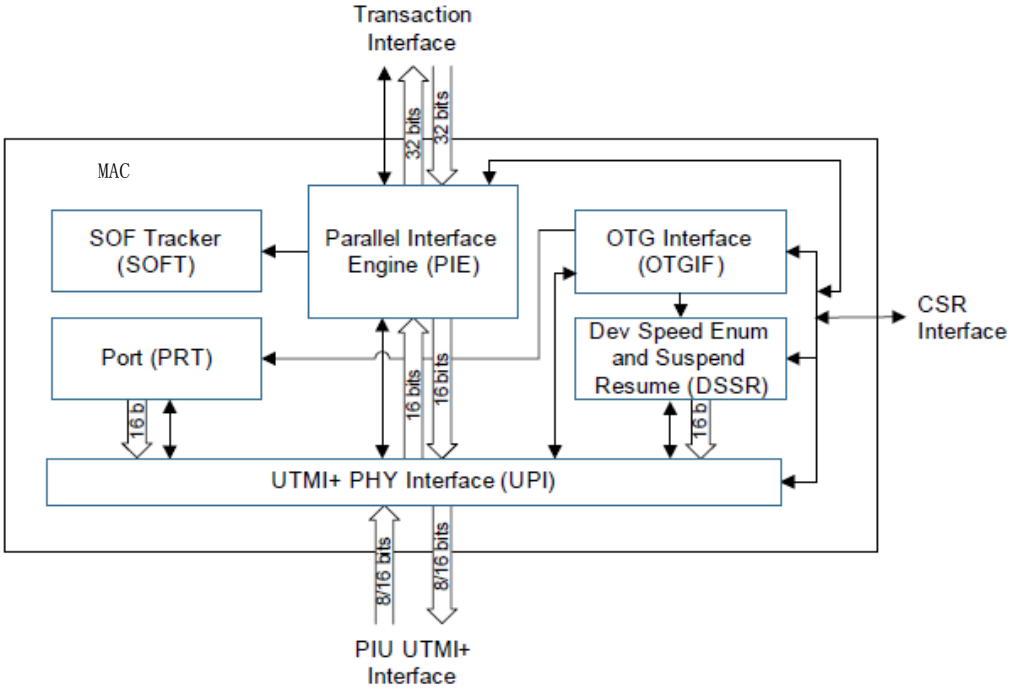


Fig. 27-12 USB OTG 2.0 Controller – MAC block diagram

Fig 27-12 shows the major MAC module components.
The major blocks are:

- Device Speed enumeration, Suspend, and Resume block (DSSR) The DSSR block is only active in Device mode. This block performs the speed enumeration, suspend, resume and remote wakeup functions in Device mode.
- Parallel Interface Engine (PIE) This block is responsible for token, data, and handshake packet generation and reception, and PID and CRC checking and generation. It generates handshake and data packets based on data

integrity and on CSR control and FIFO status information. The PIE also handles the data transfer to and from the FIFO, and the status update to the PFC and AIU.

- SOF tracker (SOFT) This block tracks SOF packets and generates SOF interrupts in Device mode. It handles missing SOFs and delayed SOFs to keep the frame number synchronization between the host and the device.
- Port (PRT) The Port block is only active in the Host mode. It is responsible for connect and disconnect detection, USB reset and speed enumeration, suspend and resume generation, remote wakeup detection, SOF generation, and High Speed Test mode handling. OTG Interface (OTGIF) The OTG Interface block handles SRP and HNP. These OTG protocols are implemented either through the regular UTMI+ interface
- UTMI+ PHY Interface (UPI) The block converts data widths for the 8-bit PHY interface and multiplexes output signals to the PHY from multiple blocks. It also implements some logic shared by multiple MAC blocks. The MAC is designed so that unused components can be removed in some configurations to reduce gate count. SOFT, DSSR, and OTGIF can be removed in a host-only configuration; PRT and OTGIF can be removed for device-only configuration.

27.4 USB OTG2.0 PHY

The USB OTG 2.0 PHY connects a USB OTG controller to a USB system. It is a complete mixed-signal IP designed to implement OTG connectivity in a System-on-Chip(SOC) design targeted to a specific fabrication process using core and 2.5-V thick-oxide devices. The USB 2.0 PHY supports the USB2.0 480-Mbps protocol and data rate, and is backward compatible with the USB 1.1 1.5-Mbps and 12-Mbps protocol and data rates.

27.4.1 Block Diagram

Fig.27-13 shows the USB OTG 2.0 PHY functional block diagram for a one-port macro.

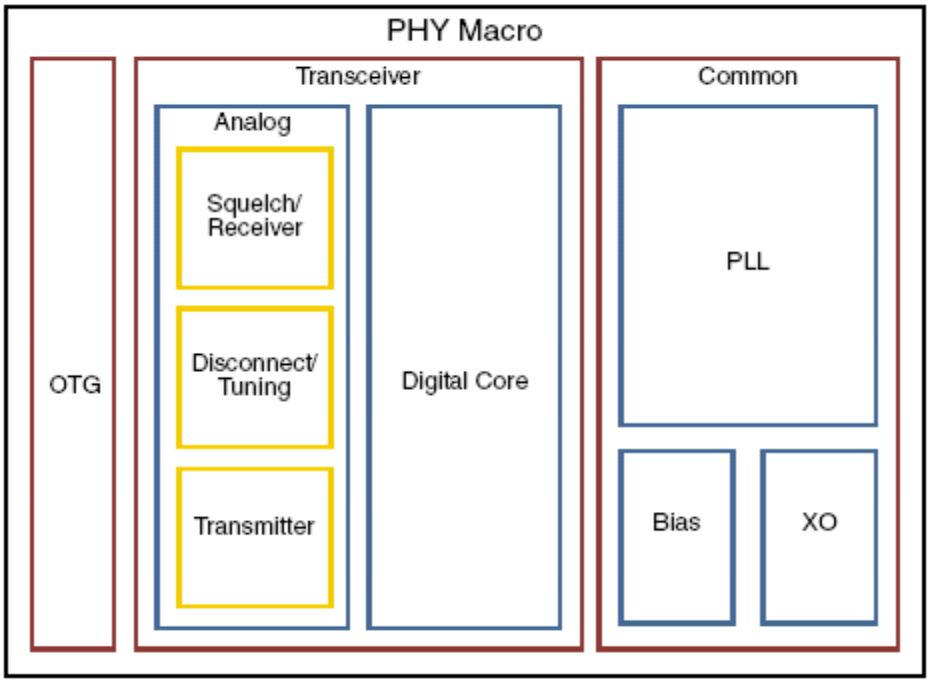


Fig. 27-13 USB OTG 2.0 PHY Architecture

The USB OTG 2.0 PHY consists of three basic components: the Common block, Transceiver block, and OTGblock.

- Common block: This block contains design components that can be reused for multiple transceivers.
- Transceiver block: This block contains the bulk of USB OTG 2.0 PHY circuitry for data processingand transfers.
- OTG block: This block enables A-devices and B-devices to initiate the Session Request Protocol(SRP), and dual-Role devices to initiate the Host Negotiation Protocol (HNP).

27.4.2 Powering Up and Powering Down

- Powering UP

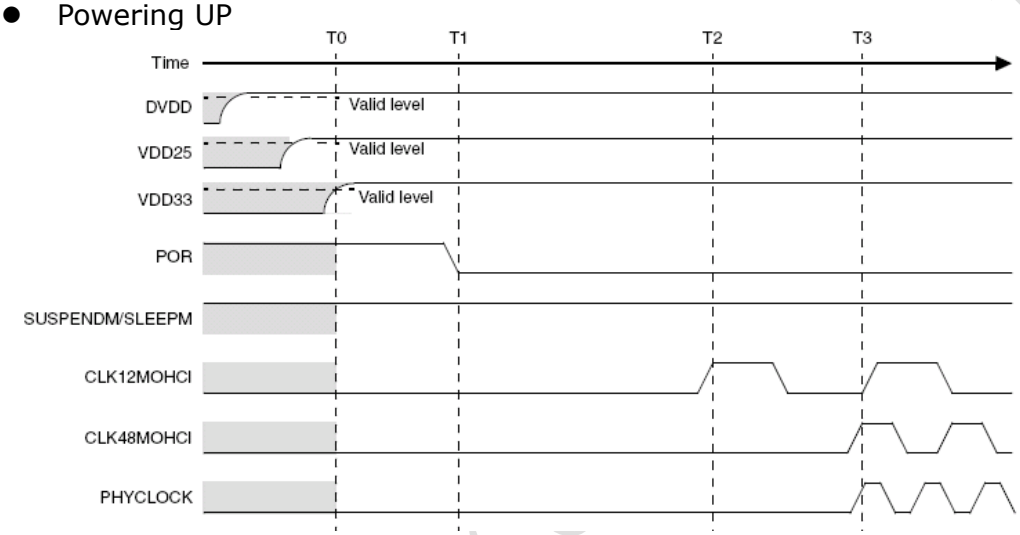


Fig. 27-14 USB OTG 2.0 PHY power supply and power up sequency

Table 27-1 USB OTG 2.0 PHY power supply timing parameter

Timing Parameter	Description	value
T0	Power-on reset (POR) is initiated	0(reference)
T1	T1 indicates when POR can be set to 1\u2019b0. (To provide examples, values for T2 and T3 are also shown where T1 = T0 + 30 \u03bcs.) In general, T1 must be \u2265 T0 + 10 \u03bcs.	T0 + 10us <=T1
T2	T2 indicates which CLK12MOHCI is available at the macro output , based on the USB OTG 2.0 PHY reference clock source	Orystal:T2<T0+620us, External board clock or CLKCORE: T2<T0+2us
GUSBCFG	T3 indicates when PHYCLOCK and CLK48MOHCI are available at the macro output, based on the USB OTG 2.0 PHY reference clock source	1)Crystal: when T1=T0+10us， T3<T1+805us = T0+815us, when T1=T0+30us,T3<T1+805us =T0+835US 2) external board clock or CLKCORE:when T1=T0+10us,T3<T1+45us=

		T0+55us, when T1=T0+30us,T3<T1+45us= T0+75us

27.4.3 Removing Power Supplies for Power Saving

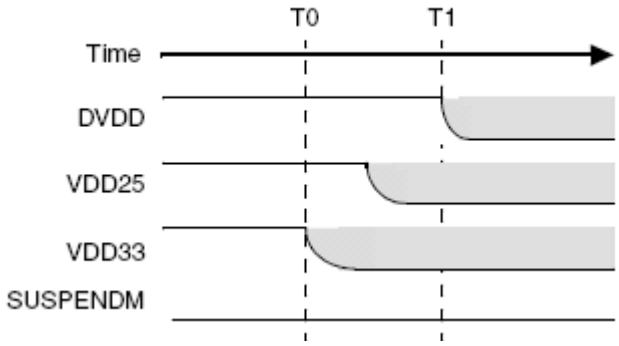


Fig. 27-15 USB OTG 2.0 PHY removing power supplies

27.5 Register description

27.5.1 Register Summary

Name	Offset	Size	Reset Value	Description
USBOTG_GOTGCTL	0x0000	W	0x00000000	Control and Status Register
USBOTG_GOTGINT	0x0004	W	0x00000000	Interrupt Register
USBOTG_GAHBCFG	0x0008	W	0x00000000	AHB Configuration Register
USBOTG_GUSBCFG	0x000c	W	0x00001400	USB Configuration Register
USBOTG_GRSTCTL	0x0010	W	0x80000000	Reset Register
USBOTG_GINTSTS	0x0014	W	0x00000000	Interrupt Register
USBOTG_GINTMSK	0x0018	W	0x00000000	Interrupt Mask Register
USBOTG_GRXSTSR	0x001c	W	0x00000000	Receive Status Debug Read
USBOTG_GRXSTSP	0x0020	W	0x00000000	Receive Status Read and Pop
USBOTG_GRXFSIZ	0x0024	W	0x00000000	Receive FIFO Size Register
USBOTG_GNPTXFSIZ	0x0028	W	0x00000000	Non-Periodic Transmit FIFO Size Register
USBOTG_GNPTXSTS	0x002c	W	0x00000000	Non-Periodic Transmit FIFO/Queue Status Register
USBOTG_GI2CCTL	0x0030	W	0x11000000	I2C Address Register
USBOTG_GPVNDCTL	0x0034	W	0x00000000	PHY Vendor Control Register
USBOTG_GGPIO	0x0038	W	0x00000000	General Purpost Input/Output Register
USBOTG_GUID	0x003c	W	0x00000000	User ID Register
USBOTG_GSNPSID	0x0040	W	0x00004f54	Core ID Register

Name	Offset	Size	Reset Value	Description
USBOTG_GHWCFG1	0x0044	W	0x00000000	User HW Config1 Register
USBOTG_GHWCFG2	0x0048	W	0x00000000	User HW Config2 Register
USBOTG_GHWCFG3	0x004c	W	0x00000000	User HW Config3 Register
USBOTG_GHWCFG4	0x0050	W	0x00000000	User HW Config4 Register
USBOTG_GLPMCFG	0x0054	W	0x00000000	Core LPM Configuration Register
USBOTG_GPWRDN	0x0058	W	0x00000000	Global Power Down Register
USBOTG_GDFIFOCFG	0x005c	W	0x00000000	Global DFIFO Software Config Register
USBOTG_GADPCTL	0x0060	W	0x00000000	ADP Timer,Control and Status Register
USBOTG_HPTXFSIZ	0x0100	W	0x00000000	Host Periodic Transmit FIFO Size Register
USBOTG_DIEPTXFn	0x0104	W	0x00000000	Device Periodic Transmit FIFO-n Size Register
USBOTG_HCFG	0x0400	W	0x00000000	Host Configuration Register
USBOTG_HFIR	0x0404	W	0x00000000	Host Frame Interval Register
USBOTG_HFNUM	0x0408	W	0x0000ffff	Host Frame Number/Frame Time Remaining Register
USBOTG_HPTXSTS	0x0410	W	0x00000000	Host Periodic Transmit FIFO/Queue Status Register
USBOTG_HAINT	0x0414	W	0x00000000	Host All Channels Interrupt Reigster
USBOTG_HAINTMSK	0x0418	W	0x00000000	Host All Channels Interrupt Mask Register
USBOTG_HPRT	0x0440	W	0x00000000	HostPort Control and Status Register
USBOTG_HCCHARn	0x0500	W	0x00000000	Host Channel-n Characteristics Register
USBOTG_HCSPLTn	0x0504	W	0x00000000	Host Channel-n Split Control Register
USBOTG_HCINTn	0x0508	W	0x00000000	Host Channel-n Interrupt Register
USBOTG_HCINTMSKn	0x050c	W	0x00000000	Host Channel-n Interrupt Mask Register
USBOTG_HCTSIZn	0x0510	W	0x00000000	Host Channel-n Transfer Size Register
USBOTG_HCDMAn	0x0514	W	0x00000000	Host Channel-n DMA Address Register
USBOTG_HCDMABn	0x051c	W	0x00000000	Host Channel-n DMA Buffer Address Register

Name	Offset	Size	Reset Value	Description
USBOTG_DCFG	0x0800	W	0x08200000	Device Cconfiguration Register
USBOTG_DCTL	0x0804	W	0x00002000	Device Control Register
USBOTG_DSTS	0x0808	W	0x00000000	Device Status Register
USBOTG_DIEPMSK	0x0810	W	0x00000000	Device IN Endpoint common interrupt mask register
USBOTG_DOEPMSK	0x0814	W	0x00000000	Device OUT Endpoint common interrupt mask register
USBOTG_DAINT	0x0818	W	0x00000000	Device All Endpoints interrupt register
USBOTG_DAINTMSK	0x081c	W	0x00000000	Device All Endpoint interrupt mask register
USBOTG_DTKNQR1	0x0820	W	0x00000000	Device IN token sequence learning queue read register1
USBOTG_DTKNQR2	0x0824	W	0x00000000	Device IN token sequence learning queue read register2
USBOTG_DVBUSDIS	0x0828	W	0x00000b8f	Device VBUS discharge time register
USBOTG_DVBUSPULSE	0x082c	W	0x00000000	Device VBUS Pulsing Timer Register
USBOTG_DTHRCTL	0x0830	W	0x08100020	Device Threshold Control Register
USBOTG_DIEPEMPMSK	0x0834	W	0x00000000	Device IN endpoint FIFO empty interrupt mask register
USBOTG_DEACHINT	0x0838	W	0x00000000	Device each endpoint interrupt register
USBOTG_DEACHINTMSK	0x083c	W	0x00000000	Device each endpoint interrupt register mask
USBOTG_DIEPEACHMSKn	0x0840	W	0x00000000	Device each IN endpoint -n interrupt Register
USBOTG_DOEPEACHMSKn	0x0880	W	0x00000000	Device each out endpoint-n interrupt register
USBOTG_DIEPCTL0	0x0900	W	0x00008000	Device control IN endpoint 0 control register
USBOTG_DIEPINTn	0x0908	W	0x00000000	Device Endpoint-n Interrupt Register
USBOTG_DIEPTSIZn	0x0910	W	0x00000000	Device endpoint n transfer size register
USBOTG_DIEPDMAn	0x0914	W	0x00000000	Device endpoint-n DMA address register
USBOTG_DTXFSTSn	0x0918	W	0x00000000	Device IN endpoint transmit FIFO status register

Name	Offset	Size	Reset Value	Description
USBOTG_DIEPDMABn	0x091c	W	0x00000000	Device endpoint-n DMA buffer address register
USBOTG_DIEPCTLn	0x0920	W	0x00000000	Device endpoint-n control register
USBOTG_DOEPCTL0	0x0b00	W	0x00000000	Device control OUT endpoint 0 control register
USBOTG_DOEPINTn	0x0b08	W	0x00000000	Device endpoint-n control register
USBOTG_DOEPTSIZn	0x0b10	W	0x00000000	Device endpoint n transfer size register
USBOTG_DOEPDMAn	0x0b14	W	0x00000000	Device Endpoint-n DMA Address Register
USBOTG_DOEPDMABn	0x0b1c	W	0x00000000	Device endpoint-n DMA buffer address register
USBOTG_DOEPCTLn	0x0b20	W	0x00000000	Device endpoint-n control register
USBOTG_PCGCR	0x0b24	W	0x200b8000	Power and clock gating control register
USBOTG_EPBUF0	0x1000	W	0x00000000	Device endpoint 0 / host out channel 0 address
USBOTG_EPBUF1	0x2000	W	0x00000000	Device endpoint 1 / host out channel 1 address
USBOTG_EPBUF2	0x3000	W	0x00000000	Device endpoint 2 / host out channel 2 address
USBOTG_EPBUF3	0x4000	W	0x00000000	Device endpoint 3 / host out channel 3 address
USBOTG_EPBUF4	0x5000	W	0x00000000	Device endpoint 4 / host out channel 4 address
USBOTG_EPBUF5	0x6000	W	0x00000000	Device endpoint 5 / host out channel 5 address
USBOTG_EPBUF6	0x7000	W	0x00000000	Device endpoint 6 / host out channel 6 address
USBOTG_EPBUF7	0x8000	W	0x00000000	Device endpoint 7 / host out channel 7 address

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

27.5.2 Detail Register Description

USBOTG_GOTGCTL

Address: Operational Base + offset (0x0000)

Control and Status Register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved

Bit	Attr	Reset Value	Description
27	RW	0x0	ChirpEn Chirp on enable This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1. If OTG_BC_SUPPORT != 1, this bit is a reserved bit.
26:22	RO	0x00	MultValidBc Multi Valued ID pin Battery Charger ACA inputs in the following order: Bit 26 - rid_float. Bit 25 - rid_gnd Bit 24 - rid_a Bit 23 - rid_b Bit 22 - rid_c These bits are present only if OTG_BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.
21	RO	0x0	reserved
20	RW	0x0	OTGVer OTG version Indicates the OTG revision. 0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. 1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.
19	RO	0x0	BSesVld B-session valid Indicates the Device mode transceiver status. 0: B-session is not valid. 1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected. Note: If you do not enabled OTG features (such as SRP and HNP), the read reset value will be 1.The vbus assigns the values internally for non-SRP or non-HNP configurations.

Bit	Attr	Reset Value	Description
18	RO	0x0	ASesVld A-session valid Indicates the Host mode transceiver status. 0: A-session is not valid 1: A-session is valid Note: If you do not enabled OTG features (such as SRP and HNP), the read reset value will be 1.The vbus assigns the values internally for non-SRP or non-HNP configurations.
17	RO	0x0	DbnTime Long/short debounce time Indicates the debounce time of a detected connection. 0: Long debounce time, used for physical connections (100 ms + 2.5 us) 1: Short debounce time, used for soft connections (2.5 us)
16	RO	0x0	ConIDSts Connector ID Status Indicates the connector ID status on a connect event. 0: The core is in A-Device mode 1: The core is in B-Device mode
15:12	RO	0x0	reserved
11	RW	0x0	DevHNPEn Device HNP Enable The application sets this bit when it successfully receives a SetFeature. SetHNPEnable command from the connected USB host. 0: HNP is not enabled in the application 1: HNP is enabled in the application
10	RW	0x0	HstSetHNPEn Host set HNP enable The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. 0: Host Set HNP is not enabled 1: Host Set HNP is enabled

Bit	Attr	Reset Value	Description
9	RW	0x0	HNPReq HNP request The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. 0: No HNP request 1: HNP request
8	RO	0x0	HstNegScs Host Negotiation Success The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. 0: Host negotiation failure 1: Host negotiation success
7:2	RO	0x0	reserved
1	RW	0x0	SesReq Session Request The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. 0: No session request 1: Session request

Bit	Attr	Reset Value	Description
0	RO	0x0	SesReqScs Session Request Success The core sets this bit when a session request initiation is successful. 0: Session request failure 1: Session request success

USBOTG_GOTGINT
Address: Operational Base + offset (0x0004)
Interrupt Register

Bit	Attr	Reset Value	Description
31:21	RO	0x0	reserved
20	W1C	0x0	MultiValueChg Multi-Valued input changed This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT = 1, otherwise it is reserved.
19	W1C	0x0	DbnceDone Debounce Done The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).
18	W1C	0x0	ADevTOUTChg A-Device Timeout Change The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.
17	W1C	0x0	HstNegDet Host Negotiation Detected The core sets this bit when it detects a host negotiation request on the USB
16:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	W1C	0x0	HstNegSucStsChng Host Negotiation Success Status Change The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure
8	W1C	0x0	SesReqSucStsChng Session Request Success Status Change The core sets this bit on the success or failure of a session request.The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.
7:3	RO	0x0	reserved
2	W1C	0x0	SesEndDet Session End Detected The core sets this bit when the utmisrp_bvalid signal is deasserted
1:0	RO	0x0	reserved

USBOTG_GAHBCFG

Address: Operational Base + offset (0x0008)

AHB Configuration Register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22	RW	0x0	NotiAllDmaWrit Notify All Dma Write Transactions This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. GAHBCFG.NotiAllDmaWrit = 1. HSOTG core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. GAHBCFG.NotiAllDmaWrit = 0. HSOTG core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.

Bit	Attr	Reset Value	Description
21	RW	0x0	RemMemSupp Remote Memory Support This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write Transfers. GAHBCFG.RemMemSupp=1. The int_dma_req output signal is asserted when HSOTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HSOTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. GAHBCFG.RemMemSupp=0. The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HSOTG Core Boundary and it does not wait for the sys_dma_done signal to complete the DATA transfers.
20:9	RO	0x0	reserved
8	RW	0x0	PTxFEmpLvl Periodic TxFIFO Empty Level Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. 0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty 1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty

Bit	Attr	Reset Value	Description
7	RW	0x0	NPTxFEmpLvl Non-Periodic TxFIFO Empty Level This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode: 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty Dedicated FIFO in device mode: 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty
6	RO	0x0	reserved
5	RW	0x0	DMAEn DMA Enable 0: Core operates in Slave mode 1: Core operates in a DMA mode This bit is always 0 when Slave-Only mode has been selected.

Bit	Attr	Reset Value	Description
4:1	RW	0x0	HBstLen Burst Length/Type This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, External DMA Mode defines the DMA burst length in terms of 32-bit words: 4'b0000: 1 word 4'b0001: 4 words 4'b0010: 8 words 4'b0011: 16 words 4'b0100: 32 words 4'b0101: 64 words 4'b0110: 128 words 4'b0111: 256 words Others: Reserved Internal DMA Mode AHB Master burst type: 4'b0000: Single 4'b0001: INCR 4'b0011: INCR4 4'b0101: INCR8 4'b0111: INCR16 Others: Reserved
0	RW	0x0	GlblIntrMsk Global Interrupt Mask The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. 1'b0: Mask the interrupt assertion to the application. 1'b1: Unmask the interrupt assertion to the application.

USBOTG_GUSBCFG
Address: Operational Base + offset (0x000c)
USB Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	CorruptTxpacket Corrupt Tx packet This bit is for debug purposes only. Never set this bit to 1.

Bit	Attr	Reset Value	Description
30	RW	0x0	ForceDevMode Force Device Mode Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. 1'b0: Normal Mode 1'b1: Force Device Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE = 0, 1 or 2. In all other cases, this bit reads 0.
29	RW	0x0	ForceHstMode Force Host Mode Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. 1'b0: Normal Mode 1'b1: Force Host Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 us is sufficient. This bit is valid only when OTG_MODE =0, 1 or 2. In allother cases, this bit reads 0.
28	RW	0x0	TxEndDelay Tx End Delay Writing a 1 to this bit enables the TxEndDelay timers in the core. 1'b0: Normal mode 1'b1: Introduce Tx end delay timers
27	RW	0x0	IC_USBTrafCtl IC_USB TrafficPullRemove Control When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling.This bit is valid only when configuration parameter OTG_ENABLE_IC_USB = 1 and register field GUSBCFG.IC_USBCap is set to 1.

Bit	Attr	Reset Value	Description
26	RW	0x0	IC_USBCap IC_USB-Capable The application uses this bit to control the IC_USB capabilities. 1'b0: IC_USB PHY Interface is not selected. 1'b1: IC_USB PHY Interface is selected. This bit is writable only if OTG_ENABLE_IC_USB=1 and OTG_FSPHY_INTERFACE!=0.The reset value depends on the configuration parameter OTG_SELECT_IC_USB when OTG_ENABLE_IC_USB = 1. In all other cases, this bit is set to 1'b0 and the bit is read only.
25	RW	0x0	ULPIIfDis ULPI Interface Protect Disable Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. Please refer to the ULPI Specification for more detail. 1'b0: Enables the interface protect circuit 1'b1: Disables the interface protect circuit
24	RW	0x0	IndPassThrough Indicator Pass Through Controls wether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. Please refer to the ULPI Specification for more detail. 1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator.

Bit	Attr	Reset Value	Description
23	RW	0x0	IndComple Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal,generating the Complement Output. Please refer to the ULPI Specification for more detail 1'b0: PHY does not invert ExternalVbusIndicator signal 1'b1: PHY does invert ExternalVbusIndicator signal
22	RW	0x0	TermSelDLPulse TermSel DLine Pulsing Selection This bit selects utmi_termselect to drive data line pulse during SRP. 1'b0: Data line pulsing using utmi_txvalid (default). 1'b1: Data line pulsing using utmi_termsel.
21	RW	0x0	ULPIExtVbusIndicator ULPI External VBUS Indicator This bit indicates to the ULPI PHY to use an external VBUS over-current indicator. 1'b0: PHY uses internal VBUS valid comparator. 1'b1: PHY uses external VBUS valid comparator. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
20	RW	0x0	ULPIExtVbusDrv ULPI External VBUS Drive This bit selects between internal or external supply to drive 5V on VBUS,in ULPI PHY. 1'b0: PHY drives VBUS using internal charge pump (default). 1'b1: PHY drives VBUS using external supply. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)

Bit	Attr	Reset Value	Description
19	RW	0x0	ULPIClkSusM ULPI Clock SuspendM This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. 1'b0: PHY powers down internal clock during suspend. 1'b1: PHY does not power down internal clock. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
18	RW	0x0	ULPIAutoRes ULPI Auto Resume This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. 1'b0: PHY does not use AutoResume feature. 1'b1: PHY uses AutoResume feature. (Valid only when RTL parameter OTG_HSPHY_INTERFACE = 2 or 3)
17	RW	0x0	ULPIFsLs ULPI FS/LS Select The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. 1'b0: ULPI interface 1'b1: ULPI FS/LS serial interface (Valid only when RTL parameters OTG_HSPHY_INTERFACE = 2 or 3 and OTG_FSPHY_INTERFACE = 1, 2, or 3)
16	RW	0x0	OtgI2CSel UTMIFS or I2C Interface Select The application uses this bit to select the I2C interface. 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals 1'b1: I2C interface for OTG signals This bit is writable only if I2C and UTMIFS were specified for Enable I2C Interface? (parameter OTG_I2C_INTERFACE = 2). Otherwise, reads return 0.

Bit	Attr	Reset Value	Description
15	RW	0x0	PhyLPwrClkSel PHY Low-Power Clock Select Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. 1'b0: 480-MHz Internal PLL clock 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS and LS modes. This bit drives the utmi_fsls_low_power core output signal, and is valid only for UTMI+ PHYs.
14	RO	0x0	reserved
13:10	RW	0x5	USBTrdTim USB Turnaround Time Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIF(SPRAM). This must be programmed to 4'h5: When the MAC interface is 16-bit UTMI+. 4'h9: When the MAC interface is 8-bit UTMI+. Note: The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.
9	RW	0x0	HNPCap HNP-Capable The application uses this bit to control the otg core's HNP capabilities. 0: HNP capability is not enabled. 1: HNP capability is enabled. This bit is writable only if an HNP mode was specified for Mode of Operation (parameter OTG_MODE). Otherwise, reads return 0.

Bit	Attr	Reset Value	Description
8	RW	0x0	SRPCap SRP-Capable The application uses this bit to control the otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. 0: SRP capability is not enabled. 1: SRP capability is enabled. This bit is writable only if an SRP mode was specified for Mode of Operation (parameter OTG_MODE). Otherwise, reads return 0.
7	RW	0x0	DDRSel ULPI DDR Select The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface. 0: Single Data Rate ULPI Interface, with 8-bit-wide data bus 1: Double Data Rate ULPI Interface, with 4-bit-wide data bus
6	RW	0x0	PHYSel USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. 0: USB 2.0 high-speed UTMI+ or ULPI PHY 1: USB 1.1 full-speed serial transceiver If a USB 1.1 Full-Speed Serial Transceiver interface was not selected (parameter OTG_FSPHY_INTERFACE = 0), this bit is always 0, with Write Only access. If a high-speed PHY interface was not selected (parameter OTG_HSPHY_INTERFACE = 0), this bit is always 1, with Write Only access. If both interface types were selected (parameters have non-zero values), the application uses this bit to select which interface is active, and access is Read and Write.

Bit	Attr	Reset Value	Description
5	RW	0x0	FSIntf Full-Speed Serial Interface Select The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. 0: 6-pin unidirectional full-speed serial interface 1: 3-pin bidirectional full-speed serial interface If a USB 1.1 Full-Speed Serial Transceiver interface was not selected (parameter OTG_FSPHY_INTERFACE = 0), this bit is always 0, with Write Only access. If a USB 1.1 FS interface was selected (parameter OTG_FSPHY_INTERFACE! = 0), then the application can set this bit to select between the 3- and 6-pin interfaces, and access is Read and Write.
4	RW	0x0	ULPI_UTMI_Sel ULPI or UTMI+ Select The application uses this bit to select either a UTMI+ interface or ULPI Interface. 0: UTMI+ Interface 1: ULPI Interface This bit is writable only if UTMI+ and ULPI was specified for High-Speed PHY Interface(s) (parameter OTG_HSPHY_INTERFACE = 3). Otherwise, reads return either 0 or 1, depending on the interface selected using the OTG_HSPHY_INTERFACE parameter.
3	RW	0x0	PHYIf PHY Interface The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. 0: 8 bits 1: 16 bits This bit is writable only if UTMI+ and ULPI were selected (parameter OTG_HSPHY_DWIDTH = 3). Otherwise, this bit returns the value for the power-on interface selected during configuration.

Bit	Attr	Reset Value	Description
2:0	RW	0x0	TOutCal HS/FS Timeout Calibration The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the line state condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full-speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times One 48-MHz PHY clock = 0.25 bit times

USBOTG_GRSTCTL
Address: Operational Base + offset (0x0010)
Reset Register

Bit	Attr	Reset Value	Description
31	RO	0x1	AHBIdle AHB Master Idle Indicates that the AHB Master State Machine is in the IDLE condition.
30	RO	0x0	DMAReq DMA Request Signal Indicates that the DMA request is in progress. Used for debug.
29:11	RO	0x0	reserved

Bit	Attr	Reset Value	Description
10:6	RW	0x00	TxFNum TxFIFO Number This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. 5'h0: Non-periodic TxFIFO flush in Host mode; Non-periodic TxFIFO flush in device mode when in shared FIFO operation. Tx FIFO 0 flush in device mode when in dedicated FIFO mode. 5'h1: Periodic TxFIFO flush in Host mode: Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation; TXFIFO 1 flush in device mode when in dedicated FIFO mode. 5'h2: Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation: TXFIFO 2 flush in device mode when in dedicated FIFO mode. ... 5'hF: Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation: TXFIFO 15 flush in device mode when in dedicated FIFO mode. 5'h10: Flush all the transmit FIFOs in device or host mode.

Bit	Attr	Reset Value	Description
5	R/WSC	0x0	TxFFlsh TxFIFO Flush This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: Read NAK Effective Interrupt ensures the core is not reading from the FIFO. Write GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are re-configured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.
4	R/WSC	0x0	RxFFlsh RxFIFO Flush The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.
3	R/WSC	0x0	INTknQFlsh IN Token Sequence Learning Queue Flush This bit is valid only if OTG_EN_DED_TX_FIFO = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.

Bit	Attr	Reset Value	Description
2	W1C	0x0	FrmCntrRst Host Frame Counter Reset The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0.
1	R/WSC	0x0	Reset A write to this bit issues a soft reset to the otg_power_dn module of the core.

Bit	Attr	Reset Value	Description
0	R/WSC	0x0	CSftRst Core Soft Reset Resets the hclk and phy_clock domains as follows: Clears the interrupts and all the CSR registers except the following register bits: PCGCCTL.RstPdwnModule PCGCCTL.GateHclk PCGCCTL.PwrClmp PCGCCTL.StopPPhyLPwrClkSelclk GUSBCFG.PhyLPwrClkSel GUSBCFG.DDRSel GUSBCFG.PHYSel GUSBCFG.FSIntf GUSBCFG.ULPI_UTMI_Sel GUSBCFG.PHYIf HCFG.FSLSPclkSel DCFG.DevSpd GGPIO GPWRDN GADPCTL All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. Any transactions on the AHB Master are terminated as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for

USBOTG_GINTSTS
Address: Operational Base + offset (0x0014)
Interrupt Register

Bit	Attr	Reset Value	Description
31	W1C	0x0	WkUpInt Resume/Remote Wakeup Detected Interrupt Wakeup Interrupt during Suspend(L2) or LPM(L1) state. During Suspend(L2): Device Mode:This interrupt is asserted only when Host Initiated Resume is detected on USB. Host Mode:This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. During LPM(L1): Device Mode:This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. Host Mode:This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.
30	W1C	0x0	SessReqInt Session Request/New Session Detected Interrupt In Host mode, this interrupt is asserted when a session request is detected from the device. In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high.
29	W1C	0x0	DisconnInt Disconnect Detected Interrupt This interrupt is asserted when a device disconnect is detected.
28	W1C	0x0	ConIDStsChng Connector ID Status Change This interrupt is asserted when there is a change in connector ID status.

Bit	Attr	Reset Value	Description
27	W1C	0x0	LPM_Int LPM Transaction Received Interrupt Device Mode : This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. Host Mode : This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt). This field is valid only if the Core LPM Configuration register's LPMCapable (LPMCap) field is set to 1.
26	RO	0x0	PTxFEmp Periodic TxFIFO Empty This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).
25	RO	0x0	HChInt Host Channels Interrupt The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.

Bit	Attr	Reset Value	Description
24	RO	0x0	PrtInt Host Port Interrupt The core sets this bit to indicate a change in port status of one of the otg core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.
23	RW	0x0	ResetDet Reset Detected Interrupt The core asserts this interrupt in Device mode when it detects a reset on the USB in Partial Power-Down mode when the device is in Suspend. This interrupt is not asserted in Host mode.

Bit	Attr	Reset Value	Description
22	W1C	0x0	FetSusp Data Fetch Suspended This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: Sets a global non-periodic IN NAK handshake, Disables In endpoints, Flushes the FIFO, Determines the token sequence from the IN Token Sequence Learning Queue, Re-enables the endpoints, Clears the global non-periodic IN NAK handshake. If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received: the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the IN token received when FIFO empty?interrupt when clearing a global IN NAK handshake.
21	W1C	0x0	incomplP Incomplete Periodic Transfer In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.

Bit	Attr	Reset Value	Description
20	W1C	0x0	incompISOIN Incomplete Isochronous IN Transfer The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA mode.
19	RO	0x0	OEPInt OUT Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.
18	RO	0x0	IEPInt IN Endpoints Interrupt The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.

Bit	Attr	Reset Value	Description
17	W1C	0x0	EPMis Endpoint Mismatch Interrupt Note: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.
16	W1C	0x0	RstrDoneInt Restore Done Interrupt The core sets this bit to indicate that the restore command after Hibernation was completed by the core. The core continues from Suspended state into the mode dictated by PCGCCTL.RestoreMode field.This bit is valid only when Hibernation feature is enabled.
15	W1C	0x0	EOPF End of Periodic Frame Interrupt Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.
14	W1C	0x0	ISOOutDrop Isochronous OUT Packet Dropped Interrupt The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.
13	W1C	0x0	EnumDone Enumeration Done The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.
12	W1C	0x0	USBRst USB Reset The core sets this bit to indicate that a reset is detected on the USB.

Bit	Attr	Reset Value	Description
11	W1C	0x0	USBSusp USB Suspend The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time.
10	W1C	0x0	ErlySusp Early Suspend The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.
9	W1C	0x0	I2CINT I2C Interrupt The core sets this interrupt when I2C access is completed on the I2C interface. This field is used only if the I2C interface was enabled . Otherwise, reads return 0.
8	W1C	0x0	ULPICKINT ULPI Carkit Interrupt This field is used only if the Carkit interface was enabled . Otherwise, reads return 0. The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. I2C Carkit Interrupt (I2CCKINT) This field is used only if the I2C interface was enabled . Otherwise, reads return 0.The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I2C Carkit interrupt in Audio mode.
7	RO	0x0	GOUTNakEff Global OUT NAK Effective Indicates that the Set Global OUT NAK bit in the Device Control register DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).

Bit	Attr	Reset Value	Description
6	RO	0x0	GINNakEff Global IN Non-Periodic NAK Effective Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Nonperiodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.
5	RO	0x0	NPTxFEmp Non-Periodic TxFIFO Empty This interrupt is valid only when OTG_EN_DED_TX_FIFO = 0. This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register(GAHBCFG.NPTxFEmpLvl).
4	RO	0x0	RxFLvl RxFIFO Non-Empty Indicates that there is at least one packet pending to be read from the RxFIFO.
3	W1C	0x0	Sof Start of (micro)Frame In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF(HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)frame number. This interrupt is seen only when the core is operating at either HS or FS.

Bit	Attr	Reset Value	Description
2	RO	0x0	OTGInt OTG Interrupt The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.
1	W1C	0x0	ModeMis Mode Mismatch Interrupt The core sets this bit when the application is trying to access: A Host mode register, when the core is operating in Device mode ; A Device mode register, when the core is operating in Host mode. The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core.
0	RO	0x0	CurMod Current Mode of Operation Indicates the current mode. 1'b0: Device mode 1'b1: Host mode

USBOTG_GINTMSK
Address: Operational Base + offset (0x0018)
Interrupt Mask Register

Bit	Attr	Reset Value	Description
31	RW	0x0	WkUpIntMsk Resume/Remote Wakeup Detected Interrupt Mask
30	RW	0x0	SessReqIntMsk Session Request/New Session Detected Interrupt Mask
29	RW	0x0	DisconnIntMsk Disconnect Detected Interrupt Mask
28	RW	0x0	ConIDStsChngMsk Connector ID Status Change Mask
27	RW	0x0	LPM_IntMsk LPM Transaction Received Interrupt Mask
26	RW	0x0	PTxFEmpMsk Periodic TxFIFO Empty Mask

Bit	Attr	Reset Value	Description
25	RW	0x0	HChIntMsk Host Channels Interrupt Mask
24	RW	0x0	PrtIntMsk Host Port Interrupt Mask
23	RW	0x0	ResetDetMsk Reset Detected Interrupt Mask
22	RW	0x0	FetSuspMsk Data Fetch Suspended Mask
21	RW	0x0	incomplPMsk_incompISOOUTMsk Incomplete Periodic Transfer Mask(Host only) Incomplete Isochronous OUT Transfer Mask(Device only)
20	RW	0x0	incompISOINMsk Incomplete Isochronous IN Transfer Mask
19	RW	0x0	OEPIntMsk OUT Endpoints Interrupt Mask
18	RW	0x0	IEPIntMsk IN Endpoints Interrupt Mask
17	RW	0x0	EPMisMsk Endpoint Mismatch Interrupt Mask
16	RW	0x0	RstrDoneIntMsk Restore Done Interrupt Mask This field is valid only when Hibernation feature is enabled.
15	RW	0x0	EOPFMsk End of Periodic Frame Interrupt Mask
14	RW	0x0	ISOOutDropMsk Isochronous OUT Packet Dropped Interrupt Mask
13	RW	0x0	EnumDoneMsk Enumeration Done Mask
12	RW	0x0	USBRstMsk USB Reset Mask
11	RW	0x0	USBSuspMsk USB Suspend Mask
10	RW	0x0	ErlySuspMsk Early Suspend Mask
9	RW	0x0	I2CIntMsk I2C Interrupt Mask
8	RW	0x0	ULPICKINTMsk_I2CCKINTMsk ULPI Carkit Interrupt Mask (ULPICKINTMsk) I2C Carkit Interrupt Mask (I2CCKINTMsk)

Bit	Attr	Reset Value	Description
7	RW	0x0	GOUTNakEffMsk Global OUT NAK Effective Mask
6	RW	0x0	GINNakEffMsk Global Non-periodic IN NAK Effective Mask
5	RW	0x0	NPTxFEmpMsk Non-periodic TxFIFO Empty Mask
4	RW	0x0	RxFLvlMsk Receive FIFO Non-Empty Mask
3	RW	0x0	SofMsk Start of (micro)Frame Mask
2	RW	0x0	OTGIntMsk OTG Interrupt Mask
1	RW	0x0	ModeMisMsk Mode Mismatch Interrupt Mask
0	RO	0x0	reserved

USBOTG_GRXSTSR

Address: Operational Base + offset (0x001c)

Receive Status Debug Read Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:21	RO	0x0	FN Frame Number (Device Only)This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.

Bit	Attr	Reset Value	Description
20:17	RO	0x0	PktSts Packet Status Indicates the status of the received packet(Host Only) 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Indicates the status of the received packet(Device only) 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved
16:15	RO	0x0	DPID Data PID Indicates the Data PID of the received packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA
14:4	RW	0x000	BCnt Byte Count Indicates the byte count of the received data packet.
3:0	RO	0x0	ChNum_EPNum Channel Number(Host) Endpoint Number(Device) (Host Only) Indicates the channel number to which the current received packet belongs. (Device Only) Indicates the endpoint number to which the current received packet belongs.

USBOTG_GRXSTSP

Address: Operational Base + offset (0x0020)

Receive Status Read and Pop Register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:21	RO	0x0	FN Frame Number (Device Only) This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.
20:17	RO	0x0	PktSts Packet Status Indicates the status of the received packet(Host Only) 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved Indicates the status of the received packet(Device only) 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved
16:15	RO	0x0	DPID Data PID Indicates the Data PID of the received OUT data packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA

Bit	Attr	Reset Value	Description
14:4	RO	0x000	BCnt Byte Count Indicates the byte count of the received data packet.
3:0	RO	0x0	ChNum_EPNum Channel Number(Host) Endpoint Number(Device) (Host Only) Indicates the channel number to which the current received packet belongs. (Device Only) Indicates the endpoint number to which the current received packet belongs.

USBOTG_GRXFSIZ

Address: Operational Base + offset (0x0024)

Receive FIFO Size Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	RxFDep RxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16, Maximum value is 32,768. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth. If Enable Dynamic FIFO Sizing? was deselected, these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected , you can write a new value in this field. You can write a new value in this field. Programmed values must not exceed the power-on value.

USBOTG_GNPTXFSIZ

Address: Operational Base + offset (0x0028)

Non-Periodic Transmit FIFO Size Register (Host mode)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	NPTxFDep Non-periodic TxFIFO For host mode, this field is always valid. For Device mode, this field is valid only when OTG_EN_DED_TX_FIFO==0. This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 This field is determined by Enable Dynamic FIFO Sizing. OTG_DFIFO_DYNAMIC = 0: These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1: The application can write a new value in this field. Programmed values must not exceed the power-on value. The power-on reset value of this field is specified by OTG_EN_DED_TX_FIFO: OTG_EN_DED_TX_FIFO = 0:The reset value is the Largest Non-periodic Tx Data FIFO Depth parameter, OTG_TX_NPERIO_DFIFO_DEPTH. OTG_EN_DED_TX_FIFO = 1:The reset value is parameter OTG_TX_HNPERIO_DFIFO_DEPTH.
15:0	RW	0x0000	NPTxFStAddr Non-periodic Transmit RAM For host mode, this field is always valid. This field contains the memory start address for Non-periodic Transmit FIFO RAM. This field is determined by Enable Dynamic FIFO Sizing?(OTG_DFIFO_DYNAMIC): OTG_DFIFO_DYNAMIC = 0 :These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1 :The application can write a new value in this field. Programmed values must not exceed the power-on value. The power-on reset value of this field is specified by Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH).

Non-Periodic Transmit FIFO Size Register (Device mode)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	INEPTxF0Dep IN Endpoint TxFIFO 0 Depth This field is valid only for Device mode and when OTG_EN_DED_TX_FIFO = 1 This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 This field is determined by Enable Dynamic FIFO Sizing? (OTG_TX_DINEP_DFIFO_DEPTH_0): OTG_DFIFO_DYNAMIC = 0—These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1—Programmed values must not exceed the power-on value. The power-on reset value of this field is specified as Largest IN Endpoint FIFO 0 Depth (parameter OTG_TX_DINEP_DFIFO_DEPTH_0).
15:0	RW	0x0000	INEPTxF0StAddr IN Endpoint FIFO0 Transmit RAM Start Address For Device mode this field is valid only when OTG_EN_DED_TX_FIFO = 0 This field contains the memory start address for IN Endpoint Transmit FIFO# 0. OTG_RX_DFIFO_DEPTH This field is determined by Enable Dynamic FIFO Sizing? (OTG_DFIFO_DYNAMIC): OTG_DFIFO_DYNAMIC = 0 —These flops are optimized, and reads return the power-on value. OTG_DFIFO_DYNAMIC = 1—The application can write a new value in this field. Programmed values must not exceed the power-on value. The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH).

USBOTG_GNPTXSTS

Address: Operational Base + offset (0x002c)

Non-Periodic Transmit FIFO/Queue Status Register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:24	RO	0x00	NPTxQTop Top of the Non-periodic Transmit Request Queue Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. Bits [30:27]: Channel/endpoint number Bits [26:25]: 2'b00: IN/OUT token 2'b01: Zero-length transmit packet (device IN/host OUT) 2'b10: PING/CSPLIT token 2'b11: Channel halt command Bit [24]: Terminate (last entry for selected channel/endpoint)
23:16	RO	0x00	NPTxQSpcAvail Non-periodic Transmit Request Queue Space Available Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. 8'h0: Non-periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available n: n locations available (0 <=n <= 8) Others: Reserved
15:0	RO	0x0000	NPTxFSpcAvail Non-periodic TxFIFO Space Avail Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. 16'h0: Non-periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 <=n <=32,768) 16'h8000: 32,768 words available Others: Reserved

USBOTG_GI2CCTL

Address: Operational Base + offset (0x0030)

I2C Address Register

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	BsyDne I2C Busy/Done The application sets this bit to 1'b1 to start a request on the I2C interface. When the transfer is complete, the core deasserts this bit to 1'b0. As long as the bit is set, indicating that the I2C interface is busy, the application cannot start another request on the interface.
30	RW	0x0	RW Read/Write Indicator Indicates whether a read or write register transfer must be performed on the interface.Read/write bursting is not supported for registers. 1'b1: Read 1'b0: Write
29	RO	0x0	reserved
28	RW	0x1	I2CDatSe0 I2C DatSe0 USB Mode Selects the FS interface USB mode. 1'b1: VP_VM USB mode 1'b0: DAT_SE0 USB mode
27:26	RW	0x0	I2CDevAdr I2C Device Address Selects the address of the I2C Slave on the USB 1.1 full-speed serial transceiver that the core uses for OTG signaling. 2'b00: 7'h2C 2'b01: 7'h2D 2'b10: 7'h2E 2'b11: 7'h2F
25	RW	0x0	I2CSuspCtl I2C Suspend Control Selects how Suspend is connected to a full-speed transceiver in I2C mode. 1'b0: Use the dedicated utmi_suspend_n pin 1'b1: Use an I2C write to program the Suspend bit in the PHY register

Bit	Attr	Reset Value	Description
24	RO	0x1	Ack I2C ACK Indicates whether an ACK response was received from the I2C Slave. This bit is valid when BsyDne is cleared by the core, after application has initiated an I2C access. 1'b0: NAK 1'b1: ACK
23	RW	0x0	I2CEn I2C Enable Enables the I2C Master to initiate I2C transactions on the I2C interface
22:16	RW	0x00	Addr I2C Address This is the 7-bit I2C device address used by software to access any external I2C Slave,including the I2C Slave on a USB 1.1 OTG full-speed serial transceiver. Software can change this address to access different I2C Slaves.
15:8	RW	0x00	RegAddr I2C Register Addr This field programs the address of the register to be read from or written to.
7:0	RW	0x00	RWData I2C Read/Write Data After a register read operation, this field holds the read data for the application. During a write operation, the application can use this register to program the write data to be written to a register. During writes, this field holds the write data.

USBOTG_GPVNDCTL

Address: Operational Base + offset (0x0034)

PHY Vendor Control Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	DisUlpiDrvr Disable ULPI Drivers This field is used only if the Carkit interface was enabled (parameter OTG_ULPI_CARKIT = 1). Otherwise, reads return 0. The application sets this bit when it has finished processing the ULPI Carkit Interrupt (GINTSTS.ULPICKINT). When set, the otg core disables drivers for output signals and masks input signal for the ULPI interface. Otg clears this bit before enabling the ULPI interface.
30:28	RO	0x0	reserved
27	R/WSC	0x0	VStsDone VStatus Done The core sets this bit when the vendor control access is done. This bit is cleared by the core when the application sets the New Register Request bit (bit 25).
26	RO	0x0	VStsBsy VStatus Busy The core sets this bit when the vendor control access is in progress and clears this bit when done.
25	R/WSC	0x0	NewRegReq New Register Request The application sets this bit for a new vendor control access.
24:23	RO	0x0	reserved
22	RW	0x0	RegWr Register Write Set this bit for register writes, and clear it for register reads.
21:16	RW	0x00	RegAddr Register Address The 6-bit PHY register address for immediate PHY Register Set access. Set to 6'h2F for Extended PHY Register Set access.

Bit	Attr	Reset Value	Description
15:8	RW	0x00	VCtrl UTMI+ Vendor Control Register Address The 4-bit register address a vendor defined 4-bit parallel output bus. Bits 11:8 of this field are placed on utmi_vcontrol[3:0]. ULPI Extended Register Address (ExtRegAddr) The 6-bit PHY extended register address.
7:0	RW	0x00	RegData Register Data Contains the write data for register write. Read data for register read, valid when VStatus Done is set.

USBOTG_GGPIO
Address: Operational Base + offset (0x0038)
General Purpost Input/Output Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	GPO General Purpose Output This field is driven as an output from the core, gp_o[15:0]. The application can program this field to determine the corresponding value on the gp_o[15:0] output.
15:0	RO	0x0000	GPI General Purpose Input This field's read value reflects the gp_i[15:0] core input value.

USBOTG_GUID
Address: Operational Base + offset (0x003c)
User ID Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	UserID Application-programmable ID field.

USBOTG_GSNPSID
Address: Operational Base + offset (0x0040)
Core ID Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00004f54	CoreID Release number of the core being used

USBOTG_GHWCFG1
Address: Operational Base + offset (0x0044)
User HW Config1 Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	epdir Endpoint Direction This 32-bit field uses two bits per endpoint to determine the endpoint direction. Endpoint Bits [31:30]: Endpoint 15 direction Bits [29:28]: Endpoint 14 direction ... Bits [3:2]: Endpoint 1 direction Bits[1:0]: Endpoint 0 direction (always BIDIR) Direction 2'b00: BIDIR (IN and OUT) endpoint 2'b01: IN endpoint 2'b10: OUT endpoint 2'b11: Reserved

USBOTG_GHWCFG2
Address: Operational Base + offset (0x0048)
User HW Config2 Register

Bit	Attr	Reset Value	Description
31	RO	0x0	OTG_ENABLE_IC_USB IC_USB mode specified for mode of operation (parameter OTG_ENABLE_IC_USB). To choose IC_USB_MODE, both OTG_FSPHY_INTERFACE and OTG_ENABLE_IC_USB must be 1.
30:26	RO	0x00	TknQDepth Device Mode IN Token Sequence Learning Queue Depth Range: 0-30
25:24	RO	0x0	PTxQDepth Host Mode Periodic Request Queue Depth 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved

Bit	Attr	Reset Value	Description
23:22	RO	0x0	NPTxQDepth Non-periodic Request Queue Depth 2'b00: 2 2'b01: 4 2'b10: 8 Others: Reserved
21	RO	0x0	reserved
20	RO	0x0	MultiProcIntrpt Multi Processor Interrupt Enabled 1'b0: No 1'b1: Yes
19	RO	0x0	DynFifoSizing Dynamic FIFO Sizing Enabled 1'b0: No 1'b1: Yes
18	RO	0x0	PerioSupport Periodic OUT Channels Supported in Host Mode 1'b0: No 1'b1: Yes
17:14	RO	0x0	NumHstChnl Number of Host Channels Indicates the number of host channels supported by the core in Host mode. The range of this field is 0-15: 0 specifies 1 channel, 15 specifies 16 channels.
13:10	RO	0x0	NumDevEps Number of Device Endpoints Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1-15.
9:8	RO	0x0	FSPhyType Full-Speed PHY Interface Type 2'b00: Full-speed interface not supported 2'b01: Dedicated full-speed interface 2'b10: FS pins shared with UTMI+ pins 2'b11: FS pins shared with ULPI pins
7:6	RO	0x0	HSPhyType High-Speed PHY Interface Type 2'b00: High-Speed interface not supported 2'b01: UTMI+ 2'b10: ULPI 2'b11: UTMI+ and ULPI

Bit	Attr	Reset Value	Description
5	RO	0x0	SingPnt Point-to-Point 1'b0: Multi-point application (hub and split support) 1'b1: Single-point application (no hub and no split support)
4:3	RO	0x0	OtgArch Architecture 2'b00: Slave-Only 2'b01: External DMA 2'b10: Internal DMA Others: Reserved
2:0	RO	0x0	OtgMode Mode of Operation 3'b000: HNP- and SRP-Capable OTG (Host and Device) 3'b001: SRP-Capable OTG (Host and Device) 3'b010: Non-HNP and Non-SRP Capable OTG (Host and Device) 3'b011: SRP-Capable Device 3'b100: Non-OTG Device 3'b101: SRP-Capable Host 3'b110: Non-OTG Host Others: Reserved

USBOTG_GHWCFG3
Address: Operational Base + offset (0x004c)
User HW Config3 Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	DfifoDepth DFIFO Depth This value is in terms of 32-bit words. Minimum value is 32 Maximum value is 32,768
15	RO	0x0	OTG_ENABLE_LPM LPM mode specified for Mode of Operation (parameter OTG_ENABLE_LPM).
14	RO	0x0	OTG_BC_SUPPORT This bit indicates the HS OTG controller support for Battery Charger. 0 : No Battery Charger Support 1 : Battery Charger support present.

Bit	Attr	Reset Value	Description
13	RO	0x0	OTG_ENABLE_HSIC HSIC mode specified for Mode of Operation (parameter OTG_ENABLE_HSIC). Value Range: 0-1 1: HSIC-capable with shared UTMI PHY interface 0: Non-HSIC-capable
12	RO	0x0	OTG_ADP_SUPPORT This bit indicates whether ADP logic is present within or external to the HS OTG controller 0: No ADP logic present with HSOTG controller 1: ADP logic is present along with HSOTG controller.
11	RO	0x0	RstType Reset Style for Clocked always Blocks in RTL 1'b0: Asynchronous reset is used in the core 1'b1: Synchronous reset is used in the core
10	RO	0x0	OptFeature Optional Features Removed Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features? 1'b0: No 1'b1: Yes
9	RO	0x0	VndctlSupt Vendor Control Interface Support 1'b0: Vendor Control Interface is not available on the core. 1'b1: Vendor Control Interface is available.
8	RO	0x0	I2CIntSel I2C Selection 1'b0: I2C Interface is not available on the core. 1'b1: I2C Interface is available on the core.

Bit	Attr	Reset Value	Description
7	RO	0x0	OtgEn OTG Function Enabled The application uses this bit to indicate theotg core's OTG capabilities. 1'b0: Not OTG capable 1'b1: OTG Capable
6:4	RO	0x0	PktSizeWidth Width of Packet Size Counters 3'b000: 4 bits 3'b001: 5 bits 3'b010: 6 bits 3'b011: 7 bits 3'b100: 8 bits 3'b101: 9 bits 3'b110: 10 bits Others: Reserved
3:0	RO	0x0	XferSizeWidth Width of Transfer Size Counters 4'b0000: 11 bits 4'b0001: 12 bits ... 4'b1000: 19 bits Others: Reserved

USBOTG_GHWCFG4

Address: Operational Base + offset (0x0050)

User HW Config4 Register

Bit	Attr	Reset Value	Description
31	RO	0x0	SGDMA Scatter/Gather DMA 1'b1: Dynamic configuration
30	RO	0x0	SGDMACon Scatter/Gather DMA configuration 1'b0: Non-Scatter/Gather DMA configuration 1'b1: Scatter/Gather DMA configuration
29:26	RO	0x0	INEps Number of Device Mode IN Endpoints Including Control Endpoint Range 0 -15 0:1 IN Endpoint 1:2 IN Endpoints .... 15:16 IN Endpoints

Bit	Attr	Reset Value	Description
25	RW	0x0	DedFifoMode Enable Dedicated Transmit FIFO for device IN Endpoints 1'b0: Dedicated Transmit FIFO Operation not enabled. 1'b1: Dedicated Transmit FIFO Operation enabled.
24	RW	0x0	SessEndFltr session_end Filter Enabled 1'b0: No filter 1'b1: Filter
23	RW	0x0	BValidFltr "b_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
22	RO	0x0	AValidFltr "a_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
21	RO	0x0	VBusValidFltr "vbus_valid" Filter Enabled 1'b0: No filter 1'b1: Filter
20	RO	0x0	IddgFltr "iddig" Filter Enable 1'b0: No filter 1'b1: Filter
19:16	RO	0x0	NumCtlEps Number of Device Mode Control Endpoints in Addition to Endpoin Range: 0-15
15:14	RO	0x0	PhyDataWidth UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+. 2'b00: 8 bits 2'b01: 16 bits 2'b10: 8/16 bits, software selectable Others: Reserved
13:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RO	0x0	EnHiber Enable Hibernation 1'b0: Hibernation feature not enabled 1'b1: Hibernation feature enabled
5	RO	0x0	AhbFreq Minimum AHB Frequency Less Than 60 MHz 1'b0: No 1'b1: Yes
4	RO	0x0	EnParPwrDown Enable Partial Power Down 1'b0: Partial Power Down Not Enabled 1'b1: Partial Power Down Enabled
3:0	RO	0x0	NumDevPerioEps Number of Device Mode Periodic IN Endpoints Range: 0-15

USBOTG_GLPMCFG
Address: Operational Base + offset (0x0054)
Core LPM Configuration Register

Bit	Attr	Reset Value	Description
31	RW	0x0	InvSelHsic HSIC-Invert Select HSIC The application uses this bit to control the otg core HSIC enable/disable. This bit overrides and functionally inverts the if_sel_hsic input port signal. If the core operates as non-HSIC-capable, it can only connect to non-HSIC-capable PHYs. If the core operates as HSIC-capable, it can only connect to HSICcapable PHYs. If the if_sel_hsic input signal is1: 1'b1: HSIC capability is not enabled. 1'b0: HSIC capability is enabled, If InvSelHsic = 1'b0: HSIC capability is enabled. If the if_sel_hsic input signal is 0: 1'b1: HSIC capability is enabled, 1'b0: HSIC capability is not enabled. This bit is writable only if an HSIC mode was specified for Mode of Operation (parameter OTG_ENABLE_HSIC). This bit is valid only if OTG_ENABLE_HSIC is enabled.

Bit	Attr	Reset Value	Description
30	RW	0x0	HSICCon HSIC-Connect The application must use this bit to initiate the HSIC Attach sequence. Host Mode: Once this bit is set, the host core configures to drive the HSIC Idle state (STROBE = 1 & DATA = 0) on the bus. It then waits for the device to initiate the Connect sequence. Device Mode: Once this bit is set, the device core waits for the HSIC Idle line state on the bus. Upon receiving the Idle line state, it initiates the HSIC Connect sequence. This bit is valid only if OTG_ENABLE_HSIC is 1, if_sel_hsic = 1 and InvSelHSIC is 0. Otherwise, it is read-only.
29:28	RO	0x0	reserved
27:25	RO	0x0	LPM_RetryCnt_Sts LPM Retry Count Status Number of LPM host retries remaining to be transmitted for the current LPM sequence.
24	RW	0x0	SndLPM Send LPM Transaction Host Mode: When the application software sets this bit, an LPM transaction containing two tokens, EXT and LPM, is sent. The hardware clears this bit once a valid response (STALL, NYET, or ACK) is received from the device or the core has finished transmitting the programmed number of LPM retries. Note: This bit must only be set when the host is connected to a local port.
23:21	R/WSC	0x0	LPM_Retry_Cnt LPM Retry Count When the device gives an ERROR response, this is the number of additional LPM retries that the host performs until a valid device response (STALL, NYET, or ACK) is received.

Bit	Attr	Reset Value	Description
20:17	RW	0x0	LPM_Chnl_Indx LPM Channel Index The channel number on which the LPM transaction must be applied while sending an LPM transaction to the local device. Based on the LPM channel index, the core automatically inserts the device address and endpoint number programmed in the corresponding channel into the LPM transaction.
16	RO	0x0	L1ResumeOK Sleep State Resume OK Indicates that the application or host can start a resume from the Sleep state. This bit is valid in the LPM Sleep (L1) state. It is set in Sleep mode after a delay of 50 us (TL1Residency). The bit is reset when SlpSts is 0 1'b1: The application/core can start resume from the Sleep state 1'b0: The application/core cannot start resume from the Sleep state

Bit	Attr	Reset Value	Description
15	RO	0x0	SlpSts Port Sleep Status Device Mode: This bit is set as long as a Sleep condition is present on the USB bus. The core enters the Sleep state when an ACK response is sent to an LPM transaction and the timer TL1TokenRetry. has expired. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the PHY Suspend input pin. The application must rely on SlpSts and not ACK in CoreL1Res to confirm transition into sleep. The core comes out of sleep: When there is any activity on the USB line_state When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig) or when the application resets or soft-disconnects the device. Host Mode: The host transitions to the Sleep (L1) state as a sideeffect of a successful LPM transaction by the core to the local port with an ACK response from the device. The read value of this bit reflects the port's current sleep status. The core clears this bit after: The core detects a remote L1 Wakeup signal The application sets the Port Reset bit or the Port L1Resume bit in the HPRT register or The application sets the L1Resume/ Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.L1WkUpInt or GINTSTS.DisconnInt, respectively). Values: 1'b0: Core not in L1 1'b1: Core in L1

Bit	Attr	Reset Value	Description
14:13	RO	0x0	CoreL1Res LPM Response Device Mode: The core's response to the received LPM transaction is reflected in these two bits. Host Mode: The handshake response received from the local device for LPM transaction. 11: ACK 10: NYET 01: STALL 00: ERROR (No handshake response)

Bit	Attr	Reset Value	Description
12:8	RW	0x00	HIRD_Thres HIRD Threshold Device Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when the HIRD value is greater than or equal to the value defined in this field (GLPMCFG.HIRD_Thres[3:0]), and HIRD_Thres[4] is set to 1'b1. Host Mode: The core asserts L1SuspendM to put the PHY into Deep Low-Power mode in L1 when HIRD_Thres[4] is set to 1'b1. HIRD_Thres[3:0] specifies the time for which resume signaling is to be reflected by the host TL1HubDrvResume2) on the USB when it detects device-initiated resume. HIRD_Thres must not be programmed with a value greater than 4'b1100 in Host mode, because this exceeds maximum TL1HubDrvResume2. Sl. No HIRD_Thres[3:0] Host mode resume time(us) 1 4'b0000 60 2 4'b0001 135 3 4'b0010 210 4 4'b0011 285 5 4'b0100 360 6 4'b0101 435 7 4'b0110 510 8 4'b0111 585 9 4'b1000 660 10 4'b1001 735 11 4'b1010 810 12 4'b1011 885 13 4'b1100 960 14 4'b1101 invalid

Bit	Attr	Reset Value	Description
7	RW	0x0	EnblSlpM Enable utmi_sleep_n For ULPI interface: The application uses this bit to write to the function control [7] in the L1 state, to enable the PHY to go into Low Power mode. For the host, this bit is valid only in Local Device mode. 1'b0: Writes to the ULPI Function Control Bit[7] are disabled. 1'b1: The core is enabled to write to the ULPI Function Control Bit[7], which enables the PHY to enter Low-Power mode. Note: When a ULPI interface is configured, enabling this bit results in a write to Bit 7 of the ULPI Function Control register. The ULPI PHY supports writing to this bit, and in the L1 state asserts SleepM when utmi_l1_suspend_n cannot be asserted. When a ULPI interface is configured, this bit must always be set if you are using the ULPI PHY. Note: For ULPI interfaces, do not clear this bit during the resume. For all other interfaces: The application uses this bit to control utmi_sleep_n assertion to the PHY in the L1 state. For the host, this bit is valid only in Local Device mode. 1'b0: utmi_sleep_n assertion from the core is not transferred to the external PHY. 1'b1: utmi_sleep_n assertion from the core is transferred to the external PHY when utmi_l1_suspend_n cannot be asserted.
6	RW	0x0	bRemoteWake RemoteWakeEnable Host Mode: The remote wakeup value to be sent in the LPM transaction's wIndex field. Device Mode: This field is updated with the received bRemoteWake LPM token's bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction.

Bit	Attr	Reset Value	Description
5:2	RW	0x0	HIRD Host-Initiated Resume Duration Host Mode: The value of HIRD to be sent in an LPM transaction. This value is also used to initiate resume for a duration TL1HubDrvResume1 for host initiated resume. Device Mode: This field is updated with the Received LPM Token HIRD bmAttribute when an ACK/NYET/STALL response is sent to an LPM transaction Sl. No HIRD[3:0] THIRD (us) 1 4'b0000 50 2 4'b0001 125 3 4'b0010 200 4 4'b0011 275 5 4'b0100 350 6 4'b0101 425 7 4'b0110 500 8 4'b0111 575 9 4'b1000 650 10 4'b1001 725 11 4'b1010 800 12 4'b1011 875 13 4'b1100 950 14 4'b1101 1025 15 4'b1110 1100 16 4'b1111 1175

Bit	Attr	Reset Value	Description
1	RW	0x0	AppL1Res LPM response programmed by application Handshake response to LPM token pre-programmed by device application software. The response depends on GLPMCFG.LPMCap. If GLPMCFG.LPMCap is 1'b0, the core always responds with a NYET. If GLPMCFG.LPMCap is 1'b1, the core responds as follows: 1: ACK. Even though an ACK is pre-programmed, the core responds with an ACK only on a successful LPM transaction. The LPM transaction is successful if: There are no PID/CRC5 errors in both the EXT token and the LPM token (else ERROR); A valid bLinkState = 0001B (L1) is received in the LPM transaction (else STALL); No data is pending in the Transmit queue (else NYET) 0: NYET. The pre-programmed software bit is overridden for response to LPM token when:(1)The received bLinkState is not L1 (STALL response); (2)An error is detected in either of the LPM token packets due to corruption (ERROR response).
0	RW	0x0	LPMCap LPM-Capable The application uses this bit to control the otg core LPM capabilities. If the core operates as a non-LPM-capable host, it cannot request the connected device/hub to activate LPM mode. If the core operates as a non-LPM-capable device, it cannot respond to any LPM transactions. 1'b0: LPM capability is not enabled. 1'b1: LPM capability is enabled. This bit is writable only if an LPM mode was specified for Mode of Operation (parameter OTG_ENABLE_LPM). Otherwise, reads return 0.

USBOTG_GPWRDN
Address: Operational Base + offset (0x0058)
Global Power Down Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:24	RO	0x00	MultValIdBC Multi Valued ID pin Battery Charger ACA inputs in the following order: Bit 26 - rid_float. Bit 25 - rid_gnd Bit 24 - rid_a Bit 23 - rid_b Bit 22 - rid_c These bits are present only if OTG_BC_SUPPORT = 1. Otherwise, these bits are reserved and will read 5'h0.
23	W1C	0x0	ADPInt ADP Interupt This bit is set whenever there is a ADP event.
22	RO	0x0	BSessVld B Session Valid This field reflects the B session valid status signal from the PHY. 1'b0: B-Valid is 0. 1'b1: B-Valid is 1. This bit is valid only when GPWRDN.PMUActv is 1.
21	RO	0x0	IDDIG This bit indicates the status of the signal IDDIG. The application must read this bit after receiving GPWRDN.StsChngInt and decode based on the previous value stored by the application. Indicates the current mode. 1'b1: Device mode 1'b0: Host mode This bit is valid only when GPWRDN.PMUActv is 1.

Bit	Attr	Reset Value	Description
20:19	RO	0x0	LineState This field indicates the current linestate on USB as seen by the PMU module. 2'b00: DM = 0, DP = 0. 2'b01: DM = 0, DP = 1. 2'b10: DM = 1, DP = 0. 2'b11: Not-defined. This bit is valid only when GPWRDN.PMUActv is 1.
18	RW	0x0	StsChngIntMsk Mask For StsChng Interrupt
17	W1C	0x0	StsChngInt This field indicates a status change in either the IDDIG or BSessVld signal. 1'b0: No Status change 1'b1: status change detected After receiving this interrupt the application should read the GPWRDN register and interpret the change in IDDIG or BSesVld with respect to the previous value stored by the application.
16	RW	0x0	SRPDetectMsk Mask For SRPDetect Interrupt
15	W1C	0x0	SRPDetect This field indicates that SRP has been detected by the PMU.This field generates an interrupt. After detecting SRP during hibernation the application should not restore the core. The application should get into the initialization process. 1'b0: SRP not detected 1'b1: SRP detected
14	RW	0x0	ConnDetMsk Mask for ConnectDet interrupt This bit is valid only when OTG_EN_PWROPT = 2.
13	W1C	0x0	ConnectDet This field indicates that a new connect has been detected 1'b0: Connect not detected 1'b1: Connect detected This bit is valid only when OTG_EN_PWROPT = 2.

Bit	Attr	Reset Value	Description
12	RW	0x0	DisconnectDetectMsk Mask For DisconnectDetect Interrupt This bit is valid only when OTG_EN_PWROPT = 2.
11	W1C	0x0	DisconnectDetect This field indicates that Disconnect has been detected by the PMU. This field generates an interrupt. After detecting disconnect during hibernation the application must not restore the core, but instead start the initialization process. 1'b0: Disconnect not detected 1'b1: Disconnect detected This bit is valid only when OTG_EN_PWROPT = 2.
10	RW	0x0	ResetDetMsk Mask For ResetDetected interrupt.This bit is valid only when OTG_EN_PWROPT = 2.
9	W1C	0x0	ResetDetected This field indicates that Reset has been detected by the PMU module. This field generates an interrupt. 1'b0: Reset Not Detected 1'b1: Reset Detected This bit is valid only when OTG_EN_PWROPT = 2.
8	RW	0x0	LineStageChangeMsk Mask For LineStateChange interrupt This bit is valid only when OTG_EN_PWROPT = 2.
7	W1C	0x0	LnStsChng Line State Change This interrupt is asserted when there is a Linestate Change detected by the PMU. The application should read GPWRDN.Linestate to determine the current linestate on USB. 1'b0: No LineState change on USB 1'b1: LineState change on USB This bit is valid only when GPWRDN.PMUActv is 1.This bit is valid only when OTG_EN_PWROPT = 2.

Bit	Attr	Reset Value	Description
6	RW	0x0	DisableVBUS The application should program this bit if HPRT0.PrtPwr was programmed to 0 before entering Hibernation. This is to indicate PMU whether session was ended before entering Hibernation. 1'b0: HPRT0.PrtPwr was not programed to 0. 1'b1: HPRT0.PrtPwr was programmed to 0.
5	RW	0x0	PwrDnSwtch Power Down Switch This bit indicates to the OTG core VDD switch is in ON/OFF state 1'b0: OTG is in ON state 1'b1: OTG is in OFF state Note: This bit must not be written to during normal mode of operation.
4	RW	0x0	PwrDnRst_n Power Down ResetN The application must program this bit to reset the OTG core during the Hibernation exit process or during ADP when powering up the core (in case the OTG core was powered off during ADP process). 1'b1: otg is in normal operation 1'b0: reset otg Note: This bit must not be written to during normal mode of operation.
3	RW	0x0	PwrDnClmp Power Down Clamp The application must program this bit to enable or disable the clamps to all the outputs of the OTG core module to prevent the corruption of other active logic. 1'b0: Disable PMU power clamp 1'b1: Enable PMU power clamp

Bit	Attr	Reset Value	Description
2	RW	0x0	Restore The application should program this bit to enable or disable restore mode from the PMU module. 1'b0: OTG in normal mode of operation 1'b1: OTG in restore mode Note: This bit must not be written to during normal mode of operation. This bit is valid only when OTG_EN_PWROPT = 2.
1	RW	0x0	PMUActv PMU Active This is bit is to enable or disable the PMU logic. 1'b0: Disable PMU module 1'b1: Enable PMU module Note: This bit must not be written to during normal mode of operation.
0	RW	0x0	PMUIntSel PMU Interrupt Select When the hibernation functionality is selected using the configuration option OTG_EN_PWR_OPT = 2, a write to this bit with 1'b1 enables the PMU to generate interrupts to the application. During this state all interrupts from the core module are blocked to the application. Note: This bit must be set to 1'b1 before the core is put into hibernation 1'b0: Internal otg_core interrupt is selected 1'b1: the external otg_pmu interrupt is selected Note: This bit must not be written to during normal mode of operation.

USBOTG_GDFIFOCFG

Address: Operational Base + offset (0x005c)

Global DFIFO Software Config Register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	EPInfoBaseAddr This field provides the start address of the EP info controller.

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	GDFIFOCfg This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non zero value to this register.The core does not have any corrective logic if the FIFO sizes are programmed incorrectly.

USBOTG_GADPCTL
Address: Operational Base + offset (0x0060)
ADP Timer,Control and Status Register

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:27	R/WSC	0x0	AR Access Request 2'b00 Read/Write Valid (updated by the core) 2'b01 Read 2'b10 Write 2'b11 Reserved
26	RW	0x0	AdpTmoutMsk ADP Timeout Interrupt Mask When this bit is set, it unmasks the interrupt because of AdpTmoutInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).
25	RW	0x0	AdpSnsIntMsk ADP Sense Interrupt Mask When this bit is set, it unmasks the interrupt due to AdpSnsInt. This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).
24	RW	0x0	AdpPrbIntMsk ADP Probe Interrupt Mask When this bit is set, it unmasks the interrupt due to AdpPrbInt.This bit is valid only if OTG_Ver = 1'b1(GOTGCTL[20]).

Bit	Attr	Reset Value	Description
23	W1C	0x0	AdpTmoutInt ADP Timeout Interrupt This bit is relevant only for an ADP probe. When this bit is set, it means that the ramp time has completed (GADPCTL.RTIM has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle. This bit is valid only if OTG_Ver = 1'b1.
22	W1C	0x0	AdpSnsInt ADP Sense Interrupt When this bit is set, it means that the VBUS voltage is greater than VadpSns value or VadpSns is reached. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
21	W1C	0x0	AdpPrbInt ADP Probe Interrupt When this bit is set, it means that the VBUS voltage is greater than VadpPrb or VadpPrb is reached. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
20	RW	0x0	ADPEn ADP Enable When set, the core performs either ADP probing or sensing based on EnaPrb or EnaSns. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
19	R/WSC	0x0	ADPRes ADP Reset When set, ADP controller is reset. This bit is auto-cleared after the reset procedure is complete in ADP controller.This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
18	RW	0x0	EnaSns Enable Sense When programmed to 1'b1, the core performs a sense operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).

Bit	Attr	Reset Value	Description
17	RW	0x0	EnaPrb Enable Probe When programmed to 1'b1, the core performs a probe operation. This bit is valid only if OTG_Ver = 1'b1 (GOTGCTL[20]).
16:6	RO	0x000	RTIM RAMP TIME These bits capture the latest time it took for VBUS to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 kHz clock cycles as follows: 0x000 - 1 cycles 0x001 - 2 cycles 0x002 - 3 cycles and so on till 0x7FF - 2048 cycles A time of 1024 cycles at 32 kHz corresponds to a time of 32 msec. (Note for scaledown ramp_timeout = prb_delta == 2'b00 => 200 cycles prb_delta == 2'b01 => 100 cycles prb_delta == 2'b01 => 50 cycles prb_delta == 2'b01 => 25 cycles.)
5:4	RW	0x0	PrbPer Probe Period These bits sets the TadpPrd as follows: 2'b00 - 0.625 to 0.925 sec (typical 0.775 sec) 2'b01 - 1.25 to 1.85 sec (typical 1.55 sec) 2'b10 - 1.9 to 2.6 sec (typical 2.275 sec) 2'b11 - Reserved (PRB_PER is also scaledown prb_per== 2'b00 => 400 ADP clocks prb_per== 2'b01 => 600 ADP clocks prb_per== 2'b10 => 800 ADP clocks prb_per==2'b11 => 1000 ADP clocks)

Bit	Attr	Reset Value	Description
3:2	RW	0x0	PrbDelta Probe Delta These bits set the resolution for RTIM value. The bits are defined in units of 32 kHz clock cycles as follows: 2'b00 - 1 cycles 2'b01 - 2 cycles 2'b10 - 3 cycles 2'b11 - 4 cycles For example if this value is chosen to 2'b01, it means that RTIM increments for every three 32Khz clock cycles.
1:0	RW	0x0	PrbDschg Probe Discharge These bits set the times for TadpDschg. These bits are defined as follows: 2'b00 4 msec (Scaledown 2 32Khz clock cycles) 2'b01 8 msec (Scaledown 4 32Khz clock cycles) 2'b10 16 msec (Scaledown 8 32Khz clock cycles) 2'b11 32 msec (Scaledown 16 32Khz clock cycles)

USBOTG_HPTXFSIZ

Address: Operational Base + offset (0x0100)

Host Periodic Transmit FIFO Size Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	PTxFSize Host Periodic TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest Host Mode Periodic Tx Data FIFO Depth (parameter OTG_TX_HPERIO_DFIFO_DEPTH). If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads returnthe power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value .
15:0	RW	0x0000	PTxFStAddr Host Periodic TxFIFO Start Address The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth specified. These parameters are: In shared FIFO operation: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH. In dedicated FIFO mode: OTG_RX_DFIFO_DEPTH + OTG_TX_HNPERIO_DFIFO_DEPTH. If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field. Programmed values must not exceed the power-on value.

USBOTG_DIEPTXFn

Address: Operational Base + offset (0x0104 + 0x4 * n) 0≤n≤14

Device Periodic Transmit FIFO-n Size Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	INEP1TxFDep IN Endpoint TxFIFO Depth This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 32,768 The power-on reset value of this register is specified as the Largest IN Endpoint FIFO number Depth (parameter OTG_TX_DINEP_DFIFO_DEPTH_n)(0 < n <= 15). If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), you can write a new value in this field.Programmed values must not exceed the power-on value .
15:0	RW	0x0000	INEP1TxFStAddr IN Endpoint FIFO1 Transmit RAM Start Address This field contains the memory start address for IN endpoint Transmit FIFOn (0 < n <= 15). The power-on reset value of this register is specified as the Largest Rx Data FIFO Depth (parameter OTG_RX_DFIFO_DEPTH). OTG_RX_DFIFO_DEPTH + SUM 0 to n-1 (OTG_DINEP_TXFIFO_DEPTH_n) For example start address of IN endpoint FIFO 1 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0. The start address of IN endpoint FIFO 2 is OTG_RX_DFIFO_DEPTH + OTG_DINEP_TXFIFO_DEPTH_0 + OTG_DINEP_TXFIFO_DEPTH_1. If Enable Dynamic FIFO Sizing? was deselected (parameter OTG_DFIFO_DYNAMIC = 0), these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected (parameter OTG_DFIFO_DYNAMIC = 1), and you have programmed a new value for RxFIFO depth, you can write that value in this field. Programmed values must not exceed the power-on value set .

USBOTG_HCFG
Address: Operational Base + offset (0x0400)
Host Configuration Register

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	PerSchedEna Enable Periodic Scheduling Applicable in Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is reset. The core will not process any periodic channels. As soon as this bit is set, the core will get ready to start scheduling periodic channels and sets HCFG.PerSchedStat. The setting of HCFG.PerSchedStat indicates the core has enabled periodic scheduling. Once HCFG.PerSchedEna is set, the application is not supposed to again reset the bit unless HCFG.PerSchedStat is set. As soon as this bit is reset, the core will get ready to stop scheduling periodic channels and resets HCFG.PerSchedStat. In non Scatter/Gather DMA mode, this bit is reserved.
25:24	RW	0x0	FrListEn Frame List Entries The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode.

Bit	Attr	Reset Value	Description
23	RW	0x0	DescDMA Enable Scatter/gather DMA in Host mode When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: GAHBCFG.DMAEn=0, HCFG.DescDMA=0 => Slave mode GAHBCFG.DMAEn=0, HCFG.DescDMA=1 => Invalid GAHBCFG.DMAEn=1, HCFG.DescDMA=0 => Buffered DMA mode GAHBCFG.DMAEn=1, HCFG.DescDMA=1 => Scatter/Gather DMA mode In non Scatter/Gather DMA mode, this bit is reserved.
22:16	RO	0x0	reserved
15:8	RW	0x00	ResValid Resume Validation Period This field is effective only when HCFG.Ena32KHzS is set. It controls the resume period when the core resumes from suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.
7	RW	0x0	Ena32KHzS Enable 32-KHz Suspend Mode This bit can only be set if the USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero. When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend.
6:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	RW	0x0	FSLSSupp FS- and LS-Only Support The application uses this bit to control the core enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device 1'b1: FS/LS-only, even if the connected device can support HS
1:0	RW	0x0	FSLSPclkSel FS/LS PHY Clock Select 2'b00: PHY clock is running at 30/60 MHz 2'b01: PHY clock is running at 48 MHz Others: Reserved

USBOTG_HFIR

Address: Operational Base + offset (0x0404)

Host Frame Interval Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	FrInt Frame Interval The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. 125 us * (PHY clock frequency for HS) 1 ms * (PHY clock frequency for FS/LS)

USBOTG_HFNUM

Address: Operational Base + offset (0x0408)

Host Frame Number/Frame Time Remaining Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	FrRem Frame Time Remaining Indicates the amount of time remaining in the current microframe (HS) or frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.
15:0	RO	0xffff	FrNum Frame Number This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features? was not selected (OTG_RM_OTG_FEATURES = 0). Otherwise, reads return the frame number value.

USBOTG_HPTXSTS

Address: Operational Base + offset (0x0410)

Host Periodic Transmit FIFO/Queue Status Register

Bit	Attr	Reset Value	Description
31:24	RO	0x00	PTxQTop Top of the Periodic Transmit Request Queue This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. Bit [31]: Odd/Even (micro)frame 1'b0: send in even (micro)frame 1'b1: send in odd (micro)frame Bits [30:27]: Channel/endpoint number Bits [26:25]: Type 2'b00: IN/OUT 2'b01: Zero-length packet 2'b10: CSPLIT 2'b11: Disable channel command Bit [24]: Terminate (last entry for the selected channel/endpoint)
23:16	RO	0x00	PTxQSpcAvail Periodic Transmit Request Queue Space Available Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. 8'h0: Periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available n: n locations available (0 <=n <= 16) Others: Reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	PTxFSpcAvail Periodic Transmit Data FIFO Space Available Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words . 16'h0: Periodic TxFIFO is full . 16'h1: 1 word available . 16'h2: 2 words available . 16'hn: n words available (where 0 . n . 32,768) . 16'h8000: 32,768 words available . Others: Reserved

USBOTG_HAINT
Address: Operational Base + offset (0x0414)
Host All Channels Interrupt Reigster

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RO	0x0000	HAINT Channel Interrupts One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15

USBOTG_HAINTMSK
Address: Operational Base + offset (0x0418)
Host All Channels Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	HAINTMsk Channel Interrupt Mask One bit per channel: Bit 0 for channel 0, bit 15 for channel 15

USBOTG_HPRT
Address: Operational Base + offset (0x0440)
Host Port Control and Status Register

Bit	Attr	Reset Value	Description
31:19	RO	0x0	reserved

Bit	Attr	Reset Value	Description
18:17	RO	0x0	PrtSpd Port Speed Indicates the speed of the device attached to this port. 2'b00: High speed 2'b01: Full speed 2'b10: Low speed 2'b11: Reserved
16:13	RW	0x0	PrtTstCtl Port Test Control The application writes a nonzero value to this field to put the port into a Test mode,and the corresponding pattern is signaled on the port. 4'b0000: Test mode disabled 4'b0001: Test_J mode 4'b0010: Test_K mode 4'b0011: Test_SE0_NAK mode 4'b0100: Test_Packet mode 4'b0101: Test_Force_Enable Others: Reserved
12	R/WSC	0x0	PrtPwr Port Power The application uses this field to control power to this port (write 1'b1 to set to 1'b1and write 1'b0 to set to 1'b0), and the core can clear this bit on an over current condition. 1'b0: Power off 1'b1: Power on
11:10	RO	0x0	PrtLnSts Port Line Status Indicates the current logic level USB data lines Bit [10]: Logic level of D+ Bit [11]: Logic level of D
9	RO	0x0	reserved

Bit	Attr	Reset Value	Description
8	RW	0x0	PrtRst Port Reset When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. 1'b0: Port not in reset 1'b1: Port in reset To start a reset on the port, the application must leave this bit set for at least the minimum duration mentioned below, as specified in the USB 2.0 specification, Section 7.1.7.5. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard. High speed: 50 ms Full speed/Low speed: 10 ms
7	R/WSC	0x0	PrtSusp Port Suspend The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively). 1'b0: Port not in Suspend mode 1'b1: Port in Suspend mode

Bit	Attr	Reset Value	Description
6	R/WSC	0x0	PrtRes Port Resume The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. 1'b0: No resume driven 1'b1: Resume driven When LPM is enabled and the core is in the L1 (Sleep) state, setting this bit results in the following behavior: The core continues to drive the resume signal until a pre-determined time specified in the GLPMCFG.HIRD_Thres[3:0] field. If the core detects a USB remote wakeup sequence, as indicated by the Port L1 Resume/Remote L1 Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.L1WkUpInt), the core starts driving resume signaling without application intervention and clears this bit at the end of the resume. The read value of this bit indicates whether the core is currently driving resume signaling. 1'b0: No resume driven 1'b1: Resume driven
5	W1C	0x0	PrtOvrCurrChng Port Overcurrent Change The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.

Bit	Attr	Reset Value	Description
4	RO	0x0	PrtOvrCurrAct Port Overcurrent Active Indicates the overcurrent condition of the port. 1'b0: No overcurrent condition 1'b1: Overcurrent condition
3	W1C	0x0	PrtEnChng Port Enable/Disable Change The core sets this bit when the status of the Port Enable bit [2] of this register changes.
2	W1C	0x0	PrtEna Port Enable A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit.The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application. 1'b0: Port disabled 1'b1: Port enabled
1	W1C	0x0	PrtConnDet Port Connect Detected The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). The application must write a 1 to this bit to clear the interrupt.
0	RO	0x0	PrtConnSts Port Connect Status 0: No device is attached to the port. 1: A device is attached to the port.

USBOTG_HCCHARn

Address: Operational Base + offset (0x0500 + 0x20 ∗ n) 0≤n≤15

Host Channel-n Characteristics Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	ChEna Channel Enable When Scatter/Gather mode is enabled 1'b0: Indicates that the descriptor structure is not yet ready. 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled, This field is set by the application and cleared by the OTG host. 1'b0: Channel disabled 1'b1: Channel enabled
30	R/WSC	0x0	ChDis Channel Disable The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.
29	RW	0x0	OddFrm Odd Frame This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.
28:22	RW	0x00	DevAddr Device Address This field selects the specific device serving as the data source or sink.

Bit	Attr	Reset Value	Description
21:20	RW	0x0	MC_EC Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes arbitration. 2'b00: Reserved This field yields undefined results. 2'b01: 1 transaction 2'b10: 2 transactions to be issued for this endpoint per microframe 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.
19:18	RW	0x0	EPType Endpoint Type Indicates the transfer type selected. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt
17	RW	0x0	LSpdDev Low-Speed Device This field is set by the application to indicate that this channel is communicating to a low-speed device.
16	RO	0x0	reserved
15	RW	0x0	EPDir Endpoint Direction Indicates whether the transaction is IN or OUT. 1'b0: OUT 1'b1: IN

Bit	Attr	Reset Value	Description
14:11	RW	0x0	EPNum Endpoint Number Indicates the endpoint number on the device serving as the data source or sink.
10:0	RW	0x000	MPS Maximum Packet Size Indicates the maximum packet size of the associated endpoint.

USBOTG_HCSPLTn

Address: Operational Base + offset (0x0504 + 0x20 * n) 0≤n≤15

Host Channel-n Split Control Register

Bit	Attr	Reset Value	Description
31	RW	0x0	SpltEna Split Enable The application sets this field to indicate that this channel is enabled to perform split transactions.
30:17	RO	0x0	reserved
16	RW	0x0	CompSplt Do Complete Split The application sets this field to request the OTG host to perform a complete split transaction.
15:14	RW	0x0	XactPos Transaction Position This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188bytes). 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).
13:7	RW	0x00	HubAddr Hub Address This field holds the device address of the transaction translator's hub.

Bit	Attr	Reset Value	Description
6:1	RO	0x0	reserved
0	RW	0x0	PrtAddr Port Address This field is the port number of the recipient transaction translator.

USBOTG_HCINTn

Address: Operational Base + offset (0x0508 + 0x20 * n) 0≤n≤15

Host Channel-n Interrupt Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	W1C	0x0	DESC_LST_ROLLIntr Descriptor rollover interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when the corresponding channel's descriptor list rolls over. For non Scatter/Gather DMA mode, this bit is reserved.
12	W1C	0x0	XCS_XACT_ERR Excessive Transaction Error This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when 3 consecutive transaction errors occurred on the USB bus. XCS_XACT_ERR will not be generated for Isochronous channels. For non Scatter/Gather DMA mode, this bit is reserved.
11	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process. BNA will not be generated for Isochronous channels. For non Scatter/Gather DMA mode, this bit is reserved.
10	W1C	0x0	DataTglErr Data Toggle Error In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
9	W1C	0x0	FrmOvrun Frame Overrun In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core

Bit	Attr	Reset Value	Description
8	W1C	0x0	BblErr Babble Error In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
7	W1C	0x0	XactErr Transaction Error Indicates one of the following errors occurred on the USB: CRC check failure, Timeout, Bit stuff error, False EOP. In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
6	WO	0x0	NYET NYET Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
5	W1C	0x0	ACK ACK Response Received/Transmitted Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
4	W1C	0x0	NAK NAK Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
3	W1C	0x0	STALL STALL Response Received Interrupt In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.
2	W1C	0x0	AHBErr AHB Error This is generated only in DMA mode when there is an AHB error during AHB read/write.The application can read the corresponding channel's DMA address register to get the error address.

Bit	Attr	Reset Value	Description
1	W1C	0x0	ChHltd Channel Halted In non Scatter/Gather DMA mode, it indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application or because of a completed transfer.In Scatter/Gather DMA mode, this indicates that transfer completed due to any of the following: EOL being set in descriptor, AHB error, Excessive transaction errors, In response to disable request by the application,Babble, Stall, Buffer Not Available (BNA)
0	W1C	0x0	XferCompl Transfer Completed For Scatter/Gather DMA mode, it indicates that current descriptor processing got completed with IOC bit set in its descriptor. In non Scatter/Gather DMA mode, it indicates that Transfer completed normally without any errors.

USBOTG_HCINTMSKn

Address: Operational Base + offset (0x050c + 0x20 * n) 0≤n≤15

Host Channel-n Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	DESC_LST_ROLLIntrMsk Descriptor rollover interrupt Mask register This bit is valid only when Scatter/Gather DMA mode is enabled. In non Scatter/Gather DMA mode, this bit is reserved.
12	RO	0x0	reserved
11	RW	0x0	BNAIntrMsk BNA (Buffer Not Available) Interrupt mask register This bit is valid only when Scatter/Gather DMA mode is enabled. In non Scatter/Gather DMA mode, this bit is reserved.
10	RW	0x0	DataTglErrMsk Data Toggle Error Mask This bit is not applicable in Scatter/Gather DMA mode.

Bit	Attr	Reset Value	Description
9	RW	0x0	FrmOvrunMsk Frame Overrun Mask This bit is not applicable in Scatter/Gather DMA mode.
8	RW	0x0	BblErrMsk Babble Error Mask This bit is not applicable in Scatter/Gather DMA mode.
7	RW	0x0	XactErrMsk Transaction Error Mask This bit is not applicable in Scatter/Gather DMA mode
6	RW	0x0	NyetMsk NYET Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
5	RW	0x0	AckMsk ACK Response Received/Transmitted Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
4	RW	0x0	NakMsk NAK Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
3	RW	0x0	StallMsk STALL Response Received Interrupt Mask This bit is not applicable in Scatter/Gather DMA mode.
2	RW	0x0	AHBErrMsk AHB Error Mask Note: This bit is only accessible when OTG_ARCHITECTURE = 2
1	RW	0x0	ChHltdMsk Channel Halted Mask
0	RW	0x0	XferComplMsk Transfer Completed Mask This bit is valid only when Scatter/Gather DMA mode is enabled. In non Scatter/Gather DMA mode, this bit is reserved.

USBOTG_HCTSIZn

Address: Operational Base + offset (0x0510 + 0x20 * n) 0≤n≤15
Host Channel-n Transfer Size Register(In Scatter/Gather DMA mode)

Bit	Attr	Reset Value	Description
31	RW	0x0	DoPng Do Ping This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol. Note: Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the channel.
30:29	RW	0x0	Pid PID The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA (non-control)/SETUP (control)
28:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:8	RW	0x000	NTD Number of Transfer Descriptors (Non Isochronous) This value is in terms of number of descriptors. Maximum number of descriptor that can be present in the list is 64. The values can be from 0 to 63. • 0 – 1 descriptor. • 63 – 64 descriptors This field indicates the total number of descriptors present in that list. The core will wrap around after servicing NTD number of descriptors for that list. (Isochronous) This field indicates the number of descriptors present in that list.μframe. The possible values for FS are • 1 – 2 descriptors • 3 – 4 descriptors • 7 – 8 descriptors • 15 – 16 descriptors • 31 – 32 descriptors • 63 – 64 descriptors The possible values for HS are • 7 – 8 descriptors • 15 – 16 descriptors • 31 – 32 descriptors • 63 – 64 descriptors • 127 – 128 descriptors • 255 – 256 descriptors

Bit	Attr	Reset Value	Description
18:0	RW	0x00000	SCHED_INFO Schedule information Every bit in this 8 bit register indicates scheduling for that microframe. Bit 0 indicates scheduling for 1st microframe and bit 7 indicates scheduling for 8th microframe in that frame. A value of 8'b11111111 indicates that the corresponding interrupt channel is scheduled to issue a token every microframe in that frame. A value of 8'b10101010 indicates that the corresponding interrupt channel is scheduled to issue a token every alternate microframe starting with second microframe. Note that this field is applicable only for periodic (Isochronous and Interrupt) channels.

Host Channel-n Transfer Size Register(In Non Scatter/Gather DMA mode)

Bit	Attr	Reset Value	Description
31	RW	0x0	DoPng Do Ping This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol. Note: Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the channel.
30:29	RW	0x0	Pid PID The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA (non-control)/SETUP (control)

Bit	Attr	Reset Value	Description
28:19	RW	0x000	PktCnt Packet Count This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN).The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion.The width of this counter is specified as Width of Packet Counters (parameter OTG_PACKET_COUNT_WIDTH).
18:0	RW	0x00000	XferSize Transfer Size For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic). The width of this counter is specified as Width of Transfer Size Counters (parameter OTG_TRANS_COUNT_WIDTH).

USBOTG_HCDMAn

Address: Operational Base + offset (0x0514 + 0x20 * n) 0≤n≤15

Host Channel-n DMA Address Register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction.

USBOTG_HCDMABn

Address: Operational Base + offset (0x051c + 0x20 * n) 0≤n≤15

Host Channel-n DMA Buffer Address Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	HCDMABn Holds the current buffer address This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

Host Channel-n DMA Decriptor Address Register

Bit	Attr	Reset Value	Description
31:N (Isoc) 31:9 (Non Isoc)	RW	0x00000000	(DMAAddr) DMA Address Non-Isochronous: This field holds the start address of the 512 bytes page. The first descriptor in the list should be located in this address. The first descriptor may be or may not be ready. The core starts processing the list from the CTD value. Isochronous: This field holds the address of the 2*(nTD+1) bytes of locations in which the isochronous descriptors are present where N is based on nTD as per Table below

31:N	N-1:3	2:0
Base Address	Offset	000

HS ISOC

nTD	N
7	6
15	7
31	8
63	9
127	10
255	11

FS ISOC

nTD	N
1	4
3	5
7	6
15	7
31	8
63	9

Bit	Attr	Reset Value	Description
N-1:3 (Isoc) 8:3 (Non Isoc)	RW	0x00000000	CTD Current Transfer Desc Non Isochronous: This value is in terms of number of descriptors. The values can be from 0 to 63. 0 – 1 descriptor. 63- 64 descriptors. This field indicates the current descriptor processed in the list. This field is updated both by application and the core. For example, if the application enables the channel after programming CTD=5, then the core will start processing the 6th descriptor. The address is obtained by adding a value of (8bytes*5=) 40(decimal) to DMAAddr. Isochronous: CTD for isochronous is based on the current frame/μframe value. Need to be set to zero by application.
2:0	RO	0x0	reserved

USBOTG_DCFG

Address: Operational Base + offset (0x0800)

Device Configuration Register

Bit	Attr	Reset Value	Description
31:26	RW	0x02	ResValid Resume Validation Period This field controls the period when the core resumes from a suspend. When this bit is set, the core counts for the ResValid number of clock cycles to detect a valid resume. This field is effective only when DCFG.Ena32KHzSusp is set.

Bit	Attr	Reset Value	Description
25:24	RW	0x0	PerSchIntvl Periodic Scheduling Interval PerSchIntvl must be programmed only for Scatter/Gather DMA mode. Description: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. When no periodic endpoints are active, then the internal DMA engine services nonperiodic endpoints, ignoring this field. After the specified time within a (micro)frame, the DMA switches to fetching for nonperiodic endpoints. 2'b00: 25% of (micro)frame. 2'b01: 50% of (micro)frame. 2'b10: 75% of (micro)frame. 2'b11: Reserved.
23	RW	0x0	DescDMA Enable Scatter/Gather DMA in Device mode When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffered DMA mode GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode

Bit	Attr	Reset Value	Description
22:18	RW	0x08	EPMisCnt IN Endpoint Mismatch Count This field is valid only in shared FIFO operation.The application programs this filed with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.
17:13	RO	0x0	reserved
12:11	RW	0x0	PerFrInt Periodic Frame Interval Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete. 2'b00: 80% of the (micro)frame interval 2'b01: 85% 2'b10: 90% 2'b11: 95%
10:4	RW	0x00	DevAddr Device Address The application must program this field after every SetAddress control command.
3	RW	0x0	Ena32KHzS Enable 32-KHz Suspend Mode When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend. This bit can only be set if USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero.

Bit	Attr	Reset Value	Description
2	RW	0x0	NZStsOUTHShk Non-Zero-Length Status OUT Handshake The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. 1'b0: Send the received OUT packet to the application (zero-length or nonzerolength) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.
1:0	RW	0x0	DevSpd Device Speed Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b10: Reserved 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)

USBOTG_DCTL

Address: Operational Base + offset (0x0804)

Device Control Register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	RW	0x0	NakOnBble Set NAK automatically on babble The core sets NAK automatically for the endpoint on which babble is received.

Bit	Attr	Reset Value	Description
15	RW	0x0	IgnrFrmNum Ignore frame number for isochronous endpoints in case of Scatte Do NOT program IgnrFrmNum bit to 1'b1 when the core is operating in Threshold mode. Note: When Scatter/Gather DMA mode is enabled this feature is not applicable to highspeed, high-bandwidth transfers. When this bit is enabled, there must be only one packet per descriptor. 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. 1: The core ignores the frame number, sending packets immediately as the packets are ready. Scatter/Gather: In Scatter/Gather DMA mode, when this bit is enabled, the packets are not flushed when an ISOC IN token is received for an elapsed frame. When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro)frames. 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame 1: Periodic transfer interrupt feature is enabled; the application can program transfers for multiple (micro)frames for periodic endpoints. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro)frames are completed.

Bit	Attr	Reset Value	Description
14:13	RW	0x1	GMC Global Multi Count GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for nonperiodic end points. 2'b00: Invalid. 2'b01: 1 packet. 2'b10: 2 packets. 2'b11: 3 packets. When Scatter/Gather DMA mode is disabled, this field is reserved. and reads 2'b00.
12	RO	0x0	reserved
11	RW	0x0	PWROnPrgDone Power-On Programming Done The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.
10	WO	0x0	CGOUTNak Clear Global OUT NAK A write to this field clears the Global OUT NAK.
9	WO	0x0	SGOUTNak Set Global OUT NAK A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.
8	WO	0x0	CGNPInNak Clear Global Non-periodic IN NAK A write to this field clears the Global Non-periodic IN NAK.

Bit	Attr	Reset Value	Description
7	WO	0x0	SGNPInNak Set Global Non-periodic IN NAK A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.
6:4	RW	0x0	TstCtl Test Control 3'b000: Test mode disabled 3'b001: Test_J mode 3'b010: Test_K mode 3'b011: Test_SE0_NAK mode 3'b100: Test_Packet mode 3'b101: Test_Force_Enable Others: Reserved
3	RO	0x0	GOUTNakSts Global OUT NAK Status 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped
2	RO	0x0	GNPINNakSts Global Non-periodic IN NAK Status 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.

Bit	Attr	Reset Value	Description
1	RW	0x0	SftDiscon Soft Disconnect The application uses this bit to signal the OTG core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.
0	RW	0x0	RmtWkUpSig Remote Wakeup Signaling When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1蝻ms after setting it.If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 us (TL1DevDrvResume) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero.

USBOTG_DSTS
Address: Operational Base + offset (0x0808)
Device Status Register

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:8	RW	0x0000	SOFFN Frame or Microframe Number of the Received SOF When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains a frame number.
7:4	RO	0x0	reserved
3	RW	0x0	ErrticErr Erratic Error The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the OTG core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.
2:1	RW	0x0	EnumSpd Enumerated Speed Indicates the speed at which the OTG core has come up after speed detection through a chirp sequence. 2'b00: High speed (PHY clock is running at 30 or 60 MHz) 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) 2'b10: Low speed (PHY clock is running at 48 MHz, internal phy_clk at 6 MHz) 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.

Bit	Attr	Reset Value	Description
0	RW	0x0	SuspSts Suspend Status In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time. The core comes out of the suspend: When there is any activity on the utmi_linestate signal, When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).

USBOTG_DIEPMSK

Address: Operational Base + offset (0x0810)

Device IN Endpoint common interrupt mask register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	NAKMsk NAK interrupt Mask
12:10	RO	0x0	reserved
9	RW	0x0	BNAInIntrMsk BNA Interrupt Mask
8	RW	0x0	TxfifoUndrnMsk Fifo Underrun Mask
7	RO	0x0	reserved
6	RW	0x0	INEPNakEffMsk IN Endpoint NAK Effective Mask
5	RW	0x0	INTknEPMisMsk IN Token received with EP Mismatch Mask
4	RW	0x0	INTknTXFEmpMsk IN Token Received When TxFIFO Empty Mask
3	RW	0x0	TimeOUTMsk Timeout Condition Mask
2	RW	0x0	AHBErrMsk AHB Error Mask
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DOEPMSK

Address: Operational Base + offset (0x0814)
Device OUT Endpoint common interrupt mask register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	NYETMsk NYET Interrupt Mask
13	RW	0x0	NAKMsk NAK Interrupt Mask
12	RW	0x0	BbleErrMsk Babble Interrupt Mask
11:10	RO	0x0	reserved
9	RW	0x0	BnaOutIntrMsk BNA interrupt Mask
8	RW	0x0	OutPktErrMsk OUT Packet Error Mask
7	RO	0x0	reserved
6	RW	0x0	Back2BackSETup Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	RO	0x0	reserved
4	RW	0x0	OUTTknEPdisMsk OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.
3	RW	0x0	SetUPMsk SETUP Phase Done Mask Applies to control endpoints only.
2	RW	0x0	AHBErrMsk AHB Error
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DAINT
Address: Operational Base + offset (0x0818)
Device All Endpoints interrupt register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	OutEPInt OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15

Bit	Attr	Reset Value	Description
15:0	RO	0x0000	InEpInt IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15

USBOTG_DAINTMSK
Address: Operational Base + offset (0x081c)
Device All Endpoint interrupt mask register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	OutEpMsk OUT EP Interrupt Mask Bits One per OUT Endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15
15:0	RW	0x0000	InEpMsk IN EP Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15

USBOTG_DTKNQR1
Address: Operational Base + offset (0x0820)
Device IN token sequence learning queue read register1

Bit	Attr	Reset Value	Description
31:8	RO	0x000000	EPTkn Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 5 Bits [27:24]: Endpoint number of Token 4 ..... Bits [15:12]: Endpoint number of Token 1 Bits [11:8]: Endpoint number of Token 0
7	RO	0x0	WrapBit Wrap Bit This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.
6:5	RO	0x0	reserved
4:0	RO	0x00	INTknWPtr IN Token Queue Write Pointer

USBOTG_DTKNQR2
Address: Operational Base + offset (0x0824)
Device IN token sequence learning queue read register2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPTkn Endpoint Token Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 13 Bits [27:24]: Endpoint number of Token 12 ..... Bits [7:4]: Endpoint number of Token 7 Bits [3:0]: Endpoint number of Token 6

USBOTG_DVBUSDIS
Address: Operational Base + offset (0x0828)
Device VBUS discharge time register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0b8f	DVBUSDis Device VBUS Discharge Time Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals:VBUS discharge time in PHY clocks / 1,024.The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width). Depending on your VBUS load, this value can need adjustment.

USBOTG_DVBUSPULSE
Address: Operational Base + offset (0x082c)
Device VBUS Pulsing Timer Register

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RW	0x000	DVBUSPulse Device VBUS Pulsing Time Specifies the VBUS pulsing time during SRP. This value equals:VBUS pulsing time in PHY clocks / 1,024.The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width).

USBOTG_DTHRCTL
Address: Operational Base + offset (0x0830)

Device Threshold Control Register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x1	ArbPrkEn Arbiter Parking Enable This bit controls internal DMA arbiter parking for IN endpoints. When thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default the parking is enabled.
26	RO	0x0	reserved
25:17	RW	0x008	RxThrLen Receive Threshold Length This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).
16	RW	0x0	RxThrEn Receive Threshold Enable When this bit is set, the core enables thresholding in the receive direction.
15:13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12:11	RW	0x0	AHBThrRatio AHB Threshold Ratio These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements. 2'b00: AHB threshold = MAC threshold 2'b01: AHB threshold = MAC threshold / 2 2'b10: AHB threshold = MAC threshold / 4 2'b11: AHB threshold = MAC threshold / 8
10:2	RW	0x008	TxThrLen Transmit Threshold Length This field specifies Transmit thresholding size in DWORDS. This field also forms the MAC threshold and specifies the amount of data, in bytes, to be in the corresponding endpoint transmit FIFO before the core can start a transmit on the USB. When the value of AHBThrRatio is 2'h00, the threshold length must be at least 8 DWORDS. If the AHBThrRatio is nonzero, the application must ensure that the AHB threshold value does not go below the recommended 8 DWORDs. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).

Bit	Attr	Reset Value	Description
1	RW	0x0	ISOThrEn ISO IN Endpoints Threshold Enable When this bit is set, the core enables thresholding for isochronous IN endpoints.
0	RW	0x0	NonISOThrEn Non-ISO IN Endpoints Threshold Enable When this bit is set, the core enables thresholding for Non Isochronous IN endpoints.

USBOTG_DIEPEMPMSK
Address: Operational Base + offset (0x0834)
Device IN endpoint FIFO empty interrupt mask register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	InEpTxfEmpMsk IN EP Tx FIFO Empty Interrupt Mask Bits These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DEACHINT
Address: Operational Base + offset (0x0838)
Device each endpoint interrupt register

Bit	Attr	Reset Value	Description
31:16	RO	0x0000	EchOutEPInt OUT Endpoint Interrupt Bits One bit per OUT endpoint: Bit 16 for OUT endpoint 0 ... Bit 31 for OUT endpoint 15
15:0	RO	0x0000	EchInEpInt IN Endpoint Interrupt Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DEACHINTMSK
Address: Operational Base + offset (0x083c)

Device each endpoint interrupt register mask

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	EchOutEpMsk OUT EP Interrupt Mask Bits One per OUT Endpoint: Bit 16 for IN endpoint 0 ... Bit 31 for endpoint 15
15:0	RW	0x0000	EchInEpMsk IN EP Interrupt Mask Bits One bit per IN Endpoint: Bit 0 for IN endpoint 0 ... Bit 15 for endpoint 15

USBOTG_DIEPEACHMSKn

Address: Operational Base + offset (0x0840 + 0x4 * n) 0≤n≤15

Device each IN endpoint -n interrupt Register

Bit	Attr	Reset Value	Description
31:14	RO	0x0	reserved
13	RW	0x0	NAKMsk NAK interrupt Mask
12:10	RO	0x0	reserved
9	RW	0x0	BNAInIntrMsk BNA interrupt Mask
8	RW	0x0	TxfifoUndrnMsk Fifo Under run Mask
7	RO	0x0	reserved
6	RW	0x0	INEPNakEffMsk IN Endpoint NAK Effective Mask
5	RW	0x0	INTknEPMisMsk IN Token received with EP Mismatch Mask
4	RW	0x0	INTknTXFEmpMsk IN Token Received When TxFIFO Empty Mask
3	RW	0x0	TimeOUTMsk Timeout Condition Mask(Non-isochronous endpoints)
2	RW	0x0	AHBErrMsk AHB Error Mask
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DOEPEACHMSKn

Address: Operational Base + offset (0x0880 + 0x4 * n) 0≤n≤15

Device each out endpoint-n interrupt register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	NYETMsk NYET interrupt Mask
13	RW	0x0	NAKMsk NAK interrupt Mask
12	RW	0x0	BbleErrMsk Babble interrupt Mask
11:10	RO	0x0	reserved
9	RW	0x0	BnaOutIntrMsk BNA interrupt Mask
8	RW	0x0	OutPktErrMsk OUT Packet Error Mask
7	RO	0x0	reserved
6	RW	0x0	Back2BackSETup Back-to-Back SETUP Packets Received Mask Applies to control OUT endpoints only.
5	RO	0x0	reserved
4	RW	0x0	OUTTknEPdisMsk OUT Token Received when Endpoint Disabled Mask Applies to control OUT endpoints only.
3	RW	0x0	SetUPMsk SETUP Phase Done Mask Applies to control endpoints only.
2	RW	0x0	AHBErrMsk AHB Error
1	RW	0x0	EPDisbldMsk Endpoint Disabled Interrupt Mask
0	RW	0x0	XferComplMsk Transfer Completed Interrupt Mask

USBOTG_DIEPCTL0

Address: Operational Base + offset (0x0900)

Device control IN endpoint 0 control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	EPEna Endpoint Enable When Scatter/Gather DMA mode is enabled, for IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. When Scatter/Gather DMA mode is disabled-such as in buffer-pointer based DMA mode-this bit indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting the following interrupts on this endpoint: Endpoint Disabled; Transfer Completed.
30	R/WSC	0x0	EPDis Endpoint Disable The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29:28	RO	0x0	reserved
27	WO	0x0	SNAK Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.
26	WO	0x0	CNAK Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:23	RO	0x0	reserved

Bit	Attr	Reset Value	Description
22	RW	0x0	TxFNum TxFIFO Number For Shared FIFO operation, this value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO. For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint 0.
21	R/WSC	0x0	Stall STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.
20	RO	0x0	reserved
19:18	RO	0x0	EPType Endpoint Type Hardcoded to 00 for control
17	RO	0x0	NAKSts NAK Status Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	reserved
15	RO	0x1	USBActEP USB Active Endpoint This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.

Bit	Attr	Reset Value	Description
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is not set. This field is not valid in Slave mode. Note: This field is valid only for Shared FIFO operations.
10:2	RO	0x0	reserved
1:0	RW	0x0	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes

USBOTG_DIEPINTn

Address: Operational Base + offset (0x0908 + 0x20 * n) 0≤n≤15

Device Endpoint-n Interrupt Register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	W1C	0x0	NYETIntrpt NYET interrupt The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint.
13	W1C	0x0	NAKIntrpt NAK interrupt The core generates this interrupt when a NAK is transmitted or received by the device.In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo.

Bit	Attr	Reset Value	Description
12	W1C	0x0	BbleErrIntrpt BbleErr (Babble Error) interrupt The core generates this interrupt when babble is received for the endpoint.
11	W1C	0x0	PktDrpSts Packet Dropped Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.
10	RO	0x0	reserved
9	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done Dependency: This bit is valid only when Scatter/Gather DMA mode is enabled.
8	W1C	0x0	TxfifoUndrn FIFO Underrun Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1; Thresholding is enabled; OUT Packet Error(OutPktErr). Applies to OUT endpoints only. This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1; Thresholding is enabled.

Bit	Attr	Reset Value	Description
7	W1C	0x0	TxFEmp Transmit FIFO Empty This bit is valid only for IN Endpoints.This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).
6	W1C	0x0	INEPNakEff IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled. Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.

Bit	Attr	Reset Value	Description
5	W1C	0x0	INTknEPMis IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in case of Scatter Gather DMA mode.
4	W1C	0x0	INTknTXFEmp IN Token Received When TxFIFO is Empty Indicates that an IN token was received when the associated TxFIFO periodic/nonperiodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.

Bit	Attr	Reset Value	Description
3	W1C	0x0	TimeOUT Timeout Condition In shared TX FIFO mode, applies to non-isochronous IN endpoints only. In dedicated FIFO mode, applies only to Control IN endpoints. In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	W1C	0x0	AHBErr AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.
1	W1C	0x0	EPDisbld Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

Bit	Attr	Reset Value	Description
0	W1C	0x0	XferCompl Transfer Completed Interrupt Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled: For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

USBOTG_DIEPTSIZn

Address: Operational Base + offset (0x0910 + 0x20 ∗ n)1≤n≤15

Device endpoint n transfer size register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:29	RW	0x0	MC Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID) Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA SETUP Packet Count (SUPCnt).Applies to control OUT Endpoints only.This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets

Bit	Attr	Reset Value	Description
28:19	RW	0x000	PktCnt Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters during IP configuration (parameter OTG_PACKET_COUNT_WIDTH). IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.
18:0	RW	0x00000	XferSize Transfer Size This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters during IP configuration (parameter OTG_TRANS_COUNT_WIDTH). The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO.OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

USBOTG_DIEPDMAn

Address: Operational Base + offset (0x0914 + 0x20 ∗ n) 0≤n≤15

Device endpoint-n DMA address register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

USBOTG_DTXFSTSn

Address: Operational Base + offset (0x0918 + 0x20 * n) 0≤n≤15

Device IN endpoint transmit FIFO status register

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	INEPTxFSpcAvail IN Endpoint TxFIFO Space Avail Indicates the amount of free space available in the Endpoint TxFIFO.Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'hn: n words available (where 0 . n . 32,768) 16'h8000: 32,768 words available Others: Reserved

USBOTG_DIEPDMABn

Address: Operational Base + offset (0x091c + 0x20 * n) 0≤n≤15

Device endpoint-n DMA buffer address register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DMABufferAddr DMA Buffer Address Holds the current buffer address.This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DIEPCTLn

Address: Operational Base + offset (0x0920 + 0x20 ∗ n) 0≤n≤14

Device endpoint-n control register

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	EPEna Endpoint Enable Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint ; For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.

Bit	Attr	Reset Value	Description
30	R/WSC	0x0	EPDis Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.
29	WO	0x0	SetD1PID Set DATA1 PID Applies to interrupt/bulk IN and OUT endpoints only.Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Set Odd (micro)frame (SetOddFr). Applies to isochronous IN and OUTendpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to odd (micro)frame. This field is not applicable for Scatter/Gather DMA mode.

Bit	Attr	Reset Value	Description
28	WO	0x0	SetD0PID Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only.Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro)frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to even (micro)frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receive descriptor structure.
27	WO	0x0	SNAK Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.

Bit	Attr	Reset Value	Description
25:22	RW	0x0	TxFNum TxFIFO Number Shared FIFO Operation:non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. 4'h0: Non-Periodic TxFIFO; Others: Specified Periodic TxFIFO number. Note: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area. Dedicated FIFO Operation:these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.
21	RW	0x0	Stall STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.

Bit	Attr	Reset Value	Description
20	RW	0x0	Snp Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RW	0x0	EPType Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt
17	RO	0x0	NAKSts NAK Status Applies to IN and OUT endpoints. Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.

Bit	Attr	Reset Value	Description
16	RO	0x0	DPID Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.Even/Odd (Micro)Frame (EO_FrNum) In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only.Indicates the (micro) rame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.

Bit	Attr	Reset Value	Description
15	R/WSC	0x0	USBActEP USB Active Endpoint Applies to IN and OUT endpoints.Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only.Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation.Note: This field is valid only for Shared FIFO operations.
10:0	RW	0x000	MPS Maximum Packet Size Applies to IN and OUT endpoints.The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

USBOTG_DOEPCTLn

Address: Operational Base + offset (0x0b00 + 0x20 ∗ n) 1≤n≤15

Device control OUT endpoint 0 control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	EPEna Endpoint Enable When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is disabled?such as for buffer-pointer based DMA mode)-this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.
30	WO	0x0	EPDis Endpoint Disable The application cannot disable control OUT endpoint 0.
29:28	RO	0x0	reserved
27	WO	0x0	SNAK Set NAK A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.
26	WO	0x0	CNAK Clear NAK A write to this bit clears the NAK bit for the endpoint.
25:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21	R/WSC	0x0	Stall STALL Handshake The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	RW	0x0	Snp Snoop Mode This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RO	0x0	EPType Endpoint Type Hardcoded to 2'b00 for control.
17	RO	0x0	NAKSts NAK Status Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit setting, the core always responds to SETUP data packets with an ACK handshake.
16	RO	0x0	reserved
15	RO	0x0	USBActEP USB Active Endpoint This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.
14:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1:0	RO	0x0	MPS Maximum Packet Size The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes

USBOTG_DOEPINTn

Address: Operational Base + offset (0x0b08 + 0x20 * n) 0≤n≤15

Device endpoint-n control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	W1C	0x0	NYETIntrpt NYET interrupt The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint.
13	W1C	0x0	NAKIntrpt NAK interrupt The core generates this interrupt when a NAK is transmitted or received by the device.In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to un-availability of data in the TXFifo.
12	W1C	0x0	BbleErrIntrpt BbleErr (Babble Error) interrupt The core generates this interrupt when babble is received for the endpoint.
11	W1C	0x0	PktDrpSts Packet Dropped Status This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. Dependency: This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.
10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	W1C	0x0	BNAIntr BNA (Buffer Not Available) Interrupt The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. Dependency: This bit is valid only when Scatter/Gather DMA mode is enabled.
8	W1C	0x0	TxfifoUndrn FIFO Underrun Applies to IN endpoints only. The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1, Thresholding is enabled, OUT Packet Error (OutPktErr). Applies to OUT endpoints only . This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet. Dependency: This interrupt is valid only when both of the following conditions are true: Parameter OTG_EN_DED_TX_FIFO==1, Thresholding is enabled.
7	W1C	0x0	TxFEmp Transmit FIFO Empty This bit is valid only for IN Endpoints. This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register(GAHBCFG.NPTxFEmpLvl)).

Bit	Attr	Reset Value	Description
6	W1C	0x0	INEPNakEff IN Endpoint NAK Effective Applies to periodic IN endpoints only. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled. Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.
5	W1C	0x0	INTknEPMis IN Token Received with EP Mismatch Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Status Phase Received For Control Write (StsPhseRcvd) This interrupt is valid only for Control OUT endpoints and only in Scatter Gather DMA mode.
4	W1C	0x0	INTknTXFEmp IN Token Received When TxFIFO is Empty Indicates that an IN token was received when the associated TxFIFO periodic/nonperiodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.

Bit	Attr	Reset Value	Description
3	W1C	0x0	TimeOUT Timeout Condition In shared TX FIFO mode, applies to non-isochronous IN endpoints only. In dedicated FIFO mode, applies only to Control IN endpoints. In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted.Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp). Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.
2	W1C	0x0	AHBErr AHB Error Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.
1	W1C	0x0	EPDisbld Endpoint Disabled Interrupt Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.

Bit	Attr	Reset Value	Description
0	W1C	0x0	XferCompl Transfer Completed Interrupt Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.

USBOTG_DOEPTSIZn

Address: Operational Base + offset (0x0b10 + 0x20 ∗ n) 1≤n≤15

Device endpoint n transfer size register

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved

Bit	Attr	Reset Value	Description
30:29	RW	0x0	MC Multi Count Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID) Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA SETUP Packet Count (SUPCnt).Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets

Bit	Attr	Reset Value	Description
28:19	RW	0x000	PktCnt Packet Count Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters during IP configuration (parameter OTG_PACKET_COUNT_WIDTH). IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.
18:0	RW	0x00000	XferSize Transfer Size This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters during IP configuration (parameter OTG_TRANS_COUNT_WIDTH). The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.

USBOTG_DOEPDMAn

Address: Operational Base + offset (0x0b14 + 0x20 ∗ n) 0≤n≤15

Device Endpoint-n DMA Address Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	DMAAddr DMA Address Holds the start address of the external memory for storing or fetching endpoint data. Note: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.

USBOTG_DOEPDMABn

Address: Operational Base + offset (0x0b1c + 0x20 * n) 0≤n≤15

Device endpoint-n DMA buffer address register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DMABufferAddr DMA Buffer Address Holds the current buffer address.This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.

USBOTG_DOEPCTLn

Address: Operational Base + offset (0x0b20 + 0x20 * n) 0≤n≤14

Device endpoint-n control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31	R/WSC	0x0	EPEna Endpoint Enable Applies to IN and OUT endpoints. When Scatter/Gather DMA mode is enabled, For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. For OUT endpoint it indicates that the descriptor structure and data buffer to receive data is setup. When Scatter/Gather DMA mode is enabled-such as for buffer-pointer based DMA mode: For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint; For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. Note: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.
30	R/WSC	0x0	EPDis Endpoint Disable Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.

Bit	Attr	Reset Value	Description
29	RO	0x0	SetD1PID Field0001 Abstract Applies to interrupt/bulk IN and OUT endpoints only.Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Set Odd (micro)frame (SetOddFr). Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to odd (micro)frame. This field is not applicable for Scatter/Gather DMA mode.
28	WO	0x0	SetD0PID Set DATA0 PID Applies to interrupt/bulk IN and OUT endpoints only.Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0.This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro)frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to even (micro)frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receive descriptor structure.
27	WO	0x0	SNAK Set NAK Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.

Bit	Attr	Reset Value	Description
26	WO	0x0	CNAK Clear NAK Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.
25:22	RW	0x0	TxFNum TxFIFO Number Shared FIFO Operation:non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. 4'h0: Non-Periodic TxFIFO; Others: Specified Periodic TxFIFO number. Note: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area. Dedicated FIFO Operation:these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.

Bit	Attr	Reset Value	Description
21	RW	0x0	Stall STALL Handshake Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.
20	RW	0x0	Snp Snoop Mode Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.
19:18	RW	0x0	EPType Endpoint Type Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt

Bit	Attr	Reset Value	Description
17	RO	0x0	NAKSts NAK Status Applies to IN and OUT endpoints. Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet.

Bit	Attr	Reset Value	Description
16	RO	0x0	DPID Endpoint Data PID Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Even/Odd (Micro)Frame (EO_FrNum). In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only. Indicates the (micro)frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.

Bit	Attr	Reset Value	Description
15	R/WSC	0x0	USBActEP USB Active Endpoint Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.
14:11	RW	0x0	NextEp Next Endpoint Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. Note: This field is valid only for Shared FIFO operations.
10:0	RW	0x000	MPS Maximum Packet Size Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

USBOTG_PCGCR

Address: Operational Base + offset (0x0b24)

Power and clock gating control register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:14	RW	0x0802e	RestoreValue Restore Value (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). Defines port clock select for different speeds. [31] if_dev_mode - 1: Device mode, core restored as device - 0: Host mode, core restored as host [30:29] p2hd_prt_spd (PRT speed) - 00: HS - 01: FS - 10: LS - 11: Reserved [28:27] p2hd_dev_enum_spd (Device enumerated speed) - 00: HS - 01: FS (30/60 MHz clk) - 10: LS - 11: FS (48 MHz clk) [26:20] mac_dev_addr (MAC device address) Device address [19] mac_termselect (Termination selection) - 0: HS_TERM (Program for High Speed) - 1: FS_TERM (Program for Full Speed) [18:17] mac_xcvrselect (Transceiver select) - 00: HS_XCVR (High Speed) - 01: FS_XCVR (Full Speed) - 10: LS_XCVR (Low Speed) - 11: LFS_XCVR (Reserved) [16] sh2pl_prt_ctl[0] - 1: prt_power enabled - 0: prt_power disabled [15:14] prt_clk_sel (Refer prt_clk_sel table)
13	RW	0x0	EssRegRestored Essential Register Values Restored (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). When a value of 1 is written to this field, it indicates that register values of essential registers have been restored.
12:10	RO	0x0	reserved

Bit	Attr	Reset Value	Description
9	RO	0x0	RestoreMode Restore Mode (Applicable only when Hibernation is enabled (OTG_EN_PWROPT=2). The application should program this bit to specify the restore mode during RESTORE POINT before programming PCGCCTL.EssRegRest bit is set. Host Mode: 1'b0: Host Initiated Resume, Host Initiated Reset 1'b1: Device Initiated Remote Wake up Device Mode: 1'b0: Device Initiated Remote Wake up 1'b1: Host Initiated Resume, Host Initiated Reset
8	RW	0x0	ResetAfterSusp Reset After Suspend Applicable in Partial power-down mode. In partial power-down mode of operation, this bit needs to be set in host mode before clamp is removed if the host needs to issue reset after suspend. If this bit is not set,then the host issues resume after suspend. This bit is not applicable in device mode and non-partial power-down mode.In Hibernation mode, this bit needs to be set at RESTORE_POINT before PCGCCTL.EssRegRestored is set. In this case, PCGCCTL.restore_mode needs to be set to wait_restore.
7	RO	0x0	L1Suspended Deep Sleep This bit indicates that the PHY is in deep sleep when in L1 state.
6	RO	0x0	PhySleep PHY in Sleep This bit indicates that the PHY is in the Sleep state.
5	RW	0x0	Enbl_L1Gating Enable Sleep Clock Gating When this bit is set, core internal clock gating is enabled in Sleep state if the core cannot assert utmi_l1_suspend_n. When this bit is not set, the PHY clock is not gated in Sleep state.

Bit	Attr	Reset Value	Description
4	RO	0x0	reserved
3	RW	0x0	RstPdwnModule Reset Power-Down Modules This bit is valid only in Partial Power-Down mode. The application sets this bit when the power is turned off. The application clears this bit after the power is turned on and the PHY clock is up.
2	RW	0x0	PwrClmp Power Clamp This bit is valid only in Partial Power-Down mode (OTG_EN_PWROPT = 1). The application sets this bit before the power is turned off to clamp the signals between the power-on modules and the power-off modules. The application clears the bit to disable the clamping before the power is turned on.
1	RW	0x0	GateHclk Gate Hclk The application sets this bit to gate hclk to modules other than the AHB Slave and Master and wakeup logic when the USB is suspended or the session is not valid. The application clears this bit when the USB is resumed or a new session starts.
0	RW	0x0	StopPclk Stop Pclk The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts.

USB_EPBUF0
Address: Operational Base + offset (0x1000)
Device endpoint 0 / host out channel 0 address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF0 Device endpoint 0 / host out channel 0 address Device IN Endpoint 0 /Host OUT Channel 0: DFIFO Write Access Device OUT Endpoint 0 /Host IN Channel 0: DFIFO Read Access

USB_EPBUF1
Address: Operational Base + offset (0x2000)
Device endpoint 1 / host out channel 1 address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF1 Device endpoint 1 / host out channel 1 address Device IN Endpoint 1/Host OUT Channel 1: DFIFO Write Access Device OUT Endpoint 1/Host IN Channel 1: DFIFO Read Access

USB_EPBUF2
Address: Operational Base + offset (0x3000)
Device endpoint 2 / host out channel 2 address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF2 Device endpoint 2 / host out channel 2 address Device IN Endpoint 2/Host OUT Channel 2: DFIFO Write Access Device OUT Endpoint 2/Host IN Channel 2: DFIFO Read Access

USB_EPBUF3
Address: Operational Base + offset (0x4000)
Device endpoint 3 / host out channel 3 address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF3 Device endpoint 3 / host out channel 3 address Device IN Endpoint 3/Host OUT Channel 3: DFIFO Write Access Device OUT Endpoint 3/Host IN Channel 3: DFIFO Read Access

USB_EPBUF4
Address: Operational Base + offset (0x5000)
Device endpoint 4 / host out channel 4 address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF4 Device endpoint 4 / host out channel 4 address Device IN Endpoint 4/Host OUT Channel 4: DFIFO Write Access Device OUT Endpoint 4/Host IN Channel 4: DFIFO Read Access

USB_EPBUF5
Address: Operational Base + offset (0x6000)
Device endpoint 5 / host out channel 5 address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF5 Device endpoint 5 / host out channel 5 address Device IN Endpoint 5/Host OUT Channel 5: DFIFO Write Access Device OUT Endpoint 5/Host IN Channel 5: DFIFO Read Access

USB_EPBUF6
Address: Operational Base + offset (0x7000)
Device endpoint 6 / host out channel 6 address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF6 Device endpoint 6 / host out channel 6 address Device IN Endpoint 6/Host OUT Channel 6: DFIFO Write Access Device OUT Endpoint 6/Host IN Channel 6: DFIFO Read Access

USB_EPBUF7
Address: Operational Base + offset (0x8000)
Device endpoint 7 / host out channel 7 address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	EPBUF7 Device endpoint 7 / host out channel 7 address Device IN Endpoint 7/Host OUT Channel 7: DFIFO Write Access Device OUT Endpoint 7/Host IN Channel 7: DFIFO Read Access

27.6 Interface description

Table 27-2 USB OTG 2.0 Interface Description

Module Pin	Direction	Pad Name	pinmux
OTG_VSSAC	AG	OTG_VSSAC	-
OTG_DVSS	DG	OTG_DVSS	-
OTG_DVDD	DP	OTG_DVDD	-
OTG_VDD25	AP	OTG_VDD25	-
OTG_DM	A	OTG_DM	-
OTG_RKELVIN	A	OTG_RKELVIN	-
OTG_DP	A	OTG_DP	-
OTG_VSSA	AG	OTG_VSSA	-
OTG_VBUS	A	OTG_VBUS	-
OTG_VDD33	AP	OTG_VDD33	-
OTG_ID	A	OTG_ID	-
otg_drv_vbus	O	GPIO0_A[5]	GRF_GPIO0A_IOMUX [10]=1

***Note**: **A**—Analog pad ; **AP**—Analog power; **AG**—Analog ground ;**DP**—Digital power ; **DG**–Digital ground;*

Chapter 28 I2S/PCM0 Controller (8 channel)

28.1 Overview

The I2S/PCM0 controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

Not only I2S but also PCM mode surround audio output (up to 7.1channel) and stereo input are supported in I2S/PCM0 controller.

- Support five internal 32-bit wide and 32-location deep FIFOs, four for transmitting and one for receiving audio data
- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2,4,6,8 channels audio transmitting in I2S and PCM mode
- Support 2 channels audio receiving in I2S and PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 3 independent LRCK signals, one for receiving and two for transmitting audio data
- Support configurable SCLK and LRCK polarity
- Support SCLK is equivalent to MCLK divided by an even number range from 2 to 64 in master mode

28.2 Block Diagram

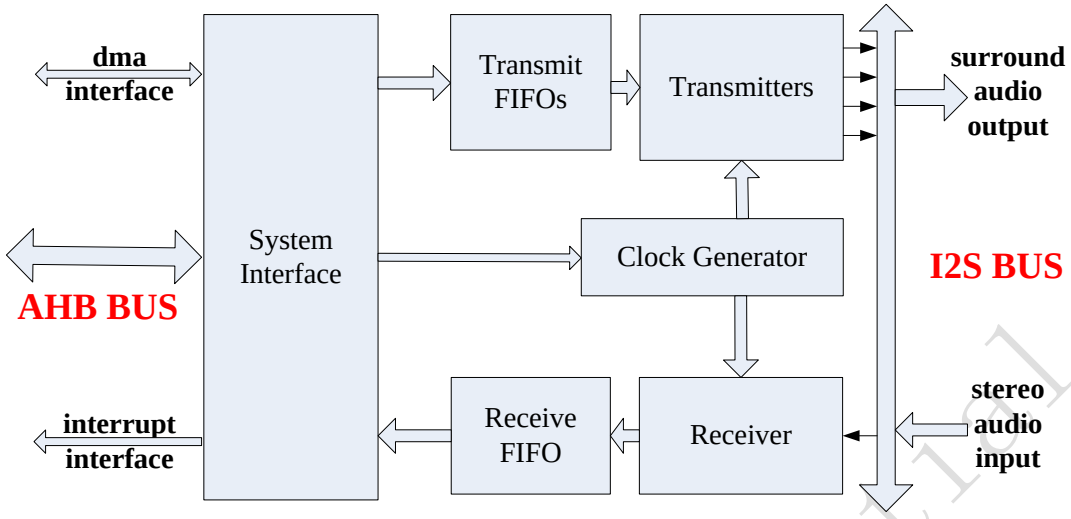


Fig. 28-1 I2S/PCM0 controller (8 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshake interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either master or slave, with I2S or PCM mode surround (up to 7.1 channel) serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFOs

The Transmit FIFOs are the buffer to store transmitted audio data. Each of the size of the four FIFOs is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

28.3 Function description

In the I2S/PCM0 controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

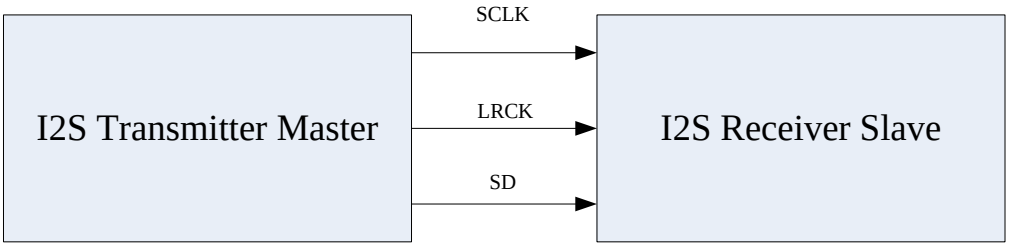


Fig. 28-2I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

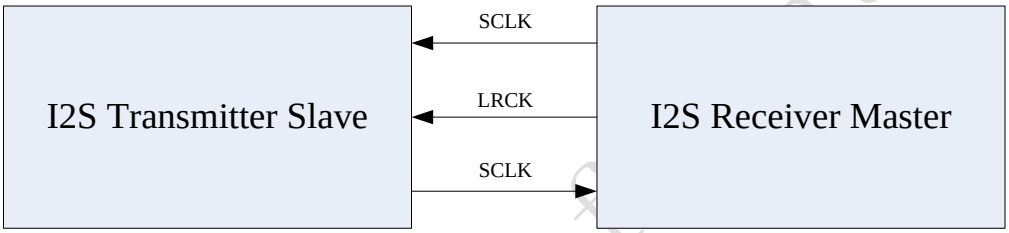


Fig. 28-3I2S transmitter-slave& receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

28.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s0_lrck_rx/i2s0_lrck_tx0) signal, it goes low to indicate left channel and high to right channel. For SD (i2s0_sdo0, i2s0_sdo1, i2s0_sdo2, i2s0_sdo3, i2s0_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

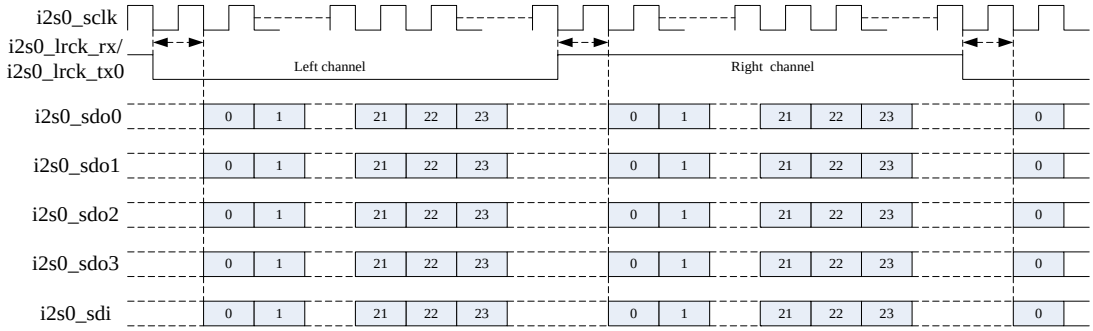


Fig. 28-4I2S normal mode timing format

28.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s0_lrck_rx / i2s0_lrck_tx0) signal, it goes high to indicate left channel and low to right channel. For SD (i2s0_sdo0, i2s0_sdo1, i2s0_sdo2, i2s0_sdo3, i2s0_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

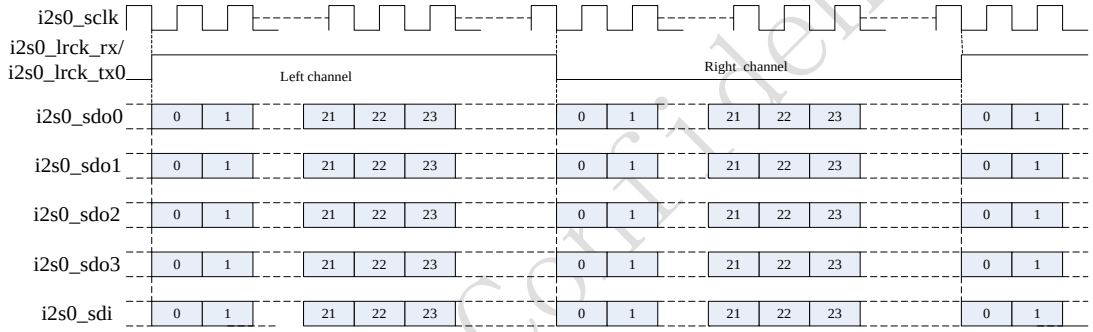


Fig. 28-5I2S left justified mode timing format

28.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s0_lrck_rx / i2s0_lrck_tx0) signal, it goes high to indicate left channel and low to right channel. For SD (i2s0_sdo0, i2s0_sdo1, i2s0_sdo2, i2s0_sdo3, i2s0_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

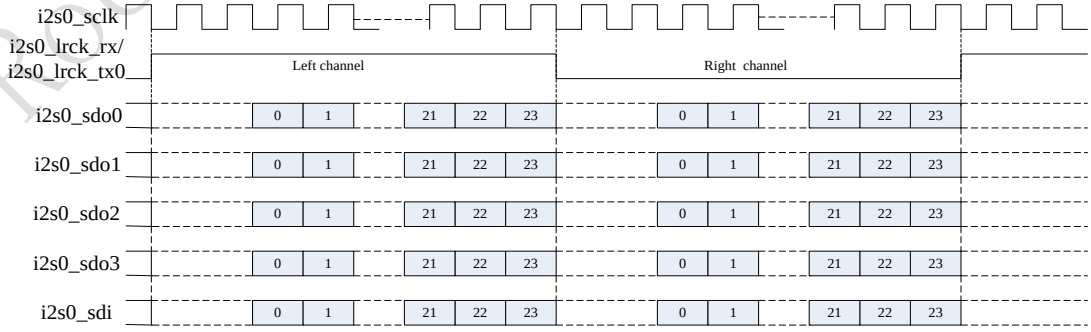


Fig. 28-6I2S right justified modetiming format

28.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s0_lrck_rx /

i2s0_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s0_sdo0, i2s0_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

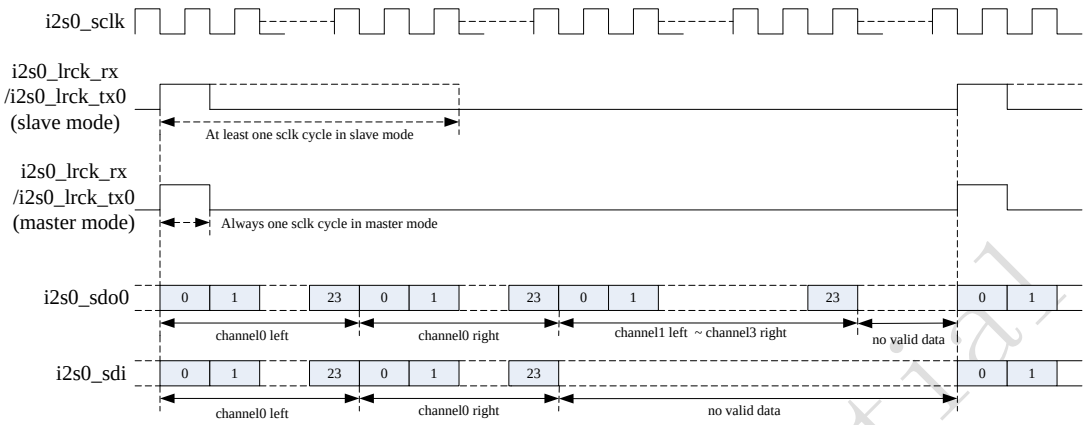


Fig. 28-7PCM early modetiming format

28.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s0_lrck_rx / i2s0_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s0_sdo0, i2s0_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

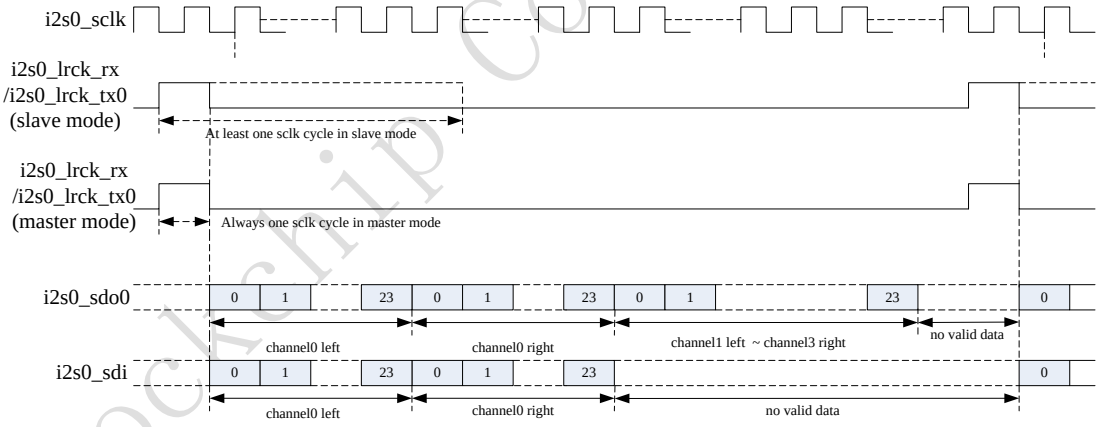


Fig. 28-8PCM late1 modetiming format

28.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s0_lrck_rx / i2s0_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s0_sdo0, i2s0_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

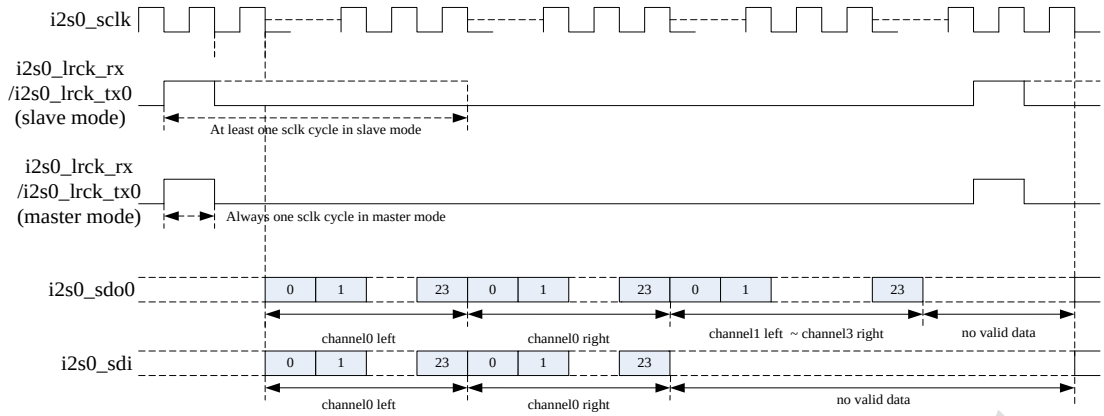


Fig. 28-9PCM late2 modetiming format

28.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s0_lrck_rx / i2s0_lrck_tx0) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s0_sdo0, i2s0_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

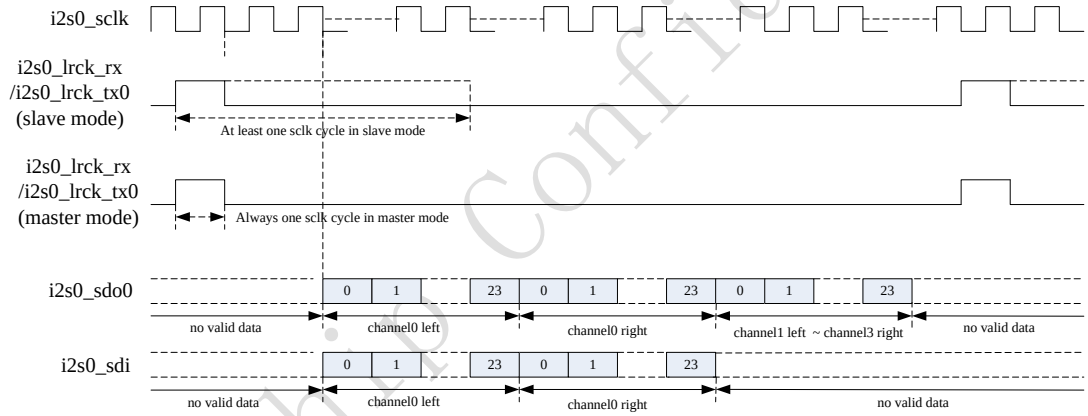


Fig. 28-10PCM late3 modetiming format

28.4 Register description

28.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
I2S0_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2S0_RXCR	0x0004	W	0x0000000f	receive operation control register
I2S0_CKR	0x0008	W	0x00071f1f	clock generation register
I2S0_FIFOLR	0x000c	W	0x00000000	FIFO level register
I2S0_DMACR	0x0010	W	0x001f0000	DMA control register
I2S0_INTCR	0x0014	W	0x01f00000	interrupt control register
I2S0_INTSR	0x0018	W	0x00000000	interrupt status register
I2S0_XFER	0x001c	W	0x00000000	Transfer Start Register
I2S0_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2S0_TXDR	0x0024	W	0x00000000	Transimt FIFO Data Register

Name	Offset	Size	Reset Value	Description
I2S0_RXDR	0x0028	W	0x00000000	Receive FIFO Data Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

28.4.2 Detail Register Description

I2S0_TXCR

Address: Operational Base + offset (0x0000)

transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right jusitified counter (Can be written only when XFER[0] bit is 0.) Only vailid in I2S Right justified format and slave tx mode is selected. Start to tramsmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel select register (Can be written only when XFER[0] bit is 0.) 0:channel 0 enable 1:channel 0 & channel 1 enable 2:channel 0 & channel 1 & channel 2 enable 3:channel 0 & channel 1 & channel 2 & channel 3 enable
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:low 16bit data valid from AHB/APB bus. 1:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel.
13	RO	0x0	reserved

Bit	Attr	Reset Value	Description
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S0_RXCR

Address: Operational Base + offset (0x0004)

receive operation control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:low 16bit data valid from AHB/APB bus. 1:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified

Bit	Attr	Reset Value	Description
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2S0_CKR

Address: Operational Base + offset (0x0008)

clock generation register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)

Bit	Attr	Reset Value	Description
24	RW	0x0	TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)
23:16	RW	0x07	MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclk frequecy / txsclk frequecy-1) 0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; …… 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; …… 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description
15:8	RW	0x1f	RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… 253: 254fs 254: 255fs 255: 256fs
7:0	RW	0x1f	TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… 253: 254fs 254: 255fs 255: 256fs

I2S0_FIFOLR

Address: Operational Base + offset (0x000c)

FIFO level register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RO	0x00	RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.
23:18	RO	0x00	TFL3 Transmit FIFO3 Level Contains the number of valid data entries in the transmit FIFO3.

Bit	Attr	Reset Value	Description
17:12	RO	0x00	TFL2 Transmit FIFO2 Level Contains the number of valid data entries in the transmit FIFO2.
11:6	RO	0x00	TFL1 Transmit FIFO1 Level Contains the number of valid data entries in the transmit FIFO1.
5:0	RO	0x00	TFL0 Transmit FIFO0 Level Contains the number of valid data entries in the transmit FIFO0.

I2S0_DMACR
Address: Operational Base + offset (0x0010)
DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2S0_INTCR
Address: Operational Base + offset (0x0014)
interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24:20	RW	0x1f	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if CSR=00; TXFIFO1 if CSR=01, TXFIFO2 if CSR=10, TXFIFO3 if CSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2S0_INTSR
Address: Operational Base + offset (0x0018)
interrupt status register

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2S0_XFER
Address: Operational Base + offset (0x001c)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2S0_CLR
Address: Operational Base + offset (0x0020)
SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX logic clear This is a self cleard bit. Write 1 to clear all receive logic.
0	RW	0x0	TXC TX logic clear This is a self cleard bit. Write 1 to clear all transmit logic.

I2S0_TXDR
Address: Operational Base + offset (0x0400~0x7FC)
Transimt FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transimt FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2S0_RXDR
Address: Operational Base + offset (0x0800~0xBFC)
Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

28.5 Timing Diagram

28.5.1 Master mode

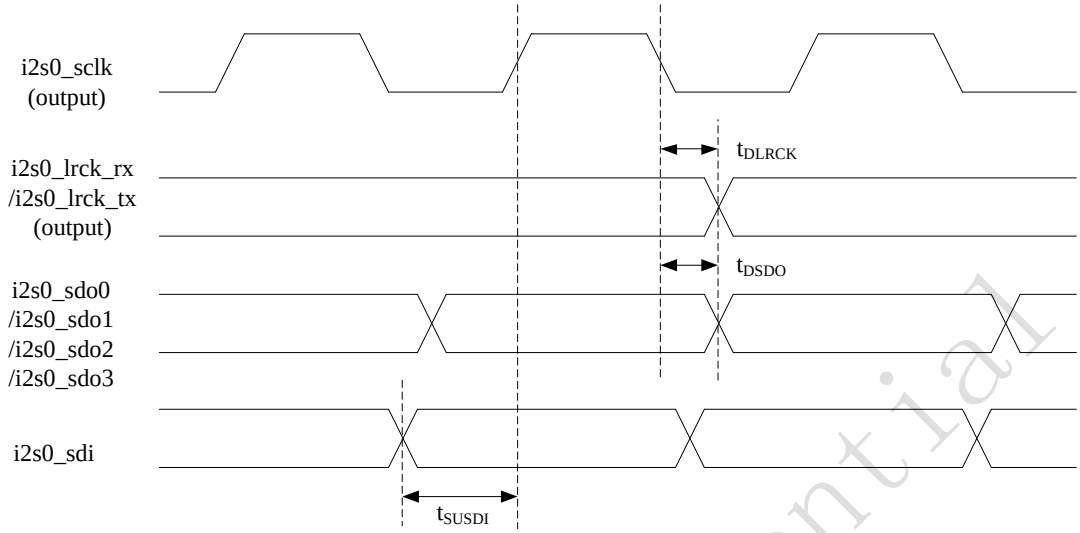


Fig. 28-11Master mode timing diagram

Table 28-1Meaning of the parameterin Fig. 28-12

Parameter	Description	min	typ	max	unit
t_{DLRCK}	i2s0_lrck_rx/i2s0_lrck_tx propagation delay from i2s0_sclk falling edge	1.65	2.42	3.38	ns
t_{DSDO}	i2s0_sdo0/i2s0_sdo1/i2s0_sdo2/i2s0_sdo3 propagation delay from i2s0_sclk falling edge	1.61	2.41	3.44	ns
t_{SUSDI}	i2s0_sdi setup time to i2s0_sclk rising edge	4.53	6.52	9.04	ns

28.5.2 Slave mode

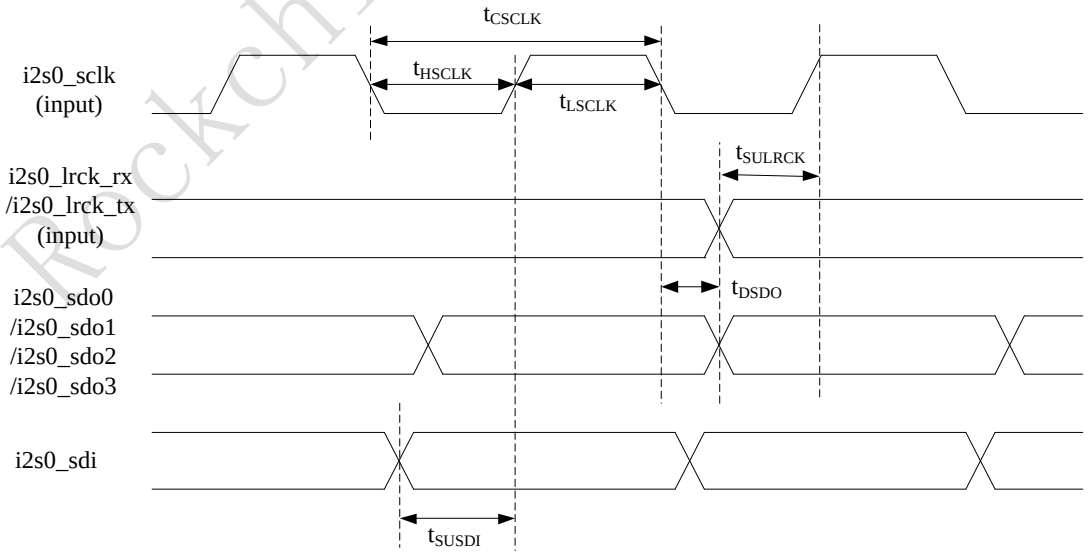


Fig. 28-12Slave mode timing diagram

Table 28-2Meaning of the parameter in Fig. 28-13

Parameter	Description	min	typ	max	unit
t_{CSCLK}	i2s0_sclk cycle time (cannot be less than 40ns)	-	-	-	ns
t_{HSCLK}	i2s0_sclk pulse width high (cannot be less than 20ns)	-	-	-	ns
t_{LSCLK}	i2s0_sclk pulse width low (cannot be less than 20ns)	-	-	-	ns
t_{SULRCK}	i2s0_lrck_rx/i2s0_lrck_tx setup time to i2s0_sclk falling edge	0.94	1.36	1.42	ns
t_{DSDO}	i2s0_sdo0/i2s0_sdo1/i2s0_sdo2/i2s0_sdo3 propagation delay from i2s0_sclk falling edge	5.37	7.89	10.96	ns
t_{SUSDI}	i2s0_sdi setup time to i2s0_sclk rising edge	0.71	0.89	0.81	ns

28.6 Interface description

Module Pin	Direction	Pad Name	IOMUX Setting
i2s0_sdi	I	GPIO0_A[7]	GRF_GPIO0A_IOMUX[14]=1'b1
i2s0_clk	O	GPIO0_B[0]	GRF_GPIO0B_IOMUX[0]=1'b1
i2s0_sclk	I/O	GPIO0_B[1]	GRF_GPIO0B_IOMUX[2]=1'b1
i2s0_lrck_rx	I/O	GPIO0_B[2]	GRF_GPIO0B_IOMUX[4]=1'b1
i2s0_lrck_tx	I/O	GPIO0_B[3]	GRF_GPIO0B_IOMUX[6]=1'b1
i2s0_sdo0	O	GPIO0_B[4]	GRF_GPIO0B_IOMUX[8]=1'b1
i2s0_sdo1	O	GPIO0_B[5]	GRF_GPIO0B_IOMUX[10]=1'b1
i2s0_sdo2	O	GPIO0_B[6]	GRF_GPIO0B_IOMUX[12]=1'b1
i2s0_sdo3	O	GPIO0_B[7]	GRF_GPIO0B_IOMUX[14]=1'b1

28.7 Application Notes

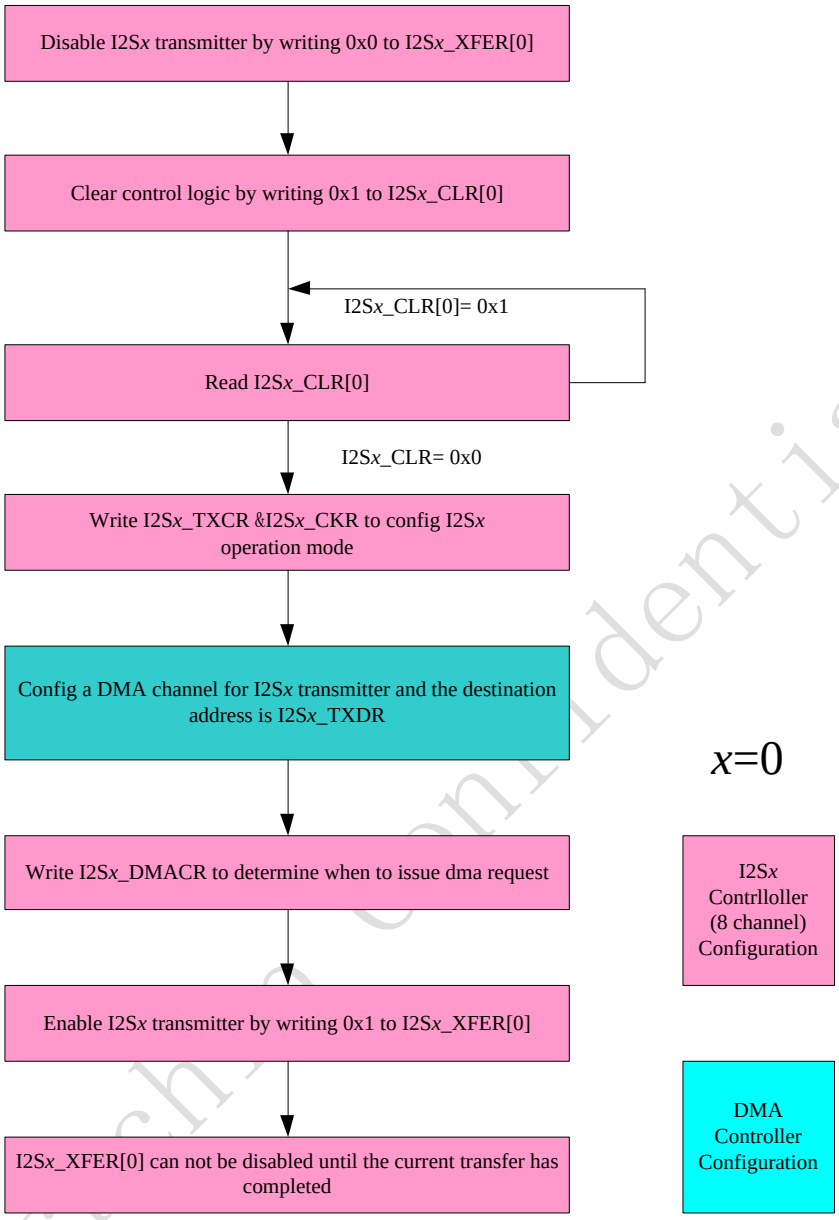


Fig. 28-13 I2S/PCM0 controller (8 channel) transmit operation flow chart

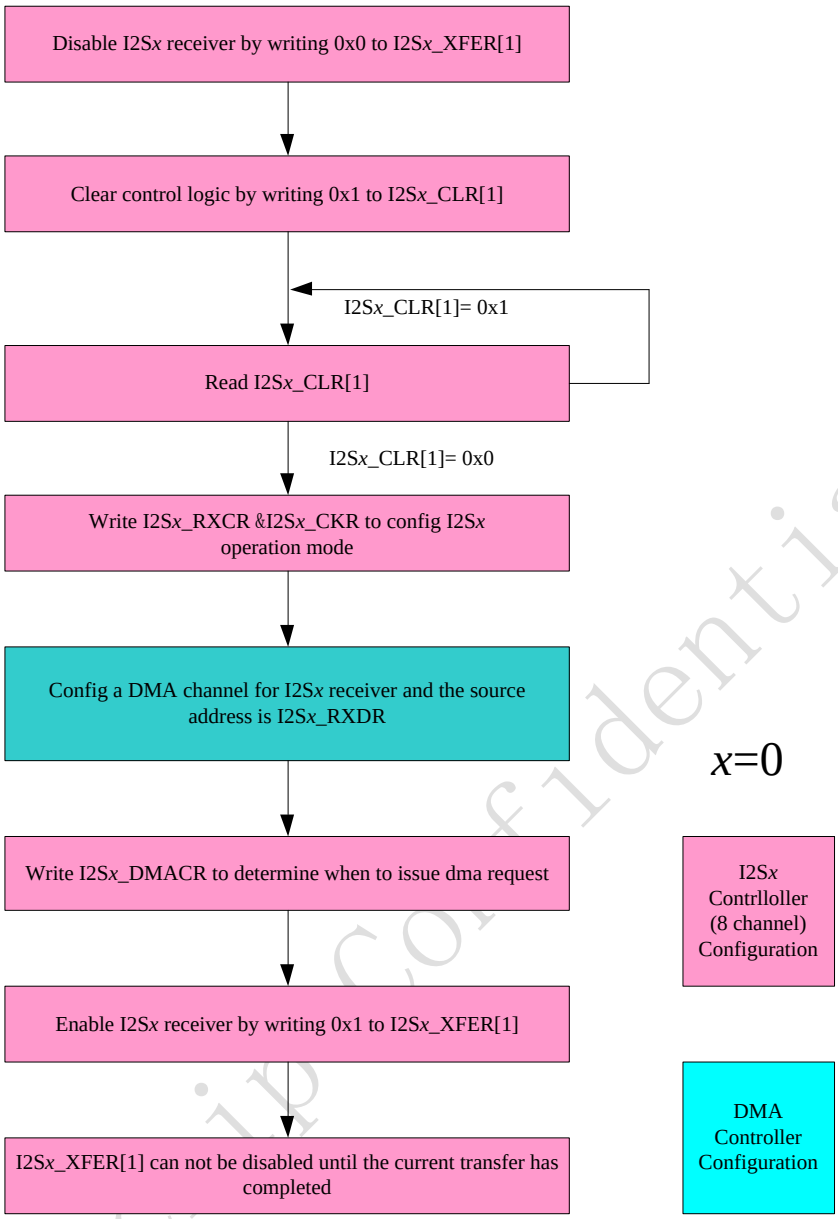


Fig. 28-14I2S/PCM0 controller (8 channel) receive operation flow chart

Chapter 29 I2S/PCM1/2 Controller (2 channel)

29.1 Overview

The I2S/PCM1/2 controller is designed for interfacing between the AHB bus and the I2S bus.

The I2S bus (Inter-IC sound bus) is a serial link for digital audio data transfer between devices in the system and be invented by Philips Semiconductor. Now it is widely used by many semiconductor manufacturers.

Devices often use the I2S bus are ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and the embedded SoC platform together and provide an audio interface solution for the system.

Not only I2S but also PCM mode stereo audio output and input are supported in I2S/PCM1/2 controller.

- Support two internal 32-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving audio data
- Support AHB bus interface
- Support 16 ~ 32 bits audio data transfer
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support 2 channels audio transmitting in I2S mode but 2,4,6,8 channels in PCM mode
- Support 2 channels audio receiving in I2S and PCM mode
- Support up to 192kHz sample rate
- Support I2S normal, left and right justified mode serial audio data transfer
- Support PCM early, late1, late2, late3 mode serial audio data transfer
- Support MSB or LSB first serial audio data transfer
- Support 16 to 31 bit audio data left or right justified in 32-bit wide FIFO
- Support two 16-bit audio data store together in one 32-bit wide location
- Support 2 independent LRCK signals, one for receiving and the other for transmitting audio data
- Support configurable SCLK and LRCK polarity
- Support SCLK is equivalent to MCLK divided by an even number range from 2 to 64 in master mode

29.2 Block Diagram

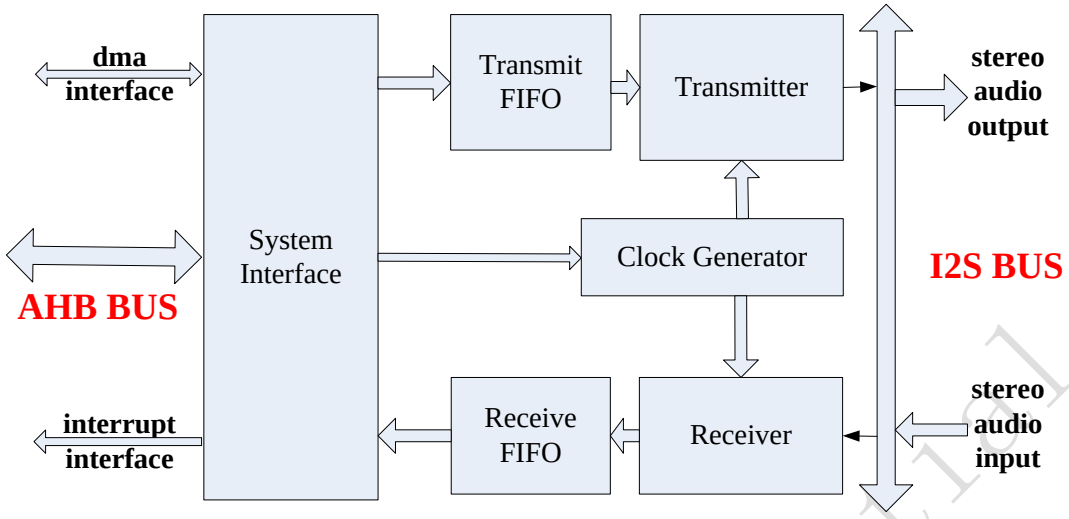


Fig. 29-1 I2S/PCM1/2 controller (2 channel) Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshake interface.

Clock Generator

The Clock Generator implements clock generation function. The input source clock to the module is MCLK_I2S, and by the divider of the module, the clock generator generates SCLK and LRCK to transmitter and receiver.

Transmitters

The Transmitters implement transmission operation. The transmitters can act as either master or slave, with I2S or PCM mode surround (up to 7.1 channel) serial audio interface.

Receiver

The Receiver implements receive operation. The receiver can act as either master or slave, with I2S or PCM mode stereo serial audio interface.

Transmit FIFO

The Transmit FIFO is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Receive FIFO

The Receive FIFO is the buffer to store received audio data. The size of the FIFO is 32bits x 32.

29.3 Function description

In the I2S/PCM1/2 controller, there are four conditions: transmitter-master & receiver-master; transmitter-master & receiver-slave; transmitter-slave & receiver-master; transmitter-slave & receiver-slave.

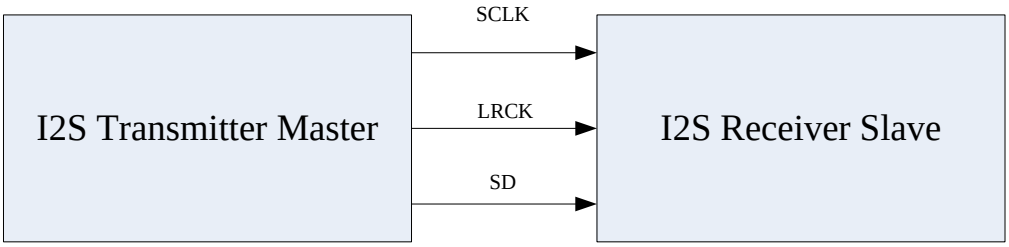


Fig. 29-2I2S transmitter-master & receiver-slave condition

When transmitter acts as a master, it sends all signals to receiver (slave), and CPU control when to send clock and data to the receiver. When acting as a slave, SD signal still goes from transmitter to receiver, but SCLK and LRCK signals are from receiver (master) to transmitter. Based on three interface specifications, transmitting data should be ready before transmitter receives SCLK and LRCK signals. CPU should know when the receiver to initialize a transaction and when to send data.

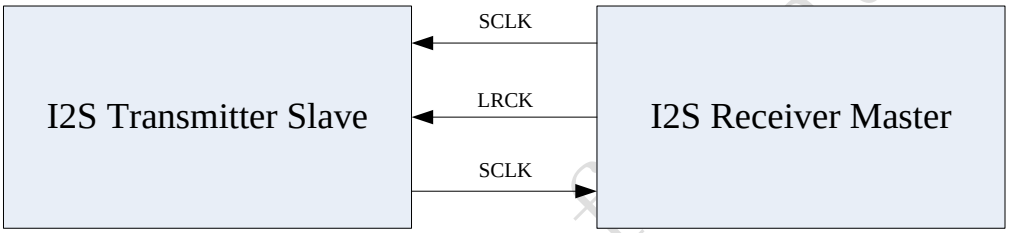


Fig. 29-3I2S transmitter-slave & receiver-master condition

When the receiver acts as a master, it sends SCLK and LRCK signals to the transmitter (slave) and receives serial data. So CPU must tell the transmitter when to start a transaction for it to prepare transmitting data then start a transfer and send clock and channel-select signals. When the receiver acts as a slave, CPU should only do initial setting and wait for all signals and then start reading data.

Before transmitting or receiving data, CPU need do initial setting to the I2S register. These includes CPU settings, I2S interface registers settings, and maybe the embedded SoC platform settings. These registers must be set before starting data transfer.

29.3.1 I2S normal mode

This is the waveform of I2S normal mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes low to indicate left channel and high to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK changes. The range of SD signal width is from 16 to 32bits.

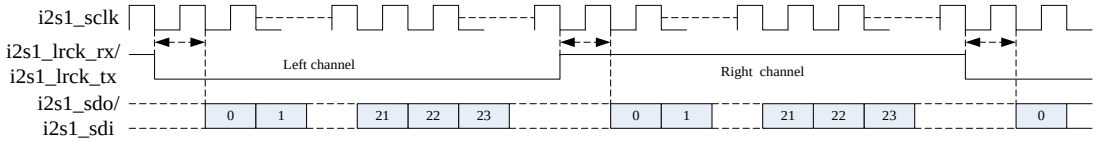


Fig. 29-4I2S normal mode timing format

29.3.2 I2S left justified mode

This is the waveform of I2S left justified mode. For LRCK (i2s1_lrck_rx / i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel.

For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK changes. The range of SD signal width is from 16 to 32bits.

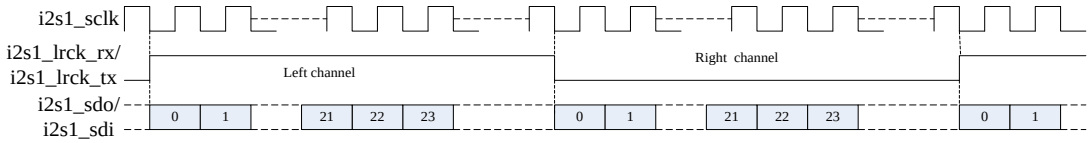


Fig. 29-5I2S left justified mode timing format

29.3.3 I2S right justified mode

This is the waveform of I2S right justified mode. For LRCK (i2s1_lrck_rx/ i2s1_lrck_tx) signal, it goes high to indicate left channel and low to right channel. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first; but different from I2S normal or left justified mode, its data is aligned to last bit at the edge of the LRCK signal. The range of SD signal width is from 16 to 32bits.

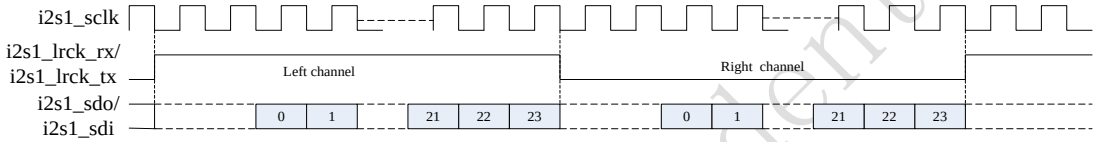


Fig. 29-6I2S right justified mode timing format

29.3.4 PCM early mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first and sends the first bit at the same time when LRCK goes high. The range of SD signal width is from 16 to 32bits.

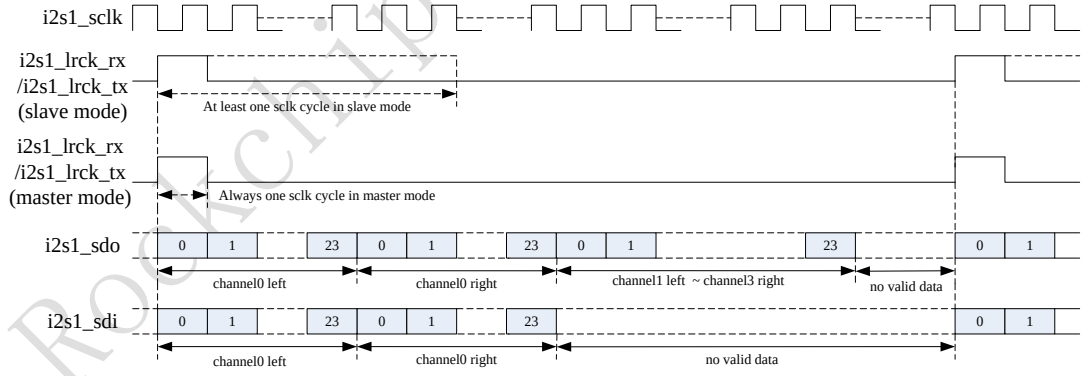


Fig. 29-7PCM early mode timing format

29.3.5 PCM late1 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first and sends the first bit one SCLK clock cycle after LRCK goes high. The range of SD signal width is from 16 to 32bits.

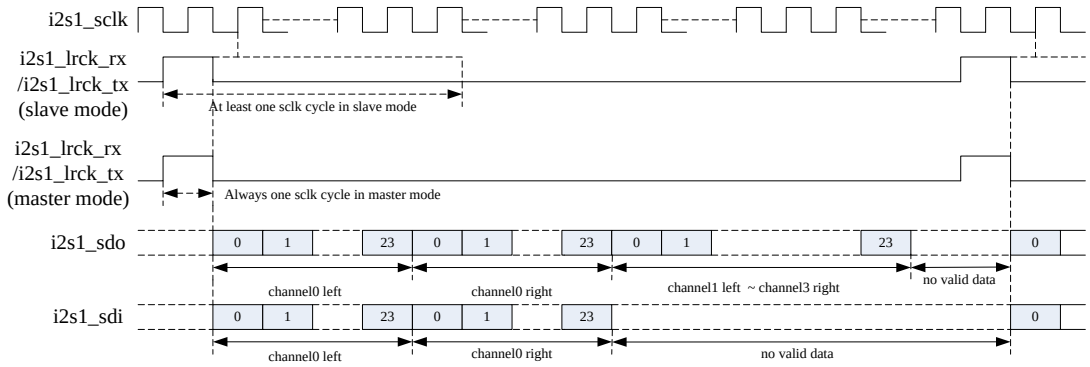


Fig. 29-8PCM late1 mode timing format

29.3.6 PCM late2 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first and sends the first bit two SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

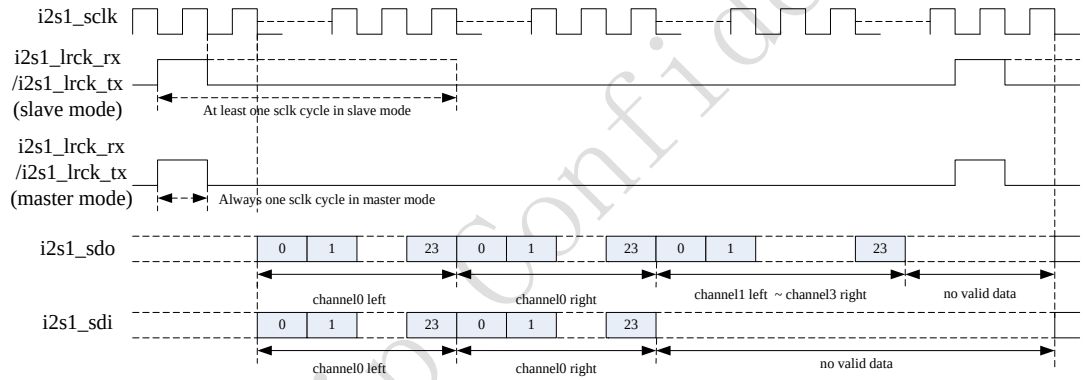


Fig. 29-9PCM late2 mode timing format

29.3.7 PCM late3 mode

This is the waveform of PCM early mode. For LRCK (i2s1_lrck_rx/i2s1_lrck_tx) signal, it goes high to indicate the start of a group of audio channels. For SD (i2s1_sdo, i2s1_sdi) signal, it transfers MSB or LSB first and sends the first bit three SCLK clock cycles after LRCK goes high. The range of SD signal width is from 16 to 32bits.

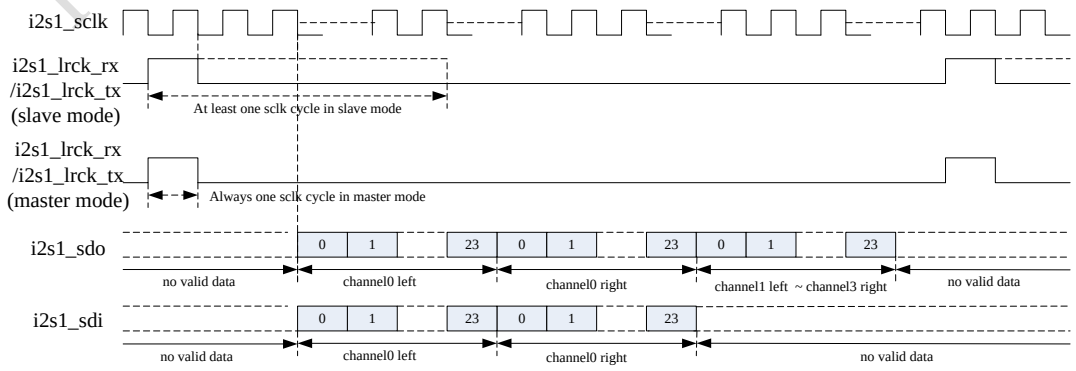


Fig. 29-10PCM late3 mode timing format

29.4 Register description

29.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
I2Sx_TXCR	0x0000	W	0x0000000f	transmit operation control register
I2Sx_RXCR	0x0004	W	0x0000000f	receive operation control register
I2Sx_CKR	0x0008	W	0x00071f1f	clock generation register
I2Sx_FIFOLR	0x000c	W	0x00000000	FIFO level register
I2Sx_DMACR	0x0010	W	0x001f0000	DMA control register
I2Sx_INTCR	0x0014	W	0x00000000	interrupt control register
I2Sx_INTSR	0x0018	W	0x00000000	interrupt status register
I2Sx_XFER	0x001c	W	0x00000000	Transfer Start Register
I2Sx_CLR	0x0020	W	0x00000000	SCLK domain logic clear Register
I2Sx_TXDR	0x0400 ~0x7FC	W	0x00000000	Transimt FIFO Data Register
I2Sx_RXDR	0x0800 ~0xBFC	W	0x00000000	Receive FIFO Data Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access, x=1,2*

29.4.2 Detail Register Description

I2Sx_TXCR
Address: Operational Base + offset (0x0000)
transmit operation control register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22:17	RW	0x00	RCNT right jusitified counter (Can be written only when XFER[0] bit is 0.) Only vailid in I2S Right justified format and slave tx mode is selected. Start to tramsmit data RCNT sclk cycles after left channel valid.
16:15	RW	0x0	CSR Channel select register Must be 2'b00.
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[0] bit is 0.) Only valid when VDW select 16bit data. 0:low 16bit data valid from AHB/APB bus. 1:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel.

Bit	Attr	Reset Value	Description
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[0] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[0] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[0] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[0] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[0] bit is 0.) 0: I2S format 1: PCM format

Bit	Attr	Reset Value	Description
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[0] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2Sx_RXCR
Address: Operational Base + offset (0x0004)
receive operation control register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14	RW	0x0	HWT Halfword word transform (Can be written only when XFER[1] bit is 0.) Only valid when VDW select 16bit data. 0:low 16bit data valid from AHB/APB bus. 1:32 bit data valid from AHB/APB bus. Low 16 bit for left channel and high 16 bit for right channel.
13	RO	0x0	reserved
12	RW	0x0	SJM Store justified mode (Can be written only when XFER[1] bit is 0.) 16bit~31bit DATA stored in 32 bits width fifo. If VDW select 16bit data, this bit is valid only when HWT select 0.Because if HWT is 1, every fifo unit contain two 16bit data and 32 bit space is full, it is impossible to choose justified mode. 0:right justified 1:left justified

Bit	Attr	Reset Value	Description
11	RW	0x0	FBM First Bit Mode (Can be written only when XFER[1] bit is 0.) 0:MSB 1:LSB
10:9	RW	0x0	IBM I2S bus mode (Can be written only when XFER[1] bit is 0.) 0:I2S normal 1:I2S Left justified 2:I2S Right justified 3:reserved
8:7	RW	0x0	PBM PCM bus mode (Can be written only when XFER[1] bit is 0.) 0:PCM no delay mode 1:PCM delay 1 mode 2:PCM delay 2 mode 3:PCM delay 3 mode
6	RO	0x0	reserved
5	RW	0x0	TFS Transfer format select (Can be written only when XFER[1] bit is 0.) 0:i2s 1:pcm
4:0	RW	0x0f	VDW Valid Data width (Can be written only when XFER[1] bit is 0.) 0~14:reserved 15:16bit 16:17bit 17:18bit 18:19bit …… 28:29bit 29:30bit 30:31bit 31:32bit

I2Sx_CKR
Address: Operational Base + offset (0x0008)
clock generation register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27	RW	0x0	MSS Master/slave mode select (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:master mode(sclk output) 1:slave mode(sclk input)
26	RW	0x0	CKP Sclk polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0: sample data at posedge sclk and drive data at negedge sclk 1: sample data at negedge sclk and drive data at posedge sclk
25	RW	0x0	RLP Receive lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)

Bit	Attr	Reset Value	Description
24	RW	0x0	TLP Transmit lrck polarity (Can be written only when XFER[1] or XFER[0] bit is 0.) 0:normal polartiy (I2S normal: low for left channel, high for right channel I2S left/right just: high for left channel, low for right channel PCM start signal:high valid) 1:oppsite polarity (I2S normal: high for left channel, low for right channel I2S left/right just: low for left channel, high for right channel PCM start signal:low valid)
23:16	RW	0x07	MDIV mclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Serial Clock Divider = Fmclk / Ftxsclk-1.(mclk frequecy / txsclk frequecy-1) 0 :Fmclk=Ftxsclk; 1 :Fmclk=2*Ftxsclk; 2,3 :Fmclk=4*Ftxsclk; 4,5 :Fmclk=6*Ftxsclk; …… 60,61:Fmclk=62*Ftxsclk; 62,63:Fmclk=64*Ftxsclk; …… 252,253:Fmclk=254*Ftxsclk; 254,255:Fmclk=256*Ftxsclk;

Bit	Attr	Reset Value	Description
15:8	RW	0x1f	RSD Receive sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Receive sclk divider= Fsclk/Frxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… 253: 254fs 254: 255fs 255: 256fs
7:0	RW	0x1f	TSD Transmit sclk divider (Can be written only when XFER[1] or XFER[0] bit is 0.) Transmit sclk divider=Ftxsclk/Ftxlrck 0~30:reserved 31: 32fs 32: 33fs 33: 34fs 34: 35fs …… 253: 254fs 254: 255fs 255: 256fs

I2Sx_FIFOLR

Address: Operational Base + offset (0x000c)

FIFO level register

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RO	0x00	RFL Receive FIFO Level Contains the number of valid data entries in the receive FIFO.
23:6	RO	0x0	reserved
5:0	RO	0x00	TFL Transmit FIFO Level Contains the number of valid data entries in the transmit FIFO0.

I2Sx_DMACR
Address: Operational Base + offset (0x0010)
DMA control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	RDE Receive DMA Enable 0 : Receive DMA disabled 1 : Receive DMA enabled
23:21	RO	0x0	reserved
20:16	RW	0x1f	RDL Receive Data Level This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1.
15:9	RO	0x0	reserved
8	RW	0x0	TDE Transmit DMA Enable 0 : Transmit DMA disabled 1 : Transmit DMA enabled
7:5	RO	0x0	reserved
4:0	RW	0x00	TDL Transmit Data Level This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TXFIFO(TXFIFO0 if CSR=00;TXFIFO1 if CSR=01,TXFIFO2 if CSR=10,TXFIFO3 if CSR=11)is equal to or below this field value.

I2Sx_INTCR
Address: Operational Base + offset (0x0014)
interrupt control register

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:20	RW	0x00	RFT Receive FIFO Threshold When the number of receive FIFO entries is more than or equal to this threshold plus 1, the receive FIFO full interrupt is triggered.
19	RO	0x0	reserved
18	WO	0x0	RXOIC RX overrun interrupt clear Write 1 to clear RX overrun interrupt.
17	RW	0x0	RXOIE RX overrun interrupt enable 0:disable 1:enable
16	RW	0x0	RXFIE RX full interrupt enable 0:disable 1:enable
15:9	RO	0x0	reserved
8:4	RW	0x00	TFT Transmit FIFO Threshold When the number of transmit FIFO (TXFIFO0 if CSR=00; TXFIFO1 if CSR=01, TXFIFO2 if CSR=10, TXFIFO3 if CSR=11) entries is less than or equal to this threshold, the transmit FIFO empty interrupt is triggered.
3	RO	0x0	reserved
2	WO	0x0	TXUIC TX underrun interrupt clear Write 1 to clear TX underrun interrupt.
1	RW	0x0	TXUIE TX underrun interrupt enable 0:disable 1:enable
0	RW	0x0	TXEIE TX empty interrupt enable 0:disable 1:enable

I2Sx_INTSR
Address: Operational Base + offset (0x0018)
interrupt status register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:18	RO	0x0	reserved
17	RO	0x0	RXOI RX overrun interrupt 0:inactive 1:active
16	RO	0x0	RXFI RX full interrupt 0:inactive 1:active
15:2	RO	0x0	reserved
1	RO	0x0	TXUI TX underrun interrupt 0:inactive 1:active
0	RO	0x0	TXEI TX empty interrupt 0:inactive 1:active

I2Sx_XFER
Address: Operational Base + offset (0x001c)
Transfer Start Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXS RX Transfer start bit 0:stop RX transfer. 1:start RX transfer
0	RW	0x0	TXS TX Transfer start bit 0:stop TX transfer. 1:start TX transfer

I2Sx_CLR
Address: Operational Base + offset (0x0020)
SCLK domain logic clear Register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	RXC RX logic clear This is a self cleard bit. Write 1 to clear all receive logic.

Bit	Attr	Reset Value	Description
0	RW	0x0	TXC TX logic clear This is a self cleard bit. Write 1 to clear all transmit logic.

I2Sx_TXDR
Address: Operational Base + offset (0x0400~0x7FC)
Transimt FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	TXDR Transimt FIFO Data Register When it is written to, data are moved into the transmit FIFO.

I2Sx_RXDR
Address: Operational Base + offset (0x0800~0xBFC)
Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXDR Receive FIFO Data Register When the register is read, data in the receive FIFO is accessed.

29.5 Timing Diagram

29.5.1 Master mode

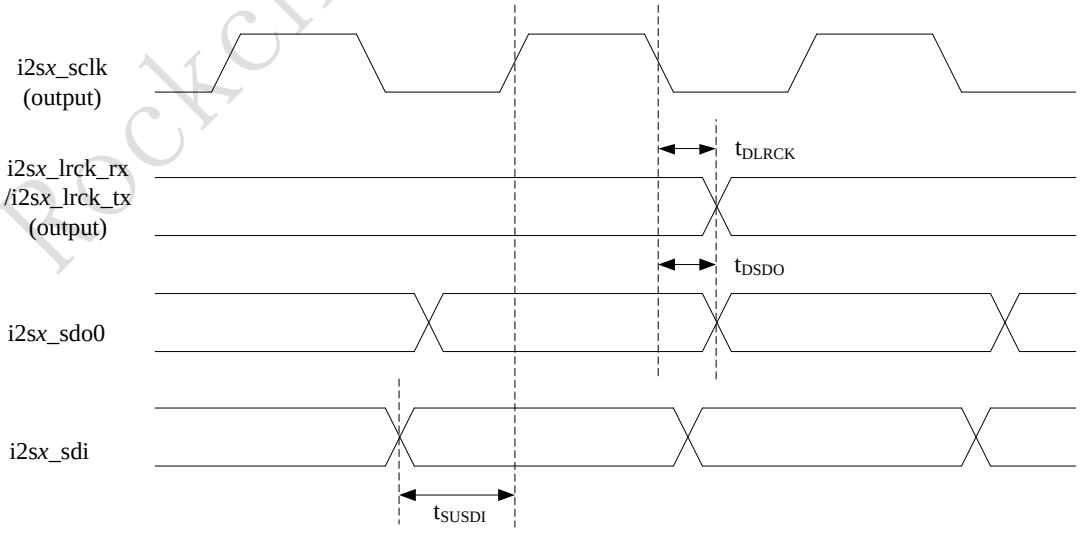


Fig. 29-11Master mode timing diagram

Table 29-1Meaning of the parameter in Fig. 29-11

Parameter	Description	min	typ	max	unit
t_{DLRCK}	i2s1_lrck_rx/i2s1_lrck_tx propagation delay from i2s1_sclk falling edge	1.17	1.63	2.18	ns
t_{DSDO}	i2s1_sdo propagation delay from i2s1_sclk falling edge	1.23	1.73	2.37	ns
t_{SUSDI}	i2s1_sdi setup time to i2s1_sclk rising edge	5.51	7.81	11.03	ns

x=1

Parameter	Description	min	typ	max	unit
t_{DLRCK}	i2s2_lrck_rx/i2s2_lrck_tx propagation delay from i2s1_sclk falling edge	1.02	1.35	1.68	ns
t_{DSDO}	i2s2_sdo propagation delay from i2s2_sclk falling edge	0.97	1.26	1.57	ns
t_{SUSDI}	i2s2_sdi setup time to i2s2_sclk rising edge	5.01	7.37	10.48	ns

x=2

29.5.2 Slave mode

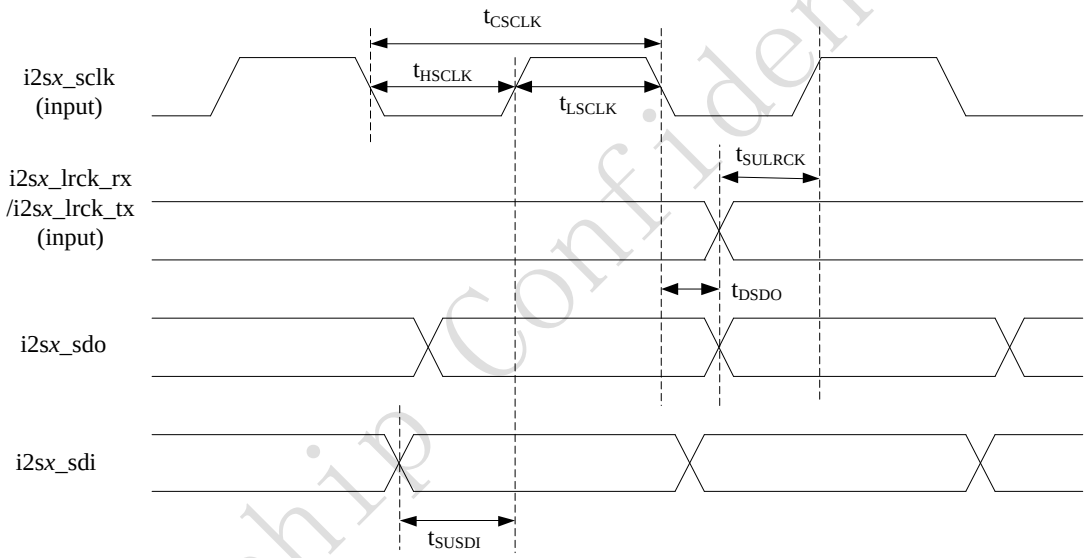


Fig. 29-12Slave mode timing diagram

Table 29-2Meaning of the parameter in Fig. 29-12

Parameter	Description	min	typ	max	unit
t_{CSCLK}	i2s1_sclk cycle time (cannot be less than 40ns)	-	-	-	ns
t_{HSCLK}	i2s1_sclk pulse width high (cannot be less than 20ns)	-	-	-	ns
t_{LSCLK}	i2s1_sclk pulse width low (cannot be less than 20ns)	-	-	-	ns
t_{SULRCK}	i2s1_lrck_rx/i2s1_lrck_tx setup time to i2s1_sclk falling edge	0.86	1.18	1.15	ns
t_{DSDO}	i2s1_sdo propagation delay from i2s1_sclk falling edge	4.97	7.24	9.91	ns
t_{SUSDI}	i2s1_sdi setup time to i2s1_sclk rising edge	0.69	0.73	0.67	ns

x=1

Parameter	Description	min	typ	max	unit

t_{CSCLK}	i2s2_sclk cycle time (cannot be less than 40ns)	-	-	-	ns
t_{HSCLK}	i2s2_sclk pulse width high (cannot be less than 20ns)	-	-	-	ns
t_{LSCLK}	i2s2_sclk pulse width low (cannot be less than 20ns)	-	-	-	ns
t_{SULRCK}	i2s2_lrck_rx/i2s2_lrck_tx setup time to i2s2_sclk falling edge	0.84	1.14	1.11	ns
t_{DSDO}	i2s2_sdo propagation delay from i2s2_sclk falling edge	4.55	6.57	9.00	ns
t_{SUSDI}	i2s2_sdi setup time to i2s2_sclk rising edge	0.72	0.98	0.84	ns

x=2

29.6 Interface description

Module Pin	Direction	Pad Name	IOMUX Setting
I2S1			
i2s1_clk	O	GPIO0_C[0]	GRF_GPIO0C_IOMUX[0]=1'b1
i2s1_sclk	I/O	GPIO0_C[1]	GRF_GPIO0C_IOMUX[2]=1'b1
i2s1_lrck_rx	I/O	GPIO0_C[2]	GRF_GPIO0C_IOMUX[4]=1'b1
i2s1_lrck_tx	I/O	GPIO0_C[3]	GRF_GPIO0C_IOMUX[6]=1'b1
i2s1_sdi	I	GPIO0_C[4]	GRF_GPIO0C_IOMUX[8]=1'b1
i2s1_sdo	O	GPIO0_C[5]	GRF_GPIO0C_IOMUX[10]=1'b1
I2S2			
i2s2_clk	O	GPIO0_D[0]	GRF_GPIO0D_IOMUX[0]=1'b1
i2s2_sclk	I/O	GPIO0_D[1]	GRF_GPIO0D_IOMUX[2]=1'b1
i2s2_lrck_rx	I/O	GPIO0_D[2]	GRF_GPIO0D_IOMUX[4]=1'b1
i2s2_lrck_tx	I/O	GPIO0_D[3]	GRF_GPIO0D_IOMUX[6]=1'b1
i2s2_sdi	I	GPIO0_D[4]	GRF_GPIO0D_IOMUX[8]=1'b1
i2s2_sdo	O	GPIO0_D[5]	GRF_GPIO0D_IOMUX[10]=1'b1

29.7 Application Notes

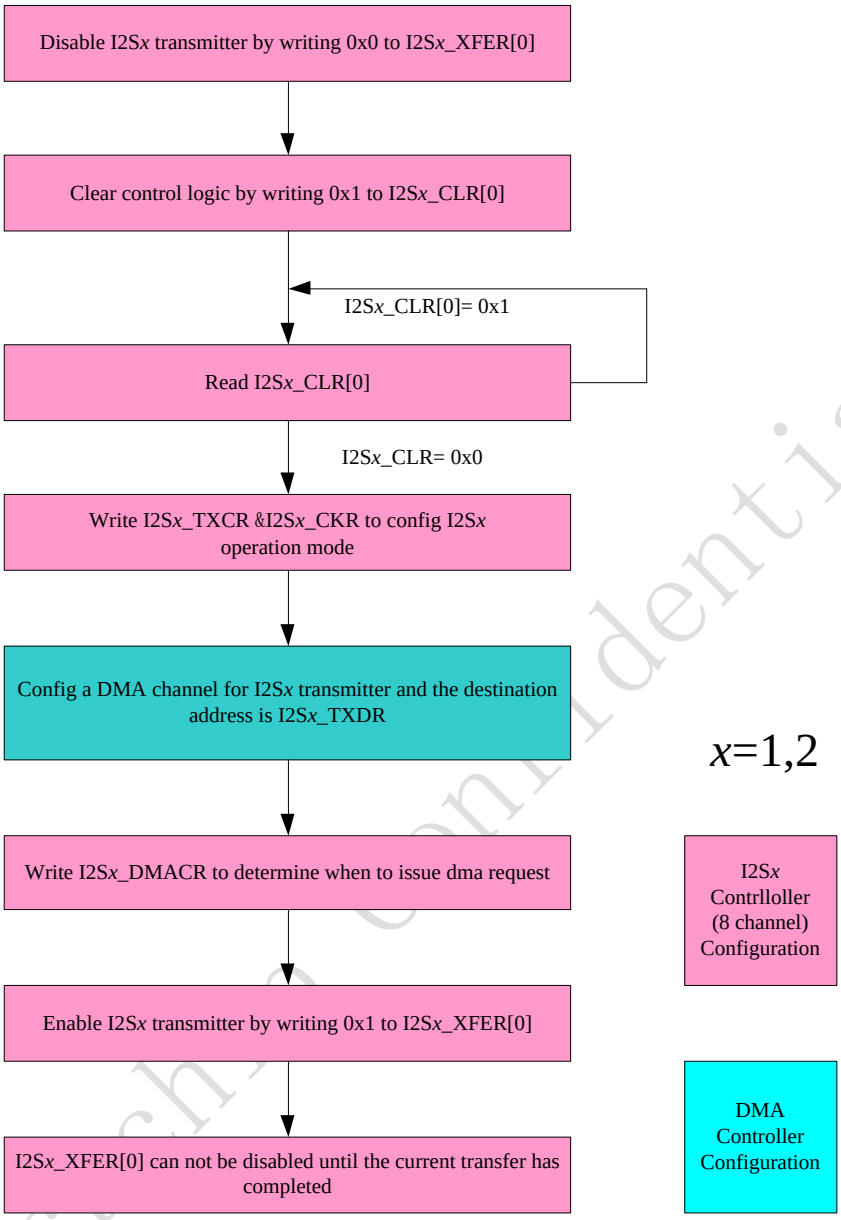


Fig. 29-13I2S/PCM1/2 controller transmit operation flow chart

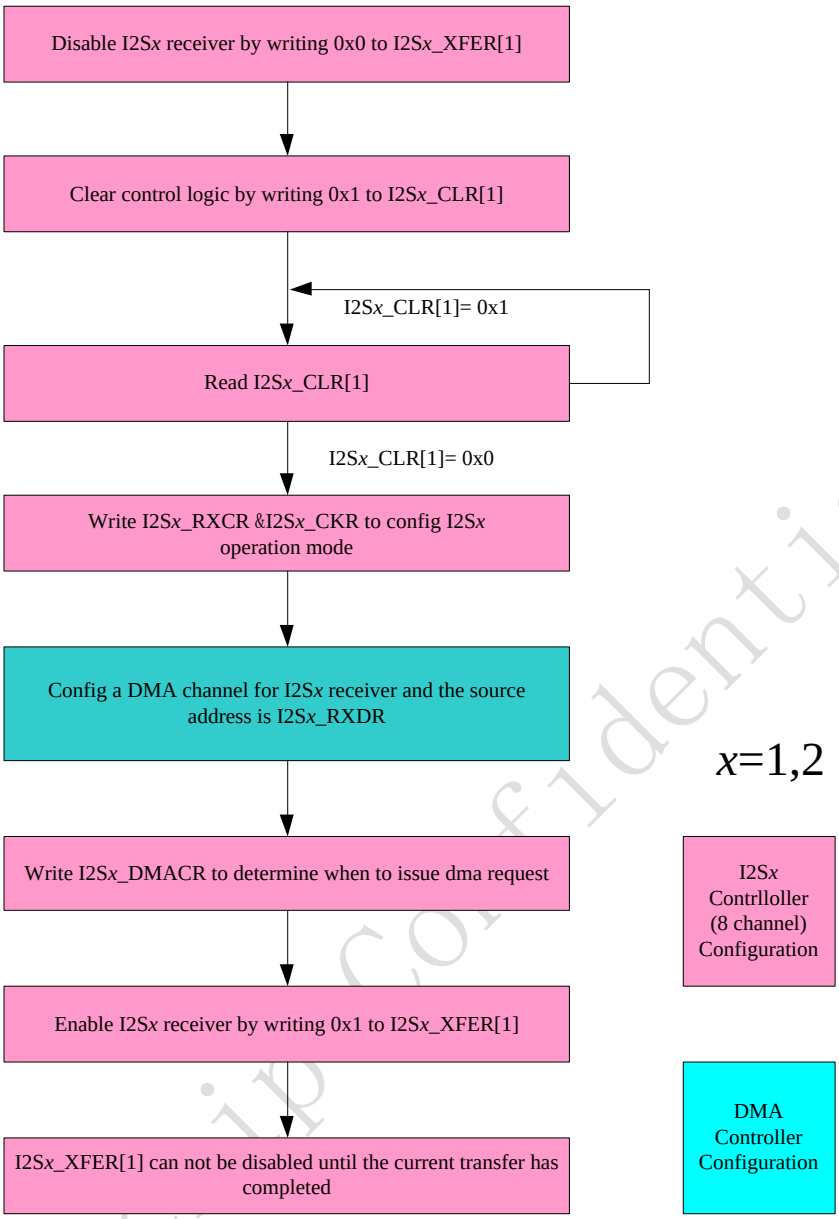


Fig. 29-14I2S/PCM1/2 controller receive operation flow chart

Chapter 30 SPDIF transmitter

30.1 Overview

The SPDIF transmitter is a self-clocking, serial, un-directional interface for the interconnection of digital audio equipment for consumer and professional applications, using linear PCM coded audio samples.

It provides the basic structure of the interface. Separate documents define items specific to particular applications.

When used in a professional application, the interface is primarily intended to carry monophonic or stereophonic programmes, at a 48 kHz sampling frequency and with a resolution of up to 24bits per sample; it may alternatively be used to carry signals sampled at 32 kHz or 44.1 kHz.

When used in a consumer application, the interface is primarily intended to carry stereophonic programmes, with a resolution of up to 20 bits per sample, an extension to 24 bits per sample being possible.

When used for other purposes, the interface is primarily intended to carry audio data coded other than as linear PCM coded audio samples. Provision is also made to allow the interface to carry data related to computer software or signals coded using non-linear PCM. The format specification for these applications is not part of this standard.

In all cases, the clock references and auxiliary information are transmitted along with the programme.

- Support one internal 32-bit wide and 32-location deep sample data buffer
- Support two 16-bit audio data store together in one 32-bit wide location
- Support AHB bus interface
- Support biphase format stereo audio data output
- Support DMA handshake interface and configurable DMA water level
- Support sample data buffer empty and block terminate interrupt
- Support combine interrupt output
- Support 16 to 31 bit audio data left or right justified in 32-bit wide sample data buffer
- Support 48, 44.1, 32kHz sample rate
- Support 16, 20, 24 bits audio data transfer

30.2 Block Diagram

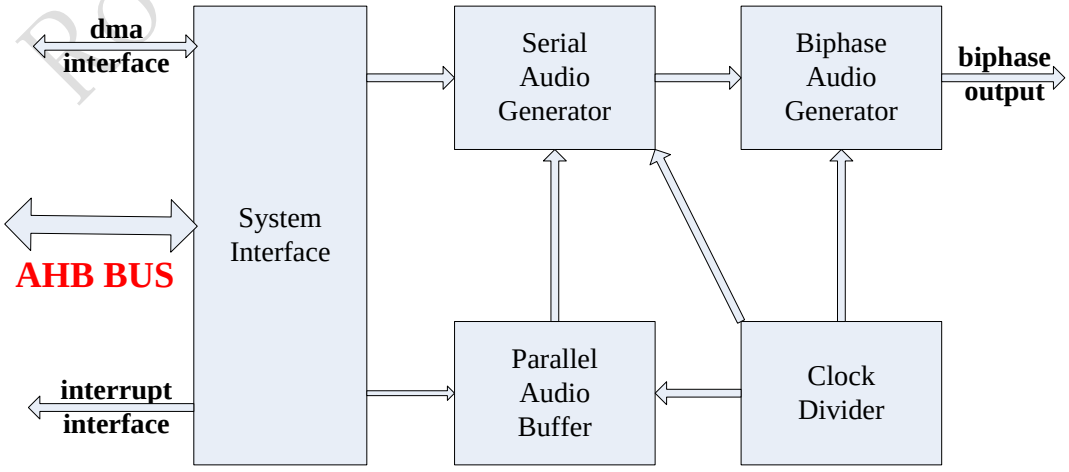


Fig. 30-1SPDIF transmitter Block Diagram

System Interface

The system interface implements the AHB slave operation. It contains not only control registers of transmitters and receiver inside but also interrupt and DMA handshake interface.

Clock Divider

The Clock Divider implements clock generation function. The input source clock to the module is MCLK, and by the divider of the module, the clock divider generates work clock for digital audio data transformation.

Parallel Audio Buffer

The Parallel Audio Buffer is the buffer to store transmitted audio data. The size of the FIFO is 32bits x 32.

Serial Audio Converter

The Serial Audio Converter reads parallel audio data from the Parallel Audio Buffer and converts it to serial audio data.

Biphase Audio Generator

The Biphase Audio Generatorreads serial audio data from the Serial Audio Converter and generates biphase audio data based on IEC-60958 standard.

30.3 Function description

30.3.1 Frame Format

A frame is uniquely composed of two sub-frames. For linear coded audio applications, the rate of transmission of frames corresponds exactly to the source sampling frequency.

In the 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive sub-frames. The first sub-frame(left channel in stereophonic operation and primary channel in monophonic operation) normally use preamble M. However, the preamble is changed to preamble B once every 192 frame to identify the start of the block structure used to organize the channel status information. The second sub-frame (right in stereophonic operation and secondary channel in monophonic operation) always use preamble W.

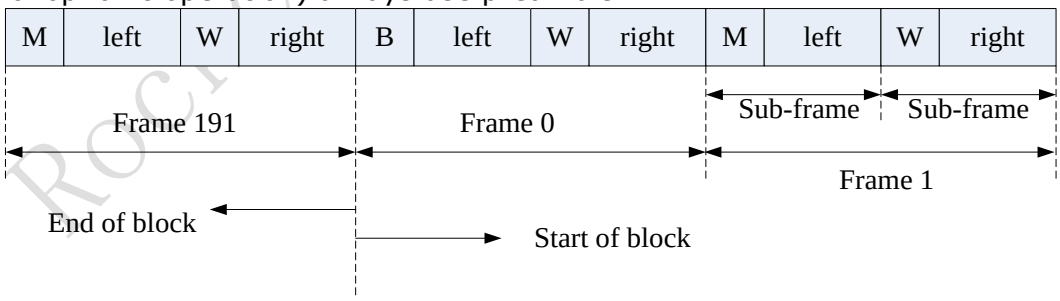


Fig. 30-2SPDIF Frame Format

In the single channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried only in the first sub-frame and may be duplicated in the second sub-frame. If the second sub-frame is not carrying duplicate data, then time slot 28 (validity flag) shall be set to logical '1' (not valid).

30.3.2 Sub-frame Format

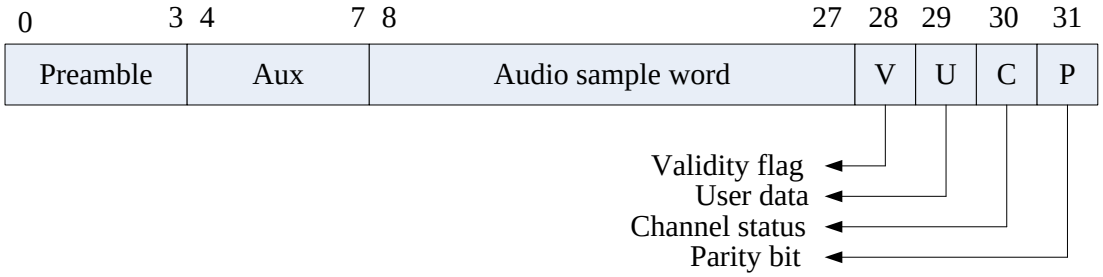


Fig. 30-3SPDIF Sub-frame Format

Each sub-frame is divided into 32 time slot, numbered from 0 to 31. Time slot 0 to 3 carries one of the three permitted preambles. Time slots 4 to 27 carry the audio sample word in linear 2's complement representation. The MSB is carried by time slot 27. When a 24-bit coding range is used, the LSB is in time slot 4. When a 20-bit coding range is used, time slot 8 to 27 carry the audio sample word with the LSB in time slot 8.Time slot 4 to 7 may be used for other application. Under these circumstances, the bits in the time slot 4 to 7 are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 24 or 20),the unused LSBs are set to a logical '0'. For a non-linear PCM audio application or a data application the main data field may carry any other information. Time slot 28 carries the validity flag associated with the main data field. Time slot 29 carries 1 bit of the user data associated with the audio channel transmitted in the same sub-frame. Time slot 30 carries one bit of the channel status words associated with the main data field channel transmitted in the same sub-frame. Time slot 31 carries a parity bit such that time slots 4 to 31 inclusive carries an even number of ones and an even number of zeros.

30.3.3 Channel Coding

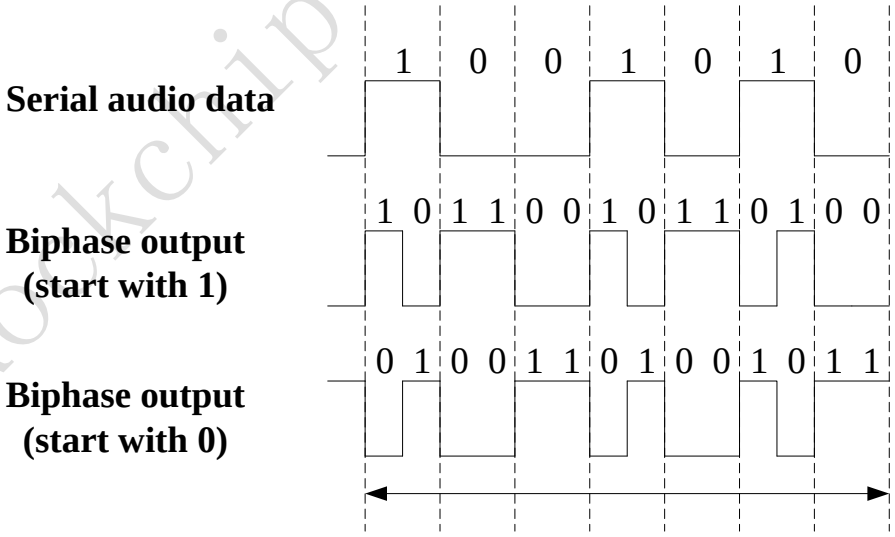


Fig. 30-4SPDIF Channel Coding

To minimize the direct current component on the transmission line, to facilitate clock recovery from the data stream and to make the interface insensitive to the polarity of connections, time slots 4 to 31 are encoded in biphase-mark.

Each bit to be transmitted is represented by a symbol comprising two consecutive binary states. The first state of a symbol is always different from the

second state of the previous symbol. The second state of the symbol is identical to the first if the bit to be transmitted is logical '0'.However, it is different from the first if the bit is logical '1'

30.3.4 Preamble

Preambles are specific patterns providing synchronization and identification of the sub-frames and blocks.

To achieve synchronization within one sampling period and to make this process completely reliable, these patterns violate the biphase-mark code rules, thereby avoiding the possibility of data imitating the preambles.

A set of three preambles is used. These preambles are transmitted in the time allocated to four time slots (time slots 0 to 3) and are represented by eight successive states. The first state of the preamble is always different from the second state of the previous symbol.

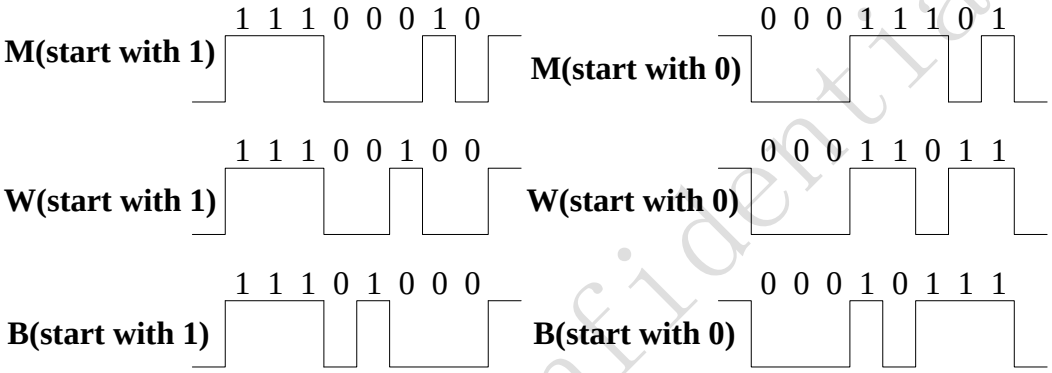


Fig. 30-5SPDIF Preamble

Like biphase code, these preambles are dc free and provide clock recovery. They differ in at least two states from any valid biphase sequence.

30.4 Register description

30.4.1 Register Summary

Name	Offset	Size	Reset value	Description
SPDIF_CFGR	0x00	W	0x0	Transfer Configuration Register
SPDIF_SDBLR	0x04	W	0x0	Sample Date Buffer Level Register
SPDIF_DMACR	0x08	W	0x0	DMA Control Register
SPDIF_INTCR	0x0C	W	0x0	Interrupt Control Register
SPDIF_INTSR	0x10	W	0x0	Interrupt Status Register
SPDIF_XFER	0x18	W	0x0	Transfer Start Register
SPDIF_SMPDR	0x20	W	0x0	Sample Data Register
SPDIF_VLDFR	0x60~0x8C	W	0x0	Validity Flag Register
SPDIF_USRDR	0x90~0xBC	W	0x0	User Data Register
SPDIF_CHNSR	0xC0~0xEC	W	0x0	Channel Status Register

Notes:
Size: B – Byte (8 bits) access, HW – Half WORD (16 bits) access, W –WORD (32 bits) access

30.4.2 Detail Register Description

SPDIF_CFGR

Address:operational base+offset(0x00)

Transfer Configuration Register

Bit	Attr	Reset Value	Description
31:24	-	-	Reserved.
23:16	RW	0x0	Fmclk/Fsdo This parameter can be caculated by Fmclk/(Fs*128). Fs=the sample frequency be wanted
15:8	-	-	Reserved.
7	W	0x0	Write 1 to clear mclk domain logic. Read return zero.
6	RW	0x0	Channel status enable
5	RW	0x0	User data enable
4	RW	0x0	Validity flag enable
3	RW	0x0	Apb valid audio data justified 0:Right justified 1:Left justified
2	RW	0x0	Halfword word transform enable 0:disable 1:enable
1:0	RW	0x0	Valid data width 00：16bit 01：20bit 10：24bit 11：reserved

SPDIF_SDBLR

Address:operational base+offset(0x04)

Sample Date Buffer Level Register

Bit	Attr	Reset Value	Description
31:6	-	-	Reserved.
5:0	R	0x0	Sample Date Buffer Level Register. Contains the number of valid data entries in the sample data buffer.

SPDIF_DMACR

Address:operational base+offset(0x08)

DMA Control Register

Bit	Attr	Reset Value	Description
31:6	-	-	Reserved.
5	RW	0x0	Transmit DMA Enable. 0 = Transmit DMA disabled 1 = Transmit DMA enabled
4:0	RW	0x0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the Sample Date Buffer is equal to or below this field value

SPDIF_INTCR

Address:operational base+offset(0x0C)
Interrupt Control Register

Bit	Attr	Reset Value	Description
31: 17	-	-	Reserved.
16	W	0x0	Block transfer terminate interrupt clear
15: 10	-	-	Reserved.
9:5	RW	0x0	Sample Date Buffer Threshold for empty interrupt
4	RW	0x0	Sample Date Buffer empty interrupt enable. 0:disable; 1: enable;
3	RW	0x0	Block transfer terminate interrupt enable. 0:disable; 1: enable;
2:0	-	-	Reserved.

SPDIF_INTSR
Address:operational base+offset(0x10)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31: 5	-	-	Reserved.
4	R	0x0	Sample Date Buffer empty interrupt status. 0:inactive; 1: active;
3	R	0x0	Block transfer terminate interrupt status. 0:inactive; 1: active;
2:0	-	-	Reserved.

SPDIF_XFER
Address:operational base+offset(0x18)
Transfer Start Register

Bit	Attr	Reset Value	Description
31: 1	-	-	Reserved.
0	RW	0x0	Transfer Start Register.

SPDIF_SMPDR
Address:operational base+offset(0x20)
Sample Data Register

Bit	Attr	Reset Value	Description
31:0	W	0x0	Sample Data Register.

SPDIF_VLDFR
Address:operational base+offset(0x60~0x8C)
Validity Flag Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	Validity Flag Register.

SPDIF_USRDR
Address:operational base+offset(0x90~0xBC)
User Data Register

Bit	Attr	Reset Value	Description
31:0	RW	0x0	User Data Register.

SPDIF_CHNSR
Address:operational base+offset(0xC0~0xEC)
Channel Status Register.

Bit	Attr	Reset Value	Description
31:0	RW	0x0	Channel Status Register.

30.5 Interface description

Module Pin	Direction	Pad Name	IOMUX Setting
spdif_tx	O	GPIO1_B[2]	GRF_GPIO1B_IOMUX[4]=1

30.6 Application Notes

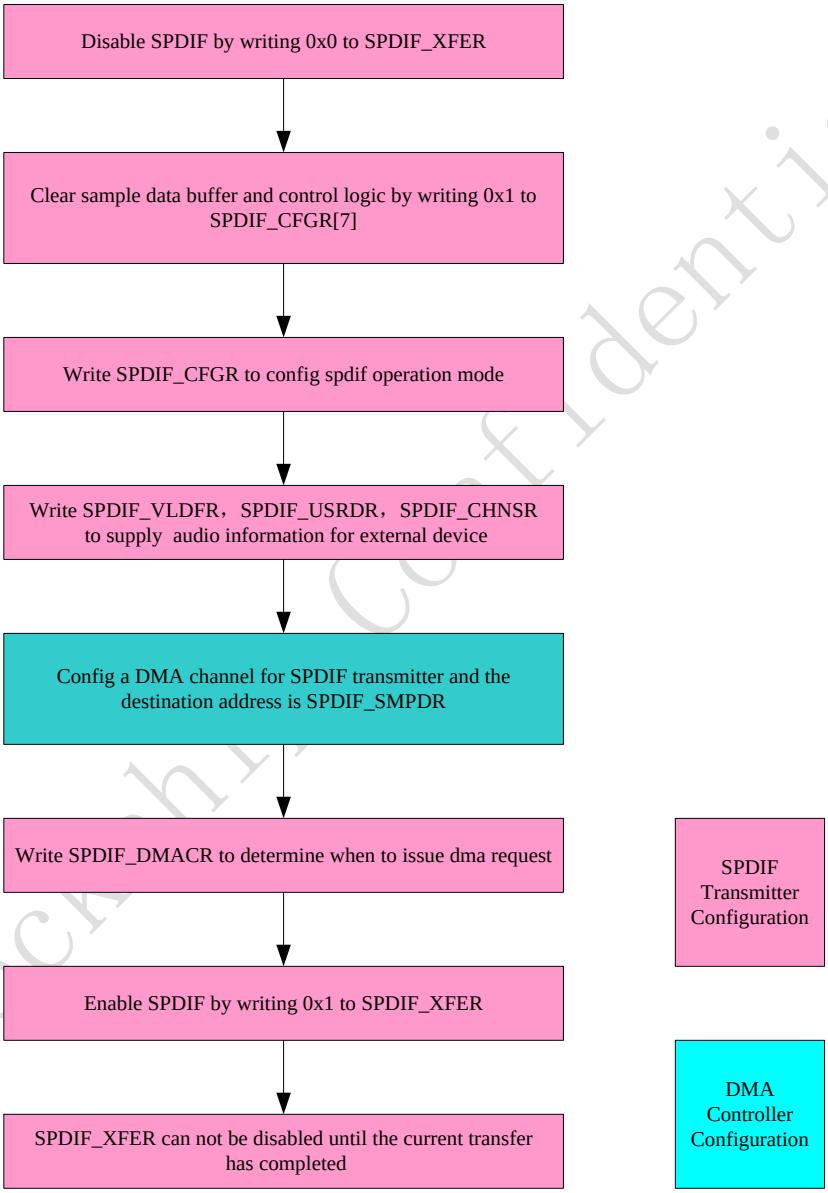


Fig. 30-6 SPDIF transmitter operation flow chart

Chapter 31 SDIO Host Controller

31.1 Overview

The SDMMC Host Controller is designed to support Secure Digital memory (SD mem - version 3.00), Secure Digital I/O(SDIO-version 3.00), Multimedia Cards(MMC-version 4.41).The SDMMC support SD Card(1/4bit), SDIO, MMC(1/4bit).

Feature

- Supports AMBA AHB interface
- Supports DMA controller for data transfers
- Supports interrupt output
- Supports SD version3.0 except SPI mode
- Supports MMC version4.41 except SPI mode
- Supports SDIO version3.0
- Supports programmable baud rate.
- Provides individual clock control to selectively turn ON or OFF clock to a card
- Supports power management and power switch. Provides individual power control to selectively turn ON or OFF power to a card
- Support DDR in 4-bit mode

31.2 Block Diagram

The SD/MMC controller consists of the following main functional blocks, which are illustrated inFig. 17-1.

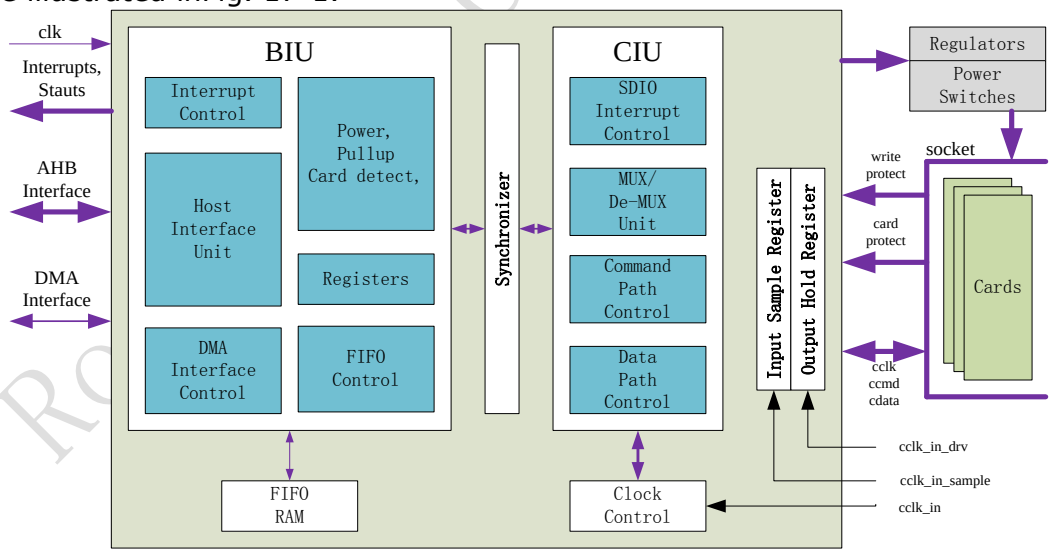


Fig. 31-1SD/MMC Controller Block Diagram

- Bus Interface Unit (BIU) – Provides AMBA AHB and DMA interfaces for register and dataread/writes.
- Card Interface Unit (CIU) – Takes care of the SD_MMC protocols and provides clockmanagement.

31.3 Function description

Ref 17.3

31.4 Register description

Ref 17.4

31.5 Timing Diagram

31.6 Interface description

31.6.1 Card-Detect and Write-Protect Mechanism

Figure 17-6 illustrates how the SD/MMC Host Controller card detection and write-protect signals are connected. Most of the SD_MMC sockets have card-detect pins. When no card is present, card_detect_n is 1 due to the pull-up. When the SD_MMC card is inserted, the card-detect pin is shorted to ground, which makes card_detect_n go to 0. Similarly in SD cards, when the write-protect switch is toward the left, it shorts the write_protect port to ground.

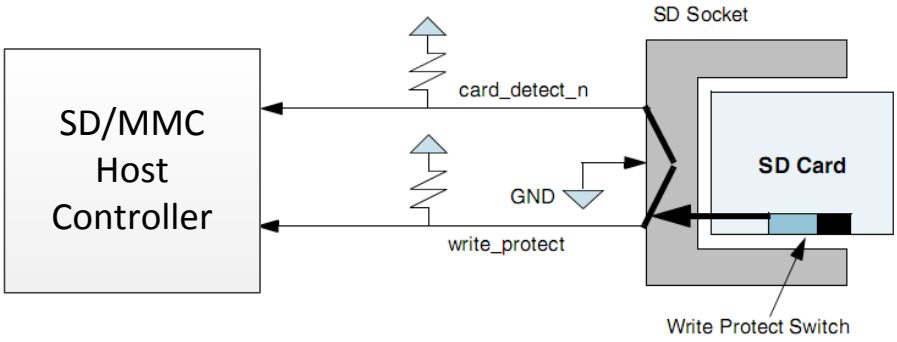


Fig. 31-2 Card-Detect and Write-Protect

31.6.2 SD/MMC Controller Termination Requirement

Fig.17-7 illustrates the SD/MMC Host Controller termination requirements, which is required to pull up ccmd and cdata lines on the SD_MMC bus. The recommended specification for pull-up on the ccmd line (Rcmd) is 4.7K - 100K for MMC, and 10K - 100K for an SD. The recommended pull-up on the cdata line (Rdat) is 50K - 100K.

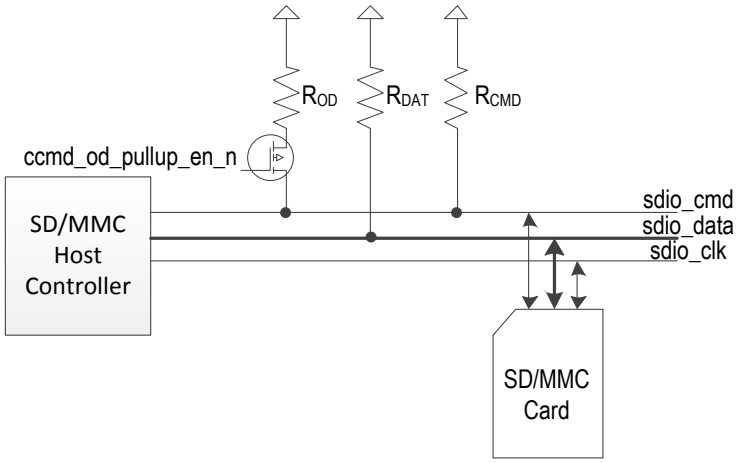


Fig. 31-3 SD/MMC Termination

Rcmd and Rod Calculation

The SD and MMC card enumeration happens at a very low frequency – 100-400KHz. Since the MMC bus is a shared bus between multiple cards, during enumeration open-drive mode is used to avoid bus conflict. Cards that drive 0 win over cards that drive "z." The pull-up in the command line pulls the bus to 1 when all cards drive "z." MMC interrupt mode also uses the pull-up. During normal data transfer, the hose chooses only one card and the card driver switches to push-pull mode.

For example, if enumeration is done at 400KHz and the total bus capacitance is 200 pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz
R = 1/(2.2 * C * 100KHz)
= 1/(2.2 x 200 x 10**-12 x 400 x 10**3)
= 1/(17.6 x 10**-5)
= 5.68K

The Rod and Rcmd should be adjusted in such a way that the effective pull-up is at the maximum 5.68K during enumeration. If there are only a few cards in the bus, a fixed Rcmd resister is sufficient and there is no need for an additional Rod pull-up during enumeration. You should also ensure the effective pull-up will not violate the Iol rating of the drivers.

In SD mode, since each card has a separate bus, the capacitance is less, typically in the order of 20-30pf (host capacitance + card capacitance + trace + socket capacitance). For example, if enumeration is done at 400KHz and the total bus capacitance is 20pf, the pull-up needed during enumeration is:

2.2 RC = rise-time = 1/400KHz
R = 1/(2.2 * C * 100KHz)
= 1/(2.2 x 20 x 10**-12 x 400 x 10**3)
= 1/(1.76 x 10**-5)
= 56.8K

Therefore, a fixed 56.8K permanent Rcmd is sufficient in SD mode to enumerate the cards.

The driver of the SD/MMC Host Controller on the "command" port needs to be only a push-pull driver. During enumeration, the SD/MMC Host Controller emulates an open-drain driver by driving only a 0 or a "z" by controlling the ccmd_out and ccmd_out_en signals.

31.6.3 SD/MMC Controller IOMUX

The SDMMC Host Controller share the pin with GPIO. In default, the pins are used for GPIO, if user want to work in sdmmc function, the user must configure the GRF registers as following table:

Table 31-1 SDMMC IOMUX Settings

Module Pin	Direction	Pad Name	IOMUX Setting
sdio_clkout	O	GPIO3_C[5]	GPIO3C_IOMUX[10]=0x1& GPIO3C_IOMUX[26]=0x1
sdio_cmd	I/O	GPIO3_C[0]	GPIO3C_IOMUX[0]=0x1& GPIO3C_IOMUX[16]=0x1
sdio_data0	I/O	GPIO3_C[1]	GPIO3C_IOMUX[2]=0x1& GPIO3C_IOMUX[18]=0x1
sdio_data1	I/O	GPIO3_C[2]	GPIO3C_IOMUX[4]=0x1& GPIO3C_IOMUX[20]=0x1
sdio_data2	I/O	GPIO3_C[3]	GPIO3C_IOMUX[6]=0x1& GPIO3C_IOMUX[22]=0x1
sdio_data3	I/O	GPIO3_C[4]	GPIO3C_IOMUX[8]=0x1& GPIO3C_IOMUX[24]=0x1
sdio_int_n	I	GPIO3_D[2]	GPIO3D_IOMUX[4]=0x1& GPIO3D_IOMUX[20]=0x1
sdio_detect_n	I	GPIO3_C[6]	GPIO3C_IOMUX[12]=0x1& GPIO3C_IOMUX[28]=0x1
sdio_write_prt	I	GPIO3_C[7]	GPIO3C_IOMUX[14]=0x1& GPIO3C_IOMUX[30]=0x1
sdio_backend	O	GPIO3_D[1]	GPIO3D_IOMUX[2]=0x1& GPIO3D_IOMUX[18]=0x1
sdio_pwr_en	O	GPIO3_D[0]	GPIO3D_IOMUX[0]=0x1& GPIO3D_IOMUX[16]=0x1

*Notes: Direction: **I**– Input, **O**– Output, **I/O**– Input/Output*

31.7 Application Notes

Ref to 17.7

Chapter 32 MAC Ethernet Interface

32.1 Overview

The VMAC Ethernet Controller providers a complete Ethernet interface from processor to a Reduced Media Independent Interface(RMII) compliant Ethernet PHY.

The VMAC includes a DMAC controller.The DMAC controller efficiently moves packet data from microprocessor's RAM,format the data for an IEEE 802.3 compliant packet and transmit the data to an Ethernet Physical Interface(PHY).It also efficiently moves packet data from RXFIFO to microprocessor's RAM.

32.1.1 Features

- IEEE 802.3u compliant Ethernet Media Access Controller
- 10Mbps and 100Mbps compatible
- Automatic retry and automatic collision frame deletion
- Reduced Media Independent Interface （RMII） for PHY connection
- Mangement Interface (MDIO) state machine for easy real-time communication with the PHY.
- Full Duplex Support
- Pause full-duplex flow-control support
- Address filtering –Broadcast/Multicast/Logic/Physical
- Complete DMA buffer management controller for minimal processor overhead
- Wake-On-LAN low-power mode support
- AHB interface to any CPU or memory

32.2 Block Diagram

32.2.1 Architecture

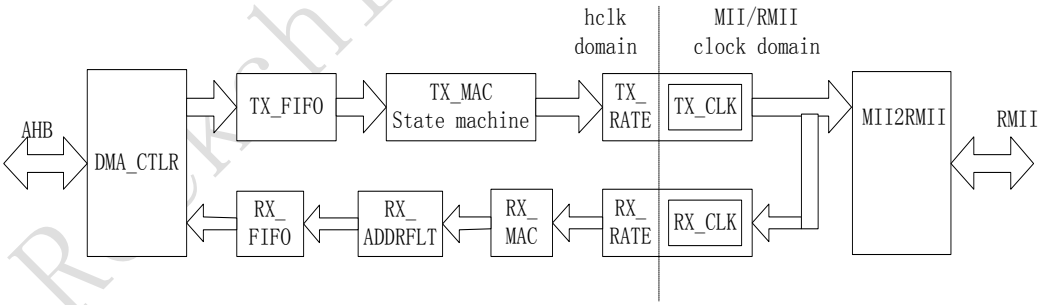


Fig. 32-1 VMAC architecture

The VMAC is broken up into nine separate functional units.These nine blocks are interconnected in the VMAC module.The block diagram shows the general flow of data and control signals between these blocks.

The DMA_CTLR controller moves data between system memory and the respective TX or RX pipelines and manges the buffer descriptors.A single state machine handles the Buffer Descriptors as well as both the RX and TX DMAs. Transmit and Receive DMAs may be interleaved as well as polling of the buffer descriptor rings.The DMA controller also includes all of the configuration,control and status registers of the VMAC as well as the MDIO state machine for communicating to the PHY.

The transmit FIFO and its controller are integrated in the TX_FIFO block.The FIFO is typically built of a 512x8bit dual-port RAM.

The TX_MAC provides all of the logic necessary to build and transmit a frame that meets the IEEE 802.3 Ethernet LAN standard.When a start of frame is detected on the data/status bus, then the TX_MAC block starts to transmit preamble data. TX_MAC delays the transmission if the receive path is active in Half-Duplex mode and ensures that the interframe period is met. In Full-Duplex mode only the interframe period is observed and no deferring process takes place.

The TX_RATE block transfers the raw packet data from the HCLK to the CLK_TX clock domains.In addition to its main function,the TX_RATE block monitors the collision signal COL from the PHY and the preamble field from TX_RATE.In case of a collision,the TX_RATE jams the data on TXD bus and notifies TX_MAC about the collision.

The RX_FIFO block is composed with a 1024x9 bit dual-port ram.

The RX_ADDRFLT block determines if the destination address matches under any of the currently-active addressing modes. While the Destination Address fileld is being verified, the incoming bytes are stored in a small FIFO. If the Destination address does not match, then the RX_ADDRFLT resets its FIFO and disregards incoming data.

The RX_MAC block provides all of the logic necessary to meet the IEEE 802.3 Ethernet LAN standard for frame reception. The RX_MAC detects the SFD pattern, verifies FCS field,senses framing errors (odd number of nibbles) and monitors the RX_ER signal, which indicates any other errors received from an external PHY.

The RX_RATE block synchronizes signals from MII CLK_RX domain to hclk domain.

The MII2RMII block transfers MII signals to RMII signals.

32.2.2 Frame Structure

Data frames transmitted shall have the frame format shown in Fig 32-2.

<inter-frame><preamble><sfd><data><efd>

Fig. 32-2 VMAC Frame structure

The preamble <preamble> begins a frame transmission. The bit value of the preamble field consist of 7 octets with the folling bit values:
10101010 10101010 10101010 10101010 10101010 10101010 10101010
The SFD (start frame delimiter) <sfd> indicates the start of a frame and follows the preamble.The bit value is 10101011.

The data in a well formed frame shall consist of N octets data.

32.2.3 RMII Interface timing diagram

1.Transmission diagram

Fig. 32-3 shows the 100Mb/s Transmission diagram. The REF_CLK frequency is 50MHz in RMII interface.

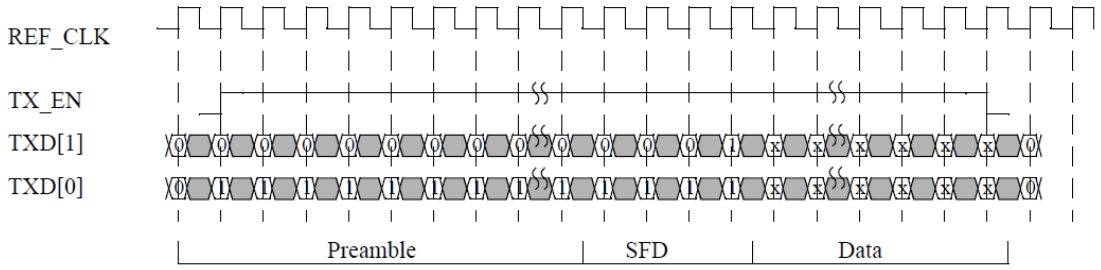


Fig. 32-3RMII transmission in 100Mb/s mode

In 10Mb/s mode, as the REF_CLK frequency is 10 times as the data rate, the value on TXD[1:0] shall be valid such that TXD[1:0] may be sampled every 10th cycle,regard-less of the starting cycle within the gRup and yield the correct frame data.

2. Receiption diagram

Fig.32-4shows the 100Mb/s reception diagram.The REF_CLK frequency is 50MHz in RMII interface.

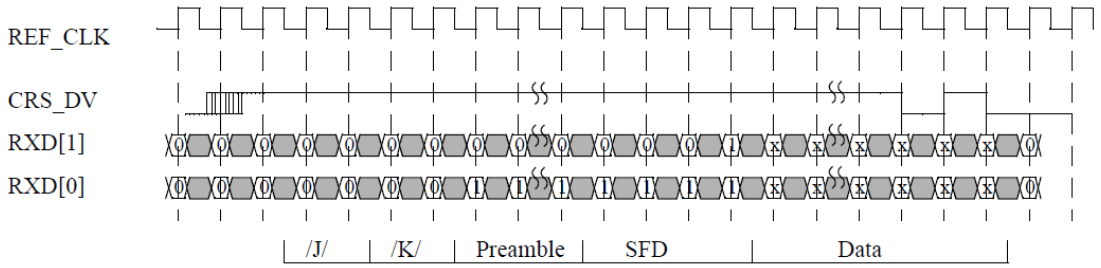


Fig. 32-4 RMII reception with no errors in 100Mb/s mode

In 10Mb/s mode, as the REF_CLK frequency is 10 times the data rate, the value on RXD[1:0] shall be valid such that RXD[1:0] may be sampled every 10th cycle,regardless of the starting cycle within the gRup and yield the correct frame data.

32.2.4 Mangement Interface

The MII management interface provides a simple,tW-wire,serial interface to connect the VMAC and a manged PHY, for the purposes of controlling the PHY and gathering status from the PHY. The management interface consists of a pair of signals that transport the management information acRss the MII bus: MDIO and MDC.

32.3 Register description

32.3.1 Register Summary

Name	Offset	Size	Reset Value	Description
EMAC_ID	0x0000	W	0x00053d02	hardware version
EMAC_STAT	0x0004	W	0x00000000	Interrupt status register
EMAC_ENABLE	0x0008	W	0x00000000	Interrupt enable register
EMAC_CONTROL	0x000c	W	0x00000000	CONTROL Register

Name	Offset	Size	Reset Value	Description
EMAC_POLLRATE	0x0010	W	0x00000000	Poll Rate register
EMAC_RXERR	0x0014	W	0x00000000	Receive Error Counters
EMAC_MISS	0x0018	W	0x00000000	Missed Packet Counter
EMAC_TXRINGPTR	0x001c	W	0x00000000	Transmit Ring Pointer address register
EMAC_RXRINGPTR	0x0020	W	0x00000000	Receive Ring Pointer address register
EMAC_ADDRL	0x0024	W	0x00000000	Ethernet MAC Address , low 32 bits
EMAC_ADDRH	0x0028	W	0x00000000	Ethernet MAC Address, high 16 bits
EMAC_LAFL	0x002c	W	0x00000000	Logical Address filter Register Low
EMAC_LAFH	0x0030	W	0x00000000	Logical Address filter Register High
EMAC_MDIO_DATA	0x0034	W	0x00000000	MDIO access register
EMAC_TXRINGPTR_READ	0x0038	W	0x00000000	Transmit Ring Pointer read-back register
EMAC_RXRINGPTR_READ	0x003c	W	0x00000000	Receive Ring Pointer read-back register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

32.3.2 Detail Register Description

EMAC_ID

Address: Operational Base + offset (0x0000)

hardware version

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RO	0x05	Revision Revision number Revision number (currently Rev 5)
15:14	RO	0x0	reserved
13:8	RO	0x3d	NOTID Ones-Complement of the ID Ones-Complement of the ID, (111101b)
7:6	RO	0x0	reserved
5:0	RO	0x02	ID Identificaiton number Identificaiton number,always 2 for VMAC

EMAC_STAT

Address: Operational Base + offset (0x0004)
Interrupt status register

Bit	Attr	Reset Value	Description
31	RW	0x0	TXPL TXPOLL Writing a one forces a poll of the transmit descriptors. Always write this bit with a one after adding a packet to the transmit BDT. Always read as zero.
30:13	RO	0x0	reserved
12	RW	0x0	MDIO MDIO Complete The register access to the PHY has completed
11	RO	0x0	reserved
10	RW	0x0	RXFL RX OVER FLOW The RXOFLOWERR counter has rolled over
9	RW	0x0	RXFR RXFRAME The RXFRAMEERR counter has rolled over
8	RW	0x0	RXCR RXCRC The RXCRCERR counter has rolled over
7:5	RO	0x0	reserved
4	RW	0x0	MSER MISSERR Missed packet counter has rolled over
3	RW	0x0	TXCH TX Chaining Error A bad conbination of FIRST and LAST bits has been encountedred. The VMAC asserts this error bit and disables the TXRN bit in the CONTROL register.
2	RO	0x0	ERR Error interrupt pending Error interrupt pending . is the logical OR of all of the other error bits in the status register (all interrupts except TXINT , RXINT and MDIO). This is a read-only bit

Bit	Attr	Reset Value	Description
1	RW	0x0	RXINT Receive Interrupt Pending RXINT is set when the VMAC clears the OWN bit of an RX BDT and LAST=1. Note that multiple RX BDTs could be cleared before the RXINT interrupt is serviced. Thus ALL RX BDTs must be processed within the interrupt. This also results in the possibility that there will be a RXINT without any corresponding BDTs with their OWN bits cleared. This is normal and the software driver must handle this case.
0	RW	0x0	TXINT Transmit Interrupt Pending TXINT is set when the OWN bit of a TX BDT is cleared to 0 by the VMAC and LAST=1. Note that multiple TX BDTs could be cleared before the TXINT interrupt is erviced. Tus ALL TX BDTs must be processed within the interrupt . This also results in the possibility that there will be a TXINT without any corresponding BDTs with their OWN bits cleared. This is normal and the software driver must handle this case.

EMAC_ENABLE
Address: Operational Base + offset (0x0008)
Interrupt enable register

Bit	Attr	Reset Value	Description
31	RW	0x0	TXPL TXPOLL TXPOLL
30:13	RO	0x0	reserved
12	RW	0x0	MDIO MDIO Complete enable MDIO Complete enable
11	RO	0x0	reserved
10	RW	0x0	RXFL RXOFLOWERR counter rolled over error enable Field0000 Description
9	RW	0x0	RXFR RXFRAMEERR counter rolled over error enable RXFRAMEERR counter rolled over error enable

Bit	Attr	Reset Value	Description
8	RW	0x0	RXCR RXFRAMEERR counter rolled over error enable RXFRAMEERR counter rolled over error enable
7:5	RO	0x0	reserved
4	RW	0x0	MSER Missed packet counter error enable Missed packet counter error enable
3	RW	0x0	TXCH TX Chaining Error enable TX Chaining Error enable
2	RW	0x0	ERR Error Interrupt Pending Enable Error Interrupt Pending Enable
1	RW	0x0	RXINT Receive interrupt pending enable Receive interrupt pending enable
0	RW	0x0	TXINT Transmit interrupt pending enable Transmit interrupt pending enable

EMAC_CONTROL
Address: Operational Base + offset (0x000c)
CONTROL Register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	RXBDTLEN RXBDTLEN Number of BDTs in the RX Ring. 1-255 allowed.
23:16	RW	0x00	TXBDTLEN TXBDTLEN Number of BDTs in the TX Ring. 1-255 allowed.
15	RW	0x0	DISAD DISADDCRC 1: disable adding the 4byte CRC(FCS) on every packet. Instead, use the ADDCRC bit in the INFO word of the Transmit Buffer descriptor to add the FCS on a packet by packet basis. 0: always add CRC and ignore the ADDCRC bit in the info field

Bit	Attr	Reset Value	Description
14	RW	0x0	DISRT DISRETRY 1: disable retries, tx will be attempted only once.
13	RW	0x0	TEST TEST used for silicon testing -always set to zero
12	RW	0x0	DIS2P DISABLE2PART 1: Disable two part deferral. Disabling 2-part deferral disables a fast Inter-packet Gap time as described in the 802.3 specification. See ANSI/IEEE Std802.3-1993 Edition , 4.2.3.2.1
11	RW	0x0	PROM PROM 0: normal mode 1: promiscuous mode = accepts all packets.
10	RW	0x0	ENFL ENBFULL 1: enable full duplex mode. This bit needs to be set to the corresponding duplex mode of the PHY chip the VMAC is connected to. The duplex mode of the PHY needs to be polled periodically to keep the VMAC duplex setting in line with the PHY.
9	RO	0x0	reserved
8	RW	0x0	DISBC DISBDCST 1: disable receive broadcast packets.
7:5	RO	0x0	reserved
4	RW	0x0	RXRN RXRUN 0: disable receive operation. If a packet is currently being received, it will complete before the receive operation is disabled. Like TXRN, RXRN can be used to safely disable packet reception.

Bit	Attr	Reset Value	Description
3	RW	0x0	TXRN TXRUN 0: disable transmit operation. This bit is only tested when the TXBDT state machine completes the current transmit operation and just before it checks the next BDT for new packet. This bit can be used to safely stop transmitting packets without affecting packet reception. A TX packet that is already in process will complete before TXRN is recongnized.
2:1	RO	0x0	reserved
0	RW	0x0	EN ENABLE 0: stop all activity. 1: enable ethernet traffic. All registers should be initialized before setting this bit. Clearing this bit to zero resets BDT rings to their first BD in the table.

EMAC_POLLRATE
Address: Operational Base + offset (0x0010)
Poll Rate register

Bit	Attr	Reset Value	Description
31:15	RO	0x0	reserved
14:0	RW	0x0000	POLLRATE POLLRATE The value programmed into this register is the number of clocks between polls times 1024. A value of 1 will cause a poll every 1024 clocks. 2=2048 clocks. 0 is not valid. A CPU clock frequency of 100MHz would typically want to program the POLLRATE register with a value of 100 which will cause a poll to occur about once every millisecond (10ns * 1024 * 100 = 1ms)

EMAC_RXERR
Address: Operational Base + offset (0x0014)
Receive Error Counters

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved

Bit	Attr	Reset Value	Description
23:16	RW	0x00	RXOFLOW Overflow Errors Number of receive packets dropped due to FIFO overflows
15:8	RW	0x00	RXFRAM RXFRAME Errors Number of receive packets dropped due to framing errors.
7:0	RW	0x00	RXCRC CRC Errors Number of receive packets dropped due to CRC errors.

EMAC_MISS
Address: Operational Base + offset (0x0018)
Missed Packet Counter

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:0	RW	0x00	MISSCNTR Missed packet counter When the counter counts up to the maximum value, it sets the MISSERR bit in the INTER register. This counter counts the number of packets that were dropped because a BD was not available. This counter is auto-zeroed when read.

EMAC_TXRINGPTR
Address: Operational Base + offset (0x001c)
Transmit Ring Pointer address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	TXRINGPTR Address of the start of the Transmit RING of Buffer Descriptors Address of the start of the Transmit RING of Buffer Descriptors. Must be on 8-byte boundaries. (ID: Bits 2-0 must be zero)

EMAC_RXRINGPTR
Address: Operational Base + offset (0x0020)
Receive Ring Pointer address register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	RXRINGPTR Address of the start of the Receive RING of BDs. Address of the start of the Receive RING of BDs. Must be on 8-byte boundaries (IE : Bits 2-0 must be zero)

EMAC_ADDRL
Address: Operational Base + offset (0x0024)
Ethernet MAC Address , low 32 bits

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	ADDRL Lower 32 bits of the ethernet MAC address Lower 32 bits of the ethernet MAC address. The address is little-endian. Thus the first byte transferred on the Ethernet wire must match bits 7:0. The second byte transferred is in bits 15:8 and so on. Thus for a Physical address transmitted in byte order of: 0x00 - 1st byte transmitted 0x11 - 2nd byte transmitted 0x22 - 3rd byte transmitted 0x33 - 4th byte transmitted 0x44 - 5th byte transmitted 0x55 - 6th byte transmitted The ADDRL register should be programmed with 0x33221100. Bit 0 is the multi-cast / Broadcast bit of the address. Bit 0 MUST be programmed with a zero.

EMAC_ADDRH
Address: Operational Base + offset (0x0028)
Ethernet MAC Address, high 16 bits

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	ADDRH Upper 16 bits of the ethernet MAC address Upper 16 bits of the ethernet MAC address. See ADDRL for more details. In the example given in ADDRL, this register would be programmed with 0x5544.

EMAC_LAFL
Address: Operational Base + offset (0x002c)
Logical Address filter Register Low

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	LAFL Low 32 bits for the Logical Address Filter Low 32 bits for the Logical Address Filter. Each bit corresponds to a hash function of the destination address of a packet. IFf the bit is set to a 1, then the packet is accepted, otherwise it is filtered out.

EMAC_LAFH
Address: Operational Base + offset (0x0030)
Logical Address filter Register High

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	LAFH High 32 bits for the Logical Address Filter High 32 bits for the Logical Address Filter. Each bit corresponds to a hash function of the destination address of a packet. If the bit is set to a 1, then the packet is accepted , otherwise it is filtered out.

EMAC_MDIO_DATA
Address: Operational Base + offset (0x0034)
MDIO access register

Bit	Attr	Reset Value	Description
31:30	RW	0x0	SFD Start of Frame Delimiter Start of Frame Delimiter, must be set to "01"
29:28	RW	0x0	OP Operation Code Operation Code, set to "10" for a read and "01" for a write.

Bit	Attr	Reset Value	Description
27:23	RW	0x00	PHY PHY address(0-31) PHY address(0-31)
22:18	RW	0x00	REG Register to access (0-31) Register to access (0-31)
17:16	RW	0x0	TA Bus Turn-Around, must be set to "10" Bus Turn-Around, must be set to "10"
15:0	RW	0x0000	DATA DATA to be written to or read from the PHY register DATA to be written to or read from the PHY register

EMAC_TXRINGPTR_READ
Address: Operational Base + offset (0x0038)
Transmit Ring Pointer read-back register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	TXRINGPTR Transmit Ring Pointer read-back register This read-only register gives the address of the current Transmit Buffer Descriptor being polled or processed by the VMAC. This allows the software to determine where in the BDT ring the VMAC is currently processing. Note that this is a hardware register and undergoes rapid changes. The value in this register should ONLY be read when TXRN is cleared to zero so that the VMAC will stop processing buffers.

EMAC_RXRINGPTR_READ
Address: Operational Base + offset (0x003c)
Receive Ring Pointer read-back register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	RXRINGPTR Receive Ring Pointer read-back register This read-only register gives the address of the current Receive Buffer Descriptor being polled or processed by the VMAC. This allows the software to determine where in the BDT ring the VMAC is currently processing buffers.

32.4 Timing Diagram

- **Management Timing Diagram**

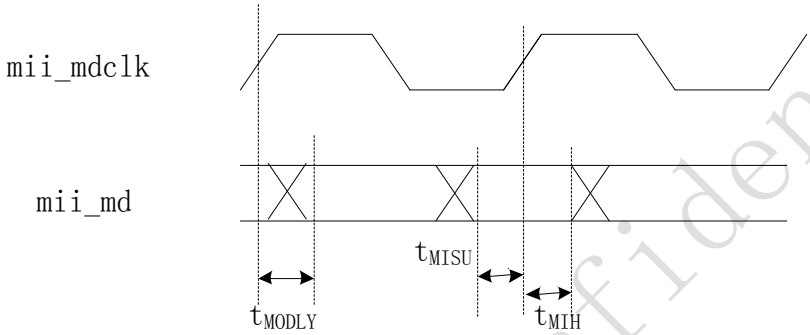


Fig. 32-5Management timing diagram

Table 32-1 Management timing parameters

Parameter	Description	min	typ	max	unit
T_{MDC}	MDC clock period	400	-	-	ns
t_{MODLY}	Mii_md output delay time from mii_mdclk rising edge	261.2			ns
t_{MISU}	Mii_md input setup time from mii_mdclk risiong edge	133.3	-	-	ns
t_{MIH}	Mii_md input hold time from mii_mdclk risiong edge	0	-	-	ns

- **RMII Timing Diagram**

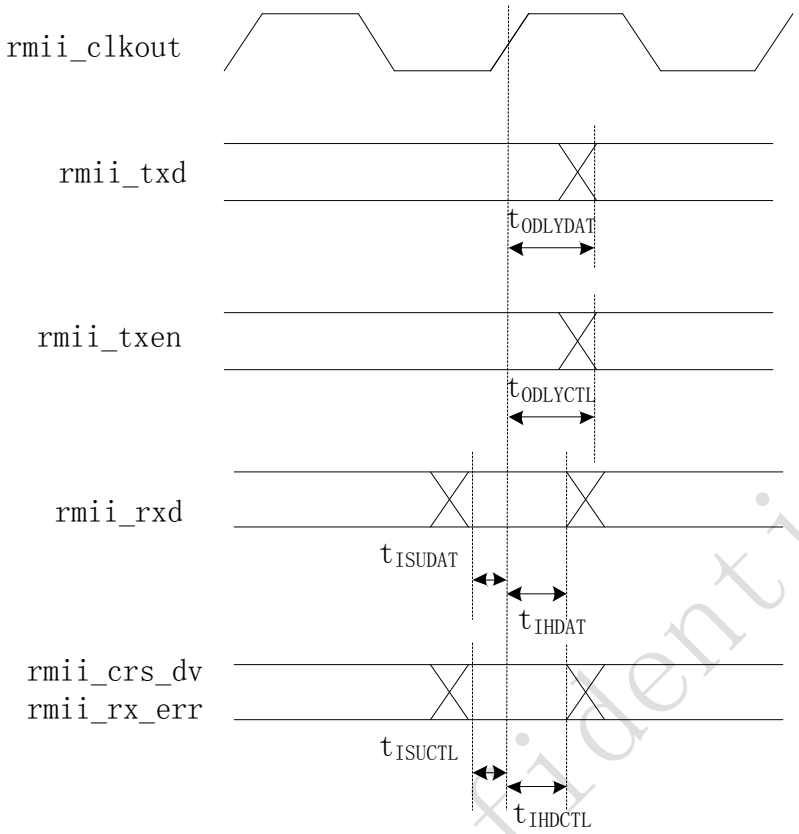


Fig. 32-6 RMII timing diagram

Table 32-2 RMii timing parameters

Parameter	Description	min	typ	max	unit
	Rmii_clkout clock frequency	-	50	-	MHz
$t_{ODLYDAT}$	Rmii txd output delay	6.061	9.403	14.044	ns
$t_{ODLYCTL}$	Rmii control signals output delay	6.131	9.48	14.134	ns
t_{ISUDAT}	rmii_rxd setup to rmii_clkout/rmii_clkin rising edge	3.083	4.035	5.489	ns
t_{IHDAT}	rmii_rxd hold to rmii_clkout/rmii_clkin rising edge	1.423	1.858	2.746	ns
t_{ISUCTL}	rmii_crs_dv, rmii_rx_err setup to rmii_clkout/rmii_clkin rising edge	3.088	4.217	5.253	ns
t_{IHCTL}	rmii_crs_dv, rmii_rx_err hold to rmii_clkout/rmii_clkin rising edge	1.425	1.873	2.6	ns

32.5 Interface Description

Table 32-3 RMII/MII Interface Description

Module pin	Direction	Pad name	IOMUX

name			
RMII interface			
rmii_clkout	O	GPIO1_C[0]	GRF_GPIO1C_IOMUX[1:0]=10
rmii_clkin	I	GPIO1_C[0]	GRF_GPIO1C_IOMUX[1:0]=11
rmii_tx_en	O	GPIO1_C[1]	GRF_GPIO1C_IOMUX[3:2]=10
rmii_txd1	O	GPIO1_C[2]	GRF_GPIO1C_IOMUX[5:4]=10
rmii_txd0	O	GPIO1_C[3]	GRF_GPIO1C_IOMUX[7:6]=10
rmii_rx_err	I	GPIO1_C[4]	GRF_GPIO1C_IOMUX[9:8]=10
rmii_crs_dv alid	I	GPIO1_C[5]	GRF_GPIO1C_IOMUX[11:10]=10
rmii_rxd1	I	GPIO1_C[6]	GRF_GPIO1C_IOMUX[13:12]=01
rmii_rxd0	I	GPIO1_C[7]	GRF_GPIO1C_IOMUX[15:14]=01
Management interface			
mii_md	I/O	GPIO1_D[0]	GRF_GPIO1D_IOMUX[1:0]=10
mii_mdclk	O	GPIO1_D[1]	GRF_GPIO1D_IOMUX[3:2]=10

32.6 Application Notes

32.6.1 Buffer Descriptors

Data is sent and received via buffers that are built in system memory. These buffers are pointed to by the Buffer Descriptor Rings that are also located in memory.These are two Buffer Descriptor Rings: the transmit (TX) and receive (RX) rings. The Transmit Ring Pointer （TXRINGPTR）is a pointer to the start of the Transmit Descriptor Ring. The RXRINGPTR register similarly is a pointer to the start of the Receive Descriptor Ring. The VMAC processes each descriptor in each ring sequentially until it reaches the last descriptor.Then the VMAC will automatically cycle back to the first descriptor in the ring.
Each Buffer Descriptor Ring in turn consist of a number of Buffer Descriptors. The Buffer Descriptor Rings must occupy a contiguous area of memory and must be aligned on 8-byte boundaries. The number of Buffer Descriptors in the Ring are programmed bu fields in the CONTROL register.

Each Buffer Descriptor (BD) within the Ring in turn points to a buffer in memory that contains the packet data. Each Buffer Descriptor consists of eight bytes of data formatted as two 32-bit Wrds. The first Wrd (INFO) contains various status information on the buffer itself. The most important field is the OWN bit which indicates whether the VMAC "owns" the buffer, or the processor "owns" the buffer. The OWN bit is a semaphore that indicates who is allowed access to the buffer and the buffer descriptor. If the VMAC owns the buffer, the processor must not make any changes to either the buffer descriptor or the contents of the buffer. If the processor owns the buffer then the VMAC will ignore the buffer and wait for the processor to release it. Once a buffer and the Buffer Descriptor have been prepared, the processor toggles the OWN bit and releases the buffer to the VMAC. The second Wrd of the BD is the PTR field which is a 32-bit byte addressable pointer to the packet data. The data can start on any bute boundary and need not be aligned. Any number of bytes of data can be contained in the buffer, every zero bytes. The length of the buffer is defined in the LENGTH field of the INFO Wrd. A buffer typically contains all of the bytes of a single packet.

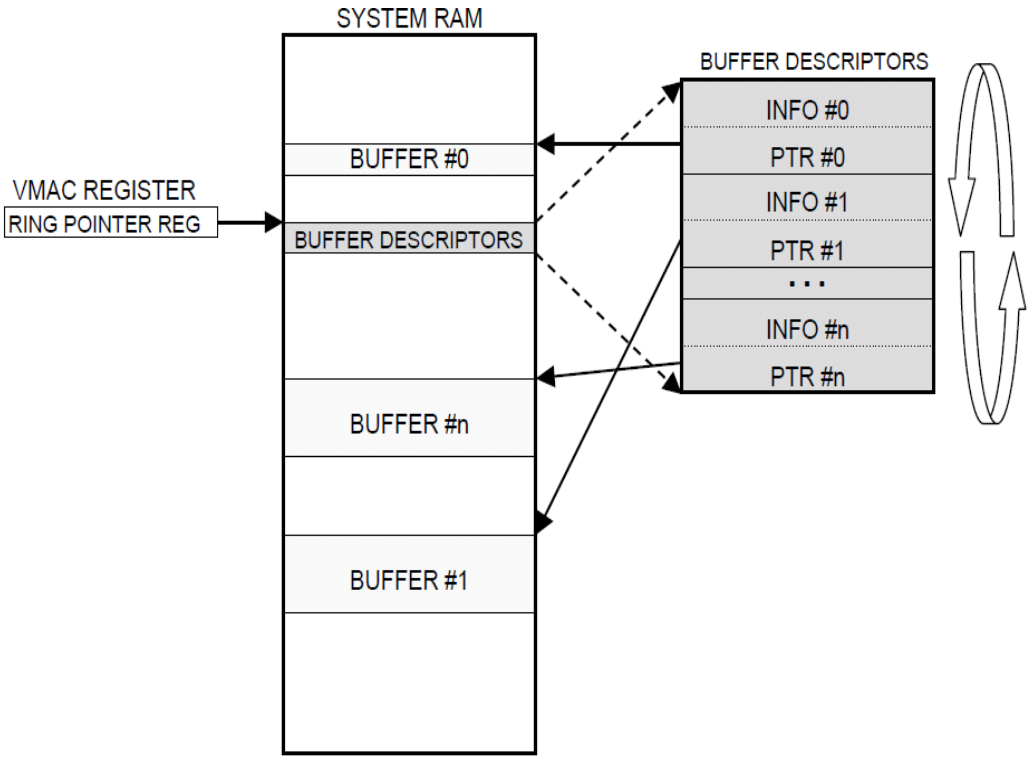


Fig. 32-7 VMAC buffer chain

Fig. 32-9 shows the relationship of the VMAC registers and the Buffer Descriptors. The RINGPTR register is shown on the left which points to the starting address of the Buffer Descriptor Ring (either TX or RX) in system memory. The Buffer Descriptor Ring takes up a very small amount of the system memory and is expanded on the right of the diagram. The Buffer Descriptor Ring has a number of Buffer Descriptors in it each containing the two 32-bit Wrds,INFO and PTR. The PTR in turn points to the start of buffer data which can be in any location in system memory. The VMAC processes one Buffer Descriptor after the next in the ring. When the last BD has been processed, it circles back to the beginning of the ring and continues processing from there.

32.6.2 Transmit Buffer Descriptor

The Transmit Buffer descriptor consists of two 32-bit Wrds, the INFO and PTR Wrds. The INFO Wrd is broken up into a number of a smaller fields described on the following page. Note that the processor fills the INFO field with one set of data, and then release the buffer to the VMAC by setting the OWN bit. Once the VMAC has completed processing this buffer, it will fill the INFO Wrd with different data and clear the OWN bit.

The processor must completely set up the Buffer Descriptor and completely fill the buffer with data before setting the OWN bit. Once the OWN bit is set, the processor must not alter the BD or buffer data.Once the OWN bit is set the VMAC will clear the OWN bit once the buffer has been sent or error condition has occurred.

1.Transmit Buffer Descriptor written by CPU

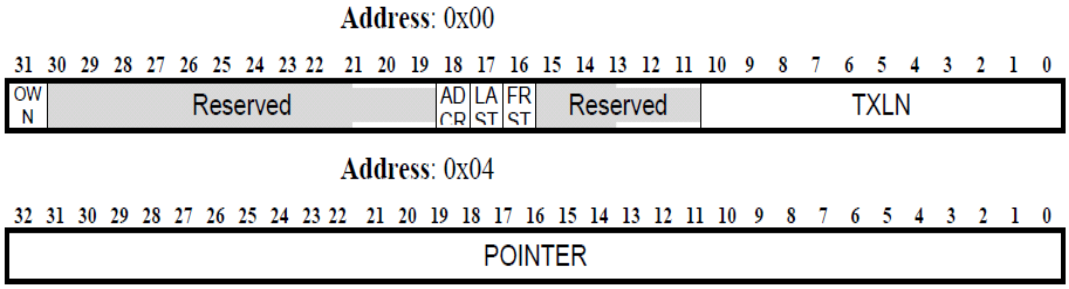


Fig. 32-8 VMAC transmit buffer descriptor written by CPU

The field descriptions for these registers are shown in the following table:

Bit	Attr	Reset Value	Description
63:32	RW	0x0	POINTER: 32 bit physical address to the start of the buffer data. Does not have to be Wrd aligned. The DMA controller will DMA bytes until it becomes Wrd aligned, then it will transfer Words. Unchanged by VMAC.
31	RW	0x0	OWN 0: buffer owned by the CPU 1: buffer owned by the VMAC
30:19	N/A	0x0	Reserved
18	RW	0x0	ADCR – ADDCRC 1: VMAC will compute and add the 4 byte CRC to the end of the packet. 0: don't add the CRC (FCS) to the end of the packet
17	RW	0x0	LAST – This bit must be set to one when it is the last buffer in a packet. Note that both FIRST and LAST are set at the same time if the buffer is large enough to hold the entire packet
16	RW	0x0	FIRST – This bit must be set to one when it is the first buffer in a packet
15:11	N/A	0x0	Reserved
10:0	RW	0x0	TXLEN – length of data in this buffer to be transmitted. Can be zero length in the case of packet-chaining, but the minimum length for a packet is 64 bytes.

2. Transmit Buffer Descriptor Written by VMAC

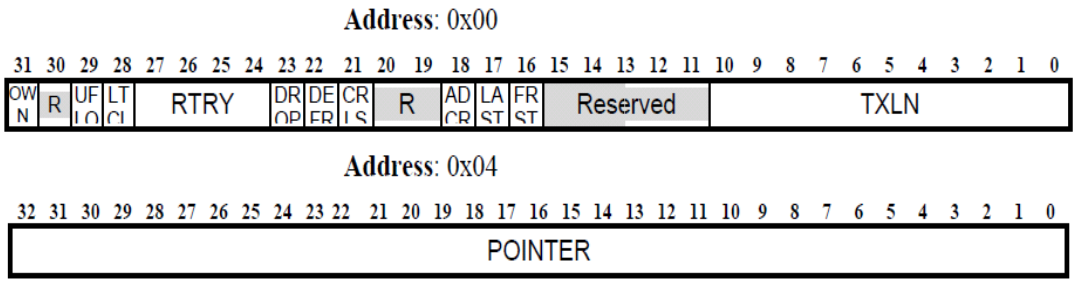


Fig. 32-9 VMAC transmit buffer descriptor written by VMAC

The field descriptions for these registers are shown in the following table:

Table 32-4 VMAC tx buffer descriptor

Bit	Attr	Reset Value	Description
63:32	R	0x0	POINTER: 32 bit physical address to the start of the buffer data. Does not have to be Wrd aligned. The DMA controller will DMA bytes until it becomes Wrd aligned, then it will transfer Wrds. Unchanged by VMAC.
31	RW	0x0	OWN – Cleared by VMAC. When this bit is cleared, the TXINT bit is set if END is also set and the transmission has complete.
30	N/A	0x0	Reserved
29	RW	0x0	UFLO – Underflow erRr, packet data corrupted and dropped because data was not available in time. Larger FIFOs, more AHB bus bandwidth or lower system latency required.
28	RW	0x0	LATECOL – Late collision error – Packet dropped due to late collision
27:24	RW	0x0	RTRY – Retry count, Number of times the packet was retried. Packet transmission is attempted up to 16 times. If the packet is transmitted successfully on the first try, the value is zero.
23	RW	0x0	DRP – More than 16 retransmissions were attempted and the packet was dropped
22	RW	0x0	DEFR – Transmission was deferred due to traffic on the wire
21	RW	0x0	CRLS- Carrier sense was lost during transmission
20:19	N/A	0x0	Rreserved
18	R	0x0	ADCR – ADDCRC 1: VMAC will compute and add the 4 byte CRC to the end of the packet. 0: don't add the CRC (FCS) to the end of the packet
17	R	0x0	LAST – This bit is one when it is the last buffer in a packet.
16	R	0x0	FIRST – This bit is one when it is the first buffer in a packet
15:11	N/A	0x0	Reserved
10:0	R	0x0	TXLEN – length of data in this buffer to be transmitted.

32.6.3 Receive Buffer Descriptor

The receive buffer descriptor is very similar to the Transmit Buffer Descriptor and again consists of two 32-bit Words, the INFO and PTR Words. The INFO Wrd is broken up into a number of smaller fields described on the following page.

Note that the processor fills the INFO field with one set of data, and then releases the buffer to the VMAC by setting the OWN bit. Once the VMAC has completed processing this buffer, it will fill the INFO Wrd with different data and clear the OWN bit.

The processor must completely set up the Buffer Descriptor before setting the OWN bit.The data buffer can be left uninitialized as VMAC will fill it with data

once a packet has been received. Once the OWN bit is set, the processor must NOT alter the Buffer Descriptor. Once the OWN bit is set the VMAC may begin operating on the buffer immediately. The VMAC will clear the OWN bit once the buffer has been filled with data or an error condition has occurred.

1.Receive Buffer Descriptor Written by VMAC

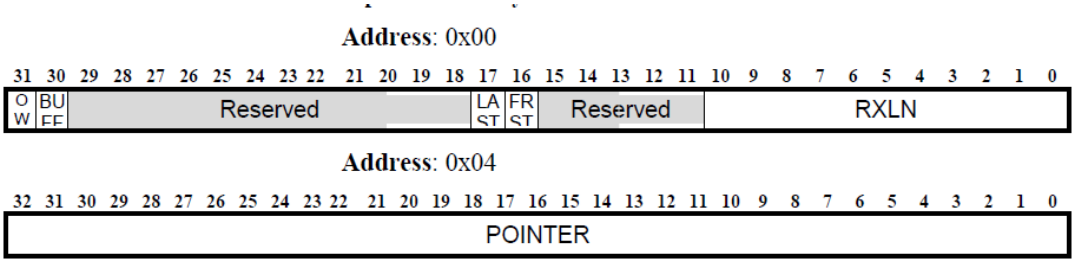


Fig. 32-10 VMAC receive buffer descriptor written by VMAC

The field descriptions for this register are shown below.

Table 32-5 VMAC- rx buffer descriptor for VMAC

Bit	Attr	Reset Value	Description
63:32	R	0x0	POINTER: 32 bit physical address to the start of the buffer data.Does not have to be Word aligned. The DMA controller will DMA bytes until it becomes Wrd aligned,then it will transfer Wrds.
31	RW	0x0	OWN 0: buffer owned by the CPU 1: buffer owned by the VMAC This bit is cleared by VMAC. When this bit is cleared and LAST is set to 1, the RXINT bit is set.
30	RW	0x0	BUFF – Buffer error, an error occurred during packet chaining, with data in one or more of the buffers in the chain corrupted.Software shold discard the entire packet and return the buffers back to the VMAC.
29:18	N/A	0x0	Reserved
17	RW	0x0	LAST – This bit must be set to one when it is the last buffer in a packet.
16	RW	0x0	FIRST – This bit must be set to one when it is the first buffer in a packet. Note that in certain error conditions, you may come acRss a FIRST bit being set without a corresponding END.In this case the previous packet must be dropped.
15:11	N/A	0x0	Reserved
10:0	RW	0x0	RXLEN – Length of the data in this buffer in bytes.Might be zero length.Note that if a packet of exactly the same length as the RXLEN is received,there is a chance that the VMAC will release the buffer without the LAST bit set in anticipation of using buffer chaining.The next buffer will be released with the LAST bit set but will be zero length.This is especially true when the CPU

			is clocking much faster than the VMAC.To prevent chaining,the CPU shold set RXLEN to a slightly larger value than ghe largest expected packet length.

2.Receive Buffer Descriptor Written by CPU

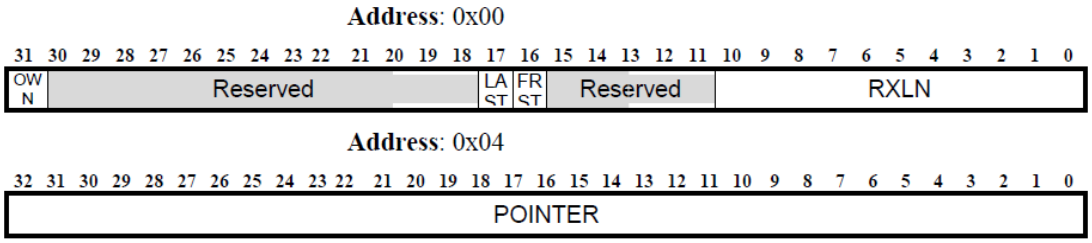


Fig. 32-11 VMAC receive buffer descriptor written by CPU

The field descriptions for this register are shown below.

Table 32-6 VMAC-rx buffer descriptor for CPU

Bit	Attr	Reset Value	Description
63:32	R	0x0	POINTER: 32 bit physical address to the start of the buffer data. Does not have to be word aligned. The DMA controller will DMA bytes until it becomes Wrd aligned, then it will transfer Wrds.
31	RW	0x0	OWN 0: Buffer owned by the CPU 1: Buffer owned by the VMAC
30:18	N/A	0x0	Rreserved
17	RW	0x0	LAST – This bit is one when it is the last buffer in a packet. CPU clears it to zero.
16	RW	0x0	FIRST – This bit is one when it is the first buffer in a packet, CPU clears it to zeR
15:11	N/A	0x0	Reserved
10:0	RW	0x0	RXLEN – Maximum length of data in this buffer in bytes. Can be zero but strongly discouraged.Minimum of 64 bytes recommended.1536 recommended unless using chaining.

32.6.4 Buffer Chaining

The length of each buffer is defined in the RXLEN or TXLEN fields within the INFO Wrd of each Buffer Descriptor. The length of a buffer can be between zero and 2047 bytes though buffer of less than 64 bytes and strongly discouraged. Larger FIFOs are highly recommended when using buffer chaining.

Buffer chaining is also useful for receive packets by using more buffers of smaller sizes.Chaining allows the use of smaller buffers which in turn creates more buffers in the same amount of memory. More receive buffers means there a lower chance of dropping important short packets and relying on upper level software to transmit the dropped packets.Note that by chaining buffers you can still assemble the packet data to be contiguous for large packets.

32.6.5 Automatic Descriptor Polling

The TX and RX BDT rings will be polled at the frequency programmed into the POLLRATE register.The DMA controller polls the BDT rings to see if the next descriptors in the chain are owned by the VMAC. There are also several instances where automatic polling of the descriptor chains will occur.

Chapter 33 High-Speed ADC /TS stream Interface

33.1 Overview

HS-ADC Interface Unit is interface unit for connecting the High Speed AD Converter to AMBA AHB bus. That implement bus speed convert at low speed AD Converter bus to high speed AHB bus. HS-ADC Interface Unit fetches the bus data by the AD converter and stores that to asynchronous FIFO after the AD clock is active when OS configure completion by DMA and HS-ADC Interface Unit. The HS-ADC Interface Unit generates the DMA request signal when data length of the asynchronous FIFO over the almost full level or almost empty level.

33.1.1 Features

- Support HS-ADC interface with 8bits/10bits data bus
- Support Transport-Stream Interface with 8bits data bus
- Support GPS interface with 2bits/4bits data bus
- Support combined interrupt output, the source includes:full interrupt, empty interrupt
- Support DMA transfers mode and that generate DMA request from the event of almost full or almost empty, etc.
- Support two channel mode:single channel and dual channel
- Support the most significant bit negation or not
- Support sign bitextention
- Support two store mode: store to high 8bits/10bits and store to low 8bits/10bits
- Support an asynchronous build-in FIFO with 128x64 size
- Support AHB burst transfer, including SINGLE, INCR4, INCR8, INCR16

33.2 Block Diagram

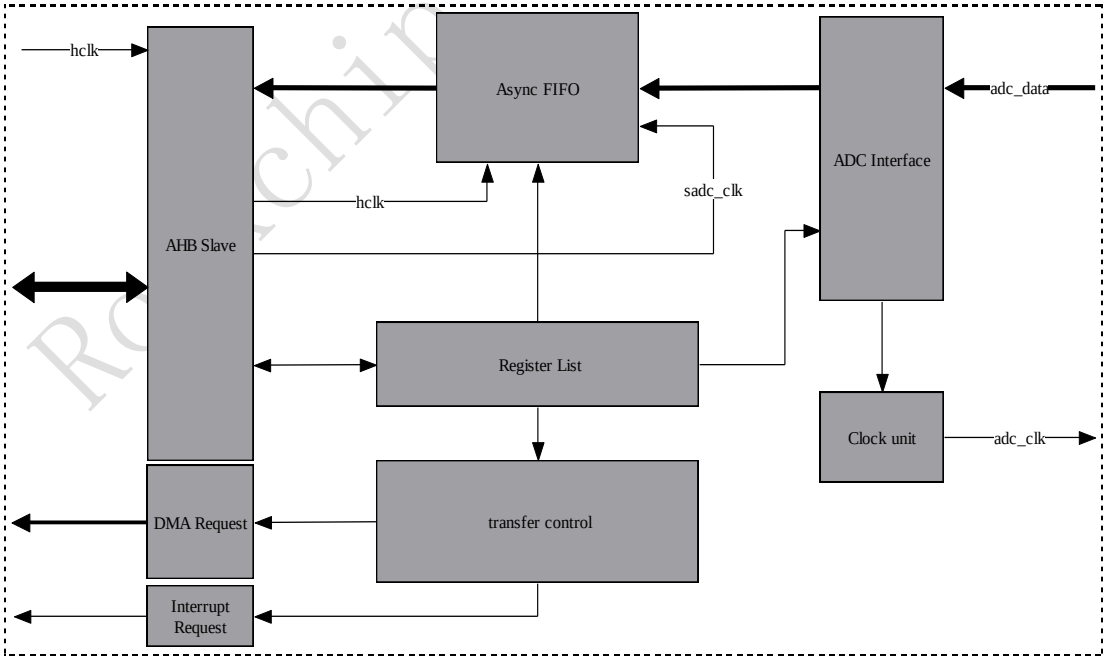


Fig. 33-1 HS-ADC/TS Interface block diagram

33.3 Function Description

This module can be configured for 3 interfaces:

1. HS-ADC interface
2. GPS interface
3. TS interface

1. HS-ADC interface

When this module is used as HS-ADC interface,user should configureGRF_GPIO2B_IOMUX and GRF_GPIO2C_IOMUX to select HS-ADC data input, and select hsadc_clock input(pgs_clkin) from pad or output hsadc_clock(hsadc_clkout) to pad. The direction of hsadc_clock is determined by software through configuring CRU_CLKSEL_CON[5:4].

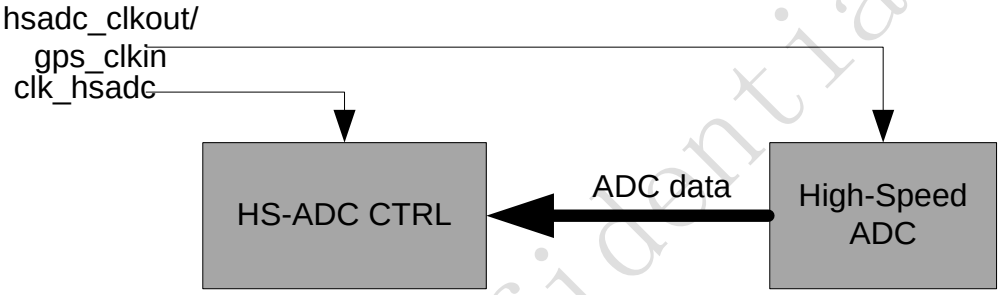


Fig. 33-2 HS-ADC application diagram

2. GPS interface

When this module is used as GPS interface, user should configureGRF_GPIO2C_IOMUX to select 2bits or 4bitsGPS data input and gps_clk as GPS clock input from pad.

Also,user should configure CRU_CLKSEL_CON[5:4] to select gps_clk as HS-ADC controller working clock source.

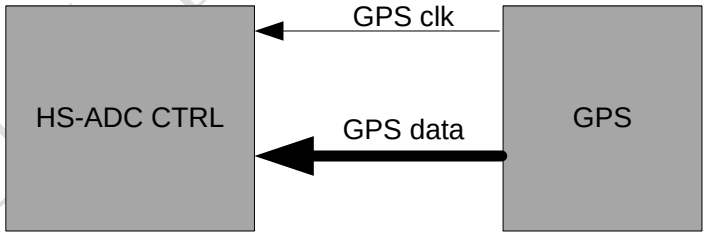


Fig. 33-3 GPS application diagram

3. TS interface

When this module is used as TS interface, user should configure GRF_GPIO2B_IOMUX and GRF_GPIO2C_IOMUX to select 8bit TS data, ts_sync, ts_valid and ts_fail input and gps_clk input as TS clock input from pad.

Also,user should configure CRU_CLKSEL_CON[5:4] to select gps_clk from pad as TS clock input.

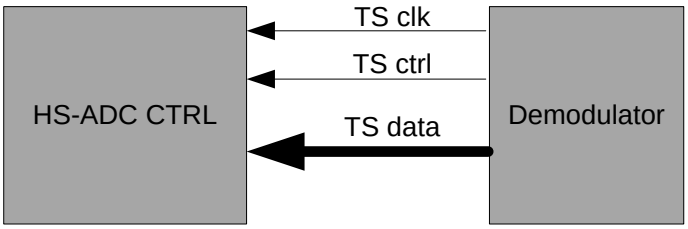


Fig. 33-4 TS application diagram

33.4 Register Description

33.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
HSADC_CTRL	0x00	RW	0x00000000	control register
HSADC_IER	0x04	RW	0x00000000	interrupt enable/mask register
HSADC_ISR	0x08	RW	0x00000000	interrupt status register
HSADC_TS_FAIL	0x0c	R	-	ts fail register
HSADC_DATA	0x10	R	-	data register

33.4.2 Detail Register Description

HSADC_CTRL
Address: Operational Base + offset (0x0000)
Control register

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:24	RW	0x0	almost_full_level Define almost full trigger level 0x0~"0xf" - configure valid range (Notes: 1 level indicate 4 entries data in the async FIFO. and this configure range mapping to 64 - 124 entries data in the async FIFO.)
23:20	RO	0x0	reserved
19:16	RW	0x0	almost_empty_level Define almost empty trigger level 0x0~"0xf" - configure valid range (Notes: 1 level indicate 4 entries data in the async FIFO. and this configure range mapping to 0 - 60 entries data in the async FIFO.)
15:13	RO	0x0	reserved
12	RW	0x0	sbex Sign Bit extent select Sign bit extent select when store width wider than interface width 0: fill 0 1: fill sign extend bit

Bit	Attr	Reset Value	Description
11	RW	0x0	gpsw GPS interface data width select 0:2bit data mode 1:4bit data mode
10	RW	0x0	ts_sync_en TS sync interface enable Field0000 Description
9	RW	0x0	ts_valid_en TS valid interface enable Enable ts interface "ts_valid" signal as data valid indicator 0:disable 1:enable
8	RW	0x0	ts_sel MPEG-TS input select 0 : MPEG-TS is Not selected 1 : MPEG-TS is selected *When gps_sel & ts_sel both equal to 0,then ADC IF is selected
7	RW	0x0	chan data input channel select 0 : single channel, ADC input from adc_data_l 1 : double channel data input from adc_data_l/adc_data_h
6	RW	0x0	stmd Store mode select 0:8bit store mode 1:16bit store mode
5	RW	0x0	dma_req_mode DMA request mode select 1 - almost full generate DMA request signal (Notes: this mode generate DMA request signal from almost full condition and cancel DMA request signal from almost empty condition. so you need configure two level by almost full level and almost empty level) 0 - almost empty generate DMA request signal (Notes: this mode generate DMA request signal from almost empty condition and that only once DMA request.)

Bit	Attr	Reset Value	Description
4	RW	0x0	smsb MSB negation select 1:negation 0:not negation
3	RW	0x0	sctl store mode select fetch the bus data by AD converter and that store to high 8-bit/10-bit or low 8-bit/10-bit at a haft word width before push to Async FIFO: "1" - store to high 8-bit/10-bit "0" - store to low 8-bit/10-bit (Notes: have sign extend if that configure of store to low 8-bit/10-bit)
2	RW	0x0	bwth The data bus width of AD converter "1" - 10-bit "0" - 8-bit
1	RW	0x0	gps_sel GPS input select 0 : GPS is Not selected 1 : GPS is selected *When gps_sel & ts_sel both equal to 0,then ADC IF is selected
0	RW	0x0	adc_en HS-ADC Interface Unit Enable Bit "1" - enable (Notes: will return 1 when the hardware started transfer) "0" - disable (Notes: other bit can be modify only the hardware return 0)

HSADC_IER
Address: Operational Base + offset (0x0004)
Interrupt control register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	int_empty_en Interrupt en/disable bit for the empty interrupt flag of async FIFO "1" - enable "0" - disable

Bit	Attr	Reset Value	Description
0	RW	0x0	int_full_en Interrupt en/disable bit for the full interrupt flag of async FIFO "1" - enable "0" - disable

HSADC_ISR
Address: Operational Base + offset (0x0008)
Interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	int_empty_stat_ind Async FIFO empty interrupt flag "1(R)" - This bit will be set to "1" when Async FIFO empty status and that only to read operation. "0(W)" - Write "0" to bit for clear the interrupt flag and that only to wrtie operation.
0	RW	0x0	int_full_stat_ind Async FIFO full interrupt flag . "1(R)" - This bit will be set to "1" when Async FIFO full status and that only to read operation. "0(W)" - Write "0" to bit for clear the interrupt flag and that only to wrtie operation.

HSADC_TS_FAIL
Address: Operational Base + offset (0x000c)
ts fail register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	ts_fail_ahb TS stream fail indicator this signal only valid when select TS stream input(mpts=1) 0:TS stream decode successfully 1:TS stream decode fail

HSADC_CGCTL
Address: Operational Base + offset (0x0010)
HSADC Clock Gating control

Bit	Attr	Reset Value	Description
31:1	RW	0x00000000	cycle_cfg clock gated cycles configuration when configure cg_enable to 1 and cycle_cfg to non-zero value,HSADC clock will be gated for cycle_cfg cycles ,then clock recover.
0	RW	0x0	cg_enable clock gating enable control 0:clock gating disable 1:clock gating enable

HSADC_DATA
Address: Operational Base + offset (0x0020)
Data register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	DATA DATA DATA

33.5 Timing Diagram

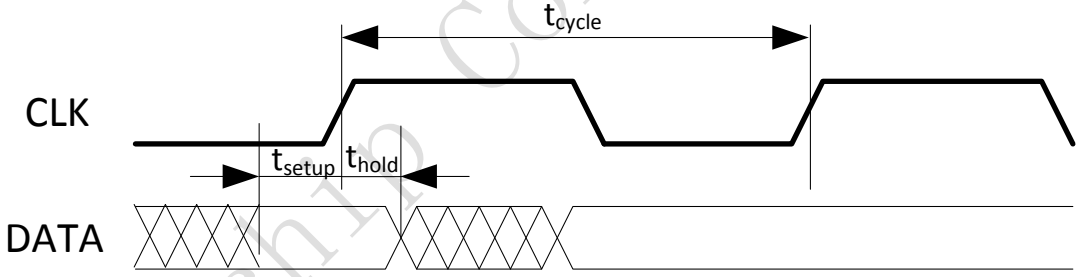


Fig. 33-5 HS-AD Interface timing diagram

Table 33-1 HS-ADC interface timing parameter

Parameter	Symbol	Min	Typ	Max	Unit
Clock Cycle	t_{cycle}	50	50	50	ns
Data to clock setup time	t_{setup}	3.123	4.211	6.366	ns
Data to clock hold time	t_{hold}	0	0	0	ns

Note:When interface configured as TS or GPS interface,CLK is input from pad,when configured as ADC interface,CLK is output to pad.

33.6 Interface Description

The following are typical iomux configurations.
For more information, please reference detailed register description in GRF chapter.
In ADC mode:

Table 33-2 IOMUX configuration in ADC mode

Module Pin	Direction	Pad Name	IOMUX Setting
hsadc_clkout/ gps_clkin	O/ I	GPIO2_C[0]	GRF_GPIO2C_IOMUX[1:0]=2'b11/GRF_GPIO2C_IOMU
adc_data_i[0]	I	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=2'b11
adc_data_i[1]	I	GPIO2_C[4]	GRF_GPIO2C_IOMUX[9:8]=2'b11
adc_data_i[2]	I	GPIO2_C[5]	GRF_GPIO2C_IOMUX[11:10]=2'b11
adc_data_i[3]	I	GPIO2_C[6]	GRF_GPIO2C_IOMUX[13:12]=2'b11
adc_data_i[4]	I	GPIO2_C[7]	GRF_GPIO2C_IOMUX[15:14]=2'b11
adc_data_i[5]	I	GPIO2_C[2]	GRF_GPIO2C_IOMUX[5:4]=2'b11
adc_data_i[6]	I	GPIO2_C[1]	GRF_GPIO2C_IOMUX[3:2]=2'b11
adc_data_i[7]	I	GPIO2_B[7]	GRF_GPIO2B_IOMUX[15:14]=2'b11
adc_data_i[8]	I	GPIO2_B[5]	GRF_GPIO2B_IOMUX[11:10]=2'b11
adc_data_i[9]	I	GPIO2_B[4]	GRF_GPIO2B_IOMUX[9:8]=2'b11

In GPS mode:

Table 33-3 IOMUX configuration in GPS mode

Module Pin	Direction	Pad Name	IOMUX Setting
gps_clkin	I	GPIO2_C[0]	GRF_GPIO2C_IOMUX[1:0]=2'b10
adc_data_i[0]	I	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=2'b11
adc_data_i[1]	I	GPIO2_C[4]	GRF_GPIO2C_IOMUX[9:8]=2'b11
adc_data_i[2]	I	GPIO2_C[5]	GRF_GPIO2C_IOMUX[11:10]=2'b11
adc_data_i[3]	I	GPIO2_C[6]	GRF_GPIO2C_IOMUX[13:12]=2'b11

In TS mode:

Table 33-4 IOMUX configuration in TS mode

Module Pin	Direction	Pad Name	IOMUX Setting
gps_clkin	I	GPIO2_C[0]	GRF_GPIO2C_IOMUX[1:0]=2'b10
adc_data_i[0]	I	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=2'b11
adc_data_i[1]	I	GPIO2_C[4]	GRF_GPIO2C_IOMUX[9:8]=2'b11
adc_data_i[2]	I	GPIO2_C[5]	GRF_GPIO2C_IOMUX[11:10]=2'b11
adc_data_i[3]	I	GPIO2_C[6]	GRF_GPIO2C_IOMUX[13:12]=2'b11
adc_data_i[4]	I	GPIO2_C[7]	GRF_GPIO2C_IOMUX[15:14]=2'b11
adc_data_i[5]	I	GPIO2_C[2]	GRF_GPIO2C_IOMUX[5:4]=2'b11
adc_data_i[6]	I	GPIO2_C[1]	GRF_GPIO2C_IOMUX[3:2]=2'b11
adc_data_i[7]	I	GPIO2_B[7]	GRF_GPIO2B_IOMUX[15:14]=2'b11
ts_valid	I	GPIO2_B[5]	GRF_GPIO2B_IOMUX[11:10]=2'b11
ts_fail	I	GPIO2_B[4]	GRF_GPIO2B_IOMUX[9:8]=2'b11
ts_sync	I	GPIO2_B[6]	GRF_GPIO2B_IOMUX[13:12]=2'b11

33.7 Application Notes

The following sections will describe the operation of DMA requests and DMA transfers:

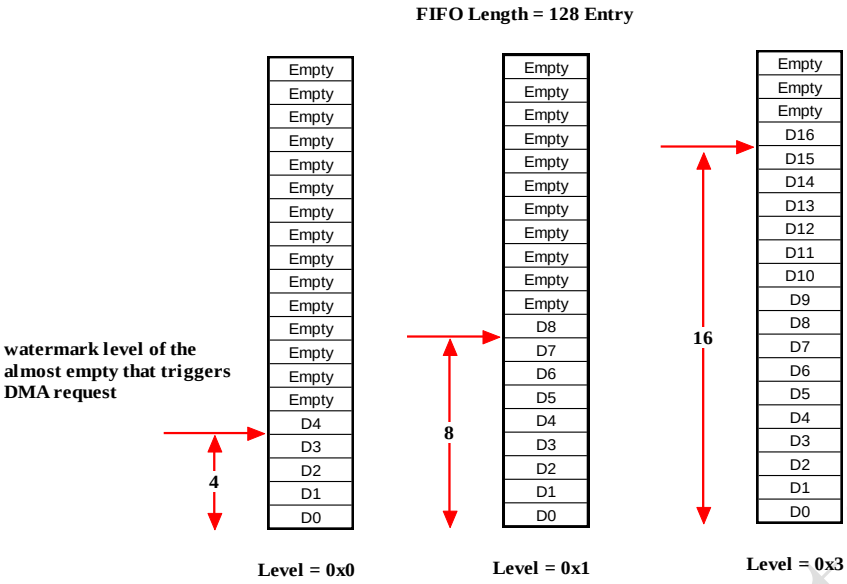


Fig. 33-6Almost empty triggers a DMA request by DMA request mode

The DMA request signal will be generatedfrom a watermark level triggerwhen data stored to FIFO over the watermark level of almost empty, where the watermark level can be configuredthrough HSADC_CTRL[19:16] by software . This DMA request mode doesn't care the watermark level of almost full. The sample for watermark level configuration is shown in figure above.

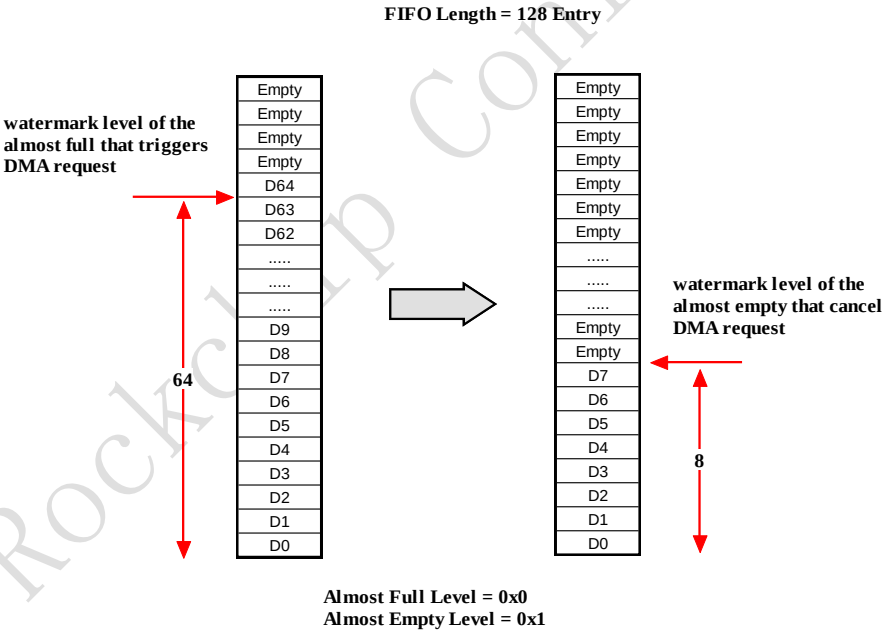


Fig. 33-7Almost full triggers a DMA request by DMA request mode

The DMA request signal will be generated from a watermark level trigger when data stored to FIFO over the watermark level of almost full. It continuesto generate request signal when the number of data in FIFO greater than watermark level of almost empty. This DMA request mode needs configuretwo watermark levels: watermark level of almost empty at the HSADC_CTRL[19:16] and watermark level of almost full at the HSADC_CTRL[27:24]. The sample for watermark level configuration is shown in figure above.

When controller works in TS mode,the interface signal ts_sync should always be used.

Chapter 34 PID-FILTER

34.1 Overview

THE PID Filter controller is used to process the orginial TS(Transport Stream) from HS-ADC. It receives the original TS from HS-ADC,and transport the filtered TS to memory.

34.1.1 Key Feature

- Support 32-bit AHB slave configuration port & buffer write
- Support PID filter up to 64 channels'PID simultaneously
- Support sync-byte detection in transport packet head
- Support 188work x 2 PingPong buffer
- Support combined interrupt and interrupt polarity/type configurable
- Support transfer handshake mechanismwith external DMA
- Support packet lost mechanismwhen bandwidth is limited

34.2 Block Diagram

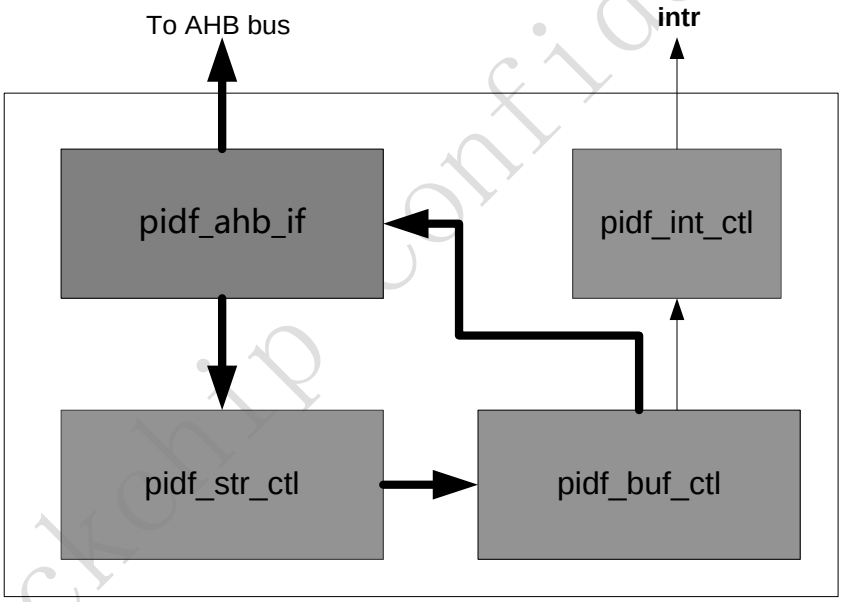


Fig. 34-1 PID-Filter block diagram

34.3 Function Description

This module is used to co-work with external DMA.The DMA receives raw TS data from HS-ADC controller and transfers them to PID-Filter, then PID-Filter controller filters specific TS packet out and triggers DMA to transfer remaining useful TS packet to memory for future usage.

Besides,when PID-Filter controller receives the unexpected head of TS packet or loses packet in TS receiving procedure,the controller is capable of outputting interrupt to processor.

34.4 Register Description

34.4.1 Register summary

Name	Offset	Size	Reset Value	Description
PIDF_GCTL	0x0000	W	0x00000000	Global Control register
PIDF_GSTA	0x0004	W	0x0000000a	Global Status register
PIDF_ICTL	0x0010	W	0x00000000	Interrupt control register
PIDF_IMR	0x0014	W	0x00000000	Interrupt mask register
PIDF_ICLR	0x0018	W	0x00000000	Interrupt clear register
PIDF_IRSR	0x001C	W	0x00000000	Raw interrupt status(pre-masking) register
PIDF_ISR	0x0020	W	0x00000000	Interrupt status register
PIDF_CHID0	0x0080	W	0x00000000	Filter out ID & enable for channel 0
PIDF_CHID1	0x0084	W	0x00000000	Filter out ID & enable for channel 1
PIDF_CHID2 ~PIDF_CHID63	0x0088 ~0x0017C	W	0x00000000	Filter out ID & enable for channel 2~31
PIDF_DR	0x0200	W	0x00000000	Data buffer write/read entrance

34.4.2 Register description

PIDF_GCTL
Address: Base + offset(0x0000)
The global control register of PID Filter controller

Bit	Attr	Reset Value	Description
31:3	R	0x0	Reserved
2	RW	0x0	Filter function bypass Filter function bypass enable. When set,data write transfer to buffer will not bypass PID filter function.
1	RW	0x0	Hardware handshake mechanisim enable 0: Hardware handshake disable 1: Hardware handshake enable
0	RW	0x0	PID filter function enable 0:PID filter disable 1:PID filter enable

PIDF_GSTA
Address: Base + offset(0x0004)
The global status register of PID Filter controller

Bit	Attr	Reset Value	Description
31:4	R	0x0	Reserved
3	R	0x1	Buffer 1 empty indicator Buffer 1 empty status indicator
2	R	0x0	Buffer 1 full indicator Buffer 1 full status indicator

1	R	0x1	Buffer 0 empty indicator Buffer 0 empty status indicator
0	R	0x0	Buffer 0 full indicator Buffer 0 full status indicator

PIDF_ICTL

Address: Base + offset(0x0010)

The interrupt control register of PID filter controller

Bit	Attr	Reset Value	Description
31:3	R	0x0	Reserved
2	RW	0x0	Interrupt type select 0:Edge-type interrupt select 1:Level-type interrupt select
1	RW	0x0	Interrupt polarity select 0: High-active interrupt select for combined int output 1: Low-active interrupt select for combined int output
0	RW	0x0	Interrupt enable 0:Interrupt disable 1:Interrupt enable

PIDF_IMR

Address: Base + offset(0x0014)

The interrupt mask register of PID filter controller

Bit	Attr	Reset Value	Description
31:4	R	0x0	Reserved
3:0	RW	0x0	Interrupt mask for each interrupt source Interrupt mask for each interrupt source 0:Interrupt source mask disable for corresponding interrupt source 1:Interrupt source mask enable for corresponding interrupt source bit0 : Sync byte error interrupt bit1 : Packet lost detected interrupt bit2 : buffer empty interrupt bit3 : buffer full interrupt

PIDF_ICLR

Address: Base + offset(0x0018)

The interrupt clear register of PID filter controller

Bit	Attr	Reset Value	Description
31:4	R	0x0	Reserved
3:0	RW	0x0	Interrupt clear for each interrupt source Interrupt clear for each interrupt source 0 : bit value will be set to "0" after write 1 to it 1 : interrupt clear for corresponding interrupt source bit0 : Sync byte error interrupt

			bit1 : Packet lost detected interrupt bit2 : buffer empty interrupt bit3 : buffer full interrupt

PIDF_IRSR
Address: Base + offset(0x001C)
The interrupt raw status register of PID filter controller

Bit	Attr	Reset Value	Description
31:4	R	0x0	Reserved
3:0	R	0x0	Interrupt raw status Interrupt raw status for each interrupt source bit0 : Sync byte error interrupt bit1 : Packet lost detected interrupt bit2 : buffer empty interrupt bit3 : buffer full interrupt

PIDF_ISR
Address: Base + offset(0x0020)
The interrupt status register of PID filter controller

Bit	Attr	Reset Value	Description
31:4	R	0x0	Reserved
3:0	R	0x0	Interrupt status Interrupt status for each interrupt source bit0 : Sync byte error interrupt bit1 : Packet lost detected interrupt bit2 : buffer empty interrupt bit3 : buffer full interrupt

PIDF_CHID0
Address: Base + offset(0x0080)
The pid index and enable for channel 0 of PID filter controller

Bit	Attr	Reset Value	Description
31:17	R	0x0	Reserved
16	RW	0x0	Channel 0 enable 0:Diable to filter out channel ID 0 1:Enable to filter out channel ID 0
15:13	R	0x0	Reserved
12:0	RW	0x0	Channel 0 PID value Channel 0 PID value to filter out

The PID_CHIDx register's address offset is (x*0x4)+0x80,and the definition of register field is the same as PID_CHID0.

PIDF_DR
Address: Base + offset(0x0200)
The data register of PID filter controller

Bit	Attr	Reset Value	Description
31:0	RW	0x0	PID Filter data register

34.5 Application Notes

34.5.1 Working Flow

Below is the typical data flow:

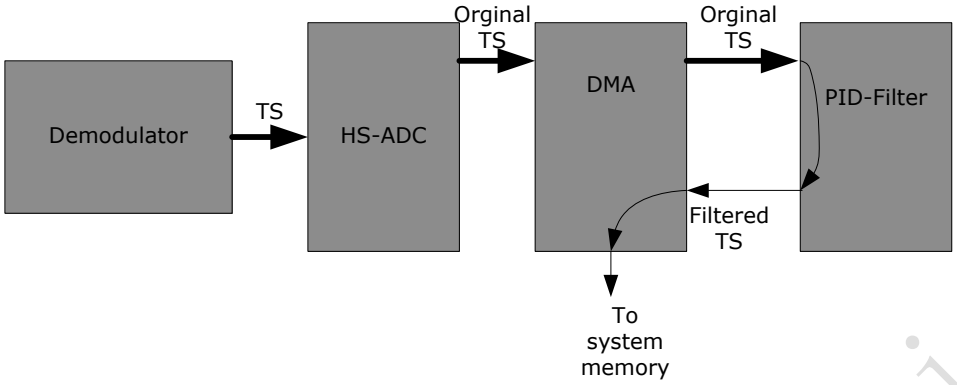


Fig. 34-2 PID-Filter data flow

Typically,PID-Filter and HS-ADC use hardware-handshake mechanism to transact with DMA.

When start to receiver TS from demodulator, user should follow the below sequence:

a. Start DMA to wait for hardware request
b. Start PID-Filter to wait for data from HS-ADC
c. Start HS-ADC to receiving

When stop to receiver TS from demodulator, user should follow the below sequence:

a. StopHS-ADC
b. Stop DMA
c. Stop PID-Filter

34.5.2 About internal buffer access

The internal ping-pong buffer in PID-Filter cannot be random accessed. User can only accessed the buffer through the register entrance (PIDF_BASE +PIDF_DR) in a particular sequence.

34.5.3 Recommendation

It’s recommended to soft reset the PID-Filter controller before restart PID-Filter working.

Chapter 35 SPI Controller

35.1 Overview

The serial peripheral interface is an APB slave device. A four-wire, full-duplex serial protocol from Motorola. There are four possible combinations for the serial clock phase and polarity. The clock phase (SCPH) determines whether the serial transfer begins with the falling edge of slave select signals or the first edge of the serial clock. The slave select line is held high when the SPI is idle or disabled. This SPI controller can work as either master or slave mode.

- Support Motorola SPI,TI Synchronous Serial Protocol and National Semiconductor Microwire interface
- Support 32-bit APB bus
- Support two internal 16-bit wide and 32-location deep FIFOs, one for transmitting and the other for receiving serial data
- Support two chip select signals in master mode
- Support 4,8,16 bit serial data transfer
- Support configurable interrupt polarity
- Support asynchronous APB bus and SPI clock
- Support master and slave mode
- Support DMA handshake interface and configurable DMA water level
- Support transmit FIFO empty, underflow, receive FIFO full, overflow, underflow interrupt and all interrupts can be masked
- Support configurable water level of transmit FIFO empty and receive FIFO full interrupt
- Support combine interrupt output
- Support up to half of SPI clock frequency transfer in master mode and one sixth of SPI clock frequency transfer in slave mode
- Support full and half duplex mode transfer
- Stop transmitting SCLK if transmit FIFO is empty or receive FIFO is full in master mode
- Support configurable delay from chip select active to SCLK active in master mode
- Support configurable period of chip select inactive between two parallel data in master mode
- Support big and little endian, MSB and LSB first transfer
- Support two 8-bit audio data store together in one 16-bit wide location
- Support sample RXD 0~3 SPI clock cycles later
- Support configurable SCLK polarity and phase
- Support fix and incremental address access to transmit and receive FIFO

35.2 Block Diagram

The SPI comprises with:

- AMBA APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller

- Register block
- Shift control and interrupt

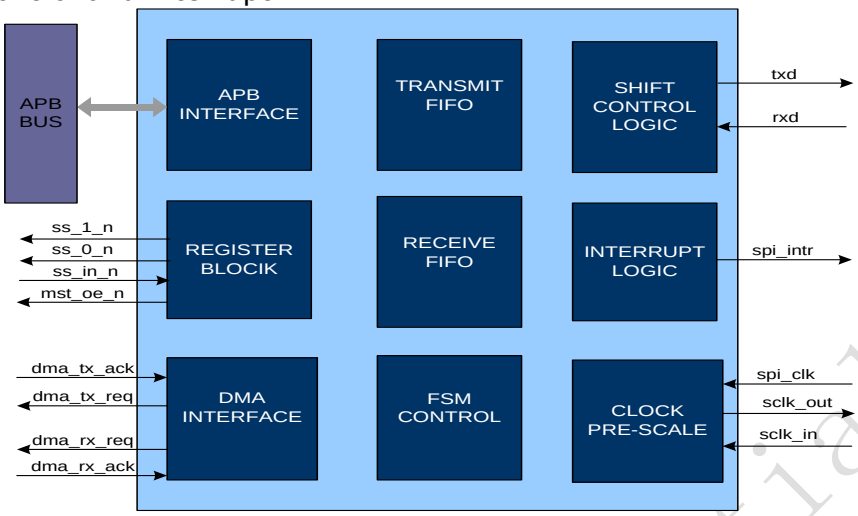


Fig. 35-1SPI Controller Block diagram

APB INTERFACE

The host processor accesses data, control, and status information on the SPI through the APB interface. The SPI supports APB data bus widths of 8, 16, and 32 bits.

DMA INTERFACE

This block has a handshaking interface to a DMA Controller to request and control transfers. The APB bus is used to perform the data transfer to or from the DMA Controller.

FIFO LOGIC

For transmit and receive transfers, data transmitted from the SPI to the external serial device is written into the transmit FIFO. Data received from the external serialdevice into the SPI is pushed into the receive FIFO. Both fifos are 32x16bits.

FSM CONTROL

Control the state's transformation of the design.

REGISTER BLOCK

All registers in the SPI are addressed at 32-bit boundaries to remain consistent with the AHB bus. Where the physical size of any register is less than 32-bits wide, the upper unused bits of the 32-bit boundary are reserved. Writing to these bits has no effect; reading from these bits returns 0.

SHIFT CONTROL

Shift control logic shift the data from the transmit fifo or to the receive fifo. This logic automatically right-justifies receive data in the receive FIFO buffer

INTERRUPT CONTROL

The SPI supports combined and individual interrupt requests, each of which can bemasked. The combined interrupt request is the ORed result of all other SPI interrupts after masking.

35.3 Function description

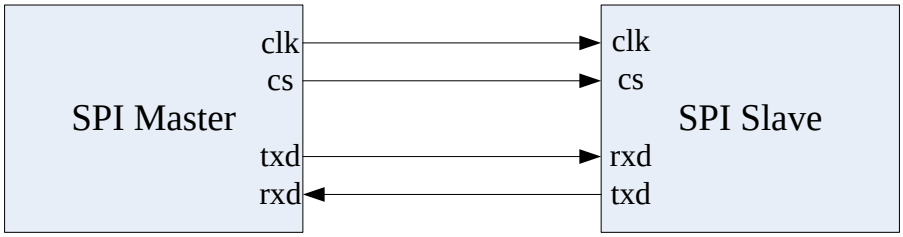


Fig. 35-2SPI Master & Slave Interconnection

The SPI controller support dynamic switching between master and slave in a system. The diagram show how the SPI controller connects with other SPI devices.

- **Operation Modes**

The SPI can be configured in the following two fundamental modes of operation: Master Mode when SPI_CTRLR0 [20] is 1'b0, Slave Mode when SPI_CTRLR0 [20] is 1'b1.

- **Transfer Modes**

The SPI operates in the following three modes when transferring data on the serial bus:

1.Transmit and Receive:

When SPI_CTRLR0 [19:18] = 2'b00, both transmit and receive logic are valid.

2.Transmit Only:

When SPI_CTRLR0 [19:18] = 2'b01, the receive data are invalid and should not be stored in the receive FIFO.

3.Receive Only:

When SPI_CTRLR0 [19:18]= 2'b10, the transmit data are invalid.

- **Clock Ratios**

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as:

Master: $F_{spi\_clk} >= 2 \times$ (maximum F_{sclk_out})

Slave: $F_{spi\_clk} >= 6 \times$ (maximum F_{sclk_in})

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4/8/16 bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge. Fig.35-3 and Fig.35-4 show a timing diagram for a single SPI data transfer with SCPH = 0. The serial clock is shown for configuration parameters SCPOL = 0 and SCPOL = 1.

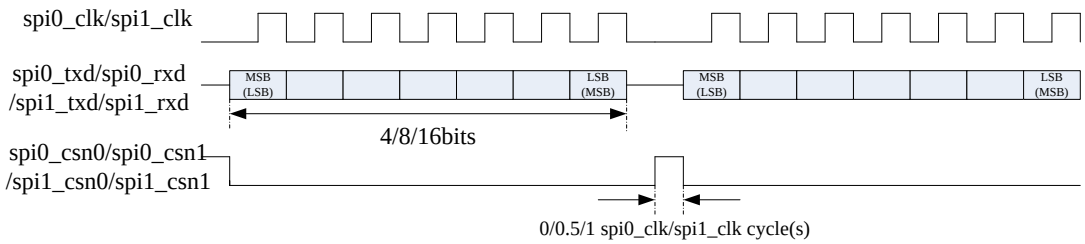


Fig. 35-3 SPI Format (SCPH=0 SCPOL=0)

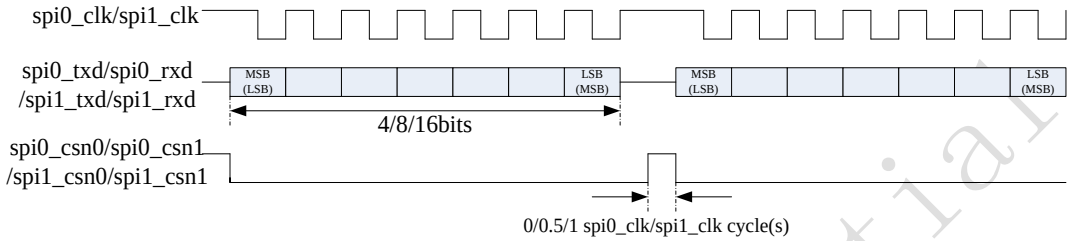


Fig. 35-4SPI Format (SCPH=0 SCPOL=1)

When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured. Fig.35-5 and Fig.35-6 show the timing diagram for the SPI format when the configuration parameter SCPH = 1.

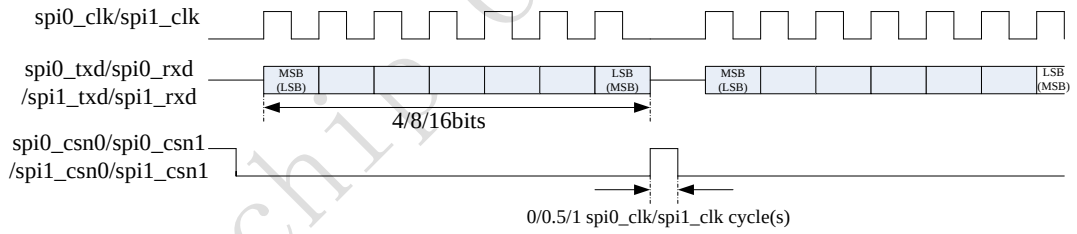


Fig. 35-5SPI Format (SCPH=1 SCPOL=0)

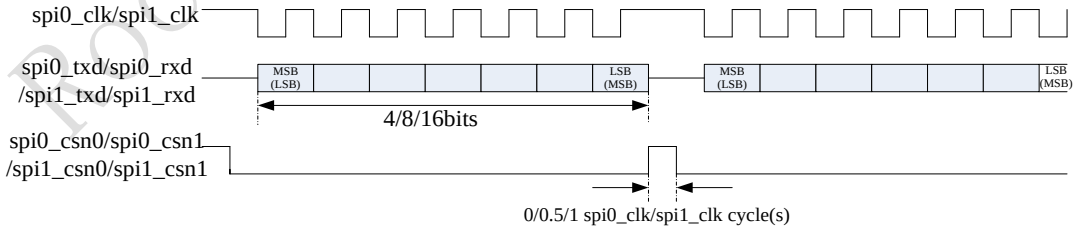


Fig. 35-6SPI Format (SCPH=1 SCPOL=1)

35.4 Register description

This section describes the control/status registers of the design. Pay attention that there are two SPI controllers in the chip: spi0 & spi1, so the base address in the following register descriptions can be either spi0 or spi1 base address.

35.4.1 Registers Summary

Name	Offset	Size	Reset value	Description
SPI_CTRLR0	0x0000	W	0x2	Control Register 0
SPI_CTRLR1	0x0004	W	0x0	Control Register 1
SPI_ENR	0x0008	W	0x0	SPI Enable Register
SPI_SER	0x000C	W	0x0	Slave Enable Register
SPI_BAUDR	0x0010	W	0x0	Baud Rate Select
SPI_TXFTLR	0x0014	W	0x0	Transmit FIFO Threshold Level
SPI_RXFTLR	0x0018	W	0x0	Receive FIFO Threshold Level
SPI_TXFLR	0x001C	W	0x0	Transmit FIFO Level Register
SPI_RXFLR	0x0020	W	0x0	Receive FIFO Level Register
SPI_SR	0x0024	W	0xC	Status Register
SPI_IPR	0x0028	W	0x0	Interrupt Polarity Register
SPI_IMR	0x002C	W	0x0	Interrupt Mask Register
SPI_ISR	0x0030	W	0x0	Interrupt Status Register
SPI_RISR	0x0034	W	0x1	Raw Interrupt Status Register
SPI_ICR	0x0038	W	0x0	Interrupt Clear Register
SPI_DMACR	0x003C	W	0x0	DMA Control Register
SPI_DMATDLR	0x0040	W	0x0	DMA Transmit Data Level
SPI_DMARDLR	0x0044	W	0x0	DMA Receive Data Level
SPI_TXDR	0x0400~0x07FC	W	0x0	Transmit FIFO Data Register
SPI_RXDR	0x0800~0x0BFC	W	0x0	Receive FIFO Data Register

Notes:
*Size: **B**– Byte (8 bits) access, **HW**– Half WORD (16 bits) access, **W**–WORD (32 bits) access*

35.4.2 Detail Register Description

SPI_CTRLR0
Address: Operational Base + offset(0x00)
Control register 0

Bit	Attr	Reset Value	Description
31:22	-	-	Reserved.
21	RW	0x0	Microwire Transfer Mode. Valid when frame format is set to National Semiconductors Microwire. 0: non-sequential transfer 1: sequential transfer
20	RW	0X0	Operation Mode. 0 –- Master Mode 1 –- Slave Mode
19:18	RW	0X0	Transfer Mode. 00 :Transmit & Receive 01 : Transmit Only 10 : Receive Only 11 :reserved
17:16	RW	0X0	Frame Format. 00: Motorola SPI

			01: Texas Instruments SSP 10: National Semiconductors Microwire 11: Reserved
15:14	RW	0X0	Rxd Sample Delay. When SPI is configured as a master, if the rxd data cannot be sampled by the sclk_out edge at the right time, this register should be configured to define the number of the spi_clk cycles after the active sclk_out edge to sample rxd data later when SPI works at high frequency. 00:do not delay 01:1 cycle delay 10:2 cycles delay 11:3 cycles delay
13	RW	0X0	Byte and Halfword Transform Valid when data frame size is 8bit. 0:apb 16bit write/read, spi 8bit write/read 1: apb 8bit write/read, spi 8bit write/read
12	RW	0X0	First Bit Mode. 0:first bit is MSB 1:first bit is LSB
11	RW	0X0	Endian Mode Serial endian mode can be configured by this bit. Apb endian mode is always little endian. 0:little endian 1:big endian
10	RW	0X0	ss_n to sclk_out delay Valid when the frame format is set to Motorola SPI and SPI used as a master. 0: the period between ss_n active and sclk_out active is half sclk_out cycles. 1: the period between ss_n active and sclk_out active is one sclk_out cycle.
9:8	RW	0X0	Chip Select Mode. Valid when the frame format is set to Motorola SPI and SPI used as a master. 00: ss_n keep low after every frame data is transferred. 01:ss_n be high for half sclk_out cycles after every frame data is transferred. 10: ss_n be high for one sclk_out cycle after every frame data is transferred. 11:reserved
7	RW	0X0	Serial Clock Polarity. Valid when the frame format is set to Motorola SPI. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high
6	RW	0X0	Serial Clock Phase. Valid when the frame format is set to Motorola SPI. 0 –Serial clock toggles in middle of first data bit 1– Serial clock toggles at start of first data bit
5:2	RW	0X0	Control Frame Size. Selects the length of the control word for the Microwire

			frame format. 0000~0010:reserved 0011:4-bit serial data transfer 0100:5-bit serial data transfer 0101:6-bit serial data transfer 0110:7-bit serial data transfer 0111:8-bit serial data transfer 1000:9-bit serial data transfer 1001:10-bit serial data transfer 1010:11-bit serial data transfer 1011:12-bit serial data transfer 1100:13-bit serial data transfer 1101:14-bit serial data transfer 1110:15-bit serial data transfer 1111:16-bit serial data transfer
1:0	RW	0X2	Data Frame Size. Selects the data frame length. 00–4bit data 01–8bit data 10–16bit data 11–reserved

SPI_CTRLR1
Address: Operational Base + offset(0x04)
Control register 1

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	RW	0x0	Number of Data Frames. When Transfer Mode is receive only, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer.

SPI_ENR
Address: Operational Base + offset(0x08)
SPI enable register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved.
0	RW	0x0	SPI Enable. Enables and disables all SPI operations. Transmit and receive FIFO buffers are cleared when the device is disabled.

SPI_SER
Address: Operational Base + offset(0x0C)
Slave enable register

Bit	Attr	Reset Value	Description
31:2	-	-	Reserved.
1:0	RW	0x0	Slave Select Enable Flag. This register is valid only when SPI is configured as a master device.

SPI_BAUDR
Address:operational base+offset(0x10)
Baud rate select

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	RW	0x0	SPI Clock Divider. This register is valid only when the SPI is configured as a master device. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fspi_clk/ SCKDV Where SCKDV is any even value between 2 and 65534. For example: for Fspi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2= 1.8432MHz

SPI_TXFTLR
Address: Operational Base + offset(0x14)
Transmit FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4:0	RW	0x0	Transmit FIFO Threshold. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

SPI_RXFTLR
Address: Operational Base + offset(0x18)
Receive FIFO Threshold Level

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4:0	RW	0x0	Receive FIFO Threshold. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

SPI_TXFLR
Address: Operational Base + offset(0x1C)
Transmit FIFO Level Register

Bit	Attr	Reset Value	Description
31:6	-	-	Reserved.
5:0	R	0x0	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

SPI_RXFLR
Address: Operational Base + offset(0x20)
Receive FIFO Level Register

Bit	Attr	Reset Value	Description

31:6	-	-	Reserved.
5:0	R	0x0	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

SPI_SR
Address: Operational Base + offset(0x24)
Status Register

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4	R	0x0	Receive FIFO Full. 0 – Receive FIFO is not full 1 – Receive FIFO is full
3	R	0x1	Receive FIFO Empty. 0 – Receive FIFO is not empty 1 – Receive FIFO is empty
2	R	0x1	Transmit FIFO Empty. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty
1	R	0x0	Transmit FIFO Full. 0 – Transmit FIFO is not full 1 – Transmit FIFO is full
0	R	0x0	SPI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. 0 – SPI is idle or disabled 1 – SPI is actively transferring data

SPI_IPR
Address: Operational Base + offset(0x28)
Interrupt Polarity Register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved.
0	RW	0x0	Interrupt Polarity Register 0:Active Interrupt Polarity Level is HIGH 1: Active Interrupt Polarity Level is LOW

SPI_IMR
Address: Operational Base + offset(0x2C)
Interrupt Mask Register

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4	RW	0x0	Receive FIFO Full Interrupt Mask 0 – spi_rxf_intr interrupt is masked 1 – spi_rxf_intr interrupt is not masked
3	RW	0x0	Receive FIFO Overflow Interrupt Mask 0 – spi_rxo_intr interrupt is masked 1 – spi_rxo_intr interrupt is not masked
2	RW	0x0	Receive FIFO Underflow Interrupt Mask 0 – spi_rxu_intr interrupt is masked 1 – spi_rxu_intr interrupt is not masked
1	RW	0x0	Transmit FIFO Overflow Interrupt Mask 0 – spi_txo_intr interrupt is masked

			1 – spi_txo_intr interrupt is not masked
0	RW	0x0	Transmit FIFO Empty Interrupt Mask 0 – spi_txe_intr interrupt is masked 1 – spi_txe_intr interrupt is not masked

SPI_ISR
Address: Operational Base + offset(0x30)
Interrupt Status Register

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4	R	0x0	Receive FIFO Full Interrupt Status 0 = spi_rxf_intr interrupt is not active after masking 1 = spi_rxf_intr interrupt is full after masking
3	R	0x0	Receive FIFO Overflow Interrupt Status 0 = spi_rxo_intr interrupt is not active after masking 1 = spi_rxo_intr interrupt is active after masking
2	R	0x0	Receive FIFO Underflow Interrupt Status 0 = spi_rxu_intr interrupt is not active after masking 1 = spi_rxu_intr interrupt is active after masking
1	R	0x0	Transmit FIFO Overflow Interrupt Status 0 = spi_txo_intr interrupt is not active after masking 1 = spi_txo_intr interrupt is active after masking
0	R	0x0	Transmit FIFO Empty Interrupt Status 0 = spi_txe_intr interrupt is not active after masking 1 = spi_txe_intr interrupt is active after masking

SPI_RISR
Address: Operational Base + offset(0x34)
Raw Interrupt Status Register

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4	R	0x0	Receive FIFO Full Raw Interrupt Status 0 = spi_rxf_intr interrupt is not active prior to masking 1 = spi_rxf_intr interrupt is full prior to masking
3	R	0x0	Receive FIFO Overflow Raw Interrupt Status 0 = spi_rxo_intr interrupt is not active prior to masking 1 = spi_rxo_intr interrupt is active prior to masking
2	R	0x0	Receive FIFO Underflow Raw Interrupt Status 0 = spi_rxu_intr interrupt is not active prior to masking 1 = spi_rxu_intr interrupt is active prior to masking
1	R	0x0	Transmit FIFO Overflow Raw Interrupt Status 0 = spi_txo_intr interrupt is not active prior to masking 1 = spi_txo_intr interrupt is active prior to masking
0	R	0x1	Transmit FIFO Empty Raw Interrupt Status 0 = spi_txe_intr interrupt is not active prior to masking 1 = spi_txe_intr interrupt is active prior to masking

SPI_ICR
Address: Operational Base + offset(0x38)
Interrupt Clear Register

Bit	Attr	Reset Value	Description
31:4	-	-	Reserved.
3	W	0x0	Clear Transmit FIFO Overflow Interrupt.
2	W	0x0	Clear Receive FIFO Overflow Interrupt.
1	W	0x0	Clear Receive FIFO Underflow Interrupt.
0	W	0x0	Clear Combined Interrupt.

SPI_DMACR
Address: Operational Base + offset(0x3C)
DMA Control Register

Bit	Attr	Reset Value	Description
31:2	-	-	Reserved.
1	RW	0x0	Transmit DMA Enable. 0 = Transmit DMA disabled 1 = Transmit DMA enabled
0	RW	0x0	Receive DMA Enable. 0 = Receive DMA disabled 1 = Receive DMA enabled

SPI_DMATDLR
Address: Operational Base + offset(0x40)
DMA Transmit Data Level

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4:0	RW	0x0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and Transmit DMA Enable (DMACR[1]) = 1.

SPI_DMARDLR
Address: Operational Base + offset(0x44)
DMA Receive Data Level

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved.
4:0	RW	0x0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and Receive DMA Enable(DMACR[0])=1.

SPI_TXDR
Address: Operational Base + offset(0x400~0x7FC)
Transimt FIFO Data Register

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	W	0x0	Transimt FIFO Data Register. When it is written to, data are moved into the transmit FIFO.

SPI_RXDR

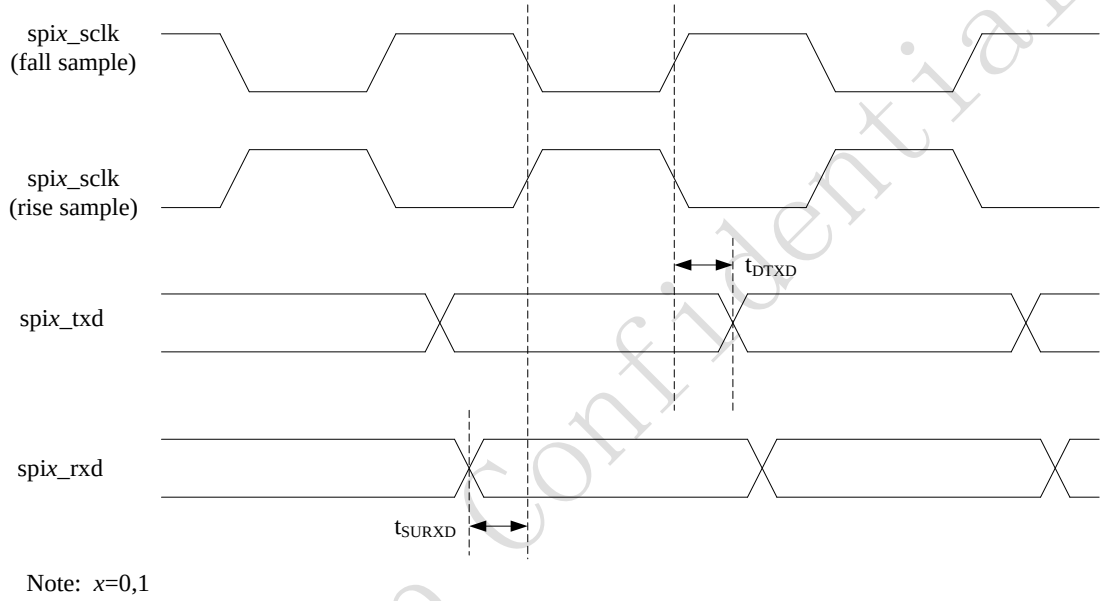
Address: Operational Base + offset(0x800~0xBFC)

Receive FIFO Data Register

Bit	Attr	Reset Value	Description
31:16	-	-	Reserved.
15:0	R	0x0	Receive FIFO Data Register. When theregister is read, data in the receive FIFO is accessed.

*Notes: Attr: **RW**– Read/writable, **R**– read only, **W**– write only*

35.5 Timing Diagram



Note: *x*=0,1

Fig. 35-7SPI controller timing diagram

Table 35-1Meaning of the parameter in Fig.35-7

Parameter	Description	min	typ	max	unit
t_{DTXD}	spix_txd propagation delay from spix_sclk drive edge	-0.639	-1.057	-1.694	ns
t_{SURXD}	spix_rxd setup time to spix_sclk sample edge	5.846	8.490	12.204	ns

Note:x=0,1

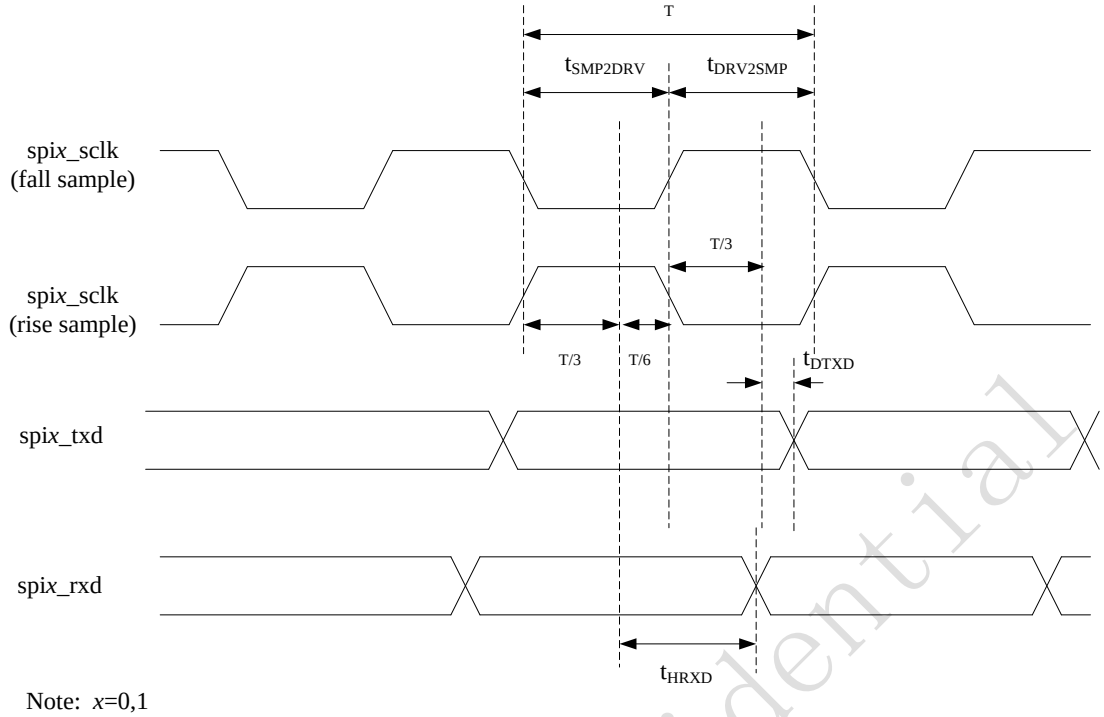


Fig. 35-8SPI controller timing diagram in slave mode

Table 35-2Meaning of the parameter in Fig.35-8

Parameter	Description	min	typ	max	unit
T	spix_sclk cycle time (cannot be less than 60 ns)	-	-	-	ns
$t_{SMP2DRV}$	spix_sclk pulse width from sample edge to drive edge (cannot be less than 30 ns)	-	-	-	ns
$t_{DRV2SMP}$	spix_sclk pulse width from drive edge to sample edge (cannot be less than 30 ns)	-	-	-	ns
t_{DTXD}	spix_txd propagation delay from T/3 after spix_sclk drive edge	6.771	10.146	14.691	ns
t_{HRXD}	spix_rxd hold time from T/3 after spix_sclk sample edge	0.238	0.489	0.676	ns

Note:x=0,1

35.6 Interface description

Table 35-3SPI interface description in master mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_clk	I/O	GPIO1_A[5]	GRF_GPIO1A_IOMUX[11:10]=10
spi0_csn0	I/O	GPIO1_A[4]	GRF_GPIO1A_IOMUX[9:8]=10
spi0_txd	O	GPIO1_A[7]	GRF_GPIO1A_IOMUX[15:14]=10
spi0_rxd	I	GPIO1_A[6]	GRF_GPIO1A_IOMUX[13:12]=10
spi1_clk	I/O	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=10
spi1_csn0	I/O	GPIO2_C[4]	GRF_GPIO2C_IOMUX[9:8]=10
spi1_txd	O	GPIO2_C[5]	GRF_GPIO2C_IOMUX[11:10]=10
spi1_rxd	I	GPIO2_C[6]	GRF_GPIO2C_IOMUX[13:12]=10
spi1_csn1	O	GPIO2_C[7]	GRF_GPIO2C_IOMUX[15:14]=10
spi0_csn1	O	GPIO4_B[7]	GRF_GPIO4B_IOMUX[14]=1

Note: spi0_csn1, spi1_csn1 can only be used in master mode

Table 35-4SPI interface description in slave mode

Module Pin	Direction	Pad Name	IOMUX Setting
spi0_clk	I/O	GPIO1_A[5]	GRF_GPIO1A_IOMUX[11:10]=10
spi0_csn0	I/O	GPIO1_A[4]	GRF_GPIO1A_IOMUX[9:8]=10
spi0_txd	O	GPIO1_A[7]	GRF_GPIO1A_IOMUX[15:14]=10
spi0_rxd	I	GPIO1_A[6]	GRF_GPIO1A_IOMUX[13:12]=10
spi1_clk	I/O	GPIO2_C[3]	GRF_GPIO2C_IOMUX[7:6]=10
spi1_csn0	I/O	GPIO2_C[4]	GRF_GPIO2C_IOMUX[9:8]=10
spi1_txd	O	GPIO2_C[5]	GRF_GPIO2C_IOMUX[11:10]=10
spi1_rxd	I	GPIO2_C[6]	GRF_GPIO2C_IOMUX[13:12]=10

35.7 Application Notes

- **Clock Ratios**

A summary of the frequency ratio restrictions between the bit-rate clock (sclk_out/sclk_in) and the SPI peripheral clock (spi_clk) are described as:

Master: F_{spi_clk}>= 2 × (maximum F_{sclk_out})

Slave: F_{spi_clk}>= 6 × (maximum F_{sclk_in})

- **Master Transfer Flow**

When configured as a serial-master device, the SPI initiates and controls all serial transfers. The serial bit-rate clock, generated and controlled by the SPI, is driven out on the sclk_out line. When the SPI is disabled (SPI_ENR = 0), no serial transfers can occur and sclk_out is held in "inactive" state, as defined by the serial protocol under which it operates.

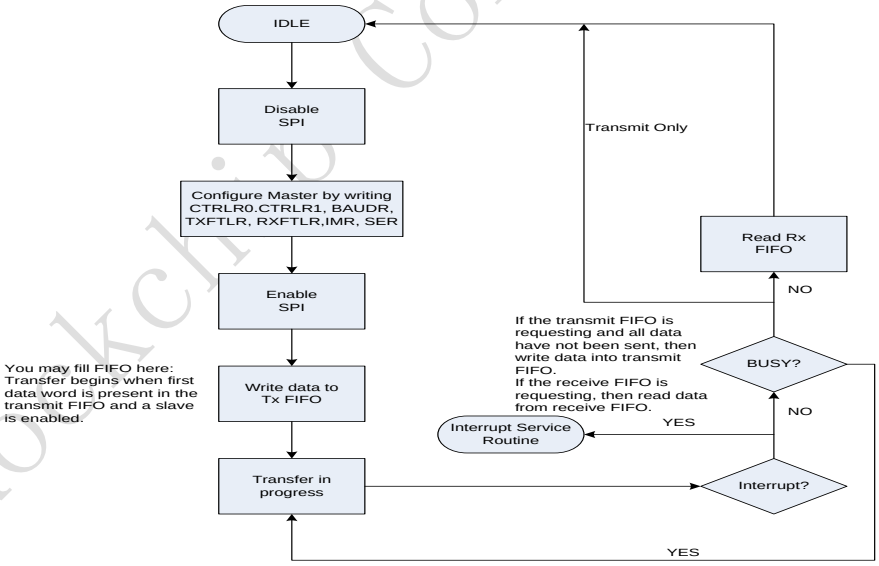


Fig. 35-9SPI Master transfer flow diagram

- **Slave Transfer Flow**

When the SPI is configured as a slave device, all serial transfers are initiated and controlled by the serial bus master.

When the SPI serial slave is selected during configuration, it enables its txd data onto the serial bus. All data transfers to and from the serial slave are regulated on the serial clock line (sclk_in), driven from the serial-master device. Data are propagated from the serial slave on one edge of the serial clock line and sampled on the opposite edge.

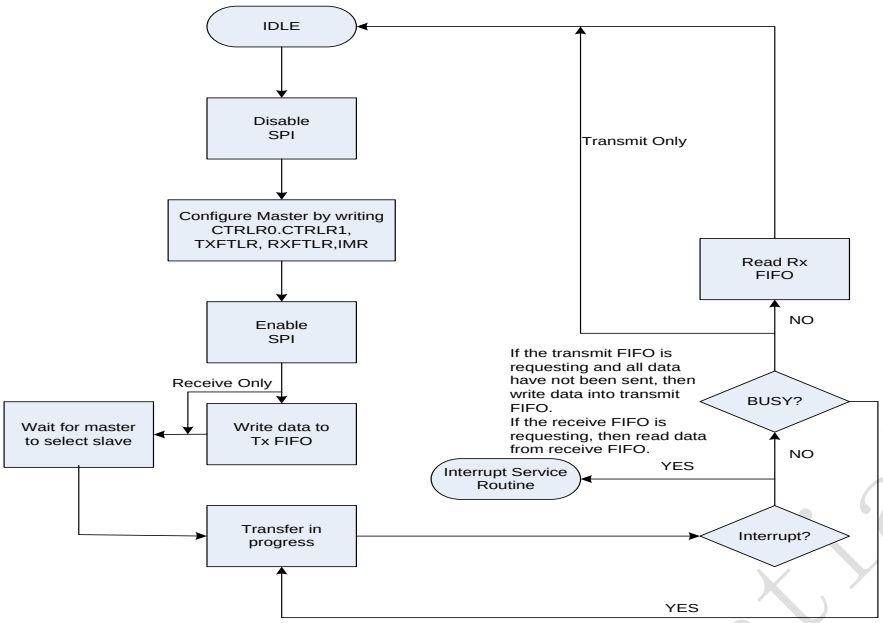


Fig. 35-10SPI Slave transfer flow diagram

Chapter 36 UART Interface

36.1 Overview

The Universal Asynchronous Receiver/Transmitter (UART) is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

36.1.1 Features

- AMBA APB interface – Allows for easy integration into a Synthesizable Components for AMBA 2 implementation.
- Support interrupt interface to interrupt controller.
- UART1/UART2/UART3 contain two 32Bytes FIFOs for data receive and transmit, UART0's two embedded FIFOs are both 64Bytes for BT transfer.
- Programmable serial data baud rate as calculated by the following:baud rate = (serial clock frequency)/(16×divisor).
- UART0 / UART1 / UART3 support auto flow-control,UART2 do not support auto flow-control.
- UART0 is in cpu system, UART1 is in alive system, UART2 / UART3 are in peri system.

36.2 Block Diagram

This section provides a description about the functions and behavior under various conditions.

The UART comprises with:

- AMBA APB interface
- FIFO controllers
- Register block
- Modem synchronization block and baud clock generation block
- Serial receiver and serial transmitter

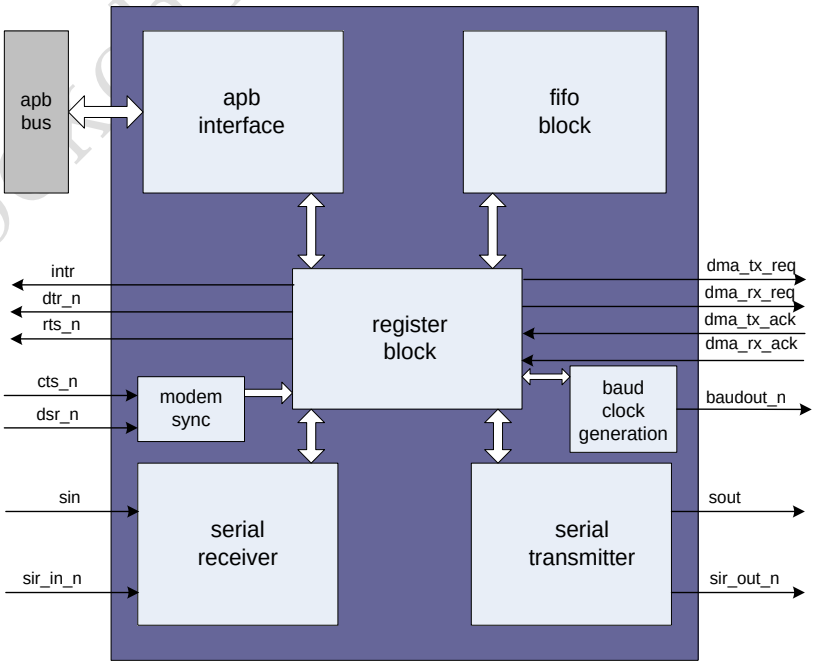


Fig. 36-1UART Architecture

APB INTERFACE

The host processor accesses data, control, and status information on the UARTthrough the APB interface. The UARTsupports APB data bus widths of 8, 16, and 32 bits.

Register block

Be responsible for the main UART functionality including control, status and interrupt generation.

Modem Synchronization block

Synchronizes the modem input signal.

FIFO block

Be responsible for FIFO control and storage (when using internal RAM) or signaling to control external RAM (when used).

Baud Clock Generator

Produces the transmitter and receiver baud clock along with the output reference clock signal (baudout_n).

Serial Transmitter

Converts the parallel data, written to the UART, into serial form and adds all additional bits, as specified by the control register, for transmission. This makeup of serial data, referred to as a character can exit the block in two forms, either serial UART format or IrDA 1.0 SIR format.

Serial Receiver

Converts the serial data character (as specified by the control register) received in either the UART or IrDA 1.0 SIR format to parallel form. Parity error detection, framing error detection and line break detection is carried out in this block.

36.3 Function description

- UART (RS232) Serial Protocol

Because the serial communication is asynchronous, additional bits (start and stop) are added to the serial data to indicate the beginning and end. An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to perform simple error checking on the received data, as shown in Figure.

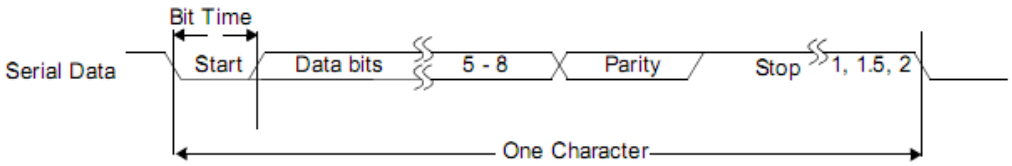


Fig. 36-2UART Serial protocol

- Baud Clock

The baud rate controlled by the serial clock (sclk or pclk in a single clock implementation) and the Divisor Latch Register (DLH and DLL). As the exact number of baud clocks that each bit was transmitted for is known, calculating the mid point for sampling is not difficult, that is every 16 baud clocks after the

mid point sample of the start bit.

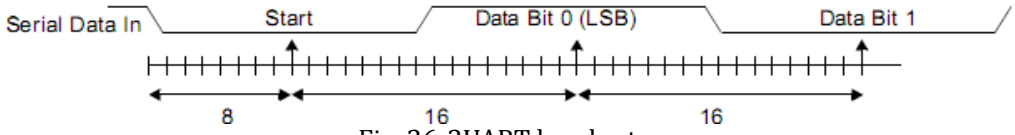


Fig. 36-3UART baud rate

- FIFO Support

1.NONE FIFO MODE

If FIFO support is not selected, then no FIFOs are implemented and only a single receive data byte and transmit data byte can be stored at a time in the RBR and THR.

2.FIFO MODE

The FIFO depth is 32,enabled by register FCR[0].

- Interrupts

The following interrupt types can be enabled with the IER register.

Receiver Error;

Receiver Data Available;

Character Timeout (in FIFO mode only);

Transmitter Holding Register Empty at/below threshold (in Programmable THRE Interrupt mode);

Modem Status;

- DMA Support

The uart supports DMA signaling with the use of two output signals (dma_tx_req_n and dma_rx_req_n) to indicate when data is ready to be read or when the transmit FIFO is empty.

The dma_tx_req_n signal is asserted under the following conditions:

a) When the Transmitter Holding Register is empty in non-FIFO mode.

b) When the transmitter FIFO is empty in FIFO mode with ProgrammableTHRE interrupt mode disabled.

c) When the transmitter FIFO is at, or below the programmed threshold with Programmable THRE interrupt mode enabled.

The dma_rx_req_n signal is asserted under the following conditions:

a) When there is a single character available in the Receive Buffer Register in non-FIFO mode.

b) When the Receiver FIFO is at or above the programmed trigger level in FIFO mode.

- Auto Flow Control

The uart can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available. If FIFOs are not implemented, then this mode cannot be selected. When Auto Flow Control mode has been selected it can be enabled with the Modem Control Register (MCR[5]). Following figure shows a block diagram of the Auto Flow Control functionality.

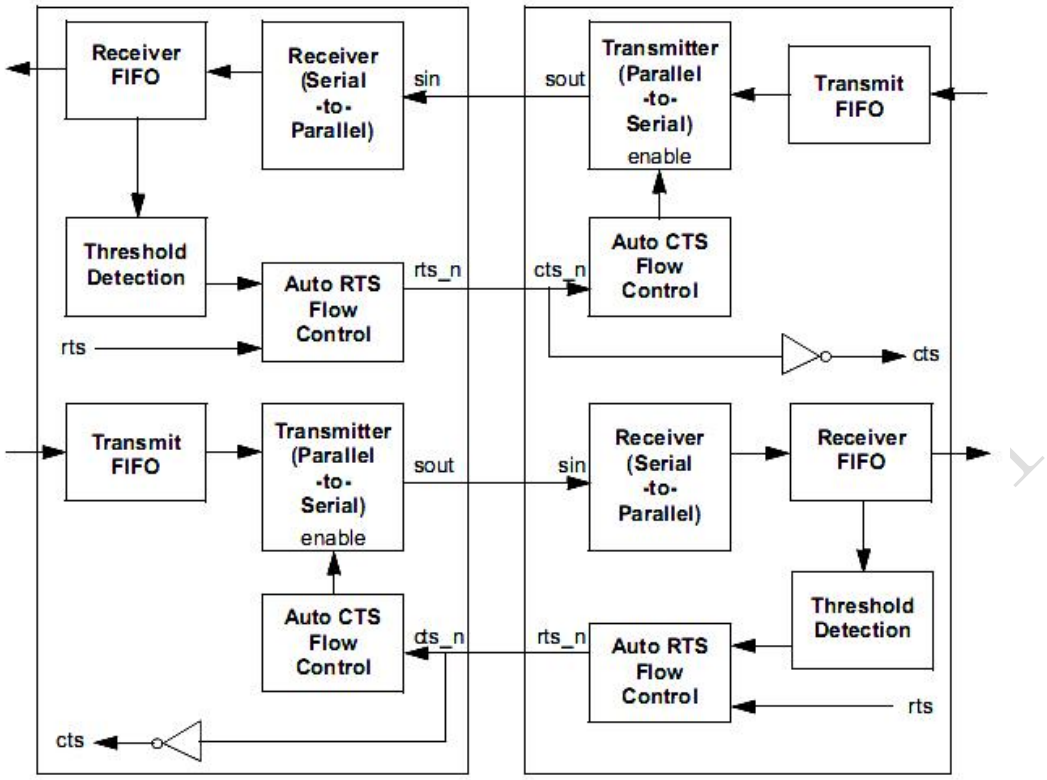


Fig. 36-4UART Auto flow control block diagram

Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

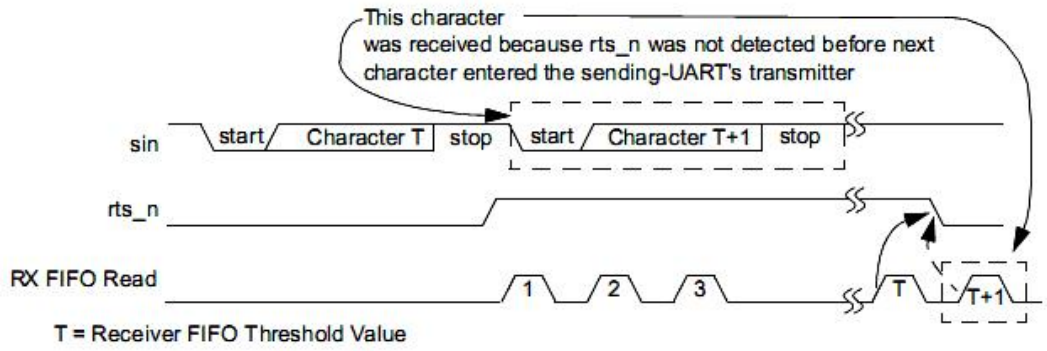


Fig. 36-5UART AUTO RTS TIMING

Auto CTS – becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit is set)
- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit is not set)

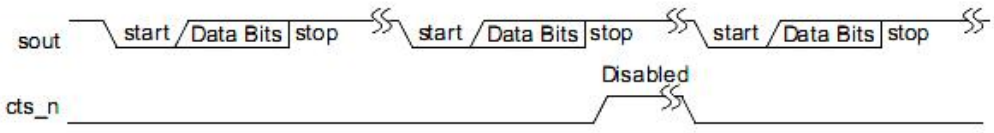


Fig. 36-6 UART AUTO CTS TIMING

36.4 Register description

There are 4 UARTs in RK30xx, and each one has its own base address.

36.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
UART_RBR	0x0000	W	0x0000_0000	Receive Buffer Register
UART_THR				Transmit Holding Register
UART_DLL				Divisor Latch (Low)
UART_DLH	0x0004	W	0x0000_0000	Divisor Latch (High)
UART_IER				Interrupt Enable Register
UART_IIR	0x0008	W	0x0000_0000	Interrupt Identification Register
UART_FCR				FIFO Control Register
UART_LCR	0x000C	W	0x0000_0000	Line Control Register
UART_MCR	0x0010	W	0x0000_0000	Modem Control Register
UART_LSR	0x0014	W	0x0000_0060	Line Status Register
UART_MSR	0x0018	W	0x0000_0000	Modem Status Register
UART_SCR	0x001c	W	0x0000_0000	Scratchpad Register
Reserved	0x0020 -2C	W	0x0000_0000	--
UART_SRBR	0x0030 -6C	W	0x0000_0000	Shadow Receive Buffer Register
UART_STHR		W	0x0000_0000	Shadow Transmit Holding Register
UART_FAR	0x0070	W	0x0000_0000	FIFO Access Register
UART_TFR	0x0074	W	0x0000_0000	Transmit FIFO Read
UART_RFW	0x0078	W	0x0000_0000	Receive FIFO Write
UART_USR	0x007C	W	0x0000_0006	UART Status Register
UART_TFL	0x0080	W	0x0000_0000	Transmit FIFO Level
UART_RFL	0x0084	W	0x0000_0000	Receive FIFO Level
UART_SRR	0x0088	W	0x0000_0000	Software Reset Register
UART_SRTS	0x008C	W	0x0000_0000	Shadow Request to Send
UART_SBCR	0x0090	W	0x0000_0000	Shadow Break Control Register
UART_SDMAM	0x0094	W	0x0000_0000	Shadow DMA Mode
UART_SFE	0x0098	W	0x0000_0000	Shadow FIFO Enable
UART_SRT	0x009C	W	0x0000_0000	Shadow RCVR Trigger
UART_STET	0x00A0	W	0x0000_0000	Shadow TX Empty Trigger
UART_HTX	0x00A4	W	0x0000_0000	Halt TX
UART_DMASA	0x00A8	W	0x0000_0000	DMA Software Acknowledge
Reserved	0x00AC -F0	W	0x0000_0000	--
UART_CPR	0x00F4	W	0x0000_0000	Component Parameter Register
UART_UCV	0x00F8	W	0x3330_372a	UART Component Version
UART_CTR	0x00FC	W	0x4457_0110	Component Type Register

Notes:
*Size: **B**– Byte (8 bits) access, **HW**– Half WORD (16 bits) access, **W**–WORD (32 bits) access*

36.4.2 Registers detail description

UART_RBR
Address: Operational Base + offset(0x00)
Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs.

UART_THR
Address: Operational Base + offset(0x00)
Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

UART_DLL
Address: Operational Base + offset(0x00)
Divisor Latch (Low)

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	Lower 8-bits of a 16-bit, read/write, Divisor Latch

			register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest Uart clock should be allowed to pass before transmitting or receiving data.

UART_DLH
Address: Operational Base + offset(0x04)
Divisor Latch (High)

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.

UART_IER
Address: Operational Base + offset(0x04)
Interrupt Enable Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7	RW	0x0	Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0 = disabled 1 = enabled
6:4	-	-	Reserved
3	RW	0X0	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0 = disabled 1 = enabled
2	RW	0X0	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0 = disabled 1 = enabled
1	RW	0X0	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. 0 = disabled 1 = enabled
0	RW	0X0	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data

			Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0 = disabled 1 = enabled

UART_IIR
Address: Operational Base + offset(0x08)
Interrupt Identification Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:6	R	0x01	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. 00 = disabled 11 = enabled
5:4	-	-	Reserved
3:0	R	0x01	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: 0000 = modem status 0001 = no interrupt pending 0010 = THR empty 0100 = received data available 0110 = receiver line status 0111 = busy detect 1100 = character timeout

UART_FCR
Address: Operational Base + offset(0x08)
FIFO Control Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:6	W	0x0	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO ¼ full 10 = FIFO ½ full 11 = FIFO 2 less than full
5:4	W	0x0	TX Empty Trigger. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO

			10 = FIFO ¼ full 11 = FIFO ½ full
3	W	0x0	DMA Mode. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected . 0 = mode 0 1 = mode 11100 = character timeout
2	W	0x0	XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected . Note that this bit is'self-clearing'. It is not necessary to clear this bit.
1	W	0x0	RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected . Note that this bit is 'self-clearing'. It is not necessary to clear this bit
0	W	0x0	FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.

UART_LCR

Address: Operational Base + offset(0x0C)

Line Control Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7	RW	0x0	Divisor Latch Access Bit.Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers
6	RW	0x0	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If MCR[6] set to one, the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to thereceiver and the sir_out_n line is forced low
5	-	-	Reserved
4	RW	0x0	Even Parity Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or

			checked.
3	RW	0x0	Parity Enable. Writeable only when UART is not busy (USR[0] is zero), always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled
2	RW	0x0	Number of stop bits. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data.If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, twostop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit
1:0	RW	0x0	Data Length Select. Writeable only when UART is not busy (USR[0] is zero), always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

UART_MCR

Address: Operational Base + offset(0x10)

Modem Control Register

Bit	Attr	Reset Value	Description
31:7	-	-	Reserved
6	RW	0x0	SIR Mode Enable. This is used to enable/disable the IrDA SIR Mode . 0 = IrDA SIR Mode disabled 1 = IrDA SIR Mode enabled
5	RW	0x0	Auto Flow Control Enable. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled
4	RW	0x0	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes.
3	RW	0x0	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0)
2	RW	0x0	OUT1

1	RW	0x0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data
0	RW	0x0	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0)

UART_LSR

Address: Operational Base + offset(0x14)

Line Status Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7	R	0x0	Receiver FIFO Error bit. This bit is relevant FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO
6	R	0x1	Transmitter Empty bit. If FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.
5	R	0x1	Transmit Holding Register Empty bit. If THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If IER[7] set to one and FCR[0] set to one respectively, the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting.
4	R	0x0	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data.
3	R	0x0	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.
2		0x0	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.
1	R	0x0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new

			data character was received before the previous data was read.
0	R	0x0	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready

UART_MSR
Address: Operational Base + offset(0x18)
Modem Status Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7	R	0x0	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n.
6	R	0x0	Ring Indicator. This is used to indicate the current state of the modem control line ri_n.
5	R	0x0	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n.
4	R	0x0	Clear to Send. This is used to indicate the current state of the modem control line cts_n.
3	R	0x0	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.
2	R	0x0	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.
1	R	0x0	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.
0	R	0x0	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.

UART_SCR
Address: Operational Base + offset(0x1C)
Scratchpad Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	RW	0x0	This register is for programmers to use as a temporary storage space.

UART_SRBR
Address: Operational Base + offset(0x30-6C)
Shadow Receive Buffer Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	R	0x0	This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial

			input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs

UART_STHR

Address: Operational Base + offset(0x30-6C)

Shadow Transmit Holding Register

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	R	0x0	This is a shadow register for the THR.

UART_FAR

Address: Operational Base + offset(0x70)

FIFO Access Register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled

UART_TFR

Address: Operational Base + offset(0x74)

Transmit FIFO Read

Bit	Attr	Reset Value	Description
31:8	-	-	Reserved
7:0	R	0x0	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR.

UART_RFW

Address: Operational Base + offset(0x78)
Receive FIFO Write

Bit	Attr	Reset Value	Description
31:10	-	-	Reserved
9	W	0x0	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
8	W	0x0	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).
7:0	W	0x0	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs not enabled, the data that is written to the RFWD is pushed into the RBR.

UART_USR
Address: Operational Base + offset(0x7C)
UART Status Register

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved
4	R	0x0	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full
3	R	0x0	Receive FIFO Not Empty. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty
2	R	0x1	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty
1	R	0x1	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	R	0x0	UART Busy. This is indicates that a serial transfer is in progress, when cleared indicates that the uart is idle or inactive. 0 = Uart is idle or inactive 1 = Uart is busy (actively transferring data)

UART_TFL
Address: Operational Base + offset(0x80)
Transmit FIFO Level

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved
4:0	R	0x0	Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO.

UART_RFL
Address: Operational Base + offset(0x84)
Receive FIFO Level

Bit	Attr	Reset Value	Description
31:5	-	-	Reserved
4:0	R	0x0	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

UART_SRR
Address: Operational Base + offset(0x88)
Software Reset Register

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2	W	0x0	XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]).
1	W	0x0	RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]).
0	W	0x0	UART Reset. This asynchronously resets the Uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset.

UART_SRTS
Address: Operational Base + offset(0x8C)
Shadow Request to Send

Bit	Attr	Reset Value	Description
31:1	-	-	
0	RW	0x0	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR.

UART_SBCR
Address: Operational Base + offset(0x90)
Shadow Break Control Register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR.

UART_SDMAM
Address: Operational Base + offset(0x94)
Shadow DMA Mode

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	Shadow DMA Mode. This is a shadow register for the

			DMA mode bit (FCR[3]).

UART_SFE
Address: Operational Base + offset(0x98)
Shadow FIFO Enable

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]).

UART_SRT
Address: Operational Base + offset(0x9C)
Shadow RCVR Trigger

Bit	Attr	Reset Value	Description
31:2	-	-	Reserved
1:0	RW	0x0	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]).

UART_STET
Address: Operational Base + offset(0Xa0)
Shadow TX Empty Trigger

Bit	Attr	Reset Value	Description
31:2	-	-	Reserved
1:0	RW	0x0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]).

UART_HTX
Address: Operational Base + offset(0Xa4)
Halt TX

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x0	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled

UART_DMASA
Address: Operational Base + offset(0xa8)
RTC counter reset register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	W	0x0	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition.

UART_UCV
Address: Operational Base + offset(0xf8)
UART Component Version

Bit	Attr	Reset Value	Description
31:0	R	0x330372a	ASCII value for each number in the version

UART_CTR
Address: Operational Base + offset(0xfc)
Component Type Register

Bit	Attr	Reset Value	Description
31:0	R	0x44570110	This register contains the peripherals identification code.

*Notes: Attr: **RW** – Read/writable, **R** – read only, **W** – write only*

36.5 Interface description

Table 36-1 UART Interface Description

Module pin	Direction	Pad name	IOMUX
UART0 Interface			
uart0_sin	I	GPIO1_A[0]	GRF_GPIO1A_IOMUX[0]=1
uart0_sout	O	GPIO1_A[1]	GRF_GPIO1A_IOMUX[2]=1
uart0_cts_n	I	GPIO1_A[2]	GRF_GPIO1A_IOMUX[4]=1
uart0_rts_n	O	GPIO1_A[3]	GRF_GPIO1A_IOMUX[6]=1
UART1 Interface			
uart1_sin	I	GPIO1_A[4]	GRF_GPIO1A_IOMUX[9:8]=01
uart1_sout	O	GPIO1_A[5]	GRF_GPIO1A_IOMUX[11:10]=01
uart1_cts_n	I	GPIO1_A[6]	GRF_GPIO1A_IOMUX[13:12]=01
uart1_rts_n	O	GPIO1_A[7]	GRF_GPIO1A_IOMUX[15:14]=01
UART2 Interface			
uart2_sin	I	GPIO1_B[0]	GRF_GPIO1B_IOMUX[0]=1
uart2_sout	O	GPIO1_B[1]	GRF_GPIO1B_IOMUX[2]=1
UART3 Interface			
uart3_sin	I	GPIO3_D[3]	GRF_GPIO3D_IOMUX[6]=1
uart3_sout	O	GPIO3_D[4]	GRF_GPIO3D_IOMUX[8]=1
uart3_cts_n	I	GPIO3_D[5]	GRF_GPIO3D_IOMUX[10]=1
uart3_rts_n	O	GPIO3_D[6]	GRF_GPIO3D_IOMUX[12]=1

36.6 Application Notes

36.6.1 None FIFO Mode Transfer Flow

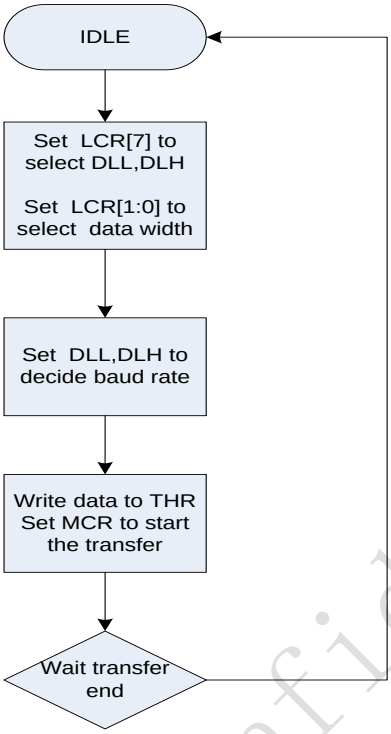


Fig. 36-7UARTnone fifo mode

36.6.2 FIFO Mode Transfer Flow

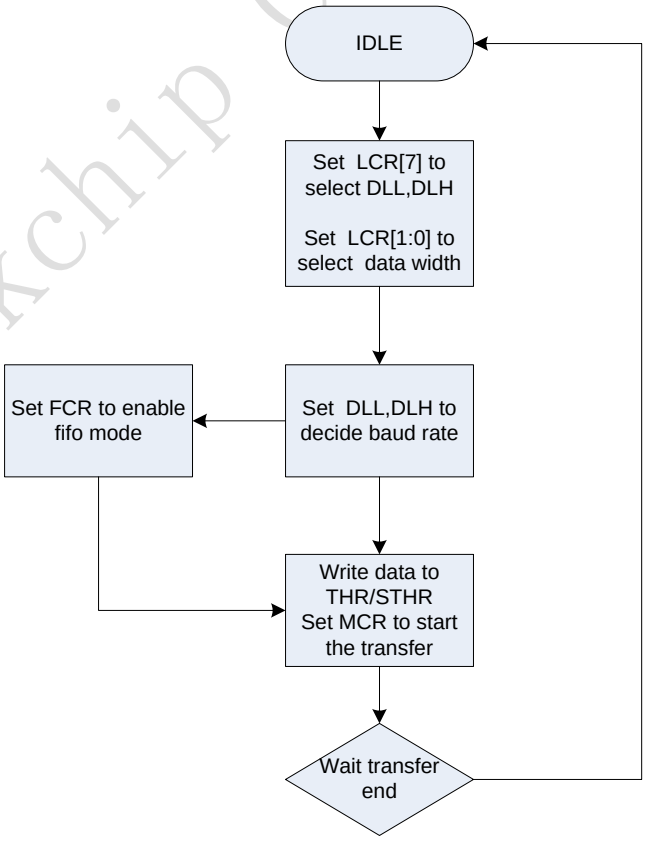


Fig. 36-8UART fifo mode

The UART is an APB slave performing:

Serial-to-parallel conversion on data received from a peripheral device.

Parallel-to-serial conversion on data transmitted to the peripheral device.

The CPU reads and writes data and control/status information through the APB interface. The transmitting and receiving paths are buffered with internal FIFO memories enabling up to 32-bytes to be stored independently in both transmit and receive modes. A baud rate generator can generate a common transmit and receive internal clock input. The baud rates will depend on the internal clock frequency. The UART will also provide transmit, receive and exception interrupts to system. A DMA interface is implemented for improving the system performance.

36.6.3 Baud Rate Calculation

- UART clock generation

Fig.36-10 shows the UART clock generation.

UART source clocks can be selected from CODEC PLL and GENERAL PLL outputs.

UART clocks can be generated by 1 to 64 division of its source clock , or can be fractionally divided again , or be provided by XIN24M.

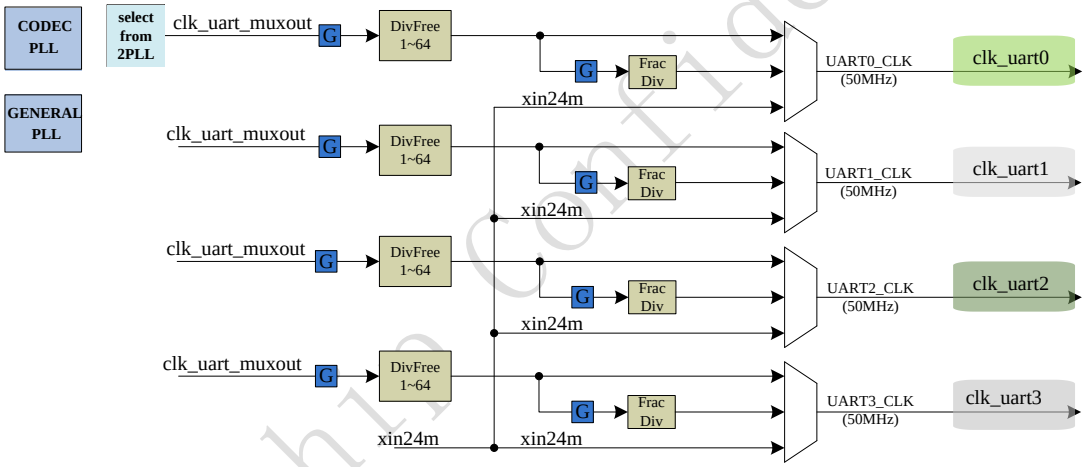


Fig. 36-9 UART clock generation

- UART baud rate configuration

Table 36-2 provides some reference configuration for different UART baud rates.

Table 36-2 UART baud rate configuration

Baud Rate	Reference Configuration
115.2 Kbps	Config GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Config UART_DLL to 8.
460.8 Kbps	Config GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Config UART_DLL to 2.
921.6 Kbps	Config GENERAL PLL to get 648MHz clock output; Divide 648MHz clock by 1152/50625 to get 14.7456MHz clock; Config UART_DLL to 1.
1.5 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 16 to get 24MHz clock; Config UART_DLL to 1

3 Mbps	Choose GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 8 to get 48MHz clock; Config UART_DLL to 1
4 Mbps	Config GENERAL PLL to get 384MHz clock output; Divide 384MHz clock by 6 to get 64MHz clock; Config UART_DLL to 1

Chapter 37 I2C Interface

37.1 Overview

The Inter-Integrated Circuit (I2C) is a two wired (SCL and SDA), bi-directional serial bus that provides an efficient and simple method of information exchange between devices. This I2C bus controller supports master mode acting as a bridge between AMBA protocol and generic I2C bus system.

37.1.1 Features

- Item Compatible with I2C-bus
- AMBA APB slave interface
- Supports master mode of I2C bus
- Software programmable clock frequency and transfer rate up to 400Kbit/sec
- Supports 7 bits and 10 bits addressing modes
- Interrupt or polling driven multiple bytes data transfer
- Clock stretching and wait state generation
- I2C0/I2C1 is in cpu system, I2C2 /I2C3 /I2C4 are in peri system

37.2 Block Diagram

This chapter provides a description about the functions and behavior under various conditions.

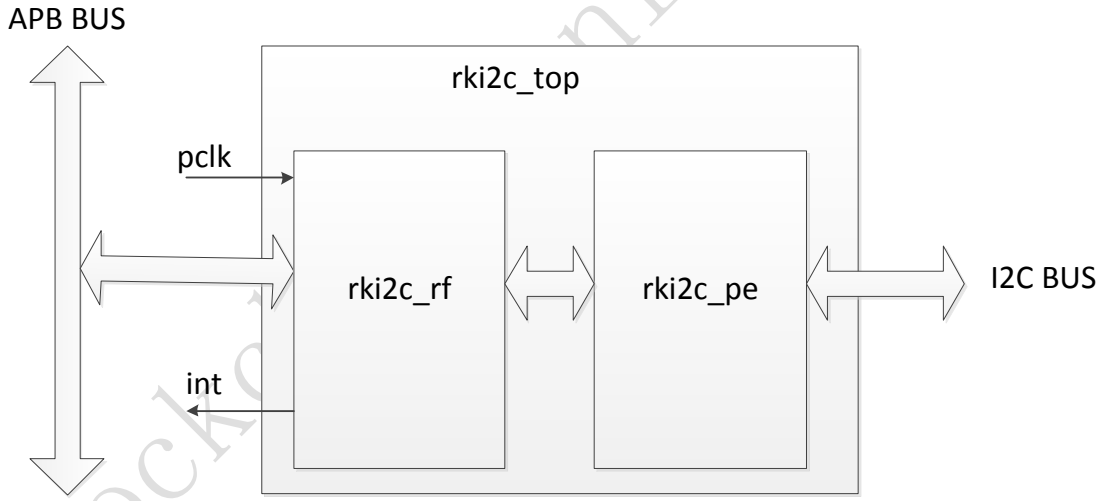


Fig. 37-1I2C architechture

I2C_rf

I2C_rf module is used to control the I2C controller operation by the host with APB interface. It implements the register set and the interrupt functionality. The CSR component operates synchronously with the pclk clock.

I2C_pe

I2C_pe module implements the I2C master operation for transmit data to and receive data from other I2C devices. The I2C master controller operates synchronously with the pclk.

I2C_top

I2C_top module is the top module of the I2C controller.

37.3 Function description

The I2C controller supports only Masterfunction. Itsupports the 7-bits/10-bits addressing mode and support general call address. The maximum clock frequency and transfer rate can be up to 400Kbit/sec.

The operations of I2C controller is divided to 2 parts and described separately: initialization and master mode programming.

Initialization

The I2C controller is based on AMBA APB bus architecture and usually is part of a SOC. So before I2C operates, some system setting & configuration must be conformed, which includes:

- I2C interrupt connection type: CPU interrupt scheme should be considered. If the I2C interrupt is connected to extra Interrupt Controller module, we need decide the INTC vector.
- I2C Clock Rate: The I2C controller uses the APB clock as the system clock so the APB clock will determine the I2C bus clock. The correct register setting is subject to the system requirement.

Master Mode Programming

SCL Clock: When the I2C controller is programmed in Master mode, the SCL frequency is determine by I2C_CLKDIV register. The SCL frequency is calculated by the following formula

SCL Divisor = 8*(CLKDIVL + CLKDIVH)

SCL = PCLK/ SCLK Divisor

Data Receiver Register Access

When the i2c controller received MRXCNT bytes data， CPU can get the datas through register RXDATA0 ~ RXDATA7. The controller can receive up to 32 byte datas in one transaction.

When MRXCNT register is written, the I2C controller will start to drive SCL to receive datas.

Transmit Trasmitter Register

Datas to transmit are written to TXDATA0~7 by CPU. The controller can transmit up to 32 byte datas in one transaction. The lower byte will be transmit first.

When MTXCNT register is written, the I2C controller will start to transmit datas.

Start Command

Write 1 to I2C_CON[3], the controller will send I2C start command.

Stop Command

Write 1 to I2C_CON[4], the controller will send I2C stop command

I2C Operation mode

There are four i2c operation mode.

When I2C_CON[2:1] is 2'b00, the controller transmit all valid datas in TXDATA0~TXDATA7 byte by byte. The controller will transmit lower byte first.

When I2C_CON[2:1] is 2'b01,the controller will transmit device address in MRXADDR first (Write/Read bit = 0) and then transmit device register address in

MRXRADDR. After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last , the controller enter receive mode .

When I2C_CON[2:1] is 2'b10, the controller is in receive mode, it will triggered clock to read MRXCNT byte datas.

When I2C_CON[2:1] is 2'b11,the controller will transmit device address in MRXADDR first (Write/Read bit = 1) and then transmit device register address in MRXRADDR . After that, the controller will assert restart signal and resend MRXADDR (Write/Read bit = 1). At last , the controller enter receive mode .

Read/Write Command

When I2C_OPMODE(I2C_CON[2:1]) is 2'b01 or 2'b11, the Read/Write command bit is decieded by controller itself.

In RX only mode (I2C_CON[2:1] is 2'b10), the Read/Write command bit is decided by MRXADDR[0].

In TX only mode (I2C_CON[[2:1] is 2'b00), the Read/Write command bit is decided by TXDATA[0].

Master Interrupt Condition

There are 7 interrupt bits in I2C_ISR register related to master mode.

Byte transfer finish interrupt (Bit 0): The bit is asserted when Master finish transferring a byte .

Byte received finish interrupt (Bit 1): The bit is asserted when Master finish receiving a byte.

MTXCNT bytes data transfer finish interrupt (Bit 2): The bit is asserted when Master finish transferring MTXCNT bytes.

MRXCNT bytes data received finish interrupt (Bit 3): The bit is asserted when Master finish receiving MRXCNT bytes.

Start interrupt (Bit 4) : The bit is asserted when Master finish asserting start command to I2C bus.

Stop interrupt (Bit 5) : The bit is asserted when Master finish asserting stop command to I2C bus.

Nak received interrupt (Bit 6): The bit is asserted when Master receive a NAK handshake.

Last byte acknowledge control

If I2C_CON[5] is 1, the I2C controller will transmit NAK handshake to slave when the last byte received in RX only mode.

If I2C_CON[5] is 0, the I2C controller will transmit ACK handshake to slave when the last byte received in RX only mode.

How to handle nak handshake received

If I2C_CON[6] is 1, the I2C controller will stop all transactions when nak handshake received. And the software should take responsibity to handle the problem.

If I2C_CON[6] is 0, the I2C controller will ignore all nak handshake received.

I2C controller data transfer waveform

- **Bit transferring**

(a) Data Validity

The SDA line must be stable during the high period of SCL, and the data on SDA line can only be changed when SCL is in low state.

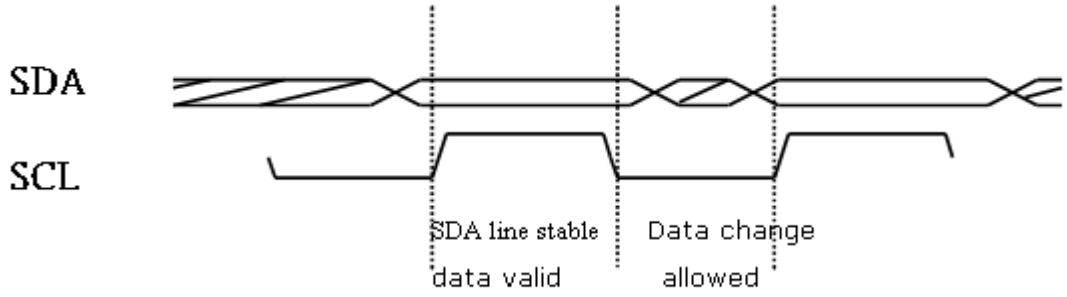


Fig. 37-2I2C DATA Validity

(b) START and STOP conditions

START condition occurs when SDA goes low while SCL is in high period. STOP condition is generated when SDA line goes high while SCL is in high state.

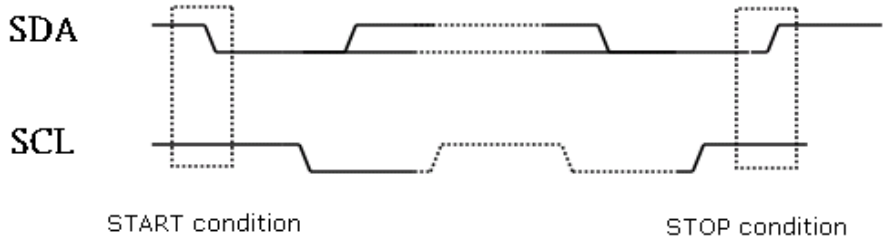


Fig. 37-3 I2C Start and stop conditions

- **Data transfer**

(a) Acknowledge

After a byte of data transferring (clocks labeled as 1~8), in 9th clock the receiver must assert an ACK signal on SDA line, if the receiver pulls SDA line to low, it means "ACK", on the contrary, it's "NOT ACK".

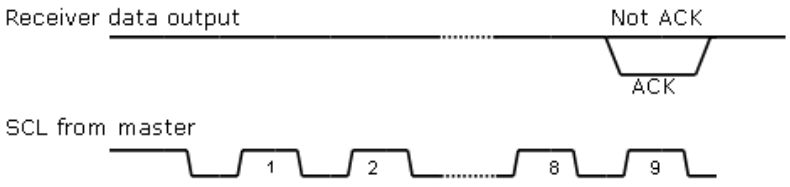


Fig. 37-4I2C Acknowledge

(b) Byte transfer

The master own I2C bus might initiate multi byte ot transfers to a slave, the transfers starts from a "START" command and ends in a "STOP"command. After every byte transfer, the receiver must reply an ACK to transmitter.

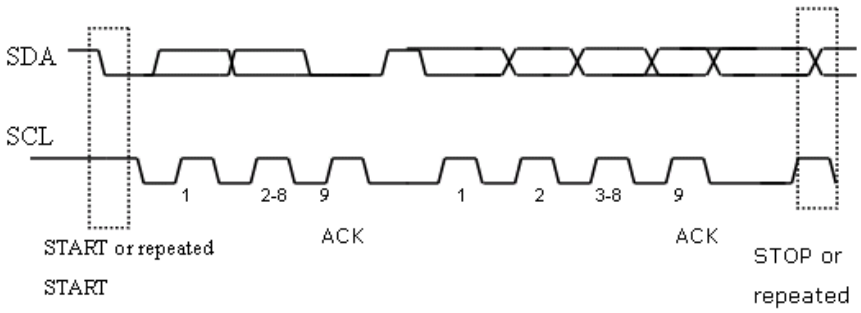


Fig. 37-5 I2C byte trasnfer

37.4 Register description

37.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
I2C_CON	0x0000	W	0x00000000	control register
I2C_CLKDIV	0x0004	W	0x00000001	Clock dividor register
I2C_MRXADDR	0x0008	W	0x00000000	the slave address accessed for master rx mode
I2C_MRXRADDR	0x000c	W	0x00000000	the slave register address accessed for master rx mode
I2C_MTXCNT	0x0010	W	0x00000000	master transmit count
I2C_MRXCNT	0x0014	W	0x00000000	master rx count
I2C_IEN	0x0018	W	0x00000000	interrupt enable register
I2C_IPD	0x001c	W	0x00000000	interrupt pending register
I2C_FCNT	0x0020	W	0x00000000	finished count
I2C_TXDATA0	0x0100	W	0x00000000	I2C tx data register 0
I2C_TXDATA1	0x0104	W	0x00000000	I2C tx data register 1
I2C_TXDATA2	0x0108	W	0x00000000	I2C tx data register 2
I2C_TXDATA3	0x010c	W	0x00000000	I2C tx data register 3
I2C_TXDATA4	0x0110	W	0x00000000	I2C tx data register 4
I2C_TXDATA5	0x0114	W	0x00000000	I2C tx data register 5
I2C_TXDATA6	0x0118	W	0x00000000	I2C tx data register 6
I2C_TXDATA7	0x011c	W	0x00000000	I2C tx data register 7
I2C_RXDATA0	0x0200	W	0x00000000	I2C rx data register 0
I2C_RXDATA1	0x0204	W	0x00000000	I2C rx data register 1
I2C_RXDATA2	0x0208	W	0x00000000	I2C rx data register 2
I2C_RXDATA3	0x020c	W	0x00000000	I2C rx data register 3
I2C_RXDATA4	0x0210	W	0x00000000	I2C rx data register 4
I2C_RXDATA5	0x0214	W	0x00000000	I2C rx data register 5
I2C_RXDATA6	0x0218	W	0x00000000	I2C rx data register 6
I2C_RXDATA7	0x021c	W	0x00000000	I2C rx data register 7

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

37.4.2 Detail Register Description

I2C_CON

Address: Operational Base + offset (0x0000)

control register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	act2nak operation when nak handshake is received 0: ignored 1: stop transaction
5	RW	0x0	ack last byte acknowledge control last byte acknowledge control in master rx mode . 0: ack 1: nak
4	W1C	0x0	stop stop enable when this bit is written to 1，I2C will generate stop signal. It cleared itself when stop operation ends.
3	W1C	0x0	start start enable when this bit is written to 1，I2C will generate start signal. It cleared itself when start operation ends.
2:1	RW	0x0	i2c_mode 00: tx only 01: tx address （device + register address） --> restart --> tx address –>rx only 10:rx only 11: tx address （device + register address, write/read bit is 1）--> restart --> tx address （device address） --> rx data
0	RW	0x0	i2c_en i2c module enable

I2C_CLKDIV

Address: Operational Base + offset (0x0004)

Clock dividor register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	CLKDIVH scl high level clock count T（SCL_HIGH）= T(PCLK) * CLKDIVH * 8
15:0	RW	0x0001	CLKDIVL scl low level clock count T（SCL_LOW）= T(PCLK) * CLKDIVL * 8

I2C_MRXADDR
Address: Operational Base + offset (0x0008)
the slave address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	addhvld address high byte valid
25	RW	0x0	addmvld address middle byte valid
24	RW	0x0	addlvld address low byte valid
23:0	RW	0x000000	saddr master address register the lowest bit indicate write or read

I2C_MRXRADDR
Address: Operational Base + offset (0x000c)
the slave register address accessed for master rx mode

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26	RW	0x0	sraddhvld address high byte valid
25	RW	0x0	sraddmvld address middle byte valid
24	RW	0x0	sraddlvld address low byte valid
23:0	RW	0x000000	sraddr slave register address accessed

I2C_MTXCNT
Address: Operational Base + offset (0x0010)
master transmit count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	mtxcnt master transmit count

I2C_MRXCNT
Address: Operational Base + offset (0x0014)
master rx count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	mrxcnt master rx count

I2C_IEN
Address: Operational Base + offset (0x0018)
interrupt enable register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	nakrcvien nak handshake received interrupt enable
5	RW	0x0	stopien stop operation finished interrupt enable
4	RW	0x0	startien start operation finished interrupt enable
3	RW	0x0	mbrfien MRXCNT data received finished interrupt enable
2	RW	0x0	mbtfien MTXCNT data transfer finished interrupt enable
1	RW	0x0	brfien byte rx finished interrupt enable
0	RW	0x0	btfien byte tx finished interrupt enable

I2C_IPD
Address: Operational Base + offset (0x001c)
interrupt pending register

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	nakrcvipd nak handshake received interrupt pending bit
5	RW	0x0	stopipd stop operation finished interrupt pending bit

Bit	Attr	Reset Value	Description
4	RW	0x0	startipd start operation finished interrupt pending bit
3	RW	0x0	mbrfipd MRXCNT data received finished interrupt pending bit
2	RW	0x0	mbtfipd MTXCNT data transfer finished interrupt pending bit
1	RW	0x0	brfipd byte rx finished interrupt pending bit
0	RW	0x0	btfipd byte tx finished interrupt pending bit

I2C_FCNT
Address: Operational Base + offset (0x0020)
finished count

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x00	fcnt finished count the count of data which has been transmitted or received for debug purpose

I2C_TXDATA0
Address: Operational Base + offset (0x0100)
I2C tx data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata0

I2C_TXDATA1
Address: Operational Base + offset (0x0104)
I2C tx data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata1

I2C_TXDATA2
Address: Operational Base + offset (0x0108)
I2C tx data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata2

I2C_TXDATA3
Address: Operational Base + offset (0x010c)
I2C tx data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata3

I2C_TXDATA4
Address: Operational Base + offset (0x0110)
I2C tx data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata4

I2C_TXDATA5
Address: Operational Base + offset (0x0114)
I2C tx data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata5

I2C_TXDATA6
Address: Operational Base + offset (0x0118)
I2C tx data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata6

I2C_TXDATA7
Address: Operational Base + offset (0x011c)
I2C tx data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	txdata7

I2C_RXDATA0
Address: Operational Base + offset (0x0200)
I2C rx data register 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata0

I2C_RXDATA1
Address: Operational Base + offset (0x0204)
I2C rx data register 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata1

I2C_RXDATA2
Address: Operational Base + offset (0x0208)
I2C rx data register 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata2

I2C_RXDATA3
Address: Operational Base + offset (0x020c)
I2C rx data register 3

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata3

I2C_RXDATA4
Address: Operational Base + offset (0x0210)
I2C rx data register 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata4

I2C_RXDATA5
Address: Operational Base + offset (0x0214)
I2C rx data register 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata5

I2C_RXDATA6
Address: Operational Base + offset (0x0218)
I2C rx data register 6

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata6

I2C_RXDATA7
Address: Operational Base + offset (0x021c)
I2C rx data register 7

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	rxdata7

37.5 Timing Diagram

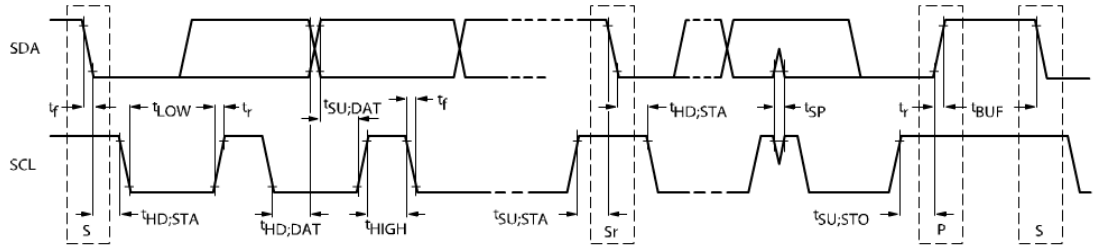


Fig. 37-6 I2C timing diagram

Table 37-1 I2C timing parameters

Parameter	Description	min	typ	max	unit
100Khz mode					
f_{SCL}	SCL clock frequency	-	100	-	KHz
$T_{HD:DAT}$	SDA hold time to rising edge of SCL	-	2.5	-	us
$T_{SU:DAT}$	SDA setup time to rising edge of SCL	-	2.5	-	us
t_{LOW}	Low period of SCL	-	5	-	us
t_{HIGH}	High period of SCL	-	5	-	us
400Khz mode					
f_{SCL}	SCL clock frequency	-	400	-	KHz
$T_{HD:DAT}$	SDA hold time to rising edge of SCL	-	0.625	-	us
$T_{SU:DAT}$	SDA setup time to rising edge of SCL	-	0.625	-	us
t_{LOW}	Low period of SCL	-	1.25	-	us
t_{HIGH}	High period of SCL	-	1.25	-	us

37.6 Interface description

Table 37-2I2C Interface Description

Module pin	Direction	Pad name	IOMUX
I2C0 Interface			
i2c0_sda	I/O	GPIO2_D[4]	GRF_GPIO2D_IOMUX[8]=1& GRF_SOC_CON1[11]=1
i2c0_scl	I/O	GPIO2_D[5]	GRF_GPIO2D_IOMUX[10]=1& GRF_SOC_CON1[11]=1
I2C1 Interface			
i2c1_sda	I/O	GPIO2_D[6]	GRF_GPIO2D_IOMUX[12]=1& GRF_SOC_CON1[12]=1
i2c1_scl	I/O	GPIO2_D[7]	GRF_GPIO2D_IOMUX[14]=1& GRF_SOC_CON1[12]=1
I2C2 Interface			
i2c2_sda	I/O	GPIO3_A[0]	GRF_GPIO3A_IOMUX[0]=1& GRF_SOC_CON1[13]=1
i2c2_scl	I/O	GPIO3_A[1]	GRF_GPIO3A_IOMUX[2]=1& GRF_SOC_CON1[13]=1
I2C3 Interface			

i2c3_sda	I/O	GPIO3_A[2]	GRF_GPIO3A_IOMUX[4]=1& GRF_SOC_CON1[14]=1
i2c3_scl	I/O	GPIO3_A[3]	GRF_GPIO3A_IOMUX[6]=1& GRF_SOC_CON1[14]=1
I2C3 Interface			
i2c4_sda	I/O	GPIO3_A[4]	GRF_GPIO3A_IOMUX[8]=1& GRF_SOC_CON1[15]=1
i2c4_scl	I/O	GPIO3_A[5]	GRF_GPIO3A_IOMUX[10]=1& GRF_SOC_CON1[15]=1

37.7 Application Notes

The I2C controller core operation flow chart below is to describe how the software configures and performs an I2C transaction through this I2C controller core. Descriptions are divided into 3 sections, transmit only mode, receive only mode, and mix mode . Users are strongly advised to following.

- Transmit only mode (I2C_CON[1:0]=2'b00)

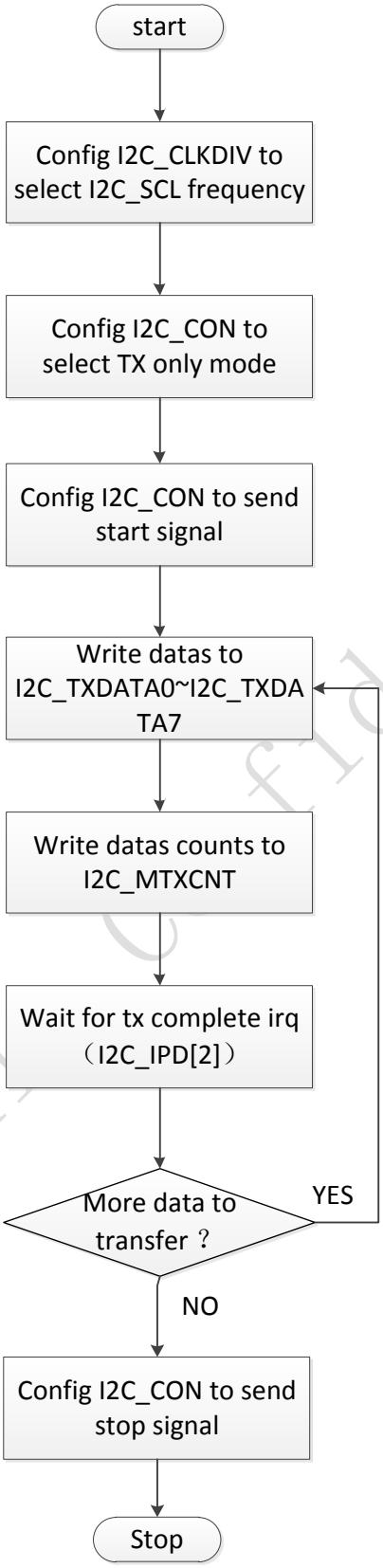


Fig. 37-7I2C Flow chat for tx only mode

- Receive only mode (I2C_CON[1:0]=2'b10)

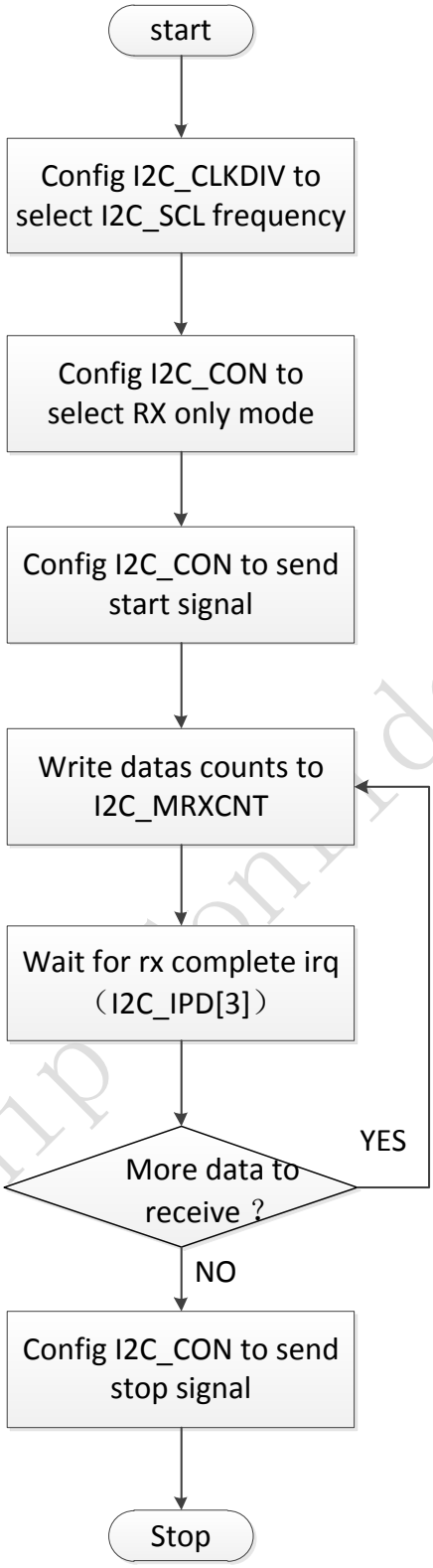


Fig. 37-8I2C Flow chat for rx only mode

- Mix mode (I2C_CON[1:0]=2'b01 or I2C_CON[1:0]=2'b10)

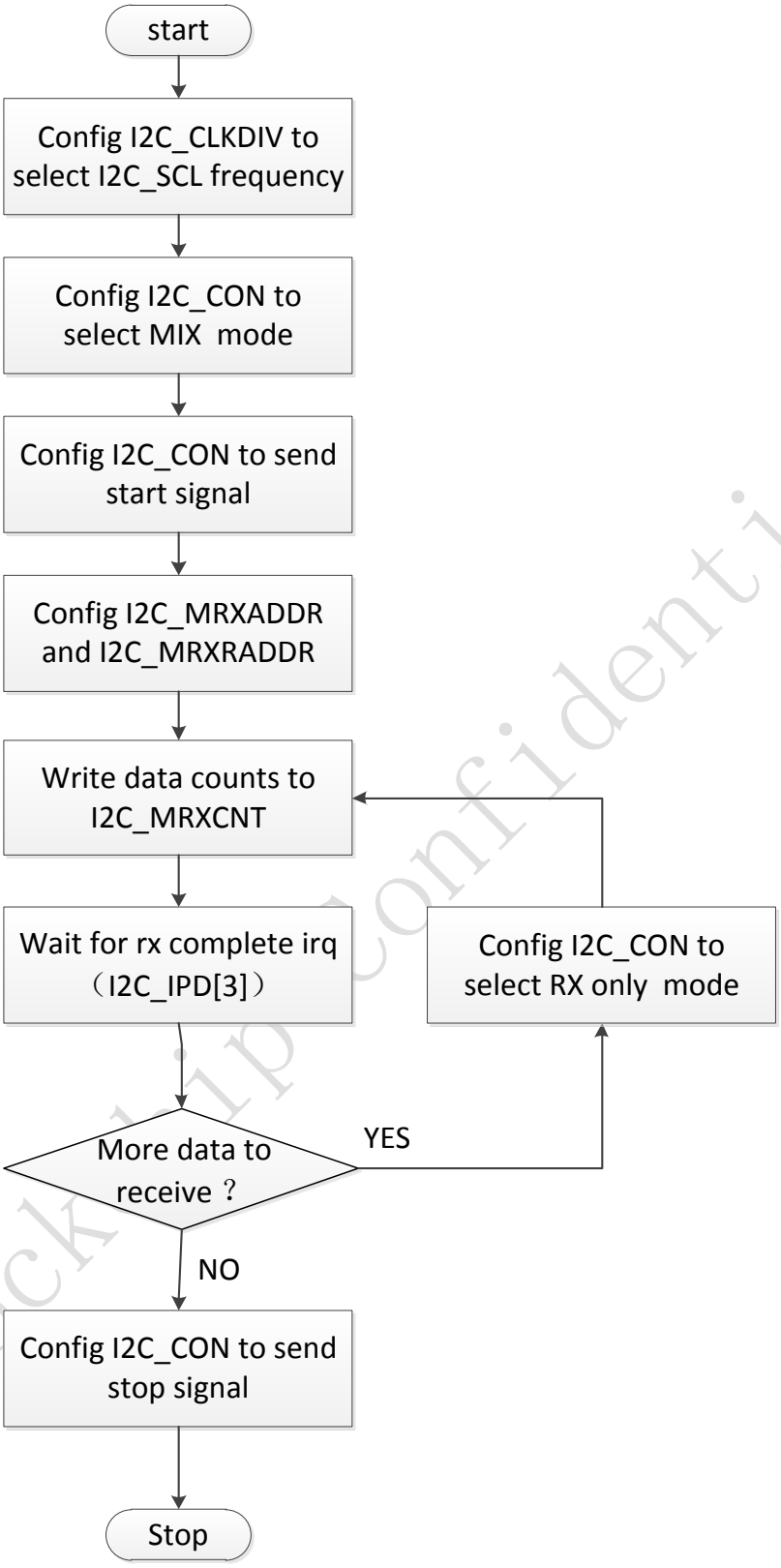


Fig. 37-9I2C Flow chat for mix mode

Chapter 38 GPIO

38.1 Overview

GPIO is a programmable General Purpose Programming I/O peripheral. This component is a APB slave device.GPIO controls the output data and direction of external I/O pads. It also can read back thedata on external pads using memory-mapped registers.

The features of GPIO are as follow:

- 32 bits APB bus width
- 32 independently configurable signals
- Separate data registers and data direction registers for each signal
- Software control for each signal, or for each bit of each signal
- Configurable interrupt mode for Port A
- Port A has 32 bits

Notes: Port A 32bits are corresponding to port A/B/C/D 8bits in Chapter1

38.2 Block Diagram

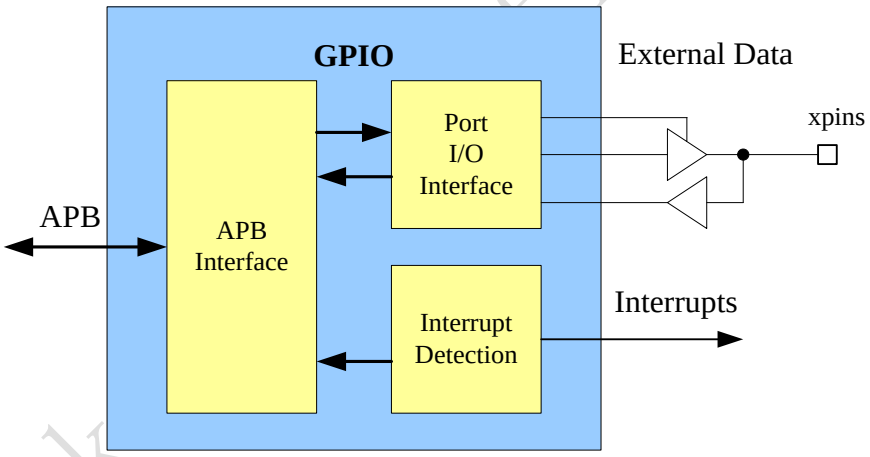


Fig. 38-1GPIO block diagram

Block descriptions:

APB Interface

The APB Interface implements the APB slave operation. It's bus width is 32 bits.

Port I/O Interface

External data Interface to or from I/O pads.

Interrupt Detection

Interrupt interface to or from interrupt controller.

38.3 Function description

38.3.1 Operation

Control Mode(software)

Under software control, the data and direction control for the signal aresourced from the data register (GPIO_SWPORTA_DR) and direction control register (GPIO_SWPORTA_DDR).

The direction of the external I/O pad is controlled by a write to the Porta datadirection register (GPIO_SWPORTA_DDR). The data written to this memory-mapped register gets mappedonto an output signal, GPIO_PORTA_DDR, of the GPIO peripheral. This output signal controls thedirection of an external I/O pad.

The data written to the Porta data register (GPIO_SWPORTA_DR) drives the output buffer of the I/O pad.External data are input on the external data signal, GPIO_EXT_PORTA. Reading the external signal register(GPIO_EXT_PORTA) shows the value on the signal, regardless of the direction. This register is read-only,meaning that it cannot be written from the APB software interface.

Reading External Signals

The data on the GPIO_EXT_PORTA external signal can always be read. The data on the external GPIO signal is read by an APB read of the memory-mapped register, GPIO_EXT_PORTA.

An APB read to the GPIO_EXT_PORTA register yields a value equal to that which is on the GPIO_EXT_PORTA signal.

Interrupts

Port A can be programmed to accept external signals as interrupt sources on any of the bits of the signal. The type of interrupt is programmable with one of the following settings:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

The interrupts can be masked by programming the GPIO_INTMASK register. The interrupt status can be read before masking (called raw status) and after masking.

The interrupts are combined into a single interrupt output signal, which has the same polarity as the individual interrupts. In order to mask the combined interrupt, all individual interrupts have to be masked. The single combined interrupt does not have its own mask bit.

Whenever Port A is configured for interrupts, the data direction must be set to Input. If the data direction register is reprogrammed to Output, then any pending interrupts are not lost.However, no new interrupts are generated.

For edge-detected interrupts, the ISR can clear the interrupt by writing a 1 to the GPIO_PORTA_EOI register for the corresponding bit to disable the interrupt. This write also clears the interrupt status and raw status registers. Writing to the GPIO_PORTA_EOI register has no effect on level-sensitive interrupts. If level-sensitive interrupts cause the processor to interrupt, then the ISR can poll the GPIO_INT_RAWSTATUS register until the interrupt source disappears, or it can write to the GPIO_INTMASK register to mask the interrupt before exiting the ISR. If the ISR exits without masking or disabling the interrupt prior to exiting, then the level-sensitive interrupt repeatedly requests an interrupt until the interrupt is cleared at the source.

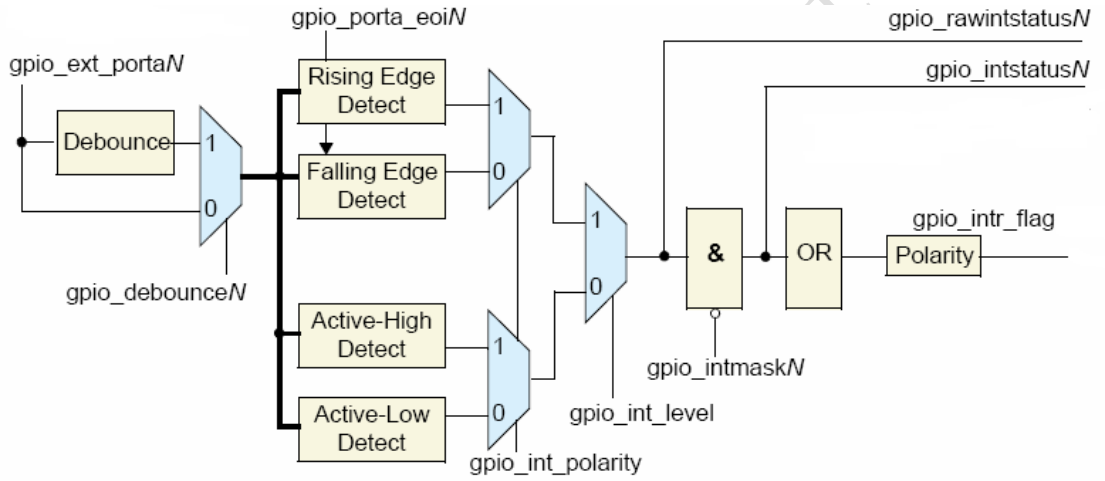


Fig. 38-2 Interrupt RTL Block Diagram

Debounce operation

Port A has been configured to include the debounce capability interrupt feature. The external signal can be debounced to remove any spurious glitches that are less than one period of the external debouncing clock.

When input interrupt signals are debounced using a debounce clock(pclk), the signals must be active for a minimum of two cycles of the debounce clock to guarantee that they are registered. Any input pulse widths less than a debounce clock period are bounced. A pulse width between one and two debounce clock widths may or may not propagate, depending on its phase relationship to the debounce clock. If the input pulse spans two rising edges of the debounce clock, it is registered. If it spans only one risingedge, it is not registered.

Synchronization of Interrupt Signals to the System Clock

Interrupt signals are internally synchronized to pclk. Synchronization topclk must occur for edge-detect signals. With level-sensitive interrupts, synchronization is optional andunder software control(GPIO_LS_SYNC).

38.3.2 Programming

Programming Considerations

● Reading from an unused location or unused bits in a particular register always returns zeros. Thereis no error mechanism in the APB.

● Programming the GPIO registers for interrupt capability, edge-sensitive or level-sensitiveinterrupts, and interrupt polarity should be completed prior to enabling the interrupts on Port A inorder to prevent spurious glitches on the interrupt lines to the interrupt controller.

● Writing to the interrupt clear register clears an edge-detected interrupt and has no effect on alevel-sensitive interrupt.

6 GPIOs' hierarchy in the chip

GPIO0, GPIO1, GPIO2 are in CPU subsystem, GPIO3, GPIO4 are in peripheral subsystem, and GPIO6 is in alive subsystem.

38.4 Register description

This chapter describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses. There are 6 GPIOs (GPIO0 ~ GPIO4, GPIO6), and each of them has same register group. Therefore, 6 GPIOs' register groups have 6 different base-address.

38.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
GPIO_SWPORTA_DR	0x0000	W	0x00000000	Port A data register
GPIO_SWPORTA_DDR	0x0004	W	0x00000000	Port A data direction register
GPIO_INTEN	0x0030	W	0x00000000	Interrupt enable register
GPIO_INTMASK	0x0034	W	0x00000000	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x0038	W	0x00000000	Interrupt level register
GPIO_INT_POLARITY	0x003C	W	0x00000000	Interrupt polarity register
GPIO_INT_STATUS	0x0040	W	0x00000000	Interrupt status of port A
GPIO_INT_RAWSTATUS	0x0044	W	0x00000000	Raw Interrupt status of port A
GPIO_DEBOUNCE	0x0048	W	0x00000000	Debounce enable register
GPIO_PORTA_EOI	0x004C	W	0x00000000	Port A clear interrupt register
GPIO_EXT_PORTA	0x0050	W	0x00000000	Port A external port register
GPIO_LS_SYNC	0x0060	W	0x00000000	Level_sensitive synchronization enable register

*Notes: Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

38.4.2 Detail Register Description

GPIO_SWPORTA_DR
Address: Operational Base + offset(0x00)
Port A data register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode. The value read back is equal to the last value written to this register.

GPIO_SWPORTA_DDR
Address: Operational Base + offset(0x04)
Port A data register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Values written to this register independently control the direction of the corresponding data bit in Port A. 0: Input (default) 1: Output

GPIO_INTEN
Address: Operational Base + offset(0x30)
Interrupt enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Allows each bit of Port A to be configured for interrupts. Whenever a 1 is written to a bit of this register, it configures thecorresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output. 0: Configure Port A bit as normal GPIO signal (default) 1: Configure Port A bit as interrupt

GPIO_INTMASK
Address: Operational Base + offset(0x34)
Interrupt mask register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. 0: Interrupt bits are unmasked (default) 1: Mask interrupt

GPIO_INTTYPE_LEVEL
Address: Operational Base + offset(0x38)
Interrupt level register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Controls the type of interrupt that can occur on Port A. 0: Level-sensitive (default) 1: Edge-sensitive

GPIO_INT_POLARITY
Address: Operational Base + offset(0x3C)
Interrupt polarity register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Controls the polarity of edge or level sensitivity that can occur on input of Port A. 0: Active-low (default) 1: Active-high

GPIO_INT_STATUS
Address: Operational Base + offset(0x40)
Interrupt status register

Bit	Attr	Reset Value	Description
31:0	R	0x00	Interrupt status of Port A

GPIO_INT_RAWSTATUS
Address: Operational Base + offset(0x44)
Raw Interrupt status register

Bit	Attr	Reset Value	Description
31:0	R	0x00	Raw interrupt of status of Port A (premasking bits)

GPIO_DEBOUNCE
Address: Operational Base + offset(0x48)
Debounce enable register

Bit	Attr	Reset Value	Description
31:0	RW	0x00	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0: No debounce (default) 1: Enable debounce

GPIO_PORTA_EOI
Address: Operational Base + offset(0x4C)
Port A clear interrupt register

Bit	Attr	Reset Value	Description
31:0	W	0x00	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this

			register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. 0: No interrupt clear (default) 1: Clear interrupt

GPIO_EXT_PORTA
Address: Operational Base + offset(0x50)
Port A external port register

Bit	Attr	Reset Value	Description
31:0	R	0x00	When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A.

GPIO_LS_SYNC
Address: Operational Base + offset(0x60)
Level_sensitive synchronization enable register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	RW	0x00	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0: No synchronization to pclk_intr (default) 1: Synchronize to pclk_intr

Chapter 39 Timer

39.1 Overview

Timer is a programmable timer peripheral. This component is an APBslave device. Timers count down from a programmed value and generate an interrupt when the count reaches zero.

The features of timer are as follow:

- Three programmable 32 bits timers
- Two operation modes: free-running and user-defined count
- Maskable for each individual interrupt

39.2 Block Diagram

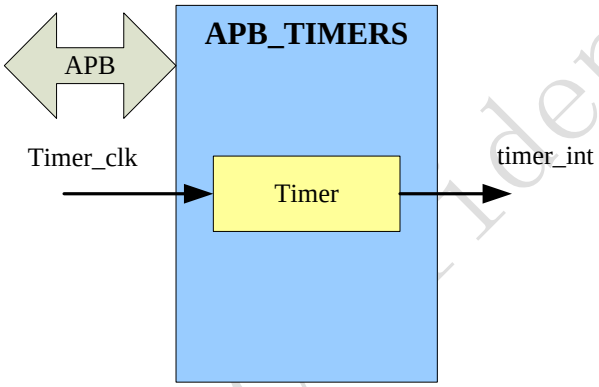


Fig. 39-1Timer Block Diagram

39.3 Function description

39.3.1 Timer clk selection

Timer0 and timer1 are in the CPU subsystem, and timer clock is 24MHz OSC; Timer2 is in the ALIVE subsystem, and timer clock is 24MHz OSC.

39.3.2 Programming sequence

1. Initialize the timer through the TIMER1_CONTROLREG register:

   a. Disable the timer by writing a "0" to the timer enable bit (bit 0); accordingly, the timer_enoutput signal is de-asserted.

   b. Program the timer mode—user-defined or free-running—by writing a "0" or "1," respectively,to the timer mode bit (bit 1).

   c. Set the interrupt mask as either masked or not masked by writing a "0" or "1," respectively, tothe timer interrupt mask bit (bit 2).

2. Load the timer counter value into the TIMER1_LOAD_COUNT register.

3. Enable the timer by writing a "1" to bit 0 of TIMER1_CONTROLREG.

Timers Usage flow

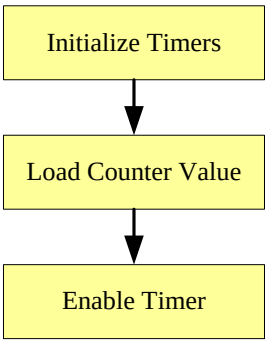


Fig. 39-2 Timer Usage Flow

39.3.3 Enabling and Disabling a Timer

You use bit 0 of the TIMER1_CONTROLREG, to either enable or disable a timer.

Enabling a Timer

If you want to enable a timer, you write a "1" to bit 0 of its TIMER1_CONTROLREG register.

Disabling a Timer

To disable a timer, write a "0" to bit 0 of its TIMER1_CONTROLREG register. When a timer is enabled and running, its counter decrements on each rising edge of its clock signal, timer_N_clk. When a timer transitions from disabled to enabled, the current value of its TIMER1_LOAD_COUNT register is loaded into the timer counter on the next rising edge of timer_N_clk.

When the timer enable bit is de-asserted and the timer stops running, the timer counter and any associated registers in the timer clock domain, such as the toggle register, are asynchronously reset.

When the timer enable bit is asserted, then a rising edge on the timer_en signal is used to load the initial value into the timer counter. A "0" is always read back when the timer is not enabled.

39.3.4 Loading a Timer Countdown Value

The initial value for each timer—that is, the value from which it counts down—is loaded into the timer using the appropriate load count register (TIMER1_LOAD_COUNT). Two events can cause a timer to load the initial count from its TIMER1_LOAD_COUNT register:

- Timer is enabled after being reset or disabled
- Timer counts down to 0

When a timer counts down to 0, it loads one of two values, depending on the timer operating mode:

- User-defined count mode – Timer loads the current value of the TIMER1_LOAD_COUNT register. Use this mode if you want a fixed, timed interrupt. Designate this mode by writing a "1" to bit 1 of TIMER1_CONTROLREG.
- Free-running mode – Timer loads the maximum value, which is dependent on the timer width; that is, the TIMER1_LOAD_COUNT register is comprised of 2TIMER_WIDTH_N – 1 bits, all of which are loaded with 1s. The timer counter wrapping to its maximum value allows time to reprogram or disable the timer before another interrupt occurs. Use this mode if you want a single timed interrupt. Designate this mode by writing a "0" to bit 1 of TIMER1_CONTROLREG.

39.4 Register description

This chapter describes the control/status registers of the design. Software should read and write these registers using 32-bits accesses.

There are 3 individual timers. (Timer0 ~ Timer2)

39.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
TIMER1_LOAD_COUNT	0x0000	W	0x00000000	Timer1 Load Count Register
TIMER1_CURRENT_VALUE	0x0004	W	0x00000000	Timer1 Current Value Register
TIMER1_CONTROLREG	0x0008	W	0x00000000	Timer1 Control Register
TIMER1_EOI	0x000C	W	0x00000000	Timer1 End-of-Interrupt Register
TIMER1_INTSTATUS	0x0010	W	0x00000000	Timer1 Interrupt Status Register
TIMERS_INTSTATUS	0x00a0	W	0x00000000	Timers Interrupt Status Register
TIMERS_EOI	0x00a4	W	0x00000000	Timers End-of-Interrupt Register
TIMERS_RAWSTATUS	0x00a8	W	0x00000000	Timers Raw Interrupt Status Register

*Notes: Size: **B** – Byte (8 bits) access, **HW** – Half WORD (16 bits) access, **W** –WORD (32 bits) access*

39.4.2 Detail Register Description

TIMER1_LOAD_COUNT
Address: Operational Base + offset(0x00)
Timer1 Load Count Register

Bit	Attr	Reset Value	Description
31:0	RW	0x1f	Value to be loaded into Timer1. This is the value from which counting commences.

TIMER1_CURRENT_VALUE
Address: Operational Base + offset(0x04)
Timer1 Current Value Register

Bit	Attr	Reset Value	Description
31:0	R	0x1f	Current Value of Timer1.

TIMER1_CONTROLREG
Address: Operational Base + offset(0x08)
Timer1 Control Register

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2	RW	0x0	Timer interrupt mask. 0: not mask 1: mask

1	RW	0x0	Timer mode. 0: free-running mode 1: user-defined count mode
0	RW	0x0	Timer enable. 0: disable 1: enable

TIMER1_EOI
Address: Operational Base + offset(0x0C)
Timer1 End-of-Interrupt Register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	R	0x0	Reading from this register returns all zeros(0) and clear interrupt from timer1

TIMER1_INTSTATUS
Address: Operational Base + offset(0x10)
Timer1 Interrupt Status Register

Bit	Attr	Reset Value	Description
31:1	-	-	Reserved
0	R	0x0	This register contains the interrupt status for timer1

TIMERS_INTSTATUS
Address: Operational Base + offset(0xa0)
Timers Interrupt Status Register

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2	R	0x0	This register contains the interrupt status for timer3
1	R	0x0	This register contains the interrupt status for timer2
0	R	0x0	This register contains the interrupt status for timer1

TIMERS_EOI
Address: Operational Base + offset(0xa4)
Timers End-of-Interrupt Register

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2:0	R	0x0	Reading from this register returns all zeros(0) and clear interrupt from all timers

TIMERS_RAWSTATUS
Address: Operational Base + offset(0xa8)
Timers Raw Interrupt Status Register

Bit	Attr	Reset Value	Description
31:3	-	-	Reserved
2	R	0x0	This register contains the interrupt status for timer3 prior to masking
1	R	0x0	This register contains the interrupt status for timer2 prior to masking
0	R	0x0	This register contains the interrupt status for timer1 prior to masking

*Notes: Attr: **RW** – Read/writable, **R** – read only, **W** – write only*

39.5 Application Notes

timer0 ~ timer2 usage flow

Timer0 and timer1 are in the CPU subsystem, timer2 is in the ALIVE subsystem, and the timer clock of timer0 ~ timer2 is 24MHz OSC. In this case, the timer_clk signal is asynchronous to pclk;

In the condition of the timer_clk signal is asynchronous to pclk. When you disable the timer enable bit (bit 0 of TIMER1_CONTROLREG), the timer_en output signal is de-asserted and, accordingly, timer_clk should stop. Then when you enable the timer, the timer_en signal is asserted and timer_clk should start running.

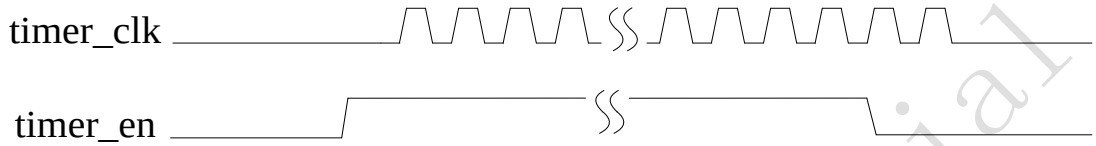


Fig. 39-3Timing of Timer_en and Timer_clk (timer_clk is async to pclk)

When the timer_clk signal is asynchronous to pclk, the timer usage flow is as follow:

1. Before using the timer, make sure that the timer_clk and timer_en are disabled. (Disable timer0 ~ timer2 clock by writing "1" into CRU_CLKGATE1_CON bit0~bit2 respectively, and disable timer_en by writing a "0" to the timer enable bit (TIMER1_CONTROLREGbit0))

2. Initialize the timer through the TIMER1_CONTROLREG register:

   a. Program the timer mode—user-defined or free-running—by writing a "0" or "1," respectively,to the timer mode bit (bit 1).
   b. Set the interrupt mask as either masked or not masked by writing a "0" or "1," respectively, tothe timer interrupt mask bit (bit 2).

3. After timer initialization, enable timer_en firstly. (enable the timer by writing a "1" to bit 0 of TIMER1_CONTROLREG)

4.After timer_en enabled, enable timer_clk. (enable timer0 ~timer2 clock by writing "0" into CRU_CLKGATE1_CONbit0~bit2 respectively.)

5. When you want to disable the timer, firstly, disable the timer_en.

6. After timer_en diabled, disable the timer_clk.

Timers Usage flow(timer_clk signal is asynchronous to pclk)

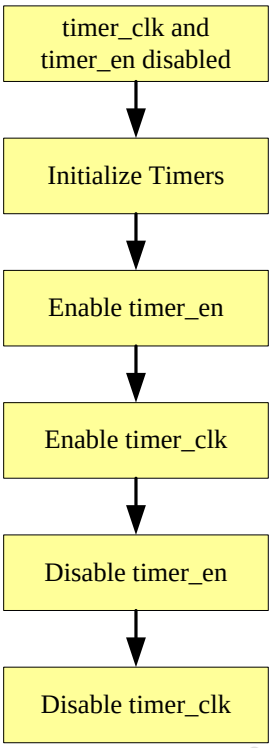


Fig. 39-4Timer0 and Timer1Usage Flow

Chapter 40 PWM

40.1 Overview

There are four PWM blocks in PWM Timer (PWM0, PWM1, PWM2 and PWM3). Each PWM block built-in 4-bit pre-scalar from PCLK. The PWM Timer supports both reference mode, which can output various duty-cycle waveforms, and capture, which can measure the duty-cycle of input waveform.

40.1.1 Features

- Built-in three 32 bit timer modulers
- Programmable counter
- Chained timer for long period purpose
- 4-channel 32-bit timer with Pulse Width Modulation(PWM)
- Support maskable interrupt

40.2 Block Diagram

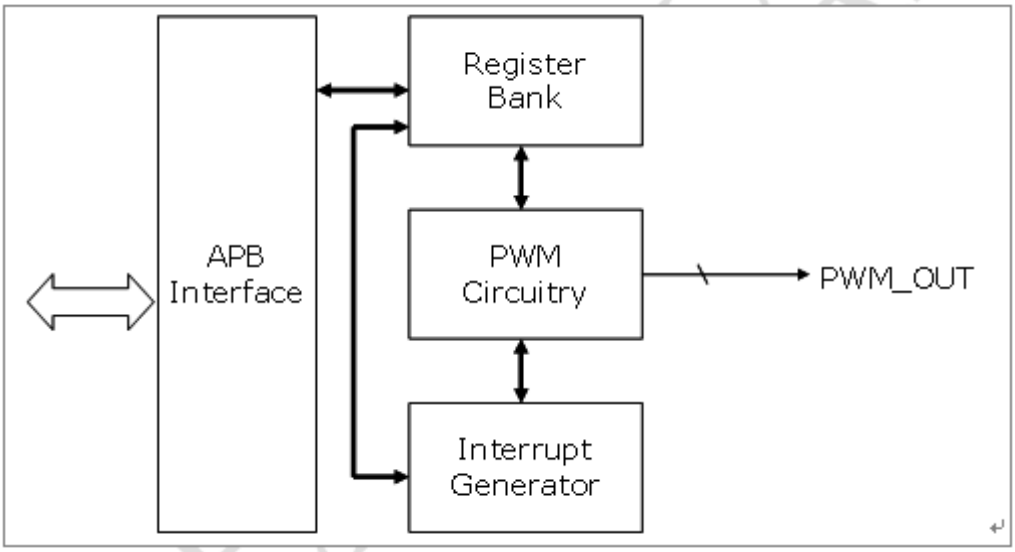


Fig. 40-1 PWM architecture

PWM Register Block

This block controls the setting of PWM mode.

PWM Circuitry

This block includes clock pre-scalar and reference comparator for PWM timer.

Interrupt Generator

This block handles the interrupt generation, masking, and clearing.

40.3 Register description

40.3.1 Register Summary

Name	Offset	Size	Reset Value	Description

Name	Offset	Size	Reset Value	Description
PWM0_CNTR	0x0000	W	0x00000000	Main counter register
PWM0_HRC	0x0004	W	0x00000000	PWM HIGH Reference/Capture register
PWM0_LRC	0x0008	W	0x00000000	PWM LOW Reference/Capture register
PWM0_CTRL	0x000c	W	0x00000000	Current value register
PWM1_CNTR	0x0010	W	0x00000000	Main counter register
PWM1_HRC	0x0014	W	0x00000000	PWM HIGH Reference/Capture register
PWM1_LRC	0x0018	W	0x00000000	PWM LOW Reference/Capture register
PWM1_CTRL	0x001c	W	0x00000000	Current value register
PWM2_CNTR	0x0020	W	0x00000000	Main counter register
PWM2_HRC	0x0024	W	0x00000000	PWM HIGH Reference/Capture register
PWM2_LRC	0x0028	W	0x00000000	PWM LOW Reference/Capture register
PWM2_CTRL	0x002c	W	0x00000000	Current value register
PWM3_CNTR	0x0030	W	0x00000000	Main counter register
PWM3_HRC	0x0034	W	0x00000000	PWM HIGH Reference/Capture register
PWM3_LRC	0x0038	W	0x00000000	PWM LOW Reference/Capture register
PWM3_CTRL	0x003c	W	0x00000000	Current value register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

40.3.2 Detail Register Description

PWMn_CNTR (n=0~3)
Address: Operational Base + offset (0x00, 0x10, 0x20, 0x30)
Main counter register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	main_pwm_timer_cnt Main PWM timer counter. Counting value ranges from 0 ~ (2^32 -1).

PWMn_HRC (n=0~3)
Address: Operational Base + offset (0x04, 0x14, 0x24, 0x34)
PWM HIGH Reference/Capture register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	high_reference_cap PWM HIGH reference/capture registers

PWMn_LRC (n=0~3)
Address: Operational Base + offset (0x08, 0x18, 0x28, 0x38)
PWM LOW Reference/Capture register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	loe_reference_cap PWM LOW reference/capture registers

PWMn_CTRL (n=0~3)
Address: Operational Base + offset (0x0C, 0x1C, 0x2C, 0x3C)
Current value register

Bit	Attr	Reset Value	Description
31:13	RO	0x0	reserved
12:9	RW	0x0	prescale_factor Prescale factor. 0000: 1/2 0001: 1/4 0010: 1/8 0011: 1/16 0100: 1/32 0101: 1/64 0110: 1/128 0111: 1/256 1000: 1/512 1001: 1/1024 1010: 1/2048 1011: 1/4096 1100: 1/8192 1101: 1/16384 1110: 1/32768 1111: 1/65536
8	RW	0x0	cap_mode_en Capture mode enable/disable 0: Disable 1: Enable
7	W1C	0x0	pwm_reset PWM reset. 0: Normal operation 1: Reset PWM
6	W1C	0x0	int_status_clr Interrupt status and clear bit. Write "1" to clear interrupt status.
5	RW	0x0	pwm_timer_int_en PWM timer interrupt enable/disable. PWM timer will assert an interrupt when PWMTx_CNTR value is equal to the value of PWMTx_LRC or PWMTx_HRC. 0: Disable 1: Enable
4	RW	0x0	single_cnt_mode Single counter mode. 0: PWMTx_CNTR is restarted after it reaches value equal to the PWMTx_LRC value. 1: PWMTx_CNTR is not increased anymore after it reaches value equal to the PWMTx_LRC value.
3	RW	0x0	pwm_output_en PWM output enable/disable. 0: Disable 1: Enable

Bit	Attr	Reset Value	Description
2:1	RO	0x0	reserved
0	RW	0x0	pwm_timer_en PWM timer enable/disable. 0: Disable 1: Enable

40.4 Interface description

Table 40-1 PWM Interface Description

Module pin	Direction	Pad name	IOMUX
pwm3	O	GPIO0_D[7]	GRF_GPIO0D_IOMUX[14]=1
pwm2	O	GPIO0_D[6]	GRF_GPIO0D_IOMUX[12]=1
pwm1	O	GPIO0_A[4]	GRF_GPIO0A_IOMUX[8]=1
pwm0	O	GPIO0_A[3]	GRF_GPIO0A_IOMUX[6]=1

Chapter 41 WatchDog

41.1 Overview

Watchdog Timer (WDT) is an APB slave peripheral that can be used to prevent system lockup that may becaused by conflicting parts or programs in a SoC.The WDT would generated interrupt or reset signal when it's counter reaches zero, then a reset controller would reset the system.

The features of WDT are as follow:

- 32 bits APB bus width
- WDT counter's clock is pclk
- 32 bits WDT counter width
- Counter counts down from a preset value to 0 to indicate the occurrence of a timeout
- WDT can perform two types of operations when timeout occurs:
  (1) Generate a system reset;
  (2) First generate an interrupt and if this is not cleared by the service routine by the time a second timeout occurs then generate a system reset
- Programmable reset pulse length
- Total 16 defined-ranges of main timeout period

41.2 Block Diagram

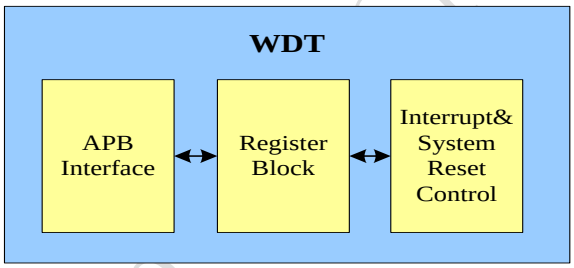


Fig. 41-1WDT block diagram

Block Descriptions:

APB Interface

The APB Interface implements the APB slave operation. It's bus width is 32 bits.

Register Block

A register block with read coherency for the current count register.

Interrupt & system reset control

An interrupt/system reset generation block comprising of a decrementing counter and control logic.

41.3 Function Description

41.3.1 Operation

Counter

The WDT counts from a preset (timeout) value in descending order to zero.

When the counter reaches zero, depending on the output response mode selected, either a system reset or an interrupt occurs. When the counter reaches zero, it wraps to the selected timeout value and continues decrementing. The user can restart the counter to its initial value. This is programmed by writing to the restart register at any time. The process of restarting the watchdog counter is sometimes referred to as kicking the dog. As a safety feature to prevent accidental restarts, the value 0x76 must be written to the Current Counter Value Register (WDT_CRR).

Interrupts

The WDT can be programmed to generate an interrupt (and then a system reset) when a timeout occurs. When a 1 is written to the response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates an interrupt. If it is not cleared by the time a second timeout occurs, then it generates a system reset. If a restart occurs at the same time the watchdog counter reaches zero, an interrupt is not generated.

System Resets

When a 0 is written to the output response mode field (RMOD, bit 1) of the Watchdog Timer Control Register (WDT_CR), the WDT generates a system reset when a timeout occurs.

Reset Pulse Length

The reset pulse length is the number of pclk cycles for which a system reset is asserted. When a system reset is generated, it remains asserted for the number of cycles specified by the reset pulse length or until the system is reset. A counter restart has no effect on the system reset once it has been asserted.

41.3.2 Programming sequence

Operation Flow Chart (Response mode=1)

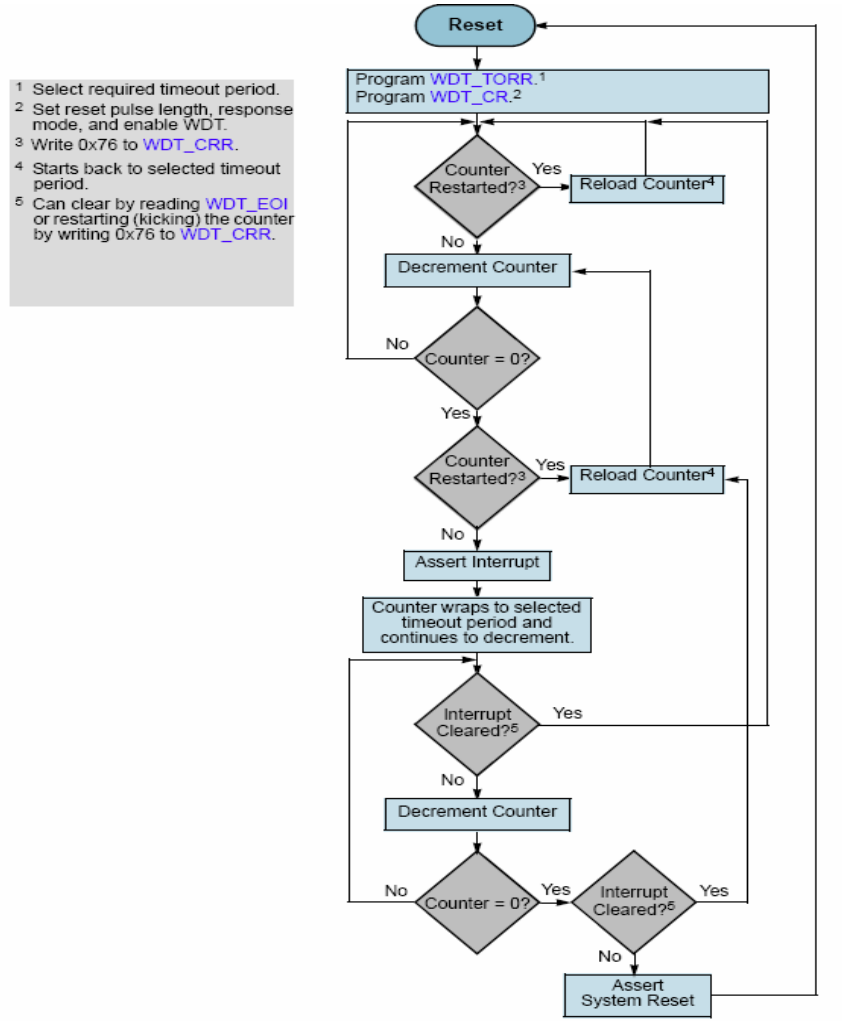


Fig. 41-2WDT Operation Flow

41.4 Register description

41.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
WDT_CR	0x0000	W	0x0000000a	Control Register
WDT_TORR	0x0004	W	0x00000000	Timeout range Register
WDT_CCVR	0x0008	W	0x00000000	Current counter value Register
WDT_CRR	0x000c	W	0x00000000	Counter restart Register
WDT_STAT	0x0010	W	0x00000000	Interrupt status Register
WDT_EOI	0x0014	W	0x00000000	Interrupt clear Register

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

41.4.2 Detail Register Description

WDT_CR

Address: Operational Base + offset (0x0000)

Control Register

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:2	RW	0x2	rst_pluse_lenth Reset pulse length. This is used to select the number of pclk cycles for which the system reset stays asserted. 000: 2 pclk cycles 001: 4 pclk cycles 010: 8 pclk cycles 011: 16 pclk cycles 100: 32 pclk cycles 101: 64 pclk cycles 110: 128 pclk cycles 111: 256 pclk cycles
1	RW	0x1	resp_mode Response mode. Selects the output response generated to a timeout. 0: Generate a system reset. 1: First generate an interrupt and if it is not cleared by the time a second timeout occurs then generate a system reset.
0	RW	0x0	wdt_en WDT enable 0: WDT disabled; 1: WDT enabled.

WDT_TORR

Address: Operational Base + offset (0x0004)

Timeout range Register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	timeout_period Timeout period. This field is used to select the timeout period from which the watchdog counter restarts. A change of the timeout period takes effect only after the next counter restart (kick). The range of values available for a 32-bit watchdog counter are: 0000: 0x0000ffff 0001: 0x0001ffff 0010: 0x0003ffff 0011: 0x0007ffff 0100: 0x000fffff 0101: 0x001fffff 0110: 0x003fffff 0111: 0x007fffff 1000: 0x00ffffff 1001: 0x01ffffff 1010: 0x03ffffff 1011: 0x07ffffff 1100: 0x0fffffff 1101: 0x1fffffff 1110: 0x3fffffff 1111: 0x7fffffff

WDT_CCVR
Address: Operational Base + offset (0x0008)
Current counter value Register

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	cur_cnt Current counter value This register, when read, is the current value of the internal counter. This value is read coherently when ever it is read

WDT_CRR
Address: Operational Base + offset (0x000c)
Counter restart Register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved

Bit	Attr	Reset Value	Description
7:0	W1C	0x00	cnt_restart Counter restart This register is used to restart the WDT counter. As a safety feature to prevent accidental restarts, the value 0x76 must be written. A restart also clears the WDT interrupt. Reading this register returns zero.

WDT_STAT
Address: Operational Base + offset (0x0010)
Interrupt status Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_status This register shows the interrupt status of the WDT. 1: Interrupt is active regardless of polarity; 0: Interrupt is inactive.

WDT_EOI
Address: Operational Base + offset (0x0014)
Interrupt clear Register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	wdt_int_clr Clears the watchdog interrupt. This can be used to clear the inter Clears the watchdog interrupt. This can be used to clear the interrupt without restarting the watchdog counter.

Chapter 42 SAR-ADC

42.1 Overview

The ADC is an 4-channel signal-ended 10-bit Successive Approximation Register (SAR) A/D Converter. It uses the supply and ground as it reference which avoid use of any external reference . It converts the analog input signal into 10-bit binary digital codes at maximum conversion rate of 100KSPS with 1MHz A/D converter clock .

42.2 Block Diagram

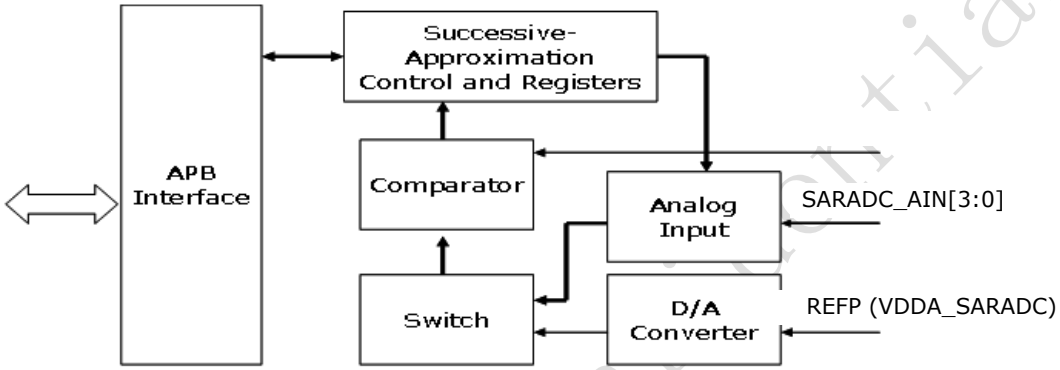


Fig. 42-1RK30xx SAR-ADC block diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input SARADC_AIN[3:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

42.3 Function Description

In RK30xx, SAR-ADC work at single-sample operation mode .

- Single-sample conversion

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC.The detailed timing diagram will be shown in the following.

42.4 Register description

42.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
SARADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
SARADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
SARADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
SARADC_DLY_PU_SOC	0x000c	W	0x00000000	delay between power up and start command

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

42.4.2 Detail Register Description

SARADC_DATA
Address: Operational Base + offset (0x0000)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:10	RO	0x0	reserved
9:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[9:0]).

SARADC_STAS
Address: Operational Base + offset (0x0004)
The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 0: ADC stop; 1: Conversion in progress.

SARADC_CTRL
Address: Operational Base + offset (0x0008)
The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable. 0: Disable; 1: Enable
4	RO	0x0	reserved
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down; 1: ADC power up and reset. start signal will be asserted（DLY_PU_SOC+2）sclk clock period later after power up
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 111 : Input source 0 (SARADC_AIN[0]) 110 : Input source 1 (SARADC_AIN[1]) 101 : Input source 2 (SARADC_AIN[2]) 100 : Input source 3 (SARADC_AIN[3]) Others : Reserved

SARADC_DLY_PU_SOC
Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x08	DLY_PU_SOC delay between power up and start command The start signal will be asserted（DLY_PU_SOC + 2）sclk clock period later after power up

42.5 Timing Diagram

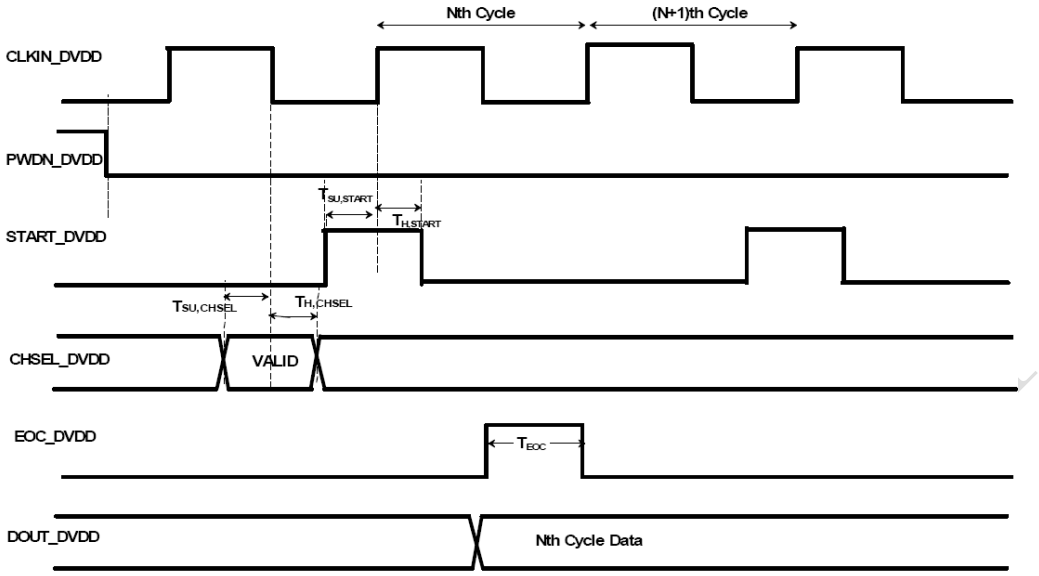


Fig. 42-2SAR-ADC timing diagram in single-sample conversion mode

42.6 Application Notes

The following is an example sequence of setting up A/D Converter, starting of conversion, and acquiring the result value.

- Power-down A/D Converter in SARADC_CTRL[3]
- Power-up A/D Converter in SARADC_CTRL[3] and select input channel of A/D Converter in SARADC_CTRL[2:0] bit
- Wait an A/D interrupt or poll the SARADC_STAS register to determine when the conversion is completed
- Read the conversion result in the SARADC_DATA register

Another, as for input clock period of SAR-ADC , it must be minimum 1000ns .

Chapter 43 TS-ADC

43.1 Overview

TS-ADC is a temperature-sensor IP providing an accuracy of +/-5 ºC without trim on the Temp sense core (for temp range -40 ºC to 125 ºC).

The IP comprises of a bipolar-based temperature-sensing cell that converts temperature into an analog voltage, and a 12-bit SAR ADC that operates on the sensed analog value to provide a 12-bit digital readout. The IP supports 2-channel input; 1 channel connected to internal temperature sensor cell and 1 channels for external temperature measurement. The sensed analog values are routed to the ADC. Internal or external sensors can be controlled by channel select bits.

43.2 Block Diagram

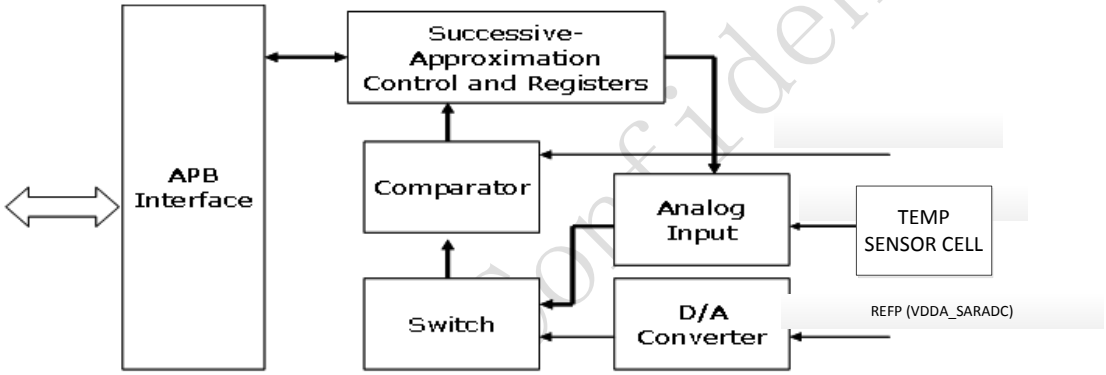


Fig. 43-1RK30xx TS-ADC block diagram

Successive-Approximate Register and Control Logic Block

This block is exploited to realize binary search algorithm, storing the intermediate result and generate control signal for analog block.

Comparator Block

This block compares the analog input SARADC_AIN[3:0] with the voltage generated from D/A Converter, and output the comparison result to SAR and Control Logic Block for binary search. Three level amplifiers are employed in this comparator to provide enough gain.

43.3 Function description

In RK30xx, TS-ADC work at single-sample operation mode .

- Single-sample conversion

This mode is useful to sample an analog input when there is a gap between two samples to be converted. In this mode START is asserted only on the rising edge of CLKIN where conversion is needed. At the end of every conversion EOC signal is made high and valid output data is available at the rising edge of EOC.The detailed timing diagram will be shown in the following.

- Temperature to code mapping

temp (C)	Code
-40	3800
-35	3792
-30	3783
-25	3774
-20	3765
-15	3756
-10	3747
-5	3737
0	3728
5	3718
10	3708
15	3698
20	3688
25	3678
30	3667
35	3656
40	3645
45	3634
50	3623
55	3611
60	3600
65	3588
70	3575
75	3563
80	3550
85	3537
90	3524
95	3510
100	3496
105	3482
110	3467
115	3452
120	3437
125	3421

43.4 Register description

43.4.1 Register Summary

Name	Offset	Size	Reset Value	Description
TSADC_DATA	0x0000	W	0x00000000	This register contains the data after A/D Conversion.
TSADC_STAS	0x0004	W	0x00000000	The status register of A/D Converter.
TSADC_CTRL	0x0008	W	0x00000000	The control register of A/D Converter.
TSADC_DLY_PU_SOC	0x000c	W	0x00000008	delay between power up and start command

*Notes:Size:**B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access*

43.4.2 Detail Register Description

TSADC_DATA

Address: Operational Base + offset (0x0000)
This register contains the data after A/D Conversion.

Bit	Attr	Reset Value	Description
31:12	RO	0x0	reserved
11:0	RO	0x000	adc_data A/D value of the last conversion (DOUT[11:0]).

TSADC_STAS
Address: Operational Base + offset (0x0004)
The status register of A/D Converter.

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RO	0x0	adc_status ADC status (EOC) 0: ADC stop; 1: Conversion in progress.

TSADC_CTRL
Address: Operational Base + offset (0x0008)
The control register of A/D Converter.

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6	RW	0x0	int_status Interrupt status. This bit will be set to 1 when end-of-conversion. Set 0 to clear the interrupt.
5	RW	0x0	int_en Interrupt enable. 0: Disable; 1: Enable
4	RW	0x0	start_convert Start of Conversion(START) Set this bit to 1 to start an ADC conversion. This bit will reset to 0 by hardware when ADC conversion has started.
3	RW	0x0	adc_power_ctrl ADC power down control bit 0: ADC power down; 1: ADC power up and reset.

Bit	Attr	Reset Value	Description
2:0	RW	0x0	adc_input_src_sel ADC input source selection(CH_SEL[2:0]). 111 : Input source 0 (SARADC_AIN[0]) 110 : Input source 1 (SARADC_AIN[1]) 101 : Input source 2 (SARADC_AIN[2]) 100 : Input source 3 (SARADC_AIN[3]) Others : Reserved

TSADC_DLY_PU_SOC
Address: Operational Base + offset (0x000c)
delay between power up and start command

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x08	DLY_PU_SOC delay between power up and start command

43.5 Timing Diagram

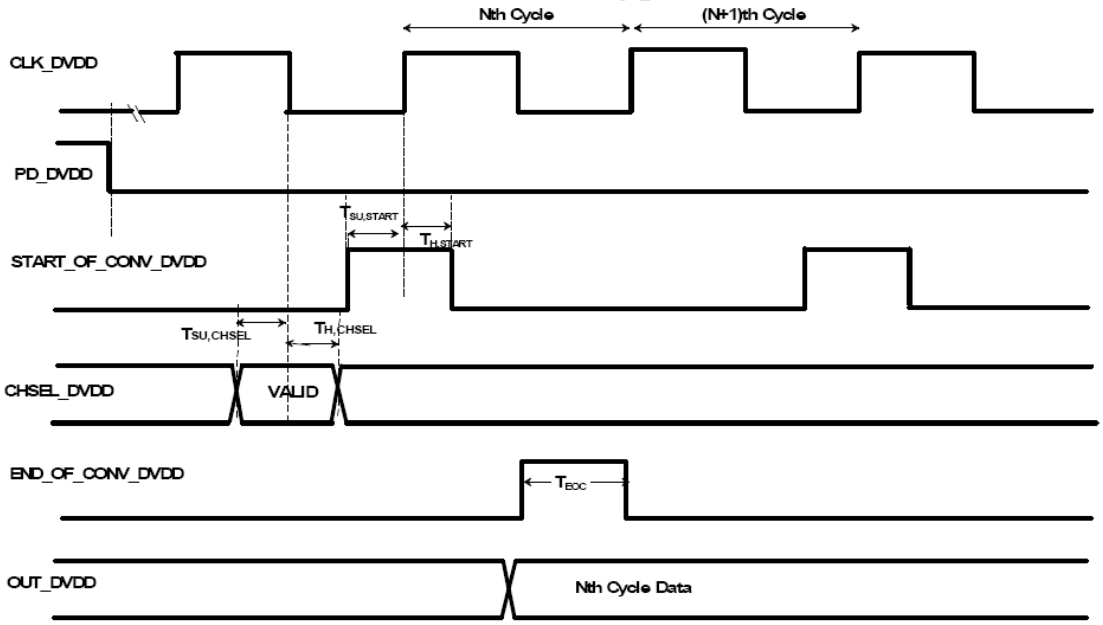


Fig. 43-2TS-ADC timing diagram in single-sample conversion mode

43.6 Application Notes

The following is an example sequence of setting up A/D Converter, starting of conversion, and acquiring the result value.

- Power-down A/D Converter in TSADC_CTRL[3]
- Power-up A/D Converter in TSADC_CTRL[3] and select input channel of A/D Converter in TSADC_CTRL[2:0] bit
- Wait an A/D interrupt or poll the TSADC_STAS register to determine

when the conversion is completed
- Read the conversion result in the TSADC_DATA register

Another, as for input clock period of TS-ADC , it must be minimum 20us .

Chapter 44 eFuse

44.1 Overview

In RK30xx, eFuse is organized as 32bits by 8 one-time programmable electrical fuses with random access interface. It is a type of non-volatile memory fabricated in standard CMOS logic process. The main features are as follows :

- Programming condition : EFUSE_VDDQ = 2.5V±10% , Program time : 10us ±1us .
- Read condition : EFUSE_VDDQ = 0V or floating or (2.5V±10%)
- Embedded power-switch
- Provide power-down and standby mode

For detailed information about eFuse, please refer to **RK30xx eFuse.pdf**。

Chapter 45 Chip Test Solution

45.1 Overview

In RK30xx, there are lots of test modes for different intention as the following table, which will be decided by signals test and testmode$_i$(i=0~3) , which is muxed with function GPIO.

Table 45-1 RK30xx test mode list

TEST	testmode3	testmode2	testmode1	testmode0	mode description
0	X	X	X	X	Normal mode
1	0	0	0	0	SOC general scan test mode
1	0	0	0	1	SOC at-speed scan test mode
1	0	0	1	0	SOC general memory bist mode / SOC general Rom bist mode
1	0	0	1	1	SOC at-speed memory bist mode / SOC at-speed Rom bist mode
1	0	1	0	1	Cortex-A9(L1) at-speed memory bist mode
1	0	1	1	0	Cortex-A9(L2) at-speed memory bist mode
1	0	1	1	1	SOC leakage test mode
1	1	0	0	0	USB PHY bist mode
1	1	0	0	1	DDR PHY test mode
1	1	0	1	0	HDMI PHY test mode
1	1	0	1	1	PLL test mode
1	1	1	0	0	eFuse test mode
1	1	1	0	1	10bit SAR-ADC test mode
1	1	1	1	0	12bit TS-ADC test mode

The key application notes are as follows：

- For SOC at-speed scan test mode, RK30xx provides six test solution with different internal test frequency, which is decided by signals test_freq_sel$_i$ (i=0~2), and need four input clocks from XIN24M,TCK,CLK32K andCLK27M input pins all with 24MHz frequency.
- For SOC at-speed memory bist mode, SOC at-speed Rom bist mode and Cortex-A9(L1/L2) at-speed memorybist mode, RK30xx provides six test solution with different test frequency, which is decided by signals test_freq_sel$_i$ (i=0~1), and only need one input clock from XIN24M input pin with 24MHz frequency.
- For SOC general scan test mode, RK30xx need three input clocks from TCK,CLK32K andCLK27M input pins with 24MHz frequency.
- For SOC general memory bist modeand SOC general rom bist mode,

RK30xx only need one input clock from XIN24M input pin with 24MHz frequency.

- For DDR PHY test mode and HDMI PHY test mode, RK30xx provides six test solution with different test frequency, which is decided by signals test_freq_sel$_i$ (i=0~1), and only need one input clock from XIN24M input pin with 24MHz frequency.
- For all other test modes for internal Analog macro inside RK30xx , only need one input clock from XIN24M input pin with 24MHz frequency.
- The reset signal for all of test modes is all from NPOR input pin directly.

In the above description, some miscellaneous signals related with test mode are not dedicated pins, always mux with function pin except TEST. The detailed information is as the following list.

Table 45-2 RK30xx iomux for misc signal in test mode

test pins	test IO	function pins	IOMUX setting
testmode0	I	GPIO6_B[0]	TEST = high level
testmode1	I	GPIO6_B[1]	TEST = high level
testmode2	I	GPIO6_B[2]	TEST = high level
testmode3	I	GPIO6_B[3]	TEST = high level
test_freq_sel0	I	GPIO6_B[5]	TEST = high level
test_freq_sel1	I	GPIO6_B[6]	TEST = high level
test_freq_sel2	I	GPIO6_B[7]	TEST = high level

For detailed information about Chip Test Solution, please refer to **RK30xx Chip Test Solution.pdf**。