EM8622L

Digital Media Processor with Multiple A/V Codec Support

Preliminary Datasheet - Draft

May, 2005

Copyright Information

Sigma Designs products are sold by description only. Sigma Designs reserves the right to make changes in circuit design and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that the datasheet is current before placing orders. Information furnished by Sigma Designs is believed to be accurate and reliable. However, no responsibility is assumed by Sigma Designs or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Sigma Designs or its subsidiaries.

Disclaimer

Sigma Designs Inc. makes no warranty, express, statutory, implied or by description regarding the information set forth herein or regarding the freedom of the described devices from intellectual property infringement. Sigma Designs Inc. makes no warranty of merchantability or fitness for any purpose.

Sigma Designs Inc. shall not be responsible for any errors that may appear in this document. Sigma Designs Inc. makes no commitment to update or keep current the information contained in this document.

Confidentiality

This document contains confidential information, trade secrets or both that are the property of Sigma Designs Inc. It is to be treated as confidential under the Non Disclosure Agreement (NDA), which has been signed by the obtainer. Reproduction or transmission in any manner to others in whole or in part is prohibited without prior written permission from Sigma Designs Inc.

Life Support Policy

The product(s) described in this document is not intended for use as critical component(s) in life support devices or systems without prior written permission from Sigma Designs Inc.

- Life support devices or systems are devices or systems which, (a) are intended as surgical implant(s) into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.

- A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Trademarks

EM8622L is a trademark of Sigma Designs Inc. All other trademarks belong to their respective companies.

Address

Sigma Designs Inc.
1221 California Circle
Milpitas CA USA - 95035
Tel: 408.262.9003
Fax: 408.262.9740
www.sigmadesigns.com

Document #: EM8622L-DS-A-03

Table of Contents

Conventions

This section presents the acronyms, abbreviations, units of measurement and other conventions used in this datasheet.

Acronyms and Abbreviations

The acronyms and abbreviations used in this datasheet are listed alphabetically in the table below:

Table 0-1. **Acronyms and abbreviations**

Acronym / Abbreviation	Definition
2D/3D	2 Dimensional/3 Dimensional
AC	Alternating Current
ADC	Analog-to-Digital Converter
ATA	AT Attachment
AV or A/V	Audio Visual
BGA	Ball Grid Array
BPP	Bits per Pixel
CMOS	Complementary Metal Oxide Semiconductor
CODEC	Coder/Decoder
CPU	Central Processing Unit
CSS	Cascading Style Sheets or Content Scrambling System
D/A	Digital-to-Analog
DAA	Data Access Arrangement
DAC	Digital-to-analog Converter
DC	Direct Current
DDR SDRAM	Double Data Rate Synchronous DRAM
DMA	Direct Memory Access
DRAM	Dynamic Random-access Memory
DSL	Digital Subscriber Line

Table 0-1. **Acronyms and abbreviations (Continued)**

Acronym / Abbreviation	Definition
DSP	Digital Signal Processor
DVD	Digital Versatile Disc or Digital Video Disc
DVI	Digital Visual Interface
EJTAG	Enhanced Joint Test Action Group
EIA	Electronic Industries Alliance
FCS	Frame Check Sequence
FIFO	First In/First Out
GPIO	General Purpose Input/Output
HDMI	High-Definition Multimedia Interface
HDTV	High-Definition Television
I	Input
I/O	Input/Output
I^2C	Inter Integrated Circuit
IDE	Integrated Drive Electronics
IEC	International Electrotechnical Commission
IPTV	Internet Protocol TV
IR	Infrared
IRQ	Interrupt Request Line
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
LCD	Liquid Crystal Display
LSB	Least Significant Bit/Byte
MIPS	Millions of Instructions per Second
MMU	Memory Management Unit
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit/Byte
O	Output
OSD	On Screen Display
P/U	Pull-up Resistor
PCB	Printed Circuit Board

Table 0-1. **Acronyms and abbreviations (Continued)**

Acronym / Abbreviation	Definition
PCI	Peripheral Component Interconnect
PID	Program Id
PIP	Picture In Picture
PKI	Public Key Infrastructure
PLL	Phase Locked Loop
PVR	Personal Video Recorder
RAM	Random Access Memory
RGB	Red Green Blue
RISC	Reduced Instruction Set Computer
ROM	Read Only Memory
RTC	Real-time Clock
S/PDIF	Sony/Philips Digital Interface
SDTV	Standard Definition Television
SPI	Synchronous Parallel Interface
SRAM	Static Random Access Memory
SSI	Synchronous Serial Interface
SSI	Server-side Include or Single-system Image
TDMX	Transport Demultiplexer
TLB	Translation Look-aside Buffer
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VOD	Video On Demand
YCbCr	Y is brightness (luma), Cb is blue minus luma (B-Y) and Cr is red minus luma (R-Y)

Units of Measurement

The units of measurement used in this datasheet are listed alphabetically in the table below:

Table 0-2. **Units of measurement**

Symbol	Unit of measurement
µA	microampere
µF	microfarad
µs	microsecond (1,000 nanoseconds)
^{0}C	degree Celsius
GB	gigabyte
bpp	Bits Per Pixel
Hz	Hertz (Cycle Per Second)
kohm	kiloohm
Kb	kilobit
KB	kilobyte (1,024 Bytes)
Kbps	kilobit per second
KBps	kilobyte per second
KHz	kilohertz
mA	milliampere
Mbps	megabit per second
MBps	megabyte (1,048,576 bits) per second
Mb	megabit
MB	megabyte (1,048,576 bytes)
MHz	megahertz (1,000 kilohertz)
ms	millisecond (1,000 micro seconds)
ns	nanosecond
V	volt
W	watt

General Conventions

Numbers and Number Bases

- Binary numbers are enclosed in single quotation marks when in text, e.g., ‘11’ designates a binary number.
- Binary numbers are written with a lower case ‘b’ suffix. e.g., 16b.
- Binary numbers larger than four digits are written with a space dividing each group of four digits, as in 1011 0101 1010b.
- All other numbers are decimal

Naming Conventions

- The register acronyms appear in capital letters such as SDIOVH or SDIN_EGD_HDG
- Register bits are listed in square brackets MSB-to-LSB separated by a colon mark, e.g., SDHE[3:0].
- TBD indicated that the values are ‘to be determined’, NA indicates ‘not available’ and NC indicates that a pin is ‘no connect’.

Datasheet Definitions

The following table gives the general definitions of the status of the datasheet:

Table 0-3. **Datasheet status definitions**

Datasheet Status	Description
Advance	Contains information on a product under development. Features, functionality, and parametric information are target goals, and are subject to change.
Preliminary	Contains information on a product under development that is not fully characterized. The parametric information contains target parameters that are subject to change.
Final	Contains information on a product that is fully characterized and is in full production.

1 Introduction

Block Diagram of EM8622L

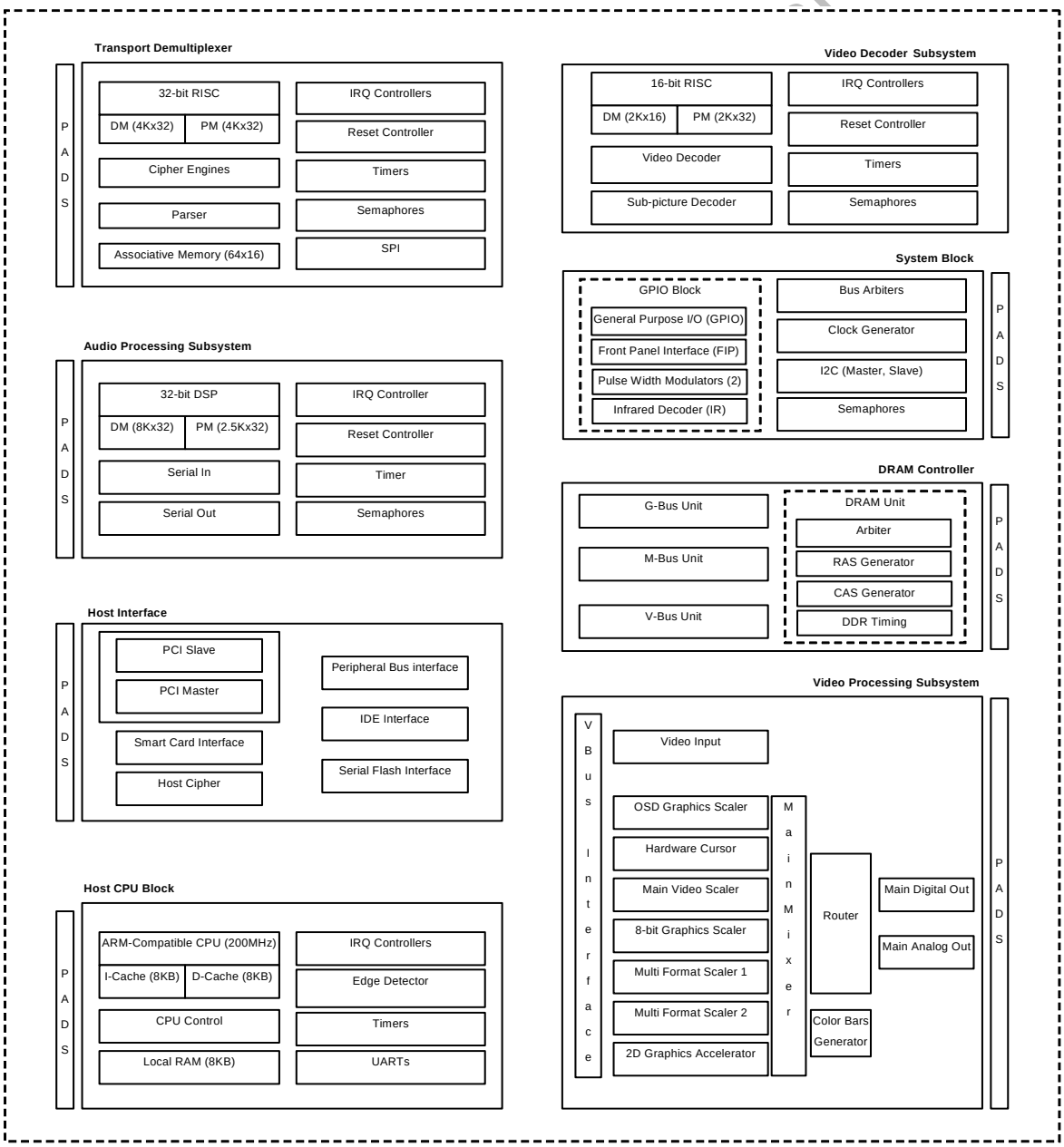


Figure 1-1. **EM8622L block diagram**

Main Features of EM8622L

- Host CPU
  - Embedded ARM-compatible processor (200MHz) for operating system, middleware and applications designed for use with applications that require ARM (version 4T) instructions
  - 16K program and 16K data memory
- IO standards
  - 32-bit PCI v2.1 (33 or 66MHz) bus master/slave/host interface with optional AES, 3 DES or DES security
  - Flexible peripheral bus supports IDE (ATA/ATAPI-4), CompactFlash, local bus and parallel flash with optional AES, 3 DES or DES security
  - IDE (ATA/ATAPI-4) or DVD loader (supports Samsung, Sanyo, Sony and Thomson DVD loaders)
  - Front panel controller interface supports NEC uPD16311, NEC uPD16312, PTC PT6311 and PTC PT6312 front panel controllers
  - Smart card interface (ISO 7816)
  - I^2C master/slave interfaces
  - Single 8-bit parallel SPI transport stream interface, or dual serial transport stream interfaces
  - I^2S interface up to 20Mbps
  - Two UARTs
  - Local bus interface supporting Ethernet chipsets, MPEG encoders and other external devices
  - Up to 48 general purpose I/O pins
- Video decoding standards
  - MPEG-1, MPEG-2 MP@HL up to 1920x1080i30 or 1920x1080p30 resolution
  - MPEG-4. 2 ASP@L5. Rectangular shape video decoding up to 1280x720p30 resolution, support for B Pictures, data partitioning and error resiliency.
  - WMV9/VC-1 MP@HL up to 1920x1080p30 resolution, progressive sources only
  - VC-1 AP@L3 up to 1920x1080p30 resolution
  - MPEG-4.10 (H.264) BP@L3 up to 720x480p30 or 720x576p25 resolution, including FMO and ASO
  - MPEG-4.10 (H.264) MP@L4.1 and HP@L4.1 up to 1920x1080i30 or 1920x1080p30 resolution
  - Baseline JPEG up to 1920x1080 resolution
  - DVD-Video and Superbit DVD
  - Error concealment, deblocking filter
  - Elementary video stream bit rate
    * MPEG-2 SDTV (HDTV): 15 (30) Mbps maximum

    * MPEG-4.2 SDTV (HDTV): 15 (30) Mbps maximum
    * MPEG-4.10 (H.264) SDTV (HDTV): 15 (30) Mbps maximum
    * WMV9/VC-1 SDTV (HDTV): 15 (30) Mbps maximum

- Video interfaces
  - Flexible video and graphics input interfaces support multiple video sources, DVI and HDMI receivers, 3D graphics chips
  - 8-bit video input interface
  - 32-bit graphics input interface
  - Video outputs
    * NTSC/PAL composite analog output with optional Macrovision v7.1.L1 protection (54MHz, 12-bit DAC)
    * NTSC/PAL s-video analog output with optional Macrovision v7.1.L1 protection (54MHz, 12-bit DACs)
    * Analog YPbPr / RGB with optional Macrovision v7.1.L1 and v1.2 protection in 480i, 576i, 480p and 576p YPbPr output modes (12-bit DACs, interlaced or progressive, SDTV or HDTV resolution)
    * 150MHz YCbCr/RGB digital video output interface; 8-bit 4:2:2 YCbCr data, 16-bit 4:2:2 YCbCr data, 24-bit 4:4:4 YCbCr data, 24-bit RGB data (888), BT.601, BT.656, or VIP 2.0, ‘video valid’ output signal, master or slave timing

- Video processing
  - Brightness, color and contrast controls for each output port
  - Hardware cursor (4096 pixels, 4-bpp, up to 255 pixels horizontally and vertically)
  - 2D graphics accelerator (up to 100 M samples per second operation for most operations)
    * Line, Rectangle, Ellipse and Circle: generate a single-color line, rectangle, ellipse or circle with optional gradient fill
    * Blend: alpha blend one rectangular region onto another
    * Move: move a rectangular region to another location
    * Replace: modified version of Move
    * Raster Operations: standard 256 Boolean operations
    * OpenType font rendering acceleration
  - 32-bit OSD with flicker filtering and scaling
  - Optional deinterlacing of interlaced sources
  - Arbitrary scaling of video and OSD up to 1920x1080 pixels
  - Alpha mixing of video, graphics, cursor and OSD
  - On screen display
    * High resolution, true-color OSD support
    * 2, 4, 7, and 8-bpp from 24-bit palette
    * Programmable OSD scaler
    * Programmable flicker filter for interfaced output modes
    * Alpha blending over video (8-bit)

- Audio interfaces
  - Audio DSP supports a wide range of audio codecs
  - Audio inputs
    * I^2S or S/PDIF serial digital audio input
  - Audio outputs
    * S/PDIF (IEC 60958) serial digital audio output for 2.0 linear PCM and compressed Dolby Digital, DTS, WMA Pro and MPEG
    * Three I^2S serial digital outputs support 5.1-channel audio
- Audio decoding standards
  - 16-bit linear PCM with HDCD support
  - MPEG-1 and MPEG-2 Layers I, II and III (MP3) 2.0
  - MPEG-2 and MPEG-4 AAC-LC 2.0
  - MPEG-2 and MPEG-4 HE-AAC 2.0
  - MPEG-4 BSAC 2.0
  - Dolby Digital 5.1
  - DTS 5.1
  - WMA9@L3 2.0, WMA9 Pro@M2 5.1
- Transport formats
  - Transport input interfaces
    * One 8-bit SPI or two SSI, with polarity control of data valid signal
    * PCI, IDE, Local Bus
  - Transport demux supports 2 dedicated PIDs (audio, video), 16 general PIDs, 64 PIDs from associative memory
  - Transport input bit rate: 40Mbps maximum (aggregate)
- Media formats
  - DVD-Video, Superbit DVD, SVCD (IEC 62107-2000), VCD 1.x and 2.0
  - DVD-R, DVD-RW, DVD+R, DVD+RW (conditional)
  - Audio CD (with optional HDCD), CD-R, CD-RW, CompactFlash
    * WMA, JPEG, MP3 and MPEG-41 AVI files using ISO 9660 or HighMAT™ format
  - Picture CD (JPEG files using ISO 9660 format)
  - Navigation software, HighMAT™ support
- Streaming formats
  - ISMA (Internet Streaming Media Alliance) MPEG-4
  - MPEG-2, MPEG-4
  - WM9 with DRM
  - MPEG-4.10 (H.264) and VC-1 AP over MPEG-2 transport

- Package
  - Packaged in 469 ball plastic ball grid array (BGA)
  - Advanced 0.13µ low voltage CMOS technology
- Power management
  - 1.2V core with 3.3V I/O (5V tolerant)
  - Low power and power down modes

Main Components of EM8622L

The EM8622L is an advanced, single-chip audio/video decoder that provides highly-integrated solutions for HDTV, IPTV, DVD, MPEG-4.10 (H.264) and WMV9/VC-1 decoding. It incorporates flexible, advanced audio/video processing, enabling cost-effective solutions for consumer appliances, such as digital media players, IPTV set-top boxes, networked DVD players and digital televisions.

The EM8622L includes optimized features for tightly embedded applications such as TV/PDP integration, streaming video endpoints, and multifunction consumer appliances. The device also includes features that enable designers to easily incorporate advanced capabilities such as A/V streaming, progressive DVD playback, Video-on-Demand (VOD), Personal Video Recording (PVR) and Picture-in-Picture (PIP) into their products.

In addition, the EM8622L supports numerous popular media formats including DVD-Video, Superbit DVD, DVD-Audio, SVCD, VCD1.x, VCD2.0 and CD/CD-R/CD-RW (audio, JPEG, MP3 and MPEG-4 AVI files). It also supports ISMA MPEG-4 streaming format, and MPEG-4 over MPEG-2 transport streaming.

The EM8622L architecture is composed of various hardware functional units – several incorporating custom-designed processor modules – interconnected by multiple high-speed synchronous data buses. Although the details of the on-chip buses are beyond the scope of this document, the primary buses and their functions are described below.

A 32-bit G-Bus connects the integrated ARM-compatible RISC processor with each functional unit. It provides access to the programmable configuration, control and status registers contained within each unit. The state of the device is initialized, controlled, and reconfigured as necessary through this bus. The G-Bus also supports direct access to the memory areas controlled by two memory controllers, PCI, flash memory areas and 32-bits of address and data (4G dwords addressable). It is an arbitrated, multi master bus.

The M-Bus provides the data path that allows each functional unit to communicate with the external memory controller. It provides a 64-bit data path resulting in a peak data bandwidth of 1332MB/sec at a system clock frequency of 166MHz. The M-Bus arbitrates access among 17 DMA channels. An arbitration algorithm provides programmable bus bandwidth to be allocated and guaranteed to each DMA channel.

The V-Bus provides a dedicated high-speed data path between the video processing engine and the external memory controller.

In addition to these device-wide buses, the various functional units typically have one or more dedicated local buses within the unit. These buses are not further described in this document.

The main modules of the EM8622L are briefly described below. The modules are discussed in detail under their respective chapters.

Host CPU Block

The host CPU block of the EM8622L contains an internal 200MHz ARM-Compatible RISC CPU with its instruction and data caches to support the embedded operating system, middle ware and applications required for the consumer appliances. The CPU is designed for use with applications that require ARM (version 4T) instructions. Typically, the µClinux embedded OS is used. Alternately, an external host CPU (such as x86, MIPS or ARM) may be used. Since MMU is not included, Linux and WinCE operating systems are not supported.

In addition to the processor itself, the host CPU block contains the following additional resources: 2Kx32 (8KB) local memory, two interrupt controllers, edge detection logic, two programmable timers and two Universal Asynchronous Receiver/Transmitters (UARTs)

DRAM Controller

The EM8622L contains double-data rate synchronous DRAM (DDR-SDRAM) controller operating at the memory clock rate. Because the DDR technology transfers data on both edges of the clock, the effective burst data bandwidth of the controller is 1.6GB/sec when using the full 32-bit interface. The DRAM controller can interface up to 128MB of external DDR SDRAM, using a 16 or 32-bit wide data bus.

System Block

The EM8622L system block contains the following modules: A GPIO block, I^2C master and slave and a clock generator. The GPIO block in turn contains General Purpose I/O (GPIO), Front Panel Interface (FIP) controller, Pulse Width Modulator (PWM) and an infrared decoder.

The general purpose I/O controller provides 16 pins of general purpose control signals and logic to help eliminate the glue logic necessary for system integration. Its functions include indicating the system operation and controlling other devices.

The Front Panel Interface (FIP) controller directly supports the NEC uPD16311, NEC uPD16312, PTC PT6311 and PTC PT6312. Other front panel controllers may be used by interfacing to the UART or the GPIO.

The infrared input allows the interfacing to an external IR receiver. The NEC and the Philips RC5/RC6 IR formats, commonly used by consumer equipment are supported.

The I^2C master and slave interfaces enable the EM8622L to read from and write to external devices. This I^2C master controller, which supports the synchronous Inter Integrated Circuits (I^2C) serial protocol, enables the host CPU to access an external I^2C slave device using a simplified register interface. A separate slave interface allows the EM8622L to be the target of I^2C transactions initiated by an external master.

The clock generator contains one audio clock, two video clocks, one system clock and one CPU clock. The clock generator creates two high speed (up to 200MHz) clocks from a 27MHz external clock using two programmable PLLs, and creates the main system clock and multiple video and audio clocks by dividing either one of the high speed clocks or the 27MHz reference.

Host Interface

The host interface unit provides the interface between the primary internal buses (G-bus and M-bus) and the PCI, the peripheral bus, smart card interface, the serial flash and the IDE interface.

The EM8622L supports both a PCI and a multimode ‘peripheral’ bus for system-level interconnection. The PCI bus implementation is a version 2.1 compliant, 32-bit wide bus capable of operating at 33 or 66MHz.

The separate Peripheral Bus Interface (PBI) can operate in several modes with programmable cycle timings, which can be varied on a cycle-by-cycle basis. The bus can operate as a general-purpose interface, or an 'ISA-like' bus for connecting external devices, or an IDE bus for attaching storage devices, or a memory bus for directly attaching asynchronous memory such as a parallel flash ROM.

When being used as a general-purpose interface, the PBI can operate in either a separate address and data mode (14 address/16 data) or a multiplexed address/data mode. In the IDE mode, the interface can support ATA device attachments.

The host block also contains an interface to an external serial flash and a smart card interface.

Video Decoder Subsystem

The EM8622L video decoder subsystem executes the video decoding algorithms supported by the EM8622L. Its architecture is a hybrid of both processor-based and hardwired logic approaches.

The video decoder engine consists of a proprietary 16-bit RISC CPU which is augmented by a number of hardware functional units. These functional units perform the most compute-intensive portions of the video decompression algorithms supported by the EM8622L.

Video Processing Subsystem

The EM8622L video processing engine provides sophisticated display processing, formatting and output capabilities.

The video processing and display unit (VPD) has extensive capabilities for retrieving graphics and video images from the memory, formatting the images as needed, mixing the images and then presenting the video stream for display in a required format. Other capabilities provided by the VPD include hardware-assisted 2D graphics acceleration, and support for an external video input port.

The 2 available video outputs consist of, digital output and analog output. The digital output supports 8, 16 or 24-bit output, RGB or YPbPr data format and the analog output supports component RGB, YPbPr, S-video and composite video. The digital and analog component outputs can each support output formats up to 1920x1080p.

Audio Processing Subsystem

The EM8622L contains an integrated audio subsystem based on a custom-designed 32-bit digital signal processor (DSP). Audio decoding and processing algorithms are implemented on the DSP. This firmware-based approach gives great flexibility for accommodating future audio standards or specialized audio requirements. The audio unit provides three I^2S output channels, one S/PDIF output channel, and one I^2S or S/PDIF audio input channel.

Transport Demultiplexer

The EM8622L includes an on-chip, RISC processor-based transport demultiplexer (TDMX) unit. The transport demultiplexer block is capable of handling up to three multiprogram bitstreams of up to 40Mbps each, with an aggregate total of up to 40Mbps.

Application Example: Networked DVD Player

The networked DVD player example below provides the typical functionality required for a networked DVD player. System integration requires very little external logic since the SMP8632 provides most of the features including:

- Progressive DVD-Video, DVD-Audio, MPEG-4.10 (H.264) and WMV9/VC-1 playback
- Interlaced or progressive YPbPr or RGB video outputs
- NTSC/PAL composite and S-video outputs
- 5.1-channel and S/PDIF audio outputs
- I^2C bus master function for controlling other chips
- ARM-Compatible CPU for operating system, middleware and applications

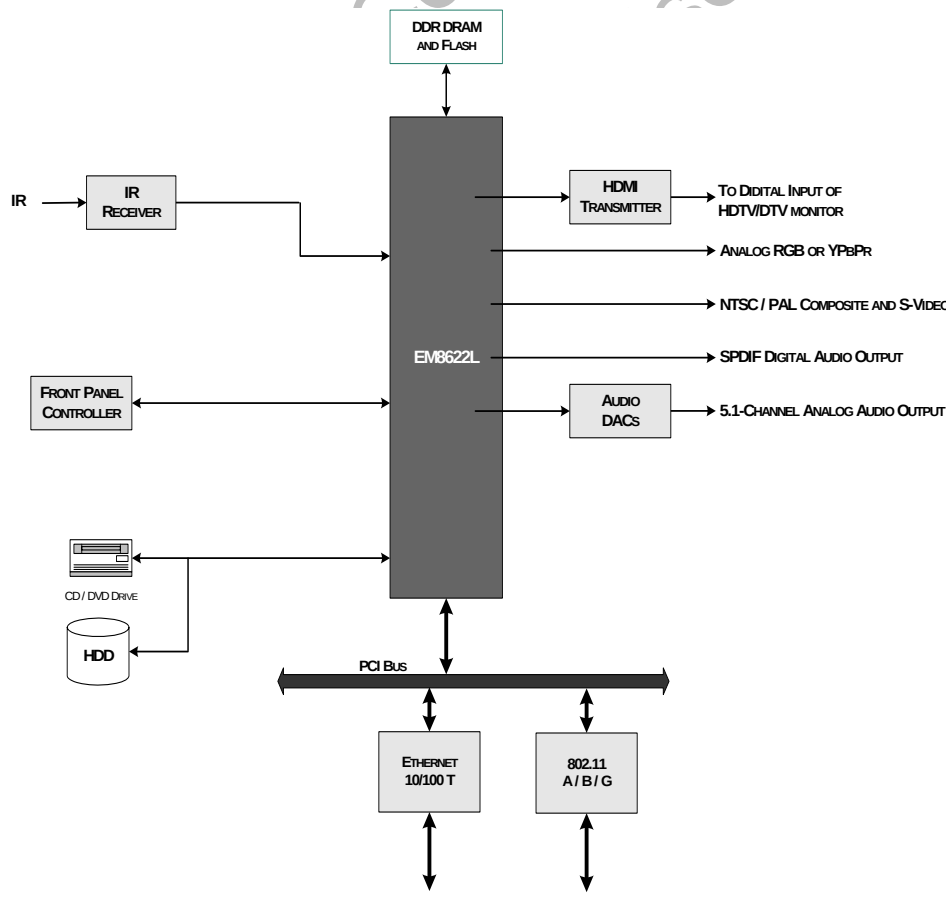


Figure 1-2. **Application example - networked DVD player**

Application Example: Digital Media Adapter or IPTV Set-top Box

The Digital Media Adapter or IPTV Set-top Box application requires very little external logic since the EM8622L provides most of the features including:

- Decoding of MPEG-1, MPEG-2, MPEG-4. 10 (H.264) and WMV9/VC-1 content
- 2D graphics, OSD and deinterlacing
- Interlaced or progressive YPbPr or RGB video outputs
- NTSC/PAL composite and S-video outputs
- 5.1-channel and S/PDIF audio outputs
- ARM-Compatible CPU for operating system, middleware and applications

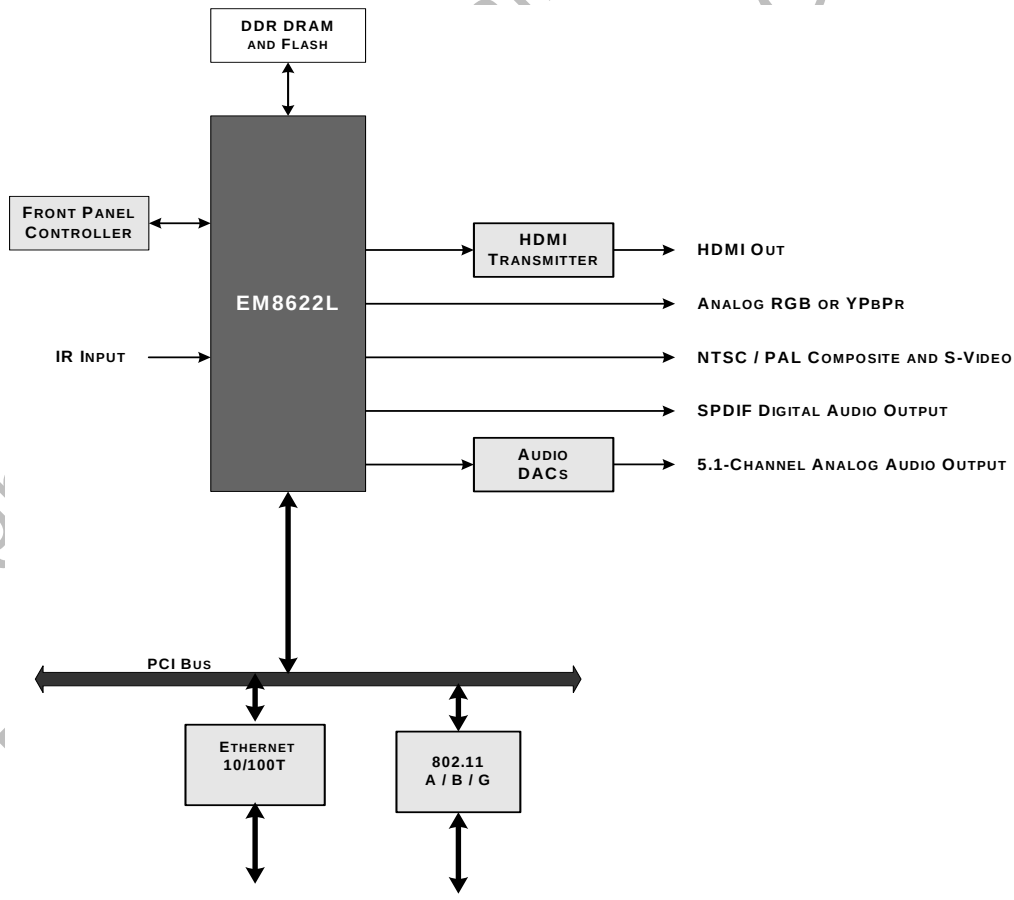


Figure 1-3. **Application example - digital media adapter or IPTV set-top box**

2 Host CPU Block

Block Diagram of Host CPU Block

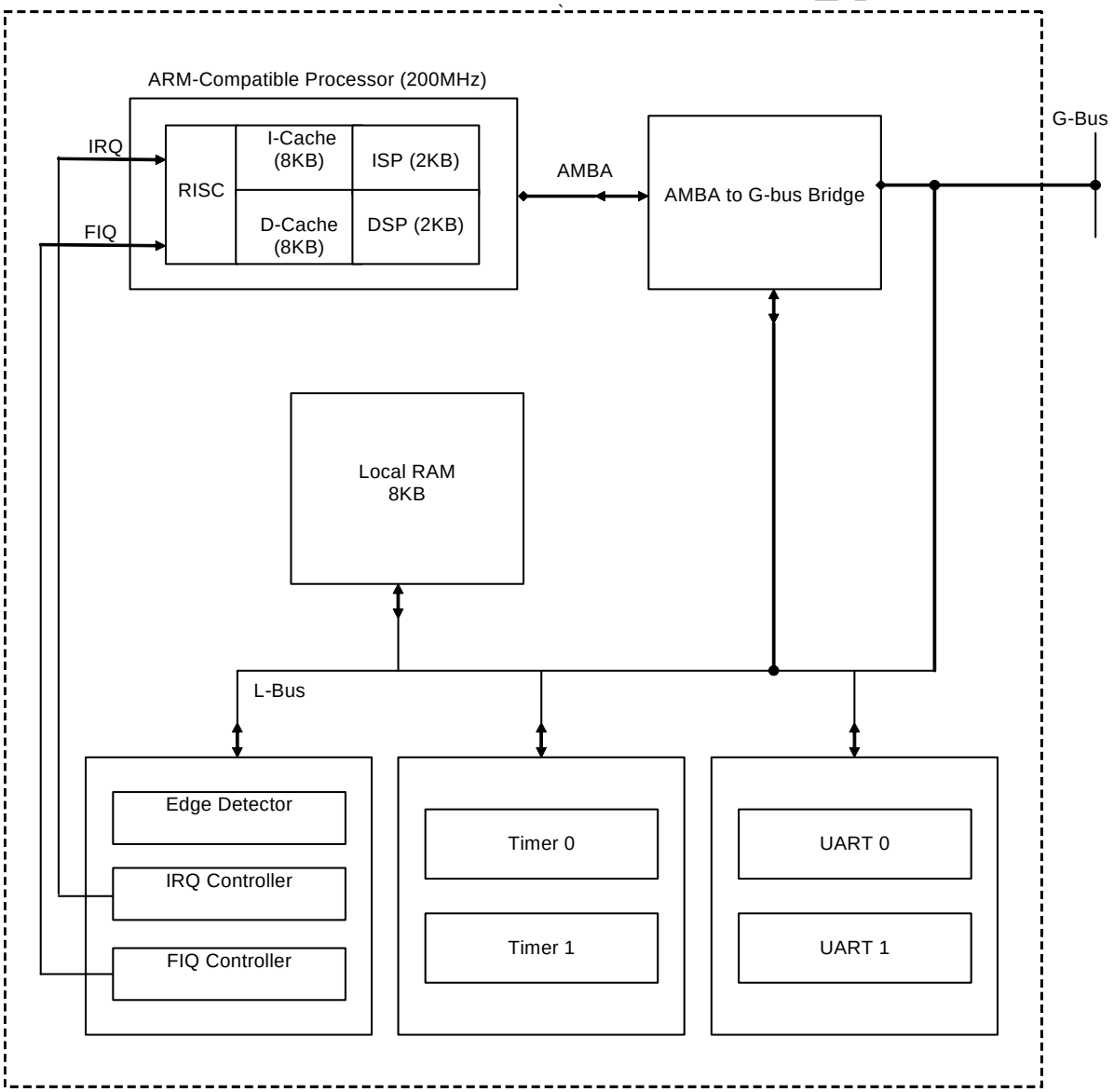


Figure 2-1. **Block diagram of host CPU block**

Introduction

The host CPU block of the EM8622L contains an internal 200MHz (~260 conforming DMIPS) ARM-compatible processor with its instruction and data caches to support the embedded operating system, middleware and applications required for the consumer products. The CPU is designed for use with applications that require ARM (version 4T) instructions. Typically, the µClinux embedded OS is used. Alternately, an external host CPU (such as x86, MIPS or ARM) may be used. Since MMU is not included, Linux and WinCE operating systems are not supported.

In addition to the RISC processor itself, the host CPU block contains the following additional resources:

- 2Kx32 (8KB) local memory
- Two interrupt controllers, IRQ and FIQ
- Edge detection logic
- Two programmable timers
- Two Universal Asynchronous Receiver/Transmitter (UARTs)

The CPU accesses the G-Bus as a master. A bridge allows the CPU to be a G-Bus master, and access all the G-Bus mapped local resources, including the DRAM through the L-Bus. Most of the processor software is executed from the DRAM. For the processor, an AMBA -to-G-Bus bridge is used.

Other components of the host CPU block reside on a local bus and can be accessed by either the RISC or other G-Bus masters. The host CPU block connects to the rest of the chip via the G-Bus.

A local RAM, accessible from the G-Bus can be used to share various types of information between the G-Bus masters. Also, critical processor code, such as reset vectors or interrupt service routines can be stored in the local RAM.

Central Processor Unit (CPU)

Introduction

The EM8622L uses a 200MHz (~260 conforming DMIPS) ARM-compatible processor.

The processor core is a high-performance, low-power, 32-bit RISC processor. It is highly portable across processes. The processor is ideally positioned to support new products for emerging segments of the digital consumer, network, systems and information management markets, enabling new tailored solutions for embedded applications. The EM8622L takes full advantage of all these features.

The EM8622L processor a 32-bit privileged resource architecture that is used with applications that require ARM (version 4T) instructions. It contains direct mapped 16KB instruction and 16KB data caches. Both caches support locking on a per-entry basis. The cache line size is 16-byte, requiring 4 sequential memory transfers on a cache line fill operation.

The processor cache allows support for up to 8 separate memory regions or ‘pages’. Each region can contain specific cacheability attributes for the instruction cache, data cache and the write buffer. In addition, multiple levels of access permissions can be attached to each memory region. Memory prefetching is supported on a per-instruction basis.

Features

- 32-bit RISC architecture
- 5-stage pipeline
- Separate, 16KB, direct-mapped instruction and data caches
  - Support for up to 8 memory regions, with memory access protection for each region
  - Data cache write-back and write-through support on a per-entry basis
  - 16-byte fixed line size with cache line locking support on a per-entry basis
  - Support for both Supervisor mode and User mode on cache accesses
  - 4-deep write buffer for data cache writebacks
  - Memory prefetching on a per-instruction basis and Memory prefetch and lock in one operation
  - Burst transfers on cache line fill operations

Block Diagram

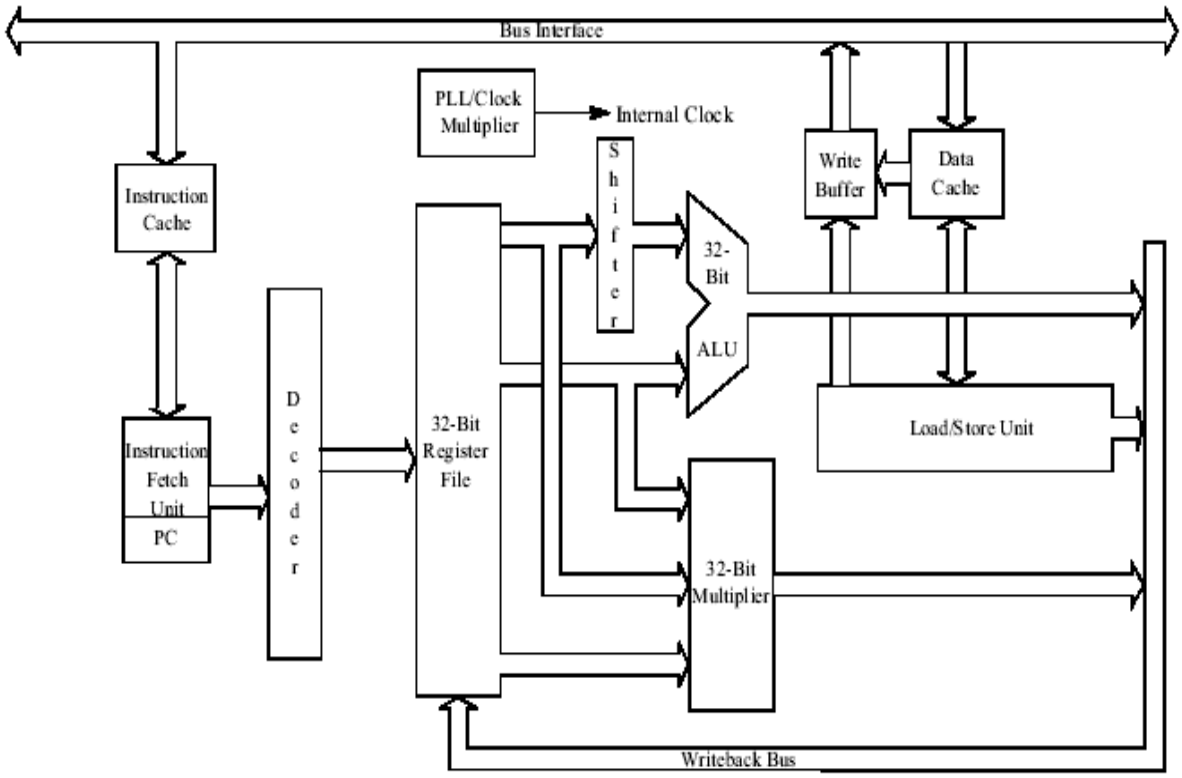


Figure 2-2. **EM8622L ARM-compatible processor block diagram**

Functional Description

The block diagram of the EM8622L ARM-compatible processor is shown above. The following subsections describe each block in the diagram.

Main Blocks of the EM8622L Processor

Instruction Fetch Unit

The instruction Fetch Unit fetches one instruction per clock cycle and contains a 3-deep FIFO for instruction storage during pipeline stalls. If the pipeline is stalled, then the instruction fetch continues until the FIFO becomes full, at which time the fetching stops.

The program counter (PC) is used to increment the address values used to access the memory. In the 32-bit mode, the counter increments by 4 each time an instruction is fetched. In the 16-bit mode, the counter increments by 2 each time an instruction is fetched. All instruction fetching is performed on the physical address value, eliminating the need for internal virtual to physical address translation.

Primary Instruction Cache

The processor's primary cache design includes a direct mapped, 16KB instruction cache. It contains a fixed line size of 128-bits and incorporates a 32-bit wide data SRAM and a Tag SRAM.

Decoder

The decoder is used to decode the instructions prior to being written to the register file. The 32-bit instructions are decoded and appropriate signals are sent to the core indicating the type of operation to be performed.

32-bit Register File

The 32-bit General Purpose (GP) Register File stores the operands and the results of a computation. The processor accesses the GP register file in one of the 6 operating modes: User, FIQ, IRQ, Supervisor, Abort and Undefined Instruction.

Shifter

The shifter performs logical and arithmetic shifting based o the type of instruction being executed. The processor's instruction set incorporates certain shift operations into the instructions and hence does not require a separate shift operation to be performed.

Arithmetic Logic Unit (ALU)

The Arithmetic Logic Unit accepts two operands and associated control signals, one from the register file and one from the shifter. The ALU processes all the operations except multiply. These include move, load/store, data processing and coprocessor operations.

32-bit Multiplier

The EM8622L processor contains a 32-bit multiplier that performs signed and unsigned multiply and multiply-accumulate operations. The multiplier requires 2 cycles to perform a multiply-accumulate operation. In the first cycle 3 operands are provided to the multiplier and the actual multiply operation is performed. In the second cycle the 4^{th} operand is provided and the accumulate operation is performed. The multiplier requires 2 operands to perform the multiply, and another 2 operands to perform the accumulate.

The processor always generates a 64-bit value on a multiply or multiply-accumulate operation. The lower 32-bits of the result are stored to a GP register whose location is defined in the instruction. The upper 32-bits are stored to the CP15 RdHi register (CP15-11). The CP15-11 is the register 11 in the CP15 control register set.

Unlike a multiply long or multiply-accumulate long operation, generating a 64-bit result on multiply and multiply-accumulate operations allows the upper and the lower halves of the 64-bit result to be written to nonsequential registers. For a long multiply or multiply-accumulate operation, the 64-bit result is stored to 2 sequential GP registers whose locations are defined in the instruction.

The EM8622L processor multiplier executes the following operations:

- Multiply: The multiply instruction (MUL) multiplies 2 signed or unsigned variables to produce a 64-bit result. The lower 32-bits of the result are stored to a GP register whose location is defined in the instruction. The upper 32-bits are stored to the CP15 RdHi register (CP15-11) using the MCR instruction. This allows the 64-bit result to be stored to 2 nonsequential registers. This instruction is executed only if the condition specified in bits 31:28 of the instruction matches the condition code status.
- Multiply-accumulate: The multiply-accumulate instruction (MLA) multiplies 2 signed or unsigned operands to produce a 64-bit result, which is added to a 3rd operand and written to the destination register. The lower 32-bits of the result are stored to a GP register whose location is defined in the instruction. The upper 32-bits are stored to the CP15 RdHi register (CP15-11) using the MCR instruction. This allows the higher-precision 64-bit result to be stored to 2 nonsequential registers. This instruction is executed only if the condition specified in bits 31:28 of the instruction matches the condition code status.
- Signed multiply long: The signed multiply long instruction (SMULL) multiplies 2 signed variables to produce a 64-bit result. The result is written to 2 sequential GP registers whose location is defined in the instruction. This instruction is executed only if the condition specified in bits 31:28 of the instruction matches the condition code status.
- Signed multiply-accumulate long: The signed multiply-accumulate long instruction (SMLAL) multiplies 2 signed variables to produce a 64-bit result, which is then added to another 64-bit value stored in 2 sequential destination GP registers. This instruction is executed only if the condition specified in bits 31:28 of the instruction matches the condition code status.
- Unsigned multiply long: The unsigned multiply long instruction (UMULL) multiplies 2 unsigned variables to produce a 64-bit result. The result is written to 2 sequential GP registers whose location is defined in the instruction. This instruction is executed only if the condition specified in bits 31:28 of the instruction matches the condition code status.
- Unsigned multiply-accumulate long: The unsigned multiply-accumulate long instruction (UMLAL) multiplies 2 unsigned variables to produce a 64-bit result, which is then added to another 64-bit value stored in 2 sequential destination GP registers. This instruction is executed only if the condition specified in bits 31:28 of the instruction matches the condition code status.

Data Cache

The primary data cache of the EM8622L includes a direct mapped, 16KB data cache. The primary data cache contains a fixed line size of 128-bits and incorporates a 32-bit wide data SRAM, a Tag SRAM, and a 1-bit wide Dirty Bit RAM.

Write Buffer

The EM8622L processor provides a 4-entry write buffer to maximize memory bus bandwidth. This buffer is used by the data cache to store modified lines to be written out to memory, and by the core to store non-cacheable data that has access to the write buffer. The entries in the cache that are marked as ‘writeback’ are written to the write buffer instead of directly to the memory. The writethrough pages are written directly to the memory and do not use the write buffer.

Load Store Unit

The EM8622L processor contains a load/store unit that controls the loading and storing of data between the data cache and the write buffer. During a data cache access, the address is generated by the Load/Store Unit of the core and driven to the cache.

The address and the data paths between the load/store unit and the write buffer are also used for non-cacheable stores that have access to the buffer.

Memory Management

The Memory Management Unit (MMU) provides an interface between the processor core and the caches. The MMU accepts only MCR and MRC instructions from the core. Although the processor executes the CDP, LDC and STC instructions, no coprocessor is implemented to respond to these instructions. Therefore, execution of these 3 instructions result in an Undefined Instruction exception.

The MMU decodes the instruction and manipulates the cache accordingly. This includes updating the contents of the caches, flushing the caches, and locking certain lines within either cache.

PLL/Clock Multiplier

The clock multiplier multiplies the input reference clock by a value of 2 to 16. The RCLK input clock to the processor is multiplied within the PLL to derive the CPU clock. The input clock can be multiplied by the following ratios: 2, 4, 6, 8, 10, 12, 14 and 16.

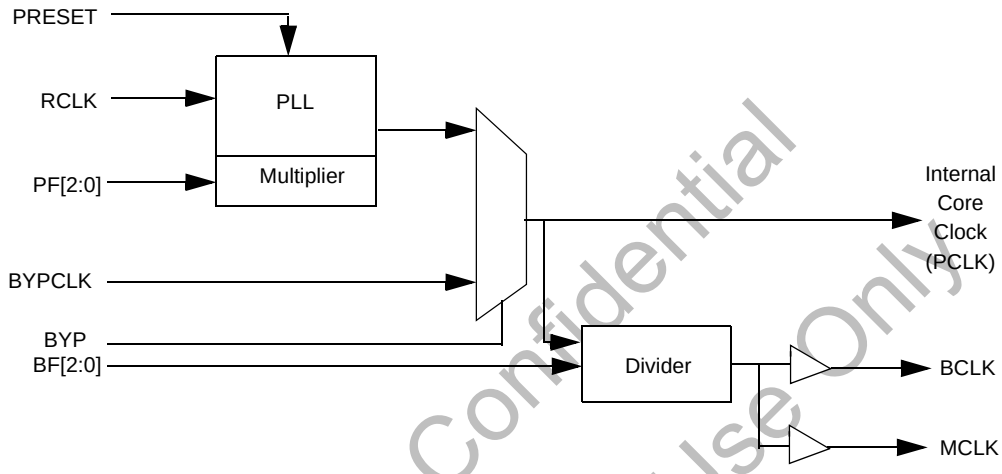


Figure 2-3. **Processor clock generator**

The RCLK input is multiplied by the ratio determined by the PF[2:0] input pins to produce the internal PCLK used by the core. A separate clock divider is used to divide the PCLK signal by the ratio determined by the BF[2:0] pins to produce the clock outputs MCLK and BCLK used to drive the bus.

Because of the skew between the RCLK and the bus clock outputs, all the devices on the system bus must synchronize to either MCLK or BCLK, when the PLL is used as the clock resource.

Alternatively, the BYP input can be asserted to bypass the PLL. In this case, an external clock BYPCLK is used. If the PLL is bypassed, the frequency of the BYPCLK is used to drive the PCLK directly.

Processor Modes of Operation

The processor contains the following modes of operation: User, FIQ, IRQ, Supervisor, Abort and Undefined Instruction.

Table 2-1. **Processor modes of operation**

Name	Operating Mode	Description
User	User	Application program
FIQ	Privileged	Fast interrupt request handler
IRQ	Privileged	Normal interrupt request handler
Supervisor	Privileged	Operating system
Abort	Privileged	Memory manager
Undefined Instruction	Privileged	Emulator for instruction set extensions

There are two basic types of operating modes, user and privileged.

User Mode

The user mode is where the application program and the other user code such as device drivers reside. The processor operates in this mode during normal operations, and only enters one of the privileged modes when an exception or interrupt occurs. The user mode is selected when the M[4:0} field contains a value of 0b10000 or 0b11111.

Privileged Modes

There are 5 types of privileged modes: IRQ mode, FIQ mode, Supervisor mode, Abort mode and Undefined Instruction mode.

IRQ Mode

The IRQ mode is a privileged mode that is entered when an external interrupt is generated on the IRQ pin. The IRQ contains a dedicated Link register (R14_IRQ) that contains the return address, and a Save Processor Status Register (SPSR_IRQ) that contains the processor state at the time the interrupt was taken. Once the interrupt has been serviced, the contents of R14_IRQ are loaded into the program counter (PC), and the contents of the SPSR_IRQ are loaded into the CPSR, allowing the program to resume execution in the mode specified in the SPSR_IRQ.

FIQ Mode

The FIQ mode is a privileged mode that allows for faster interrupt processing than the IRQ mode by providing 5 additional dedicated general purpose registers (R8_FIQ through R12_FIQ) that the interrupt handler can use for temporary storage. The FIQ mode is entered when an external interrupt is generated on the FIQ pin. Like the IRQ mode, the FIQ mode also contains a dedicated Link Register (R14_FIQ) that contains the return address, and a Save Processor Status register (SPSR_FIQ) that contains the processor state at the time the interrupt was taken. Once the interrupt has been serviced, the contents of R14_FIQ are loaded into the program counter (PC), and the contents of SPSR_FIQ are loaded into the CPSR, allowing the program to resume execution in the User mode.

Supervisor Mode

The supervisor mode is a privileged mode entered through the execution of the software Interrupt (SWI) instruction. Certain memory spaces not available in the User mode can be accessed in the Supervisor mode. In addition, many core maintenance functions are performed in the supervisor mode. The Supervisor mode contains a dedicated Link register (R14_SVC) that contains the return address, and a Save Processor Status register that contains the processor state at the time the interrupt was taken. Once the interrupt has been serviced, the contents of the Link register are loaded into the program counter (PC), and the contents of the Save Processor Status register are loaded into the CPSR, allowing the program to resume execution in the User mode.

Abort Mode

The abort mode is a privileged mode that is entered when the processor must abort an operation. There are 4 types of abort operations:

- External instruction abort
- External data abort
- Internal instruction abort
- Internal data abort

An external data or instruction abort is initiated when an external logic asserts the ABORT pin to the processor. An instruction abort occurs when the processor attempts to fetch instruction from an invalid or restricted address. An instruction operation is indicated by the processor driving the nOPC pin low. A data abort occurs when the processor attempts to store or load instructions to or from am invalid or restricted address. A data operation is indicated by the processor driving the nOPC pin high.

The MMU can also perform an integral instruction or data abort by checking the address generated by the processor against its own access permissions.

Internal and external abort operations are logically OR'd within the processor core to provide a single internal abort signal. The MMU compares the 32-bit address with the access permissions. If there is a match between the access permissions provided by the processor, then the MMU generates an internal abort. The internal abort signal is then logically OR'd with the external ABOT pin. If either of these signals is asserted, then an abort signal is sent to the core.

The abort mode contains a dedicated Link register (R14_ABORT) that contains the return address, and a Save Processor Status register that contains the processor state at the time the abort was taken. Once the abort request has been serviced, the contents of R14_ABORT are loaded into the program counter (PC), and the contents of the Save Processor Status register are loaded into the CPSR, allowing the program to resume execution in the User mode.

Undefined Instruction Mode

The undefined instruction mode is a privileged mode that is entered under either of the following 2 conditions:

- When no coprocessor responds to a coprocessor instruction generated by the processor
- When bits 27:25 of the 32-bit instruction contain a value of 0b011, and bit 3 is 0b1, indicating an access to undefined instruction space.

The undefined instruction mode contains a dedicated Link register (R14_UND) that contains the return address, and a Save Processor Status register that contains the processor state at the time the interrupt was taken. Once the abort request has been serviced, the contents of R14_UND are loaded into the program counter (PC), and the contents of the Save Processor Status register are loaded into the CPSR, allowing the program to resume execution in the User mode.

Processor Pipeline

The EM8622L processor core is a high-performance, single-issue RISC architecture that implements a 5-stage pipeline:

1. Fetch stage - Instruction prefetch.
2. Decode stage - Instruction decode and read source registers from multiported register file.
3. Execute stage: Generate memory read address and perform ALU/MAC operation.
4. Memory stage: Read data input bus and ALU/MAC result.

5. Writeback stage: Writeback to register file and load write buffer.

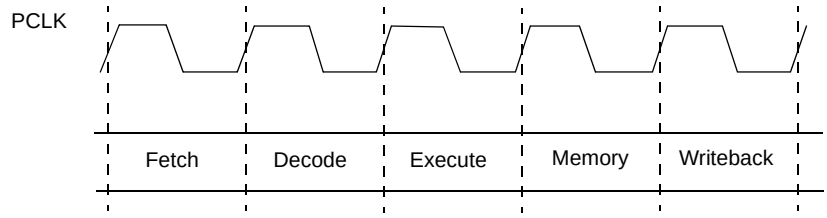


Figure 2-4. **EM8622L processor pipeline block diagram**

Fetch Stage

During the fetch stage the Instruction Fetch Unit retrieves the instruction from the Instruction Cache and passes it to the decoder.

Decode Stage

In the decode stage, 32-bit instructions are decoded and the appropriate internal signals are driven to indicate the type of operation to be performed. If the processor is operating in a 16-bit mode, then a 16-bit instruction is translated into a 32-bit instruction that is decoded by the 32-bit decoder during this stage. The result of the operation is written to the register file.

Execute Stage

In the execute stage the instruction operands are read from the register file and passed to the ALU or 32-bit multiplier depending on the type of the operation. Most ALU operations require only one PCLK cycle to complete. A multiply instruction occupies the Execute stage for 3 PCLK cycles, which stalls the next instruction. The latency for a multiply is constant, and not data-dependent.

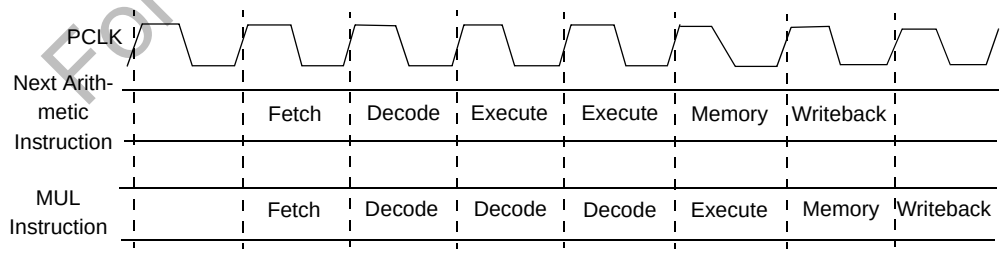


Figure 2-5. **Pipeline usage during a multiply instruction**

The next instruction stalls in the Decode stage for 2 clocks because the MUL instruction is using the Execute stage. Once the multiplication is complete and the MUL instruction propagates to the Memory stage, the next instruction can move to the Execute stage. This diagram assumes that the next instruction is an arithmetic instruction since only one Execute cycle is required.

Memory Stage

In the memory stage the data cache is accessed and store data is written to the buffer in the write back mode. The address and the data paths between the Load/Store Unit and the Write Buffer are used for non-cacheable stores that have access to the buffer.

Writeback Stage

During the writeback stage the data from the load/store unit, the ALU, or the 32-bit multiplier is written back to the register file.

Cycle Timings

The following table provides a summary of the minimum cycle times for the following operations:

Table 2-2. **EM8622L processor cycle times**

Pipeline Operation	Number of PCLK Cycles
Multiply	3
Multiply-Accumulate	3
Load after store	3
Store after load	1
Back-to-back loads	1
Back-to-back stores	1

Local Memory

Introduction

The local RAM is a 2Kx32 memory that supports byte, word and dword accesses. This SRAM can be used to share various data items between the G-Bus masters, as well as to store critical RISC CPU code such as, reset vectors or interrupt service routines (ISRs). The local RAM is accessible from the G-Bus and from the CPU L-Bus.

Features

- 2Kx32 memory
- Supports byte, word and dword accesses
- Supports 8, 16 and 32-bit reads and writes
- Critical RISC CPU code such as, reset vectors or interrupt service routines (ISRs) can be stored.

Functional Description

The internal SRAM consists of 8KB of memory which is accessible by the processor for instruction fetches as well as data reads and writes. This memory can be used by the software as a general data scratchpad, or can be downloaded with the code in order to run performance critical software.

The internal memory consists of a 2048 x 32-bit synchronous SRAM. Accesses to this memory are performed in a single processor clock cycle so that no wait states are required. The internal memory logic supports 8, 16 and 32-bit reads and writes.

The SRAM introduces 1 wait state on the L-Bus on all the read transactions, and one non-dword write transaction. The dword writes have 0 wait state.

Memory Remapping

After power-up, the internal memory is mapped to addresses 60000 to 61FFF. A remap bit is provided in the CPU configuration register to remap the internal memory address space from its startup address mapping to the beginning of ROM space. Remapping the internal memory to ROM space allows the firmware to execute the code from the internal SRAM for faster access and greater flexibility. When the internal SRAM is remapped, the first 8KB of the external ROM is accessible at addresses 0040_0000 to 0040_1FFF.

The registers CPU_RESET_VEC, CPU_UNDEF_VEC, CPU_SWI_VEC, CPU_INSTR_ABORT, CPU_DATA_ABORT, CPU_IRQ_VEC and CPU_FIQ_VEC are the memory locations used to store the reset and exception vectors for the CPU. They should be programmed with the op-code of a jump instruction to the appropriate exception handling routine. At power up, all registers are 0.

Register Map

Local Memory Registers

Table 2-3. **Host CPU block local memory registers**

Address[1]	Register Name	R/W/A[2]	Description
+0000	CPU_RESET_VEC	R/W	CPU Reset Vector Register
+0004	CPU_UNDEF_VEC	R/W	CPU Undefined Vector Register
+0008	CPU_SWI_VEC	R/W	CPU Software Interrupt Vector Register
+000C	CPU_INSTR_ABORT	R/W	CPU Instruction Abort Register
+0010	CPU_DATA_ABORT	R/W	CPU Data Abort Register
+0014	Reserved		
+0018	CPU_IRQ_VEC	R/W	CPU IRQ Vector Register
+001C	CPU_FIQ_VEC	R/W	CPU FIQ Vector Register

1. Address refers to G-Bus byte address relative to the host CPU block base.
2. Read/Write/Auto update.

Interrupt Controller

Introduction

The interrupt controller block is the central location for interrupt handling. It accepts the interrupts generated by the internal blocks and generates the processor interrupts. The interrupt controller block can handle up to 31 hardware and 16 software interrupts.

The interrupts are handled by an IRQ controller and a FIQ controller. Both the controllers are identical. If an interrupt is enabled in both the blocks, then nIRQ and nFIQ will be asserted simultaneously. As nFIQ has a higher priority than nIRQ, the processor will go into a fast interrupt state.

Each interrupt controller can be enabled or disabled individually or globally. A 2-level interrupt priority selection can be implemented.

Features

- Two interrupt controllers, IRQ and FIQ
- Enabling/disabling of individual interrupts
- Global enable/disable
- 2-level interrupt priority selection

Block Diagram

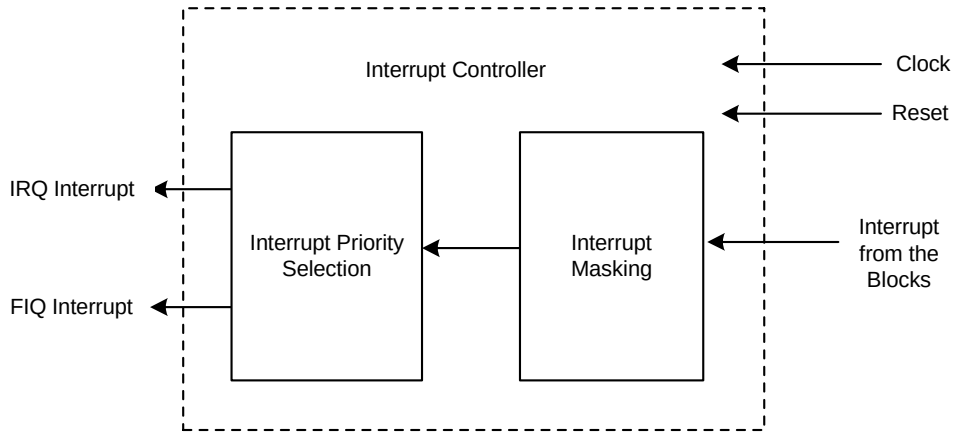


Figure 2-6. **Interrupt controller block diagram**

Functional Description

The interrupt controller allows each interrupt to be asserted as two interrupt levels, IRQ or FIQ. This allows the interrupt to be used for application-specific needs, while still fulfilling the real-time requirements. Each interrupt may be individually masked; alternatively, all interrupts may be masked without altering the individual interrupt masks. Interrupt sources include the following: Timer 1 and 0, UART 1 and 0, PIO 1 and 0, I^2C master, DVD loader, IDE interface, I^2S interface and the MPEG core.

The interrupt control for all the processor interrupts is provided in a central location. Each interrupt may be enabled or disabled individually, or a global enable/disable may be enforced. Interrupts must be cleared at the source once the service request is served. All the blocks with cascaded interrupt control provide an interrupt mask for each interrupt source and each source can be cleared individually. This is illustrated below:

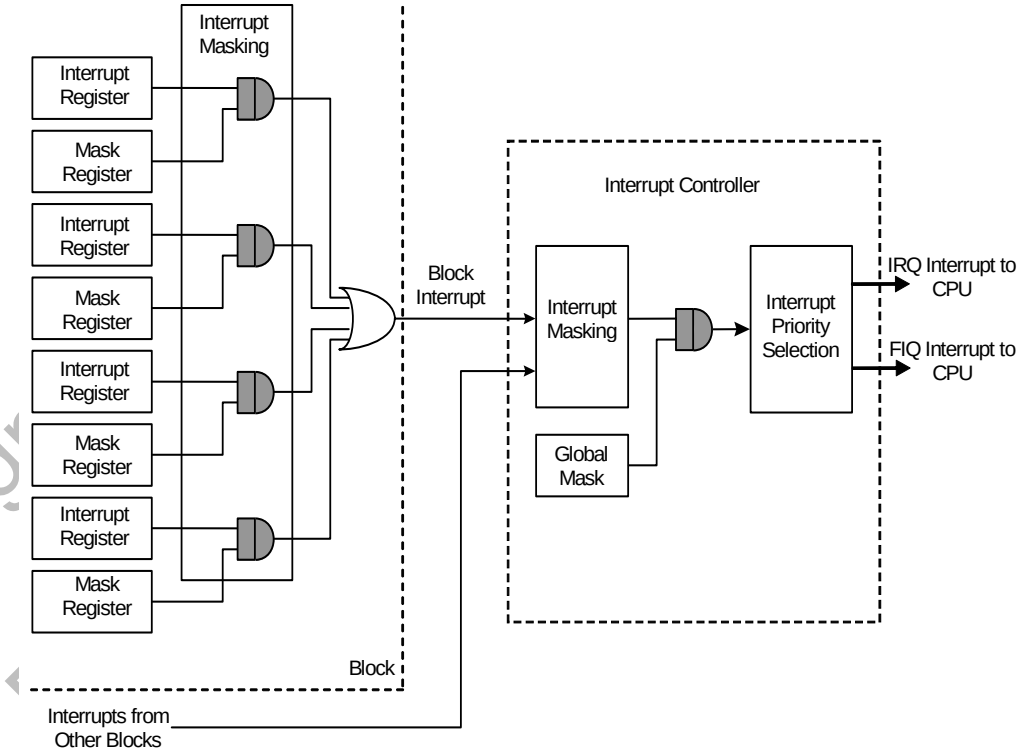


Figure 2-7. **Cascaded interrupt structure**

Interrupt Priority

Each interrupt may be assigned a priority of IRQ or FIQ, allowing the software to customize the priority of each block.

Global Disable

The interrupt controller provides a global disable control bit. This feature can be used to avoid interrupting critical portions of the code before completion. The global disable does not affect the individual interrupt masks. The software need not perform a save and restore. This saves time by reducing the code size and the interrupt latency if an interrupt is asserted when all the other interrupts are disabled.

Control (mask) Blocks

The control (masking) blocks allow the 31 hardware (possibly edge-detected) interrupt lines to be masked, and routed to two CPU interrupt lines. They also contain registers to generate 6 software interrupts.

Register Map - Control Block

Table 2-4. **Control block registers**

Address[1]				
IRQ	**FIQ**	**Register Name[2]**	**R/W/A[3]**	**Description**
+E000	+E100	CPU_XXX_STATUS	R/A	CPU IRQ/FIQ Status Register
+E004	+E104	CPU_XXX_RAWSTAT	R/A	CPU IRQ/FIQ Raw Status Register
+E008	+E108	CPU_XXX_ENABLESET	R/W/A	CPU IRQ/FIQ Enable Set Register
+E00C	+E10C	CPU_XXX_ENABLECLR	R/W	CPU IRQ/FIQ Enable Clear Register
+E010	+E110	CPU_XXX_SOFTSET	R/W	CPU IRQ/FIQ Software Set Register
+E014	+E114	CPU_XXX_SOFTCLR	W	CPU IRQ/FIQ Software Clear Register

1. Address refers to G-Bus byte address relative to the host CPU block base.
2. XXX is IRQ/FIQ depending on the address.
3. Read/Write/Auto update

Processor Timer Interrupt

The CPU has 6 interrupt inputs and an internal timer capable of generating an interrupt. The processor interrupt inputs 0, 1 and 2 are connected to the interrupt control blocks previously described. The processor interrupt inputs 3 and 4 are connected to 0 (inactive) and the processor interrupt input 5 is connected to the processor timer interrupt output.

Edge Detector

The CPU interrupt controller block receives up to 31 hardware interrupt sources. An edge detector receives these 31 lines, and if needed, an edge detection is performed on selected lines. The output of the edge detector consists of 31 ‘latched’ interrupt lines, driven to two identical control (masking) blocks. Each control block can independently mask each of its 31 inputs, plus one software controlled interrupt.

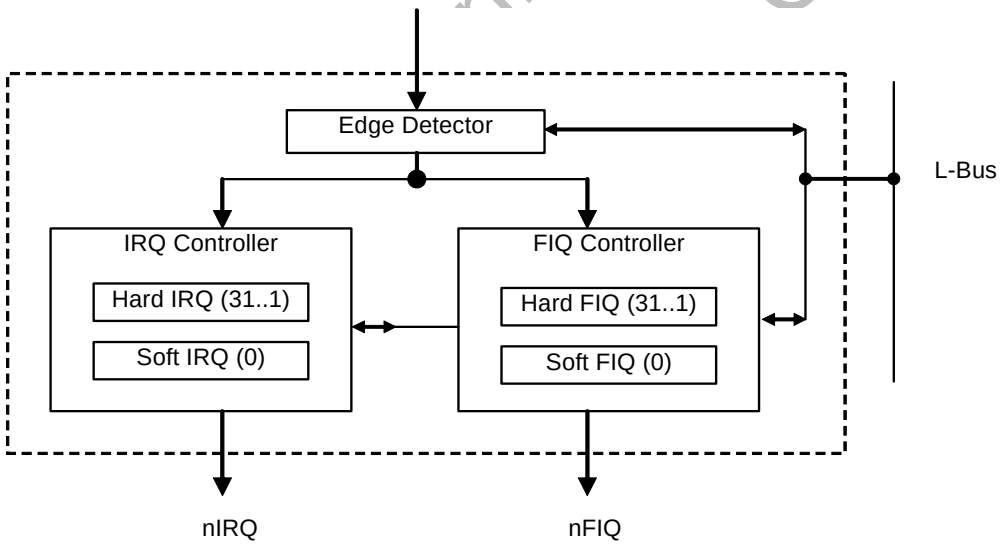


Figure 2-8. **Edge detector**

The 31 interrupts are assigned as shown in the following table:

Table 2-5. **Interrupt sources**

Bit	Source
0	Unusable
1	UART 0
2	UART 1
3	Reserved for keyboard
4	Serial flash
5	Timer 0
6	Timer 1
7	Reserved for Timer 2
8	Real-time clock
9	Host interface channel W0
10	Host interface channel W1
11	Host interface channel R0
12	Host interface channel R1
13	PCI INTA#
14	PCI INTB#
15	PCI INTC#
16	PCI INTD#
17	Reserved
18	Reserved
19	Reserved
20	PCI local bus fault
21	Reserved for external interrupt (GPIO pin)
22	I^2C
23	Graphics accelerator
24	VSYNC 0 (Composite analog output)
25	VSYNC 1 (Composite analog output)
26	VSYNC 2 (MAin analog output)
27	VSYNC 3 (Digital output)
28	VSYNC 4 (Gfxin-vsync: start of v-blanking)

Table 2-5. **Interrupt sources (Continued)**

Bit	Source
29	VSYNC 5 (Gfxin-vsync: end of v-blanking)
30	VSYNC 6 (Vidin-vsync: start of v-blanking)
31	VSYNC 7 (Vidin-vsync: end of v-blanking)

For each of the 31 interrupt lines, two bits called rise and fall select the operating mode according to the following table:

Table 2-6. **Operating mode selection for interrupt lines**

Ride	Fall	Operating Mode
0	0	Level sensitive (active high)
1	1	Level sensitive (active low = inverted)
1	0	Rising edge sensitive
0	1	Falling edge sensitive

Two configuration registers can be read or written, to set or verify the operating mode of all the 31 lines. In order to set/clear the operating mode of certain lines (without changing the others), a read-modify-write operation can be performed on these registers.

Reading the status register returns the individual interrupt lines values at the output of the edge detector. Reading the edge raw status register returns the individual interrupt lines values at the input of the edge detector. For interrupts lines that are in ‘level sensitive, active high’ mode, the two status registers will return the same value.

Writing the edge raw status register will clear the capture register (for each bit written as 1, of the corresponding interrupt). This should be done before exiting an interrupt service routine to avoid re-entering, only on edge sensitive interrupts.

Register Map - Edge Detector

Table 2-7. **Edge detector registers**

Address[1]	Register Name	R/W/A[2]	Description
+E200	CPU_EDGE_STATUS	R/W	CPU Edge Detector Status Register
+E204	CPU_EDGE_RAWSTAT	R/W/A	CPU Edge Detector Raw Status Register
+E208	CPU_EDGE_CONFIG_RISE	R/W	CPU Edge Detector Configuration Rise Register
+E20C	CPU_EDGE_CONFIG_FALL	R/W	CPU Edge Detector Configuration Fall Register
+E210	CPU_EDGE_CONFIG_RISE_SET	R/W	CPU Edge Detector Configuration Rise Set Register
+E214	CPU_EDGE_CONFIG_RISE_CLR	R/W	CPU Edge Detector Configuration Rise Clear Register
+E218	CPU_EDGE_CONFIG_FALL_SET	R/W	CPU Edge Detector Configuration Fall Set Register
+E21C	CPU_EDGE_CONFIG_FALL_CLR	R/W	CPU Edge Detector Configuration Fall Clear Register

1. Address refers to G-Bus byte address relative to the host CPU block base.
2. Read/Write/Auto update.

Timers

Introduction

The EM8622L has a timer block implemented in the host CPU block. The timer block contains two independent timers, Timer 0 and Timer 1, each with two modes of operation periodic or free-running. They are identical except that, Timer 0 is driven by the system clock, while Timer 1 receives the external 27MHz clock source. Each timer is a 16-bit counter which decrements on each input clock (an optional divide-by-16 or divide-by-256 pre-scaler is supported). When the counter reaches zero, an interrupt is generated and the initial count value is automatically reloaded.

Features

- Two timers
- Supports independent clock pre-scale for each timer
- Independent interrupt for each timer
- Supports two modes of operation, periodic and free-running

Block Diagram

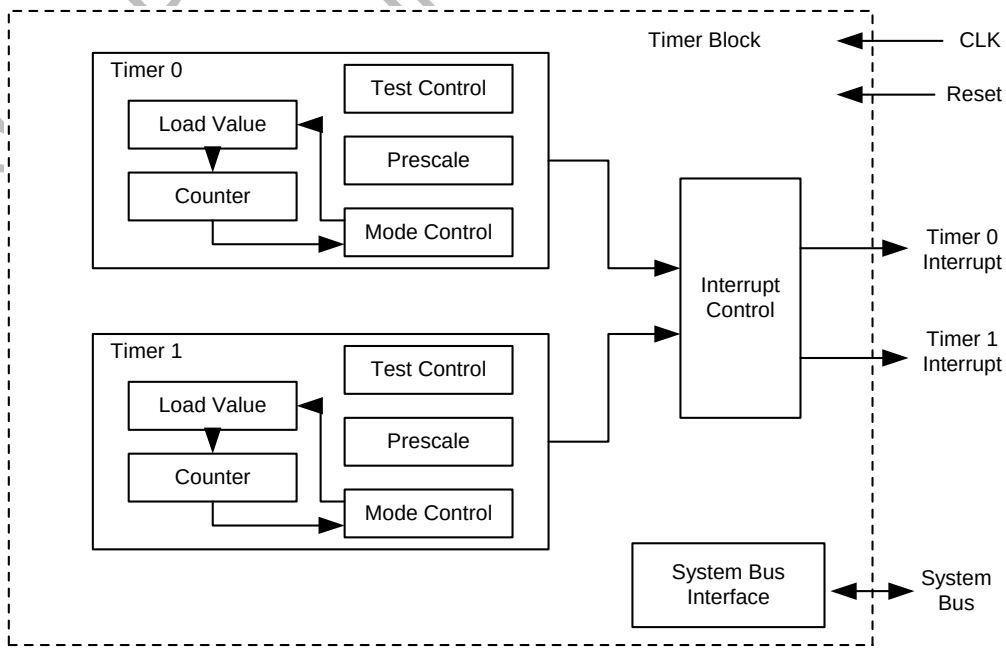


Figure 2-9. **Timer block diagram (system overview)**

Functional Description

Timers are primarily used to accurately track long periods of time (such as during time-outs and internal performance monitoring), freeing the processor for more important tasks. Two timers provide great flexibility, allowing small delay times to be programmed in one timer while the second is used for time-out functions, or to allow multiple concurrent tasks access to independent timers.

The timer block includes two timers with scalable processor ticks. They count down from a user-loaded value, which must be reloaded to avoid a time-out. Each timer has an independent clock pre-scale function and an independent interrupt. They can be set to a periodic or a free-running mode.

The system bus interface block interfaces the timer registers to the processor through the system bus.

The timer is implemented as a 16-bit down counter. The timer counts down from the count value programmed in the timer load value register; on reaching zero an interrupt is generated. Depending on the mode, the timer then reloads the load value, continues counting or stops. The timer load value register can be changed at any time; the timer will begin counting from the new value written to the this register.

The timer load value registers hold the initial count value that is reloaded each time the counter reaches 0, when operating in the periodic mode. The value in the timer counter value registers decrement on each clock cycle until they reach 0. They are then automatically reloaded with the values in the timer load value registers, and the interrupt bit INT 5/6 is set. The values can be overwritten at any time, whatever the mode.

The timer control registers allow the selection between the periodic mode or the free-running mode. Pre-scale sets the clock divisor to 1 (00), 16 (01) or 256 (10).

Timer Control

The timer powers up disabled. It can be enabled or disabled using the timer control register. When the timer is disabled, it stops counting and retains its current value. When it is enabled again, it resumes counting from its current value. The timer can be reset at any time to the value in the timer load value register using the timer control register. The counter value immediately loads, irrespective of the timer being enabled.

The interrupts from the timer can be cleared through the timer clear register. Reading the interrupt bit will return the current status of the interrupts. Writing a ‘1’ to the corresponding interrupt bits clears the interrupts.

Load Value/Counter

A counter operating off of the system clock drives the timer. The load value register programmed by the firmware controls its period. This register contains the load value for the timer. In all the modes except the free-running mode, this value is loaded into the timer counter before counting down. It may be updated at any time; the new value will be written to the counter immediately. Writing load value of 0 will disable the timer except in the free-running mode; in free-running mode, this register value is ignored.

Mode Control

Two modes of operation is available for the timers, namely, periodic and free-running. The timer control register controls the timer reloading and disabling.

Periodic

In the periodic mode, the timer counts down to zero from the load value. An interrupt is generated when the count is zero. After reaching zero, the load value is reloaded into the timer and the timer counts down again. A load value of zero disables the timer.

Free-running

In the free-running mode, the timer counts down to zero from FFFFh. An interrupt is generated when the count is zero. After reaching zero, FFFFh is reloaded into the timer. The load register is ignored in this mode. This mode is identical to the periodic mode with a load value of 65535. When the timer is first enabled, it begins counting down from its current value, and not from FFFFh.

Interrupt Control

Each timer may generate an interrupt when it times out. The interrupts from the timers can be cleared through the timer clear registers. Reading the interrupt bit will return the current status of the interrupts. Writing a ‘1’ to the corresponding interrupt bits clears the interrupts.

Clock Pre-scale

A clock pre-scale is provided for the timer clock. When used, the pre-scale divides the system clock by powers of two, from 2^2 to 2^{16} in order to achieve higher resolution or longer timer periods as defined below:

Table 2-8. **Timer clock pre-scale**

Value[1]	Timer clock frequency
0	System clock
1	System clock / 4
2	System clock / 8
3 . .	System clock / 16 . .
14	System clock / 32,768
15	System clock / 65,536

1. The pre-scale value should not be changed unless the timer is disabled.

Register Map

Timer Registers

Table 2-9. **Timer registers**

Address[1]	Register Name	R/W/A[2]	Description
+C500	CPU_TIMER0_LOAD	R/W	Timer 0 Load Value Register
+C504	CPU_TIMER0_VALUE	R/W/A	Timer 0 Counter Value Register
+C508	CPU_TIMER0_CTRL	R/W	Timer 0 Control Register
+C50C	CPU_TIMER0_CLR	W	Timer 0 Clear Register
+C600	CPU_TIMER1_LOAD	R/W	Timer 1 Load Value Register
+C604	CPU_TIMER1_VALUE	R/W/A	Timer 1 Counter Value Register
+C608	CPU_TIMER1_CTRL	R/W	Timer 1 Control Register
+C60C	CPU_TIMER1_CLR	W	Timer 1 Clear Register

1. Address refers to G-Bus byte address relative to the host CPU block base.
2. Read/Write/Auto update.

UARTs

Introduction

The EM8622L contains two instances of Universal Asynchronous Receiver-Transmitter (UART), UART 0 and UART 1, for supporting asynchronous serial communications.

The two UARTs, UART0 and UART1 are functionally equivalent to the industry-standard 16550, and differ only in a few minor configuration/control register definitions (listed in the Functional Description section below). Each UART includes an independent baud rate generator. Baud rates supported are dependent on the frequency of operation. The baud rate generator may either use the system clock (200MHz typically) or the 27MHz external clock as its reference.

The UARTs also provide debugging with full modem support that allows simultaneous connection to remote systems.

Features

- Supports modem communication support
- Supports the 'FIFO mode' in which the transmitter and receiver are each buffered with 16 byte FIFO's to reduce the number of interrupts presented to the CPU
- Holding and shift registers eliminate the need for precise synchronization between the CPU and the serial data
- Supports independently controlled transmit, receive, line status and data set interrupts
- Supports 300 - 115.2kbps baud rate (higher rate will depend upon the clock frequency). The programmable baud generator divides any input clock by 1 to (2^{16} - 1) and generates the 16 x clock.
- Supports fully programmable serial-interface with the following characteristics:
  - 5, 6, 7 or 8-bit characters
  - Even, odd, stick or no-parity bit generation and detection
  - 1, 1 ½ or 2-stop bit generation
- Supports false start bit detection
- Supports complete status reporting capabilities
- Supports line break generation and detection

- Supports the following internal diagnostic capabilities:
  - Loopback controls for communications link fault isolation
  - Break, parity, overrun, framing error simulation
- Supports full prioritized interrupt system controls
- Supports receive buffer interrupts for empty, half-full and byte-received
- Supports transmit buffer interrupt for empty

Block Diagram

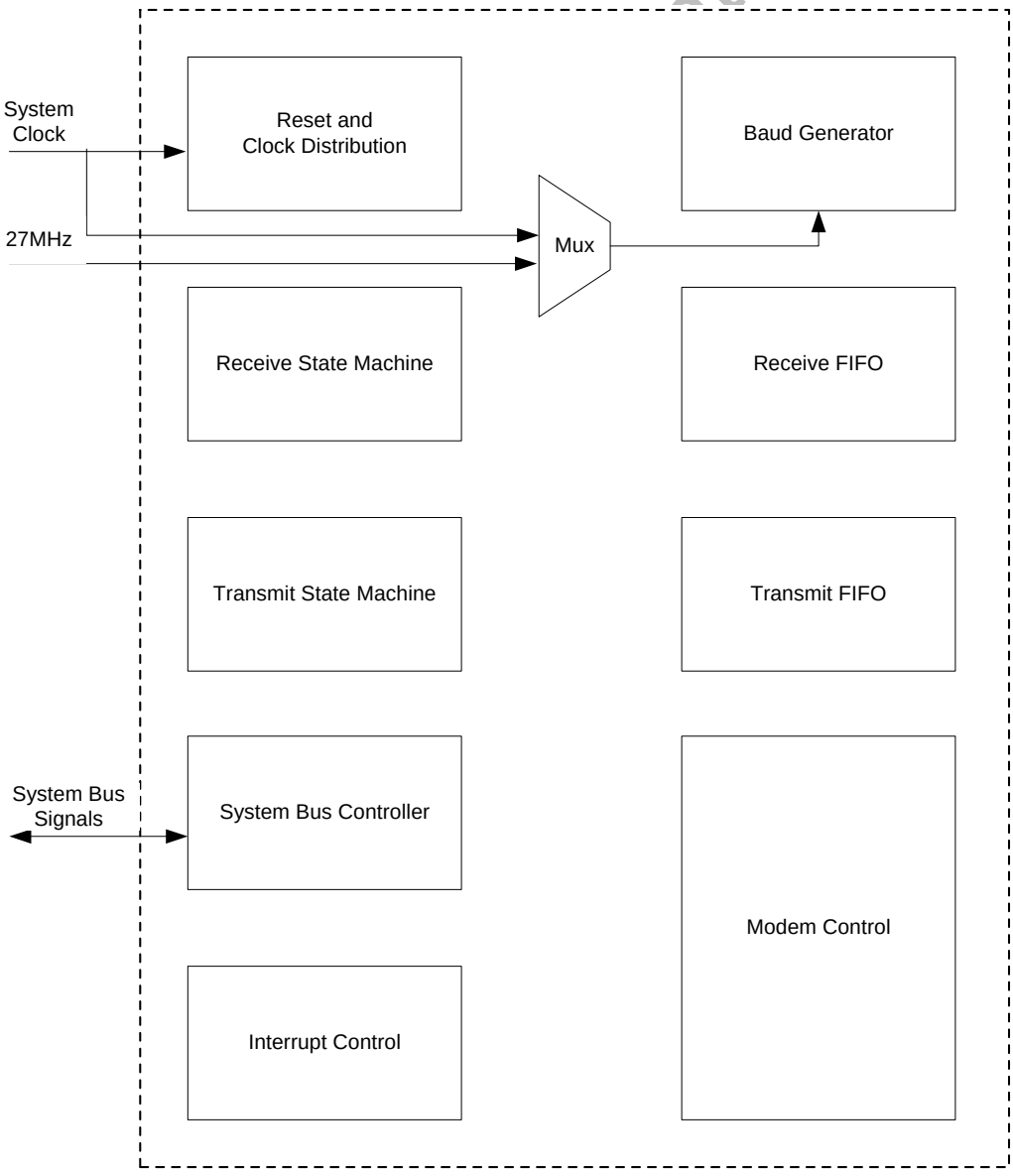


Figure 2-10. **UART block diagram**

Functional Description

The Universal Asynchronous Receiver/Transmitter is a peripheral device providing serial communications capabilities. It converts the internal data from parallel to serial format (or vice versa) for bidirectional transmission down a single cable. The device receives a character, generates an interrupt, and stores it in a buffer until the next character is ready. The CPU must fetch the information and clear the interrupt flag before the next character is received.

The two UARTs, UART 0 and UART 1 in the EM8622L are almost compatible with the industry standard 16550. The differences between the EM8622L UARTs and the 16550 are listed below:

- Registers are 4-byte apart in the EM8622L UART versus the 1-byte in the 16550
- The UART in the EM8622L is connected to INT 1
- CPU_UART_RXD and CPU_UART_TXD registers are at two different addresses
- The 16-bit register CPU_UART_CLKDIV is the concatenation of the Divisor Latch (MS) and Divisor Latch (LS) registers on the 16550
- The register CPU_UART_CLKSEL either selects the system clock (typically 200MHz), or the external clock (27MHz) as the clock source for the baud rate generator

The UART block provides two independent UARTs, UART 0 and UART 1 for serial communication and debugging. The UART includes full modem support, allowing simultaneous connections to remote systems. This UART is compatible with generic UART devices used on PCs and other systems. Proper start, parity, and stop bits are appended to characters transmitted on the TXD output pin. Similarly, the characters received at the RXD input pin are stripped of the extra bits enveloping them. Receiver and transmitter logic runs on the clock derived from the main clock input (divided by 16) divided by the value in the clock divider control register.

The UART performs serial-to-parallel conversion on data characters received from an external device or a modem, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read the complete status of the UART at any time during the functional operation. Status information reported includes the type and condition of the transfer operations being performed by the UART, as well as any error conditions (parity, overrun, framing or break interrupt).

The UART supports serial data formats consisting of 5-8 data bits, 1-2 stop bits and even/odd/stick/no parity. The baud rate generator uses a selectable clock source, and the UART blocks support baud rates from 300 to 115.2kbps. Its 16-byte transmit buffer interrupts for empty data, and the 16-byte receiver interrupts for empty, half-full, and byte-received data. The UART also detects false start bits, breaks and supports modem communications.

The UART can be put into an alternate mode (FIFO mode) to relieve the CPU of excessive software overhead. In this mode internal FIFOs are activated allowing 16-byte (plus 3-bit of error data per byte in the RCVR FIFO) to be stored in both the receive and the transmit modes.

UART Data Formats

The UART includes a programmable baud rate generator that is capable of dividing the timing reference clock input by divisors of 2 to 65535 and by 16. The UART has complete modem-control capability, and a processor-interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link.

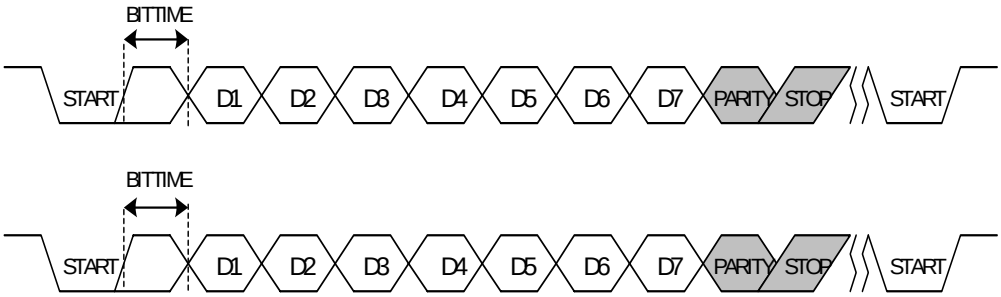
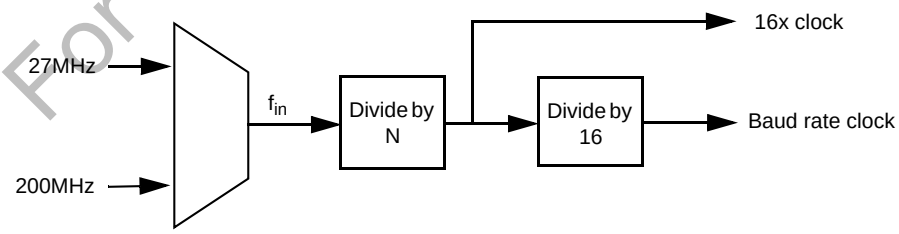


Figure 2-11. **UART data formats**



Baud rate ($f_{baud\ rate}$) = Input clock frequency (f_{in}) / (16xN) = bits/sec, where N = 1 to 2^{16}-1

Bit time = 1 / Baud rate = sec/bit

Start bit

TXD and RXD are normally high. To transmit a character drive the TXD line low for one bit time. The receiver always samples the RXD line; when it detects a start bit it starts shifting a new character in.

Data

A character can be programmed for 5-8 data bits. Both the receiving and transmitting UARTs should be programmed for the same settings or communication fails.

Parity

The parity generation and checking can be enabled or disabled. If the parity is disabled, then no parity bit is transmitted and the receiver will not expect a parity bit. If the parity is enabled, then it can be an even, odd or a stick parity.

- Even parity: The parity bit is 1 if the character has an odd number of 1's
- Odd parity: The parity bit is 1 if the character has an even number of 1's
- Stick parity: The parity bit can be forced to be either '1' or '0'

Stop Bit

The stop bits ('1') are the last bits to be transmitted/received for each character. The number of stop bits can be programmed to 1, 1½ or 2-bit times. The stop bits act as spacers between characters when transmitted back to back. Both the receiving and the transmitting UARTs need to be programmed for the same settings. The communication may fail if the number of stop bits expected by the receiver is greater than the number of stop bits actually received.

Break

A break is detected if the RXD line is held low longer than character time, which is the time taken to transmit or receive a character including start, parity and stop bits. This usually happens if the RXD line is disconnected or if the transmitting UART forces a break or is turned off. To force a break, the break bit in the line control register needs to be set. An interrupt is generated if a break is detected.

Modes

While only a few modes are standard, nearly limitless combinations are possible. Any of the following variables can be combined to create distinct modes: baud rate, FIFO/non-FIFO, data bits, stop bits and parity.

Interrupts

UART operation and line speed are controlled by the UART line control register and a few other registers. The UARTS generate data ready (DR) or character time-out, buffer empty (THRE), line status (OE, PE, FE, AND BI), and modem status (DCTS, DDSR, RI, and DCD) interrupts.

The data ready interrupt (DR) will be asserted when the receiver buffer depth is equal to the number of characters programmed in the trigger register.

The THRE interrupt will be asserted if the transmitter buffer depth is less than or equal to the number of characters programmed in the trigger register.

The line status interrupt is asserted when a receive overrun occurs (OE) or if the received parity is different from the expected value (PE) or, if a valid stop bit is not detected (FE) or, if a break is received when the RXD signal is at a low state for more than one character transmission time from start bit to stop bit (BI).

the modem status interrupt bit dcts, is asserted when the ctsn (Clear to Send) pin changes, or when the DSRN (data set ready) pin changes or, when the RIN (ring indicator) pin is at a low value or, when the DCDN (data carrier detect) pin is at a low value.

If more than one category of interrupt is asserted, only the highest priority ID will be given. The line and modem status interrupts are cleared by reading the corresponding status register (LSR, MSR). The receiver data available interrupt will be cleared when all the data is read from the receiver buffer or the FIFO. The transmitter buffer empty will be cleared when the data is written to the TXD register, or if the IIR register is read and the THRE interrupt ID is set. The transmit and receive FIFOs have the effective depth of sixteen characters when enabled and a depth of one character when disabled.

GPIO Function

There are seven UART pins which can individually be used as GPIOs if the UART functionality is not required. The following table shows the relationship between the UART signals and the GPIOs:

DTR	RTS	TXD	DCD	DSR	CTS	RXD
GPIO6	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	GPIO0

CPU_UART_GPIO_MODE: This register selects the UART mode/GPIO mode (0 = UART, 1 = GPIO) for each pin.

Common values are:

- 7F00 = All pins are in the UART mode (power-up value)
- 7F7F = All pins are in the GPIO mode (UART function is disabled)
- 7F11 = TXD-RXD are in the UART mode (allowing a 2 wire serial protocol), and the other pins are in the GPIO mode.

The CPU_UART_GPIO_MODE register must be written at address xx38, but read at xx3C. The CPU_UART_GPIO_DIR register determines the direction of the GPIOs (0 = input, 1 = output). This affects the pins in the GPIO mode only. The CPU_UART_GPIO_DATA register gives the value of the GPIO pin.

Register Map

UART Registers

Table 2-10. **UART registers**

Address[1]	Register Name	R/W/A[2]	Description
+C100/C200	CPU_UART_RXD	R/A	Receive Buffer Register
+C104/C204	CPU_UART_TXD	W	Transmit Buffer Register
+C108/C208	CPU_UART_INTEN	R/W	Interrupt Enable Register
+C10C/C20C	CPU_UART_INTID	R/W	Interrupt Identification Register
+C110/C210	CPU_UART_FIFOCTL	R/W	FIFO Control Register
+C114/C214	CPU_UART_LINECTL	R/W	Line Control Register
+C118/C218	CPU_UART_MODEMCTL	R/W	Modem Control Register
+C11C/C21C	CPU_UART_LINESTAT	R/W	Line Status Register
+C120/C220	CPU_UART_MODEMSTAT	R/W	Modem Status Register
+C124/C224	CPU_UART_SCRATCH	R/W	Scratch Register
+C128/C228	CPU_UART_CLKDIV	R/W	UART Clock Divider Control Register
+C12C/C22C	CPU_UART_CLKSEL	R/W	UART Clock Select Control Register
+C130/C230	CPU_UART_GPIO_DIR	R/W	GPIO Direction Register
+C134/C234	CPU_UART_GPIO_DATA	R/W	GPIO Data Register
+C138/C238	CPU_UART_GPIO_MODE	R/W	GPIO Mode Register

1. Address refers to G-Bus byte address relative to the host CPU block base.
2. Read/Write/Auto update.

Pin Description

UART Pins

Table 2-11. **UART pin descriptions**

Pin Name	Ball ID	Direction	Description
UART0_TX	A10	B	UART 0 transmit data output
UART0_RX	D12	B	UART 0 receive data input
UART0_RTS	B10	B	UART 0 request to send. Flow control signal.
UART0_CTS	C12	B	UART 0 clear to send. Flow control signal.
UART0_DCD	B11	B	UART 0 data carrier detect. Data set status signal.
UART0_DTR	E11	B	UART 0 data terminal ready. Data terminal status signal.
UART0_DSR	A11	B	UART 0 data set ready. Data set status signal.
UART1_TX	D7	B	UART 1 transmit data output
UART1_RX	A9	B	UART 1 receive data input
UART1_RTS	E8	B	UART 1 request to send. Flow control signal.
UART1_CTS	B9	B	UART 1 clear to send. Flow control signal.
UART1_DCD	E9	B	UART 1 data carrier detect. Data set status signal.
UART1_DTR	E7	B	UART 1 data terminal ready. Data terminal status signal.
UART1_DSR	B8	B	UART 1 data set ready. Data set status signal.

Electrical Characteristics

UART DC characteristics

Table 2-12. **UART DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}(1)	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
V_{IH}[2]	Input high voltage	V	2.0		5.5
V_{IL}(2)	Input low voltage	V	-0.3		0.8

1. Parameter applies to all pins configured as outputs.
2. Parameter applies to all pins configured as inputs.

Timing Diagrams

Baud rate ($f_{baud\ rate}$) = Input clock frequency (f_{in}) / (16xN) = bits/sec, where N = 1 to 2^{16}-1

Bit time = 1/Baud rate = sec/bit

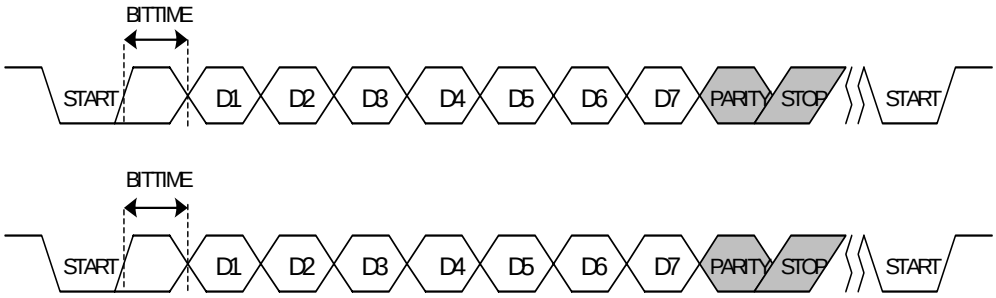


Figure 2-12. **UART timing diagram**

3 DRAM Controller

Block Diagram of DRAM Controller

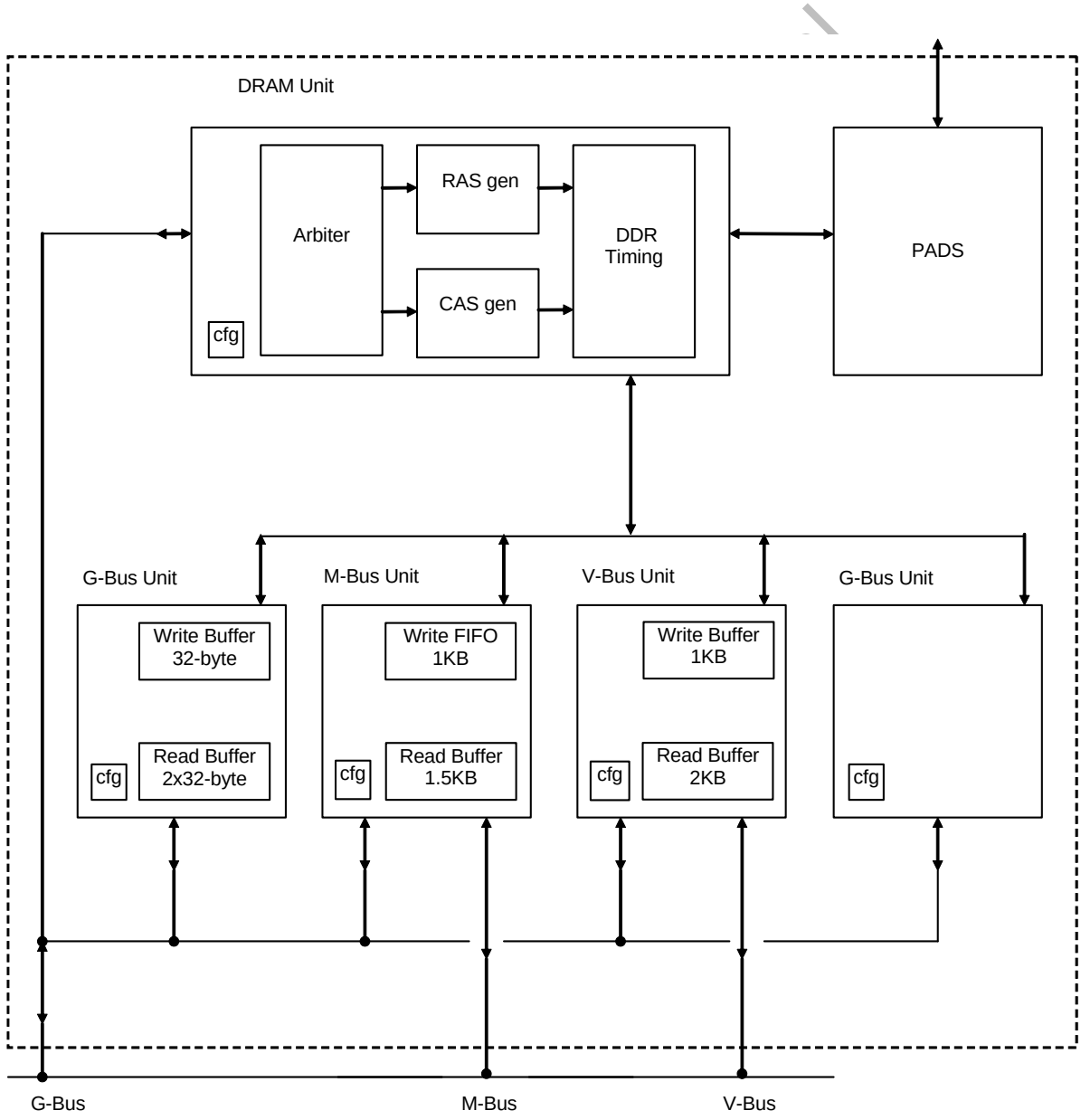


Figure 3-1. **DRAM controller block diagram**

Introduction

The EM8622L contains double-data rate synchronous DRAM (DDR-SDRAM) controller operating at 200MHz. Because the DDR technology transfers data on both edges of the clock, the effective burst data bandwidth of the controller is 1.6GB/sec when using the full 32-bit interface. Thus, the DRAM controller can interface up to 128MB of external DDR SDRAM, with a burst data bandwidth of 1.6GB/sec. No more than 80% of the memory bandwidth should be used (~1.2GB/sec for 32-bit interface). The contents of the DRAM memory may also be encrypted (proprietary algorithm) to protect the contents.

Compatible external memory devices attached to the memory interface provide the data storage capability necessary for all the functions performed by the EM8622L. These functions include:

- Multiple audio, video and data buffers
- Temporary data storage for hardware function blocks
- Program and data memory for the on-chip CPUs, DSPs and RISC processors
- Transport stream buffers

The DRAM controller supports memory devices which conform to JEDEC standard JESD79. The electrical interface to the external memories supports JEDEC SSTL_2 compliant signaling.

Features

- Double-data rate synchronous DRAM (DDR-400)
- 16 or 32-bit DRAM interface
  - 16-bit: SDTV applications
  - 32-bit: HDTV applications (minimal graphics)
- Effective burst data bandwidth of the controller is up to 1.6GB/sec
- Interfaces up to 128MB of external DDR SDRAM
- Supports memory devices which conform to JEDEC standard JESD79
- Electrical interface to the external memories supports JEDEC SSTL_2 compliant signaling

Functional Description

The EM8622L DRAM controller can interface to up to 128MB of external DDR SDRAM, using a 16 or 32-bit wide data bus. The DRAM controller interfaces with the different modules of the EM8622L through,

- A G-Bus for flat addressing of the DRAM, and configuration registers.
- An M-Bus for DMA transfers to/from all the blocks except the video output block
- A V-Bus for DMA transfers to/from the video output block
- A direct load port to interface to the video decoder for motion compensation loads

As shown in the block diagram, the memory controller connects to all four of the primary device buses. Each bus connects to the controller via an interface unit, which contains command queues as well as FIFOs for read and/or write data. The interface units in turn connect to the DRAM unit which actually controls the external memory.

Any G-Bus master can program the configuration registers of all four units. Also, a G-Bus master can access the entire DRAM space via the G-Bus unit.

The DRAM controller occupies two sections on the G-Bus space: A 256MB section for direct memory accesses and a 64KB section for configuration register accesses.

Within the DRAM unit, an arbiter analyzes the pending requests from the many DMA channels in the various functional units. Based on the channel priority, the bank access requested and the currently available banks, the arbiter determines order of the service requested by the various channels. The DRAM unit is designed to maximize utilization of the DRAM, while simultaneously controlling read and write access latency.

The EM8622L device uses a sophisticated delay-lock loop (DLL) technique to optimize critical timings in the DRAM interface. These timings are automatically updated on a periodic basis to maintain the critical characteristics under changing voltage and temperature conditions, and are not accessible to the customers.

Memory Configurations

The 32-bit DRAM controller interface can connect to either 32, 64 or 128MB of DDR SDRAM using a 16 or 32-bit wide data bus. Only a single ‘bank’ of SDRAM is supported, i.e. memory ‘depth’ expansion is not possible.

Table 3-1. **Supported memory configurations**

Total memory (MB)	Bus Width	Memory chip configuration	Rows	Columns
32	16	One 16Mx16	8192	512
64	16	One 32Mx16	8192	512
64	32	Two 16Mx16	8192	512
128	32	Two 32Mx16	8192	1024

Many factors affect the external memory configuration needed to support a particular application. In addition to the basic requirement of picture buffer areas needed to support the video decoding process and any necessary graphics planes, other storage requirements must also be considered.

Additional factors which come into play may include:

- Font sets and graphics elements which must be maintained in the off-screen display memory
- Multiple graphics buffers required for ‘instant’ updates (animation effects)
- Use of video and graphics input ports
- Nature of processing tasks being performed on the GP RISC CPU

Determining the optimum memory configuration requires that the application requirements be well defined. More demanding applications may require an analysis of the memory bandwidth utilization.

The DRAM registers control the DRAM refresh timing, CAS latency, and DRAM organization (total size, row/column size and the number [1x32 or 2x16] of DDR chips used). They also control the programmable delay lines affecting the data output delay (write transactions) and the input delay (read transactions). The registers allow the monitoring of the DRAM bandwidth usage.

Memory Connection Diagrams

This section provides the connection diagrams for a variety of supported external memory configurations. These diagrams show basic signal connections only; they do not address detailed electrical issues such as termination networks.

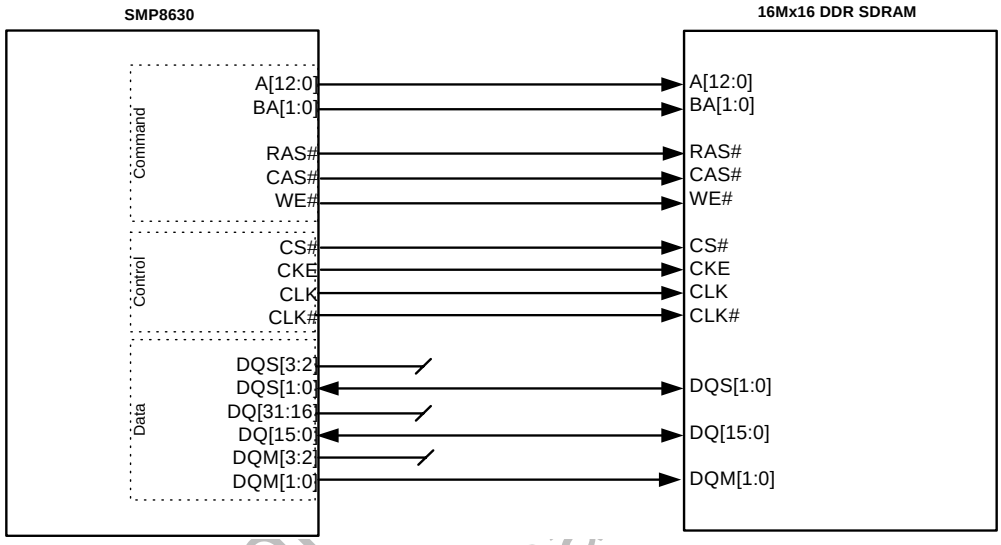


Figure 3-2. **Case 1: 32MB memory; 16-bit bus width**

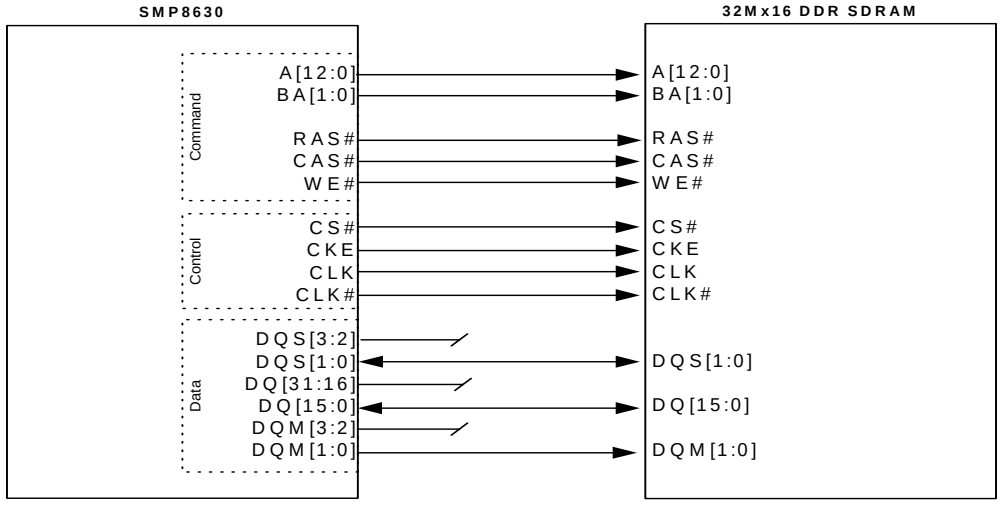


Figure 3-3. **Case 2: 64MB memory; 16-bit bus width**

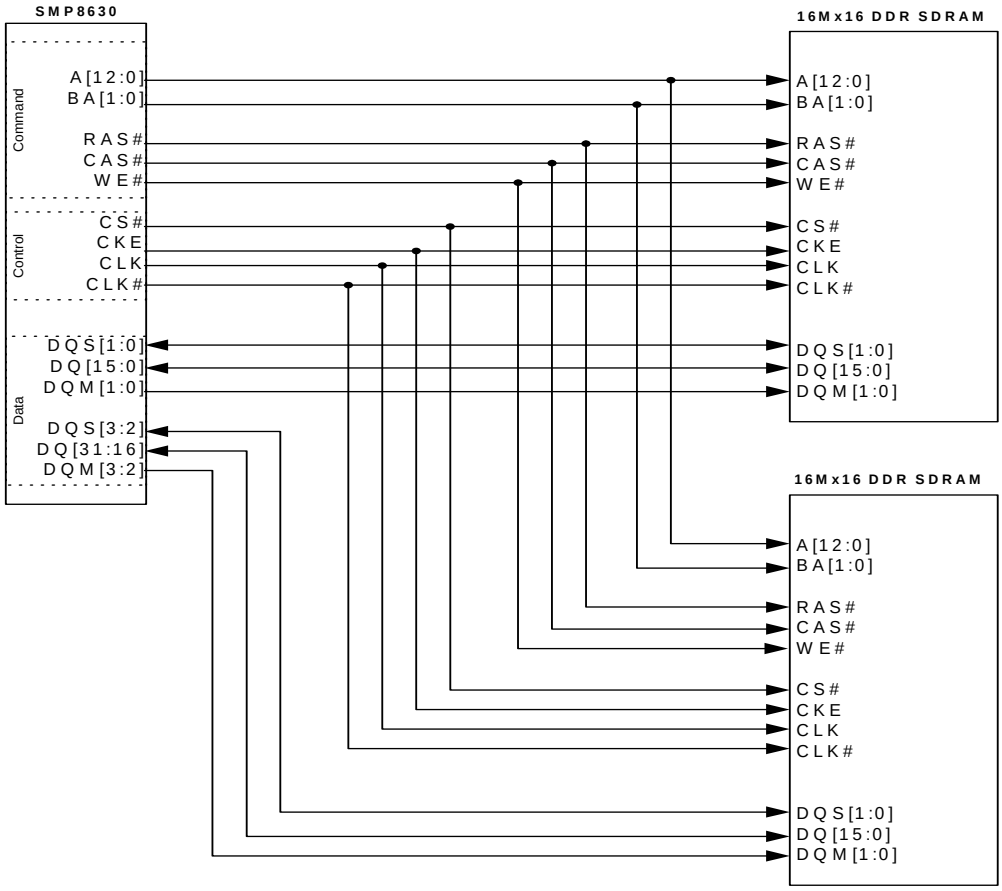


Figure 3-4. **Case 3: 64MB memory; 32-bit bus width**

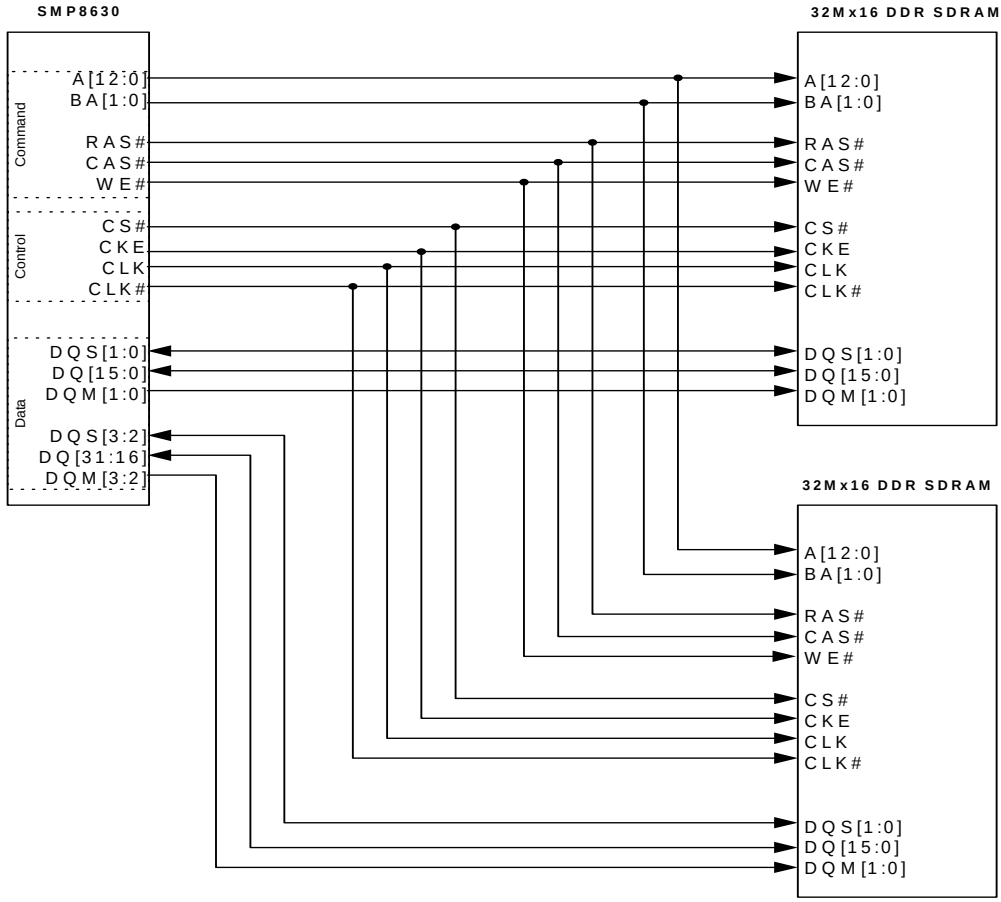


Figure 3-5. **Case 4: 128MB memory; 32-bit bus width**

Memory Bandwidth Considerations

For complex applications, the available memory bandwidth may limit what can be done with EM8622L. In these cases the total available memory bandwidth (assuming DDR-400 DRAM) is as follows:

- 32-bit interface, maximum memory bandwidth is 1.6GB/s
- Should not exceed 80% of maximum, resulting in 1.3GB/s available memory bandwidth

Table 3-2. **Memory bandwidth requirements**

Function	Format	Memory bandwidth needed
MPEG-4.10 (H.264) video decode and display	1920 x 1080 x 30 1280 x 720 x 60 720 x 480 x 60	1.1GB/s max 1.0GB/s max 0.3GB/s max
VC-1 video decode and display	1920 x 1080 x 30 1280 x 720 x 60 720 x 480 x 60	TBD
MPEG-2 video decode and display	1920 x 1080 x 30 1280 x 720 x 60 720 x 480 x 60	TBD
Displaying 32-bit data graphics,	1920 x 1080 x 30 x 32/8 1280 x 720 x 60 x 32/8 720 x 480 x 60 x 32/8 720 x 480 x 30 x 32/8	0.25GB/s 0.22GB/s 0.08GB/s 0.04GB/s
Displaying 24-bit data graphics	1920 x 1080 x 30 x 24/8 1280 x 720 x 60 x 24/8 720 x 480 x 60 x 24/8 720 x 480 x 30 x 24/8	0.19GB/s 0.17GB/s 0.06GB/s 0.03GB/s
Displaying 16-bit data graphics	1920 x 1080 x 30 x 16/8 1280 x 720 x 60 x 16/8 720 x 480 x 60 x 16/8 720 x 480 x 30 x 16/8	0.12GB/s 0.11GB/s 0.04GB/s 0.02GB/s
Displaying 8-bit data graphics	1920 x 1080 x 30 x 8/8 1280 x 720 x 60 x 8/8 720 x 480 x 60 x 8/8 720 x 480 x 30 x 8/8	0.06GB/s 0.06GB/s 0.02GB/s 0.01GB/s

For example, displaying a HD MPEG-4.10 (H.264) video + SD MPEG-4.10 (H.264) + 32-bit HD graphics + 8-bit HD subpicture requires, 1.1 + 0.3 + 0.25 + 0.06 = 1.7GB/s of memory bandwidth to display it on the TV. This does not include the additional memory bandwidth required for writing the graphics and the subpicture data, deinterlacing, audio, and the CPU/XPU requirements. Note that the display on the TV must always be at 30 frames or 60 fields per second.

To assist in calculating the needed memory bandwidth for various applications, additional typical memory bandwidth values for various other operations are:

Table 3-3. **Typical memory bandwidth values**

Function	Format	Typical memory bandwidth needed
Deinterlacing	1920 x 1080 x 30 720 x 480 x 60	TBD
Audio Decoding - Dolby Digital decoding - Dolby Digital Plus decoding - Dolby Lossless decoding - DTS decoding - DTS-HD decoding - MPEG Layers I, II decoding - MPEG Layer III (mp3) decoding - MPEG-4 AAC-LC decoding - MPEG-4 HE-AAC decoding		TBD
Video decoding - MPEG-2 SD decoding - MPEG-2 HD decoding - MPEG-4.2 SD decoding - MPEG-4.2 HD decoding - MPEG-4.10 (H.264) SD decoding - MPEG-4.10 (H.264) HD decoding		TBD
DVD decoding		TBD
BD-ROM decoding		TBD
HD-DVD decoding		TBD
Transport demultiplexer		TBD

Pin Description

DRAM Controller Pins

Table 3-4. **DRAM controller pin descriptions**

Pin Name	Ball ID	Direction	Description
DRAM_DQ31	K20	B	Memory data bus bit 31 (MSB)
DRAM_DQ30	J22	B	Memory data bus bit 30
DRAM_DQ29	K19	B	Memory data bus bit 29
DRAM_DQ28	J21	B	Memory data bus bit 28
DRAM_DQ27	H22	B	Memory data bus bit 27
DRAM_DQ26	J20	B	Memory data bus bit 26
DRAM_DQ25	G23	B	Memory data bus bit 25
DRAM_DQ24	H21	B	Memory data bus bit 24
DRAM_DQ23	F22	B	Memory data bus bit 23
DRAM_DQ22	E23	B	Memory data bus bit 22
DRAM_DQ21	H19	B	Memory data bus bit 21
DRAM_DQ20	G20	B	Memory data bus bit 20
DRAM_DQ19	F21	B	Memory data bus bit 19
DRAM_DQ18	E22	B	Memory data bus bit 18
DRAM_DQ17	F20	B	Memory data bus bit 17
DRAM_DQ16	E21	B	Memory data bus bit 16
DRAM_DQ15	D22	B	Memory data bus bit 15
DRAM_DQ14	C23	B	Memory data bus bit 14
DRAM_DQ13	B23	B	Memory data bus bit 13
DRAM_DQ12	C22	B	Memory data bus bit 12
DRAM_DQ11	E20	B	Memory data bus bit 11
DRAM_DQ10	F19	B	Memory data bus bit 10
DRAM_DQ9	D21	B	Memory data bus bit 09
DRAM_DQ8	C21	B	Memory data bus bit 08
DRAM_DQ7	C20	B	Memory data bus bit 07
DRAM_DQ6	A22	B	Memory data bus bit 06
DRAM_DQ5	A21	B	Memory data bus bit 05

Table 3-4. **DRAM controller pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
DRAM_DQ4	B20	B	Memory data bus bit 04
DRAM_DQ3	C19	B	Memory data bus bit 03
DRAM_DQ2	A20	B	Memory data bus bit 02
DRAM_DQ1	B19	B	Memory data bus bit 01
DRAM_DQ0	A19	B	Memory data bus bit 00 (LSB)
DRAM_A12	L22	O	Memory address bit 12. Memory address bus provides the SDRAM with the row address for the active commands, and the column address and the auto-precharge value for read/write commands.
DRAM_A11	M19	O	Memory address bit 11
DRAM_A10	N23	O	Memory address bit 10
DRAM_A9	M21	O	Memory address bit 09
DRAM_A8	M23	O	Memory address bit 08
DRAM_A7	N21	O	Memory address bit 07
DRAM_A6	P23	O	Memory address bit 06
DRAM_A5	P21	O	Memory address bit 05
DRAM_A4	R23	O	Memory address bit 04
DRAM_A3	P19	O	Memory address bit 03
DRAM_A2	P20	O	Memory address bit 02
DRAM_A1	P22	O	Memory address bit 01
DRAM_A0	N20	O	Memory address bit 00
DRAM_BA1	M22	O	Bank address output 1. BA[1:0] define to which SDRAM bank an active, read, write or precharge command is being applied.
DRAM_BA0	M20	O	Bank address output 0
DRAM_DQS3	G22	B	Data strobes for bits [31:24] of memory data bus. Output by the EM8622L with the write data; output by the SDRAM with the read data.
DRAM_DQS2	G21	B	Data strobes for bits [23:16] of memory data
DRAM_DQS1	D20	B	Data strobes for bits [15:08] of memory data
DRAM_DQS0	D19	B	Data strobes for bits 07:00 of memory data
DRAM_DM3	F23	O	Write data mask for bits [31:24] of memory data. Write data is masked when the corresponding DM bit is sampled high by the SDRAM during a write access.
DRAM_DM2	H20	O	Write data mask for bits [23:16] of memory data

Table 3-4. **DRAM controller pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
DRAM_DM1	B22	O	Write data mask for bits [15:08] of memory data
DRAM_DM0	A23	O	Write data mask for bits [07:00] of memory data
DRAM_RAS#	L20	O	Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested.
DRAM_CAS#	K22	O	Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested.
DRAM_WE#	J23	O	Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested.
DRAM_CK	K23	O	Non-inverted clock output. CK and CK# are a differential clock signal pair. SDRAM address and control signals are sampled on the crossing of the positive edge of CK and the negative edge of CK#. SDRAM output (read) data is referenced to the crossings of CK and CK# in both directions.
DRAM_CK#	L23	O	Inverted clock output. CK and CK# are a differential clock signal pair. SDRAM address and control signals are sampled on the crossing of the positive edge of CK and the negative edge of CK#. SDRAM output (read) data is referenced to the crossings of CK and CK# in both directions.
DRAM_CKE	L21	O	Clock enable. A high level activates, and a low level deactivates the internal clock signals in the SDRAM. This signal is driven low to activate the SDRAM power-down modes.
DRAM_CS#	K21	O	Chip select. Commands are masked when CS# is sampled high by the SDRAM.
DRAM_VREFSSTL3	B21	I	SSTL-2 voltage reference input. For the SDRAM interface pins to support the SSTL_2 signaling specifications, a DC voltage of 1.25V should be applied to these pins.
DRAM_VREFSSTL2	D23	I	SSTL-2 voltage reference input
DRAM_VREFSSTL1	H23	I	SSTL-2 voltage reference input
DRAM_VREFSSTL0	N22	I	SSTL-2 voltage reference input

Electrical Characteristics

DRAM DC Electrical Characteristics

Table 3-5. **I2C master and slave interface DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA			
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA			
V_{IH}[2]	Input high voltage	V			
V_{IL}[2]	Input low voltage	V			

1. Parameter applies to all pins configured as outputs.
2. Parameter applies to all pins configured as inputs.

Timing Diagrams

DRAM Timing Diagrams

The DDR SDRAM controller conforms to the interface specifications defined by the JEDEC standard, JESD79, Release2 (May 2002). All the I/O associated with the DDR SDRAM interfaces conform to the SSTL_2, Class II specification. All generated and required timings conform to the requirements of the JESD79 specification for the ‘DDR333’ speed grade.

Table 3-6. **DRAM interface timing parameters**

Parameter	Units	Minimum	Typical	Maximum
t_{CK}	ns		6.66	
t_{CH}	ns	3		3.66
t_{CL}	ns	3		3.66
t_{IS}	ns	0.9		
t_{IH}	ns	0.9		
t_{DSH}	ns	1.66		
t_{DSS}	ns	1.66		
t_{DS}	ns	0.45		
t_{DH}	ns	0.45		

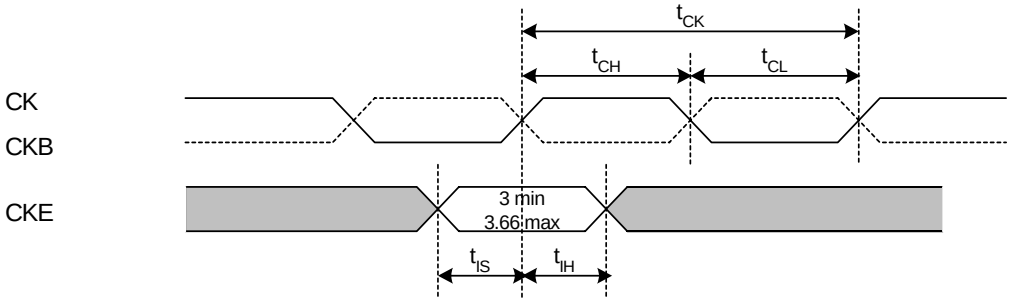


Figure 3-6. **CK/CKB/CKE timing diagram**

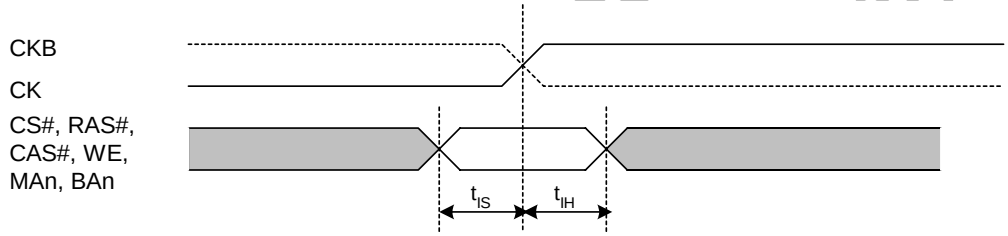


Figure 3-7. **Command timing diagram**

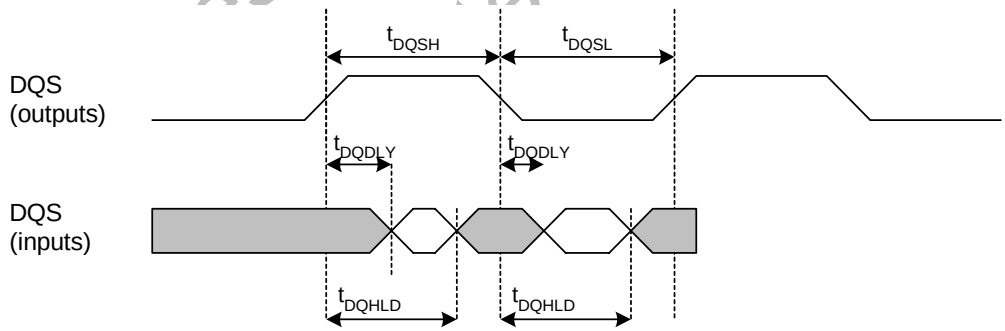


Figure 3-8. **DQ/DQS read timing diagram**

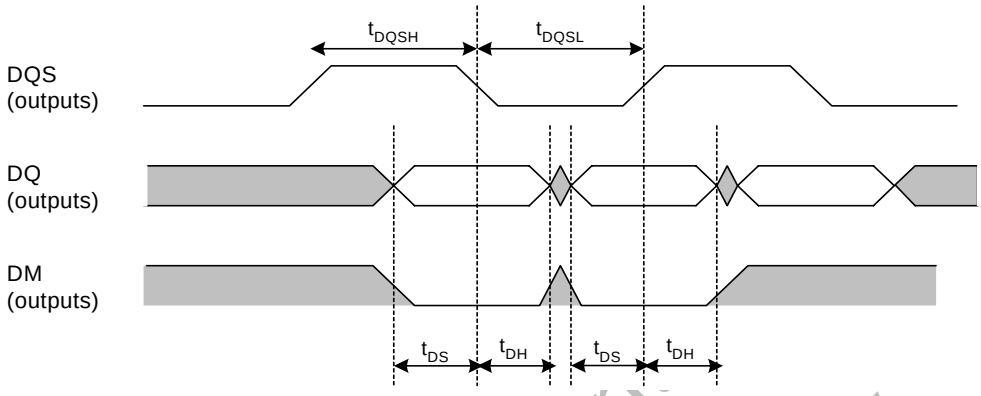


Figure 3-9. **DQ/DQS/DM write timing diagram**

4 System Block

Block Diagram of System Block

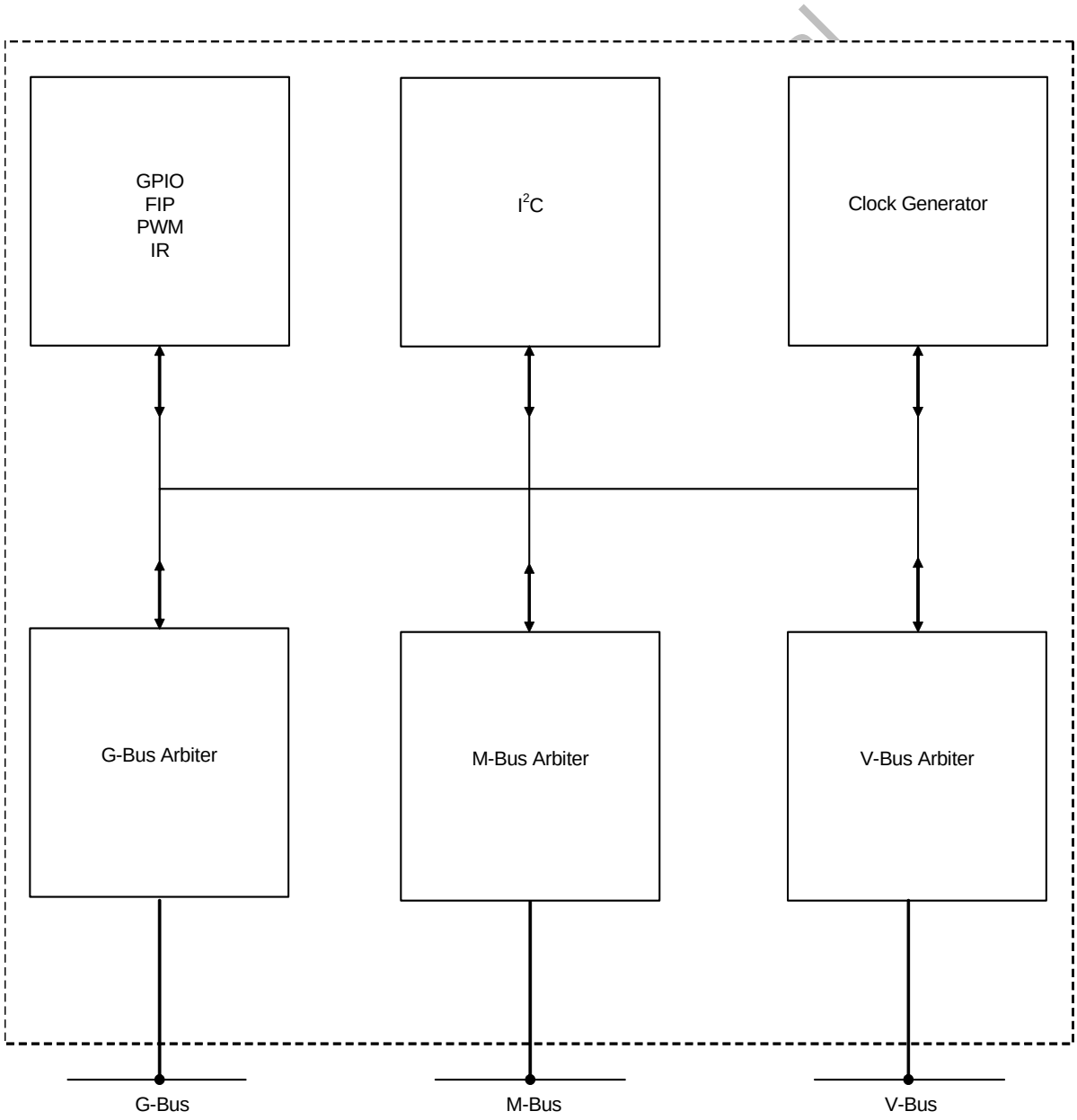


Figure 4-1. **Block diagram of system block**

Introduction

The EM8622L system block contains the following modules:

- A GPIO block containing General Purpose I/O (GPIO), Front Panel Interface (FIP) Controller, Pulse Width Modulator (PWM) and infrared decoder
- I^2C master and slave
- A clock generator
- A G-Bus arbiter that supports 5 masters: MIPS CPU, PCI, MPEG0, MPEG1 and TDMX
- An M-Bus arbiter with 17 channels
- A V-Bus arbiter with 15 channels

Many EM8622L sub-blocks require shared accesses to the internal system buses. The bus arbiters allocate the access to the individual system buses based on the priority of the device.

The system block of EM8622L contains three bus arbiters; G-Bus, M-Bus and the V-Bus arbiters. G-Bus arbiter supports five masters: MIPS CPU, PCI, MPEG0, MPEG1 and TDMX. The M-Bus arbiter has 17 configuration registers, each corresponding to a M-Bus channel. The V-Bus arbiter has 15 configuration registers, each corresponding to a V-Bus channel.

A G-Bus to L-Bus bridge, allows each of the above blocks to be configured through the G-Bus. The DRAM controller has an internal local bus (L-Bus) connecting one master (G-Bus to L-Bus bridge) with four slaves: The G-Bus, M-Bus and the V-Bus units.Through the G-Bus to L-Bus bridge and the L-Bus, any G-Bus master can program configuration registers of all the units. Also, a G-Bus master can access the entire DRAM space through the bridge, the L-Bus and the G-Bus unit.

As shown in the block diagram, the system block connects to each of the three primary device data buses previously mentioned. The system controller unit provides the arbitration logic for the four primary internal buses. The M-Bus and V-Bus arbiters implement a bandwidth allocation algorithm allowing each bus master to be assigned a certain minimum and maximum bandwidth allocation in increments of 2^{-8} of the total available bandwidth.

When certain constraints are met in the allocation of minimum and maximum bandwidth, each master can be guaranteed to receive its minimum bandwidth, and often receives more.

GPIO Block

General Purpose I/O (GPIO)

The general purpose I/O controller provides 16 pins of general purpose control signals and logic to help eliminate the glue logic necessary for system integration. Its functions include indicating the system operation and controlling other devices.

The 16 GPIO pins are individually configurable as inputs or outputs. Pins default to input mode after a device reset. The registers inside configure the direction and output value of each pin. GPIO pins 14 and 15 each have an associated pulse-width modulator (PWM) block which can be enabled in place of the digital output mode. When enabled, the PWM block produces a pulse train on the output pin with a duty cycle controlled by the value of a configuration register. The duty cycle control is a 12-bit value, so the output duty cycle can vary from 0 to $1-2^{-12}$. A simple digital-to-analog converter can be implemented using an external RC filter to integrate the pulse train.

Features

- Supports 16 independent level outputs
- Supports independent polarity controls
- Supports edge detect interrupt on any input transition

Block Diagram

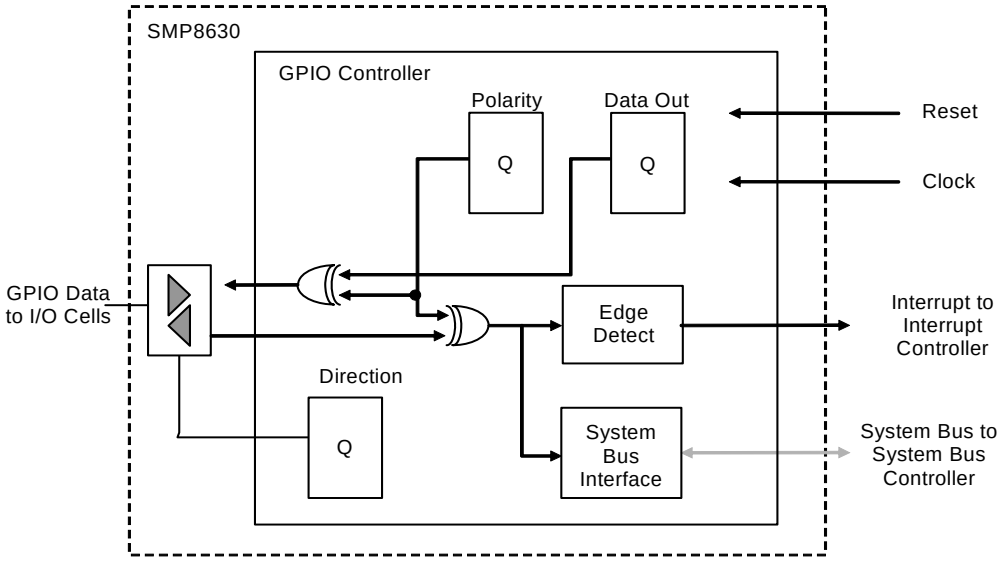


Figure 4-2. **GPIO controller block diagram**

The GPIO block controls the GPIO 16 pins, GPIO[15:0]. These pins can be assigned the following functions:

GPIO: After reset, all the 16 pins have the GPIO function. The direction of each pin can be set by programming the direction register. the default for all the GPIO pins following reset is to be configured as inputs. The value of each pin can be read/written through the data register.

Alternate functions: When an alternate function is enabled (e.g Master I^2C), the corresponding pins (e.g. GPIO[1] and GPIO[0]) are routed to a dedicated hardware block and can no longer be used as GPIOs. There are five fixed-pin alternate functions (PWM0, PWM1, FIP, I^2C Slave and I^2C Master) which can be independently enabled.

Selectable inputs: These are input functions only (e.g. PCI interrupt input A). They are pre-assigned to a pin at reset, but can be re-assigned to another pin using the control register. The pin should have the GPIO function, and be programmed as an input.The following figure shows the GPIO alternate function and selectable inputs in detail:

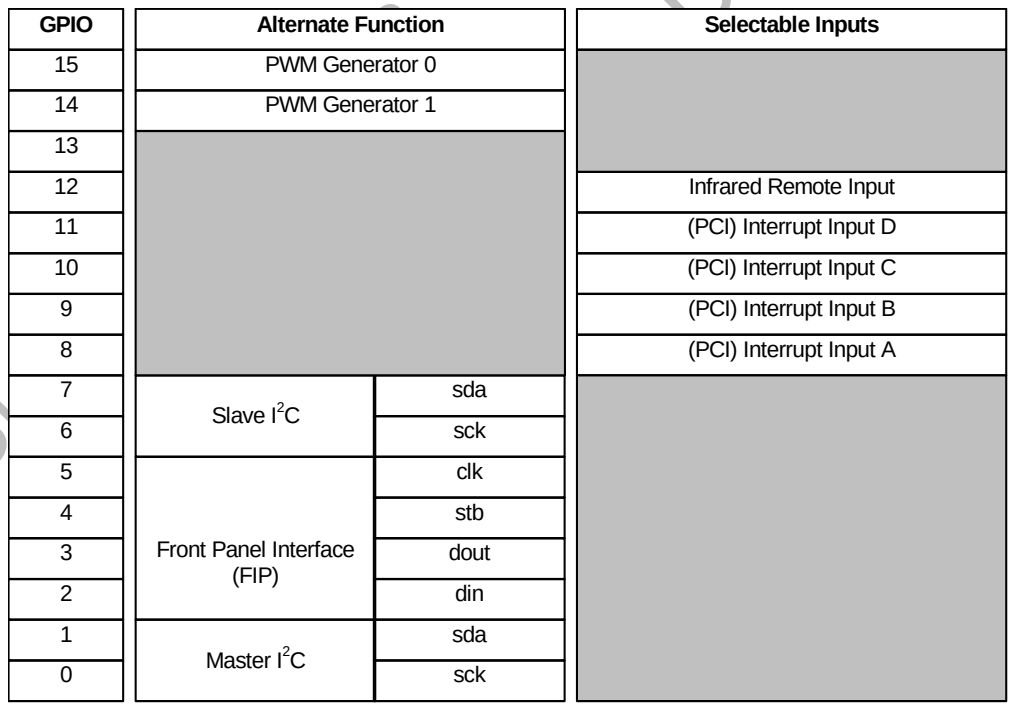


Figure 4-3. **GPIO alternate and selectable inputs**

General Purpose I/O Controller

The GPIO is a bidirectional interface that can be used for general interfacing applications, such as controlling the external hardware and sensing the external conditions. The controller has 16 independently programmable pins. Each pin can be programmed as an input or output, possesses independent polarity controls, and supports an edge detect interrupt for input transitions. The GPIO direction register controls the data direction of the 16 GPIO pins (0 = Input, 1 = Output).

The data register controls the data output on the 16 GPIO pins. It can be written in the same manner as the direction register. The data register bits are used for driving or sensing the static signals on the GPIO pins. To drive a value onto a GPIO pin, the corresponding bit in the GPIO direction register must be set. If the corresponding direction bit is set, then the value written to the bit in the GPIO data register will be driven at the pin. A read of this register returns the value of the signals currently on the GPIO pins.

The GPIO PWM registers control the PWM generators connected to GPIO15/GPIO14 pins. An external low pass filter is used to generate a 4096 level analog voltage. The minimum pulse width at the PWM output is equal to 2(div+1) system clock periods.When the bit E is set, the PWM generator is enabled, forcing GPIO15 to an output (Overrides dir[15] and data[15])

Note: The 8 I/O pins of any UART block can be shared to provide additional GPIO functionality. For more details see the section on UART in the chapter *Host CPU Block*.

Independent level-sensitive I/O

The 16 pins can be set to either input or output modes using the firmware. In the output mode, the pins may be set to a high or a low voltage. Each bit has an independent polarity control to keep the pin voltage independent of the logical signal level.

GPIO interrupt

An interrupt may be generated when any input-only pin transitions from a low to a high state or vice versa with respect to the polarity flag. This interrupt is useful if the GPIO events are infrequent or unpredictable (such as, from a keypad or a remote control button). If it is not needed, then the interrupt can be masked in the interrupt controller.

Register Map

GPIO Registers

Table 4-1. **GPIO registers**

Address[1]	Register Name	R/W/A[2]	Description
+0500	SYS_GPIO_DIR	R/W	GPIO Direction Register
+0504	SYS_GPIO_DATA	R/W	GPIO Data Register
+0510	SYS_GPIO15_PWM	R/W	GPIO15 PWM Register
+0514	SYS_GPIO14_PWM	R/W	GPIO14 PWM Register

1. Address refers to G-Bus byte address relative to the system block register base.
2. Read/Write/Auto update.

Pin Description

GPIO pins

The EM8622L provides 16 general-purpose I/O (GPIO) pins which may be individually configured as inputs or outputs and used for design-specific applications.

Table 4-2. **GPIO pin description**

Pin Name	Ball ID	Direction	Description
GPIO15	A4	B	General-purpose IO pin 15
GPIO14	C5	B	General-purpose IO pin 14
GPIO13	B4	B	General-purpose IO pin 13
GPIO12	D5	B	General-purpose IO pin 12
GPIO11	A3	B	General-purpose IO pin 11
GPIO10	C4	B	General-purpose IO pin 10
GPIO9	B3	B	General-purpose IO pin 9
GPIO8	A2	B	General-purpose IO pin 8
GPIO7	E5	B	General-purpose IO pin 7
GPIO6	C3	B	General-purpose IO pin 6
GPIO5	D4	B	General-purpose IO pin 5
GPIO4	B2	B	General-purpose IO pin 4
GPIO3	C2	B	General-purpose IO pin 3
GPIO2	A1	B	General-purpose IO pin 2

Table 4-2. **GPIO pin description**

Pin Name	Ball ID	Direction	Description
GPIO1	B1	B	General-purpose IO pin 1
GPIO0	C1	B	General-purpose IO pin 0

Electrical Characteristics

GPIO Electrical Characteristics

Table 4-3. **GPIO DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
V_{IH}[2]	Input high voltage	V	2		5.5
V_{IL}[2]	Input low voltage	V	-0.3		0.8

1. Parameter applies to all pins configured as outputs.
2. Parameter applies to all pins configured as inputs.

Front Panel Controller Interface (FIP)

Introduction

The Front Panel Controller Interface (FIP) directly supports the NEC uPD16311, NEC uPD16312, PTC PT6311 and PTC PT6312. Other front panel controllers may be used by interfacing to the UART or the GPIO.

Features

- Supports NEC uPD16311 and uPD16312
- Supports PT6311 and PT6312
- Interrupt driven
- Works on a fixed clock rate of 27MHz

Block Diagram

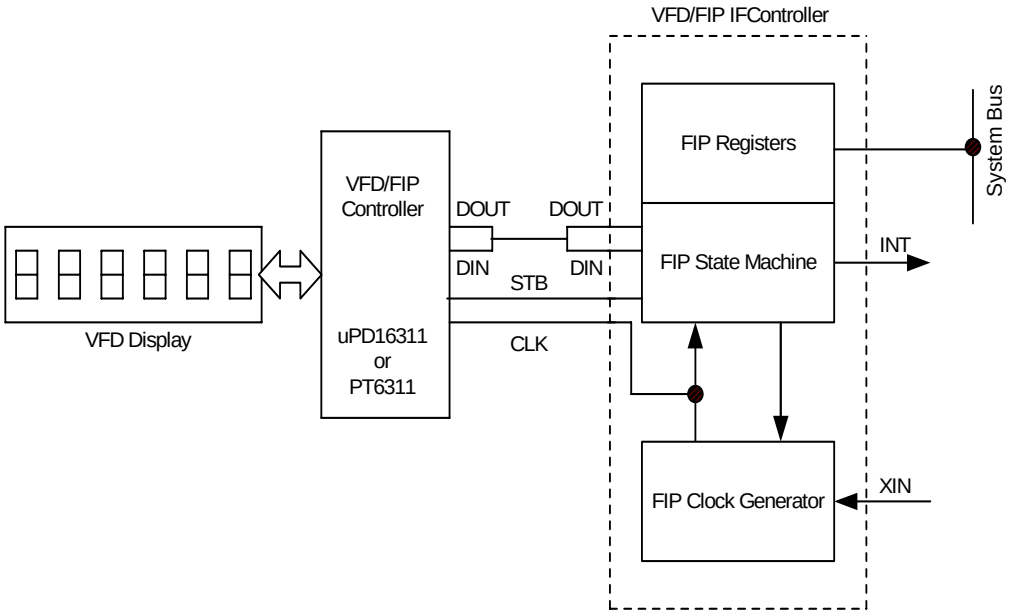


Figure 4-4. **FIP block diagram**

Functional Description

This block is designed to connect directly to the NEC uPD16311, NEC uPD16312, PT6311 or PT6312 VFD/FIP controller chips. The VFD/FIP controller chips mentioned are commonly used to control VFD displays used in consumer appliances. When enabled PIO[5:2] are used as control pins DOUT, DIN, STB and CLK.

In some applications DIN and DOUT are connected together on both sides (EM8622L and the VFD controller/FIP controller chip) to save one communication line. In this case, the tristate mode needs to be enabled and an external pull-up resistor connected.

The FIP command register is used to provide a command to a VFD/FIP device to set the display mode, data read/write mode, address of the display memory and the display control. The display data register stores the display data. The LED data is stored by the LED data register.

The key data registers are used to the store key data. The FIP controller will serially shift out the key data from its key input storage RAM after a write to the FIP Command register. The switch data register is used to store the switch input data. It is valid after a write to the FIP command register.

The configuration register controls the enabling of the FIP controller interface. The configuration register value is used to divide the input clock and generate the FIP clock (default = 27d).

Frequency (FIP_CLK) = Frequency (XIN) / (FIP_CLK_DIV + 1); where, FIP_CLK_DIV should be greater than 2, and values 0, 1 and 2 default to 2.

The configuration register also issues an interrupt on the completion of read or write transactions. The interrupt status register is used to clear the interrupts. It is recommended to clear this register before enabling the interrupts.

Register Map

FIP Registers

Table 4-4. **FIP registers**

Address[1]	Register Name	R/W/A[2]	Description
+0540	FIP_COMMAND	R/W	FIP Command Register
+0544	FIP_DISPLAY_DATA	R/W	FIP Display Data Register
+0548	FIP_LED_DATA	R/W	FIP LED Data Register
+054C	FIP_KEY_DATA1	R	FIP Key Data1 Register
+0550	FIP_KEY_DATA2	R	FIP Key Data2 Register
+0554	FIP_SWITCH_DATA	R	FIP Switch Data Register
+0558	FIP_CONFIG	R/W	FIP Configuration Register
+055C	FIP_INT	R/C[3]	FIP Interrupt Status Register

1. Address refers to G-Bus byte address relative to the system block register base.
2. Read/Write/Auto update.
3. Read/Clear

Pin Description

FIP Pins

Table 4-5. **FIP pin description**

Pin Name	Ball ID	Direction	Description
GPIO2	A1	B	FIP serial data input
GPIO3	C2	B	FIP serial data output
GPIO4	B2	B	FIP data strobe
GPIO5	D4	B	FIP serial I/O clock

Electrical Characteristics

FIP Electrical Characteristics

Table 4-6. **FIP DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	27	46
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
V_{IH}[2]	Input high voltage	V	2		5.5
V_{IL}[2]	Input low voltage	V	-0.3		0.8

1. Parameter applies to all pins configured as outputs.
2. Parameter applies to all pins configured as inputs.

FIP AC Electrical Characteristics

Table 4-7. **FIP AC characteristics**

Symbol	Description	Units	Min	Typ	Max

Timing Diagram

TBD

Infrared Decoder

Introduction

The infrared input allows the interfacing to an external IR receiver. The NEC, Phillips RC5/RC6 Mode 6A IR formats, commonly used by consumer equipment are supported.

Features

- Supports RC5 and RC6 Mode 6A formats
- Supports RC5 extended format
- Supports NEC format
- Interrupt driven
- Contains error detection

Block Diagram

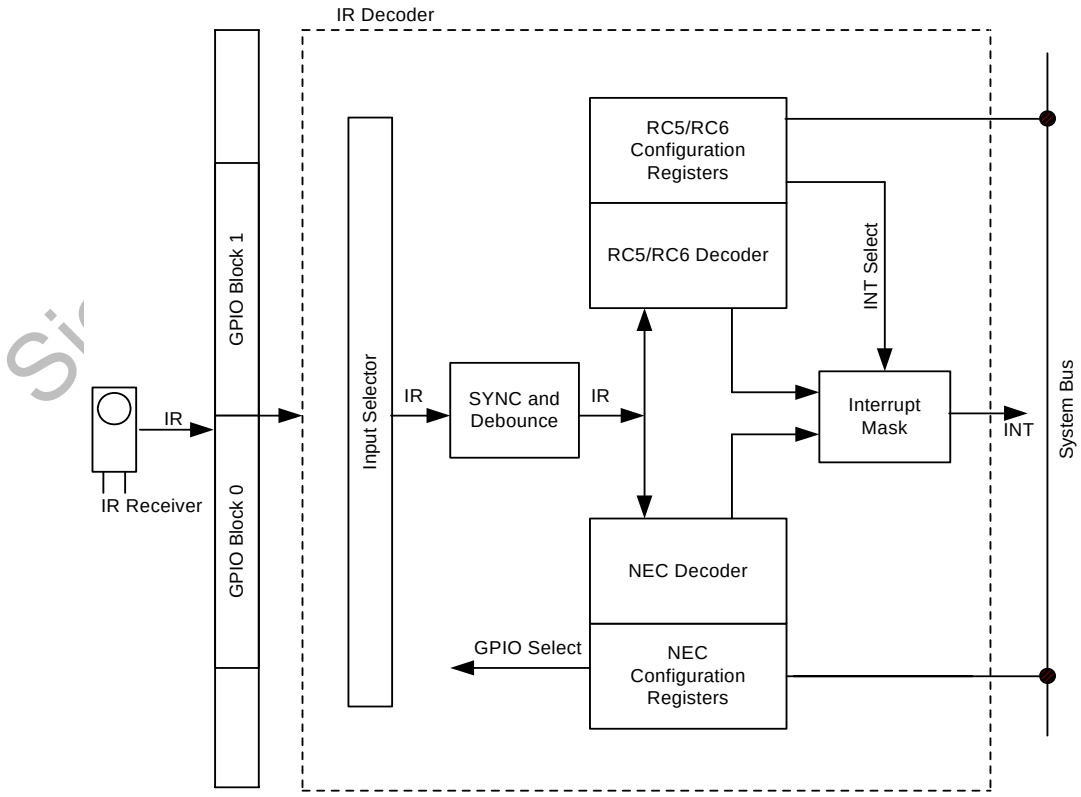


Figure 4-5. **Infrared decoder block diagram**

Functional Description

This block enables the user to receive the IR scan codes from a RC5/RC6 Mode 6A/NEC standard compliant remote control.

The control register sets the number of bits to be captured in one frame. It also sets the pre-devider value for the NEC IR decoder. Using this register a GPIO pin may be selected to be used as an input for the NEC and RC5/RC6 Mode 6A IR decoder block.

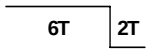
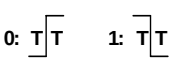
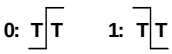
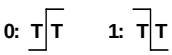
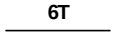
The decoder data register contains the last scan code captured. This register can be used to detect whether a RC5/RC6 Mode 6A scan code, or a NEC scan code was received.

Supported RC6 Mode 6A Formats

LB	SB	MB [2:0]	T	SCC [7:0] or LCC [15:0]	SCIF [X*8-1:0]; X={1,2} LCIF [X*8-1:0] X={3, ... , 16}	SFT
HEADER				CTRL FIELD	INFORMATION FIELD	LO

RC6A frame

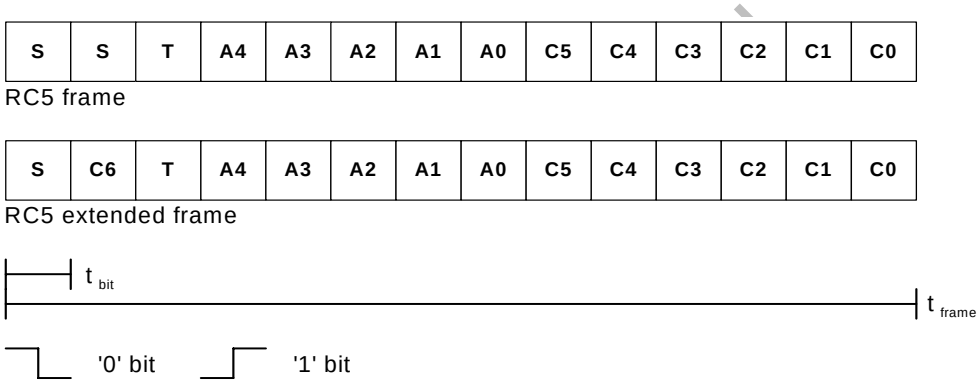
T = 444.44 us

LB	Header Leader bit, duration 8T (6T high, 2T low)	
SB	Header Start bit, duration 2T (T high, T low)	
MB [2:0]	Header Mode bits, duration 3 * 2T, in Mode 6A MB[2:0]='110'	
T	Header Trailer bit, duration 4T, in Mode 6A T= '0'	
SCC [7:0] or LCC [15:0]	Short Customer Code, duration 8 * 2T, SCC [7] = 0 Long Customer Code, duration 16 * 2T, LCC[15] = 1	
SCIF [X*8-1:0]; X={1,2} LCIF [X*8-1:0] X={3, ... , 16}	Short Customer Information Field, duration X*8*2T X={1,2} Long Customer Information Field, duration X*8*2T X={3, ... , 16}	
SFT	Lead out = Signal Free Time, duration 6T	

Supported RC5 Formats (Standard and Extended)

A frame is emitted from a RC5 compliant IR remote control once a key is pressed. If the key is held down, the same frame will be sent repeatedly. If the same key is pressed again (after releasing), the same frame will be sent again, except that the T bit will be inverted.

The extended RC5 format supports 128 commands rather than 64. This is achieved by using the 2nd start bit of the standard RC5 format as bit C6.



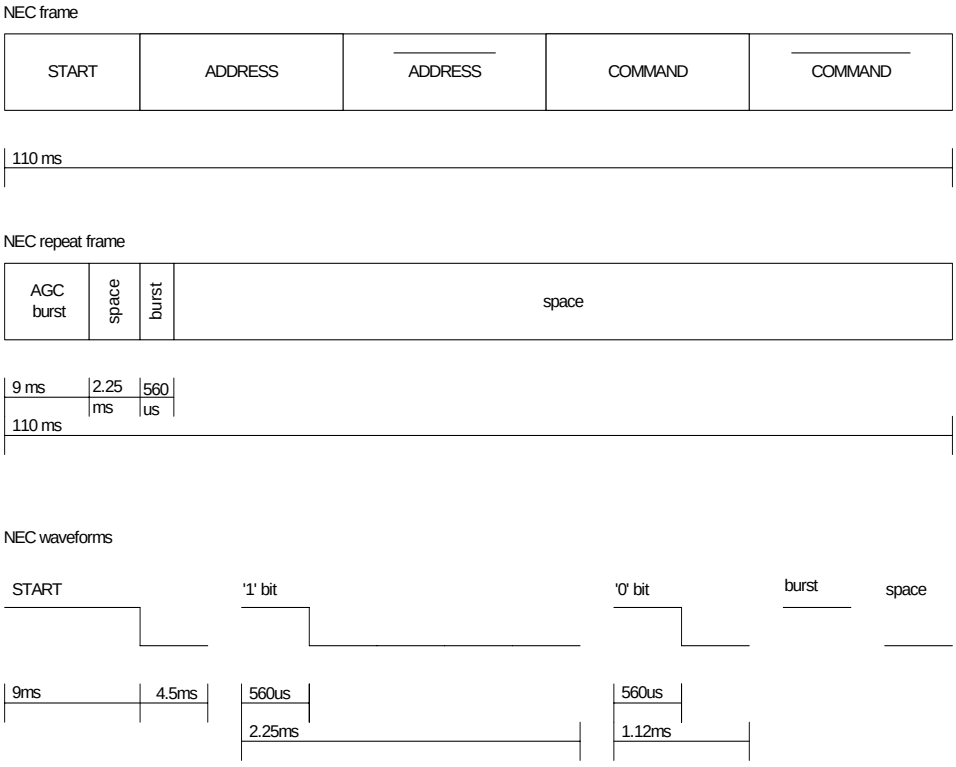
S: Start bit, T: Toggle bit, A[4:0]: 5 system address bits, C[5:0]: 6 system command bits (supported in the RC5 standard format), C[6:0]: 7 system command bits (supported only in the RC5 extended format)

Table 4-8. **RC5 timing parameters**

	Minimum	Typical	Maximum	Units
t_{bit}	1.334	1.778	2.222	ms
t_{frame}		24.889		ms

Supported NEC Format

A frame is emitted from a NEC compliant IR remote control once a key is pressed. After the START condition is transmitted, ADDRESS and ADDRESS# are sent (LSB first). The ADDRESS# represents the bit compliment of ADDRESS. Then the COMMAND and COMMAND# are transmitted wherein COMMAND# represents the bit compliment of COMMAND; transmission is LSB first. If the key is held down, a repeat frame is being sent by the remote.



START: Start condition, ADDRESS: 8-bit system address, COMMAND: 8-bit system command

Table 4-9. **NEC timing parameters**

	Minimum	Typical	Maximum	Units
t_{frame}		110		ms

Register Map

Infrared Decoder Registers

Table 4-10. **Infrared decoder registers**

Address[1]	Register Name	R/W/A/C[2]	Description
+0518	IR_NEC_CONTROL	R/W	Infrared NEC Control Register
+051C	IR_NEC_DECODER_DATA	R	Infrared NEC Decoder Data Register
+0520	IR_RC5_DECODER_CONTROL	R/W	Infrared RC5 Decoder Control Register
+0524	IR_RC5_DECODER_CLK_DIV	R/W	Infrared RC5 Decoder Clock Divisor Register
+0528	IR_RC5_DECODER_DATA	R	Infrared RC5 Decoder Data Register
+052C	IR_INT_STATUS	R/C	Infrared Interrupt Status Register

1. Address refers to G-Bus byte address relative to the system block register base.
2. Read/Write/Auto update/Clear.

Pin Description

Infrared Decoder Pin

Table 4-11. **Infrared decoder pin description**

Pin Name	Ball ID	Direction	Description
GPIO12	D5	B	Default pin assigned to the IR decoder input function (may be mapped to any other GPIO pin under the software control).

Electrical Characteristics

Infrared Decoder DC Characteristics

Table 4-12. **Infrared decoder DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	27	46
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
V_{IH}[2]	Input high voltage	V	2		5.5
V_{IL}[2]	Input low voltage	V	-0.3		0.8

1. Parameter applies to all pins configured as outputs.
2. Parameter applies to all pins configured as inputs.

Infrared Decoder AC Characteristics

Table 4-13. **Infrared decoder AC characteristics**

Symbol	Description	Units	Min	Typ	Max

Inter Integrated Circuits (I^2C)

Introduction

The I^2C master and slave interfaces enable the EM8622L to read from and write to external devices. The I^2C master controller, which supports the synchronous Inter Integrated Circuits (I^2C) serial protocol, enables the host CPU to access an external I^2C slave device using a simplified register interface. A separate slave interface allows the EM8622L to be the target of I^2C transactions initiated by an external master. Both interfaces accommodate bidirectional data transfer, have programmable address width of up to 8-bit with sequential byte read or write capability, and generate interrupts whenever bytes are transmitted or received.

Features

- Supports synchronous Inter Integrated Circuits (I^2C) serial protocol
- Supports bidirectional data transfer
- Supports programmable address width up to 8-bit
- Capable of sequential byte read or write
- Generates interrupts when bytes are received/transmitted
- Supports programmable I^2C bus clock rate
- Supports transmission of device address and register address to do device, page and address selection to perform read and write accesses.
- Each of the interfaces support 100Kbps or 400Kbps bit rates

Block Diagram

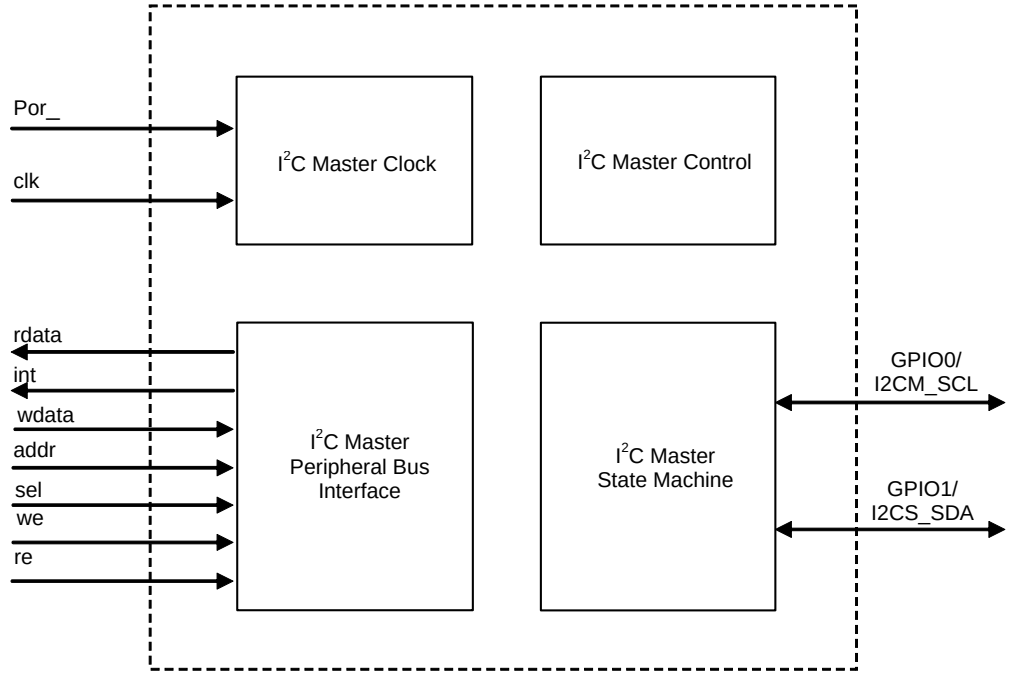


Figure 4-6. **I²C master block diagram**

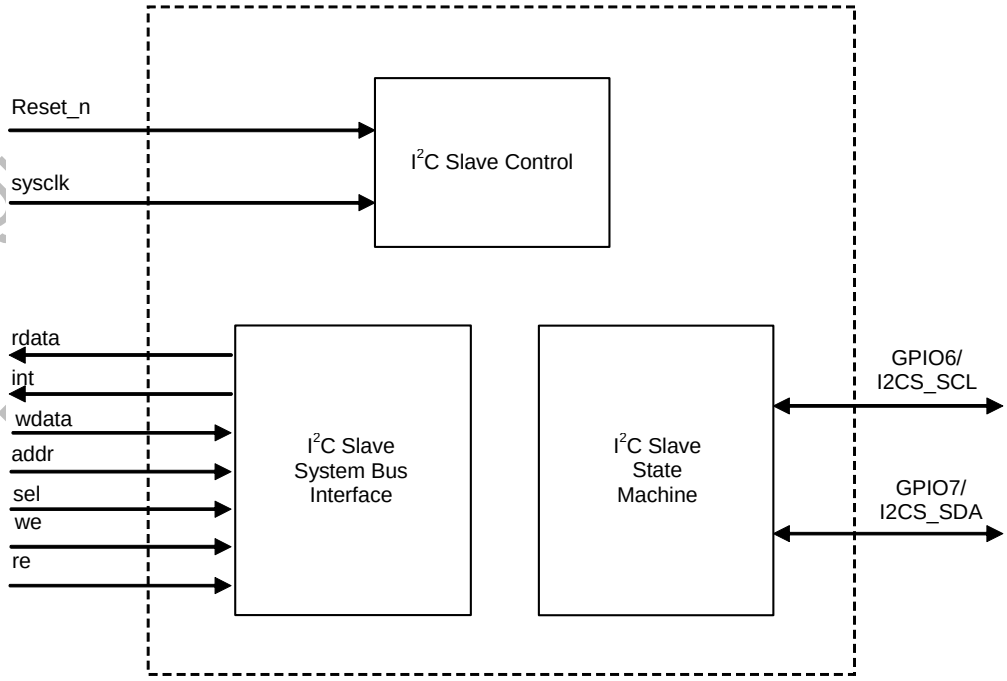


Figure 4-7. **I²C slave block diagram**

Functional Description

I²C Master

This I²C master controller enables the host CPU to access an external I²C slave device using a simplified register interface.

A configuration register controls the status of the I²C master. When enabled, the I²C master connects to the GPIO pins. When disabled, the state machine is reset and the I²C master disconnects from the GPIO port. When the bit DEVADRDIS is set to '1', the I²C host controller will not transfer the device address. When set to '0', the device address will be transmitted before the register address or data. When the bit REGADRDIS is set to '1', the I²C host controller will not transfer the register address data, but will only read or write the serial data. When set to '0', normal transfers will occur with the address being shifted out followed by the data being read or written.The register address register stores the address to be sent to the external I²C slave devices, if the bit ADRDIS is not set.

First, the software will configure the I²C controller by programming the register CONFIG and selecting the proper value of the register CLKDIV. Then, the software may program the device address register DEVADR with an address and/or data to be written to or read from the external I²C slave device. To access an external I²C slave device, the CPU should program the registers DEVADR and/or ADR/DATAOUT. To do a burst read or a burst write the register BYTECNT needs to be set as well.

The direction and the start of the transfer is determined by a bit in the register STARTXFER. Writing to this bit starts the I²C state machine. It will handle all the proper hardware signalling.

The state machine starts by asserting a start condition on the I²C bus, followed by shifting the seven bits contained in the register DEVADR. It checks for an acknowledge signal from the addressed slave device. If there is no acknowledgement, then a bit is set in the register I²C STATUS. In case the bit ADRDIS was set to '0', then the state machine will take the data from the register ADR. Then in the case of a write it will shift out the eight bits in the register DATAOUT or, in case of a read it will shift the data into the register DATAIN.

The I²C bus consists of two bidirectional lines, the serial data (SD) line, and the serial clock (SCLK) line. A start condition on the I²C bus is indicated by asserting the SD low when the SCLK is high. Subsequent device address and data are shifted at the rate of 1-bit per SCLK (see the I²C master timing diagram).

The transmitter on the I^2C bus sets the data on the SD line during the low phase of the clock, the receiver on the I^2C bus samples the data on during the high phase of the clock. When the host I^2C controller is done transmitting or receiving data on the I^2C bus it will assert a stop condition by releasing the SD line during the high phase of the SCLK. The clock divider register is used to generate the I^2C bus SCLK by applying the following equation: SCLK = SYS_CLK / (2 x CLKDIV).

When the start transfer register is written to, the I^2C state machine starts a read/write access on the I^2C bus. A read/write direction bit is sent out to the I^2C slave devices. It is written with a '1' for reads, and written with a '0' for writes. For a sequential read/write the bit RWDIR needs to be programmed only once to start the transmission of the sequence.

Setting the bit DUMMY to '1' will enable the DUMMY write, where no data transfer is required. This is useful when the host CPU is trying to do a random read from particular I^2C slave memory devices, the I^2C state machine would shift the device address followed by the address, if not disabled, then returns to IDLE without shifting out the data in the register DATAOUT. This bit should be written to '0' for normal I^2C bus accesses.

At the end of a transfer a 'stop condition' or a 'no stop condition' is issued. Therefore the next transfer will begin with a 'repeated start'. The value written to the byte count register (plus one) will indicate the number of bytes to be written to or read from the external I^2C slave devices. It is used when the firmware is doing a sequential read or write. After every sequence, this register needs to be reprogrammed.

During the I^2C writes, as soon as the register STARTXFER is written to by the processor, a start condition is automatically asserted. When the transfer starts, the bit I2CIDLE in the register STATUS is cleared to 0. The device address in the bit DEVADR will be serially shifted out followed by the bit RWDIR in the register STARTXFER, and the 8-bit write data is stored in the register DATAOUT.

Once the transaction is completed, the I^2C host controller asserts a stop condition. Then the I2CIDLE bit will be set to '1'. During a sequential write to an external I^2C slave device, the bit DOUTEMPTY is set in the register STATUS after every byte is shifted out. The state machine then holds the I^2C bus SCLK low, until the host CPU supplies new data to the register DATAOUT. The I^2C state machine keeps repeating the loop until all the bytes have been shifted out.

During the I^2C reads, as soon as the register STARTXFER is written to by the processor, a Start Condition is automatically asserted on the I^2C bus. When the transfer starts, the bit I2CIDLE in the register STATUS is cleared to 0. The device address in the bit DEVADR will be serially shifted out followed by the bit RWDIR. The external device begins driving the read data on the serial data line, and the I^2C host controller will start sampling the data and store them in the register DATAIN. The transaction is completed, as soon as the I^2C host controller initiates a stop condition. The bit I2CIDLE will be set to '1'. During a read from the external device, the 8 bits which are received from the external I^2C slave devices are stored in the data in register. The bit DATARDY in the register STATUS will be set to '1' when this register contains valid data.

During a sequential read from an external I^2C slave device, the bit DATAREADY is set in the register STATUS after every byte is shifted in. The state machine then holds the I^2C bus SCLK low until the host CPU reads the register DATAIN. The I^2C state machine keeps repeating the loop until all bytes have been shifted in.

I^2C Slave

The SDA and SCL are the two pins that control the operation of the 2-wire-bus. The SDA is the bidirectional open-drain serial data IO and the SCL is the serial interface input clock controlled by the master. The EM8622L acts as a slave device and the device address register is used to select the 7-bit slave address of the EM8622L. This register also enables the I^2C slave.

The data received and transmitted through the SDA must be stable while the SCL is high. The data on SDA can change their states only when the SCL is low. The start signal is the falling edge of SDA when the SCL is high. The start signal informs all the slave devices that a data transfer sequence is initiated. In contrast, the stop signal is signalled by the rising of the SDA while the SCL is high. When the serial interface is not active, the logic level of the SDA and the SCL are both high, due to external pull-up resistors.

The first data transferred after the start signal is the 7-bit slave address and the bit R/W. When the bit R/W is high, the master reads from the registers in the slave. When the bit R/W is low, the master writes to the registers inside the slave. If the transmitted slave address matches the address programmed by the address register, then the EM8622L acknowledges being selected by bringing the SDA low on the ninth pulse of the SCL clock. Otherwise, the EM8622L does not acknowledge, and both the SDA and the SCL are high and the device remains in inactive mode.

In order to write the data to the EM8622L, the master needs to provide the 7-bit slave address and set the bit R/W to 0. Then the master needs to provide the data to the EM8622L. An interrupt will be generated and the EM8622L software will need to read the data register. If an external master reads the device, then this 8-bit data will be sent to that master. During a write from the external device, the 8-bit data that was received from the external I^2C master device is stored here. The bit DATARDY in the register STATUS will be set to '1' when this register contains valid data. It means that the register DATAIN has valid data in it. This bit is cleared when the register DATAIN is read by the host CPU.

If the I^2C slave controller data has not been read fast enough by the host CPU before the next byte from the external I^2C master, then an 'overflow' error occurs and an interrupt is asserted. The register DATAIN needs to be read once to clear it from the corrupted (last) data. This way the register DATAIN, the status register, a pending 'overflow' interrupt and 'data ready' are all cleared for a new transaction.

If the host does not send new bytes to the I^2C slave before the master read access, then an 'underflow' error occurs and an interrupt is asserted. The interrupt is cleared when the slave is supplied with new the data.

If a master is reading, and the data is not ready (inside the I^2C slave), then the I^2C slave can hold the bus until the data becomes ready by forcing the I^2C clock to low. The bus hold register can be used for this bus holding function.

When the master reads from the slave, and the data is not ready, the slave will hold the bus if the bit HOLDEN is 1. At the same time it will start a time-out counter (count down) by loading the value bit HOLDCOUNT. The slave will end the bus holding when either the data is read, or the counter expires (becomes zero). If the counter expires before the data is ready, then an 'underflow' interrupt is issued.

If the interrupt mode bit is 0, then the 'doutempty' interrupt is issued as soon as the current data is read by an I^2C master. That means that the bus hold register always contains the data provided that the interrupt is serviced. It also means that, a succession of bytes needs be provided to the slave. Inside the slave module are two registers, DATAOUT and SHIFT. When the slave detects that the master is starting a read transfer, it moves the data from the register DATAOUT to the register SHIFT and immediately issues a 'doutempty' interrupt to the host; so that the host will have more time to service the interrupt (interrupt service latency). Thus, there will always be an extra interrupt at the end of the transfer.

If the interrupt mode bit is 1, then the ‘doutempty’ interrupt is issued only if a byte is requested by a master read. When the slave detects that the master is starting a read transfer, it issues a ‘doutempty’ interrupt to the host and holds the I^2C bus (by forcing the I^2C clock to low) until, either the data is provided by the host CPU or the bus holding times out. Thus, there will be no extra interrupt, but the I^2C bus speed may reduce.

Also, in the I^2C master write, after receiving one byte the slave will hold the bus until the host CPU reads the byte. Here the overflow does not occur and the bytes are not missed.

Register Maps

I^2C Master Registers

Table 4-14. **I^2C master registers**

Address[1]	Register Name	R/W/A[2]	Description
+0580	I2C_MASTER_CONFIG	R/W	I^2C Master Configuration Register
+0584	I2C_MASTER_CLK_DIV	R/W	I^2C Master Clock Divisor Register
+0588	I2C_MASTER_DEV_ADDR	R/W	I^2C Master Device Address Register
+058C	I2C_MASTER_ADR	R/W	I^2C Master Address Register
+0590	I2C_MASTER_DATAOUT	R/W	I^2C Master Data Out Register
+0594	I2C_MASTER_DATAIN	R	I^2C Master Data In Register
+0598	I2C_MASTER_STATUS	R	I^2C Master Status Register
+059C	I2C_MASTER_STARTXFER	R/W	I^2C Master Start Transfer Register
+05A0	I2C_MASTER_BYTE_CNT	R/W	I^2C Master Byte Count Register
+05A4	I2C_MASTER_INTEN	R/W	I^2C Master Interrupt Enable Register
+05A8	I2C_MASTER_INT	R/C[3]	I^2C Master Interrupt Register

1. Address refers to G-Bus byte address relative to the system block register base.
2. Read/Write/Auto update.
3. Read/Clear.

I²C Slave Registers

Table 4-15. **I²C slave registers**

Address[1]	Register Name	R/W/A[2]	Description
+05C0	I2C_SLAVE_ADDR_REG	R/W	I²C Slave Device Address Register
+05C4	I2C_SLAVE_DATAOUT	R/W	I²C Slave Data Out Register
+05C8	I2C_SLAVE_DATAIN	R	I²C Slave Data In Register
+05CC	I2C_SLAVE_STATUS	R	I²C Slave Status Register
+05D0	I2C_SLAVE_INTEN	R/W	I²C Slave Interrupt Enable Register
+05D4	I2C_SLAVE_INT	R/C[3]	I²C Slave Interrupt Register
+05D8	I2C_BUS_HOLD	R/W	I²C Bus Hold Register

1. Address refers to G-Bus byte address relative to the system block register base.
2. Read/Write/Auto update.
3. Read/Clear.

Pin Description

I²C Master and Slave Interface Pins

Table 4-16. **I²C master and slave interface pin descriptions**

Pin Name	Ball ID	Direction	Description
GPIO0	C1	B	I2CM_SCL. I²C master interface serial clock
GPIO1	B1	B	I2CM_SDA. I²C master interface serial data
GPIO6	C3	B	I2CS_SCL. I²C slave interface serial clock
GPIO7	E5	B	I2CS_SDA. I²C slave interface serial data

Electrical Characteristics

I²C Master and Slave Interface DC Characteristics

Table 4-17. **I²C master and slave interface DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	27	46
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
V_{IH}[2]	Input high voltage	V	2		5.5
V_{IL}[2]	Input low voltage	V	-0.3		0.8

1. Parameter applies to all pins configured as outputs.
2. Parameter applies to all pins configured as inputs.

I²C Master and Slave Interface AC Characteristics

Table 4-18. **I²C master and slave interface AC characteristics**

Symbol	Description	Units	Min	Typ	Max

Timing Diagrams

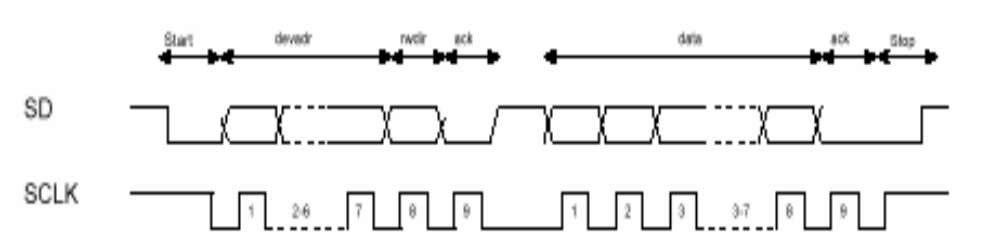


Figure 4-8. **I²C master transfer timing diagram**

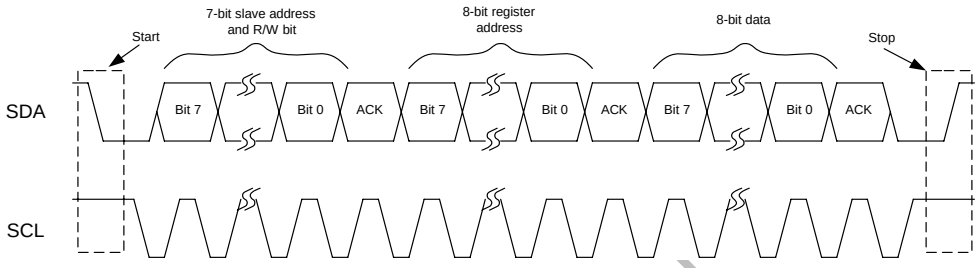


Figure 4-9. **I²C slave transfer timing diagram**

Clock Generator

Introduction

The clock generator contains audio clocks, video clocks, system clock (200MHz) and a CPU clock. The clock generator creates two high speed (up to 200MHz) clocks from a 27MHz external clock using two programmable PLLs, and creates the main system clock and multiple video and audio clocks by dividing either one of the high speed clocks or the 27MHz reference.

The clock generator module receives seven different clock inputs, and generates 13 separate clock and timer tick outputs. The clock generator module contains four independent frequency synthesizers based on phase-locked loops (PLLs). Every PLL except one, can use any of the seven input sources as its reference clock; PLL0 always receives its reference clock from the XTAL_IN pin (nominal 27MHz source) and generates the internal system clock.

Features

- Creates two high speed clocks (up to 200MHz)
- Contains audio clocks, video clocks, system clock and a CPU clock
- Supports programmable PLLs

Block Diagram

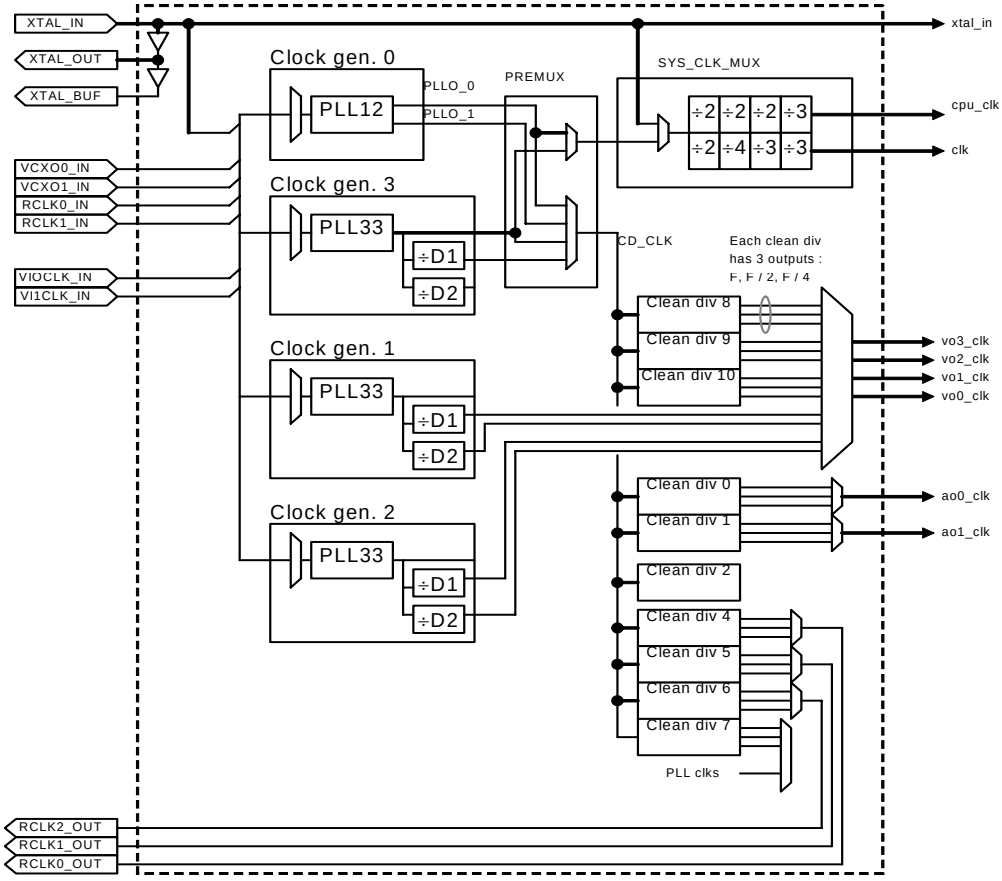


Figure 4-10. **Clock generator block diagram**

Functional Description

The three PLL registers determine the multiplying and dividing factors. When the PLL bypass bit is set, the post-dividers are driven by the PLL input (rather than the PLL output). The direct output is not affected by the bypass. The PLL output frequency is equal to, $F_{Out} = F_{In}$ x (N+2) / (M+2). The PLL input source selection is encoded according to the following table:

0	1	2	3	4	5	6	7
PWRDWN	XTAL_IN	VCXO0_IN	VCXO1_IN	RCLK0_IN	RCLK1_IN	VI0_CLK	VI1_CLK

The divider registers determine the post-dividers ratio (must be between 2 and 15). The PLL output (PLL input when bypass is activated) is divided by D1 to generate CK1, and divided by D2 to generate CK2. The clean dividers generate an output frequency equal to, F_{Out} = SysClk / (2 + divider x 2^{-27}).

Register Map

Clock Generation Registers

Table 4-19. **Clock generation registers**

Address[1]	Register Name	R/W/A[2]	Description
+0000	SYS_CLKGEN0_PLL	R/W	System Clock Generator 0 PLL Register
+0008	SYS_CLKGEN1_PLL	R/W	System Clock Generator 1 PLL Register
+000C	SYS_CLKGEN1_DIV	R/W	System Clock Generator 1 Divider Register
+0010	SYS_CLKGEN2_PLL	R/W	System Clock Generator 2 PLL Register
+0014	SYS_CLKGEN2_DIV	R/W	System Clock Generator 2 Divider Register
+0018	SYS_CLKGEN3_PLL	R/W	System Clock Generator 3 PLL Register
+001C	SYS_CLKGEN3_DIV	R/W	System Clock Generator 3 Divider Register
+0034	SYS_SYSCLK_PREMUX	R/W	System Clock Premux Register
+0038	SYS_AVCLK_MUX	R/W	System AV Clock Mux Register
+003C	SYS_SYSCLK_MUX	R/W	System Clock Mux Register

1. Address refers to G-Bus byte address relative to the clock generator base.
2. Read/Write/Auto update.

Clean Dividers Registers

Table 4-20. **Clean dividers registers**

Address[1]	Register Name	R/W/A[2]	Description
+0080	SYS_CLEANDIV0_DIV	R/W	System Clock Clean Divider 0 Divider Register
+0084	SYS_CLEANDIV0_CTRL	R/W/A	System Clock Clean Divider 0 Control Register
+0088	SYS_CLEANDIV1_DIV	R/W	System Clock Clean Divider 1 Divider Register
+008C	SYS_CLEANDIV1_CTRL	R/W/A	System Clock Clean Divider 1 Control Register
+0090	SYS_CLEANDIV2_DIV	R/W	System Clock Clean Divider 2 Divider Register
+0094	SYS_CLEANDIV2_CTRL	R/W/A	System Clock Clean Divider 2 Control Register
+0098	Reserved		

Table 4-20. **Clean dividers registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+009C	Reserved		
+00A0	SYS_CLEANDIV4_DIV	R/W	System Clock Clean Divider 4 Divider Register
+00A4	SYS_CLEANDIV4_CTRL	R/W/A	System Clock Clean Divider 4 Control Register
+00A8	SYS_CLEANDIV5_DIV	R/W	System Clock Clean Divider 5 Divider Register
+00AC	SYS_CLEANDIV52_CTRL	R/W/A	System Clock Clean Divider 5 Control Register
+00B0	SYS_CLEANDIV6_DIV	R/W	System Clock Clean Divider 6 Divider Register
+00B4	SYS_CLEANDIV6_CTRL	R/W/A	System Clock Clean Divider 6 Control Register
+00B8	SYS_CLEANDIV7_DIV	R/W	System Clock Clean Divider 7 Divider Register
+00BC	SYS_CLEANDIV7_CTRL	R/W/A	System Clock Clean Divider 7 Control Register
+00C0	SYS_CLEANDIV8_DIV	R/W	System Clock Clean Divider 8 Divider Register
+00C4	SYS_CLEANDIV8_CTRL	R/W/A	System Clock Clean Divider 8 Control Register
+00C8	SYS_CLEANDIV9_DIV	R/W	System Clock Clean Divider 9 Divider Register
+00CC	SYS_CLEANDIV9_CTRL	R/W/A	System Clock Clean Divider 9 Control Register
+00D0	SYS_CLEANDIV10_DIV	R/W	System Clock Clean Divider 10 Divider Register
+00D4	SYS_CLEANDIV10_CTRL	R/W/A	System Clock Clean Divider 10 Control Register

1. Address refers to G-Bus byte address relative to the clock generator base.
2. Read/Write/Auto update.

Clock Counter Registers

Table 4-21. **Clock counter registers**

Address[1]	Register Name	R/W/A[2]	Description
+0040	SYS_CLK_CNT	R/W/A	System Clock Counter Register
+0044	SYS_RND_CNT	R/W/A	System Clock Random Generator Counter Register
+0048	SYS_XTAL_IN_CNT	R/W	System XTAL In Counter Register

Table 4-21. **Clock counter registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+004C	SYS_CNT_CFG	R/W	System Clock Counter Configuration Register
+0050	SYS_CFG_CNT0	R/W/A	System Clock Configuration Counter 0 Register
+0054	SYS_CFG_CNT1	R/W/A	System Clock Configuration Counter 1 Register
+0058	SYS_CFG_CNT2	R/W/A	System Clock Configuration Counter 2 Register
+005C	SYS_CFG_CNT3	R/W/A	System Clock Configuration Counter 3 Register
+0060	SYS_CFG_CNT4	R/W/A	System Clock Configuration Counter 4 Register

1. Address refers to G-Bus byte address relative to the clock generator base.
2. Read/Write/Auto update.

Pin Description

Clock Generator Pins

Table 4-22. **Clock generator pin description**

Pin Name	Ball ID	Direction	Description
XTAL_IN	A8	I	27MHz crystal oscillator input.
XTAL_OUT	A7	O	Crystal oscillator output.
XTAL_BUF	A6	O	Buffered crystal oscillator output
VCXO0_IN	B7	I	Input from external VCXO #0
VCXO1_IN	B6	I	Input from external VCXO #1
RCLK0_IN	C7	I	PLL reference clock input #0
RCLK1_IN	C6	I	PLL reference clock input #1
RCLK0_OUT	D6	O	Auxiliary clock output #0
RCLK1_OUT	E1	O	Auxiliary clock output #1
RCLK2_OUT	E2	O	Auxiliary clock output #2

Electrical Characteristics

Clock Generator DC Characteristics

Table 4-23. **Clock generator DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	27	46
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
V_{IH}[2]	Input high voltage	V	2		5.5
V_{IL}[2]	Input low voltage	V	-0.3		0.8

1. Parameter applies to RCLK0_OUT, RCLK1_OUT and RCLK2_OUT.
2. Parameter applies to RCLK0_IN, RCLK1_IN, VCXO0_IN and VCXO1_IN.

5 Host Interface

Block Diagram of Host Interface

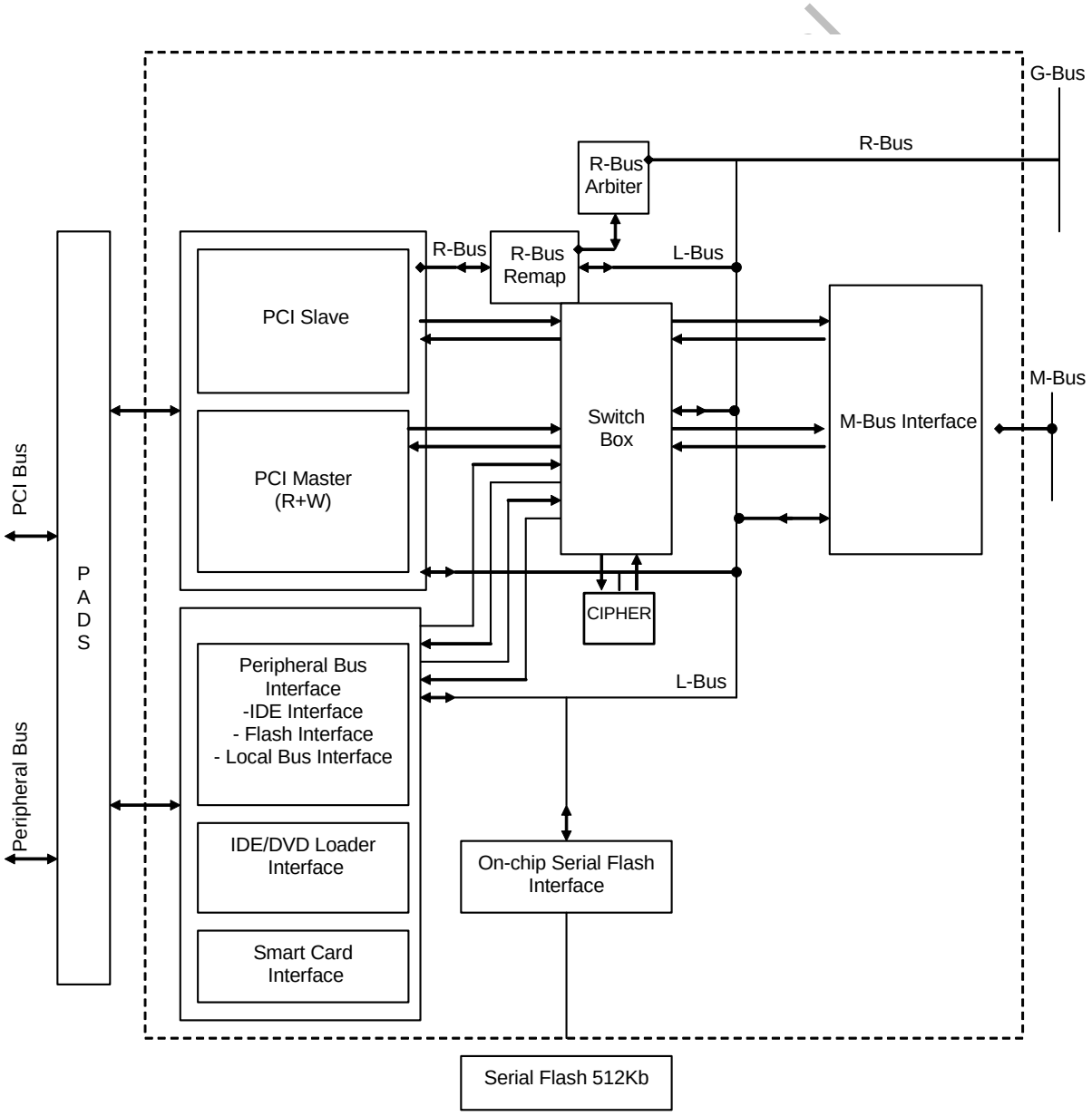


Figure 5-1. **Host interface block diagram**

Introduction

The host interface unit provides the interface between the primary internal buses (G-bus and M-bus) and the PCI, the peripheral/IDE and an external 512Kb serial flash. As a G-Bus slave, the host interface occupies two sections of the G-Bus space: A 64KB section for configuration registers accesses, and a 8MB section for linear access to an external flash memory.

The EM8622L supports both a PCI and a multimode ‘peripheral’ bus for system-level interconnection. The PCI bus implementation is a version 2.1 compliant, 32-bit wide bus capable of operating at 33 or 66MHz. The electrical interface supports both 3.3V and 5V signaling. The interface supports both master and target operation, as well as ACPI power management (v.2.2). On-chip logic optionally implements the PCI bus configuration and arbitration functions (up to four external masters) so that no external PCI ‘host’ device is needed. This allows the design of small, tightly-embedded systems in which the EM8622L hosts the PCI bus.

A separate Peripheral Bus Interface (PBI) is also supported. This interface can operate in several modes with programmable cycle timings which can be varied on a cycle-by-cycle basis. The bus can operate as a general-purpose interface, or as an ‘ISA-like’ bus for connecting external devices, or as an IDE bus for attaching storage devices, or as a memory bus for directly attaching an asynchronous memory such as a parallel flash ROM.

When being used as a general-purpose interface, the PBI can operate in either a separate address and data mode (14 address/16 data) or a multiplexed address/data mode. The timings of all the relevant parameters (address setup, command pulse widths, etc.) are programmable to support the external device requirements. The interface also supports an asynchronous IORDY input for additional timing control.

In the IDE mode, the interface can support ATA/ATAPI-4 device attachments. Two devices (master and slave) may be connected, but only one can be used at any given time; they will not work simultaneously. Both PIO and DMA transfer modes are supported. The IDE interface allows CD, DVD or hard disk drives to be attached directly.

The host interface contains a smart card interface for ISO/IEC7816 standard asynchronous protocols. The cipher block accelerates the AES and DES (3DES) functions.

The EM8622L supports both parallel and serial flash memories. The parallel flash memory is connected via the peripheral bus interface. The serial flash is also connected externally.

PCI Master/Slave/Host Interface

Introduction

The 32-bit PCI master/slave/host interface (33 or 66MHz) is designed for both reads and writes with programmable burst length, and is compliant with PCI v2.1 specifications. It supports 3.3V and 5V operation and ACPI power management from the PCI v2.2 specification. Up to three external PCI devices may be controlled by the EM8622L in the PCI host mode. Content over this interface may be optionally AES, 3DES or DES encrypted/decrypted. To further secure the content when an external host is used, a 'secure PCI' mode can control the regions of the DRAM that the external host CPU can access.

Features

- Supports both reads and writes
- Supports programmable burst length
- Compliant with PCI v2.1 specifications
- PCI host supports up to three external PCI devices
- Supports AES, 3DES or DES encrypted/decrypted content

Functional Description

The host, as a PCI master, can perform G-Bus accesses through the PCI slave interface, the R-Bus and the R-Bus to G-Bus bridge. The entire G-Bus space (128MB) or a portion of it is mapped into the PCI space (configured through startup bits). The R-Bus is a single master (the PCI interface), single slave (the R-Bus to G-Bus bridge) bus.

The PCI slave block contains specific PCI configuration registers, and registers for the DRAM read/write operations. When these registers are accessed, the data is sent to or received from the M-Bus interface through 8-bit ports. The M-Bus interface, programmed by any G-Bus master through the G-Bus to L-Bus bridge and L-Bus will exchange the data with the DRAM controller via the M-Bus.

The PCI master interface (which is also programmed through the L-Bus), can initiate data block moves to/from the host memory to 8-bit ports connected to the M-Bus interface.

An external PCI host can access the G-bus address space via the PCI slave interface. The slave interface also provides the PCI configuration register block and registers for the DRAM read/write operations. This allows for direct SDRAM access through the PCI slave interface. The PCI master interface can initiate block data moves between the SDRAM and the PCI host memory.

The host uses the PCI configuration space registers to auto-detect and configure PCI devices. Prior to the host reading the configuration block, several registers in the block must be initialized. Initialization of these registers is performed by the boot code stored in the external serial flash memory. The boot code establishes the following nominal register values:

Table 5-1. **Default values for PCI register fields**

PCI Configuration Register Field	Default Value (Hex)
VendorID	0x1105
DeviceID	0x8600
Class Code	0x048000
RevisionID	0x01
Subsystem Vendor ID	0x1105
Subsystem Device ID	0x0000

The EM8622L can function as the PCI host. To support this, it includes the PCI bus arbitration logic and 3 REQ#/GNT# pairs. In addition, it adds 3 IDSEL pins to support agent configuration cycle selection. These added pins and capabilities allow up to 3 external PCI masters and/or slave devices to be connected to the EM8622L PCI bus without requiring an external host.

The following figures shows the general connection topology for the arbitration pins and IDSEL pins in the EM8622L:

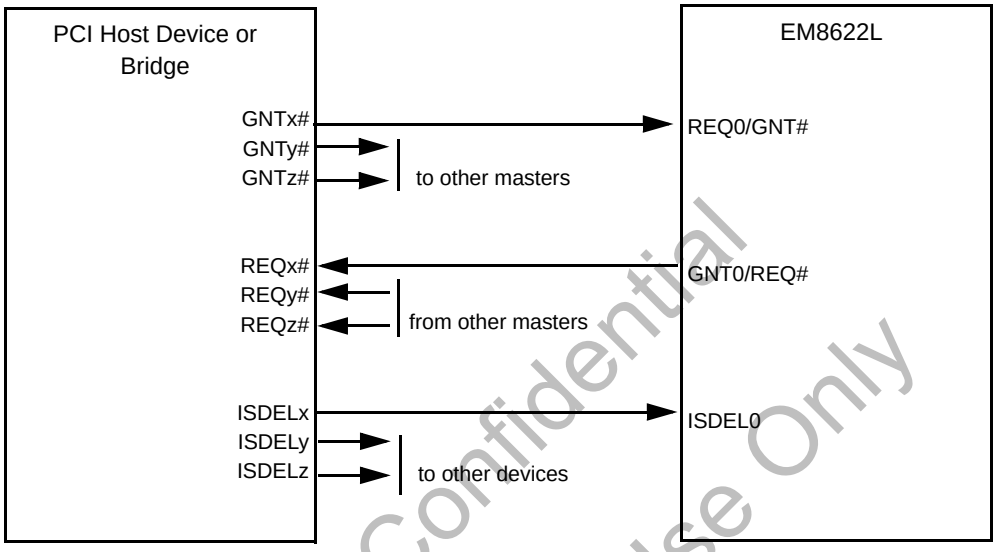


Figure 5-2. **PCI arbitration pin connection diagram (PCI device)**

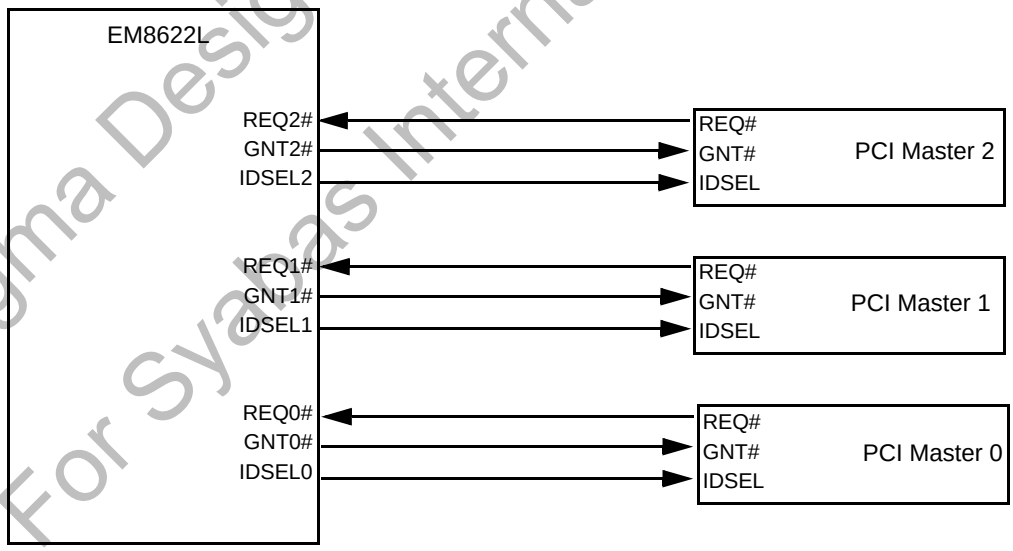


Figure 5-3. **PCI arbitration pin connection diagram (PCI host)**

PCI Slave Access

Direct Access

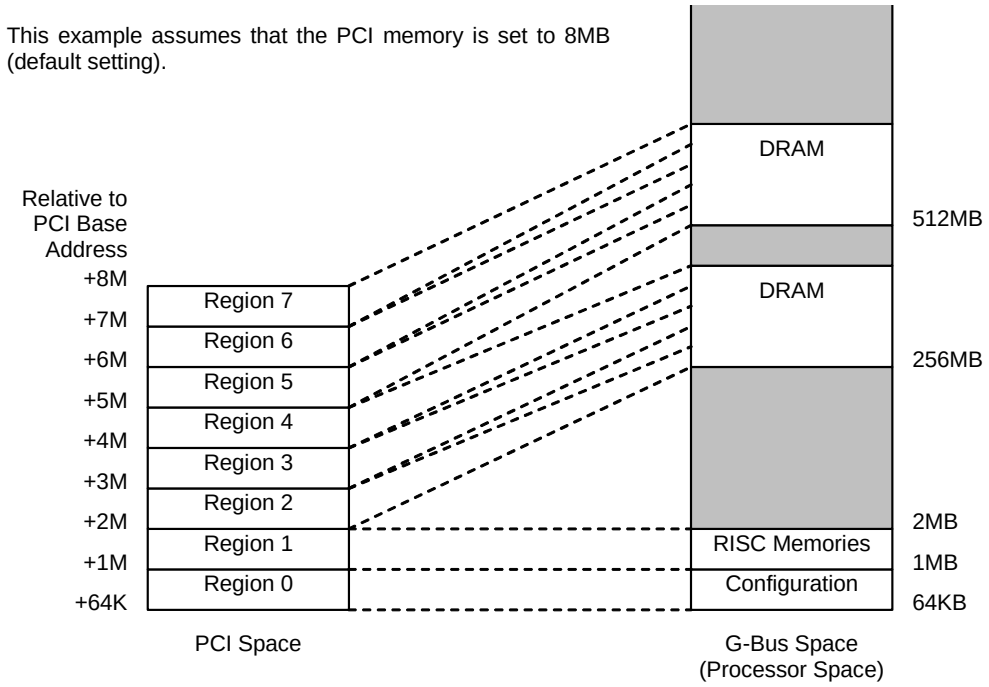


Figure 5-4. **PCI slave direct access (8MB default setting)**

When the EM8622L is used on a PCI expansion board, the host can access all the configuration registers, internal memories and external DDR using the PCI slave access. At boot time, the EM8622L requests between 1MB to 128MB of memory space (default is 8MB). This memory space is split into 8 regions. Each region can be mapped to a specific area of the processor memory space, using the region base address registers. Region 0 is hard-wired to start at address 0. The first 64K of the Region 0 is used for the region base address registers, the interrupt register, the slave DMA access and the time-out control registers, and cannot be used to access the G-Bus.

The region base address registers can also be accessed from the G-Bus, so the CPU can change them. This can be useful when that EM8622L is used as a host, and master-capable devices attached to the EM8622L need access to particular regions of the EM8622L memory.

DMA Access

DMA transfers to the external DDR can be performed using the PCI slave access. Prior to writing or reading any data from the special DMA locations described below, the switch-box and the M-Bus interface should be set up for a DMA transfer. The DMA transfers are best performed using the PCI master access.

Time-out Control

Time-out control is the same for direct access and for the DMA access. In direct access, time-out can occur if the G-Bus is too slow to respond. This is most likely to happen with very slow G-Bus slaves, such as the serial flash controller. In DMA access, time-out can occur if the switchbox is not routing the PCI slave, or if the M-Bus interface is not expecting any data from the switchbox.

PCI Device Configuration

The PCI configuration space registers are used by the host to auto-detect and configure the devices on the bus. Some parameters must be setup before the host reads the registers: Vendor ID, Device ID, Class code, Revision ID, Subsystem vendor ID and Subsystem device ID.

Configuration should be performed by the boot code stored in the external serial flash memory, in less than 2^{24} PCI clock periods. Once the configuration is complete, the boot code will write 1 to the bit VLD of the register HOST_REG2.

The Vendor ID should always be set to 0x1105 (denoting ‘Sigma Designs, Inc.). The top 16 bits of the class code should always be set to 0x0480 (denoting ‘Other Multimedia Device’). The Revision ID shall be incremented for each new tape-out of he EM8622L. The values from the memory size field of register PCI_REG3 are defined as follows:

000	001	010	011	100	101	110	111
1MB	2MB	4MB	8MB	16MB	32MB	64MB	128MB

The other PCI configuration registers are read-only. They can be accessed by writing the number of the register (DWORD address) in the select part of the register PCI_REG3, and the contents of the selected register can be read from the register PCI_CONFIG.

PCI Master Access

Transferring Data from the Host to the EM8622L (Read Transaction)

To transfer a block of data from the host to the EM8622L, the switchbox and the M-Bus interface are programmed with the destination address and size in the EM8622L DDR. Then, the PCI master is programmed with the address in the host memory. In the case of a PC, the address is the physical address of the source data in the memory. The transfer is completed when the register READ_DMA_COUNTER is 0. The software then writes a 0 to the register READ_DMA_ENABLE.

Transferring Data from the EM8622L to the Host (Write Transaction)

To transfer a block of data from the EM8622L to the host, the switchbox and the M-Bus interface are programmed with the source address and size in the EM8622L DDR. Then, the PCI master is programmed with the address in the host memory. In the case of a PC, the address is the physical address of the destination in the memory. The transfer is completed when the register WRITE_DMA_COUNTER is 0. The software then writes a 0 to the register WRITE_DMA_ENABLE.

PCI Host

The PCI host regroups two functionalities:

1. PCI bus arbitration: 2-level priority round-robin arbitration of the bus ownership.
2. PCI host bridge: Low-latency PCI master (through G-Bus) available to the host system.

PCI Arbiter

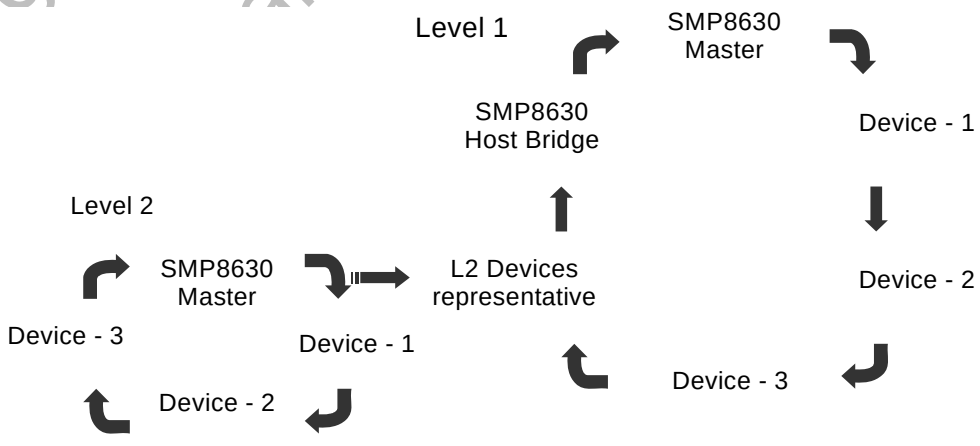


Figure 5-5. **PCI arbiter**

There are 2 loops, Level 1 and Level 2, and 6 agents. Two of them are always part of the Level 1 loop (host bridge and L2 representative) and the 4 others may be part of either loops (exclusive selection). The Level 2 representative is not an agent by itself but one of the agents of the Level 2 loop. It makes requests when any Level 2 agent makes a request.

The arbitration is request based. If requests are asserted, then the bus is granted to the first device requesting in the priority list. The arrows indicate the dynamics of the priority order. In each loop, given a current master agent, the arrow starting from this agent points at the highest priority potential next agent and so on. For example, when all the agents are level 1, and the current master is Device-1 then, the priority order for the next grant is Device-2, Device-3, Host Bridge, EM8622L Master and Device-1.

When no new agent is requesting and the current master is done, the bus is parked on the current master. When the current level 1 master is the L2 representative, the level 2 loop toggles. Each level-versatile agent is attributed a programmable time-out that restricts its control of the bus after the address phase. A broken agent time-out gives 16 clocks to each master to start the address phase once the bus is idle.

A super request mode is available to the host bridge agent. It makes its request higher in the priority in any case. After a super request, the highest priority next agent is the agent next to the one that was ‘interrupted’ by the host.

PCI Host Bridge

The host bridge allows low latency G-Bus controlled master transactions on the PCI bus. Each transaction is a single address phase/single data phase (max one dword at one address). It can be of configuration read/write, IO read/write or a memory read/write access, as defined by the G-Bus address range being used. In the priority scheme, the host is usually like any other agent, unless the super request bit is turned on.

PCI Configuration Access

A direct mapping between the G-Bus address and the PCI address is implemented. It allows the host to perform type0 and type1 configuration accesses on the PCI bus. During a type 0 access, the device number can be programmed within the G-Bus address or through the L-Bus registers. If the L-Bus register agent selection is used, then the bit DEVICE # should be set to 0. Inversely if the G-Bus address agent selection is used, then the ‘agent select’ bits of the register HOST_REG2 should be set to 0.

During the type 1 access, the bit DEVICE # must be programmed through the G-Bus address. It is recommended to use the direct G-Bus address mapping. During an IO access, the G-Bus byte enable bits are mapped to PCI_AD[1:0].

Agent Detection

Upon power up to detect the PCI agents in the system the EM8622L boot software resets the PCI bus. This is accomplished using a GPIO pin. After setting the registers MAMBO_IS_HOST and HOST_REG2, the Device ID and the Vendor ID of the selected device are read at the address of the host memory. If the agent is not present or the data is not ready within the configuration retry cycles, then an interrupt is generated. The register HOST_REG2 contains the interrupt status. At the end of the agent detection process, the software has the Device IDs and Vendor IDs of all the devices in the system. It then has to configure each one of the present devices, before using them.

Once the devices are configured for agent configuration registers, their I/O and memory spaces can be accessed by reading and writing to registers IOSPACE and MEMORYSPACE.

Note: The software must also be careful to map all the devices in the lower 512MB of the PCI memory space, since the EM8622L can only use 512MB on the PCI bus.

Register Map

PCI Slave Registers

PCI Slave Direct Access Registers

Table 5-2. **PCI slave direct access registers**

Address[1]	Register Name	R/W/A[2]	Description
+9000	REGION_0_BASE	R	Region 0 Base Address Register
+9004	REGION_1_BASE	R/W	Region 1 Base Address Register
+9008	REGION_2_BASE	R/W	Region 2 Base Address Register
+900C	REGION_3_BASE	R/W	Region 3 Base Address Register
+9010	REGION_4_BASE	R/W	Region 4 Base Address Register
+9014	REGION_5_BASE	R/W	Region 5 Base Address Register
+9018	REGION_6_BASE	R/W	Region 6 Base Address Register
+901C	REGION_7_BASE	R/W	Region 7 Base Address Register
+9010	PCI_IRQ_STATUS	R/A	PCI Interrupt Status Register
+9014	PCI_IRQ_SET	R/W	PCI Interrupt Set Register
+9018	PCI_IRQ_CLEAR	R/W	PCI Interrupt Clear Register

Table 5-2. **PCI slave direct access registers**

Address[1]	Register Name	R/W/A[2]	Description
+FF88[3]	PREFETCH	R/A	Prefetch Register
+FF8C[3]	DISC-LAT	R/W	Discard Latency Register
+FFFC[3]	ABORT	R/W	Abort Register

1. Address refers to G-Bus byte address relative to the PCI base address.
2. Read/Write/Auto update.
3. Address refers to G-Bus byte address relative to the host register base address.

PCI Slave Time-out Control Registers

Table 5-3. **PCI slave time-out control registers**

Address[1]	Register Name	R/W/A[2]	Description
+8000	TIMEOUT_VALUE	R/W	Time-out Value Register
+8004	TIMEOUT_STATUS	R/A	Time-out Status Register
+8008	TIMER_COUNTER	R/W/A	Timer Counter Register
+800C	TIMER_TEST_REGISTER	W	Timer Test Register
+8010	WAKEUP_REGISTER	W	Wake Up Register

1. Address refers to G-Bus byte address relative to the PCI base address.
2. Read/Write/Auto update.

PCI Slave Device Configuration Registers

Table 5-4. **PCI slave device configuration registers**

Address[1]	Register Name	R/W/A[2]	Description
+FED4	HOST_REG2	R/W	Host Region 2 Register
+FEE8	PCI_REG0	R/W	PCI Region 0 Register
+FEEC	PCI_REG1	R/W	PCI Region 1 Register
+FEF0	PCI_REG2	R/W	PCI Region 2 Register
+FEF4	PCI_REG3	R/W	PCI Region 3 Register
+FEF8	PCI_CONFIG	R/A	PCI Configuration Register

1. Address refers to G-Bus byte address relative to the PCI base address.
2. Read/Write/Auto update.

PCI Master Access Registers

Table 5-5. **PCI master access registers**

Address[1]	Register Name	R/W/A[2]	Description
+FEC0	READ_DMA_ADDRESS	R/W/A	Read DMA Address Register
+FEC4	READ_DMA_COUNTER	R/W/A	Read DMA Counter Register
+FEC8	READ_DMA_ENABLE	R/W	Read DMA Enable Register
+FECC	DMA_REV_ORDER	R/W	DMA Revision Order Register
+FED8	WRITE_DMA_ADDRESS	R/W/A	Write DMA Address Register
+FEDC	WRITE_DMA_COUNTER	R/W/A	Write DMA Counter Register
+FEE0	WRITE_DMA_ENABLE	R/W	Write DMA Enable Register
+FEE4	DMA_BURST	R/W	DMA Burst Register

1. Address refers to G-Bus byte address relative to the host register base address.
2. Read/Write/Auto update.

PCI Host Registers

Table 5-6. **PCI host registers**

Address[1]	Register Name	R/W/A[2]	Description
+FE90	MAMBO_IS_HOST	R/W	Mambo Is Host Register
+FED0	HOST_REG1	R/W	Host Region1 Register
+FED4	HOST_REG2	R/W	Host Region2 Register
+FE80	HOST_REG3	R/W	Host Region3 Register
+FE84	HOST_REG4	R/W	Host Region4 Register
+FE94	HOST_REG5	R/W	Host Region5 Register
+1000_0000-+10FF_FFFC	CONFIGURATION	R/W	Configuration Register
+1800_0000-+2000_0000	I/O SPACE	R/W	I/O Space Register
+2000_0000-+4000_0000	MEMORY SPACE	R/W	Memory Space Register

1. Address refers to G-Bus byte address relative to the host register base address.
2. Read/Write/Auto update.

Pin Descriptions

PCI Pins

Table 5-7. **System bus pin descriptions**

Pin name	Ball ID	Direction	Description
PCI_AD31	AC5	B	PCI address/data pin 31 (MSB). The address and the data are multiplexed on the AD pins during the memory and the I/O operations on the PCI bus.
PCI_AD30	AB5	B	PCI address/data pin 30
PCI_AD29	AA5	B	PCI address/data pin 29
PCI_AD28	Y5	B	PCI address/data pin 28
PCI_AD27	AC6	B	PCI address/data pin 27
PCI_AD26	AB6	B	PCI address/data pin 26
PCI_AD25	AA6	B	PCI address/data pin 25
PCI_AD24	Y6	B	PCI address/data pin 24
PCI_AD23	W7	B	PCI address/data pin 23
PCI_AD22	AC8	B	PCI address/data pin 22
PCI_AD21	AB8	B	PCI address/data pin 21
PCI_AD20	AA8	B	PCI address/data pin 20
PCI_AD19	Y8	B	PCI address/data pin 19
PCI_AD18	W8	B	PCI address/data pin 18
PCI_AD17	AC9	B	PCI address/data pin 17
PCI_AD16	AB9	B	PCI address/data pin 16
PCI_AD15	AB11	B	PCI address/data pin 15
PCI_AD14	AA11	B	PCI address/data pin 14
PCI_AD13	Y11	B	PCI address/data pin 13
PCI_AD12	W11	B	PCI address/data pin 12
PCI_AD11	AC12	B	PCI address/data pin 11
PCI_AD10	AB12	B	PCI address/data pin 10
PCI_AD9	AA12	B	PCI address/data pin 9
PCI_AD8	Y12	B	PCI address/data pin 8
PCI_AD7	AB13	B	PCI address/data pin 7
PCI_AD6	AA13	B	PCI address/data pin 6

Table 5-7. **System bus pin descriptions (Continued)**

Pin name	Ball ID	Direction	Description
PCI_AD5	Y13	B	PCI address/data pin 5
PCI_AD4	W13	B	PCI address/data pin 4
PCI_AD3	AC14	B	PCI address/data pin 3
PCI_AD2	AB14	B	PCI address/data pin 2
PCI_AD1	AA14	B	PCI address/data pin 1
PCI_AD0	Y14	B	PCI address/data pin 0 (LSB)
PCI_CBE3#	AC7	B	Command/byte enable pin 3. During the address phase of a transaction the CBE(3:0)# defines the PCI command. During the data phase each signal indicates whether the associated data byte will be transferred. The CBE3# applies to PCI_AD(31:24).
PCI_CBE2#	AA9	B	Command/byte enable pin 2. Applies to PCI_AD(23:16).
PCI_CBE1#	AC11	B	Command/byte enable pin 1. Applies to PCI_AD(15:8).
PCI_CBE0#	AC13	B	Command/byte enable pin 0. Applies to PCI_AD(7:0).
PCI_FRAME#	Y9	B	Cycle frame pin. Current initiator asserts the FRAME# pin to indicate the start and duration of a transaction.
PCI_DEVSEL#	AB10	B	Device select pin. A target asserts DEVSEL# when it decodes its address.
PCI_TRDY#	AC10	B	Target ready pin. The currently addressed target asserts TRDY# to indicate that it is ready to complete a transaction.
PCI_IRDY#	W9	B	Initiator ready pin. The current bus master asserts IRDY# to indicate that it is ready to complete a transaction.
PCI_PAR	Y10	B	Parity. Driven by an initiator (write) or currently-addressed target (read) to create even parity across AD(31:0) and CBE(3:0)#.
PCI_CLK	AC3	I	Clock input for PCI interface section. Either 33MHz or a 66MHz clock signal.
PCI_INTA#	AB3	B	Interrupt A pin. Asserted by a PCI agent to request an interrupt.
PCI_STOP#	AA10	O	Stop pin. Asserted by an addressed target to request the bus master to terminate the current transaction in progress.
PCI_REQ2#	W5	I	PCI host mode: Bus Request for master #2. Asserted low by external master requesting PCI bus transaction. PCI Device Mode: Unused, make no connection.

Table 5-7. **System bus pin descriptions (Continued)**

Pin name	Ball ID	Direction	Description
PCI_REQ1#	W4	I	PCI Host Mode: Bus Request for Master #1. Asserted low by external master requesting PCI bus transaction. PCI Device Mode: Unused, make no connection.
PCI_REQ0#/GNT#	Y4	I	PCI host mode: Bus request for master #0. Asserted low by an external master requesting a PCI bus transaction. PCI device mode: Bus grant input from an external PCI host.
PCI_GNT2#	AA4	O	PCI host mode: Bus grant for master #2. Asserted low to grant the PCI ownership to an external master. PCI device mode: Unused, make no connection.
PCI_GNT1#	AB4	O	PCI host mode: Bus grant for master #1. Asserted low to grant the PCI ownership to an external master. PCI device mode: Unused, make no connection.
PCI_GNT0#/REQ#	AC4	O	PCI host mode: Bus grant for master #0. Asserted low to grant the PCI ownership to an external master. PCI device mode: Bus request output to an external PCI host.
PCI_IDSEL2	Y7	O	PCI host mode: ID Select for the PCI device #2. Asserted low by the EM8622L to indicate a configuration cycle to a PCI device. PCI device mode: Unused, make no connection.
PCI_IDSEL1	AA7	O	PCI host mode: ID select for PCI device #1. Asserted low by the EM8622L to indicate a configuration cycle to a PCI device. PCI Device Mode: Unused, make no connection.
PCI_IDSEL0	AB7	B	PCI host mode: ID select for PCI device #0. Asserted low by the EM8622L to indicate a configuration cycle to a PCI device. PCI device mode: Asserted low by an external PCI host to indicate a configuration cycle for the EM8622L.

Electrical Characteristics

PCI DC Characteristics

Table 5-8. **PCI interface DC characteristics**

Symbol	Parameter	Units	Min	Max
V_{IH}	Input high voltage	V	0.5xVDD_3V3	5.5
V_{IL}	Input low voltage	V	-0.3	0.3xVDD_3V3
I_{IL}	Input leakage (condition: 0 < VIN < 3.3V)	uA		± 10
V_{OH}	Output high voltage	V	0.9xVDD_3V3	
V_{OL}	Output low voltage	V		0.1xVDD_3V3
C_{IN}	Input pin capacitance	pF		10
C_{CLK}	CLK pin capacitance	pF	5	10
C_{IDSEL}	IDSEL pin capacitance	pF		8
L_{PIN}	Pin inductance	nH		20

PCI AC Characteristics

The EM8622L PCI I/O buffers fully conform to the electrical characteristics specified in the sections '*Electrical Specification*' and '*66MHz PCI Specification*' of the PCI Local Bus Specification, Revision 2.2. The PCI bus specification provides a comprehensive set of AC characteristics which compliant devices must meet. These characteristics are specified in terms of V/I curves, output slew rates, and detailed measurement procedures, and are not reproduced in this document. The I/O cells used in the EM8622L provide compliance with these specifications by design and characterization.

Timing Parameters

Certain PCI bus timing parameters have different values depending on whether the signal is ‘bused’ or ‘point-to-point’. The point-to-point signals are the REQ# and GNT# signals. The timing parameters which differ between these two groups are identified as ‘BUS’ or ‘PTP’ in the table below. For exact measurement conditions for all the timing parameters, refer to the PCI Local Bus Specification, Revision 2.2, Section 7.6.4.3.

Table 5-9. **PCI timing parameters**

Symbol	Parameter	Units	Min	Max
T_{CYC}	PCI_CLK cycle time (measured at 0.4xVDD_3V3 level)	ns	15	
T_{HIGH}	PCI_CLK high time (measured at 0.5xVDD_3V3 level)	ns	6	
T_{LOW}	PCI_CLK low time (measured at 0.3xVDD_3V3 level)	ns	6	
T_{VAL}	CLK to Signal Valid Delay (BUS)	ns	2	6
T_{VAL}	CLK to Signal Valid Delay (PTP)	ns	2	6
T_{ON}	Float to Active Delay	ns	2	
T_{OFF}	Active to Float Delay	ns		14
T_{SU}	Input Set Time to CLK (BUS)	ns	3	
T_{SU}	Input Set Time to CLK (PTP)	ns	5	
T_{H}	Input Hold Time from CLK	ns	0	

Timing Diagrams

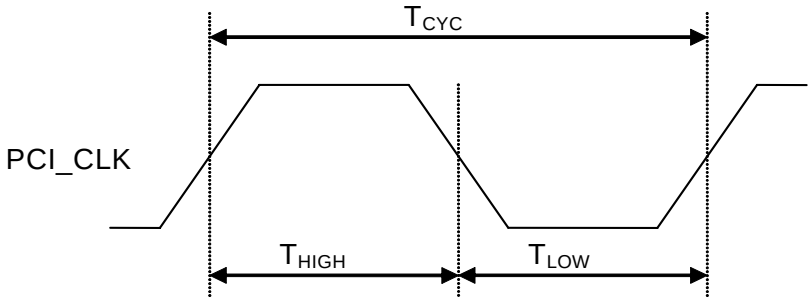


Figure 5-6. **PCI CLK timing diagram**

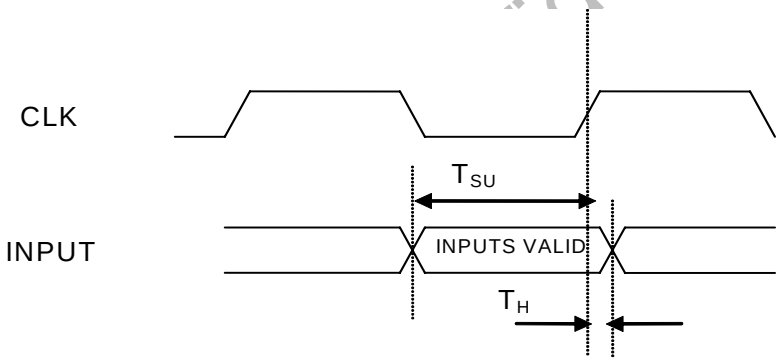


Figure 5-7. **PCI input timing diagram**

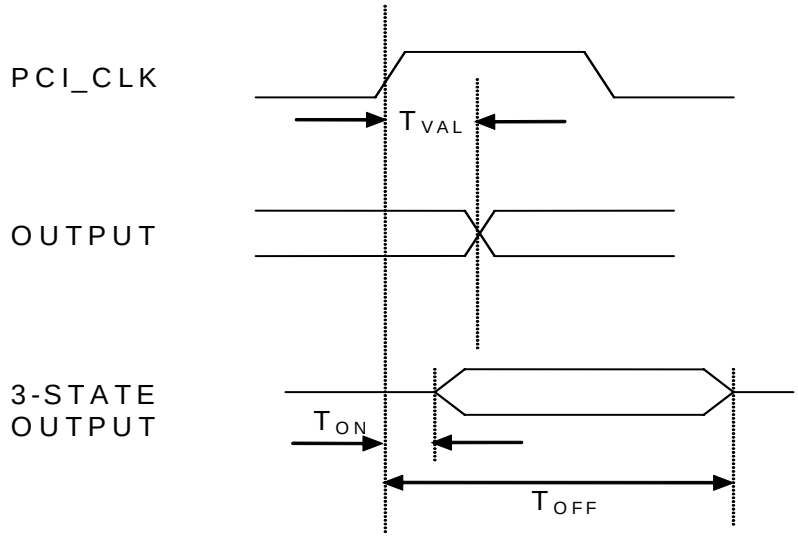


Figure 5-8. **PCI output timing diagram**

Peripheral Bus Interface (PBI)

Introduction

The EM8622L has an external Peripheral Bus Interface (PBI), which supports multiple protocols based on asynchronous (ISA like) transactions. This bus can be used to connect slave devices such as:

- One or two IDE hard drives (CD-ROM, DVD drives)
- Parallel flash memory (up to 64MB per chip select)
- ISA compatible chips (Ethernet, MPEG2 encoder, etc.)

The PBI is basically a bridge between the G-Bus and the PB. It is the only master on the PB. A DMA engine performs repetitive accesses on the PB and transfers the data to/from the DRAM. The DMA engine is compatible with the IDE DMA mode.

IDE Interface

The IDE (ATA/ATAPI-4) interface enables CD or DVD drives, hard disc drives (HDD), CompactFlash or memory stick readers to be easily incorporated into a system. It supports both PIO and DMA. Two IDE devices may be connected to the IDE port, although only one at a time may be used. The content over this interface may be optionally AES, 3DES or DES encrypted/decrypted.

Flash Interface

The 8-bit parallel flash interface supports up to 16MB (per chip select) of NOR Flash memory with 24 address bits. The flash content can be optionally AES, 3DES or DES encrypted.

Local Bus Interface

The Local Bus interface supports either 16 data bits and 14 address bits, or 16 multiplexed data/address bits. This interface may be used to connect to an external MPEG encoder chip or 802.11a/b/g chip.The content over this interface may be optionally AES, 3DES or DES encrypted/decrypted.

Features

- Supports a wide range of external devices like Ethernet chip sets, MPEG encoders, 802.11 chips and other external devices
- Supports up to 4 devices
- Capable of handling single register read/write transactions from/to a slave device
- DMA transfers (FIFO read/write) between an external chip and the host memory or the MPEG decoder memory are possible
- Two FIFO read ports can be used for video and audio
- Supports a write FIFO to send data to an external chip set

Block Diagram

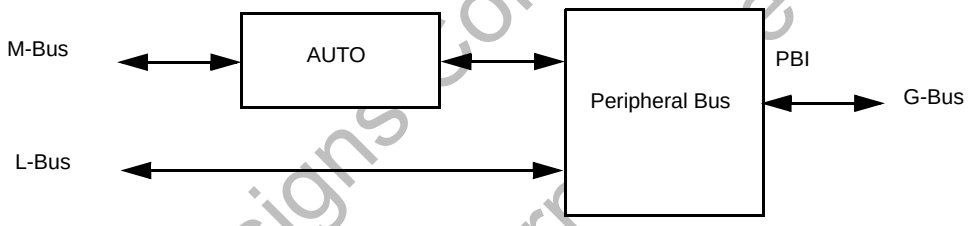


Figure 5-9. **Peripheral Bus Interface (PBI) block diagram**

Functional Description

The peripheral bus interface allows Local Bus peripherals, IDE peripherals, Compact-Flash, and parallel flash memory to be connected to the device using a single bus. There are four chip selects available to share between the flash memory and the local bus peripherals.

The Peripheral Bus Interface (PBI) in the EM8622L can operate in several modes with programmable cycle timings which can be varied on a cycle-by-cycle basis. The bus can operate as a general-purpose, ‘ISA-like’ local bus for connecting external devices, an IDE bus for attaching storage devices, or as a memory bus for attaching asynchronous parallel flash memory.

The master-only bus can be used to connect slave devices such as:

- IDE hard disk drives, CD-ROM drives and DVD drives
- Supports direct attachment of 8-bit or 16-bit NOR type parallel flash memory (up to 64MB per chip select)

- Supports ISA compatible devices (Ethernet controller, MPEG encoder, etc.)

When being used as a general-purpose interface, the PBI can operate in either a separate address and data mode (14 address/16 data local bus) or a multiplexed address/data mode. The timings of all the relevant parameters (address setup, command pulse widths, etc.) are programmable to support the external device requirements. The interface also supports an asynchronous IORDY input for additional timing control.

When being used as an IDE interface, the IDE controller interfaces one or two external IDE devices and DDR memory. It provides a simple, standard interface to mass storage peripherals. The IDE interface is a common feature of consumer-level DVD and HDD drives; the EM8622L provides a direct interface to these drives. It supports a parallel flash memory connected via the PBI. In the IDE mode, the interface can support ATA device attachments. Two devices (master and slave) may be connected, but only one can be used at any given time; they will not work simultaneously. Both PIO and DMA transfer modes are supported. The IDE interface allows CD, DVD or hard disk drives to be directly connected.

When used to connect an external asynchronous memory, the PBI supports an 8-bit data bus width and 24-bit addressing (16MB).

PBI address mapping

Besides configuration registers, the PBI occupies two areas in the G-Bus space:

- A 2048-byte register area. The register area consists of four 512-byte sections. Depending on which section is accessed, one of the four chip selects (PB_CS#[3:0]) is activated. For example, a G-Bus access to address 2_0600 activates PB_CS#[3].
- A 256MB memory area. The memory also consists of four (64MB) sections, each activating one PB_CS#[3:0] line.

G-Bus accesses to these areas are translated into the PB transactions by the PBI. For each section of the memory area, the user must configure 4 parameters through the register CS_CONFIG, the PB data width (8/16-bit), the multiplexed/non multiplexed address data, the packed/unpacked and the IDE/non IDE parameters.

The first two parameters indicate the mode (as described in the PBI signals paragraph). The packed/unpacked flag dictates the behavior of the PBI when the G-Bus transaction width does not match the P-Bus data width. The G-Bus is 32-bit wide, and supports byte, word and dword transactions, with word and dword transactions respectively aligned on an address multiple of 2 and 4.

When the packed flag is set, a G-Bus transaction may generate more than one PB transaction, according to the following table. The base indicates the G-Bus address of the concerned section of the memory area.

Table 5-10. **PBI data transactions**

G-Bus Transaction Type	PB Data Width = 8	PB Data Width = 16
byte	1 PB transaction PB address = G-Bus byte address - Base	No transaction generated on the PB
word	2 PB transactions 1st PB address = G-Bus byte address - Base 2nd PB address = 1st PB address + 1	1 PB transaction PB address = (G-Bus byte address - Base)/2
dword	4 PB transactions 1st PB address = G-Bus byte address - Base 2nd PB address = 1st PB address + 1 3rd PB address = 1st PB address + 2 4th PB address = 1st PB address + 3	2 PB transactions 1st PB address = (G-Bus byte addr - Base)/2 2nd PB address = 1st PB address + 1

When the packed flag is cleared, only the dword G-Bus transactions are allowed (the G-Bus byte address being a multiple of 4), and each G-Bus transaction generates one PB transaction.

If the PB data width = 8, then the PB data is the LSB of the G-Bus data. During the write transactions, the 3 MSB of the G-Bus data are discarded and during the read transactions, they read 0.

If the PB data width = 16, then the PB data is the least significant word of the G-Bus data. During the write transactions, the most significant word of the G-Bus data is discarded and during the read transactions, it reads 0. In both the cases, the PB address = (G-Bus byte addr – Base) / 4.

The four sections of the register area use the same parameters as their memory area counterparts, with the exception that the accesses are always unpacked regardless of the pack/unpack configuration bit.

PB Registers

The PB timing registers define the 7 timing sets (see timing diagrams for a description of T_A, T_B, T_C and T_D). The PB use timing registers are registers are associated with the PB timing registers, and define for which type of transaction each timing set is used. When a PB transaction occurs, the PBI checks if the type of the transaction matches the description in the register PB_USE_TIMING0. If so, then the register PB_TIMING0 is used. Otherwise, the register PB_USE_TIMING1 is checked, and so on. If no match occurs, then the register PB_DEFAULT timing is used.

The automode registers are used to setup a DMA transfer between the PB and the EM8622L DRAM. Some applications using the PB DMA are,

- IDE device: To transfer the data sectors between an IDE device (hard disk, CD ROM) and the DRAM
- Flash memory: To load the data blocks from the flash memory to the DRAM.
- External MPEG encoder: To read the compressed bitstream from an MPEG encoder into the DRAM.

The data is transferred through two dedicated 8-bit ports (one for each direction) of the PBI, the host interface switchbox and the M-Bus interface. On the PBI side, a DMA transfer is setup using the two automode registers. The automode control register indicates the number of PB accesses to perform. In the 8-bit mode, count bytes are transferred. In the 16-bit mode, count words (2 x count bytes) are transferred. The direction bit gives the direction of the data transfer between the DRAM and the PB.

PBI Timing

Non Multiplexed Mode

The PBI transactions in both multiplexed and the non-multiplexed modes are generated in distinct phases. The phases are timed using programmed numbers of system clock periods to define each interval. In the diagrams below, the T_A, T_B, T_C and T_D parameters represent the programmable values. A separate set of timing parameters is associated with each of the four PB_CSBx outputs; four different sets of cycle timings can be defined and selectively issued to the attached devices. The asynchronously-sampled IORDY signal can be deasserted by an addressed device to extend the transaction

.

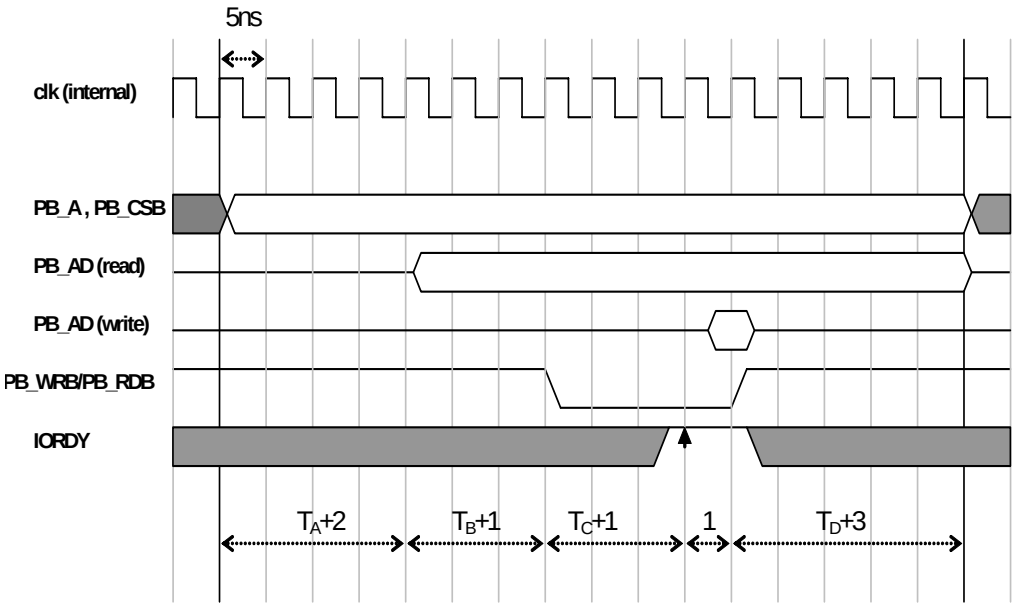


Figure 5-10. **Functional timing diagram: non-multiplexed modes (default parameters)**

A non-multiplexed (address data) PB transaction is configured through a set of 4 timing parameter: T_A, T_B, T_C and T_D. In the above diagram $T_A = T_B = T_C = T_D = 2$ (default value). Also, during a read transaction, PB_WR# remains high. During a write transaction, PB_RD# remains high.

For a non-multiplexed PBI transaction:

1. At the beginning of the transaction, PB_A and one of the four PB_CS# signals are asserted.
2. T_A+2 clock cycles later, the data is driven on PB_AD if the transaction is a write.
3. T_B+1 clock cycles later, PB_WR# or PB_RD# are asserted respectively for a write or read transaction.
4. T_C+1 clock cycles later, PB_IORDY# is sampled. The sampling of PB_IORDY# will continue until it is high.
5. One clock cycle later, PB_WR# (or PB_RD#) is de-asserted. In the case of a read transaction, the read data is sampled at this point.
6. T_D+3 clock cycles later, the transaction terminates. The PB_A and PB_CS# may change if a new transaction follows.

Multiplexed Mode

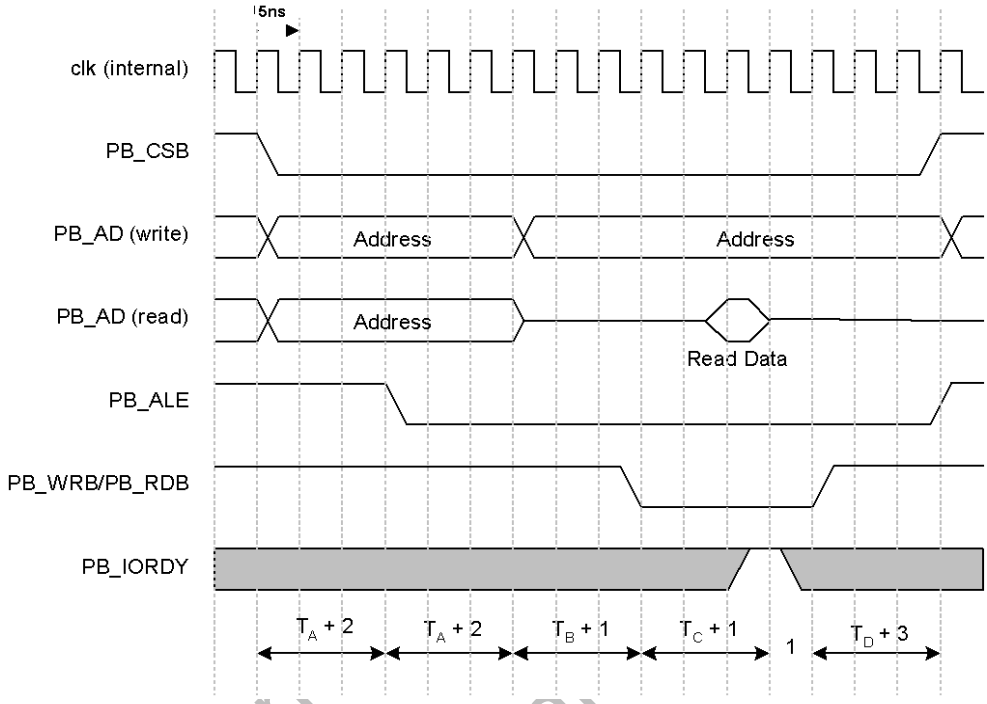


Figure 5-11. **Functional timing diagram: multiplexed modes**

For a multiplexed PBI transaction:

1. At the beginning of the transaction, PB_AD is driven with address and one of the four PB_CS# signals are asserted.
2. T_A+2 clock cycles later, PB_ALE is taken active (low).
3. T_A+2 clock cycles later, PB_AD is driven with the write data for a write transaction, or floated for a read transaction.
4. T_B+1 clock cycles later, PB_WR# or PB_RD# are asserted respectively, for a write or a read transaction.
5. T_C+1 clock cycles later, PB_IORDY# is sampled. The sampling of PB_IORDY# continues until it is high.
6. One clock cycle later, the asserted command is de-asserted. For a read transaction, the read data is sampled at this point.
7. T_D+3 clock cycles later, the transaction terminates. The PB_AD and PB_CS# may change if a new transaction follows.

PBI Signals

The PBI supports 4 address/data modes:

- Mode 8/24: 8-bit data/24-bit address, non multiplexed. Typically used with an 8bit flash memory (up to 16MB).
- Mode 16/16: 16-bit data/16-bit address, non multiplexed. Typically used with an IDE device (Note: IDE devices require only 3 address bits)
- Mode 16/16mux: 16-bit data/16-bit address, multiplexed. Typically used with an external MPEG2 encoder (up to 128KB).
- Mode 16/24mux: 16-bit data/24-bit address, multiplexed (up to 32MB).

The following table shows the PB signals, with the functionality of the PB_A[15:0] and PB_AD[15:0] pins for each mode. Please note that the PB mode can change on a cycle-to-cycle basis, allowing disparate devices to be simultaneously connected.

Table 5-11. **PBI address/data pin descriptions**

Mode	**8/24**		**16/16**		**16/16 mux**		**16/24 mux**	
Signal	**Dir**	**Function**	**Dir**	**Function**	**Dir**	**Function**	**Dir**	**Function**
PB_A[15:8]	O	addr[23:8]	O	addr[15:8]	O	Not used	O	addr [23:16]
PB_A[7:0]	O	addr[15:8]	O	addr[7:0]	O	Not used	O	addr [15:8]
PB_AD[15:8]	B	data[7:0]	B	data[15:8]	B	addr/ data[15:8]	B	data[15:8]
PB_AD[7:0]	O	addr[7:0]	B	data[7:0]	B	addr/ data[7:0]	B	addr/ data[7:0]

Other pins associated with the PBI and their descriptions are as follows:

Table 5-12. **PBI control pin descriptions**

Signal	Direction	Description
PB_CS#[3:0]	O	Four chip selects (active low)
PB_RD#	O	Read strobe (active low)
PB_WR#	O	Write strobe (active low)
PB_IORDY#	I	Ready - An accessed device can extend the PB transaction by pulling this line low through an open collector.
PB_ALE	O	Address latch enable. In multiplexed address/data mode, this signal indicates when to latch the address.
PB_DIR#	O	Indicates the direction of the current transaction (may be used to control external transceivers).
PB_DMAREQ	I	An external IDE device can request a DMA transfer using this signal
PB_DMAACK	O	DMA acknowledge

Register Map

PBI Configuration Registers

Table 5-13. **PBI configuration registers**

Address[1]	Register Name	R/W/A[2]	Description
+0800	PB_TIMING0	R/W	Peripheral Bus Timing 0 Register
+0804	PB_TIMING1	R/W	Peripheral Bus Timing 1 Register
+0808	PB_TIMING2	R/W	Peripheral Bus Timing 2 Register
+080C	PB_TIMING3	R/W	Peripheral Bus Timing 3 Register
+0810	PB_TIMING4	R/W	Peripheral Bus Timing 4 Register
+0814	PB_TIMING5	R/W	Peripheral Bus Timing 5 Register
+0818	PB_DEFAULT_TIMING	R	Peripheral Bus Default Timing Register
+081C	PB_USE_TIMING0	R/W	Peripheral Bus Use Timing 0 Register
+0820	PB_USE_TIMING1	R/W	Peripheral Bus Use Timing 1 Register
+0824	PB_USE_TIMING2	R/W	Peripheral Bus Use Timing 2 Register
+0828	PB_USE_TIMING3	R/W	Peripheral Bus Use Timing 3 Register
+082C	PB_USE_TIMING4	R/W	Peripheral Bus Use Timing 4 Register
+0830	PB_USE_TIMING5	R/W	Peripheral Bus Use Timing 5 Register

Table 5-13. **PBI configuration registers**

Address[1]	Register Name	R/W/A[2]	Description
+0834	PB_CS_CONFIG	R/W	Peripheral Bus CS Configuration Register

1. Address refers to G-Bus byte address relative to the host interface register base.
2. Read/Write/Auto update.

PBI Automode Registers

Table 5-14. **PBI automode registers**

Address[1]	Register Name	R/W/A[2]	Description
+0840	PB_AUTOMODE_START_ADDRESS	R/W	Peripheral Bus Automode Start Address Register
+0844	PB_AUTOMODE_CONTROL	R/W	Peripheral Bus Automode Control Register

1. Address refers to G-Bus byte address relative to the host interface register base.
2. Read/Write/Auto update.

IDE/Flash Interface Registers

Table 5-15. **IDE/Flash interface registers**

Address[1]	Register Name	R/W/A[2]	Description
+0000	IDE_DATA	R/W	IDE Data Register
+0004	IDE_ERROR	R/W	IDE Error Register
+0008	IDE_COUNT	R/W	IDE Counter Register
+000C	IDE_START_SECTOR	R/W	IDE Start Sector Register
+0010	IDE_CYLINDER_LO	R/W	IDE Cylinder Low Register
+0014	IDE_CYLINDER_HI	R/W	IDE Cylinder High Register
+0018	IDE_HEAD_DEVICE	R/W	IDE Head Device Register
+001C	IDE_CMD_STAT	R/W	IDE CMD Status Register
+0200	Reserved		
+0204	Reserved		
+0208	Reserved		
+020C	Reserved		
+0210	Reserved		
+0214	Reserved		

Table 5-15. **IDE/Flash interface registers**

Address[1]	Register Name	R/W/A[2]	Description
+0218	IDE_IRQ_STAT	R/W	IDE Interrupt Status Register
+021C	IDE_CMD_STAT	R/W	IDE CMD Status Register

1. Address refers to G-Bus byte address relative to the host interface register base.
2. Read/Write/Auto update.

Pin Description

PBI Pins

Table 5-16. **PBI pin descriptions**

Pin Name	Ball ID	Direction	Description
PB_AD15	Y18	B	PBI multiplexed address/data bit 15 (MSB). Used as multiplexed address and data bus for 16/16mux and 24/8 mux modes. Used as a data bus in 24/8 and 16/16 modes. AD(7:0) forms low-order byte of address in 24/8 mode.
PB_AD14	AB20	B	Multiplexed address/data bit 14
PB_AD13	Y19	B	Multiplexed address/data bit 13
PB_AD12	Y16	B	Multiplexed address/data bit 12
PB_AD11	AA20	B	Multiplexed address/data bit 11
PB_AD10	AC22	B	Multiplexed address/data bit 10
PB_AD9	W17	B	Multiplexed address/data bit 9
PB_AD8	AB21	B	Multiplexed address/data bit 8
PB_AD7	AC23	B	Multiplexed address/data bit 7
PB_AD6	AB22	B	Multiplexed address/data bit 6
PB_AD5	AB23	B	Multiplexed address/data bit 5
PB_AD4	Y20	B	Multiplexed address/data bit 4
PB_AD3	AA21	B	Multiplexed address/data bit 3
PB_AD2	W19	B	Multiplexed address/data bit 2
PB_AD1	Y21	B	Multiplexed address/data bit 1
PB_AD0	AA22	B	Multiplexed address/data bit 0 (LSB)
PB_A15	AC16	O	PBI address bit 15 (MSB). Provides address for non-multiplexed bus cycles.
PB_A14	AC17	O	PBI address bit 14

Table 5-16. **PBI pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
PB_A13	AB16	O	PBI address bit 13
PB_A12	AA16	O	PBI address bit 12
PB_A11	AB17	O	PBI address bit 11
PB_A10	AA17	O	PBI address bit 10
PB_A9	AC18	O	PBI address bit 9
PB_A8	W15	O	PBI address bit 8
PB_A7	AC19	O	PBI address bit 7
PB_A6	AA18	O	PBI address bit 6
PB_A5	AB18	O	PBI address bit 5
PB_A4	AC20	O	PBI address bit 4
PB_A3	AC21	O	PBI address bit 3
PB_A2	Y17	O	PBI address bit 2
PB_A1	AB19	O	PBI address bit 1
PB_A0	AA19	O	PBI address bit 0 (LSB)
PB_CS3#	AA23	O	Chip select #3 (active low). Pins PB_CSB3:0 become active for accesses to specified regions of the PB memory space.
PB_CS2#	V19	O	Chip select #2 (active low)
PB_CS1#	W20	O	Chip select #1 (active low)
PB_CS0#	Y22	O	Chip select #0 (active low)
PB_ALE	Y23	O	Address latch enable
PB_DMARQ	W23	I	DMA request
PB_DMAACK#	V22	O	DMA acknowledge
PB_DIR#	V21	O	Transfer direction (low = read, high = write)
PB_IORDY	W22	I	Device ready. A device can extend a transfer cycle by pulling this line low prior to the IORDY sampling point, and holding it low until the device is ready to complete the cycle.
PB_RD#	V20	O	Read command (active low)
PB_WR#	W21	O	Write command (active low)

Electrical Characteristics

PBI DC Characteristics

Table 5-17. **PBI DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}(1)	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
V_{IH}[2]	Input high voltage	V	2.0		5.5
V_{IL}(2)	Input low voltage	V	-0.3		0.8

1. Parameter applies to all outputs and bi-directional pins when driving.
2. Parameter applies to all inputs and bidirectional pins when receiving.

PBI AC Characteristics

Table 5-18. **PBI AC characteristics**

Symbol	Description	Units	Min	Typ	Max

Timing Diagrams

AC Timings

The PBI can operate in several modes, and allows interaction with a variety of external devices. The PBI controller is clocked by the system clock (166MHz nominal), but programmable parameters allow custom adjustments of the actual bus timings.

All the PBI output timings are synchronized to the internal system clock. The variations in the system clock frequency affect the bus timings. The following diagram shows the timings for the bus input signals PB_ADN (read cycles) and PB_IOCHRDY.

PB_IORDY is an asynchronous input. Failure to meet setup and hold times may result in a failure to recognize inputs, but will not cause system failure.

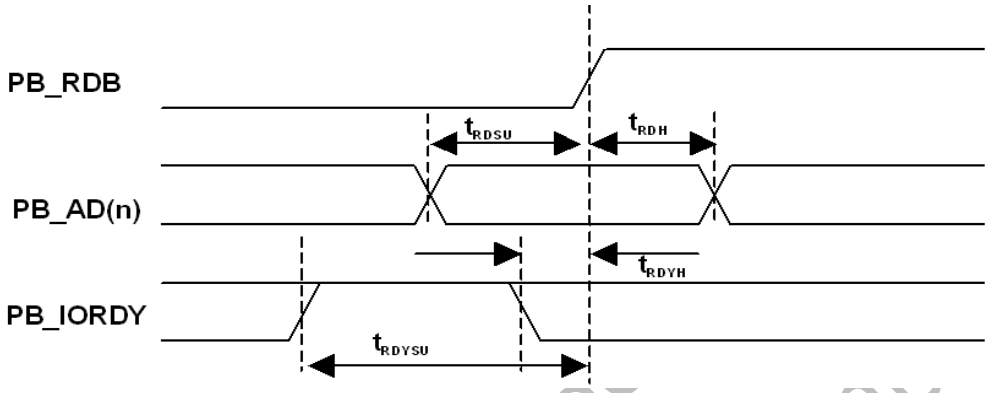


Figure 5-12. **PBI input timing diagram**

IDE/DVD Loader Interface

Introduction

The IDE/DVD Loader interface may be configured to be either an IDE or DVD loader interface.

The IDE (ATA/ATAPI-6) interface enables CD or DVD drives, hard disc drives (HDD), CompactFlash or memory stick readers to be easily incorporated into a system. It supports both PIO and DMA. Two IDE devices may be connected to the IDE port, although only one at a time may be used.

Content over this interface may be optionally AES, 3DES or DES encrypted/decrypted. The 8-/16-bit parallel DVD loader interface is compatible with a variety of loaders.

Features

- IDE interface may be configured to be either an IDE or DVD loader interface
- Supports ATA/ATAPI-6 device attachments
- Supports transfer rates of up to 100MB/sec (UltraDMA mode 4)
- Supports both PIO and DMA transfer modes
- Allows direct attachment of CD, DVD, CompactFlash, memory stick readers or hard disk drives

Block Diagram

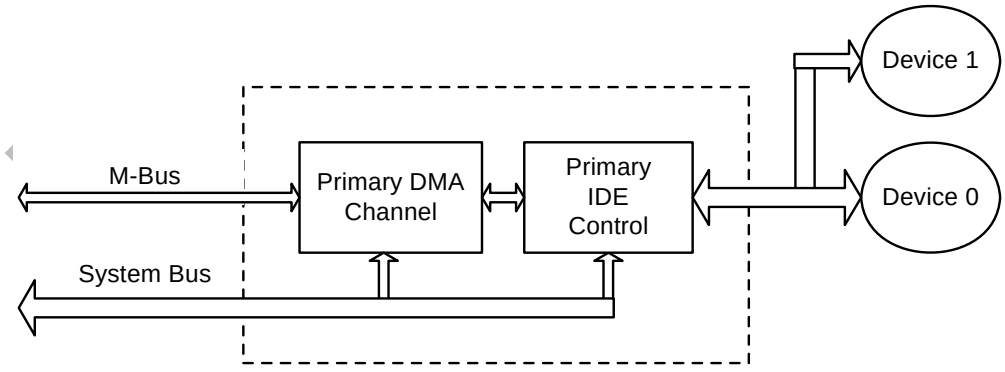


Figure 5-13. **IDE interface block diagram**

Functional Description

The IDE controller interfaces between up to two external IDE devices and the DDR memory. It provides a simple, standard interface to mass storage peripherals. The IDE (also referred to as ATA) interface is a common feature of consumer-level DVD and HDD drives; the EM8622L provides a direct interface to these drives.

The EM8622L can support peak transfer rates of up to 100MB/sec. using the synchronous, dual-clock-edge protocol.

The interface provides support for the cable ID function, which allows systems to be designed to accommodate either the standard 40-conductor cable, or the optional 80-conductor cable.

The IDE controller gets data from the IDE drive and sends it to the DDR memory. The IDE interface is primarily used to connect mass storage peripherals to the system. The flexible, low-cost ATA interface is a standard addition to consumer-level DVD drives. The EM8622L provides a direct interface to these IDE drives. The ATA drives are memory-mapped devices.

IDE Interface

The pre-fetch and post write sub-module controls the data read/write of the external ATA devices during the read sectors, the write sectors, the read multiple and the write multiple commands. When data pre-fetching and posting are enabled, accesses to the IDE data port use the FIFO in the DMA channels.

The data is pre-fetched on 512-byte boundaries into the FIFO during the data port reads. The data is written into the FIFO during data port writes. During the read commands data transfer, writes to any register in the external ATA/ATAPI device other than the data port causes the FIFO to be cleared and pre-fetched data to be lost. During the write commands with write posting enabled, writes to any register other than the data port register causes data in the FIFO to be flushed to the external device, before the write is allowed to continue.

IDE Control

The IDE interface is controlled by the primary IDE control register. It also selects the timing characteristics of the IDE cycle for the PIO and the standard bus master transfers. The I/O transactions targeting the IDE ATA register blocks (command and control blocks) are positively decoded and drive the IDE interface.

Two IDE timing registers control the timing characteristics for the primary drive 0 and primary drive1 on the IDE channel. They allow the programming of independent operating modes for each IDE agent. The UltraDMA control register enables each individual channel and drive for UltraDMA transfers (both ATA/33 and ATA/66).

DMA Channel

The DMA channels are identical blocks. The DMA channels interface the channel interface block and an IDE controller. The IDE controller block strobes data into and out of the 64x32-bit FIFO in the DMA channel block based on the setting of the read/write direction bit in the register BMIC. These channels contain the bus master IDE control registers mapped to the I/O space.

The DMA activity starts when the host software sets the start/stop bit in the register BMIC. The DMA channel state machine starts moving to/from the system memory from/to the 64x32-bit FIFO in the corresponding channel. The source/destination address in the DDR memory is provided by the register IDE_DMAPTR. The size of the transfer is specified in the register IDE_DMALEN.

When all the data transfer for the command is complete, the ATA device asserts an interrupt. The DMA channel waits until the last data words have been moved to/from the system memory before setting the interrupt bit and clearing the bus master IDE active bit of the register BMIS.

Register Maps

IDE Controller Configuration Registers

Upon reset, the IDE controller sets its internal registers to a predetermined default state, which represents the minimum functionality feature set required to bring up the system. It is the responsibility of the firmware to properly program the configuration registers to achieve optimal system performance.

Table 5-19. **IDE controller configuration registers**

Address[1]	Register Name	R/W/A[2]	Description
0834	IDESRC	R/W	IDE Slew Rate Control Register
0838	PRI_DRV1UDMATIM1	R/W	Primary Drive 1 UltraDMA Timing Register 1
083C	PRI_DRV1UDMATIM2	R/W	Primary Drive 1 UltraDMA Timing Register 2
0840	PRI_IDECTL	R/W	Primary IDE Control Register
0844	PRI_DRV0TIM	R/W	Primary Drive 0 Timing Register
0848	PRI_DRV1TIM	R/W	Primary Drive 1 Timing Register
084C	IDEMISC	W	IDE Miscellaneous Register
0850	IDESTATUS	R	IDE Status Register
0854	UDMACTL	R/W	UltraDMA Control Register
0858	PRI_DRV0UDMATIM1	R/W	Primary Drive 0 UltraDMA Timing Register 1
085C	PRI_DRV0UDMATIM2	R/W	Primary Drive 0 UltraDMA Timing Register 2
08C4	PREF_ST	R	Prefetch FIFO Status Register

1. Address refers to G-Bus byte address relative to the host interface register base.
2. Read/Write/Auto update.

IDE Device Registers

The primary channel devices command and control block registers window through the IDE block.

Table 5-20. **IDE device registers**

Address[1]	Register Name	R/W/A[2]	Description
0866	Reserved		
0870-77	Reserved		
08E6	IDE_DEV_CTL	R/W	Primary IDE Control Block Register
08F0-F7		R/W	Primary IDE Command Block Register

1. Address refers to G-Bus byte address relative to the host interface register base.
2. Read/Write/Auto update.

IDE DMA Channel Registers

Table 5-21. **IDE DMA channel registers**

Address[1]	Register Name	R/W/A[2]	Description
0900	BMIC	R/W	Bus Master IDE Command Register
h0904	BMIS	R/W	Bus Master IDE Status Register
0908	Reserved		
09E0	IDE_DMAPTR	R/W	IDE DMA Address Register
09E4	IDE_DMALEN	R/W	IDE DMA Length Register
09F0	PIO_PREFETCH_DATA	R/W	PIO Prefetch Data Register

1. Address refers to G-Bus byte address relative to the host interface register base.
2. Read/Write/Auto update.

Pin Description

IDE Pins

The IDE/ATA interface provides a dedicated port for the attachment of standard storage devices.

Table 5-22. **IDE interface pin descriptions**

Pin Name	Ball ID	Direction	Description
IDE_D15	A16	B	IDE data bit 15 (MSB)
IDE_D14	B16	B	IDE data bit 14
IDE_D13	A17	B	IDE data bit 13
IDE_D12	B17	B	IDE data bit 12
IDE_D11	D16	B	IDE data bit 11
IDE_D10	C17	B	IDE data bit 10
IDE_D9	D17	B	IDE data bit 9
IDE_D8	D18	B	IDE data bit 8
IDE_D7	E18	B	IDE data bit 7
IDE_D6	E16	B	IDE data bit 6
IDE_D5	C18	B	IDE data bit 5
IDE_D4	B18	B	IDE data bit 4
IDE_D3	A18	B	IDE data bit 3
IDE_D2	C16	B	IDE data bit 2
IDE_D1	D15	B	IDE data bit 1
IDE_D0	C15	B	IDE data bit 0 (LSB)
IDE_A2	B13	O	IDE bus address bit 2
IDE_A1	D13	O	IDE bus address bit 1
IDE_A0	C13	O	IDE bus address bit 0
IDE_CS1#	A12	O	Chip select #1
IDE_CS0#	A13	O	Chip select #0
IDE_IOW#	D14	O	Write command (active-low)
IDE_IOR#	A15	O	Read command (active-low)
IDE_IORDY	C14	I	IDE cycle extension input
IDE_DMARQ	B15	I	DMA request

Table 5-22. **IDE interface pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
IDE_ACK#	B14	O	DMA acknowledge
IDE_INTRQ	A14	I	IDE device interrupt request
IDE_NPCBLID	B12	I	Cable ID input. Used to detect the type of cable assembly in use.

Electrical Characteristics

IDE Interface DC Characteristics

Table 5-23. **IDE interface DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA			
I_{OL}[a]	Low level output current (@V_{OL} = 0.4V)	mA			
V_{IH}[2]	Input high voltage	V			
V_{IL}[b]	Input low voltage	V			

1. Parameter applies to all outputs and bidirectionals when driving.
2. Parameter applies to all inputs and bidirectionals when receiving.

IDE Interface AC Characteristics

Table 5-24. **IDE interface AC characteristics**

Symbol	Description	Units	Min	Typ	Max

Timing Diagram

TBD

Smart Card Interface

Introduction

The XPU block in the EM8622L contains a smart card interface. The smart card interface supports ISO/IEC7816 standard asynchronous protocols. It may be used for applications requiring DVB-CSA conditional access. The smart card interface controller is connected to the card reader via an IC card interface, which performs all the supply, protection and control functions.

Features

- Supports ISO/IEC7816 standard asynchronous protocols
- May be used for applications requiring DVB-CSA conditional access

Block Diagram

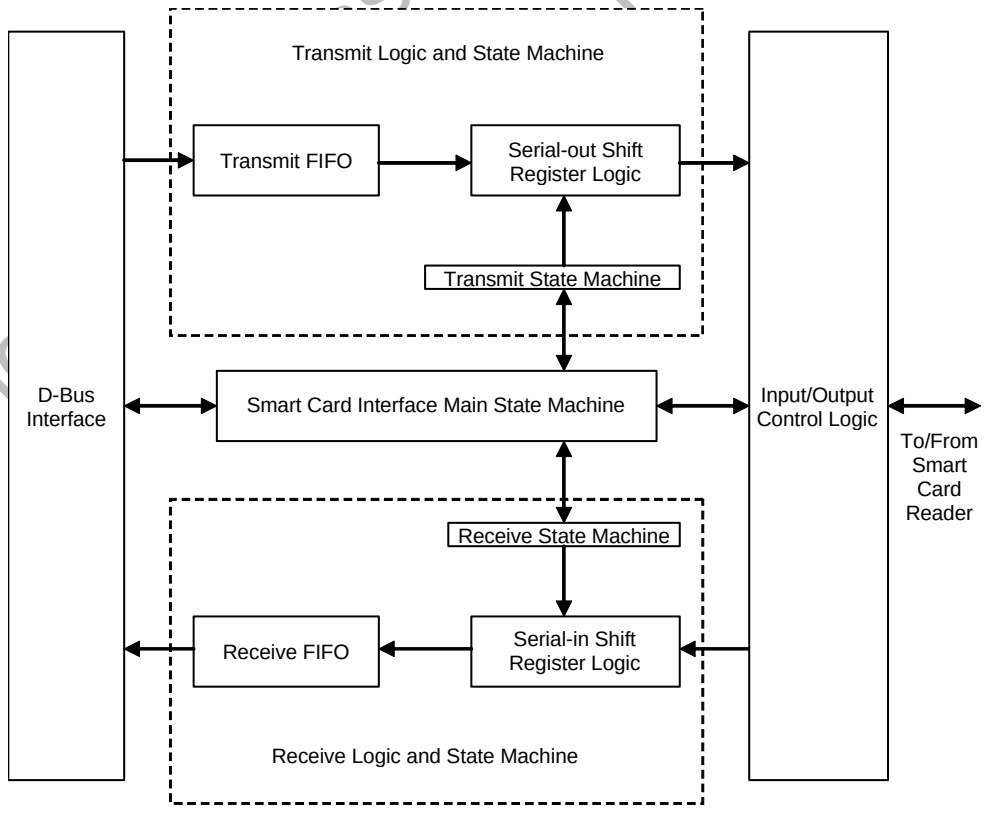


Figure 5-14. **Smart card interface block diagram**

Functional Description

The smart card module contains a D-Bus interface, transmit logic, main state machine and output control logic sub modules. The D-Bus interface module contains all the registers. The transmit logic contains the state machine, transmit fifo and the transmit output shift register logic sub modules.

The transmit state machine waits for the byte(s) from the D-Bus interface. When it gets a byte, it tries to send it to the smart card reader. When it completes the transfer it asserts an interrupt to indicate the completion of the transmission. After the interrupt has been asserted, the D-Bus interface (CPU) supplies the next byte. If the FIFO is enabled, then more bytes can be written into the FIFO. This reduces the number of interrupts to the CPU and increases the performance.

The main state machine of the card controls the state of the card by doing the card activation, cold reset, warm reset, clock stopping and the deactivation of the card.

The main smart card interface state machine contains the following states:

1. Unpowered idle state: Initial state after reset.
2. Activate card state: The card is activated.
3. Cold reset state: Cold reset is applied to the card.
4. Powered idle state: The card is powered and ready for use (read/write).
5. Warm reset state: Warm reset is applied to the card.
6. Clock stop state: The clock to the card is stopped.
7. Deactivate card state: The card is deactivated.

The default state after reset is the Unpowered idle state. The state machine should be in the powered idle state in order to communicate to the card. The initialization sequence from the unpowered idle to the powered idle state can be done either entirely by the hardware or by both the hardware and the software.

If the software performs the initialization sequence, then it should change the state by writing into the STATE_REG register.

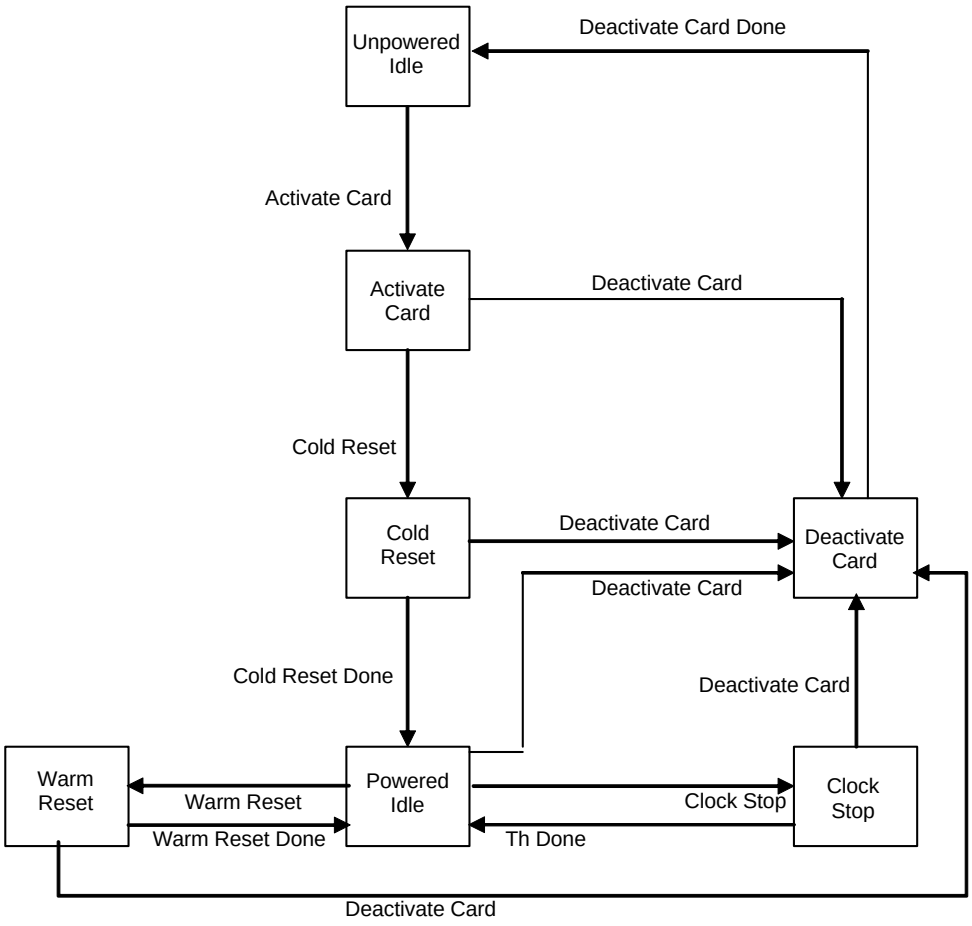


Figure 5-15. **Smart card interface state machine**

The receive logic contains the receive state machine, receive fifo and the receive input shift register logic. The receive state machine waits for any activity on the smart card data line. As soon as it detects a transfer, it tries to catch the byte(s) in the shift register. After receiving one byte, it interrupts the CPU. If the receive FIFO is enabled, then more bytes can be read from the smart card into the FIFO. This reduces the number of interrupts to the CPU and increases the performance.

Before any write/read to the smart card, registers CLK_HIGH_VAL, CLK_LOW_VAL, EGT_ETU_REG, PARAM_REG, SCARD_CTRL_REG, SOFT_OUT_REG and INTEN_REG should be programmed to the desired values.

To write to the smart card without the FIFO, data is written to the registers TX_BYTE_REG or TX_WORD_REG or TX_DWORD_REG. This starts the write protocol to the smart card. After the data is transmitted an interrupt is asserted. More data can be sent in the sane manner given above.

To write to the smart card with the FIFO, write data to the registers TX_BYTE_REG or TX_WORD_REG or TX_DWORD_REG. Up to 16 bytes can be written into the transmit FIFO (FIFO depth = 16 bytes). The register STATE_REG gives the number of bytes in the transmit FIFO. If this is less than 16, then more bytes need to be written into the FIFO.

To read from the smart card without the FIFO, wait for the RX_DONE interrupt. This interrupt is set once one byte is received. Then, read the data from the register RX_BYTE_REG. To receive more data repeat the steps given above.

To read from the smart card with the FIFO, wait for the RX_DONE interrupt. This interrupt is set once the receive FIFO reaches the RFIFO_THRESHOLD value. Then, read the data from the registers RX_BYTE_REG or RX_WORD_REG or RX_DWORD_REG. Read until the FIFO is not empty. To receive more data repeat the steps given above.

Writing into the register TX_BYTE_REG sends one byte to the smart card. Writing into the register TX_WORD_REG sends two bytes to the smart card. Writing into the TX_DWORD_REG register sends four bytes to the smart card.

If FIFO is enabled, then the data will be written into the FIFO. Then, from the FIFO the data will be sent to the smart card.

Reading the register RX_BYTE_REG gets one byte from the smart card. Reading the register RX_WORD_REG gets two bytes from the smart card. Reading the register RX_DWORD_REG gets four bytes from the smart card.

The register CLK_HIGH_VAL gives the smart card output clock high value in number of sysclks. The register CLK_LOW_VAL gives the smart card output clock low value in number of sysclks. The register TIMEOUT_LOAD gives the time-out value programmed in number of smart card clocks. The register PARAM_REG gives the Tc, Tb, Tg and Th counter load values programmed in number of 256 smart card clocks. The register EGT_ETU_REG gives the EGT and ETU count load values. The ETU load value is programmed in number of smart card clocks and the EGT load value is programmed in number of ETU units.

By using the register SOFT_OUT_REG, the smart card pin values can be directly controlled. The smart card control register activates/deactivates the card. The smart card state register controls the states of the smart card interface state machine.

Register Map

Smart Card Interface Registers

Table 5-25. **Smart card interface registers**

Address[1]	Register Name	R/W/A[2]	Description
+C300	TX_BYTE_REG	W	Smart Card Transmit Byte Register
+C304	TX_WORD_REG	W	Smart Card Transmit Word Register
+C308	TX_DWORD_REG	W	Smart Card Transmit Dword Register
+C30C	Reserved		
+C310	RX_BYTE_REG	R	Smart Card Receive Byte Register
+C314	RX_WORD_REG	R	Smart Card Receive Word Register
+C318	RX_DWORD_REG	R	Smart Card Receive Dword Register
+C31C	Reserved		
+C320	CLK_HIGH_VAL	R/W	Smart Card Clock High Value Register
+C324	CLK_LOW_VAL	R/W	Smart Card Clock Low Value Register
+C328	TIMEOUT_LOAD	R/W	Smart Card Time-out Value Register
+C32C	PARAM_REG	R/W	Smart Card Parameters Register
+C330	EGT_ETU_REG	R/W	Smart Card EGT and ETU Register
+C334	SOFT_OUT_REG	R/W	Smart Card Software Output Register
+C338	SCARD_CTRL_REG	R/W	Smart Card Control Register
+C33C	STATE_REG	R/W	Smart Card State Register
+C340	INT_REG	R/C	Smart Card Interrupt Register
+C344	INTEN_REG	R/W	Smart Card Interrupt Enable Register
+C348	ALT_ETU_CNT	R	Alternate ETU Count Register
+C34C	STATUS_REG	R	Smart Card Status Register
+C35C	CRC_REG	R	Smart Card CRC Register
+C350	CRC_INIT_REG	R/W	Smart Card CRC Initial Value Register

1. Address refers to G-Bus byte address relative to the XPU block register base.
2. Read/Write/Auto update.

Pin Description

Smart Card Interface Pins

Table 5-26. **Smart card interface pin descriptions**

Pin Name	Ball ID	Direction	Description
UART0_RTS	B10	B	SCARD_RST. Smart card interface reset.
UART0_DTR	E11	B	SCARD_CLK. Smart card interface clock.
UART0_CTS	C12	B	SCARD_FCB. Smart card interface FCB.
UART0_DSR	A11	B	SCARD_IO. Smart card interface IO.

Host Cipher

The host cipher contains two encryption/decryption units, AES and DES.

DES Encryption/Decryption Unit

The DES encryption/decryption unit can either encrypt or decrypt a 64-bit data block using the DES or the Triple DES algorithm. The supported modes are DES ECB, CBC, OFB and Triple DES TECB, TCBC and TOFB, both in encryption and decryption. The configuration is unique for a group of 64-bit data blocks. The number of blocks in a data group is 1 to 255. The IV register is used in the OFB and CBC chaining modes (ignored in the ECB mode). It must be programmed before the encryption/decryption is started, and gets automatically updated after each block is processed. The IV is updated without user intervention in the following modes: OFB, TOFB, CBC, TCBC both encryption and decryption. These modes are chained modes in which the output depends on the outputs of the previous blocks. The IV value at the end of a group is the value which would be required for an additional block in the group. It means that Encrypting 100 data blocks in CBC mode and then again 100 data blocks in CBC mode without writing any value in the IV between the two executions will be exactly the same as encrypting in one shot the 200 data blocks. This is to over run the limitation of 255 blocks in a group. There is thus no limitation in actual groups in either modes. The number of key pairs is 8 (even/odd).

Key format

The Triple DES/DES algorithm uses a 56-bit key. However, 8-bit are added to the key (in key[63:0] bits 0, 8…56) to get a 64-bit key (these bits are the parity control bits). These parity bits are not used during encryption or decryption. However, a 64 bit key must be written in the registers. The value of bits 0, 8, 16, 24, 32, 40, 48, 56 of key[63:0] can be anything. Thus, the encryption using the key = 0000000000000000 and the key = 0101010101010101 will exactly be the same.

Data Format

The data interface (input and output) is a standard valid/ready interface. Data block[63:0] = 64'h0011223344556677. The bytes will be sent in the order: 00 11 22 33 44 55 66 77. Thus the left most byte of the data block is sent for. The left most byte of the output block will be out first. If the encrypted/decrypted bytes received are in the order: 00, 11, 22, 33, 44, 55, 66 and 77, then output enciphered, deciphered block is 64'h0011223344556677.

Table 5-27. **Performance**

	DES - 64-bit to 8-byte	Triple DES - 64- bit to 8- byte
One Block	50 Cycles	86 Cycles
Several blocks	34 Cycles	49 Cycles

The total number of clock cycles represents the time between the start of the ciphering and the rising edge of the DES_END. If the number of the blocks is higher than one, then the blocks are received and sent while the other blocks are encrypted.

AES Encryption/Decryption Unit

The AES encryption/decryption unit can either encrypt or decrypt a 128, 194 or 256-bit data block using the AES algorithm. The supported modes are ECB, OFB, CBC, CFB and CTR, both in encryption and decryption. The supported key lengths are 128, 194 and 256-bit independent of the data block size. The configuration is unique for a group of data blocks. The number of blocks in a data group is up to 255. The number of key pairs is 3 (even/odd).

The key registers (0x1E60 - 0x1E67) are write only while the IV registers is read and write capable. Although writing to the key register is permitted, as the AES module is activated, the contents of the key and IV vector registers are over written with the contents in the code word RAM. When read from the key register, the D-Bus will return values of zeroes. Read of IV vector registers is permitted at the end of each block ciphering, to enable the continuation of the block ciphering in the chained modes.

In the ECB mode the IV registers are ignored. In the OFB, CBC and the CFB modes, these registers are used for IV storage. In the CTR mode these registers store the counter value.

The IV or counter is updated by the module in the following modes: OFB, CBC, CFB, and CTR (during both encryption and decryption). When the cipher is in a chaining mode, the output depends on the outputs of the previous blocks. The IV value at the end of a group is the value which would be required for an additional block in the group. It means that encrypting 100 data blocks in OFB mode, and then again 100 data blocks in any chaining modes without writing any value in the IV between the 2 executions will be exactly the same as encrypting in one step the 200 data blocks. This over runs the limitation of 255 blocks in a group. Thus there is no limitation in actual group size in any chaining mode.

In the CTR mode, the IV is composed of 3 components: a nonce (typically first 32-bit vector), an initial vector and a 32-bit counter. The counter is incremented for every data block. At the end of the group, the value taken by the counter is the value which would be needed for an additional block in the previous group.

Data Format

The data interface (input and output) is a standard valid/ready interface. Data block[127:0] = 128'h00112233445566778899aabbccddeeff. The bytes will be sent in the order: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff. The left most byte of the data block is sent first. The left most byte of the output block will be out first. If the encrypted/decrypted bytes received are in the order, 00, 11, 22, 33, 44, 55, 66, 77...ff, then the output enciphered, deciphered block is 128'h00112233445566778899aabbccddeeff.

Table 5-28. **Performance**

Size of block (byte)	16	16	16	24	24	24	32	32	32
Size of Key[1]	16	24	32	16	24	32	16	24	32
Total Cycles	95	115	130	150	160	170	225	230	245

1. In clock cycle (200MHz clock)

Serial Flash Interface

The EM8622L supports both parallel flash memory (connected via the peripheral bus interface) or SPI-compatible serial flash memory connected via a dedicated 4-wire interface. The interface is designed to support a large number of serial flash types and optimized for high-speed continuous reads.

The interface operates only with signaling corresponding to CPOL=0 and CPHA=0. Supported READ command values are 0x03, 0xE3 and 0xFF. The number of address bits must be 8, 9, 16, 24 or 32, followed by 0 to 127 stuff bits.

Pin Description

Serial Flash Interface Pins

Table 5-29. **Serial flash interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SF_CLK	AC15	O	Serial flash interface clock output
SF_CSB	AB15	B	Serial flash interface chip enable out. Input at reset for reading configuration straps.
SF_DI	AA15	I	Serial flash interface serial data in.
SF_DO	Y15	B	Serial flash interface serial data out. Input at reset for reading configuration straps.

Electrical Characteristics

Serial Flash Interface DC Characteristics

Table 5-30. **Serial flash interface DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}[a]	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
V_{IH}[2]	Input high voltage	V	2.0		5.5
V_{IL}[b]	Input low voltage	V	-0.3		0.8

1. Parameter applies to all outputs and bidirectionals when driving.
2. Parameter applies to all inputs and bidirectionals when receiving.

Serial Flash Interface AC Characteristics

Table 5-31. **Serial flash interface AC characteristics**

Symbol	Description	Units	Min	Typ	Max

6 Video Decoder Subsystem

Block Diagram of Video Decoder Subsystem

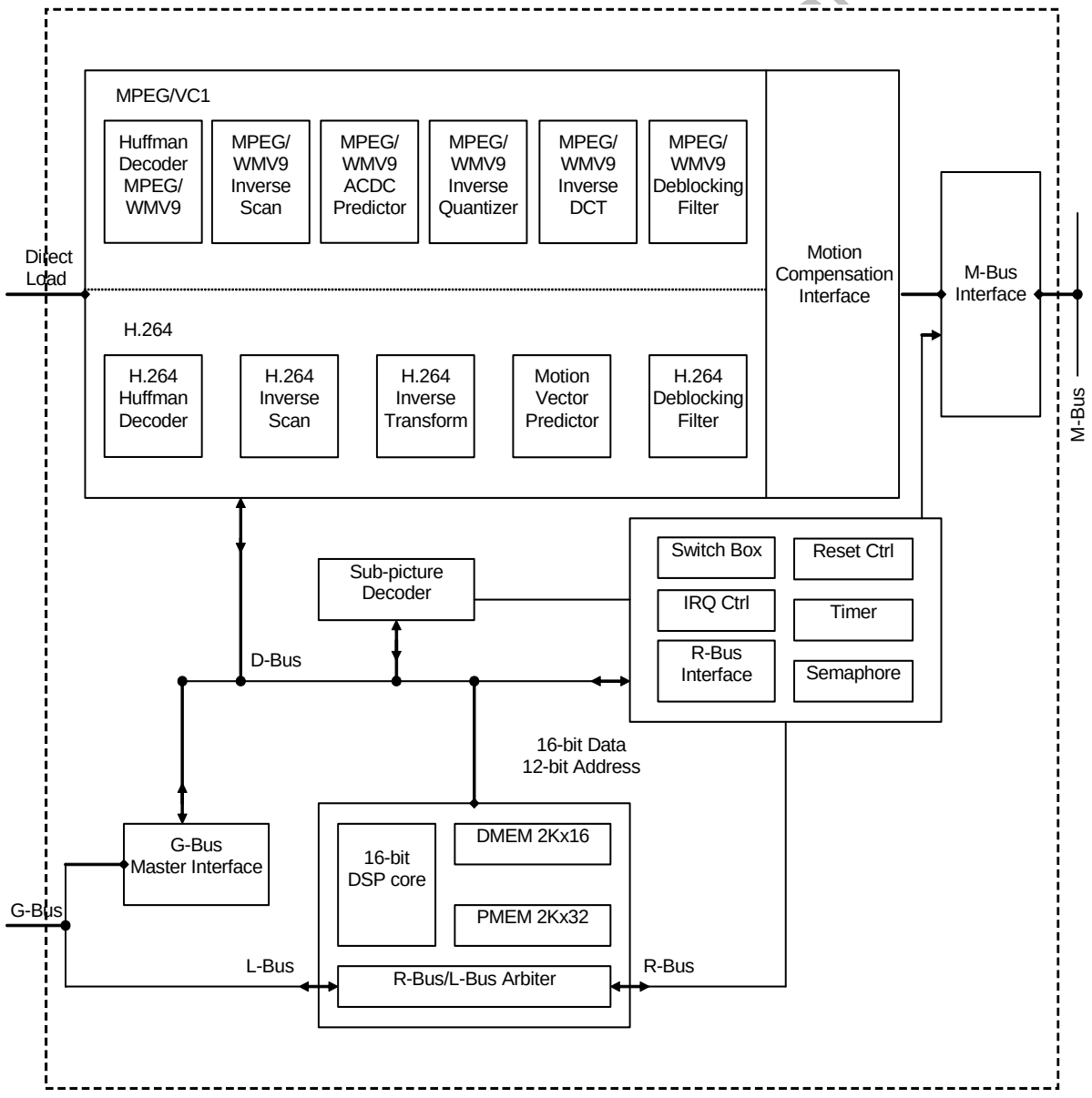


Figure 6-1. **Video decoder subsystem block diagram**

Introduction

The EM8622L video decoder subsystem executes the video decoding algorithms supported by the EM8622L. Its architecture is a hybrid of both processor-based and hard-wired logic approaches. The proprietary RISC CPU runs at 300MHz, and the hardware-based data path processing runs at 200MHz.

The EM8622L can decode MPEG-4.10 (H.264), SMPTE VC-1, Microsoft Windows Media 9 (WM9), MPEG-2, MPEG-1 and DVD. The maximum number of simultaneous programs that can be decoded and displayed depends on the source format and resolution. The EM8622L can also support the requirements for MSTV IPTV applications.

Table 6-1. **Supported E8622L video decoding formats**

	MPEG-2	MPEG-4.2	WMV9	VC-1		MPEG-4.10 (H.264)	
Max profile and level	MP@HL	ASP@L5	MP@HL	MP@HL	AP@L3	BP@L3	MP@L4.1 HP@L4.1
1080i30	2		N/A	N/A	2	N/A	2
1080p30	2		2	2	2	N/A	2
720p60	2	2	2	2	2	N/A	2
480p60 576p50	2	2	2	2	2	2	2
480i30 576i25	2	2	N/A	N/A	2	2	2

The video decoder engine consists of a proprietary 16-bit RISC CPU which is augmented by a number of hardware functional units.These functional units perform the most compute-intensive portions of the video decompression algorithms supported by the EM8622L. The video RISC CPU executes certain portions of the algorithms, and prepares the data for, and coordinates the execution of the hardware units.

This architecture represents a carefully engineered trade-off between the hardware complexity, operating clock frequency and adaptability.

Appropriate microcode (supplied by Sigma Designs Inc.) is loaded into the video RISC program memory from the DRAM to perform the required video decompression.

The high memory bandwidth requirements created by the motion compensation features of the various algorithms is addressed by the ‘Direct Load Port’ feature of the engine. This port provides a dedicated, prioritized, high-bandwidth port to the system memory.

Features

- MPEG-1
- MPEG-2 MP@HL up to 1920x1080p24, 1920x1080i30 or 1280x720p60 resolution
- MPEG-4.2 ASP@L5. Rectangular shape video decoding up to 1280x720p30 resolution, support for B Pictures, data partitioning and error resiliency. No support for global motion compensation (GMC).
- WMV9/VC-1 MP@HL up to 1280x720p60 (<90% of the P-picture macroblocks have 4 motion vectors) or 1920x1080p24 resolution. Up to 1280x720p30 or 1920x1080p24 resolution if unconstrained 4MV bitstream.
- VC-1 AP@L3 up to 1920x1080i30 (<60% of the B-picture macroblocks have 4 motion vectors), 1920x1080p24 or 1280x720p60 resolution. Up to 1920x1080i25, 1920x1080p24 or 1280x720p60 or resolution if unconstrained 4MV bitstream.
- MPEG-4.10 (H.264) BP@L3 up to 720x480p30 or 720x576p25 resolution, including FMO and ASO
- MPEG-4.10 (H.264) MP@L4.1 and HP@L4.1 up to 1920x1080p24, 1920x1080i30 or 1280x720p60 resolution (constrained to match BD and HD-DVD specifications), 180 Mbin/s maximum for CABAC streams
- Hardware accelerated Baseline JPEG decoding
- DVD-Video and Superbit DVD
  - CSS decryption
  - 16:9 and 4:3 playback, letterbox, 3:2 pull-down
  - Multiple angles and sub-picture
- Error concealment, deblocking filter
- Elementary video stream bit rate
  - MPEG-2 SDTV (HDTV): 20 (40) Mbps maximum
  - MPEG-4.2 SDTV (HDTV): 20 (40) Mbps maximum
  - MPEG-4.10 (H.264) SDTV (HDTV): 20 (40) Mbps maximum
  - WMV9/VC-1 SDTV (HDTV): 20 (40) Mbps maximum

MPEG-4.10 (H.264) SEI and VUI

Table 6-2. **Supported MPEG-4.10 (H.264) SEI (Supplemental enhancement information) messages and VUI (video usability information)**

	Supported	Not supported	Application Responsible for Implementation	Not Applicable
Buffering period SEI message	X			
Picture timing SEI message	X			
Pan-scan rectangle SEI message	X			
Filler payload SEI message	X			
User data registered by ITU-T Recommendation T.35 SEI message			X	
User data unregistered SEI message			X	
Recovery point SEI message syntax	X			
Decoded reference picture marking repetition SEI message			X	
Spare picture SEI message		X		
Scene information SEI message		X		
Sub-sequence information SEI message		X		
Sub-sequence layer characteristics SEI message		X		
Sub-sequence characteristics SEI message		X		
Full-frame freeze SEI message			X	
Full-frame freeze release SEI message			X	
Full-frame snapshot SEI message			X	
Progressive refinement segment start SEI message			X	
Progressive refinement segment end SEI message			X	
Motion-constrained slice group set SEI message				X
All VUI content	X			

7 Video Processing Subsystem

Block Diagram of Video Processing Subsystem

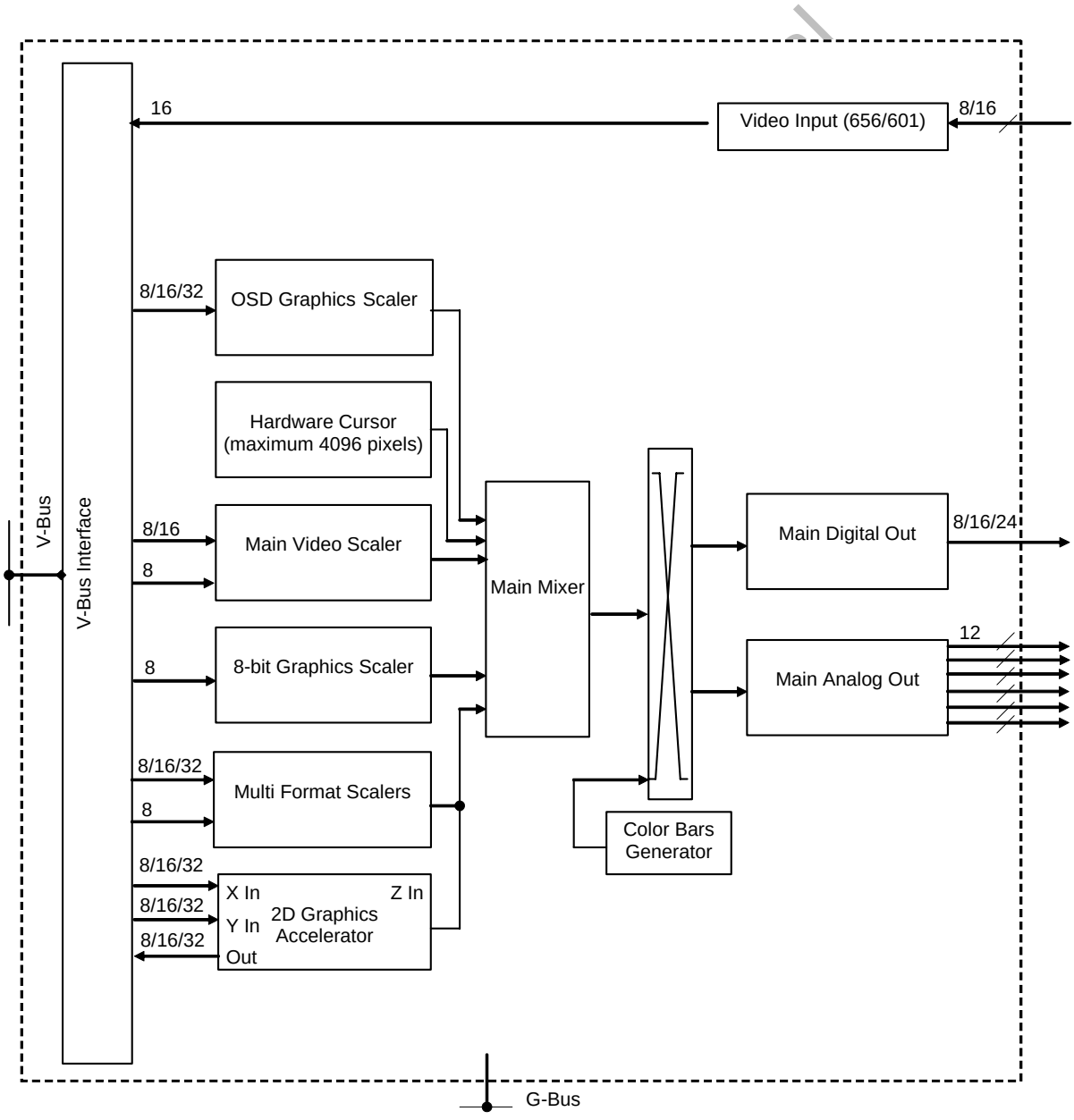


Figure 7-1. **Video processing subsystem block diagram**

Introduction

The EM8622L video processing engine provides sophisticated display processing, formatting and output capabilities.

The video processing block includes 5-way or 6-way alpha mixing of,

- A video plane
- A video/graphic plane
- An OSD (graphic only) plane
- An 8-bit graphic plane
- A hardware cursor (always on top)

Each video and graphics picture retrieved from the memory can be converted to other formats as necessary, and scaled independently. Independent brightness, contrast, and saturation controls are implemented for each of the two video streams. The OSD graphics scaler includes adaptive flicker filtering with edge detection.

The video engine also includes support for 2D graphics acceleration. The accelerator is a bit-block transfer mechanism for moving, filling, merging and expanding rectangular regions of display memory without the processor intervention after the initial setup. A pixel rate of one pixel per system clock can be supported when executing the acceleration operations.

Multiple video/graphics streams are merged in a video formatter. The formatter can alpha-blend up to four of the five available sources for each pixel.

The two available video outputs consist of,

- Main digital output: Supports 8, 16 or 24-bit output, RGB or YPbPr data format.
- Main analog output: Supports component RGB, component YUV, S-video, or composite output.

The main digital, main analog and the component analog output can each support output formats of up to 1920x1080p.

Features

Video Processing

- Hardware cursor (4096 pixels, 4-bpp, up to 255 pixels horizontally and vertically)
- Supports a 2D graphics accelerator with OpenType font rendering acceleration (up to 100M samples per second operation for most operations)
- Accelerator supports alpha/color merging alpha modulation
- Supports line, rectangle, ellipse and circle to generate a single-color line, rectangle, ellipse or circle with optional gradient fill
- Supports blend to alpha blend of one rectangular region onto another
- Supports move to move a rectangular region to another location
- Supports replace (a modified version of move)
- Supports raster operations (standard 256 boolean operations)
- Supports a 32-bit OSD with flicker filtering and scaling
- Supports optional deinterlacing of interlaced sources
- Supports arbitrary scaling of video and OSD of up to 1920x1080 pixels
- Supports scaling and alpha mixing of video, OSD, sub picture, graphics and hardware cursor
- Supports graphic channels with gamma correction

Video Input and Output Interfaces

- Supports a flexible 40-bit video/graphic input interface
- Supports brightness, color and contrast controls for each output port
- Supports a 150MHz RGB/YCbCr digital video output interface with color temperature control
- Supports NTSC/PAL composite analog output with optional Macrovision v7.1.L1 protection (54MHz, 12-bit DAC)
- Supports NTSC/PAL s-video analog output with Macrovision v7.1.L1 protection (54MHz, 12-bit DACs)
- Supports analog YPbPr / RGB with Macrovision v7.1.L1 and v1.2 protection in 480i, 576i, 480p and 576p YPbPr output modes (12-bit DACs, interlaced or progressive, SDTV or HDTV resolution)
- Supports 8-bit 4:2:2 YCbCr data

- Supports 16-bit 4:2:2 YCbCr data
- Supports 24-bit 4:4:4 YCbCr data
- Supports 24-bit RGB data (888)
- Supports BT.601, BT.656, or VIP 2.0, ‘video valid’ output signal
- Supports master or slave timing

Functional Description

Video Processing

The video processing and display unit (VPD) has extensive capabilities for retrieving graphics and video images from the memory, formatting the images as needed, mixing the images and then presenting the video stream for display in a required format. Other capabilities provided by the VPD include hardware-assisted 2D graphics acceleration, and support for an external video input port. The primary functions performed include:

- Up- or down-scaling a video or graphics image to a specified resolution
- Data format conversion (color lookup, color space conversion, etc.)
- Independent positioning of each scaled image
- Ordering and merging (alpha-blending) of selected video and graphics image planes
- Brightness, contrast and saturation control
- TV encoding to NTSC and/or PAL standards
- Selection of analog out mode (CVBS, S-video, component RGB, component YPbPr)
- Video digital-to-analog conversion
- Digital video output formatting
- External video input interface
- 2D graphics acceleration

Data Formats

The EM8622L supports video and graphics data stored in a variety of formats. For each pixel any four of the planes can be blended. The various formats are depicted below:

The video plane is used for the main video source. It supports the 4:1:1, 4:2:0 and 4:2:2 YCbCr formats. Both standard definition and high-definition resolutions are supported.

The graphics plane is used for graphics images or a second video source. It supports the RGB and 4:2:0, 4:1:1, 4:2:2 and 4:4:4 YCbCr formats. Only the standard definition resolutions are supported.

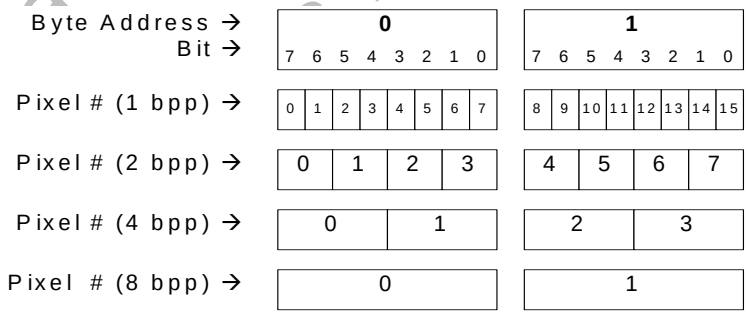
For the RGB data, 4 palletized color depths are supported: 2 colors (1-bpp), 4 colors (2-bpp), 16 colors (4-bpp) and 256 colors (8-bpp). A 256x32 look-up table converts the 1, 2, 4 or 8-bit code into 24-bit YCbCr plus 8-bit alpha. A 16-bpp format is available that supports the following formats: 565 RGB, 1555 ARGB and 4444 ARGB. 24-bit 888 RGB and 32-bit 8888 ARGB formats are also available.

Picture Buffer Formats

Graphics – 1/2/4/8-bpp, with LUT

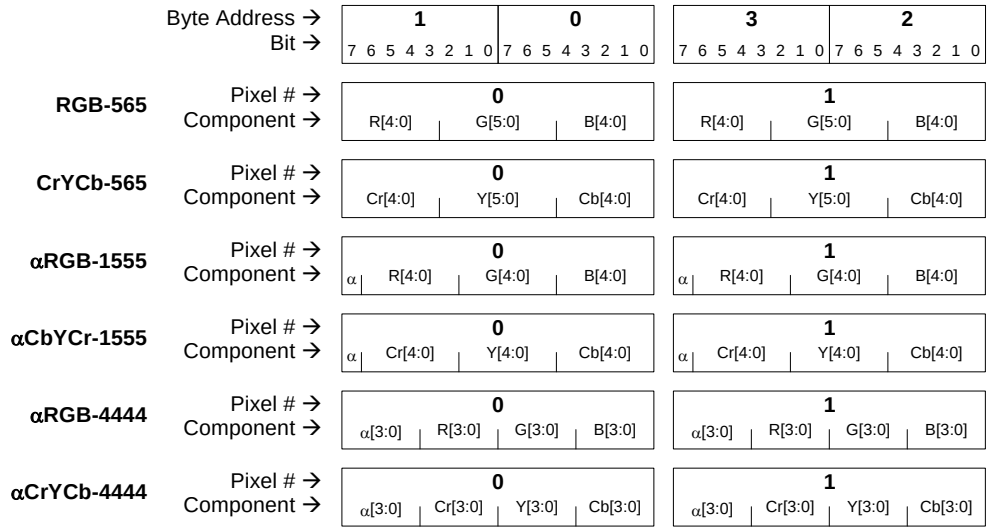
LUT formats with 1/2/4-bpp are supported by the graphics accelerator (Y input only).

LUT formats with 8-bpp is supported by the graphics accelerator (X and Y inputs, output) as well as the video output OSD channel



Graphics - 16-bpp

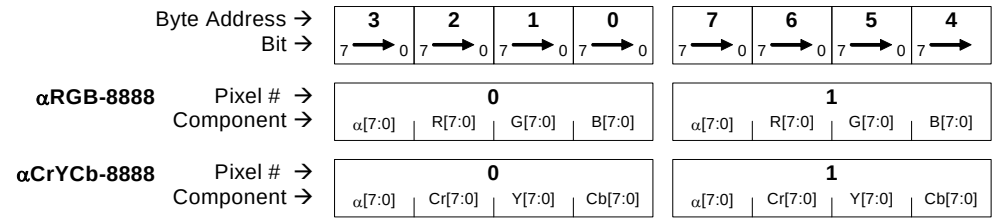
There are six formats using 16-bpp. All are supported by the 2D graphics accelerator (X and Y inputs, output) and by the video output OSD channel.



In the above diagram bytes are swapped (1, 0, 3, 2, 5, 4…). When the processor (or another G-Bus master) reads a 16-bit word from the DRAM, it will receive the low address byte in low position, and the high address byte in the high position. Thus, the 16-bit word will appear as shown in the diagram.

Graphics - RGB 32-bit format (RGB-32)

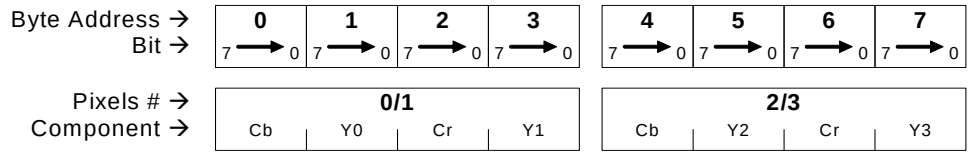
These modes are supported by the 2D graphics accelerator (source input, destination input and destination output) and by the video output OSD channel.



Video-Luminance

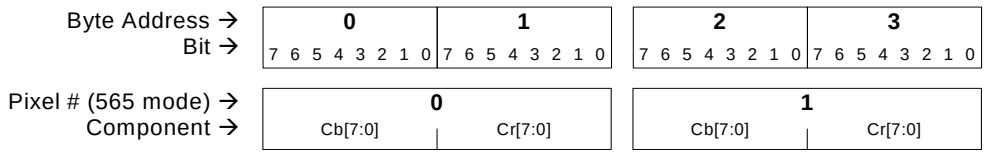
When an MPEG bit stream is decoded, the resulting frames are in the 4:2:0 or the 4:2:2 format. In either case, the luminance and the chrominance are stored in separate buffers (one buffer for luminance and one buffer for Cb and Cr).

In the luminance buffer, data is stored as 8-bpp, with increasing byte address corresponding to increasing pixel X coordinates, except on tile crossing boundaries.



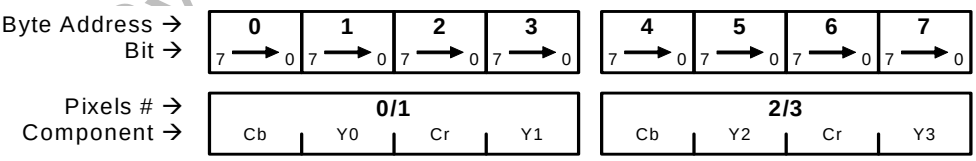
Video-Chrominance

In the chrominance, data is stored as 16-bpp, as shown below. A chrominance buffer typically uses the same number of bytes per line, as the associated luminance buffer in the 4:2:2 or the 4:2:0 modes (half the number of pixels, but two components). However, in the 4:2:0 mode, there are half as many lines of chrominance compared to the luminance.



Video – Interleaved 4:2:2

This format is used for the video coming from the video input ports. Each line contains 2xN pixels of luminance, N pixels of Cb and N pixels of Cr interleaved as follows:



OSD Graphics Scaler

The OSD enables full-screen menus, images and the text to be blended over the video, graphics and sub picture. The OSD scaler block supports all the graphics data formats, and performs programmable edge adaptive scaling with directional filtering. The H and V-scalers can operate in either 2-tap or 4-tap mode. In the 2-tap mode, the input line resolution of up to 2048 pixels is supported for all the input data formats. In the 4-tap mode, the memory size limitations restrict the input line resolution to 1024 pixels in 32-bpp data formats.

Table 7-1. **OSD line resolutions**

Tap	Pixel Resolution	Line Resolution
2-tap	16-bpp	2048
	24/32-bpp	2048
4-tap	16-bpp	2048
	24/32-bpp	1920
		1440
		1024

The OSD scaler also implements a 3-tap flicker reduction filter (scalers must be in 4-tap mode). Three sets of coefficients provide light, moderate, and strong flicker reduction options, in addition to no filtering. Four palletized color depths are supported: 2 colors (1-bpp), 4 colors (2-bpp), 16 colors (4-bpp) and 256 colors (8-bpp). A 256x32 look-up table converts the 1, 2, 4 or 8-bit code into 24-bit YCbCr plus 8-bit alpha.

A 16-bpp format is available that supports the following formats: 565 RGB, 1555 ARGB and 4444 ARGB. 24-bit 888 RGB and 32-bit 8888 ARGB formats are also available.

When decoding high-definition programs, the OSD is limited to 8-bit HD resolution or 32-bit SD resolution (upscaled to HD resolution).

Hardware Cursor

The hardware cursor block generates a small picture for the main mixer block. It supports up to 4096 4-bit pixels and feeds into the alpha mixer. An arbitrary bitmap is stored in 4-bpp format in a 512x32 on-chip SRAM. No external memory bandwidth is required to support the cursor. It may be organized as any size, up to 255 pixels horizontally or vertically. A 16x24 lookup table converts the 4-bit cursor data to 24-bit (6-6-6-6 format) YCbCr plus 8-bit alpha. Each video component is multiplied by four, and the α value is extended to 8-bit before being sent to the main mixer.

The horizontal and vertical dimensions of the cursor picture are constrained as follows:

- X size less than or equal to 255
- Y size less than or equal to 255
- Total pixels (X x Y) less than or equal to 4096

Multi-format Scalers 1 and 2(VCR/CRT/GFX)

The multi-format scaler is a general-purpose scaling unit which can accept all the supported video and graphics data formats (except 8-bit graphics). A 256x32 lookup table in the scaler supports the color expansion in 1, 2, 4 and 8-bpp input modes.

The scaler processes the video stream in the YC domain, and implements 2-tap H and V-scalers with a scaling range of 0.25 to infinity. Additional downscaling is possible by employing a pre-downscaler before the 2-tap scalers. The scaler can support the differing chrominance sample alignment of MPEG-1 versus MPEG-2. Due to the memory limitations the pixel/line input (line buffer level) resolutions are constrained.

Table 7-2. **Multi format scaler line resolutions**

Tap	Pixel resolution	Line resolution
4-tap	16-bpp	2048
	24/32-bpp	2048
		1280
		1024

The multi-format scaler is limited to standard definition input data only. The scaler is used whenever the type 2 deinterlacing is enabled, and can also be used to support picture-in-picture mode, or to provide scaled input data to the 2D accelerator unit.

Main Video Scaler

The main video scaler is similar to the multi-format scaler in general structure with several important differences:

- Only video input data formats are supported (no graphics)
- 4-tap H and V-scalers are implemented rather than 2-tap
- A special deinterlacing mode supports vertical scaling using a 2-tap filter combining previous and current field data
- HD (ITU 709) and SD (ITU 601) colorimetry conversion is supported (both directions)

Deinterlacing

Three types of deinterlacing (or progressive scan conversion) are available when interlaced sources are to be viewed on a progressive display.

The intra-field mode uses scan line interpolation to convert an interlaced source into progressive.

The flag mode uses flags within the MPEG bitstream. This mode will not work with the digital video input port since the video data is not MPEG compressed.

The inter-field mode examines the content over multiple fields, to determine how best to convert it into the progressive mode. Due to memory bandwidth requirements, only one interlaced source at a time may be processed using this algorithm. Two deinterlacing modes are available, Type 1 and Type 2.

Type 1: This algorithm is not motion adaptive. A frame N is built from the fields N-1 and N. Both the fields are upscaled, and a certain amount of data from the field N-1 is inserted in the output frame, giving a fixed proportion for the whole frame.

Type 2: This mode is motion adaptive. It uses 3 fields. A frame N is built from fields N-1, N and N+1. The lumas of the fields N-1 and N+1 (same polarity) are locally compared (4x2 blocks), resulting in a local motion detection. The fields N and N-1 are both upscaled and information from field N-1 is locally inserted in the output frame depending on how much motion is locally detected. Like most motion adaptive algorithms, it continuously shifts between inter-field deinterlacing when there is little motion, and intra-field deinterlacing when there is more motion.

The main video scaler V-Bus channels have a ‘dual field’ operation mode. In this mode, 2 fields are simultaneously sent to the main scaler, by alternating their lines: line 0 of field 0, line 0 of field 1, line 1 of field 0, line 1 of field 1 and line 2 of field 0. In the type 1 mode, fields N-1 and N must be read that way from the DRAM. In the type 2 mode, fields N-1 and N+1 must be read that way from DRAM. At the same time, field N must be sent to an available multi-scaler channel. The main video scaler will then output frame N-1 (upscaled from field N-1) with a modulated alpha indicating how much insertion is desired. In the same way, the multi scaler channel will output a frame N (upscaled from field N). The upscaled frame N-1 should then be blended on top of the upscaled frame N resulting into the motion adaptive de-interlaced output frame N.

The type 2 algorithm supports concurrent de-interlacing and re-sizing: for instance, if the input is 480, I then the display is 720 P. It could also be used to scale from one interlaced format to another interlaced format: for instance 480 I to 1080 I, or 480 I to 576 I.

De-interlacing Type 1

The following figure illustrates the deinterlacing algorithm. In this figure, there is no additional scaling on top of deinterlacing and the vertical scaling phase is 0.

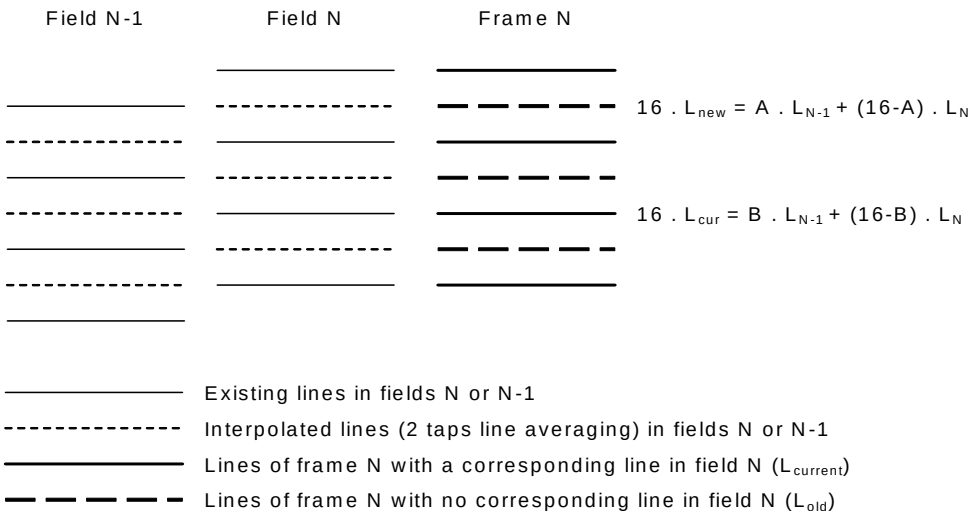


Figure 7-2. **De-interlacing type 1**

Here ‘A’ represents the fraction of data from the field N-1 used to create the new lines in the frame N (i.e. lines that do not exist in the field N). ‘B’ represents the fraction of data from the field N-1 inserted in the existing lines of field/frame N. The parity of the frame being deinterlaced must be programmed. The vertical scaling factor/phase must be set using the frame dimensions as input dimensions.

De-interlacing Type 2

Contrary to type 1, the type 2 de-interlacing uses three blocks within the video output:

- The main video scaler which inputs fields N-1, N+1 and outputs an intra field directional filtering de-interlaced frame N-1 with a motion modulated alpha.
- The multi scaler which inputs field N and outputs an intra field directional filtering de-interlaced frame.
- The mixer which blends the intra field directional filtering frame N-1 on top of the intra field directional filtering frame N (with directional filtering).

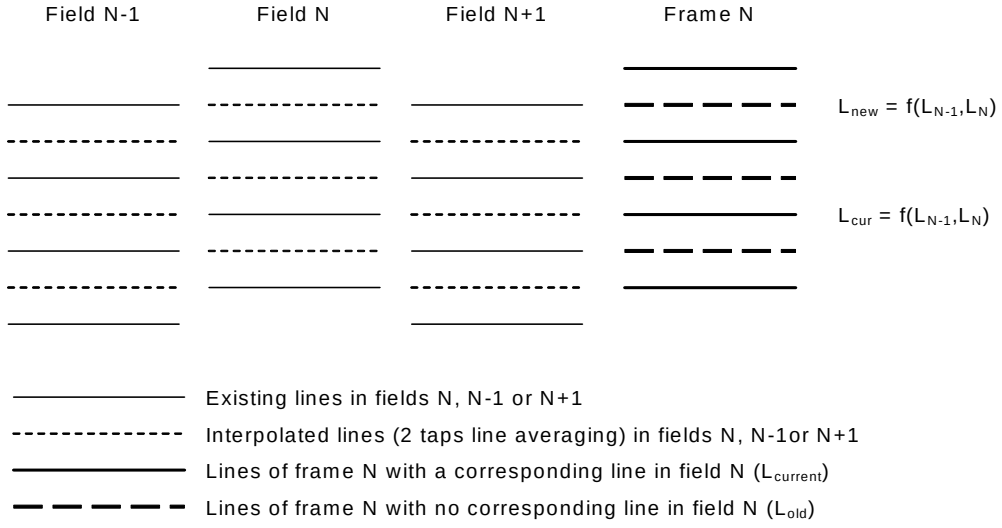


Figure 7-3. **De-interlacing type 2**

8-bit Graphics Scaler

This scaler is used to support sub-pictures, subtitles and closed captioning. Resolutions of up to 1920x1080 are supported.

For DVD-Video subpictures, the sub-picture block receives a picture from the DRAM, as a 8-bpp stream. The 8-bit sub picture data address 4x32, 16x32 or a 256x32 programmable lookup table to generate 24-bit YCbCr and 8-bit of alpha information. This allows support for DVB subtitles and EIA closed captioning. The 32-bit stream is sent to an adaptive H/V scaler, which can scale the picture by a ratio of 0.25 to infinity. Before the stream is output to the mixer, it can optionally be color-converted from YUV SD to YUV HD format.

The horizontal downscale block can reduce the number of pixel per line (pixel skipping) by a fractional ratio, and allows the sub-picture to be reduced to an arbitrarily small horizontal size.

The line buffer acts as a FIFO, and together with the adaptive XY scaling block, allows an arbitrary resizing of the sub-picture (limited to a down-scaling by 4 in either direction). The ‘conversion’ block can convert the video color space from YCC-SD to YCC-HD. The output of the sub-picture block is a 32-bpp stream (Cr,Y,Cb).

Graphics Accelerator

The graphic accelerator is composed of the following:

1. A bitmap accelerator to execute the bitmap operations like blending two images, moving a graphic in memory, combining graphics etc.
2. A vectorial accelerator to generate a two colors per pixel mask from a vectorial definition (e.g. true type fonts).
3. The gradient generator.

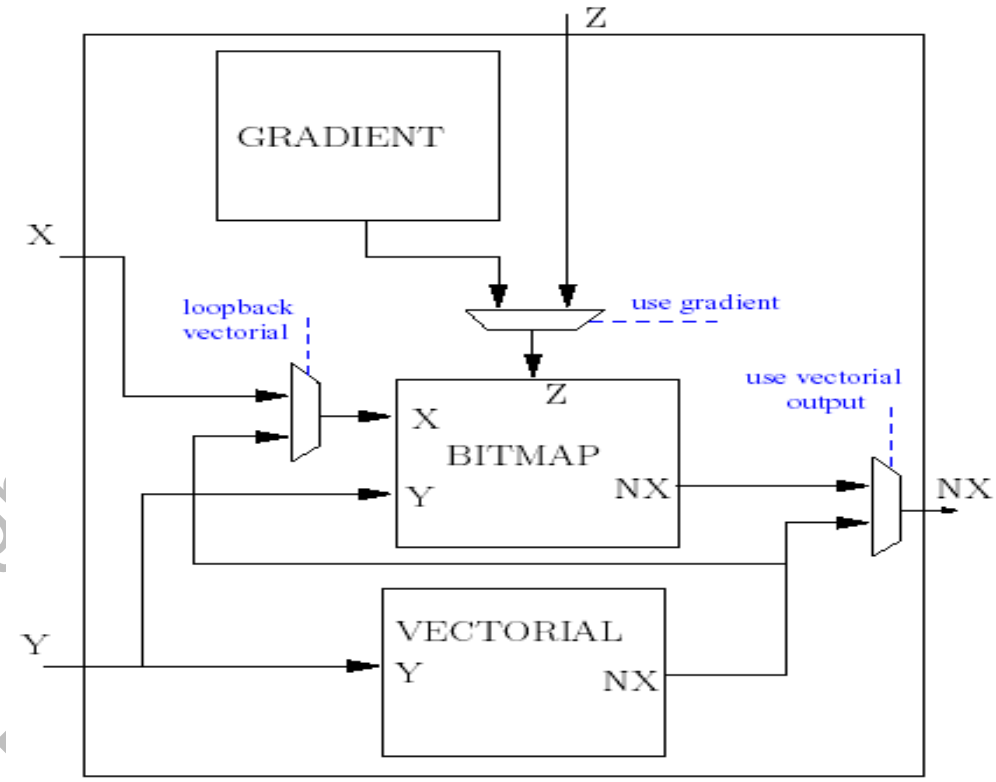


Figure 7-4. **Graphics accelerator block diagram**

The above figure shows the different utilization of the graphic accelerator and the corresponding data paths. The bitmap or the vectorial blocks can be used individually. The vectorial can be an input of the bitmap block and so cutting a texture. Moreover, instead of loading a texture from the memory, the bitmap can use a generated gradient. Therefore it is possible to generate a vectorial image with a gradient filling.

Graphics Accelerator (Bitmap Mode)

The graphics accelerator uses two DRAM read channels and one DRAM write channel to implement high speed pixel operations on rectangular regions of memory. An additional input to the accelerator is coupled to the output of the multi-scaler, giving it scaling and format conversion capabilities.

The graphics accelerator provides five basic operations:

1. Fill: Generates a single-color filled rectangle in the memory.
2. Blend: Alpha blends one rectangular region on top of another.
3. Move: Moves a rectangular region to another location.
4. Replace: Similar to move, but does not write transparent pixels.
5. Raster Operations: Implements a standard 256 raster operation set on the source data.

In the move and the replace modes, the accelerator can merge the color object with an alpha field. The graphics accelerator also includes a vectorial mode unit which greatly accelerates the rendering of vector mode fonts such as OpenType, TrueType, and Post-Script. This engine renders vector fonts in the following steps:

1. Data defining the glyph outline is read from the system DRAM via a V-bus DMA channel and is loaded into a FIFO.
2. A ‘control point extraction’ unit parses the incoming data and re-orders the data on a per-control-point basis.
3. The outline points data is transformed as required in a transformation matrix to effect rotations, etc.
4. The transformed outline points are then scaled as required in a scaling unit.
5. An ‘outline drawing’ unit then draws the outline in 64x64 pixel areas (2-bpp) and stores the results in the on-chip memory. The resulting outline regions are then filled (1-bpp) and the 64x64 regions are written to DRAM via a V-bus DMA channel

The unit is capable of rendering linear or second-order outlines directly, and handles third-order outlines if necessary by substituting a sequence of second-order segments.

X, Y input and Z output channels

The X channel can input:

1. 1/2/4/8-bpp alpha formats in which case,
   - 1 and 2-bpp data address a 4x8-bit alpha LUT
   - 4-bpp true alpha data is extended to 8-bit
   - 8-bpp is full true alpha information.
   - The X channel output is 8-bpp alpha data, that is used for alpha-color ‘merging’
2. 16/24/32-bpp color (with or without alpha) formats. The X channel output is 32-bpp.

The Y channel can input,

1. 1/2/4/8-bpp color LUT formats.
2. 16/24/32-bpp true color formats.

The Y channel has a 256x32-bit LUT that can be used to decode the 1/2/4/8-bpp formats. Depending on how the registers VO_GRAPH_ACC_CONTROL or OUTPUT_FORMAT are set,

1. The incoming 1/2/4/8-bpp data can be output as, 8-bpp or 16-bpp (the raw content format of the LUT if the sub mode is set to 16-bpp) or 32-bpp (whatever the content format of the LUT is).
2. The incoming 16-bpp data can be output as 16-bpp (unchanged) or 32-bpp.
3. Incoming 24/32-bpp data is output as 32-bpp.

Output formats that are not 32-bpp allow move/replace commands to be performed without altering the original DRAM format. However, even though the data is output in a non 32-bpp expanded format, the associated alpha information is processed and output along the original data. This is useful to determine wether the pixel is transparent or not, in case of a replace command.

The Z channel is the 32-bpp output of the GFX multi format scaler.

Graphic Accelerator (Vectorial Mode)

The font engine renders TrueType-like glyphs in 3 steps:

1. Reads a pre-parsed glyph from the DRAM (this data can also be provided directly by the G-Bus) and pre-processes the points stream.
2. Renders the glyph outline and fills the character.
3. A 1-bpp (black and white) bitmap is then generated in the DRAM.

The data defining the glyph outline is read from the DRAM through the V-Bus R10 channel and is cached. The engine is configured through the G-Bus. The ‘control point extraction’ re-orders the data on a per control point basis. Then, the outline points coordinates undergo an affine transformation. Some points needed for the calculations are inserted and the coordinates are scaled for the output device. The ‘outline drawing’ module draws the outline in a 64x64 pixel area (2-bpp) and stores it in a 1KB memory. Then the bitmap is filled (64x64x1-bpp) and each area is written to the DRAM through the W2 channel (8-byte x X line rectangle transfers). A state machine supervises all the memory accesses and the pixel areas sub-division.

The input data can be read from the DRAM or be written in the cache by a G-Bus master. The data read through the V-Bus or programmed by the G-Bus is stored in the cache. The glyph header (end point of each contour) is stored in the cache during the entire outline rendering process. This header must be a 4-byte multiple in order to be easily addressable. The outline control points information (flags and coordinates) is stored in a FIFO way, in the remaining part of the cache. The ‘control point extraction’ module inputs the data stream read in the cache. The purpose is to separate the data into different buses relative to:

- The X coordinates (16-bit)
- The Y coordinates (16-bit)
- The ‘End of contour’ and ‘end of glyph’ flags (2-bit)
- The ‘On/off curve’ and ‘cubic curve’ flags (2-bit).

In the case of a composite glyph, the coordinates of the points undergo an affine transformation defined by a 6-coefficient matrix. These coefficients (a, b, c, d, e, f) are calculated by the CPU and programmed by the G-Bus.

The module ‘point insertion’ inserts:

- The off-curve control points created when transforming a cubic Bezier curve into 4 quadratic ones. These points are the centers of the centers of the cubic curves points and they can be determined directly by an equation for each of them, depending on the three cubic vectors.
- An on-curve point beginning the contour when it is not the case.
- The on-curve point implied between two consecutive off-curve points in the case of TrueType outlines. This point is the middle of the two off-curve points.

As the coordinates are still relative ones, the middle point is just obtained by dividing the coordinates vector by 2 and repeating it twice (with a carry for odd numbers).

When scaling, the original coordinates in FUnits are converted into pixel units. This step is done by multiplying the coordinates by the scale factor. The scale factor is calculated by the CPU and programmed by the G-Bus. This multiplication increases the range of X-Y coordinates (16-bit to19-bit).

This module also scales other FUnits data, instead of control points X-Y coordinates, and the bounding box of the glyph (xmin, xmax, ymin and ymax). This data is programmed by the G-Bus.

Once the coordinates of the control points of the outline have been scaled, the outline is drawn in a 64x64 pixel area (2-bpp). First, the relative coordinates are accumulated and added to the previous coordinates generating absolute coordinates. By this, both the types can be used to draw the outline lines and curves, and the lines or curves that do not get in the area are skipped.

After a step of the lines or curves division into elementary segments, each intersection of these segments forming the outline with the working area scan lines is calculated. In the outline memory a 2-bit information for each touched pixel is stored.

- Dropout mode: The pixel bit indicates if there is a contour boundary, and if the pixel is set, then the flag indicates the direction of the contour. If the pixel is not set and the flag is, then there will be a dropout.
- No dropout control: These 2 bits indicates the ‘weight’ of the contour direction (01: 1 contour direction A or 3 direction B, 10: 2 contours direction A or B, 11: 1 contour direction B or 3 direction A).

A dropout and an intersecting contour can not be managed at the same time, as both use the same flag. There can not be more than 3 intersecting contours (2-bit). When the outline drawing is completed in the area, the final bitmap is filled and written to the DRAM through W2 channel (1-bpp, black and white).

The font rendering the state machine supervises all the memory accesses required by the process. The state machine deals with several tasks:

- Divides the glyph correctly into 64x64 pixel working areas and provides the current coordinates to the outline drawing module.
- Controls the font data read from the DRAM or programmed by the G-Bus. If the FIFO does not contain this data, then the information is retrieved again for each working area.
- Controls the writing operation of the glyph areas to the DRAM when they are valid.
- Initializes the outline RAM at the first launch of the graphic accelerator
- Controls the sub-modules reset signal

Graphic Accelerator (Gradient Fill)

The gradient block is placed as an input of the bitmap graphic accelerator and is seen as a texture generator. This block does the gradient filling of a rectangular area of 2048 maximum pixel width. The gradient can be of two kinds linear or radial.

Linear graduation is along a direction. The direction can be the horizontal axis, the vertical axis or a combination of the horizontal and the vertical. The linear gradient is the superposition of a horizontal and a vertical gradient. The first color of the gradient is in the upper left hand corner, whereas the second color of the gradient is in the bottom right hand corner of a rectangle. This rectangle is not necessary the drawing window area. Usually the drawing window is inside this rectangle, but it can be larger and in that case the same pattern will be repeated.

The size of the rectangle, and the position of the second color, is given by the horizontal and the vertical scale factor: The scale factors are the inverse of the dimensions along the gradient. These scale factors parameter the orientation of the gradient direction, for horizontal gradient the scale factor 2 = 0 and for vertical gradient the scale factor = 0.

In a linear gradient mode, the color is pondered by the distance between the two color points. The horizontal and vertical gradient filling are symmetrical.

The radial graduation is defined by a center and two radius. The two radius define a ring. The ring is filled by a radial gradient from one color to another. The color outside the ring can be transparent or the perimeter color. The interior and the exterior can be set independently.

Main Mixer

The main mixer receives the picture streams from each of the five sources. Each incoming stream passes through a positioning block, which places the input picture at a specified horizontal and vertical position within the active display window.

Within the mixer, the positioned streams can be assigned to any of the layers 0-5. The highest priority (layer 5, the top layer) is always assigned to the hardware cursor. For each pixel, lower layers beginning with layer 1 are alpha-blended with higher layers. If any pixel is transparent, then the sort moves down the layer order until a non-transparent pixel is found. The layer 3 is alpha-blended with the final result to form the output pixel.

The output pixel stream ultimately consists of the merged combination of the four highest priority non-transparent layers. The output of the main mixer is sent to the display routing block. For each output pixel, the 6 input pixels are ordered according to the global plane order.

Positioning blocks

There are eight positioning blocks (one per input stream). The purpose of the positioning block is, to position the input picture within a large frame as shown in the diagram:

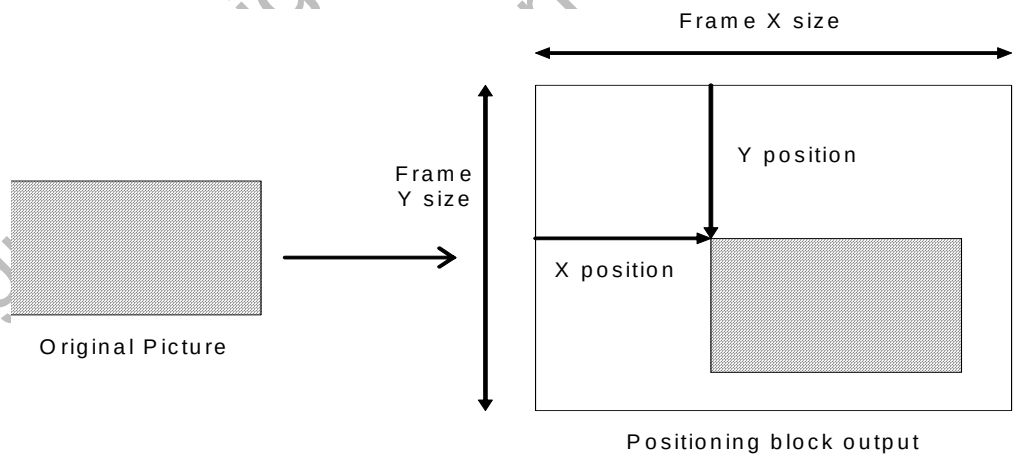


Figure 7-5. **Positioning blocks**

Each positioning block has four parameters:

1. A 13-bit signed X position (-4096 to 4095).
2. A 12-bit signed Y position (-2048 to 2047).
3. A 12-bit X active window size (0 to 4095).
4. A 11-bit Y active window size (0 to 2047).

The X and Y position parameters are signed to allow the top-left corner of the input picture to be positioned outside of the output frame (the left and/or top portion of the input picture is not displayed). If the discarded portion of the input picture is too large, then some bandwidth problems will appear. In this case, the input picture should be cut in the V-Bus interface.

Display Routing Block

The display router is essentially a crossbar switch, which couples the two input video streams to any of the two display output blocks. The two input sources to the display router are, the main mixer output and the color bars generator.

Each output (digital output /main analog output/CAV (component analog) output/CVBS output) is either disabled or assigned one source (labeled ‘00’ to ‘11’). Each source can thus have from 0 to 4 associated outputs. It should be made sure that the associated outputs (sharing the same source) are compatible.

The color bars generator generates the regular color bars pattern (in Y-Cb-Cr) for testing purposes. The image size and the intensity (75% or 100%) are programmable a. The white color can be PAL-like or NTSC-like. The generation is enabled as soon as the block is put in ‘run’ reset level. There are 8 bars of equal width per line.

Video Output Interface

Display Controllers

Each video output port may be independently operated as either a video timing master or a slave. In the master mode, the display controller generates the HSYNC and the VSYNC from an internal or external video clock. In the slave mode, the display controller receives the HSYNC and the VSYNC using either an internal or external video clock.

Commonly used output resolutions and frame refresh rates include:

- 704/720 x 480i @ 29.97 and 30Hz
- 704/720 x 480p @ 59.94 and 60Hz
- 704/720 x 576i @ 25Hz
- 704/720 x 576p @ 50Hz
- 1280 x 720p @ 50, 59.94 and 60Hz
- 1366 x 768p @ 50, 59.94 and 60Hz
- 1024 x 1024p @ 50, 59.94 and 60Hz

- 1920 x 1080i @ 25, 29.97 and 30Hz
- 1920 x 1080p @ 50, 59.94 and 60Hz

The output video may be optionally rotated by 90 degrees (xy swap). This enables low-cost PDA-style LCD panels to be used for portable video players.

As the video may be scaled to any resolution, these represent only the standard consumer resolutions. Resolutions unique to fixed-resolution displays, such as LCD, DLP and PDP are easily accommodated.

VBI Support on Video Outputs

The VBI data is present on the composite output, the Y channel of the s-video output and the Y channel of the YPbPr output. Two of the GPIO pins may be used for adding the appropriate DC offsets to pins 8 and 16 of a SCART connector to indicate the program aspect ratio information. The GPIO pins may also be used to control the Line 1, Line 2, and Line 3 signals, and monitor the Plug Detect signal, for the EIAJ CP-4120 DTerminal interface.

Closed Captioning

NTSC closed captioning on lines 21 and 284, as defined by EIA-608 is supported.

EIA-608 and ETSI EN 301 775 (DVB) closed captioning data present in the MPEG-2 bit-stream may be added to the 480i outputs on lines 21 and 284 when applicable.

Wide screen Signaling and CGMS

480i wide screen signaling (WSS) and copy generation management (CGMS-A) on lines 20 and 283, as defined by IEC 61880-1 and EIAJ CPR1204, is supported. CGMS-A data is also present on line 284 per EIA-608B.

480p wide screen signaling (WSS) and copy generation management (CGMS-A) on line 41, as defined by EIAJ CPR1204-1, EIA-805 (Oct. 2000) and IEC 61880-2, is supported.

576i wide screen signaling (WSS) and copy generation management (CGMS-A) on line 23, as defined by ETSI EN 300 294 and ITU-R BT.1119, is supported.

576p wide screen signaling (WSS) and copy generation management (CGMS-A) on line 43, as defined by IEC 62375, is supported.

EIA-805 (Oct. 2000) CGMS-A data is supported when in 720p or 1080i output mode.

Teletext

PAL teletext data may be output on lines 6-22 and 318-335 or NTSC teletext data may be output on lines 10-21 and 272- 284.

Digital Video Output

The digital video output has the following features:

- 8/16-bit (4:2:2) or 24-bit (4:4:4) digital output.
- Includes a 3x3 matrix multiplier to support color space conversion (YCbCr/RGB), brightness/contrast/saturation/hue controls and SD/HD colorimetry correction.
- LUT controlled gamma and color temperature control
- Optional DDR output (2x12-bit)

The digital output block controls an 8/16/24-bit digital video output interface. Within the block are provisions for color space conversion, brightness/contrast/saturation adjustment, 4:4:4 to 4:2:2 conversion, timing generation, and output data formatting.

The output interface can operate in 8, 16 or 24-bit mode. The maximum pixel clock frequency for the digital video output is 148.5MHz. So, output resolutions up to 1920x1080p60 can be supported. In the 8 or 16-bit modes, the output port can operate in BT.601 (separate HSYNC and VSYNC signals) or BT.656 (embedded sync using SAV/EAV codes) modes, or in the VIP 2.0-compatible mode. Support is also provided for a 12-bit, double-clocked output mode. It also supports the 27MHz 720(1440) x 480i and 720(1440) x 576i 24-bit YCbCr or RGB formats for HDMI compatibility. The unit contains a complete video timing generator for operating as a sync master, or the interface may be slaved to H/V sync timing from any of these sources, analog output or an external digital video device.

A 3x3 matrix multiplier with programmable coefficients provides optional YCbCr/RGB color space conversion. The matrix multiplier is followed by the addition of three programmable constants to the output components to provide optional brightness (or ‘black level’) control.

When outputting anything other than 480i or 576i video, the composite and S-video analog outputs are blanked. The supported YCbCr and RGB output formats include:

- 8-bit 4:2:2 YCbCr
- 16-bit 4:2:2 YCbCr
- 24-bit 4:4:4 YCbCr

- 24-bit 4:4:4 YCbCr (12-bit, 2x multiplexed over two words)
- 24-bit RGB
- 24-bit RGB (12-bit, 2x multiplexed over two words)

When the BT.601 format is used, the port can operate in the master or the slave timing mode. When the BT.656 format is used, the port can operate in the master timing mode only. A ‘valid video’ output signal is available for the BT.601, BT.656, and VIP 2.0 formats. VBI data is not present on this output port due to the variety of standards used. This port may also be used to interface to, and drive progressively, LCD and PDP panels within a digital television.

When generating the HDTV outputs, the ability to independently adjust the sync timing of this output port enables the adjusting of the HDMI embedded sync to align with either the falling edge or the center (rising edge) of the YPbPr tri-level sync. This can prevent the HDMI picture from being shifted by 2.3% relative to analog picture in some TVs.

When outputting RGB data, 3 programmable 256x8 lookup tables are available to adjust the gamma to match the requirements of the LCD and the PDP panels or provide black level adjustment. Additional circuitry is also provided to enable the color temperature and the white balance calibration.

Gamma Correction LUT

Gamma correction is achieved through 3 Look Up Tables (1 for each component R/G/B). The 3 LUT are accessed simultaneously by the GBUS master, but are addressed separately by the data flow.

Temperature Control LUT

After RGB gamma correction, a luminance component Y’ is derived for each (R’,G’,B’) pixel. The latter addresses a LUT containing delta (d) adjustments per component. Plasma televisions, in particular, tend to alter the R,G & B components resulting in black & white images showing some levels of color. The delta (d) adjustments are 2’s complement signed and have 8 bits each. Their range is therefore [-128,+127]. The output of the temperature control LUT is defined as follows:

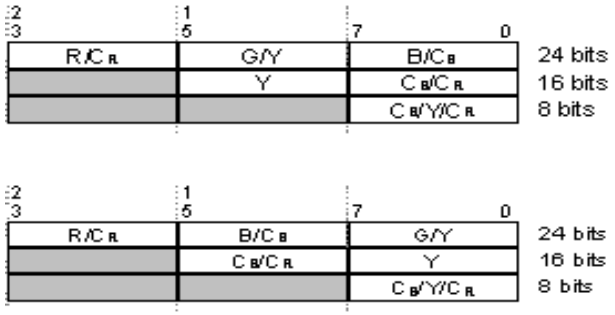
R’ -> R’ + dR’

G’ -> G’ + dG’

B’ -> B’ + dB’

Digital Video Output Formats

In the single data rate, depending on the selection made by the software either one of the following 2 configurations are supported.



In the double data rate, it is basically the first single data rate configuration, with the 12 lower bits sent first and the 12 upper bits sent later.

Timing Description

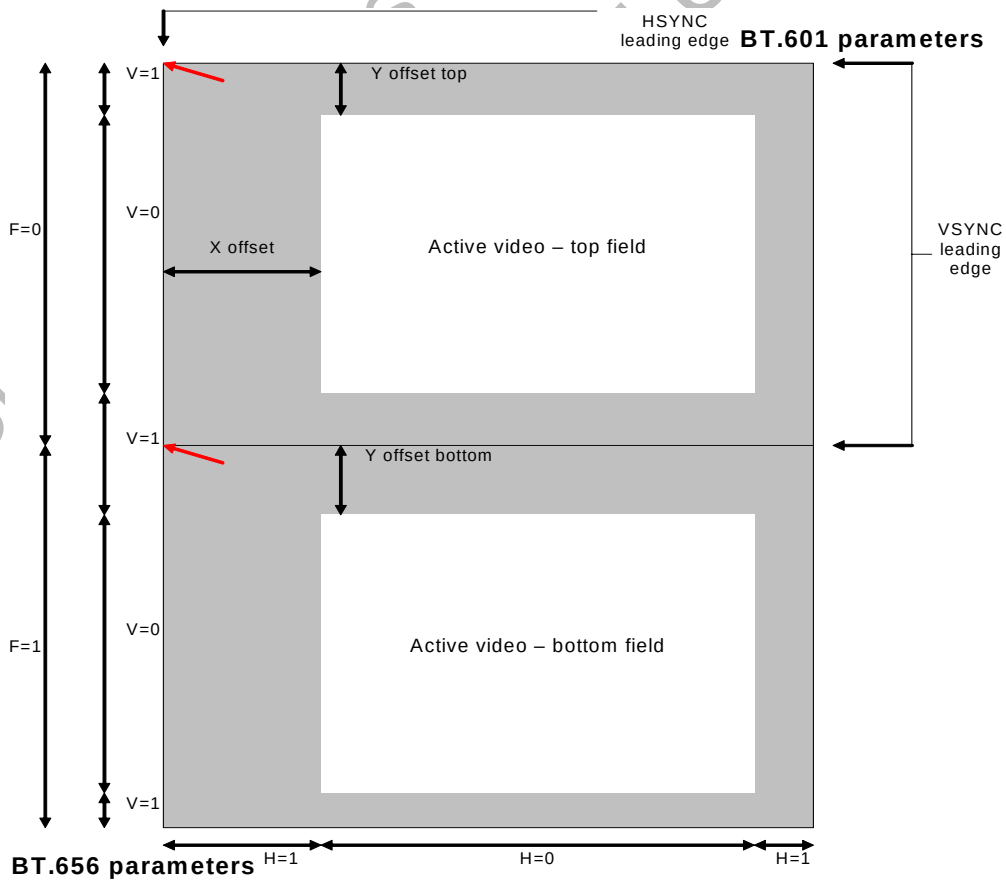


Figure 7-6. **Main digital output timing description**

Analog Video Output

The analog video output block controls 6 video output DACs to provide video in the composite, S-video (Y/C) and the component (RGB or YPbPr) formats. The block contains an independent video timing generator. It can also operate as a slave to other sync sources. The maximum sample rate of the 12-bit DACs is 148.5MHz, corresponding to a 1920x1080p60 format.

One set of three 12-bit DACs (the VO2 block) provides the component video output (selectable as either RGB or YPbPr). The other set of three 12-bit DACs (the VO1 block) is associated with the integrated NTSC/PAL TV encoder and provides simultaneous S-video (Y and C) and composite (CVBS) output. Each 3-DAC group can be independently disabled when not used for significant power savings.

Component Outputs

The component outputs support optional Macrovision v7.1.L1 and v1.2 protection and DAC attenuation compensation for 720p and 1080i/p output resolutions. When generating 480i or 576i video, the DACs operate at 54 MHz; for 480p or 576p video, the DACs operate at 108 MHz; for 1920x1080i or 1280x720p video, the DACs operate at 74.25 MHz. The analog video outputs are capable of driving a doubly-terminated 75-ohm load.

Supported component output formats are:

- RGB SCART
- RGB sync-on-green
- SMPTE GBR
- YUV Betacam
- YUV M2 SMPTE YUV
- Consumer YPbPr (no sync on Pb and Pr): EIA-775.1, EIA-775.2 and EIA-775.3

Independent brightness (or “black level”), contrast (or “picture” or “white level”), saturation (or “color”) and hue (or “tint”) controls are provided for the component output. Colorimetry conversion from BT.709 (HDTV) to BT.601 (SDTV) can also be performed.

The TV encoder includes several digital processing functionalities, such as SDTV/EDTV content 4x over-sampling, programmable color space conversion and brightness-contrast – saturation-hue control, and 8 levels sharpness control (along with notch filtering) for SDTV modes. It can be sync master or slave.

Configuration register bits can be used to select a number of features and characteristics:

- DAC enable or power-down
- H- and V-sync source
  - Video input port
  - Digital video output interface
  - Main analog output block (master)
- H- and V-sync polarity
- SDTV/HDTV mode
- Progressive/interlaced output (component output)
- Closed caption/extended closed caption enable/disable
- Sharpness/notch filter enable/disable and gain/attenuation control
- Luma and chroma filter options
- Composite mode video system: NTSC-M, PAL-B/G, PAL-60, PAL-M
- NTSC pedestal on/off
- Component output mode format
  - RGB SCART
  - RGB sync-on-green
  - SMPTE RGB
  - YUV Betacam
  - YUV M2 SMPTE YUV

Color space conversion and brightness/contrast/saturation control are provided as they are in the digital video output block.

There are two modes light display and full display. In the light display mode, the main analog output is the only analog output. It has 2 internal data paths (CVBS/S-video and CAV) with separated controls (different matrixes in particular). The CVBS/S-video data path and the CAV path (VO2) feed the 3 DACS (VO1). Each path can be enabled or disabled independently. There are therefore 3 possible modes:

1. CAV only enabled
2. CVBS/S-video only enabled
3. Both paths enabled

It is very important to notice that when both the paths are enabled, they display the same content at the same resolution. Since the CVBS/S-video supports only composite resolutions, the main analog output can handle only composite resolutions when the 2 data paths are simultaneously enabled (CAV = 480i and CVBS/S-video = NTSC or CAV = 576i and CVBS/S-video = PAL). Also when a path is disabled, its DACs are turned off.

Video Input Interfaces

The video input block inputs 8 or 16-bit video data and interfaces 656/601 (with VVLD or not) protocols. The video data is buffered and then sent to the V-Bus interface on a 16-bit bus. The video input port supports the receiving and storing of the externally-supplied digital video to the DRAM via an 8-bit input interface, using the Bt.656 protocols. This block can input 16/24/32-bpp data using either 656 or 601 protocols. The maximum input frequency is 148.5MHz.

Digital Video Input

This interface is designed to capture 8-bit 4:2:2 YCbCr digital video data in the BT.601 or BT.656 format from a NTSC/PAL decoder, 480p/576p YPbPr A/D converter or a CMOS camera chip. It supports clock rates up to 54MHz and resolutions of up to 720x576p. Sliced (binary) teletext data from select NTSC/PAL video decoders may also be captured and saved to memory for further processing. The closed captioning and wide screen signaling sliced (binary) VBI data may be read from the NTSC/PAL video decoder via the I^2C interface). An interrupt is generated at the beginning of each active video field or frame.

Video Format

The video format enables the interfacing to an external NTSC/PAL decoder or YPbPr A/D chip. Data in 8 or 16-bit YCbCr and in the BT.601 or BT.656 formats is supported. The 16-bit support requires the use of a digital video input.

Clock rates up to 75MHz and resolutions up to 1920x1080i or 1280x720p are supported. Capture of HD resolution data requires the use of the 16-bit interface and the use of a digital video input.

Sliced (binary) teletext data from select NTSC/PAL video decoders may also be captured and saved to memory for further processing. Closed captioning and wide screen signaling sliced (binary) VBI data may be read from the NTSC/PAL video decoder via the I^2C interface. An interrupt is generated at the beginning of active video each field or frame.

When the BT.601 format is used, the port can operate in master or slave timing mode. When the BT.656 format is used, the port can only operate in slave timing mode. A “valid video” input signal is available for both the BT.601 and BT.656 formats.

Graphics Format

The graphics format enables the interfacing to an external 3D graphics or DVI/HDMI receiver chip. 8, 16, 24, or 32-bit YCbCr or RGB video or graphics data in the BT.601 or BT.656 formats is supported.

Clock rates up to 150MHz and resolutions up to 1920x1080p are supported. The supported YCbCr and RGB input formats include:

- 8-bit 4:2:2 YCbCr
- 16-bit 4:2:2 YCbCr
- 24-bit 4:4:4 YCbCr
- 32-bit 4:4:4 YCbCr + alpha
- 16-bit ARGB (4444, 1555, or 565)
- 24-bit ARGB (888, 8565, or 5676)
- 24-bit RGB data (12-bit 2x multiplexed, 888 over two words, “C” and “I” version), Intel compatible
- 32-bit ARGB (8888)
- 32-bit RGB data (16-bit 2x multiplexed, 8888 over two words, “C” and “I” version)

When the BT.601 format is used, the port can operate in either the master or the slave timing mode. When the BT.656 format is used, the port can operate in the slave timing mode only. A ‘valid video’ input signal is available for both the BT.601 and BT.656 formats.

RGB input ranges of 0-255 (PC mode) or 16-235 (video mode) are supported. The 0-255 RGB data is converted to 16- 235 RGB data before mixing with the video.

Register Maps

Video Processing Registers

Multi-format Scaler Registers

Table 7-3. **Multi-format scaler registers**

Address[1]					
GFX	**CRT**	**VCR**	**Register Name[2]**	**R/W/A[3]**	**Description**
+0700	+0600	+0500	VO_XXX_FORMAT_HDS	R/W	Video Output GFX/CRT/VCR Format HDS Register
+0704	+0604	+0504	VO_XXX_OUTPUT_SIZE	R/W	Video Output GFX/CRT/VCR Output Size Register
+0708	+0608	+0508	VO_XXX_SCALE_FACTOR	R/W	Video Output GFX/CRT/VCR Scale Factor Register
+070C	+060C	+050C	VO_XXX_PHASE	R/W	Video Output GFX/CRT/VCR Phase Register
+0710	+0610	+0510	VO_XXX_ALPHA_ROUTING	R/W	Video Output GFX/CRT/VCR Alpha Routing Register
+0714	+0614	+0514	VO_XXX_KEY_COLOR	R/W	Video Output GFX/CRT/VCR Key Color Register
+0718	+0618	+0518	VO_XXX_BCS	R/W	Video Output GFX/CRT/VCR BCS Register
+071C	+061C	+051C	VO_XXX_STRIP_EDGE	R/W	Video Output GFX/CRT/VCR Strip Edge Register
+0720	+0620	+0520	VO_XXX_NONLINEAR_0	R/W	Video Output GFX/CRT/VCR Nonlinear 0 Register
+0724	+0624	+0524	VO_XXX_NONLINEAR_1	R/W	Video Output GFX/CRT/VCR Nonlinear 1 Register
+0728	+0628	+0528	VO_XXX_TILING	R/W	Video Output GFX/CRT/VCR Tiling Register

1. Address refers to G-Bus byte address relative to the video output base.
2. XXX is GFX/CRT/VCR depending on the address.
3. Read/Write/Auto update

Main Video Scaling Registers

Table 7-4. **Main video scaling registers**

Address[1]	Register Name	R/W/A[2]	Description
+0400	VO_MAIN_FORMAT_HDS	R/W	Video Output Main Format HDS Register
+0404	VO_MAIN_OUTPUT_SIZE	R/W	Video Output Main Output Size Register
+0408	VO_MAIN_SCALE_FACTOR	R/W	Video Output Main Scale Factor Register
+040C	VO_MAIN_SCALE_PHASE	R/W	Video Output Main Scale Phase Register
+0410	VO_MAIN_ALPHA_DEINT_ROUTING	R/W	Video Output Main Alpha Deint Routing Register
+0414	VO_MAIN_DEINT2	R/W	Video Output Main Deint 2 Register
+0418	VO_MAIN_BCS	R/W	Video Output Main BCS Register
+041C	VO_MAIN_PULLDOWN	R/W	Video Output Main Pull-down Register
+0420	VO_MAIN_STRIP_FILTER	R/W	Video Output Main Strip Filter Register
+0424	VO_MAIN_NONLINEAR_0	R/W	Video Output Main Nonlinear 0 Register
+0428	VO_MAIN_NONLINEAR_1	R/W	Video Output Main Nonlinear 1 Register

1. Address refers to G-Bus byte address relative to the video output base.
2. Read/Write/Auto update.

OSD Scaler Registers

Table 7-5. **OSD scaler registers**

Address[1]	Register Name	R/W/A[2]	Description
+0300	VO_OSD_FORMAT_HDS	R/W	Video Output OSD Format HDS Register
+0304	VO_OSD_OUTPUT_SIZE	R/W	Video Output OSD Output Size Register
+0308	VO_OSD_SCALE_FACTOR	R/W	Video Output OSD Scale Factor Register
+030C	VO_OSD_SCALE_PHASE_FLICKER	R/W	Video Output OSD Scale Phase Flicker Register
+0310	VO_OSD_ALPHA_ROUTING	R/W	Video Output OSD Alpha Routing Register
+0314	VO_OSD_KEY_COLOR	R/W	Video Output OSD Key Color Register
+9000	VO_OSD_LUT0	R/W	Video Output OSD LUT 0 Register

Table 7-5. **OSD scaler registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+9004	VO_OSD_LUT1	R/W	Video Output OSD LUT 1 Register
+90FC	VO_OSD_LUT255	R/W	Video Output OSD LUT 255 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Hardware Cursor Registers

Table 7-6. **Hardware cursor registers**

Address[1]	Register Name	R/W/A[2]	Description
+0100	VO_CURSOR_SIZE_CTRL	R/W	Video Output Cursor Size Control Register
+0140	VO_CURSOR_ LUT0	R/W	Video Output Cursor LUT 0 Register
+017C	VO_CURSOR_LUT15	R/W	Video Output Cursor LUT 15 Register
+8000	VO_CURSOR_PIX0	R/W	Video Output OSD Cursor Pixel0 Register
+87FC	VO_CURSOR_PIX511	R/W	Video Output OSD Cursor Pixel 511 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Main Mixer Registers

Table 7-7. **Main mixer registers**

Address[1]	Register Name	R/W/A[2]	Description
+0800	VO_MIX_GFX_POS	R/W	Video Output Main Mixer GFX Position Register
+0804	VO_MIX_ CRT_POS	R/W	Video Output Main Mixer CRT Position Register
+0808	VO_MIX_VCR_POS	R/W	Video Output Main Mixer VCR Position Register
+080C	VO_MIX_SP_POS	R/W	Video Output Main Mixer Sub-picture Position Register
+0810	VO_MIX_MV_POS	R/W	Video Output Main Mixer Main Video Position Register
+0814	VO_MIX_ OSD_POS	R/W	Video Output Main Mixer OSD Position Register
+0818	VO_MIX_GIN_POS	R/W	Video Output Main Mixer Position GIN Register

Table 7-7. **Main mixer registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+081C	VO_MIX_CUR_POS	R/W	Video Output Main Mixer Cursor Position Register
+0820	VO_MIX_INDEX	R/W	Video Output Index Register
+0824	VO_MIX_ FRAME _SIZE	R/W	Video Output Frame Size Register
+0828	VO_MIX_ BACKGROUND	R/W	Video Output Background Color Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Graphic Accelerator (Bitmap Mode) Registers

Table 7-8. **Graphic accelerator (bitmap mode) registers**

Address[1]	Register Name	R/W/A[2]	Description
+0A00	VO_GRAPH_ACC_X_FORMAT	R/W	Video Output Graphic Accelerator X Format Register
+0A04	VO_GRAPH_ACC_X_ALPHA	R/W	Video Output Graphic Accelerator X Alpha Register
+0A08	VO_GRAPH_ACC_X_KEYCOLOR	R/W	Video Output Graphic Accelerator X Key Color Register
+0A0C	VO_GRAPH_ACC_Y_FORMAT	R/W	Video Output Graphic Accelerator Y Format Register
+0A10	VO_GRAPH_ACC_Y_KEYCOLOR	R/W	Video Output Graphic Accelerator Y Key Color Register
+0A14	VO_GRAPH_ACC_CONTROL	R/W	Video Output Graphic Accelerator Control Register
+0A18	VO_GRAPH_ACC_FILL	R/W	Video Output Graphic Accelerator Fill Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Graphic Accelerator (Vectorial Mode) Registers

Table 7-9. **Graphic accelerator (vectorial mode) registers**

Address[1]	Register Name	R/W/A[2]	Description
+0A80	VO_GRAPH_ACC_MODE_CONTROL	R/W	Video Output Graphic Accelerator Mode Control Register
+0A84	VO_GRAPH_ACC_DRAM_READ_ACCESS	R/W	Video Output Graphic Accelerator DRAM Read Access Register
+0A88	VO_GRAPH_ACC_DRAM_WRITE_ACCESS	R/W	Video Output Graphic Accelerator DRAM Write Access Register
+0A8C	VO_GRAPH_ACC_X_BOUNDING_BOX	R/W	Video Output Graphic Accelerator X Bounding Box Register
+0A80	VO_GRAPH_ACC_Y_BOUNDING_BOX	R/W	Video Output Graphic Accelerator Y Bounding Box Register
+0A84	VO_GRAPH_ACC_SCALING_AND_CONTOURS	R/W	Video Output Graphic Accelerator Scaling and Contours Register
+0A88	VO_GRAPH_ACC_MATRIX_COEFFS	R/W	Video Output Graphic Accelerator Matrix Coefficients Register
+0A9C	VO_GRAPH_ACC_MATRIX_COEFFS	R/W	Video Output Graphic Accelerator Matrix Coefficients Register
+0AA0	VO_GRAPH_ACC_MATRIX_COEFFS	R/W	Video Output Graphic Accelerator Matrix Coefficients Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Video Input Registers

Video Input Registers

Table 7-10. **Video input registers**

Address[1]	Register Name	R/W/A[2]	Description
+0B00	VO_VID_IN_FORMAT2	R/W	VO Video Input Format Register
+0B34	VO_VID_IN_DATA_SIZE	R/W	VO Video Input Data Size Register
+0B04	VO_VID_IN_DATA_XOFFSET	R/W	VO Video Input Data X-offset Register
+0B08	VO_VID_IN_DATA_YOFFSET	R/W	VO Video Input Data Y-offset Register
+0B0C	VO_VID_IN_HZ_SYNC	R/W	VO Video Input Horizontal Sync Register
+0B10	VO_VID_IN_VT_SYNC	R/W	VO Video Input Vertical Sync Register
+0B14	VO_VID_IN_SYNC_COORD	R/W	VO Video Input Sync Coordinates Register
+0B18	VO_VID_IN_TOP_VBI	R/W	VO Video Input Top VBI Register

Table 7-10. **Video input registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+0B1C	VO_VID_IN_BOT_VBI	R/W	VO Video Input Bottom VBI Register
+0B20	VO_VID_IN_VBI_SIZE	R/W	VO Video Input VBI Size Register
+0B28	VO_VID_IN_VBI_VSM	R/W	VO Video Input VBI VSM Register
+0B2C	VO_VID_IN_VBI_VOFFSET	R	VO Video Input VBI V-offset Register
+0B30	VO_VID_IN_COUNTERS	R/W	VO Video Input Counters Register
+0B24	VO_VID_IN_COUNTERS2	R	VO Video Input Counters 2 Register
+0B38	VO_VID_IN_FORMAT2	R	VO Video Input Format 2 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Graphic Input Registers

Table 7-11. **Graphic input registers**

Address[1]	Register Name	R/W/A[2]	Description
+0C00	VO_GRAPH_IN_FORMAT	R/W	VO Video Input Format Register
+0C34	VO_GRAPH_IN_FORMAT2	R/W	VO Video Input Data Size Register
+0C04	VO_GRAPH_IN_ALPHA_ROUTING	R/W	VO Video Input Data X-offset Register
+0C08	VO_GRAPH_IN_KEY_COLOR	R/W	VO Video Input Data Y-offset Register
+0C0C	VO_GRAPH_IN_DATA_SIZE	R/W	VO Video Input DAta Size Register
+0C10	VO_GRAPH_IN_DATA_XOFFSET	R/W	VO Graphic Input Data X-offset Register
+0C14	VO_GRAPH_IN_DATA_YOFFSET	R/W	VO Graphic Input Data Y-offset Register
+0C18	VO_GRAPH_IN_HZ_SYNC	R/W	VO Graphic Input Top VBI Register
+0C1C	VO_GRAPH_IN_VT_SYNC	R/W	VO Graphic Input Vertical Sync Register
+0C20	VO_GRAPH_IN_SYNC_COORD	R/W	VO Graphic Input Sync Coordinates Register
+0C24	VO_GRAPH_IN_SYNC_OFFSET	R/W	VO Graphic Input Sync Offset Register
+0C28	VO_GRAPH_IN_TOP_VBI	R/W	VO Graphic Input Top VBI Register
+0C2C	VO_GRAPH_IN_BOT_VBI	R	VO Graphic Input Bottom VBI Register
+0C3C	VO_GRAPH_IN_VBI_SIZE	R/W	VO Graphic Input VBI Size Register
+0C40	VO_GRAPH_IN_VBI_VSM	R/W	VO Graphic Input VBI VSM Register

Table 7-11. **Graphic input registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+0C44	VO_GRAPH_IN_VBI_VOFFSET	R/W	VO Graphic Input VBI V-offset Register
+0C30	VO_GRAPH_IN_COUNTERS	R/W	VO Graphic Input Counters Register
+0C38	VO_GRAPH_IN_FORMAT2	R/W	VO Graphic Input Format 2 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Color Bars Generator Registers

Table 7-12. **Color bars generator registers**

Address[1]	Register Name	R/W/A[2]	Description
+0D00	VO_COLOR_BARS_CTRL	R/W	Video Output Color Bars Control Register
+0D04	VO_COLOR_BARS_SIZE	R/W	Video Output Color Bars Size Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Display Routing Register

Table 7-13. **Display routing register**

Address[1]	Register Name	R/W/A[2]	Description
+1208	VO_ROUTING_CTRL	R/W	Video Output Routing Control Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Video Output Registers

Digital Output Registers

Table 7-14. **Digital output registers**

Address[1]	Register Name	R/W/A[2]	Description
+0E00	VO_DIGIT_OUT_CONV0	R/W	VO Digital Output Conversion 0 Register
+0E04	VO_DIGIT_OUT_CONV1	R/W	VO Digital Output Conversion 1 Register
+0E08	VO_DIGIT_OUT_CONV2	R/W	VO Digital Output Conversion 2 Register
+0E0C	VO_DIGIT_OUT_CONV3	R/W	VO Digital Output Conversion 3 Register
+0E10	VO_DIGIT_OUT_CONV4	R/W	VO Digital Output Conversion 4 Register
+0E14	VO_DIGIT_OUT_CONV5	R/W	VO Digital Output Conversion 5 Register
+0E20	VO_DIGIT_OUT_FORMAT	R/W	VO Digital Output Format Register
+0E24	VO_DIGIT_OUT_XOFFSET	R/W	VO Digital Output X-offset Register
+0E28	VO_DIGIT_OUT_YOFFSET	R/W	VO Digital Output Y-offset Register
+0E2C	VO_DIGIT_OUT_HZ_SYNC	R/W	VO Digital Output Horizontal Sync Register
+0E30	VO_DIGIT_OUT_VT_SYNC	R/W	VO Digital Output Vertical Sync Register
+0E34	VO_DIGIT_OUT_VSYNC_COORD	R/W	VO Digital Output Vsync Coordinates Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Main Analog Output Registers

Table 7-15. **Main analog output registers**

Address[1]	Register Name	R/W/A[2]	Description
+0F00	VO_MAIN_ANALOG_CONV0	R/W	VO Main Analog Output Conversion 0 Register
+0F04	VO_MAIN_ANALOG_CONV1	R/W	VO Main Analog Output Conversion 1 Register
+0F08	VO_MAIN_ANALOG_CONV2	R/W	VO Main Analog Output Conversion 2 Register
+0F0C	VO_MAIN_ANALOG_CONV3	R/W	VO Main Analog Output Conversion 3 Register

Table 7-15. **Main analog output registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+0F10	VO_MAIN_ANALOG_CONV4	R/W	VO Main Analog Output Conversion 4 Register
+0F14	VO_MAIN_ANALOG_CONV5	R/W	VO Main Analog Output Conversion 5 Register
+0F18	VO_MAIN_ANALOG_XOFFSET_FIELD	R/W	VO Main Analog Output X-offset Field Register
+0F1C	VO_MAIN_ANALOG_OFFSET	R/W	VO Main Analog Output Offset Register
+0F20	VO_MAIN_ANALOG_CVBS_CONV0	R/W	VO Main Analog Output CVBS Conversion 0 Register
+0F24	VO_MAIN_ANALOG_CVBS_CONV1	R/W	VO Main Analog Output CVBS Conversion 1 Register
+0F28	VO_MAIN_ANALOG_CVBS_CONV2	R/W	VO Main Analog Output CVBS Conversion 2 Register
+0F2C	VO_MAIN_ANALOG_CVBS_CONV3	R/W	VO Main Analog Output CVBS Conversion 3 Register
+0F20	VO_MAIN_ANALOG_CVBS_CONV4	R/W	VO Main Analog Output CVBS Conversion 4 Register
+0F24	VO_MAIN_ANALOG_CVBS_CONV5	R/W	VO Main Analog Output CVBS Conversion 5 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Component Analog Output Registers

Table 7-16. **Component analog output registers**

Address[1]	Register Name	R/W/A[2]	Description
+1000	VO_COMPONENT_OUT_CONV0	R/W	VO Component Output Conversion 0 Register
+1004	VO_COMPONENT_OUT_CONV1	R/W	VO Component Output Conversion 1 Register
+1008	VO_COMPONENT_OUT_CONV2	R/W	VO Component Output Conversion 2 Register
+100C	VO_COMPONENT_OUT_CONV3	R/W	VO Component Output Conversion 3 Register
+1010	VO_COMPONENT_OUT_CONV4	R/W	VO Component Output Conversion 4 Register
+1014	VO_COMPONENT_OUT_CONV5	R/W	VO Component Output Conversion 5 Register
+1018	VO_COMPONENT_OUT_XOFFSET_FIELD	R/W	VO Component Output X-offset Field Register

Table 7-16. **Component analog output registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+101C	VO_COMPONENT_OUT_YOFFSET	R/W	VO Component Output Y-Offset Register
+1040	VO_COMPONENT_OUT_TV_CONFIG	R/W	VO Component Output TV Configuration Register
+1044	VO_COMPONENT_OUT_TV_SIZE	R/W	VO Component Output TV Size Register
+1048	VO_COMPONENT_OUT_TV_HSYNC	R/W	VO Component Output TV Hsync Register
+104C	VO_COMPONENT_OUT_TV_VSYNC_O_0	R/W	VO Component Output TV Vsync O 0 Register
+1050	VO_COMPONENT_OUT_TV_VSYNC_O_1	R/W	VO Component Output TV Vsync O 1 Register
+1054	VO_COMPONENT_OUT_TV_VSYNC_E_0	R/W	VO Component Output TV Vsync E 0 Register
+1058	VO_COMPONENT_OUT_TV_VSYNC_E_1	R/W	VO Component Output TV Vsync E 1 Register
+105C	VO_COMPONENT_OUT_TV_HD_HSYNC_INFO	R/W	VO Component Output TV HD Hsync Information Register
+1060	VO_COMPONENT_OUT_TV_HD_VSYNC	R/W	VO Component Output TV HD Vsync Register
+1064	VO_COMPONENT_OUT_TV_CGMS	R/W	VO Component Output TV CGMS Register
+1068	VO_COMPONENT_OUT_TV_CC_AGC	R/W	VO Component Output TV CC AGC Register
+106C	VO_COMPONENT_OUT_TV_TEST_CONFIG	R/W	VO Component Output TV Test Configuration Register
+1080	VO_COMPONENT_OUT_TV_MV_N_0_22	R/W	VO Component Output TV MV N 0, 22 Register
+1084	VO_COMPONENT_OUT_TV_MV_N_1_2_3_4	R/W	VO Component Output TV MV N 1, 2, 3, 4 Register
+1088	VO_COMPONENT_OUT_TV_MV_N_5_6_7_8	R/W	VO Component Output TV MV N 5, 6, 7, 8 Register
+108C	VO_COMPONENT_OUT_TV_MV_N_9_10_11	R/W	VO Component Output TV MV N 9, 10, 11 Register
+1090	VO_COMPONENT_OUT_TV_MV_N_12_13_14	R/W	VO Component Output TV MV N 12, 13, 14 Register
+1094	VO_COMPONENT_OUT_TV_MV_N_15_16_17_18	R/W	VO Component Output TV MV N 15, 16, 17, 18 Register
+1098	VO_COMPONENT_OUT_TV_MV_N_19_20_21	R/W	VO Component Output TV MV N 19, 29, 21 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Composite Analog Output Registers

Table 7-17. **Composite analog output registers**

Address[1]	Register Name	R/W/A[2]	Description
+1100	VO_COMPOSITE_OUT_BCS	R/W	VO Composite Output BCS Register
+1104	VO_COMPOSITE_OUT_XOFFSET	R/W	VO Composite Output X-offset Register
+1108	VO_COMPOSITE_OUT_YOFFSET	R/W	VO Composite Output Y-offset Register
+1140	VO_COMPOSITE_OUT_TV_CONFIG	R/W	VO Composite Output TV Configuration Register
+1144	VO_COMPOSITE_OUT_TV_SIZE	R/W	VO Composite Output TV Size Register
+1148	VO_COMPOSITE_OUT_TV_HSYNC	R/W	VO Composite Output TV Hsync Register
+114C	VO_COMPOSITET_OUT_TV_VSYNC_O_0	R/W	VO Composite Output TV Vsync O 0 Register
+1150	VO_COMPOSITE_OUT_TV_VSYNC_O_1	R/W	VO Composite Output TV Vsync O 1 Register
+1154	VO_COMPOSITE_OUT_TV_VSYNC_E_0	R/W	VO Composite Output TV Vsync E 0 Register
+1158	VO_COMPOSITE_OUT_TV_VSYNC_E_1	R/W	VO Composite Output TV Vsync E 1 Register
+1164	VO_COMPOSITE_OUT_TV_CGMS	R/W	VO Composite Output TV CGMS Register
+1168	VO_COMPOSITE_OUT_TV_CC_AGC	R/W	VO Composite Output TV CC AGC Register
+116C	VO_COMPOSITE_OUT_TV_TEST_CONFIG	R/W	VO Composite Output TV Test Configuration Register
+1180	VO_COMPOSITE_OUT_TV_MV_N_0_22	R/W	VO Composite Output TV MV N 0, 22 Register
+1184	VO_COMPOSITE_OUT_TV_MV_N_1_2_3_4	R/W	VO Composite Output TV MV N 1, 2, 3, 4 Register
+1188	VO_COMPOSITE_OUT_TV_MV_N_5_6_7_8	R/W	VO Composite Output TV MV N 5, 6, 7, 8 Register
+118C	VO_COMPOSITE_OUT_TV_MV_N_9_10_11	R/W	VO Composite Output TV MV N 9, 10, 11 Register
+1190	VO_COMPOSITE_OUT_TV_MV_N_12_13_14	R/W	VO Composite Output TV MV N 12, 13, 14 Register
+1194	VO_COMPOSITE_OUT_TV_MV_N_15_16_17_18	R/W	VO Composite Output TV MV N 15, 16, 17, 18 Register
+1198	VO_COMPOSITE_OUT_TV_MV_N_19_20_21	R/W	VO Composite Output TV MV N 19, 29, 21 Register

1. Address refers to G-Bus byte address relative to the video output block base.
2. Read/Write/Auto update.

Pin Description

Digital Video Input Pins

Table 7-18. **Digital video input pin descriptions**

Pin Name	Ball ID	Direction	Description
VI1_P7	E4	I	Video input port pixel bus bit 7
VI1_P6	G5	I	Video input port pixel bus bit 6
VI1_P5	D3	I	Video input port pixel bus bit 5
VI1_P4	F4	I	Video input port pixel bus bit 4
VI1_P3	E3	I	Video input port pixel bus bit 3
VI1_P2	D2	I	Video input port pixel bus bit 2
VI1_P1	F3	I	Video input port pixel bus bit 1
VI1_P0	D1	I	Video input port pixel bus bit 0
VI1_CLK	F1	I	Video input port pixel clock signal. Active edge programmable.

Digital Video Output Pins

Digital Video Output Pins

Table 7-19. **Digital video output pin descriptions**

Pin Name	Ball ID	Direction	Description
VO0_P23	G4	O	Video output pixel bus bit 23
VO0_P22	H4	O	Video output pixel bus bit 22
VO0_P21	G3	O	Video output pixel bus bit 21
VO0_P20	G2	O	Video output pixel bus bit 20
VO0_P19	J5	O	Video output pixel bus bit 19
VO0_P18	H3	O	Video output pixel bus bit 18
VO0_P17	G1	O	Video output pixel bus bit 17
VO0_P16	H2	O	Video output pixel bus bit 16
VO0_P15	J3	O	Video output pixel bus bit 15
VO0_P14	H1	O	Video output pixel bus bit 14
VO0_P13	J4	O	Video output pixel bus bit 13
VO0_P12	K4	O	Video output pixel bus bit 12

Table 7-19. **Digital video output pin descriptions**

Pin Name	Ball ID	Direction	Description
VO0_P11	J2	O	Video output pixel bus bit 11
VO0_P10	J1	O	Video output pixel bus bit 10
VO0_P9	K3	O	Video output pixel bus bit 9
VO0_P8	K2	O	Video output pixel bus bit 8
VO0_P7	L5	O	Video output pixel bus bit 7
VO0_P6	K1	O	Video output pixel bus bit 6
VO0_P5	L4	O	Video output pixel bus bit 5
VO0_P4	L3	O	Video output pixel bus bit 4
VO0_P3	L2	O	Video output pixel bus bit 3
VO0_P2	L1	O	Video output pixel bus bit 2
VO0_P1	N5	O	Video output pixel bus bit 1
VO0_P0	M4	O	Video output pixel bus bit 0
VO0_HS	M3	B	Video output port Hsync input or output. Polarity programmable.
VO0_VS	M2	B	Video output port Vsync input or output. Polarity programmable.
VO0_VLD	N4	O	Video output port Data Valid signal. Active high.
VO0_CLK	M1	O	Video output port pixel clock signal

Main Analog Video Output Pins

Table 7-20. **Main analog output pin descriptions**

Pin Name	Ball ID	Direction	Description
VO1_Y	N1	O	Analog video output. Output Y (luminance) signals in the component YUV or S-video mode, or G signal in the component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO1_U	R1	O	Analog video output. Outputs Pb signal in the component YUV mode, the composite signal in S-video mode, or B signal in component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO1_V	P1	O	Analog video output. Outputs Pr signal in component YUV mode, C (chrominance) signal in S-video mode, or R signal in component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO1_RSET	R5	O	DAC current set pin. A resistor (140 ohm, 1% tolerance) connected between this pin and the ground sets the full-scale DAC current.
VO1_VREF	T5	B	Video DAC current source reference voltage (1.25V nominal)
VO1_AVDD	R4	I	Video output analog block 3.3V power supply connection
VO1_AVDD_Y	N3	I	Y-channel DAC power supply. Connect to a 3.3V nominal supply.
VO1_AVDD_U	R3	I	U-channel DAC power supply. Connect to a 3.3V nominal supply.
VO1_AVDD_V	P3	I	V-channel DAC power supply. Connect to a 3.3V nominal supply.
VO1_AVSS	P4	I	Video output analog block ground connection
VO1_AVSS_Y	N2	I	Y-channel DAC ground connection
VO1_AVSS_U	R2	I	U-channel DAC ground connection
VO1_AVSS_V	P2	I	V-channel DAC ground connection

Component Analog Output Pins

Table 7-21. **Component analog output pin descriptions**

Pin Name	Ball ID	Direction	Description
VO2_Y	T1	O	Analog video output. Outputs Y (luminance) signal in component YUV mode, or G signal in the component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO2_U	V1	O	Analog video output. Outputs Pb signal in component YUV mode, or B signal in component the RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO2_V	U1	O	Analog video output. Outputs Pr signal in component YUV mode, or R signal in the component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO2_RSET	U5	O	DAC current set pin. A resistor (140 ohm, 1% tolerance) connected between this pin and the ground sets the full-scale DAC current.
VO2_VREF	T4	B	Video DAC current source reference voltage (1.25V nominal)
VO2_AVDD	U4	I	Video output analog block 3.3V power supply connection
VO2_AVDD_Y	T3	I	Y-channel DAC power supply. Connect to a 3.3V nominal supply.
VO2_AVDD_U	V3	I	U-channel DAC power supply. Connect to a 3.3V nominal supply.
VO2_AVDD_V	U3	I	V-channel DAC power supply. Connect to a 3.3V nominal supply.
VO2_AVSS	V4	I	Video output analog block ground connection
VO2_AVSS_Y	T2	I	Y-channel DAC ground connection
VO2_AVSS_U	V2	I	U-channel DAC ground connection
VO2_AVSS_V	U2	I	V-channel DAC ground connection

Electrical Characteristics

Digital Video Input DC Characteristics

Table 7-22. **Digital video input DC characteristics**

Symbol	Description	Units	Min	Typ	Max
V_{IH}	Input high voltage	V	2		5.5
V_{IL}	Input low voltage	V	-0.3		0.8
R_{PU}[1]	Pullup resistor value	kohms	49	63	93

1. Parameter applies to all pins VI1 pins.

Digital Video Output DC Characteristics

Table 7-23. **Digital video output DC electrical characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}(1)	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
I_{OH}[2]	High level output current (@V_{OH} = 2.4V)	mA	25	48	74
I_{OL}(2)	Low level output current (@V_{OL} = 0.4V)	mA	18	28	33
V_{IH}[3]	Input high voltage	V	2.0		5.5
V_{IL}(3)	Input low voltage	V	-0.3		0.8

1. Parameter applies to pins VO0_P(23:0), VO0_VLD. Also VO0_HS, VO0_VS when configured as outputs.
2. Parameter applies to pins VO0_CLK.
3. Parameter applies to pins VO0_HS, VO0_VS when configured as inputs.

Analog Video Output DC Characteristics

Table 7-24. **Analog video output electrical characteristics**

Symbol	Description	Units	Min	Typ	Max
	DAC resolution	bits	12	12	12
R_{SET}	Current set resistor	W		140	
I_{FS}	Full scale output current	mA			
	Video level error (R_{SET} tolerance 1% or less)	%			5
AVDD	DAC power supply	V	3.15	3.3	3.45

Timing Diagrams

Digital Video Timing (Graphic/Video Inputs, Digital Output)

Setup and Hold Times

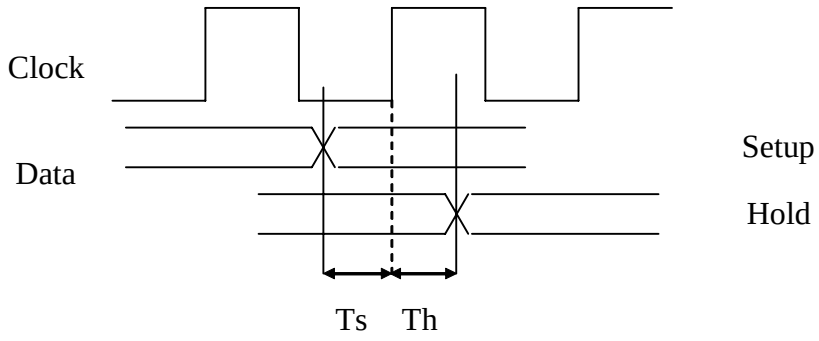


Figure 7-7. **Setup and hold timing diagram**

Ts = 1 ns (minimum), Th = 0 ns (minimum).

HSYNC/VSYNC Timing

Interlaced Mode, Top/Bottom Signalization

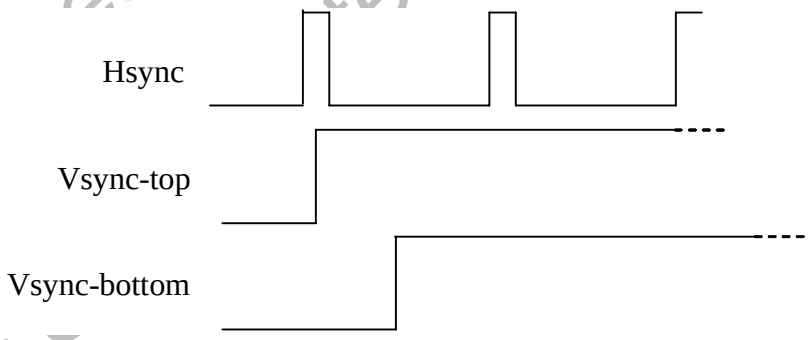


Figure 7-8. **HSYNC/VSYNC timing diagram (interlaced mode, top/bottom signalization)**

In this diagram both the sync pulses are active high. Both active low and active high polarities for the Hsync and the Vsync are supported.

In the case of a top field, the Vsync leading edge (low to high in this diagram) occurs while the Hsync is active. This includes the standard signalization where the Hsync and the Vsync leading edges occur during the same clock cycle. A top field is signaled by FIELD_ID = 0. In the case of a bottom field, the Vsync leading edge occurs while the Hsync is inactive. A bottom field is signaled by FIELD_ID = 1.

Progressive mode

The Hsync/Vsync signalization in the progressive mode is similar to the top field signalization in the interlaced mode.

Sync Generation

The sync generators use the following parameters:

The HZ_TOTAL_SIZE is the number of cycles between 2 Hsyncs, that is the number of cycles included in a line.

The VT_TOTAL_SIZE is the number of lines (or Hsyncs) between 2 Vsyncs of the same field ID (in a frame).

In the interlaced mode, the TOP_FIELD_HEIGHT is the number of half lines between a top and the next bottom Vsyncs.

The HS_POLARITY and VS_POLARITY allow to control the polarity of the Hsync and Vsync pulses.

Note: Typically, in the interlaced mode, VT_TOTAL_SIZE is an odd number and TOP_FIELD_HEIGHT = VT_TOTAL_SIZE. In any case, the TOP_FIELD_HEIGHT should always be odd. In the progressive mode, TOP_FIELD_HEIGHT = 0.

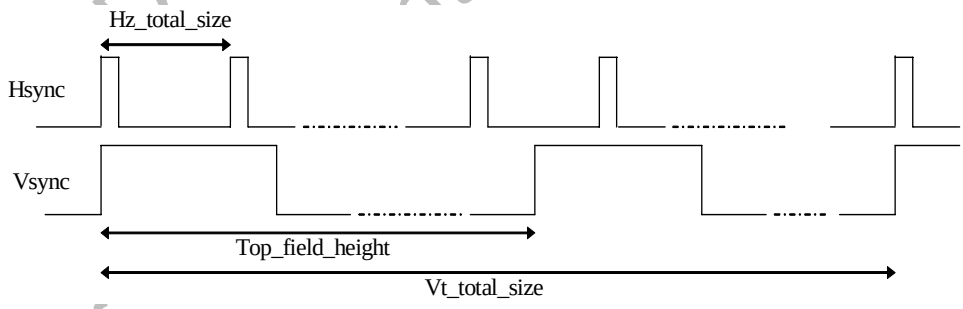


Figure 7-9. **Active high Hsync and Vsync in the interlaced mode**

Active Picture Positioning

A picture is a field in the interlaced mode and a frame in the progressive mode. The position of a picture relative to the Hsync and the Vsync in all the inputs (601 mode) and the output is achieved through the registers X_OFFSET, Y_OFFSET_TOP (used for the interlaced top fields and in the progressive mode) and Y_OFFSET_BOTTOM (used only in the interlaced mode).

The register X_OFFSET positions the start of the active video relative to the leading edge of the Hsync (in clock cycles). The register Y_OFFSET positions the start of the active video relative to the leading edge of the Vsync (in lines). The parameter Y_OFFSET_BOTTOM is used relative to the Hsync leading edge following the Vsync leading edge. The use of these parameters is illustrated in the following diagram:

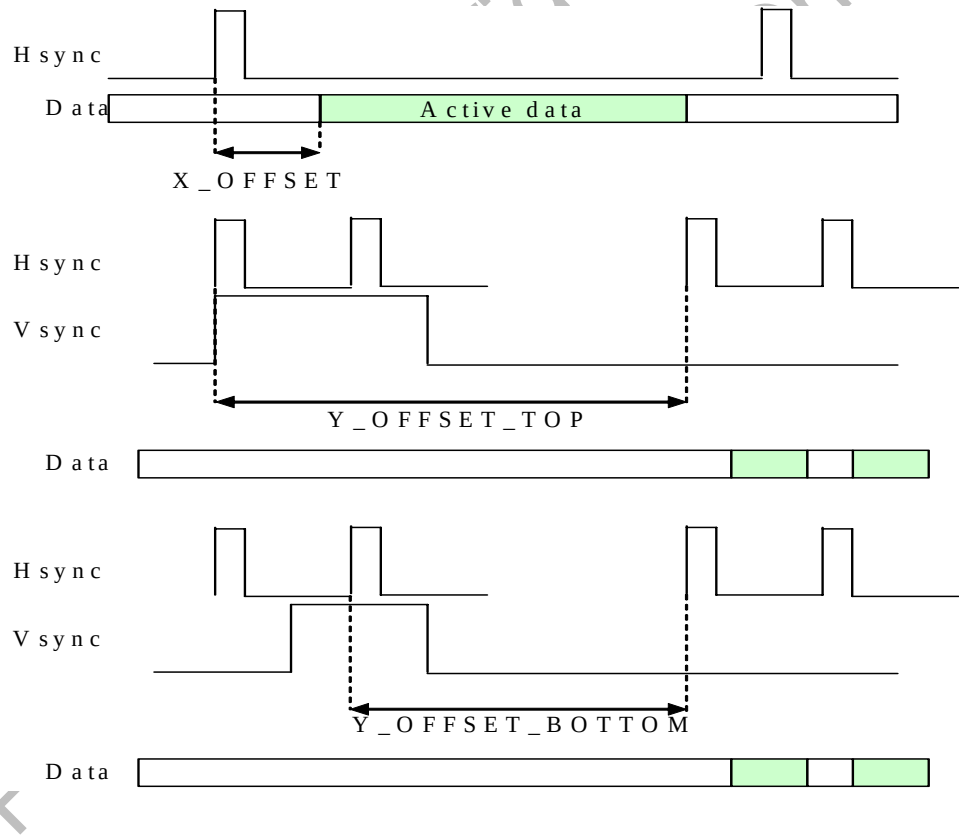


Figure 7-10. **Active picture positioning timing diagram**

Active Picture Dimensions

In the digital output, the dimensions of the active picture are implied by the stream of data input by the output blocking (coming from the router).

In the inputs, in the 656 mode, the picture dimensions are indicated by the SAV/EAV flags. In the 601 mode, the picture dimensions are specified by the X size (size of an active line in pixel clocks) and the Y size (number of lines of active picture in Hsyncs).

8 Audio Processing Subsystem

Block Diagram of Audio Processing Subsystem

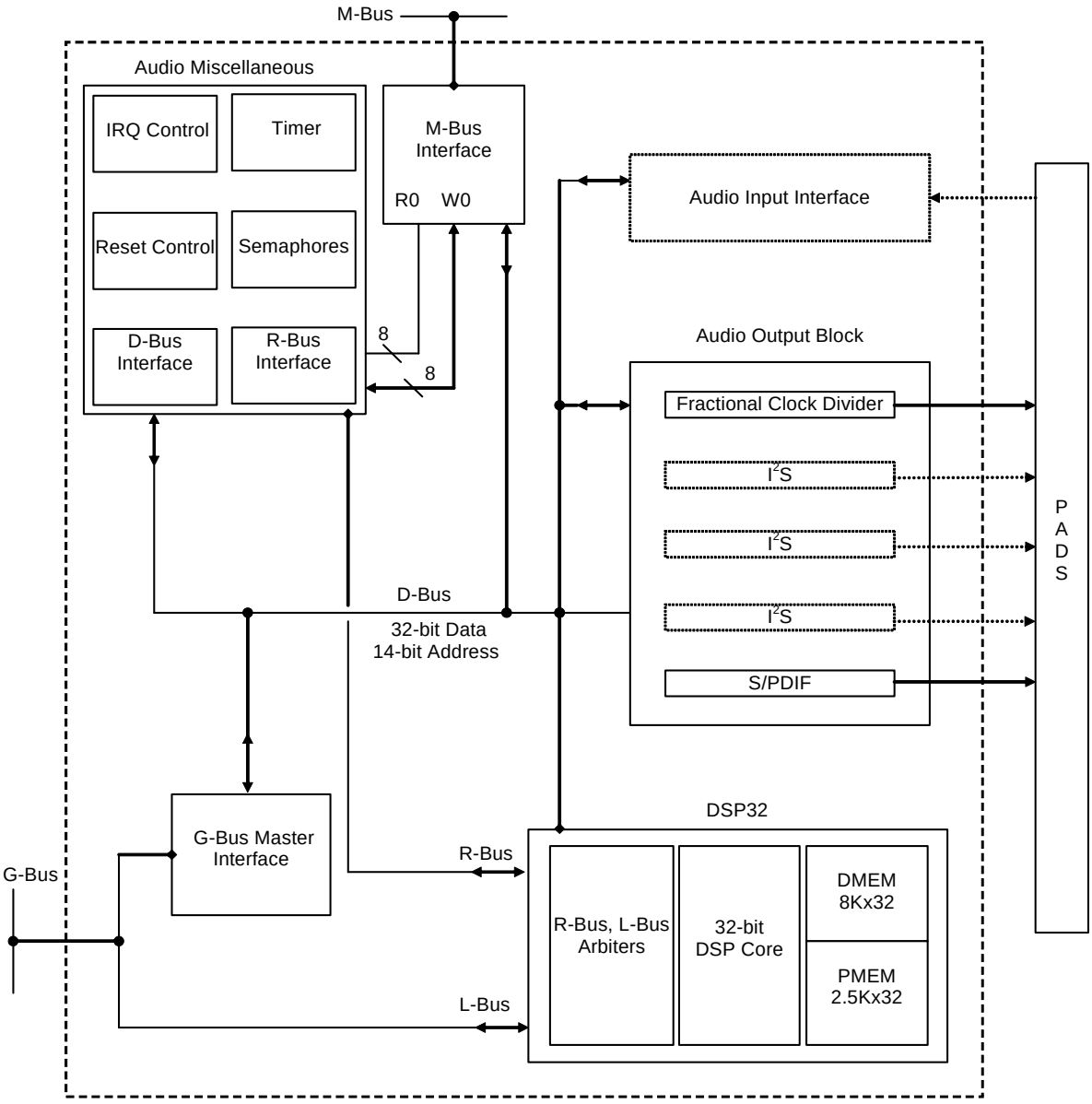


Figure 8-1. **Audio processing subsystem block diagram**

Introduction

The EM8622L contains an integrated audio subsystem based on a custom-designed 32-bit digital signal processor (DSP). The audio decoding and processing algorithms are implemented on the DSP. This firmware-based approach gives great flexibility for accommodating future audio standards or specialized audio requirements.

The decoder is capable of supporting the following audio formats. In most cases, in order to receive code for a specific audio codec, the user must be a codec licensee. A list of currently available audio codecs is available from Sigma Designs.

- MPEG-1 and MPEG-2 Layers I, II and III (MP3) 2.0
- MPEG-2 BC multi-channel Layers I, II and III 5.1
- MPEG-2 and MPEG-4 AAC-LC 2.0
- Dolby Digital 5.1
- DTS 5.1
- WMA9@L3 2.0, WMA9 Pro@M2 5.1

Features

- Custom built 32-bit DSP
- 7.1-channel audio digital output
- Supports one S/PDIF (IEC 60958) digital audio output
- Supports I^2S digital audio input
- Supports audio services required for the digital TV applications

Functional Description

The audio block of the EM8622L contains a 32-bit DSP, with the following peripherals mapped on the D-Bus:

- Audio Miscellaneous: Includes the DMA logic to transfer data between the DSP program/data memories and the DRAM (operates together with the M-Bus interface). Also contains a programmable timer, reset/irq control registers and semaphores.
- M-Bus interface: The DRAM side of the RISC to/from the DRAM.
- G-Bus interface: Allows the DSP to become a G-Bus master and access the G-Bus resources.
- Audio input: Receives a two channel serial audio stream (I^2S or S/PDIF formats). The audio data is shifted into registers that can be read by the DSP.
- Audio output: Sends an eight/six/two channel serial audio stream (I^2S or S/PDIF formats). The audio data is shifted from registers that is written by the DSP.

Audio Processing Capabilities

The audio unit provides 3 I^2S output channels, one S/PDIF output channel, and one I^2S audio input channel. The audio decoder supports the following audio formats:

For MPEG-1 and MPEG-2 Layers I, II and III, bit rates up to 448kbps (Layer I), 384kbps (Layer II) or 320kbps (Layer III) are supported with sample rates of 16, 22.05, 24, 32, 44.1 and 48KHz. Single channel, dual channel, joint stereo, and stereo modes are supported.

For MPEG-2 and MPEG-4 AAC-LC, bit rates up to 384kbps are supported with sample rates of 7.35, 8, 11.025, 12, 16, 22.05, 24, 32, 44.1 and 48KHz.

For Dolby Digital, bit rates up to 640kbps are supported with sample rates of 32, 44.1 and 48KHz. Multi-channel (up to 5.1) operation is supported with optional downmixing to 2 channel Dolby Pro Logic.

For DTS, bit rates of 384, 768 and 1509kbps are supported with sample rates of 8, 16, 22.05, 24, 32, 44.1, 48, 96 and 192KHz. Multi-channel (up to 5.1) operation is supported with optional downmixing to two-channel.

WMA9 CBR bit rates up to 192Kbps, WMA9 VBR bit rates up to 360Kbps and WMA9 Pro (CBR and VBR) bit rates up to 768Kkbps are supported. Sample rates of 44.1, 48 and 96KHz are supported.

PCM audio processing supports 16, 20 and 24 bits per sample, up to 5.1 channels with optional downmixing to 2 channels, and sample rates up to 192KHz (2.0 channels) or 96KHz (5.1 channels).

A bilingual mode is also supported (English on right or ‘sub’ channel and another language on the left or ‘main’ channel). Either language may be output onto both the left and right channels.

As the EM8622L does not support 6.1-channel audio outputs, Dolby Digital EX is decoded to 5.1 channels since it is compatible with Dolby Digital 5.1. Alternately, the S/PDIF output may be used to connect to a Dolby Digital EX compatible surround sound receiver for decoding the 6.1-channel audio.

Additional capabilities include:

- Downloading of new firmware
- Mixing I^2S or S/PDIF input with decoded audio
- Mixing LPCM data with dual decoded amd mixed audio
- Adjust output level of each channel in each codec at downmix
- When decoding dual audio streams, audio S/PDIF and I^2S outputs can be from either stream, or from a mixed stream
- 2x and 4x upsampling
- De-emphasis for 48 and 44.1kHz sampled audio
- Attenuate the audio
- Convert from 44.1ksps to 48ksps LPCM

DTV Audio Services

The supported audio services for the DTV include:

- 1.0, 2.0 or 7.1 channels of complete main (CM) mixed with 1.0 channels of either visually impaired (VI), hearing impaired (HI), commentary (C) or voice-over (VO)
- 1.0, 2.0 or 7.1 channels of music and effects (ME) mixed with 1.0 or 2.0 channels of dialog (D) and also mixed with 1.0 channels of either visually impaired (VI), hearing impaired (HI), commentary (C) or voice-over (VO)
- 1.0, 2.0 or 7.1 channels of visually impaired (VI), hearing impaired (HI) or commentary (C)
- 1.0 channels of emergency (E)

When the 1.0 channel visually impaired (VI), hearing impaired (HI), commentary (C) or voice-over (VO) is available as an associated service, it may be mixed with the 7.1 center channel audio (7.1 output mode) or with the stereo audio (2.0 output mode).

During an emergency message all other audio services are muted and the emergency audio is output onto both the primary and the secondary stereo audio outputs (all the other output channels are muted).

All the audio services use the same audio compression format. The supported compression formats include Dolby Digital (AC3), MPEG-1, MPEG-2 and MPEG-4. The user adjustable dynamic range controls and mixing controls for the main audio and associated audio services are also supported.

The audio block connects to other EM8622L on-chip components via the G-Bus and the M-Bus. Several local buses connect the modules contained within the audio block.

DSP

The DSP block is built around a proprietary 32-bit DSP core. It has the following features:

- 200MHz cycle time (system clock). All the instructions execute in one cycle, except for the branching instructions (bra, call, trap, ret etc.), which execute in 1 or 2 cycles and load store instructions with wait states.
- Separate program and data spaces (respectively accessed through the system bus and the D-Bus). The system bus is used for all the instruction fetches and the D-Bus is used through the load/store instructions. Data space: 24Kword, 48KB; Program space: 6Kword, 12KB

- The initial program or data loads before the processor is started. Once the processor is running, it fetches additional program/data through a DMA mechanism from external memory to arbitrary parts of program/data space.
- The first half of the data space has 0 wait states (memory access), second half of data space has one (or more) wait states (IO access)
- The stack is located in the data memory
- The two R-Bus interfaces allow an external bus master to access the P-Bus and D-Bus through the RISC
- 16/32-bit instruction set. Frequently used 32-bit instructions have 16-bit equivalent instructions that can be used to minimize the code space. All the 16-bit instructions have 32-bit equivalent instructions. The instruction set consists of three groups:
  - Data processing: ALU and DSP instructions.
  - Memory: Data memory load/stores.
  - Control: Flow control (sbra, bra, call, trap, ret, rti), push/pop, sp and stat operations.

Audio Miscellaneous

The audio miscellaneous block consists of an interrupt controller, reset controller, D-Bus and R-Bus interfaces, a timer and semaphores.

During a DMA transfer from the DRAM to the RISC Program/Data space, bytes received from the DRAM are assembled into dwords. The resulting dwords are written to the R-Bus. For each R-Bus transaction, the register MISC_DR_ADD is incremented by 1 and the register MISC_DR_CNT is decremented by 1 until it reaches 0.

The register MISC_DR_ADD specifies the R-Bus address used during the DMA from external the DRAM to the RISC program/data space. When the bit 14 =1, the data space is selected. When the bit 14=0, the program space is selected.

The register MISC_DR_CNT gives the number of the R-Bus transactions (the DMA transfer size). Writing a non-zero value to this register starts the DMA transfer, assuming that the M-Bus interface has already been programmed.

The registers MISC_DW_MODE, MISC_DW_ADD and MISC_DW_CNT control the DMAs from the RISC program/data spaces to DRAM. Their operation is identical to the read transfers.

The register MISC_TIMER_DIV needs to be programmed with the desired timer period. The resulting timer frequency is given by: F = FSYSCLK / (4 x MISC_TIMER_DIV).

For example, if FSYSCLK = 148.5MHz and MISC_TIMER_DIV = 7425, then the timer will generate an interrupt at 5KHz.

When the register MISC_TIMER_DIV is written to, the same value is also written into the register MISC_TIMER_CNT. The value 0, disables the timer operation.

The register MISC_TIMER_CNT is the timer count down register. When this register reaches 1, it is reloaded with the register MISC_TIMER_DIV and a timer interrupt is generated. Writing to the register MISC_TIMER_CNT does not affect the counter, but clears the timer interrupt. This is typically done when entering the timer interrupt service routine.

Audio Input Interface

The EM8622L has one digital audio input port that may be configured to be either I2S or S/PDIF compatible. Either the I^2S or the S/PDIF input may be used, but not both simultaneously. The I^2S input operates only as a slave; it is not capable of operating as a master.

The audio input interface allows the DSP to receive and process the audio input from the external sources. The audio input may be received as either I^2S (3-wire synchronous) or S/PDIF (1-wire self-clocked) audio data.

The I^2S input port is designed to capture 2.0-channel PCM audio input data. The S/PDIF input port may capture 2.0-channel PCM audio data or any compressed audio data as defined by the S/PDIF specification and supported by the chip. The input audio may be mixed with other audio and the result output onto the audio outputs.

The audio-in interface has two channels organized as one pair: (L0,R0). Once the input modes are set (once at the beginning of each session), the audio input interface will issue an interrupt to the DSP as each new sample is made available. The interrupt service routine reads the two samples and then writes in the L0 register. The L0 register then causes the interrupt line to become inactive until the next sample set is made available.

The audio input block contains 5 registers: Two data registers (one per channel, in which the new audio samples can be read), one configuration register which must be programmed at the beginning of a recording session and two status registers (one for global information, and one for the S/PDIF channel status bits of the current S/PDIF input, if available). The data registers are left aligned (32-bit) and MSB first.

The audio-in module is on when the reset input is 1, and tries to decode the audio sample in the adapted input module (I^2S or S/PDIF) according to the bit INSEL of the configuration register.

At the beginning of a recording session, the register SO_CONF must be programmed to match the expected input. If it is set to select S/PDIF then no more configuration is needed

as the S/PDIF automatically detects the sampling rate. If I^2S is selected, then the other I^2S configuration must be set properly.

Whatever is the mode chosen, when a new frame is available, the IRQ output is set, the RISC can then read the new data in the data registers and write to the SO_L0_DATA register to deactivate the IRQ. In both the modes, the register SO_STATUS contains the value of the IRQ bit as bit 0. In the S/PDIF mode, the bit SPDIF_ON, indicates if a good S/PDIF signal is detected. This bit is set to 0 in the I^2S mode.

Serial Output Interface

The audio output interface provides the DSP with the ability to output the decoded samples in either the I^2S format for connection to external DAC(s), or in the S/PDIF format for connection to an external S/PDIF transmitter or optical interface. The audio output has 8 channels organized as 4 pairs: L0/R0, L1/R1, L2/R2, L3/R3. The first 6 channels (3 pairs) drive the 3 I^2S output blocks while the 4^{th} pair drives the S/PDIF output block.

In addition to outputting the same 2-channel PCM audio data as the left/right I^2S digital outputs, the S/PDIF output can transmit compressed Dolby Digital, DTS, WMA9 Pro and MPEG audio data. SCMS (Serial Copy Management System) is supported.

The configuration register bits provide substantial flexibility in defining the audio output characteristics which include:

- Enable/disabling each I^2S and S/PDIF channel
- Defining I^2S data output alignment
- Defining audio bit clock (ACLK) output polarity
- Defining frame clock (LRCLK) output polarity
- Selecting I^2S 16/32-bit mode
- Defining I^2S bit order (MSB/LSB first)
- Selecting audio bit clock source

Once the sampling rate, output modes and the miscellaneous configuration registers are set (once at the beginning of a session), the audio output interface will issue an interrupt to the DSP at the sampling rate (32KHz, 44.1KHz, 48KHz or 96KHz). The interrupt service routine must write 8 samples into 8 dedicated registers (one per channel). This causes the interrupt line to become inactive until the next sample set is required. The audio output block has two groups of registers:

- Data registers: There are 8 data registers, one per channel. Typically, the DSP will write the new decoded samples in these registers at a rate equal to the sampling rate (e.g. 44.1KHz, 48KHz.)
- Configuration registers: These registers are programmed once at the beginning of a decoding session.

The S/PDIF output can be programmed/controlled by the software using the register SO_CH_CTRL. The register SO_AUDIO_CLK_DIV specifies the fractional audio clock divider control bits. The output clock frequency is, FDIV_CLK = FCLK x M / [2 x (M+N)]

For example,

If FCLK = 27MHz, M = 1024 and N = 101 then, FDIV_CLK = 48000 x 256

If FCLK = 27MHz, M = 1568 and N = 307 then, FDIV_CLK = 44100 x 256

If FCLK = 27MHz, M = 2048 and N = 1327 then, FDIV_CLK = 32000 x 256

Note: To reduce the audio clock jitter, clean dividers located in the system block can be used instead of the fractional clock divider.

I²S Output Format

The I²S interface operates in the 32-bit (LRCLK = ACLK/64) or 16-bit mode (LRCLK = ACLK/32). For the 16-bit mode, only bits D23 to D8 are output starting with bit 8 (LSB first) or bit 23 (MSB first).

Table 8-1. **I²S data alignment in the 32-bit mode**

Slot	LSB First Align = 0	LSB First Align=1	LSB First Align=30	MSB First Align=0	MSB First Align=1	MSB First Align=30
0	0	0	D22	D23	0	0
1	0	0	D23	D22	D23	0
2	0	0	0	D21	D22	0
3	0	0	0	D20	D21	0
4	0	0	0	D19	D20	0
5	0	0	0	D18	D19	0
6	0	0	0	D17	D18	0
7	0	D0	0	D16	D17	0

Table 8-1. **I²S data alignment in the 32-bit mode (Continued)**

Slot	LSB First Align = 0	LSB First Align=1	LSB First Align=30	MSB First Align=0	MSB First Align=1	MSB First Align=30
8	D0	D1	0	D15	D16	0
9	D1	D2	0	D14	D15	0
10	D2	D3	0	D13	D14	0
11	D3	D4	0	D12	D13	0
12	D4	D5	0	D11	D12	0
13	D5	D6	0	D10	D11	0
14	D6	D7	0	D9	D10	0
15	D7	D8	0	D8	D9	0
16	D8	D9	0	D7	D8	0
17	D9	D10	0	D6	D7	0
18	D10	D11	0	D5	D6	0
19	D11	D12	0	D4	D5	0
20	D12	D13	0	D3	D4	0
21	D13	D14	0	D2	D3	0
22	D14	D15	0	D1	D2	0
23	D15	D16	0	D0	D1	0
24	D16	D17	0	0	D0	0
25	D17	D18	0	0	0	0
26	D18	D19	0	0	0	0
27	D19	D20	0	0	0	0
28	D20	D21	0	0	0	0
29	D21	D22	0	0	0	0
30	D22	D23	0	0	0	D23
31	D23	0	0	0	0	D22

Note: The frequency of LRCLK is identical in both the 16-bit and the 32-bit modes, and is equal to sclkin/256 (or 384). However, the frequency of the ACLK (bit rate) is equal to sclkin/4 (or 6) in the 32-bit mode and equal to sclkin/8 (or 12) in the 16-bit mode. Also, if align = 4 and MSB firs t= 0, then the I²S data is aligned with the S/PDIF data. In the 16-bit mode align = 1 (should be) for the I²S compatible mode.

S/PDIF (IEC958) Output Format

The S/PDIF output consists of a series of data blocks. Each block consists of 192 frames and each frame consists of two sub-frames corresponding to the left and right channels.

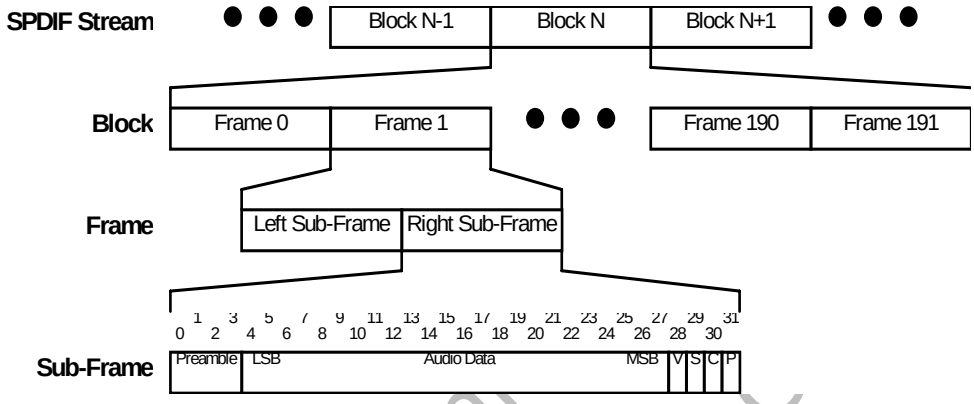


Figure 8-2. **S/PDIF data format**

Data bits and preamble coding

Each subframe consists of 32 bits, coded as a sequence of 64 half-bits. Except for the preamble, data bits are coded as follows:

- 0 is coded ‘11’ (if preceding half-bit = 0) or ‘00’ (if preceding half-bit = 1)
- 1 is coded ‘10’ (if preceding half-bit = 0) or ‘01’ (if preceding half-bit = 1)

There are three types of preambles:

- B (left sub-frame of the first frame of a block)
- M (left sub-frame of any frame except the first frame of a block)
- W (right sub-frame of any frame)

The three preamble types are coded as follows:

- Preamble B is coded ‘11101000’ (if preceding half-bit = 0) or ‘00010111’ (if preceding half-bit = 1)
- Preamble M is coded ‘11100010’ (if preceding half-bit = 0) or ‘00011101’ (if preceding half-bit = 1)
- Preamble W is coded ‘11100100’ (if preceding half-bit = 0) or ‘00011011’ (if preceding half-bit = 1)

Audio Data

The audio data field carries the 24-bit value written in the SO_LS_DATA/SO_RS_DATA registers. The LSB is transmitted first.

- Bit 0 of the register S0_LS_DATA (or the register S0_RS_DATA) appears at the bit position 4 within the S/PDIF sub-frame
- Bit 23 of the register S0_LS_DATA (or the register S0_RS_DATA) appears at the bit position 27 within the S/PDIF sub-frame

When 16-bit or 20-bit PCM samples are transmitted, they are left aligned on bit 23 of the registers SO_LS_DATA / SO_RS_DATA.

Validity Bit (Subframe Bit 28)

This bit (bit 28 of a sub-frame) is normally set to low to indicate valid data. The S/PDIF interface sets this bit when a data underflow occurs.

Subcode Data (Subframe Bit 29)

This bit (bit 29 of a sub-frame) is controlled by bit 19 of the register SO_CH_STRL, (default value= 0).

Channel Status Information (Subframe Bit 30)

This bit is used to carry side information on the S/PDIF stream. The S/PDIF specification requires the channel status bit to be equal for both the subframes of a frame. Consequently, 192 bits of the channel status information are transmitted during a S/PDIF data block. A register contains the channel status information transmitted during the first 32 frames of a block, LSB first. During the remaining 160 (192-32) frames the channel status bit is set to 0.

Parity (Subframe Bit 31)

This bit is generated by the S/PDIF interface so that bits 4 to 31 (inclusive) of the sub-frame contain an even number of 1’s (and 0’s).

Register Maps

Audio Miscellaneous Registers

Table 8-2. **Audio miscellaneous registers**

Address[1]	Register Name	R/W/A[2]	Description
+3E80	MISC_DR_MODE	R/W	DRAM Read Byte Assemble Mode Register
+3E81	MISC_DR_CNT	R/W/A	DRAM Read Byte Count Register
+3E82	MISC_DR_ADD	R/W/A	DRAM Read Address Register
+3E83	Reserved		
+3E84	MISC_DW_MODE	R/W	DRAM Write Byte Assemble Mode Register
+3E85	MISC_DW_CNT	R/W/A	DRAM Write Byte Count Register
+3E86	MISC_DW_ADD	R/W/A	DRAM Write Address Register
+3E87	Reserved		
+3E88	MISC_RESET0	R/W	Reset 0 Register
+3E89	MISC_RESET1	R/W	Reset 1 Register
+3E8A	MISC_INTERRUPT	R/W	Interrupt Register
+3E8B	MISC_TIMER_DI	R/W	Timer Divisor Load Register
+3E8C	MISC_TIMER_CNT	R/A	Timer Value (current) Register
+3E8D	Reserved		
+3E8E	Reserved		
+3E8F	Reserved		

1. Address refers to G-Bus byte address relative to the audio block base.
2. Read/Write/Auto update.

Audio Input Interface Registers

Table 8-3. **Audio input interface registers**

Address[1]	Register Name	R/W/A[2]	Description
+3E40	SI_L0_DATA	R/W	Audio Input Left Channel Data Register
+3E41	SI_R0_DATA	R	Audio Input Right Channel Data Register
+3E42	SI_STATUS	R	Audio Input Status Register
+3E43	SI_CONF	R/W	Audio Input Configuration Register
+3E44	SI_SPDIF_STATUS	R	S/PDIF Status Register

1. Address refers to G-Bus byte address relative to the audio block base.
2. Read/Write/Auto update.

Audio Output Interface Registers

Table 8-4. **Audio output interface registers**

Address	Register Name	R/W	Description
+3E00	SO_L0_DATA	R/W	I^2S Pair 0 Left Channel Data Register
+3E01	SO_R0_DATA	R/W	I^2S Pair 0 Right Channel Data Register
+3E02	SO_L1_DATA	R/W	I^2S Pair 1 Left Channel Data Register
+3E03	SO_R1_DATA	R/W	I^2S Pair 1 Right Channel Data Register
+3E04	SO_L2_DATA	R/W	I^2S Pair 2 Left Channel Data Register
+3E05	SO_R2_DATA	R/W	I^2S Pair 2 Right Channel Data Register
+3E06	SO_LS_DATA	R/W	S/PDIF Left Channel Data Register
+3E07	SO_RS_DATA	R/W	S/PDIF Right Channel Data Register
+3E08	SO_CH_INTR	R/W	Channel Interrupt Register
+3E09	SO_CH_CTRL	R/W	Channel Control Register
+3E0A	SO_SPDIF_CH_STAT	R/W	S/PDIF Channel Status Register
+3E0B	SO_L3_DATA	R/W	I^2S Pair 3 Left Channel Data Register
+3E0C	SO_R3_DATA	R/W	I^2S Pair 3 Right Channel Data Register
+3E0D	Reserved		
+3E0E	SO_AUDIO_CLK_DIV	R/W	Audio fractional clock divider register (M, N)

Pin Description

Audio Input Interface Pins

Table 8-5. **Audio input interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SI_BCLK	AC1	I	I^2S audio input audio bit clock. Maximum frequency is 12.288MHz.
SI_DATA	AB2	I	I^2S audio input audio data
SI_LRCLK	AC2	I	I^2S audio input frame clock

Audio Output Interface Pins

Table 8-6. **Audio output interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SO0_BCLK	Y1	O	I^2S audio bit clock output. Maximum frequency is 12.288MHz.
SO0_MCLK	Y2	O	Audio clock generator master clock output. Maximum frequency is 50MHz.
SO0_LRCLK	AA1	O	I^2S audio frame clock output
SO0_DATA3	Y3	O	I^2S channel 3 audio data output
SO0_DATA2	AA2	O	I^2S channel 2 audio data output
SO0_DATA1	AB1	O	I^2S channel 1 audio data output
SO0_SPDIF	AA3	O	S/PDIF audio data output

Electrical Characteristics

Audio Input Interface DC Characteristics

Table 8-7. **Audio input interface DC electrical characteristics**

Symbol	Description	Units	Min	Typ	Max
V_{IH}	Input high voltage	V	2		5.5
V_{IL}	Input low voltage	V	-0.3		0.8
R_{PU}	Pullup resistor value	kohm	49	63	93

Audio Output Interface DC Characteristics

Table 8-8. **Audio output interface DC electrical characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}(1)	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
I_{OH}[2]	High level output current (@V_{OH} = 2.4V)	mA	25	48	74
I_{OL}(2)	Low level output current (@V_{OL} = 0.4V)	mA	18	28	33

1. Parameter applies to all SO0 pins except SO0_SPDIF.
2. Parameter applies to SO0_SPDIF.

Audio Input Interface AC Characteristics

Table 8-9. **Audio input interface DC timing characteristics**

Symbol	Description	Units	Min	Typ	Max
V_{IH}	Input high voltage	V	2		5.5
V_{IL}	Input low voltage	V	-0.3		0.8
R_{PU}	Pullup resistor value	kohm	49	63	93

Audio Output Interface AC Characteristics

Table 8-10. **Audio output interface DC timing characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	25	40
I_{OL}(1)	Low level output current (@V_{OL} = 0.4V)	mA	9	14	17
I_{OH}[2]	High level output current (@V_{OH} = 2.4V)	mA	25	48	74
I_{OL}(2)	Low level output current (@V_{OL} = 0.4V)	mA	18	28	33

1. Parameter applies to all SO0 pins except SO0_SPDIF.
2. Parameter applies to SO0_SPDIF.

Timing Diagram

I²S Serial-in Timing Diagram

There are four bit rates supported in the I²S audio-in, 16, 20, 24 and 32. In the following diagram x = 15, 19, 23 or 31 for 16, 20, 24 or 32-bit rates respectively.

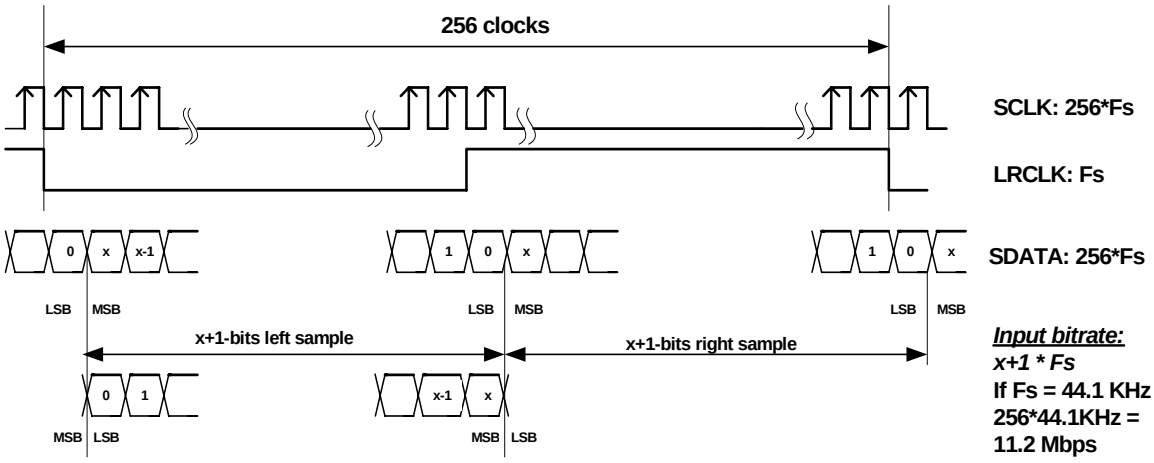


Figure 8-3. **I²S audio-in timing diagram**

9 Transport Demultiplexer

Block Diagram of Transport Demultiplexer

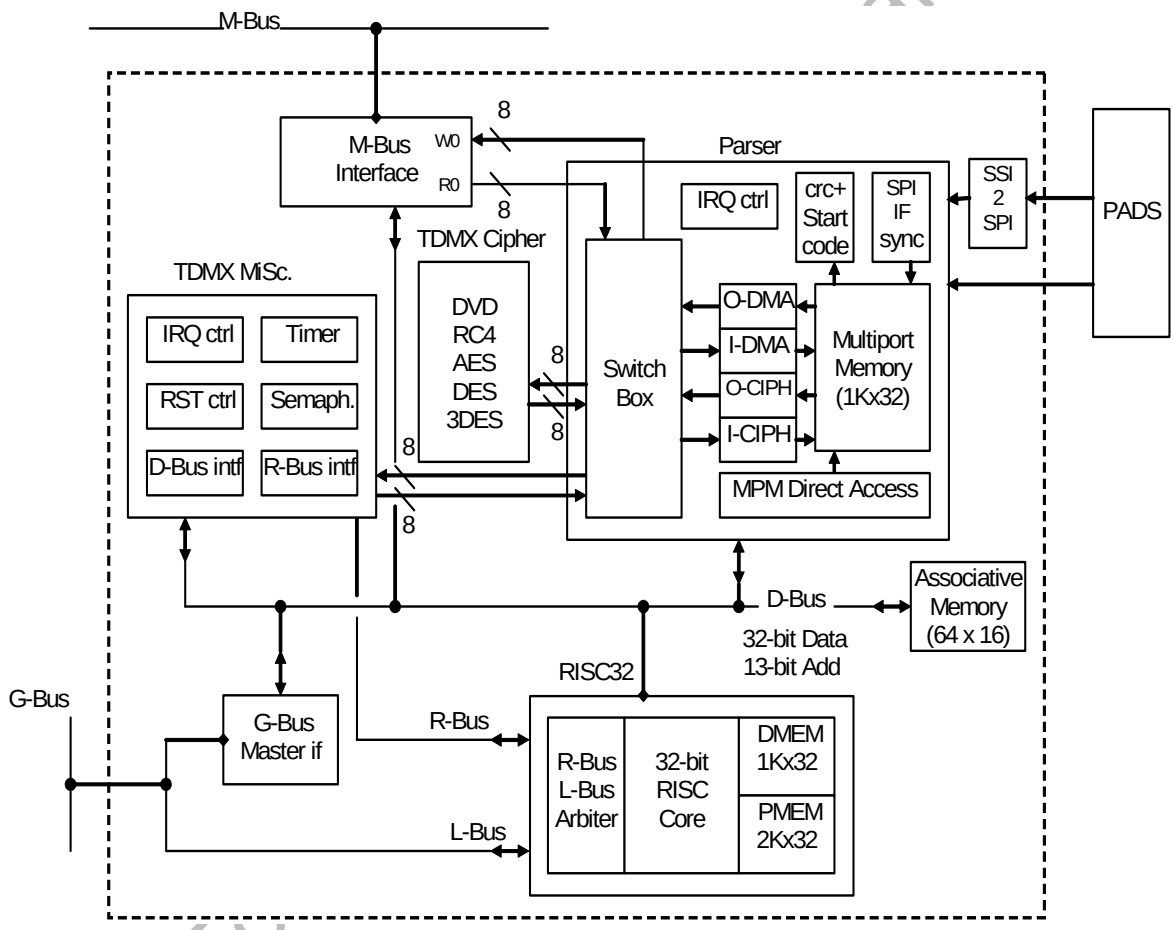


Figure 9-1. **Transport demultiplexer block diagram**

Introduction

The EM8622L includes an on-chip, RISC processor-based transport demultiplexer (TDMX) unit. The transport demultiplexer block is capable of handling up to three multi-program bitstreams of up to 40Mbps each, with an aggregate total of up to 40Mbps. The on-chip demultiplexing supports the following formats:

- DVD-Video, Superbit DVD, DVD-Audio, SVCD and VCD bitstreams
- HD-DVD, BD-ROM and BD-RE bitstreams
- MPEG-1 system bitstreams, MPEG-2 transport and program bitstreams
- MPEG-4 file, MPEG-4.2 over MPEG-2 transport bitstreams
- MPEG-4.10 (H.264) over MPEG-2 transport
- WMV9 ASF or AVI files, VC-1 AP over MPEG-2 transport

The TDMX unit consists primarily of a dedicated 32-bit RISC processor and a parser block along with specialized support hardware. The RISC has direct access to the memory. The TDMX interfaces with the rest of the chip via the G-Bus and M-Bus datapaths. It has additional internal buses to connect the various components.

The primary components of the TDMX unit are the 32-bit RISC processor and the Parser block as shown. The parser block contains a multiport memory (MPM), organized as 1K by 32 bits (4 KB) with 6 independent access ports. The 6 access ports of the MPM are connected to:

- SPI input module
- Input DMA unit
- Output DMA unit
- CRC/start code unit
- Input cipher DMA unit
- Output cipher DMA unit

Additional components of the TDMX unit include:

- A 64 entry associative memory, to quickly retrieve PIDs
- 32 hardware section filters; with 14-byte filtering depth. Each filter can be individually configured to be either a match or a range filter.
- A synchronizer for the SPI data (since the SPI input clock is different from the internal clock)
- A G-Bus to L-Bus bridge to allow any G-Bus master to access the RISC PMEM, DMEM, and all the modules mapped on the D-Bus.
- A G-Bus master interface to allow the RISC to perform G-Bus master accesses
- An M-Bus interface to convert the 8-bit streams going in and out of the parser to the M-Bus protocol
- A Cipher to encrypt or decrypt streams using 8 different algorithms: RC4, DES, Triple DES, AES, DVD-CSS, C2, DVB-CSA and Multi2.
- An SSI to SPI unit which converts a serial stream from the SPI to the SPI unit in the Parser to a parallel stream

The bit stream data may be input via the PCI bus, IDE interface, Local Bus interface etc. The bitstreams are loaded into the DRAM, processed by the transport demultiplexer RISC (including any decryption) and the result is written to the DRAM. The audio and video decoder DSPs then read the data from the DRAM and process it. The bitstreams and the data not used by the EM8622L (such as EPG, etc.) are written to the DRAM and passed on to the middleware for processing.

DVD-Video Decryption

The EM8622L includes hardware CSS decryption and supports DVD-Video CSS procedural specifications. It also fully supports DVD-Video control features such as 16:9 and 4:3 aspect ratios, up to 8 language sound tracks, 32 subtitle settings, letterboxing, pan and scan, multiple angles and 3:2 pulldown.

Features

- Streaming support includes MPEG-2, MPEG-4, ISMA MPEG-4, WM9 and MPEG-4.10 (H.264) and VC-1 over MPEG-2 transport
- Dedicated demultiplexer RISC
  - 32-bit 200MHz RISC CPU core
  - 16KB Data and 16KB Program memory
  - Timers and Interrupt controller support
- Compliant DTV standards
  - ISO/IEC 13818-1:2000
  - ATSC: A-65B
  - DVB: ETS 300 428 v.1.4.1
  - ARIB: STD-B32 Part 3
  - OpenCable®
- Transport stream input interface
  - Any combination of up to 2 transport or program streams, up to 40Mbps total aggregate
  - Hardware/software configurable one 8-bit parallel and two serial input interfaces
  - Programmable clock and data valid polarity
- Conditional access
  - One NRSS-A (ISO-7816 form factor) or smart card interface
  - One NRSS-B (PCMCIA PC card form factor) interface for DVB-CI, ATSC A-70 (requires external SCM CiMAX chip)
- On-chip ciphers
  - * DES with CBC, ECB and OFB mode support
  - Block size: multiple of 64-bit
  - Key sizes: 56-bit
  - 8 key pair (even/odd) key table
  - * 3DES with TCBC, TECB and TOFB mode support
  - Block size: multiple of 64-bit
  - Key sizes: 56-bit (x3)
  - 8 key pair (even/odd) key table
  - * AES with CBC, CFB, CTR, ECB and OFB mode support
  - Block sizes: multiple of 128, 192 or 256-bit
  - Key sizes: 128, 192 or 256-bit
  - 3 key pair (even/odd) key table

    * RC4 (stream cipher)
  - Key sizes: 8 to 256-bit (increments of 8-bit)
  - Can encrypt/decrypt segments up to 65,535 bytes at a time.
  - 4 key pair (even/odd) key table
    * DVB CSA (decryption only)
  - 8 key pair (even/odd) key table
    * ARIB Multi-2 (decryption only)
  - 8 key pair (even/odd) key table
    * C2
  - CSS for DVD-Video
  - CPPM for DVD-Audio
  - CPRM for playback of recordable DVD
- PID filtering
  - Up to 128 programmable PID filters, aggregate
  - Five additional fixed PIDs for filtering of PAT, CAT, PCR, MGT and Null packets per channel
  - Flexible duplicated or erroneous packet control under software configuration
  - Splicing control of up to 4 PIDs for directed channel change applications
- Section filtering
  - Up to 32 hardware based 12-byte match mask and 12-byte range filters per channel
  - Filters can be logically linked in order to construct a longer or more logically completed filter
- ECM filtering
  - Up to 8 ECM filters implemented by the RISC firmware
  - Programmable even/odd ECM filtering
- PSI/SI/PSIP processing
  - 32-bit hardware based CRC checking at the end of the section
  - Version number check for filtering out redundant sections (firmware)
- PES header filtering
  - PTS extraction
  - Private data PES stream type identification
  - DSM-CC support for datacasting

Functional Description

The parser block consists of a multi-port memory (MPM) which contains circular buffers for up to 4 transport streams, and processing blocks which can have simultaneous access to this memory.

The SPI unit receives the data from the PADS, and stores it into the MPM. The input DMA unit reads the data from the DRAM, and stores it into the MPM. The CRC unit calculates the CRC of a data block the MPM, and determine the validity of the block. Also, the CRC block can parse a data block and determine the presence of MPEG2 system start-codes. The Output DMA unit reads the data (typically ES) from the MPM and sends it to DRAM. It also performs the DVD decryption. A DVB/DSS decryption reads a block of data from the MPM, decrypt it and store it back into the MPM.

The RISC has full access to the MPM, and performs all parsing functions. The Input Cipher DMA reads the data from the Cipher, and stores it into the MPM. The Output DMA unit reads the data from the MPM and sends it to the Cipher.

The IRQ controller handles Iciph, Ociph, Idma, Odma and Cipher Interrupts. The SSI to SPI unit receives the data from the PADS (serial), converts it into a parallel stream for the SPI unit.

Multi Port Memory (MPM)

The MPM memory has a total capacity of 4KB. It is physically implemented using four 1024x8 single-port RAMs to allow single cycle access of a 32-bit word at any byte position. It is logically accessible as 4 circular buffers of 1KB.

The MPM block has 5 ports, with an arbiter to select on each clock cycle, which port will have access to the RAMs (The RAMs themselves are single port). The port priority is defined as (highest to lowest):

- SPI unit - Write only
- RISC (through address translation block) – Read/Write
- CRC – Read only
- Output Cipher DMA – Read only
- Input Cipher DMA – Write only
- Output DMA – Read only
- Input DMA – Write only

The 32-bit wide, 64 words deep tag memory holds the packet attributes of the packets stored in the MPM memory. This tag information can be accessed through the D-Bus with the assigned address space. The valid address for the last written tag address can be viewed through the status registers in the PID filtering block. Therefore, tracking the register value gives the number of packets newly written to the MPM since the last tag memory access.

The next destination of the packets in the MPM is determined based on either the scrambling control status from the packet header, or the packet type as indicated by the packet attribute. If the packet is a non-PSI packet, then the RISC firmware routes the payload data to the internal deciphering block.

If the section filter enable attribute is set, then the RISC firmware enables the section filtering for the PSI packets. The section filtering can be initiated by writing the section header data to the hardware section filter register, along with the filter enable attribute value that is obtained from the tag memory.

Once the deciphering process is completed (either by external the CAM or the internal descrambler), all the payload data in the MPM will be in a clear-text format. The RISC can determine whether any other packet processing needs to be done, before posting the data to the external DRAM.

The final destination of each packet and additional packet parsing are resolved according to the packet attribute information such as, the routing information and the packet types. The packet type attribute elects the firmware filtering actions (e.g. PES filtering for AV data, section/ECM filtering for system data etc.), while the routing information attributes decide the depth of the stream parsing depending on the playback type.

The following diagram shows two transactions on a MPM port (the first one has 0 wait states, the second one has 2 wait states):

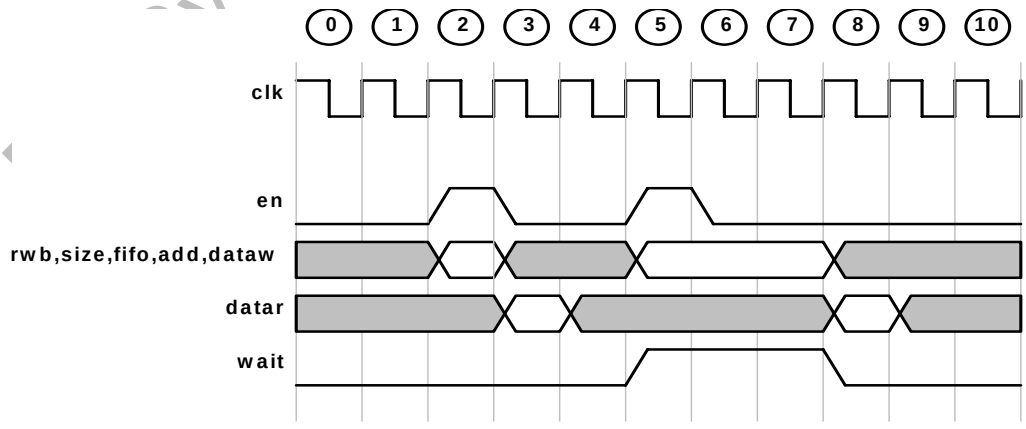


Figure 9-2. **MPM timing diagram**

The timing of a MPM port is very similar to the RISC D-Bus timing. The difference is that on read operations, the data is returned by the MPM exactly one cycle later. As a result, the address translation block (which interfaces between the RISC D-Bus and one of the MPM ports) will have to extend the wait signal given from the MPM port by one cycle prior to sending it to the RISC (only for read operations).

MPM Direct Access Block

The MPM direct access block allows the RISC to directly read/write to the MPM memory. First, a fifo number must be specified in the appropriate register. Then, the RISC can read/write a group of 4 consecutive bytes by accessing one of the locations specified in a register.

The byte with the lowest byte address in the FIFO (the earliest in the bitstream) comes at the most significant position of the 32-bit dword read/written by the RISC.

SPI Unit

The SPI unit receives the incoming byte stream from the SPI external interface. The input stream can consist of up to four transport streams (or SPI channels) multiplexed on a byte-by-byte basis. The channel number is indicated on SPI_CH[1:0] for each byte transferred.

The SPI/SSI input, may be configured as either one 8-bit parallel (SPI) or two serial (SSI) interfaces. AES, 3DES, DES, DVB-CS or MULTI-2 encrypted content may be received over this interface and decrypted.

The SPI unit receives the data from the dedicated SPI pins and stores them in the MPM. The input DMA unit receives the data from the DRAM and stores them in the MPM. The CRC unit reads the data from the MPM and calculates a CRC. It can also search for a start code or for a specific byte. The output DMA block then reads the data from the MPM and sends it to the DRAM, optionally searching for a start code on the fly. The input cipher DMA unit receives the data from the cipher and stores it in the MPM. The output cipher DMA block reads the data from the MPM and sends them to the cipher, optionally searching for a start code on the fly.

The following figure illustrates the SPI input protocol when the input stream contains only one TS bitstream.

The SPI unit has four write pointer registers (corresponding to the four FIFOs implemented in the MPM), which are used to fill the MPM with the incoming SPI data.

This unit can also receive data from 2 independent streams from serial interface. In this case, a bit in the register Serial Stream Input Enable must be set. The serial data inputs are internally converted to an 8-bit byte before the byte is placed in the buffer (MPM). A channel ID of 0 is associated to the first serial input, and a channel ID of 1 is associated to the second. The SPI_CH signals are ignored in serial input mode.

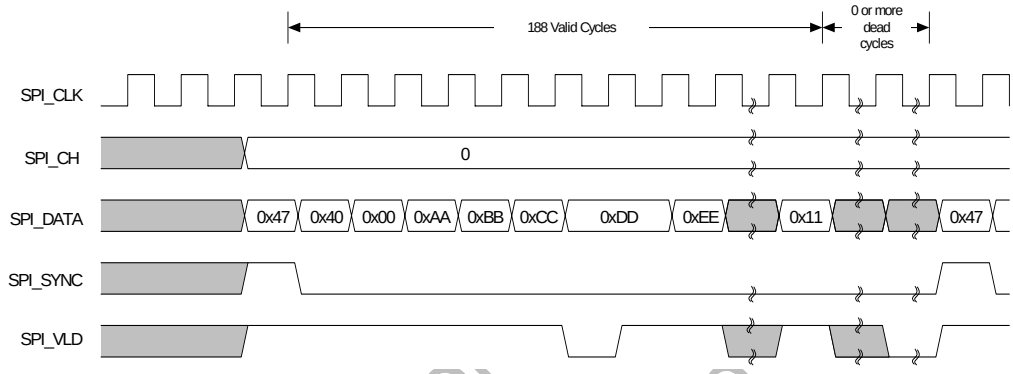


Figure 9-3. **Single channel parallel SPI input**

Regardless of the input mode setting, the input data is accepted only when the corresponding valid signal is asserted at the time the data is on the input bus. The framing of TS packets must be maintained, i.e. the sync byte value of 0x47 must be seen at every 188th valid cycle for each channel, along with sync and valid signal being asserted for the sync byte. Otherwise, there will be a potential glitch on the rendering of the data due to the data loss during transport packet re-synchronization by the demux.

The following figure shows when the SPI parallel input is used for accepting two TS streams multiplexed into one. There is no requirement on the number of dead cycles between two adjacent TS packets as long as a constant bit rate is maintained at the SPI input.

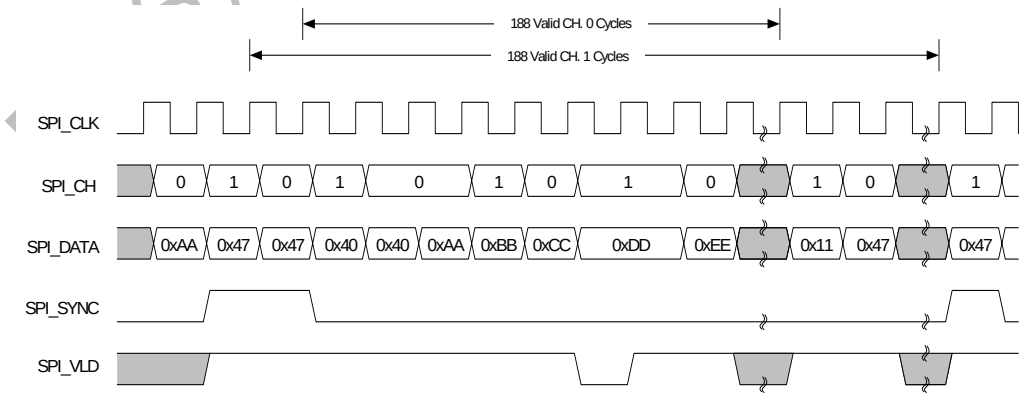


Figure 9-4. **Multi-channel parallel SPI input**

CRC unit

The CRC unit performs three functions.

- Calculates the CRC of a data block stored in the MPM (to verify the integrity of the PSI/SI tables in MPEG-2 transport stream packets).
- Searches MPM data blocks for embedded start codes.
- Searches for 4-byte strings in MPM to locate PES start codes.

When inactive (i.e., ready to receive new commands), the CRC unit drives the INT7 high. If the INT7 is unmasked before a command is sent to the CRC unit, then the interrupt will occur immediately. The CRC unit is controlled using 4 registers.

Input DMA Unit

The input DMA unit receives a byte stream from the switchbox, and writes the data into the MPM. When inactive (i.e., ready to receive new commands), the input DMA unit drives the INT5 high. If the INT5 is unmasked before a command is sent to the input DMA unit, then the interrupt will occur immediately. The input DMA unit is configured using 2 registers.

Output DMA Unit

The output DMA unit can read a data block from the MPM, and send it to the switchbox as a byte stream. When inactive (i.e., ready to receive new commands), the output DMA unit drives the INT2 high. If the INT2 is unmasked before a command is sent to the output DMA unit, the interrupt will occur immediately.

Input Cipher MPM Channel

The input Cipher DMA unit receives a byte stream from the switchbox, and writes the data into the MPM. The data stream received from the switchbox is from the Cipher. The input cipher MPM channel is configured using 2 registers.

Output Cipher MPM Channel

The output Cipher DMA unit can read a data block from the MPM and send it to the switchbox as a byte stream. The data stream sent to the switchbox is routed to (depending on the switchbox configuration): either the M-Bus interface (transfer from the MPM to the DRAM - optional DVD decryption), or the input DMA module (transfer from the MPM to the MPM).

Parser IRQ Controller

The interrupt controller block is the central location for interrupt handling. It accepts the interrupts generated by the internal blocks and generates the parser interrupts. The parser generates 5 interrupts:

1. I-DMA MPM channel
2. O-DMA MPM channel
3. I-Cipher MPM channel
4. O-Cipher MPM channel
5. Cipher block

They are muxed into 2 RISC interrupts:

1. Parser IRQ1 (RISC IRQ2)
   - O-Cipher
   - I-Cipher
   - Cipher
2. Parser IRQ1 (RISC IRQ2)
   - I-DMA
   - O-DMA

A rising edge on an input interrupt loads a corresponding flag. The interrupts are enabled, and the flags read and cleared through a D-Bus register

SSI to SPI

This module handles two serial streams coming from the transport-demux pads interface. The two bit-streams are first converted to parallel streams over 8 bits in their respective clock domain. The resulting byte streams are then transferred to the system clock domain, and are tagged with a channel ID (0: stream 1, 1: stream 2). These are eventually multi-plexed into a single byte stream similar to a regular parallel input stream. The latter feeds the SPI module.

Associative Memory

When performing MPEG-2 transport stream demultiplexing, the RISC processor receives packets of data identified by a 13-bit field called PacketID (PID). For each received packet, a decision needs to be made whether to discard the packet or process the payload.

To speed up the decision process, associative memory is used to store a table of up to 64 active PIDs. A received PID can be referenced quickly in associative memory. If the PID value is stored, the packet is considered active and continues to be processed.

The associative memory is typically updated when Program Association/Map Table (PAT/PMT) packets are received, and queried (via 'reverse reads') when other packets are received. A continuity counter verification is implemented whenever a reverse read is performed for a given PID, by programming a counter along with the PID. The counter is compared with the previously programmed counter + 1 for that PID. The newly programmed counter is then stored.

The associative memory module includes a 64x20 register table. Each memory location is used to store: a 13-bit PID, a 2-bit channel (which support up to 4 streams), a 4-bit counter and an 'empty' status bit hat indicates (high) that the location does not contain a PID. Four active commands mapped on 4 D-Bus locations are used to access or modify the table content.

Cipher

The Ciphering module contains the different way to encrypt/decrypt stream data. It supports the following algorithms:

- DES/Triple DES – (ECB, OFB, CBC)
- AES – (ECB, OFB, CTR, CBC, CFB)
- RC4
- DVD
- DVD-CSA
- Multi-2 – (ECB, OFB, CBC, CFB)
- C2 – (ECB, CBC)

The cipher RAM provides a storage space for multiple key sets and the initialization vectors. The keys for all the ciphers, except for the DVD and C2, are written to the cipher RAM by the CPU in a secure manner. The DVD keys are written directly to the register space. The read access to any key address space is disabled by the hardware. The CPU informs the demultiplexer RISC of the key sets to be used for descrambling the contents by means of key indices, channel ID and the PID.

DES Encryption/Decryption Unit

The DES encryption/decryption unit can either encrypt or decrypt a 64-bit data block using the DES or the Triple DES algorithm. The supported modes are DES ECB, CBC, OFB and Triple DES TECB, TCBC and TOFB, both in encryption and decryption. The configuration is unique for a group of 64-bit data blocks. The number of blocks in a data group is 1 to 255. The IV register is used in the OFB and CBC chaining modes (ignored in the ECB mode). It must be programmed before the encryption/decryption is started, and gets automatically updated after each block is processed. The IV is updated without user intervention in the following modes: OFB, TOFB, CBC, TCBC both encryption and decryption. These modes are chained modes in which the output depends on the outputs of the previous blocks. The IV value at the end of a group is the value which would be required for an additional block in the group. It means that Encrypting 100 data blocks in CBC mode and then again 100 data blocks in CBC mode without writing any value in the IV between the two executions will be exactly the same as encrypting in one shot the 200 data blocks. This is to over run the limitation of 255 blocks in a group. There is thus no limitation in actual groups in either modes. The number of key pairs is 8 (even/odd).

Key format

The Triple DES/DES algorithm uses a 56-bit key. However, 8-bit are added to the key (in key[63:0] bits 0, 8…56) to get a 64-bit key (these bits are the parity control bits). These parity bits are not used during encryption or decryption. However, a 64 bit key must be written in the registers. The value of bits 0, 8, 16, 24, 32, 40, 48, 56 of key[63:0] can be anything. Thus, the encryption using the key = 0000000000000000 and the key = 0101010101010101 will exactly be the same.

Data Format

The data interface (input and output) is a standard valid/ready interface. Data block[63:0] = 64'h0011223344556677. The bytes will be sent in the order: 00 11 22 33 44 55 66 77. Thus the left most byte of the data block is sent for. The left most byte of the output block will be out first. If the encrypted/decrypted bytes received are in the order: 00, 11, 22, 33, 44, 55, 66 and 77, then output enciphered, deciphered block is 64'h0011223344556677.

Table 9-1. **Performance**

	DES - 64-bit to 8-byte	Triple DES - 64- bit to 8- byte
One Block	50 Cycles	86 Cycles
Several blocks	34 Cycles	49 Cycles

The total number of clock cycles represents the time between the start of the ciphering and the rising edge of the DES_END. If the number of the blocks is higher than one, then the blocks are received and sent while the other blocks are encrypted.

AES Encryption/Decryption Unit

The AES encryption/decryption unit can either encrypt or decrypt a 128, 194 or 256-bit data block using the AES algorithm. The supported modes are ECB, OFB, CBC, CFB and CTR, both in encryption and decryption. The supported key lengths are 128, 194 and 256-bit independent of the data block size. The configuration is unique for a group of data blocks. The number of blocks in a data group is up to 255. The number of key pairs is 3 (even/odd).

The key registers (0x1E60 - 0x1E67) are write only while the IV registers is read and write capable. Although writing to the key register is permitted, as the AES module is activated, the contents of the key and IV vector registers are over written with the contents in the code word RAM. When read from the key register, the D-Bus will return values of zeroes. Read of IV vector registers is permitted at the end of each block ciphering, to enable the continuation of the block ciphering in the chained modes.

In the ECB mode the IV registers are ignored. In the OFB, CBC and the CFB modes, these registers are used for IV storage. In the CTR mode these registers store the counter value.

The IV or counter is updated by the module in the following modes: OFB, CBC, CFB, and CTR (during both encryption and decryption). When the cipher is in a chaining mode, the output depends on the outputs of the previous blocks. The IV value at the end of a group is the value which would be required for an additional block in the group. It means that encrypting 100 data blocks in OFB mode, and then again 100 data blocks in any chaining modes without writing any value in the IV between the 2 executions will be exactly the same as encrypting in one step the 200 data blocks. This over runs the limitation of 255 blocks in a group. Thus there is no limitation in actual group size in any chaining mode.

In the CTR mode, the IV is composed of 3 components: a nonce (typically first 32-bit vector), an initial vector and a 32-bit counter. The counter is incremented for every data block. At the end of the group, the value taken by the counter is the value which would be needed for an additional block in the previous group.

Data Format

The data interface (input and output) is a standard valid/ready interface. Data block[127:0] = 128'h00112233445566778899aabbccddeeff. The bytes will be sent in the order: 00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff. The left most byte of the data block is sent first. The left most byte of the output block will be out first. If the encrypted/decrypted bytes received are in the order, 00, 11, 22, 33, 44, 55, 66, 77...ff, then the output enciphered, deciphered block is 128'h00112233445566778899aabbccddeeff.

Table 9-2. **Performance**

Size of block (byte)	16	16	16	24	24	24	32	32	32
Size of Key[1]	16	24	32	16	24	32	16	24	32
Total Cycles	95	115	130	150	160	170	225	230	245

1. In clock cycle (200MHz clock)

RC4 Encryption/Decryption Unit

The RC4 encryption/decryption unit can encrypt or decrypt data up to 65535 bytes using the RC4 algorithm. This module handles all the key lengths from 8 to 256-bit with 8-bit increments.

Data Format

If the given key is 48-bit long, 7B311D8415F0 then, key_1[31:0]= 32'h7B311D84, key_2[31:0]= 32'h15F00000 and the Length_key [8:0] = 030.

Table 9-3. **Performance**

	Clock Cycle
First initialization	258 clock cycles
Second initialization	1538 clock cycles
Encrypt\Decrypt first byte	8 clock cycles
Encrypt\Decrypt following byte	6 clock cycles

DVD Decryption

The DVD decryption unit is controlled by 5 registers. The controller register indicates the number of bytes to process, typically 2048 for a DVD sector. If the count is programmed to 4095 (reset value), then the DVD decryption block is transparent. The start indicates how many bytes must be transferred before the decryption begins, typically 128 for a DVD sector. In a 2048-byte encrypted sector, the first 128-byte are clear and the last 1920-byte are encrypted. Before starting the decryption, the 40-bit title key must be programmed in the key registers. The two state registers allow a backup of the decryption block internal state. This is required when a context switch occurs while a DVD sector is only partially processed.

DVB-CSA

Typically the 64-bit control words, which are used to ciphering the MPEG-2 bitstream contents by the authoring head end, (scrambling authorization module and conditional access system) are also encrypted and embedded in the same bitstream by the means of EMM and ECM. The ECM carries the control word, while the EMM carries the entitlement messages in conjunction with the key to deciphering keys for the control words. These scrambled control words are then extracted and deciphered by the authorized decoders. The following figure depicts the key delivery scheme.

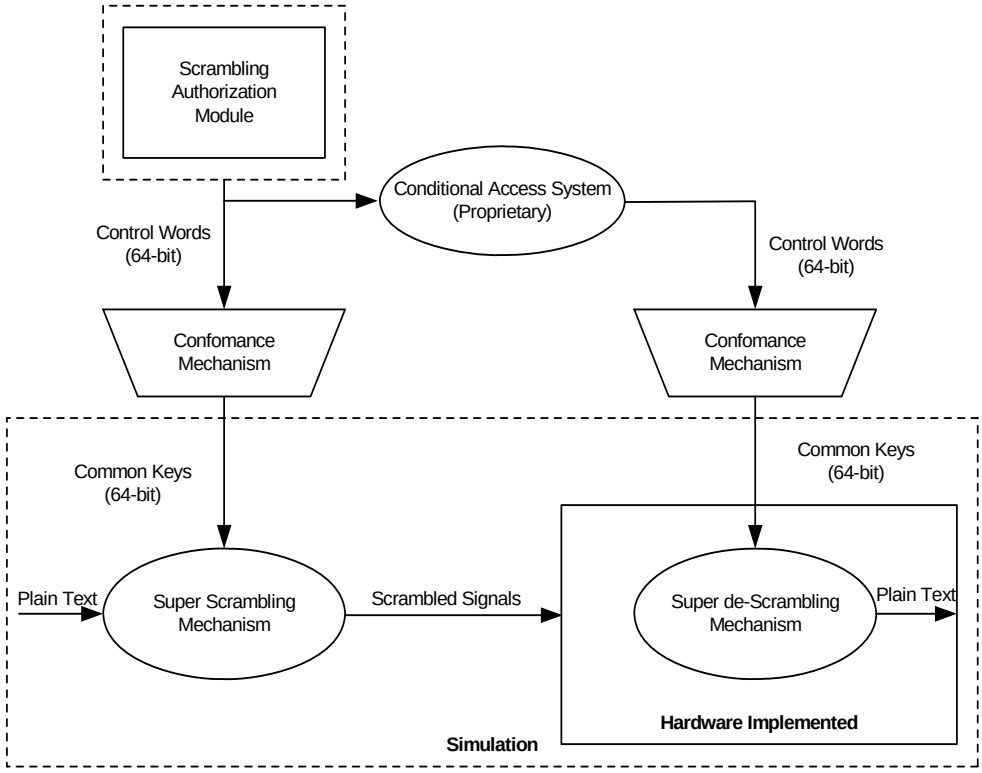


Figure 9-5. **Overview of DVB-CSA**

Once the control words are deciphered, the common keys are obtained by the conformance mechanism. This conformance mechanism is a simple combination of addition and modulus operations that can be easily performed by the secure processor before the keys are put to use. The hardware is designed to work with a set of common keys, which has 64-bit in length derived from the 64-bit control word.

Key Transfer

In general, the control words are produced in a pair, namely even and odd keys by the head-end authoring tools and this pair of keys are used to scramble in an alternative fashion. One set of the keys is used to scramble the program contents for a fixed time frame. The other set of the keys, is used to scramble the content for the next time frame. These are transmitted through the bitstream. This alternate key transmission mechanism provides the system with enough time for the key delivery, computation and the initialization of the hardware.

The set of the key pair used to scramble the contents is indicated at the scrambling control field in the corresponding transport packet header. Hence, the RISC firmware must program the key registers with the appropriate set of the keys, before initiating the deciphering process by the hardware.

Setting up DMA channels

Because the DVB CSA allows the payload being scrambled at either TS-level or PES level, the DMA channels to (ODMA), or out of (IDMA) the decipher hardware must be properly programmed with the payload byte counts based on the scrambling layer information. This, scrambling layer information should come from either the security core or the host CPU (as a result of the table parsing).

When the payload is scrambled at TS-level, the payload length is described with the packet length (188-byte) subtracted by the TS headers (4-byte) and the adaptation field length. However, if the packet is scrambled at the PES-level, the PES header length must be excluded from the payload byte count in addition to the ones for the TS-layer payload length computation. Thereby, the PES header will not be scrambled.

The DVB impose the following rules in order to facilitate the decoder hardware implementation in case the authoring head-end chooses the PES-layer scrambling schemes.

1. The PES header is not scrambled.
2. The TS packets containing parts of a scrambled PES packet do not contain an adaptation field, with the exception of the TS packet containing the end of the PES packet.
3. The first byte of a PES packet header is the first byte of the TS packet payload.

In addition, a couple of rules described below will also be applied irrespective of whether the payload is scrambled at the TS-layer or the PES-layer.

1. The scrambled PES header does not span over multiple TS packets.
2. The TS packet carrying the start of a scrambled PES packet is filled by the PES header and the first part of the PES packet payload.
3. The end of the PES packet payload is aligned with the end of the TS packet by inserting an adaptation field of suitable size.

Multi-2

The Multi-2 module provides the encryption and the decryption of the Multi-2 algorithm in all the chaining modes. The module can do an encryption or a decryption in ECB, CBC, CFB and OFB mode with an iteration number from 1 to 255, for data size 1-byte to 255-byte.

Though the size of the base data block is 64-bit, it is not necessary for the data size to be a multiple of 64-bit for encryption. Moreover, a packet is not always likely to be a multiple of 8-byte. The base unit is a byte, therefore the module interface for data path will be based on 1-byte. The module loads data by 8-byte and processes it by 8 bytes block. If less than 8-byte remain, then this residual bytes are processed in an OFB mode.

The Multi-2 configuration is done through the D-Bus. The parameters are written into the cipher dedicated RAM. Writing into the configuration register (size of data to process, iteration number for the basic Multi-2 algorithm and the chaining mode) starts the module. The module begins to read the system key, the data key and the initial value from the RAM. It then computes the working key, reads the data, processes the data and writes the result back to the RAm. Except for the system keys, the current data, the initial value and the status bit can be read back through the D-Bus. As the last byte of the input data is processed, the module puts one of its outputs (end) high.

Performance

The performance measures the number of cycles to compute an encryption or a decryption once the module has all the parameters needed to do the computation. Therefore the time needed by the module to load the keys and the initial value from the cipher RAM (minimum 24 cycles, 12 values to load, 2 cycles per value) is not taken in the total. Moreover, the following performance asserts no delay in receiving the data from the input and no delay in sending the data to the output.

C2 Block Cipher

The C2 Block deciphers the contents that are generated by the Content Protection for Pre-recorded Media (CPPM) licensed authoring devices. This block supports 5 different ciphering modes namely, C2-D, C2-G, C2E, C2-ECBC and C2-DCBC. All the 5 modes are required during the deciphering process. The mode C2-ECBC is used when content is Content Protection for Recordable Media (CPRM) protected and the EM8622L is used as the encoder.

The functionality of this block is largely partitioned into 3 independent logical blocks: C2 register, C2 control and C2 stream blocks.

The C2 register block is responsible for the D-Bus interface to configure the necessary parameters including the Secret Constant RAM and other data blocks to be processed. Once the D-Bus programming is completed, the C2 control block uses the configuration parameters to generate the data flow control timings based on its state machine.

The stream key generation and data ciphering are performed by the C2 stream block. Depending on the mode of operation, the ciphered data will be routed to either the output port of the cipher or the D-Bus registers.

When the cipher is in the CBC mode, the ciphering is applied to the data from the input cipher ports, allowing the chaining mode for the successive input data blocks. Unlike the CBC modes in other cipher modules (like AES or Multi-2), the cipher key is updated for each data block. Hence in C2, the CBC chaining mode is called the converted CBC mode as the cipher key is chained as opposed to the ciphered data in the other cipher modules.

Once the C2 cipher block configuration is completed, it starts accepting the input data block from the input ports using the ready/valid protocol. Due to the C2-CBC algorithm, the first block takes 16 clock cycles longer then consecutive data blocks to generate the stream key. Two internal 64-bit buffers are provided at the block IO ports in order to eliminate the delays caused by the 64-bit block conversion from eight 8-bit bytes.

Performance

For a data block of 240,or 1920 bytes, the C2 ciphering in the CBC mode will consume,

Num_cycles = 20 + (4 x [240-1]) + (30 x 240) + (2 x 240) + 16 = 8672 clock cycles.

This yields 1.77bits/cycle of total ciphering performance or 1.88bits/cycle without considering the IO transactions.

Unlike the CBC operation, the ECB mode and the one-way function always apply the ciphering process block-by-block. Num_cycles = 2 + 20 + 30 + 1 = 53 cycles/block.

Section/ECM Filtering

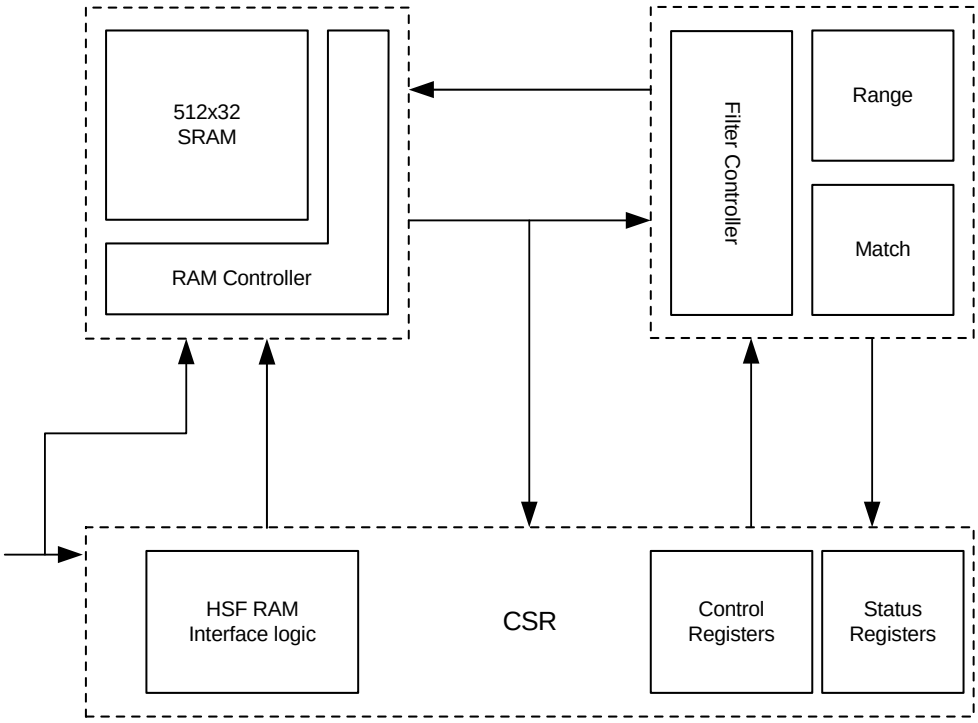


Figure 9-6. **Hardware section filter**

This demultiplexer supports various filtering schemes that applies to the sections from PSI/SI/PSIP and private tables. The section filtering schemes are typically applied to the clear text packets after the PID filtering and the de-scrambling operations. The section headers embedded in the payload part of all general PIDs from the PID bank as well as the dedicated, fixed PIDs can be filtered. The packets that need to be filtered are indicated by the saved packet attribute set during the PID filtering. Their packet types are defined as system or private data. The RISC collects this information by reading the TAG memory.

The hardware performs 32 match or range section-filtering for more complicated section header filtering. The RISC performs simple ECM filtering. However, the hardware can be set up to perform ECM filtering as well.

The section filter is a combination of 32 12-byte match/mask filters, or 12-byte range filters. Simultaneous operation of both types of section filters is permitted by indicating the applied filter type in the Link byte of the filter. The filtering vectors are stored in the HSF RAM.

Section Filtering

Match/Mask Section Filtering

Each of the 32 filters can be applied to any of the general PID entries in the PID bank or other dedicated PSI/SI PIDs from each channel. Each filter is made of 12 sets of match, mask and mode bytes. Each bit of the 12-byte section filter is configured to mask or match (either a positive match or a negative match as indicated in the mode vector). The filtering operation consists of a bit-wise comparison between the section header and a compare vector.

A byte is taken from the section header and compared bit-by-bit, against the corresponding byte from the compare vector. This bit-wise comparison is carried out in two independent operations, one for positive and one for negative match operation according to the mode vector setting. A logic state 1 in the mode vector represents a positive operation, and a logic state 0 represents a negative operation.

The comparison is considered a match if all the bits match in the positive mode and at least one of the bits does not match in the negative mode. A mask is applied bit-wise to the results from the comparison. The result vector is AND operated with the mask vector in order to sort out only the match result of interest.

It is possible to enable multiple filters for the same section header. In this case, all the applied filtering outputs will be logically ANDed (or ORed) by the RISC firmware. This results in the section to be posted to the memory if all (or any) the selected filters pass.

The HSF RAM holds the vectors for the section filtering. Each section filter consists of a set of 12-byte compare (C vector), mode (N vector), mask (M vector) values and a link. The link is used to chain the filters in a linked list for comparison. Linking increases the filtering depth, but the processing time is longer. To reduce the load of the RISC CPU, the firmware filters 4-byte at a time and to stops the filtering processing when a mismatch is found. It moves on to the next filter, if the filter is not the end-of-link.

Each filter consumes 0x30 bytes in the system address space. The filter starting address can be obtained quickly by multiplying the filter number n with 0x30 (n times 0x30).

Range Filter

The section range filtering compares the values of a byte from a section header, against a range specified for the byte in the section range filter. A section range filter consists of a minimum compare vector (Cmin), a maximum compare vector (Cmax), a minimum mask vector (MinM), a maximum mask vector (MaxM), function vector (FN) and a link value (LNK). Each of the Cmin and Cmax filtering vectors is 12-byte, FN is 12-bit and LNK is 8-bit, and are written by the applications in the HSF RAM area.

The operational range function is defined by the value of the FN byte. Each bit of the FN byte represents the corresponding compare byte range check operation. If a bit in FN is 0, then the operation on the corresponding byte is an ‘inner range’ check, or else it is an ‘outer range’ check.

ECM Filtering

The ECM filter mechanism allows selective transfer of new ECM messages to the external DRAM. It is used to eliminate the repeated ECM information from reaching the host CPU. The RISC supports up to 8 independent ECM filters, if the ECM bit CONSTRUCT follows the private section syntax format, or when the section is PES packetized as defined in ISO/ISE 13818-1.

The filtering will be performed by comparing the table id or the first byte of the PES payload. The STREAM_ID will be a part of the filtering process if the section is PES packetized. The host sets the 8-bit PID attributes, in addition to the 3-bit PID PTI for each ECM PID with proper values, before applying the ECM filters to the packets.

Usually, broadcasters transmit even or odd ECM packets, updated every few seconds in an alternating fashion. At power-up (or after user changes the programs), the host configures the PID attributes to receive both ECM in order to descramble the selected program properly. Once both the ECM’s are obtained, the host configures the attribute settings again in order to receive the odd or even ECM information.

Pin Description

Transport Demultiplexer Pins

The SPI (synchronous parallel interface) supports transport stream input to the EM8622L. The SPI interface conforms to the EN 50083-9 specification, with the addition of support for up to four concurrent transport streams multiplexed on a byte-by-byte basis. The SPI_CH1 and SPI_CH0 bits form a 2-bit channel identifier which associates each transferred byte with one of four separate input streams.

Table 9-4. **SPI transport stream interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SPI_CH0	R22	I	SPI channel ID bit 2
SPI_CH1	R21	I	SPI channel ID bit 1
SPI_CLK	U23	I	SPI port clock input data is transferred on the positive-going edge of this clock.

Table 9-4. **SPI transport stream interface pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
SPI_D0	T22	I	SPI input data bit 0 (LSB). When the port is operated in the SSI mode, this pin is the SSI0_DATA input.
SPI_D1	T21	I	SPI input data bit 1. When the port is operated in the SSI mode, this pin is the SSI0_CLK input.
SPI_D2	U22	I	SPI input data bit 2. When the port is operated in the SSI mode, this pin is the SSI0_SYNC input.
SPI_D3	T20	I	SPI input data bit 3. When the port is operated in the SSI mode, this pin is the SSI0_VLD input.
SPI_D4	V23	I	SPI input data bit 4. When the port is operated in the SSI mode, this pin is the SSI1_DATA input.
SPI_D5	U21	I	SPI input data bit 5. When the port is operated in the SSI mode, this pin is the SSI1_CLK input.
SPI_D6	U20	I	SPI input data bit 6. When the port is operated in the SSI mode, this pin is the SSI1_SYNC input.
SPI_D7	T19	I	SPI input data bit 7 (MSB). When the port is operated in the SSI mode, this pin is the SSI1_VLD input.
SPI_SYNC	R20	I	SPI sync (active high). Identifies the first byte of a packet.
SPI_VLD	T23	I	SPI data valid (active high). Indicates valid transport packet bytes.

Electrical Characteristics

SPI Interface DC Characteristics

Table 9-5. **SPI interface DC characteristics**

Symbol	Description	Units	Min	Typ	Max
V_{IH}	Input high voltage	V	2		5.5
V_{IL}	Input low voltage	V	-0.3		0.8

10 Pin Information

Pin Layout

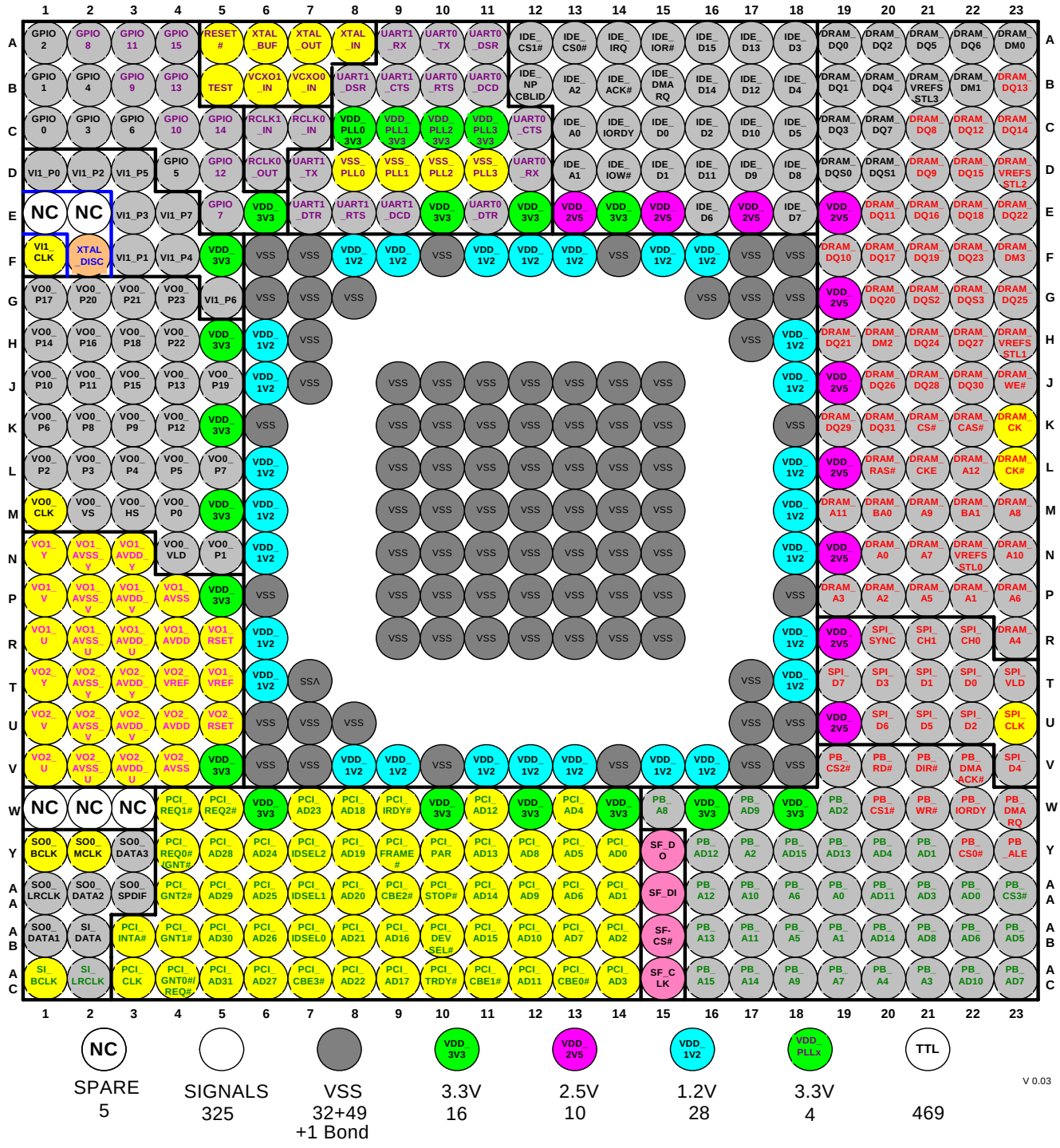


Figure 10-1. **EM8622L pin diagram**

Pin Listing by Ball Id

Table 10-1. **EM8622L pin listing by ball id**

Ball Id	Pin Name
A1	GPIO2
A2	GPIO8
A3	GPIO11
A4	GPIO15
A5	RESET#
A6	XTAL_BUF
A7	XTAL_OUT
A8	XTAL_IN
A9	UART1_RX
A10	UART0_TX
A11	UART0_DSR
A12	IDE_CS1#
A13	IDE_CS0#
A14	IDE_IRQ
A15	IDE_IOR#
A16	IDE_D15
A17	IDE_D13
A18	IDE_D3
A19	DRAM_DQ0
A20	DRAM_DQ2
A21	DRAM_DQ5
A22	DRAM_DQ6
A23	DRAM_DM0
B1	GPIO1
B2	GPIO4
B3	GPIO9
B4	GPIO13
B5	TEST
B6	VCXO1_IN
B7	VCXO0_IN
B8	UART1_DSR
B9	UART1_CTS
B10	UART0_RTS
B11	UART0_DCD
B12	IDE_NPCBLID
B13	IDE_A2
B14	IDE_ACK#
B15	IDE_DMARQ
B16	IDE_D14
B17	IDE_D12
B18	IDE_D4
B19	DRAM_DQ1
B20	DRAM_DQ4
B21	DRAM_VREFSSTL3
B22	DRAM_DM1
B23	DRAM_DQ13
C1	GPIO0
C2	GPIO3
C3	GPIO6
C4	GPIO10
C5	GPIO14
C6	RCLK1_IN
C7	RCLK0_IN
C8	VDD_PLL03V3
C9	VDD_PLL13V3
C10	VDD_PLL23V3
C11	VDD_PLL33V3
C12	UART0_CTS
C13	IDE_A0
C14	IDE_IORDY
C15	IDE_D0
C16	IDE_D2
C17	IDE_D10
C18	IDE_D5
C19	DRAM_DQ3
C20	DRAM_DQ7
C21	DRAM_DQ8
C22	DRAM_DQ12
C23	DRAM_DQ14
D1	VI1_P0
D2	VI1_P2
D3	VI1_P5
D4	GPIO5
D5	GPIO12
D6	RCLK0_OUT
D7	UART1_TX
D8	VSS_PLL0
D9	VSS_PLL1
D10	VSS_PLL2
D11	VSS_PLL3
D12	UART0_RX
D13	IDE_A1
D14	IDE_IOW#
D15	IDE_D1

Ball Id	Pin Name	Ball Id	Pin Name	Ball Id	Pin Name
D16	IDE_D11	F1	VI1_CLK	G16	VSS
D17	IDE_D9	F2	XTAL_DISC	G17	VSS
D18	IDE_D8	F3	VI1_P1	G18	VSS
D19	DRAM_DQS0	F4	VI1_P4	G19	VDD_2V5
D20	DRAM_DQS1	F5	VDD_3V3	G20	DRAM_DQ20
D21	DRAM_DQ9	F6	VSS	G21	DRAM_DQS2
D22	DRAM_DQ15	F7	VSS	G22	DRAM_DQS3
D23	DRAM_VREFSSTL2	F8	VDD_PLL01P2	G23	DRAM_DQ25
E1	NC	F9	VDD_1V2	H1	VO0_P14
E2	NC	F10	VSS	H2	VO0_P16
E3	VI1_P3	F11	VDD_1V2	H3	VO0_P18
E4	VI1_P7	F12	VDD_1V2	H4	VO0_P22
E5	GPIO7	F13	VDD_1V2	H5	VDD_3V3
E6	VDD_3V3	F14	VSS	H6	VDD_1V2
E7	UART1_DTR	F15	VDD_1V2	H7	VSS
E8	UART1_RTS	F16	VDD_1V2	H17	VSS
E9	UART1_DCD	F17	VSS	H18	VDD_1V2
E10	VDD_3V3	F18	VSS	H19	DRAM_DQ21
E11	UART0_DTR	F19	DRAM_DQ10	H20	DRAM_DM2
E12	VDD_3V3	F20	DRAM_DQ17	H21	DRAM_DQ24
E13	VDD_2V5	F21	DRAM_DQ19	H22	DRAM_DQ27
E14	VDD_3V3	F22	DRAM_DQ23	H23	DRAM_VREFSSTL1
E15	VDD_2V5	F23	DRAM_DM3	J1	VO0_P10
E16	IDE_D6	G1	VO0_P17	J2	VO0_P11
E17	VDD_2V5	G2	VO0_P20	J3	VO0_P15
E18	IDE_D7	G3	VO0_P21	J4	VO0_P13
E19	VDD_2V5	G4	VO0_P23	J5	VO0_P19
E20	DRAM_DQ11	G5	VI1_P6	J6	VDD_1V2
E21	DRAM_DQ16	G6	VSS	J7	VSS
E22	DRAM_DQ18	G7	VSS	J9	VSS
E23	DRAM_DQ22	G8	VSS	J10	VSS

Ball Id	Pin Name	Ball Id	Pin Name	Ball Id	Pin Name
J11	VSS	L2	VO0_P3	M18	VDD_1V2
J12	VSS	L3	VO0_P4	M19	DRAM_A11
J13	VSS	L4	VO0_P5	M20	DRAM_BA0
J14	VSS	L5	VO0_P7	M21	DRAM_A9
J15	VSS	L6	VDD_1V2	M22	DRAM_BA1
J18	VDD_1V2	L9	VSS	M23	DRAM_A8
J19	VDD_2V5	L10	VSS	N1	VO1_Y
J20	DRAM_DQ26	L11	VSS	N2	VO1_AVSS_Y
J21	DRAM_DQ28	L12	VSS	N3	VO1_AVDD_Y
J22	DRAM_DQ30	L13	VSS	N4	VO0_VLD
J23	DRAM_WE#	L14	VSS	N5	VO0_P1
K1	VO0_P6	L15	VSS	N6	VDD_1V2
K2	VO0_P8	L18	VDD_1V2	N9	VSS
K3	VO0_P9	L19	VDD_2V5	N10	VSS
K4	VO0_P12	L20	DRAM_RAS#	N11	VSS
K5	VDD_3V3	L21	DRAM_CKE	N12	VSS
K6	VSS	L22	DRAM_A12	N13	VSS
K9	VSS	L23	DRAM_CK#	N14	VSS
K10	VSS	M1	VO0_CLK	N15	VSS
K11	VSS	M2	VO0_VS	N18	VDD_1V2
K12	VSS	M3	VO0_HS	N19	VDD_2V5
K13	VSS	M4	VO0_P0	N20	DRAM_A0
K14	VSS	M5	VDD_3V3	N21	DRAM_A7
K15	VSS	M6	VDD_1V2	N22	DRAM_VREFSSTL0
K18	VSS	M9	VSS	N23	DRAM_A10
K19	DRAM_DQ29	M10	VSS	P1	VO1_V
K20	DRAM_DQ31	M11	VSS	P2	VO1_AVSS_V
K21	DRAM_CS#	M12	VSS	P3	VO1_AVDD_V
K22	DRAM_CAS#	M13	VSS	P4	VO1_AVSS
K23	DRAM_CK	M14	VSS	P5	VDD_3V3
L1	VO0_P2	M15	VSS	P6	VSS

Ball Id	Pin Name	Ball Id	Pin Name	Ball Id	Pin Name
P9	VSS	R23	DRAM_A4	V2	VO2_AVSS_U
P10	VSS	T1	VO2_Y	V3	VO2_AVDD_U
P11	VSS	T2	VO2_AVSS_Y	V4	VO2_AVSS
P12	VSS	T3	VO2_AVDD_Y	V5	VDD_3V3
P13	VSS	T4	VO2_VREF	V6	VSS
P14	VSS	T5	VO1_VREF	V7	VSS
P15	VSS	T6	VDD_1V2	V8	VDD_1V2
P18	VSS	T7	VSS	V9	VDD_1V2
P19	DRAM_A3	T17	VSS	V10	VSS
P20	DRAM_A2	T18	VDD_1V2	V11	VDD_1V2
P21	DRAM_A5	T19	SPI_D7	V12	VDD_1V2
P22	DRAM_A1	T20	SPI_D3	V13	VDD_1V2
P23	DRAM_A6	T21	SPI_D1	V14	VSS
R1	VO1_U	T22	SPI_D0	V15	VDD_1V2
R2	VO1_AVSS_U	T23	SPI_VLD	V16	VDD_1V2
R3	VO1_AVDD_U	U1	VO2_V	V17	VSS
R4	VO1_AVDD	U2	VO2_AVSS_V	V18	VSS
R5	VO1_RSET	U3	VO2_AVDD_V	V19	PB_CS2#
R6	VDD_1V2	U4	VO2_AVDD	V20	PB_RD#
R9	VSS	U5	VO2_RSET	V21	PB_DIR#
R10	VSS	U6	VSS	V22	PB_DMAACK#
R11	VSS	U7	VSS	V23	SPI_D4
R12	VSS	U8	VSS	W1	NC
R13	VSS	U17	VSS	W2	NC
R14	VSS	U18	VSS	W3	NC
R15	VSS	U19	VDD_2V5	W4	PCI_REQ1#
R18	VDD_1V2	U20	SPI_D6	W5	PCI_REQ2#
R19	VDD_2V5	U21	SPI_D5	W6	VDD_3V3
R20	SPI_SYNC	U22	SPI_D2	W7	PCI_AD23
R21	SPI_CH1	U23	SPI_CLK	W8	PCI_AD18
R22	SPI_CH0	V1	VO2_U	W9	PCI_IRDY#

Ball Id	Pin Name	Ball Id	Pin Name	Ball Id	Pin Name
W10	VDD_3V3	Y18	PB_AD15	AB3	PCI_INTA#
W11	PCI_AD12	Y19	PB_AD13	AB4	PCI_GNT1#
W12	VDD_3V3	Y20	PB_AD4	AB5	PCI_AD30
W13	PCI_AD4	Y21	PB_AD1	AB6	PCI_AD26
W14	VDD_3V3	Y22	PB_CS0#	AB7	PCI_IDSEL0
W15	PB_A8	Y23	PB_ALE	AB8	PCI_AD21
W16	VDD_3V3	AA1	SO0_LRCLK	AB9	PCI_AD16
W17	PB_AD9	AA2	SO0_DATA2	AB10	PCI_DEVSEL#
W18	VDD_3V3	AA3	SO0_SPDIF	AB11	PCI_AD15
W19	PB_AD2	AA4	PCI_GNT2#	AB12	PCI_AD10
W20	PB_CS1#	AA5	PCI_AD29	AB13	PCI_AD7
W21	PB_WR#	AA6	PCI_AD25	AB14	PCI_AD2
W22	PB_IORDY	AA7	PCI_IDSEL1	AB15	SF_CS#
W23	PB_DMARQ	AA8	PCI_AD20	AB16	PB_A13
Y1	SO0_BCLK	AA9	PCI_CBE2#	AB17	PB_A11
Y2	SO0_MCLK	AA10	PCI_STOP#	AB18	PB_A5
Y3	SO0_DATA3	AA11	PCI_AD14	AB19	PB_A1
Y4	PCI_REQ0#/GNT#	AA12	PCI_AD9	AB20	PB_AD14
Y5	PCI_AD28	AA13	PCI_AD6	AB21	PB_AD8
Y6	PCI_AD24	AA14	PCI_AD1	AB22	PB_AD6
Y7	PCI_IDSEL2	AA15	SF_DI	AB23	PB_AD5
Y8	PCI_AD19	AA16	PB_A12	AC1	SI_BCLK
Y9	PCI_FRAME#	AA17	PB_A10	AC2	SI_LRCLK
Y10	PCI_PAR	AA18	PB_A6	AC3	PCI_CLK
Y11	PCI_AD13	AA19	PB_A0	AC4	PCI_GNT0#/REQ#
Y12	PCI_AD8	AA20	PB_AD11	AC5	PCI_AD31
Y13	PCI_AD5	AA21	PB_AD3	AC6	PCI_AD27
Y14	PCI_AD0	AA22	PB_AD0	AC7	PCI_CBE3#
Y15	SF_DO	AA23	PB_CS3#	AC8	PCI_AD22
Y16	PB_AD12	AB1	SO0_DATA1	AC9	PCI_AD17
Y17	PB_A2	AB2	SI_DATA	AC10	PCI_TRDY#

Ball Id	Pin Name
AC11	PCI_CBE1#
AC12	PCI_AD11
AC13	PCI_CBE0#
AC14	PCI_AD3
AC15	SF_CLK
AC16	PB_A15
AC17	PB_A14
AC18	PB_A9
AC19	PB_A7
AC20	PB_A4
AC21	PB_A3
AC22	PB_AD10
AC23	PB_AD7

Pin Listing by Pin Name

Table 10-2. **EM8622L pin listing by pin name**

Pin Name	Ball Id
DRAM_A0	N20
DRAM_A1	P22
DRAM_A10	N23
DRAM_A11	M19
DRAM_A12	L22
DRAM_A2	P20
DRAM_A3	P19
DRAM_A4	R23
DRAM_A5	P21
DRAM_A6	P23
DRAM_A7	N21
DRAM_A8	M23
DRAM_A9	M21
DRAM_BA0	M20
DRAM_BA1	M22
DRAM_CAS#	K22
DRAM_CK	K23
DRAM_CK#	L23
DRAM_CKE	L21
DRAM_CS#	K21
DRAM_DM0	A23
DRAM_DM1	B22
DRAM_DM2	H20
DRAM_DM3	F23
DRAM_DQ0	A19
DRAM_DQ1	B19
DRAM_DQ10	F19
DRAM_DQ11	E20
DRAM_DQ12	C22
DRAM_DQ13	B23
DRAM_DQ14	C23
DRAM_DQ15	D22
DRAM_DQ16	E21
DRAM_DQ17	F20
DRAM_DQ18	E22
DRAM_DQ19	F21
DRAM_DQ2	A20
DRAM_DQ20	G20
DRAM_DQ21	H19
DRAM_DQ22	E23
DRAM_DQ23	F22
DRAM_DQ24	H21
DRAM_DQ25	G23
DRAM_DQ26	J20
DRAM_DQ27	H22
DRAM_DQ28	J21
DRAM_DQ29	K19
DRAM_DQ3	C19
DRAM_DQ30	J22
DRAM_DQ31	K20
DRAM_DQ4	B20
DRAM_DQ5	A21
DRAM_DQ6	A22
DRAM_DQ7	C20
DRAM_DQ8	C21
DRAM_DQ9	D21
DRAM_DQS0	D19
DRAM_DQS1	D20
DRAM_DQS2	G21
DRAM_DQS3	G22
DRAM_RAS#	L20
DRAM_VREFSSTL0	N22
DRAM_VREFSSTL1	H23
DRAM_VREFSSTL2	D23
DRAM_VREFSSTL3	B21
DRAM_WE#	J23
GPIO0	C1
GPIO1	B1
GPIO10	C4
GPIO11	A3
GPIO12	D5
GPIO13	B4
GPIO14	C5
GPIO15	A4
GPIO2	A1
GPIO3	C2
GPIO4	B2
GPIO5	D4
GPIO6	C3
GPIO7	E5
GPIO8	A2
GPIO9	B3
IDE_A0	C13
IDE_A1	D13

Pin Name	Ball Id	Pin Name	Ball Id	Pin Name	Ball Id
IDE_A2	B13	PB_A0	AA19	PB_AD9	W17
IDE_ACK#	B14	PB_A1	AB19	PB_ALE	Y23
IDE_CS0#	A13	PB_A10	AA17	PB_CS0#	Y22
IDE_CS1#	A12	PB_A11	AB17	PB_CS1#	W20
IDE_D0	C15	PB_A12	AA16	PB_CS2#	V19
IDE_D1	D15	PB_A13	AB16	PB_CS3#	AA23
IDE_D10	C17	PB_A14	AC17	PB_DIR#	V21
IDE_D11	D16	PB_A15	AC16	PB_DMAACK#	V22
IDE_D12	B17	PB_A2	Y17	PB_DMARQ	W23
IDE_D13	A17	PB_A3	AC21	PB_IORDY	W22
IDE_D14	B16	PB_A4	AC20	PB_RD#	V20
IDE_D15	A16	PB_A5	AB18	PB_WR#	W21
IDE_D2	C16	PB_A6	AA18	PCI_AD0	Y14
IDE_D3	A18	PB_A7	AC19	PCI_AD1	AA14
IDE_D4	B18	PB_A8	W15	PCI_AD10	AB12
IDE_D5	C18	PB_A9	AC18	PCI_AD11	AC12
IDE_D6	E16	PB_AD0	AA22	PCI_AD12	W11
IDE_D7	E18	PB_AD1	Y21	PCI_AD13	Y11
IDE_D8	D18	PB_AD10	AC22	PCI_AD14	AA11
IDE_D9	D17	PB_AD11	AA20	PCI_AD15	AB11
IDE_DMARQ	B15	PB_AD12	Y16	PCI_AD16	AB9
IDE_IOR#	A15	PB_AD13	Y19	PCI_AD17	AC9
IDE_IORDY	C14	PB_AD14	AB20	PCI_AD18	W8
IDE_IOW#	D14	PB_AD15	Y18	PCI_AD19	Y8
IDE_IRQ	A14	PB_AD2	W19	PCI_AD2	AB14
IDE_NPCBLID	B12	PB_AD3	AA21	PCI_AD20	AA8
NC	E1	PB_AD4	Y20	PCI_AD21	AB8
NC	E2	PB_AD5	AB23	PCI_AD22	AC8
NC	W1	PB_AD6	AB22	PCI_AD23	W7
NC	W2	PB_AD7	AC23	PCI_AD24	Y6
NC	W3	PB_AD8	AB21	PCI_AD25	AA6

Pin Name	Ball Id	Pin Name	Ball Id	Pin Name	Ball Id
PCI_AD26	AB6	PCI_REQ2#	W5	SPI_D7	T19
PCI_AD27	AC6	PCI_STOP#	AA10	SPI_SYNC	R20
PCI_AD28	Y5	PCI_TRDY#	AC10	SPI_VLD	T23
PCI_AD29	AA5	RCLK0_IN	C7	TEST	B5
PCI_AD3	AC14	RCLK0_OUT	D6	UART0_CTS	C12
PCI_AD30	AB5	RCLK1_IN	C6	UART0_DCD	B11
PCI_AD31	AC5	RESET#	A5	UART0_DSR	A11
PCI_AD4	W13	SF_CLK	AC15	UART0_DTR	E11
PCI_AD5	Y13	SF_CS#	AB15	UART0_RTS	B10
PCI_AD6	AA13	SF_DI	AA15	UART0_RX	D12
PCI_AD7	AB13	SF_DO	Y15	UART0_TX	A10
PCI_AD8	Y12	SI_BCLK	AC1	UART1_CTS	B9
PCI_AD9	AA12	SI_DATA	AB2	UART1_DCD	E9
PCI_CBE0#	AC13	SI_LRCLK	AC2	UART1_DSR	B8
PCI_CBE1#	AC11	SO0_BCLK	Y1	UART1_DTR	E7
PCI_CBE2#	AA9	SO0_DATA1	AB1	UART1_RTS	E8
PCI_CBE3#	AC7	SO0_DATA2	AA2	UART1_RX	A9
PCI_CLK	AC3	SO0_DATA3	Y3	UART1_TX	D7
PCI_DEVSEL#	AB10	SO0_LRCLK	AA1	VCXO0_IN	B7
PCI_FRAME#	Y9	SO0_MCLK	Y2	VCXO1_IN	B6
PCI_GNT0#/REQ#	AC4	SO0_SPDIF	AA3	VDD_1V2	F9
PCI_GNT1#	AB4	SPI_CH0	R22	VDD_1V2	F11
PCI_GNT2#	AA4	SPI_CH1	R21	VDD_1V2	F12
PCI_IDSEL0	AB7	SPI_CLK	U23	VDD_1V2	F13
PCI_IDSEL1	AA7	SPI_D0	T22	VDD_1V2	F15
PCI_IDSEL2	Y7	SPI_D1	T21	VDD_1V2	F16
PCI_INTA#	AB3	SPI_D2	U22	VDD_1V2	H6
PCI_IRDY#	W9	SPI_D3	T20	VDD_1V2	H18
PCI_PAR	Y10	SPI_D4	V23	VDD_1V2	J6
PCI_REQ0#/GNT#	Y4	SPI_D5	U21	VDD_1V2	J18
PCI_REQ1#	W4	SPI_D6	U20	VDD_1V2	L6

Pin Name	Ball Id	Pin Name	Ball Id	Pin Name	Ball Id
VDD_1V2	L18	VDD_3V3	H5	VO0_P12	K4
VDD_1V2	M6	VDD_3V3	K5	VO0_P13	J4
VDD_1V2	M18	VDD_3V3	M5	VO0_P14	H1
VDD_1V2	N6	VDD_3V3	P5	VO0_P15	J3
VDD_1V2	N18	VDD_3V3	V5	VO0_P16	H2
VDD_1V2	R6	VDD_3V3	W6	VO0_P17	G1
VDD_1V2	R18	VDD_3V3	W10	VO0_P18	H3
VDD_1V2	T6	VDD_3V3	W12	VO0_P19	J5
VDD_1V2	T18	VDD_3V3	W14	VO0_P2	L1
VDD_1V2	V8	VDD_3V3	W16	VO0_P20	G2
VDD_1V2	V9	VDD_3V3	W18	VO0_P21	G3
VDD_1V2	V11	VDD_PLL01P2	F8	VO0_P22	H4
VDD_1V2	V12	VDD_PLL03V3	C8	VO0_P23	G4
VDD_1V2	V13	VDD_PLL13V3	C9	VO0_P3	L2
VDD_1V2	V15	VDD_PLL23V3	C10	VO0_P4	L3
VDD_1V2	V16	VDD_PLL33V3	C11	VO0_P5	L4
VDD_2V5	E13	VI1_CLK	F1	VO0_P6	K1
VDD_2V5	E15	VI1_P0	D1	VO0_P7	L5
VDD_2V5	E17	VI1_P1	F3	VO0_P8	K2
VDD_2V5	E19	VI1_P2	D2	VO0_P9	K3
VDD_2V5	G19	VI1_P3	E3	VO0_VLD	N4
VDD_2V5	J19	VI1_P4	F4	VO0_VS	M2
VDD_2V5	L19	VI1_P5	D3	VO1_AVDD	R4
VDD_2V5	N19	VI1_P6	G5	VO1_AVDD_U	R3
VDD_2V5	R19	VI1_P7	E4	VO1_AVDD_V	P3
VDD_2V5	U19	VO0_CLK	M1	VO1_AVDD_Y	N3
VDD_3V3	E6	VO0_HS	M3	VO1_AVSS	P4
VDD_3V3	E10	VO0_P0	M4	VO1_AVSS_U	R2
VDD_3V3	E12	VO0_P1	N5	VO1_AVSS_V	P2
VDD_3V3	E14	VO0_P10	J1	VO1_AVSS_Y	N2
VDD_3V3	F5	VO0_P11	J2	VO1_RSET	R5

Pin Name	Ball Id	Pin Name	Ball Id	Pin Name	Ball Id
VO1_U	R1	VSS	J7	VSS	N9
VO1_V	P1	VSS	J9	VSS	N10
VO1_VREF	T5	VSS	J10	VSS	N11
VO1_Y	N1	VSS	J11	VSS	N12
VO2_AVDD	U4	VSS	J12	VSS	N13
VO2_AVDD_U	V3	VSS	J13	VSS	N14
VO2_AVDD_V	U3	VSS	J14	VSS	N15
VO2_AVDD_Y	T3	VSS	J15	VSS	P6
VO2_AVSS	V4	VSS	K6	VSS	P9
VO2_AVSS_U	V2	VSS	K9	VSS	P10
VO2_AVSS_V	U2	VSS	K10	VSS	P11
VO2_AVSS_Y	T2	VSS	K11	VSS	P12
VO2_RSET	U5	VSS	K12	VSS	P13
VO2_U	V1	VSS	K13	VSS	P14
VO2_V	U1	VSS	K14	VSS	P15
VO2_VREF	T4	VSS	K15	VSS	P18
VO2_Y	T1	VSS	K18	VSS	R9
VSS	F6	VSS	L9	VSS	R10
VSS	F7	VSS	L10	VSS	R11
VSS	F10	VSS	L11	VSS	R12
VSS	F14	VSS	L12	VSS	R13
VSS	F17	VSS	L13	VSS	R14
VSS	F18	VSS	L14	VSS	R15
VSS	G6	VSS	L15	VSS	T7
VSS	G7	VSS	M9	VSS	T17
VSS	G8	VSS	M10	VSS	U6
VSS	G16	VSS	M11	VSS	U7
VSS	G17	VSS	M12	VSS	U8
VSS	G18	VSS	M13	VSS	U17
VSS	H7	VSS	M14	VSS	U18
VSS	H17	VSS	M15	VSS	V6

Pin Name	Ball Id
VSS	V7
VSS	V10
VSS	V14
VSS	V17
VSS	V18
VSS_PLL0	D8
VSS_PLL1	D9
VSS_PLL2	D10
VSS_PLL3	D11
XTAL_BUF	A6
XTAL_DISC	F2
XTAL_IN	A8
XTAL_OUT	A7

Miscellaneous Pins

Miscellaneous Pins

Miscellaneous Pins

Table 10-3. **Miscellaneous pin descriptions**

Pin Name	Ball ID	Direction	Description
RESET#	A5	I	Device reset input. Active low.
TEST	B5	I	Test mode input. Tie to VSS for normal operation.
XTAL_DISC	F2	I	Used for manufacturing purposes only
NC	E1	-	No connect
NC	E2	-	No connect
NC	W1	-	No connect
NC	W2	-	No connect
NC	W3	-	No connect

11 System Specifications

Electrical Specifications

The absolute maximum ratings should not be exceeded as permanent damage to the device may occur. The operation of the device at the absolute maximum ratings is not implied. The recommended operating conditions provide the range of operating parameters within which the device will perform to specifications.

The DC and AC characteristics which apply to the specific interfaces of the device are listed under the section 'Electrical Characteristics' in their respective chapters. This section groups signals by interface, and lists corresponding signal names, ball numbers, and other relevant information.

Absolute Maximum Ratings

Table 11-1. **Absolute maximum ratings**

Parameter	Symbol	Unit	Minimum	Maximum
DC supply voltage	VDD_1V2	V		1.4
DC supply voltage	VDD_2V5	V		3.0
DC supply voltage	VDD_3V3	V		3.7
DC input voltage	V_{Imax}	V	-0.3	5.5
Storage temperature	T_{STG}	^0C	-65	150

Recommended Operating Conditions

Table 11-2. **Recommended operating conditions**

Parameter	Symbol	Unit	Minimum	Typical	Maximum
DC supply voltage	VDD_1V2	V	1.08	1.2	1.32
DC supply voltage	VDD_2V5	V	2.4	2.5	2.6
DC supply voltage	VDD_3V3	V	3.15	3.3	3.45
Operating ambient temperature	T_{OP}	^0C			70

Power

The EM8622L operates from 3.3, 2.5, and 1.2 volt DC power supplies. The 1.2 volt rail powers the digital core of the device. The 2.5 volt rail powers the SSTL_2 DRAM I/O interface. The 3.3 volt rail powers all the other digital I/O and the analog blocks. Subsequent tables list power supply specifications and the ball locations of the power and ground connections.

Power Supply Sequencing Considerations

To minimize the risk of latch-up or transient high current consumption at power-up, all the supply inputs should rise to their nominal voltages simultaneously. This is easily accomplished by deriving all the supply voltages from a single higher voltage supply using the linear regulators. When this is not possible, the following sequencing rules should be used:

1. Vdd_3p3 should rise before Vdd_1p2.
2. Vdd_2p5 should rise before Vdd_1p2.

Even if this sequencing is followed, the Vdd_1p2 volt supply should follow the 2.5V and 3.3V rails by no more than 500µS. One approach to minimizing the latch-up exposure when power supply sequencing cannot be easily controlled or guaranteed, is to use external Schottky diodes. Attaching two external Schottky diodes eliminates the need for the above sequencing rules.

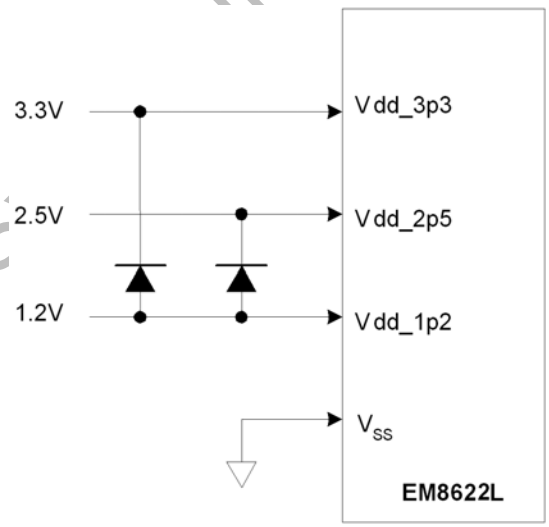


Figure 11-1. **Power supply sequencing with two external Schottky diodes**

Power Consumption

Table 11-3. **Power consumption**

Parameter	Symbol	Unit	Typical
1.2V supply current, normal operation	$I_{dd\_1.2}$	mA	700
3.3V supply current, normal operation, all DAC's disabled	$I_{dd\_2.5}$	mA	33
3.3V supply current, I/O	$I_{dd\_3.3}$	mA	25
3.3V supply current, PLL	$I_{dd\_3.3}$	mA	2
3.3V supply current, component VDACs enabled	$I_{dd\_3.3}$	mA	TBD
3.3V supply current, Y/C/CVBS VDACs enabled	$I_{dd\_3.3}$	mA	TBD

Listing of Ground Pins

The following table lists the EM8622L ground pins:

Table 11-4. **Ground pins (V_{SS})**

Pin Name	Ball ID	Direction	Description
VSS	F10	I	Ground. Zero volt reference for 1.2, 2.5 and 3.3V supplies
VSS	F14	I	Ground
VSS	F17	I	Ground
VSS	F18	I	Ground
VSS	F6	I	Ground
VSS	F7	I	Ground
VSS	G16	I	Ground
VSS	G17	I	Ground
VSS	G18	I	Ground
VSS	G6	I	Ground
VSS	G7	I	Ground
VSS	G8	I	Ground
VSS	H17	I	Ground
VSS	H7	I	Ground
VSS	J10	I	Ground
VSS	J11	I	Ground

Table 11-4. **Ground pins (V_{SS}) (Continued)**

Pin Name	Ball ID	Direction	Description
VSS	J12	I	Ground
VSS	J13	I	Ground
VSS	J14	I	Ground
VSS	J15	I	Ground
VSS	J7	I	Ground
VSS	J9	I	Ground
VSS	K10	I	Ground
VSS	K11	I	Ground
VSS	K12	I	Ground
VSS	K13	I	Ground
VSS	K14	I	Ground
VSS	K15	I	Ground
VSS	K18	I	Ground
VSS	K6	I	Ground
VSS	K9	I	Ground
VSS	L10	I	Ground
VSS	L11	I	Ground
VSS	L12	I	Ground
VSS	L13	I	Ground
VSS	L14	I	Ground
VSS	L15	I	Ground
VSS	L9	I	Ground
VSS	M10	I	Ground
VSS	M11	I	Ground
VSS	M12	I	Ground
VSS	M13	I	Ground
VSS	M14	I	Ground
VSS	M15	I	Ground
VSS	M9	I	Ground
VSS	N10	I	Ground

Table 11-4. **Ground pins (V_{SS}) (Continued)**

Pin Name	Ball ID	Direction	Description
VSS	N11	I	Ground
VSS	N12	I	Ground
VSS	N13	I	Ground
VSS	N14	I	Ground
VSS	N15	I	Ground
VSS	N9	I	Ground
VSS	P10	I	Ground
VSS	P11	I	Ground
VSS	P12	I	Ground
VSS	P13	I	Ground
VSS	P14	I	Ground
VSS	P15	I	Ground
VSS	P18	I	Ground
VSS	P6	I	Ground
VSS	P9	I	Ground
VSS	R10	I	Ground
VSS	R11	I	Ground
VSS	R12	I	Ground
VSS	R13	I	Ground
VSS	R14	I	Ground
VSS	R15	I	Ground
VSS	R9	I	Ground
VSS	T17	I	Ground
VSS	T7	I	Ground
VSS	U17	I	Ground
VSS	U18	I	Ground
VSS	U6	I	Ground
VSS	U7	I	Ground
VSS	U8	I	Ground
VSS	V10	I	Ground

Table 11-4. **Ground pins (V_{SS}) (Continued)**

Pin Name	Ball ID	Direction	Description
VSS	V14	I	Ground
VSS	V17	I	Ground
VSS	V18	I	Ground
VSS	V6	I	Ground
VSS	V7	I	Ground
VSS_PLL3	D11	I	Ground. Dedicated connection for PLL#3
VSS_PLL2	D10	I	Ground. Dedicated connection for PLL#2
VSS_PLL1	D9	I	Ground. Dedicated connection for PLL#1
VSS_PLL0	D8	I	Ground. Dedicated connection for PLL#0

Volt Power Rail

1.2V Power Rail

Table 11-5. **1.2V power rail (VDD_1V2)**

Pin Name	Ball ID	Direction	Description
VDD_1V2	F11	I	1.2V (nominal) power supply
VDD_1V2	F12	I	1.2V (nominal) power supply
VDD_1V2	F13	I	1.2V (nominal) power supply
VDD_1V2	F15	I	1.2V (nominal) power supply
VDD_1V2	F16	I	1.2V (nominal) power supply
VDD_PLL01P2	F8	I	1.2V (nominal) power supply
VDD_1V2	F9	I	1.2V (nominal) power supply
VDD_1V2	H18	I	1.2V (nominal) power supply
VDD_1V2	H6	I	1.2V (nominal) power supply
VDD_1V2	J18	I	1.2V (nominal) power supply
VDD_1V2	J6	I	1.2V (nominal) power supply
VDD_1V2	L18	I	1.2V (nominal) power supply
VDD_1V2	L6	I	1.2V (nominal) power supply
VDD_1V2	M18	I	1.2V (nominal) power supply
VDD_1V2	M6	I	1.2V (nominal) power supply

Table 11-5. **1.2V power rail (VDD_1V2) (Continued)**

Pin Name	Ball ID	Direction	Description
VDD_1V2	N18	I	1.2V (nominal) power supply
VDD_1V2	N6	I	1.2V (nominal) power supply
VDD_1V2	R18	I	1.2V (nominal) power supply
VDD_1V2	R6	I	1.2V (nominal) power supply
VDD_1V2	T18	I	1.2V (nominal) power supply
VDD_1V2	T6	I	1.2V (nominal) power supply
VDD_1V2	V11	I	1.2V (nominal) power supply
VDD_1V2	V12	I	1.2V (nominal) power supply
VDD_1V2	V13	I	1.2V (nominal) power supply
VDD_1V2	V15	I	1.2V (nominal) power supply
VDD_1V2	V16	I	1.2V (nominal) power supply
VDD_1V2	V8	I	1.2V (nominal) power supply
VDD_1V2	V9	I	1.2V (nominal) power supply

2.5V Power Rail

Table 11-6. **2.5V power rail (VDD_2V5)**

Pin Name	Ball ID	Direction	Description
VDD_2V5	E13	I	2.5V (nominal) power supply
VDD_2V5	E15	I	2.5V (nominal) power supply
VDD_2V5	E17	I	2.5V (nominal) power supply
VDD_2V5	E19	I	2.5V (nominal) power supply
VDD_2V5	G19	I	2.5V (nominal) power supply
VDD_2V5	J19	I	2.5V (nominal) power supply
VDD_2V5	L19	I	2.5V (nominal) power supply
VDD_2V5	N19	I	2.5V (nominal) power supply
VDD_2V5	R19	I	2.5V (nominal) power supply
VDD_2V5	U19	I	2.5V (nominal) power supply

3.3V Power Rail

Table 11-7. **3.3V power rail (VDD_3V3)**

Pin Name	Ball ID	Direction	Description
VDD_3V3	E10	I	3.3V (nominal) power supply
VDD_3V3	E12	I	3.3V (nominal) power supply
VDD_3V3	E14	I	3.3V (nominal) power supply
VDD_3V3	E6	I	3.3V (nominal) power supply
VDD_3V3	F5	I	3.3V (nominal) power supply
VDD_3V3	H5	I	3.3V (nominal) power supply
VDD_3V3	K5	I	3.3V (nominal) power supply
VDD_3V3	M5	I	3.3V (nominal) power supply
VDD_3V3	P5	I	3.3V (nominal) power supply
VDD_3V3	V5	I	3.3V (nominal) power supply
VDD_3V3	W10	I	3.3V (nominal) power supply
VDD_3V3	W12	I	3.3V (nominal) power supply
VDD_3V3	W14	I	3.3V (nominal) power supply
VDD_3V3	W16	I	3.3V (nominal) power supply
VDD_3V3	W18	I	3.3V (nominal) power supply
VDD_3V3	W6	I	3.3V (nominal) power supply
VDD_PLL33P3	C11	I	3. V (nominal) power supply. Dedicated power connection for PLL#3
VDD_PLL23P3	C10	I	3.3V (nominal) power supply. Dedicated power connection for PLL#2
VDD_PLL13P3	C9	I	3.3V (nominal) power supply Dedicated power connection for PLL#1
VDD_PLL03P3	C8	I	3.3V (nominal) power supply. Dedicated power connection for PLL#0

Mechanical Specifications

Package Drawing

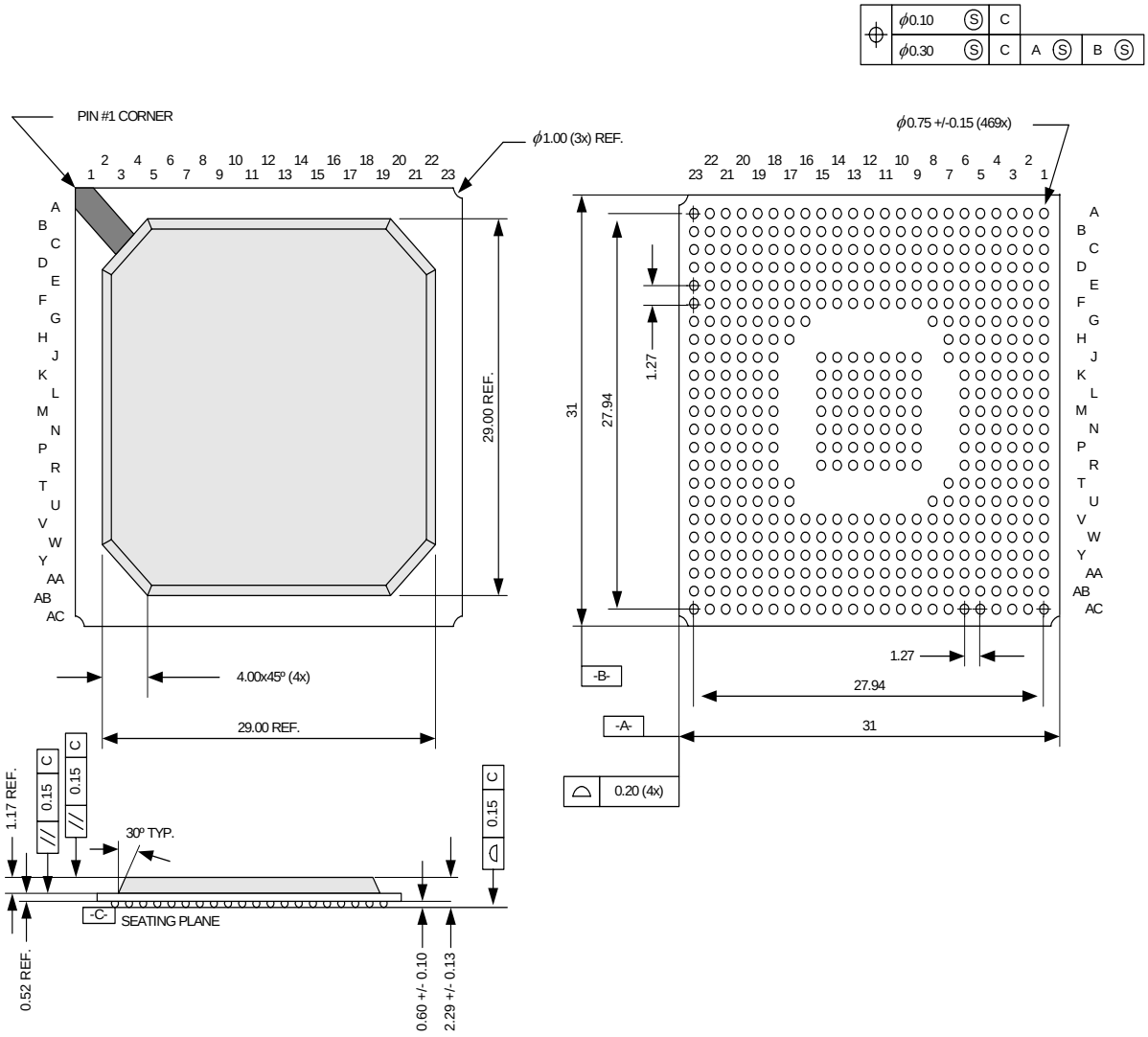


Figure 11-2. **Package diagram (469 BGA)**

Thermal Specifications

Package Thermal Characteristics

Table 11-8. **Package thermal characteristics**

Parameter	Symbol	Unit	Minimum	Typical	Maximum
Thermal resistance, junction to ambient	Theta J-A	^{0}C/W		27	
Thermal resistance, junction to case	Theta J-C	^{0}C/W			
Lead temperature, soldering, 10sec maximum		^{0}C			300

12 Appendix

DDR SDRAM Layout Considerations

Careful PCB (Printed Circuit Board) design is essential to achieve reliable operation of the DDR SDRAM subsystem. This section provides some specific recommendations to follow which help minimize layout-related problems.

Signal Topology

With multiple memory chips, the data and the command signals are routed in a daisy chain topology, while the control and the clock signals should be routed point-to-point.

The DDR signals should not be routed on any layer that is not directly adjacent to a common reference plane. Signals should be routed within the same data group on the same layer only.

To help reduce the crosstalk, the data and the data strobes should be isolated from the address and the command signals. This is accomplished by routing different signal types on separate layers of the PCB.

Signal Length/Width

There are four data signal groups. Within each data signal group, the following signal length matching guidelines should be followed:

DRAMX_D[31:24] and DRAMX_DQM3 must be within +/- 6.5 mm [25 mil] of DRAMX_DQS3

DRAMX_D[23:16] and DRAMX_DQM2 must be within +/- 6.5 mm [25 mil] of DRAMX_DQS2

DRAMX_D[15:8] and DRAMX_DQM1 must be within +/- 6.5 mm [25 mil] of DRAMX_DQS1

DRAMX_D[7:0] and DRAMX_DQM0 must be within +/- 6.5 mm [25 mil] of DRAMX_DQS0

Signals DRAMX_CK and DRAMX_CK# are a differential pair requiring exactly the same length of routing from the controller pad to the memory chip pin. This pair length must be less than 125 mm.

Differential pairs should be routed in a 7 mil trace width, with 8 mil spacing. Other signals should be routed in a 5 mil trace width, with 5 mil spacing. Clearance between the data group signals and the other group signals should be at least 0.5 mm [20 mil] to minimize the crosstalk.

Signal Termination

With careful layout, parallel termination (i.e., a resistor from the signal line to a 1.25 V termination supply voltage) is not required. However, care must be taken to properly terminate the differential CK/CK# pair (discussed below). The series termination resistors of value 15 to 33 ohm are recommended on all data, command and control signals (not required on address group signals). Resistor packs can be used for signals within the data group. The series resistors should be placed close to the memory chip (maximum 12.7 mm/500 mils from the pin). The control signals must NOT be placed within the same resistor packs as data, strobe or command signals.The command signals must NOT be placed within the same resistor packs as data, strobe or control signals.

CK/CK# Termination

Use of a resistor connected in a shunt configuration between the CK and CK# differential clock signals dramatically improves the signal integrity and is recommended. In addition, a capacitor of approximately 10 pF connected in parallel with the shunt resistor provides an additional (although smaller) benefit. When the CK/CK# pair is split into a ‘Y’ to drive two DDR-DRAMs, each branch should have a shunt termination. A resistor value of 470 ohms has been found to provide good results on reference boards designed by Sigma Designs.

1.25 V Vref Generation

A resistor voltage divider (2% precision or better) is adequate to generate the 1.25 V Vref voltage for the SSTL-2 interface from the 2.5 V power supply. A divider using resistors of approximately 1 k-ohm is appropriate. Vref should be decoupled to both Vss and the VDD_2V5 supply at each Vref pin of both the EM8622L and the DDR-DRAMs. Also, the Vref trace should have distributed decoupling along it’s length, using ceramic (MLC) components to reduce the ESL and the ESR.

List of Figures

5 Host Interface

6 Video Decoder Subsystem

7 Video Processing Subsystem

8 Audio Processing Subsystem

9 Transport Demultiplexer

10 Pin Information

11 System Specifications

List of Tables

5 Host Interface

6 Video Decoder Subsystem

7 Video Processing Subsystem

8 Audio Processing Subsystem

9 Transport Demultiplexer

10 Pin Information

11 System Specifications

Sales Offices

USA Headquarters
Sigma Designs, Inc.
1221 California Circle
Milpitas, CA 95035
+1 (408) 262-9003
+1 (408) 957-9740 FAX
www.sigmadesigns.com
sales@sdesigns.com

Europe
Sigma Designs, Inc.
49, Rue des Moissonneurs
Brussels, Belgium 1040
+32 496 501234
+32 234 72260 FAX

Hong Kong
Sigma Designs (Asia) Ltd.
Unit 1516, Tower 1, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
+852 2401 7388
+852 2610 2177 FAX

China
Sigma Designs China
Suite 218, South Block
Changsong Building
Tairan Industry Zone
Shenzhen, PRC 518040
Postcode 518031
+86 755 83435669
+86 755 83435629 FAX

Japan
Sigma Designs Japan
4-16-8 Nakahara,
Mitaka-shi,
Tokyo 181-0005, Japan
+81 422 79 3067
+81 422 79 3067 FAX

Taiwan
Sigma Designs Taiwan
Far East World Center, C Tower
8F-8, No. 79, Sec. 1
Hsin Tai Wu Road
Hsichih, Taipei Hsien, Taiwan
+886 2 2698 2066
+886 2 2698 2099 FAX

Distributors

Japan
Macnica, Inc.
Brilliant Technologies Company
Macnica Bldg. No 1, 1-6-3 Shin-Yokohama
Kohoku-Ku, Yokohama-City, 222-8561 Japan
+ 81 45 470 9831
+ 81 45 470 9832 FAX
www.btc.macnica.co.jp

Korea
Uniquest Korea
8th Floor Dongsung Bldg.
158-24, Samsung-Dong
Kangnam-Ku, Seoul 135-090, Korea
+82 2 3452 6622
+82 2 3452 6623 FAX
www.uniquest.co.kr

License Requirements

Dolby Digital, Dolby Surround Pro Logic II, Dolby Headphone and/or Dolby Virtual Speaker features require license from Dolby. HighMAT, WMA and/or WMT feature require license from Microsoft. MPEG video features require license(s) from MPEG-LA. MPEG audio features require license(s) from Via Licensing, Sisvel and/or Thomson. DTS feature requires license from DTS. DVD-Audio feature requires that the customer be a 4C CPPM, Verance ARIS/SOLANA-4C and Dolby MLP Lossless licensee. Other DVD features require licenses from DVD CCA (for CSS), DVD FLLC (for logo and format) and Philips. Customer must be a Macrovision licensee.Sigma Designs Worldwide

Copyright Information