SMP8634

Secure Media Processor with Multiple A/V Codec Support

Datasheet

November 21, 2006

Document #: SMP8634-DS-A-07

Draft version: November 21, 2006
Adds information about rev. C

Copyright Information

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.

Export Control

Due to the inclusion of cryptographic functions, Sigma Designs has applied to the US Department of Commerce, Bureau of Industry and Security (BIS) for authorization to ship the SMP8634 and SMP8635 to customers around the world. These devices have been authorized by the BIS for license exception ENC under sections 740.17 (A) and (B3). The EECN number under which this exception was granted is 5A002A.1. The following Web site lists all countries where this license exception applies (see *country group B, as in B3 above*): http://www.gpo.gov/bis/ear/pdf/740spir.pdf.

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

Secure Media Processor 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 #: SMP8634-DS-A-07

Table of Contents

Conventions

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

0.1 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)

0.2 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

0.3 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.

Note: 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 or N/A indicate ‘not available’ and NC indicates that a pin is ‘no connect’.

1 Introduction

1.1 Block Diagram of SMP8634

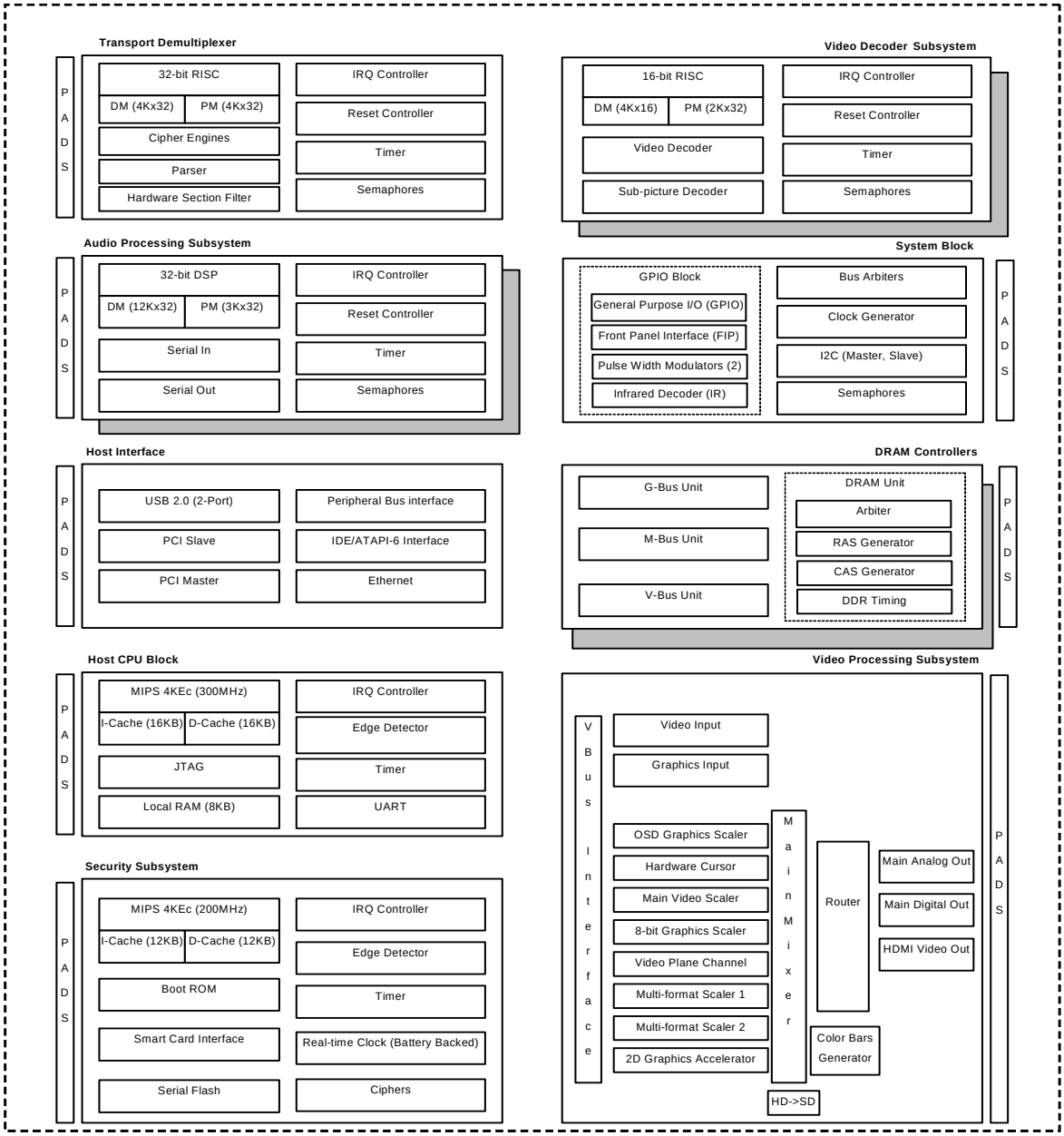


Figure 1-1 **SMP8634 block diagram**

1.2 Main Features of SMP8634

- Host CPU
  - Embedded 300MHz MIPS 4KEc for operating system, middleware and applications
  - 16K program and 16K data caches
  - Interrupt controller and timers
- Highly secure DRM/CA solution
  - Separate 200MHz MIPS 4KEc to securely execute DRM and conditional access software
  - 24K program and 16K data memory
  - Encrypted DRAM and flash memory, secure boot loader, secure PKI, encrypted I/O interfaces, battery backed-up real-time clock
  - Supports a wide variety of DRM and conditional access solutions
- IO standards
  - 32-bit PCI v2.1 (33 or 66MHz) bus master/slave/host
  - Peripheral bus supports IDE (PIO modes), CompactFlash, Local Bus and 8/16-bit NOR Flash
  - Dedicated IDE controller supports ATA/ATAPI-6 modes, UDMA mode 5 up to and including
  - Supports Ethernet 10/100, USB2.0
  - Front panel controller interface supports NEC uPD16311, NEC uPD16312, PTC PT6311 and PTC PT6312 front panel controllers
  - ISO 7816 smart card interface
  - I^2C master/slave interfaces
  - Dual 8-bit parallel SPI transport stream interface, or triple serial transport stream interface
  - I^2S interface up to 20Mbps
  - Two UARTs
  - Local bus interface supports 802.11 chips, MPEG encoders and other external devices
  - 32 general purpose I/O pins
- Video decoding standards
  - Video decoding of MPEG-1, MPEG-2 MP@HL, MPEG-4.2 ASP@L5 (720p), MPEG-4.10 (H.264) BP@L3, MP@L4.0 and HP@L4.0, WMV9 MP@HL, SMPTE 421M (VC-1) MP@HL and AP@L3.
  - 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/SMPTE 421M (SMPTE 421M (VC-1)) SDTV (HDTV): 20 (40) Mbps maximum

• 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/TrueType font rendering acceleration
- 32-bit OSD with flicker filtering and scaling
- Optional deinterlacing of interlaced sources
- Arbitrary scaling of video and graphics up to 1920x1080 pixels
- Alpha mixing of video, graphics, cursor and OSD
- Picture-in-Picture (PIP) support

• Video interfaces

- Flexible 40-bit video/graphics input interface supports multiple video sources, DVI and HDMI receivers, 3D graphics chips
- NTSC/PAL composite and s-video outputs with optional Macrovision protection
- Analog YPbPr/RGB video outputs with optional Macrovision protection in 480i, 576i, 480p and 576p YPbPr output modes (12-bit DACs, interlaced or progressive, SDTV or HDTV resolution)
- Integrated HDMI v1.1 video output interface includes HDCP content protection (requires external HDMI transmitter)
- 150MHz YCbCr/RGB digital video output interface
  * 8-bit and 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

- Audio processing and interfaces
  - Two audio DSPs
  - I^2S and S/PDIF digital audio input interface
  - Each DSP includes five I^2S serial digital outputs to support 7.1-channel plus 2.0-channel downmixed audio simultaneously
  - Digital serial S/PDIF (IEC 60958) audio output
- Transport formats
  - Transport input interfaces
    * Two 8-bit SPI or three SSI, with polarity control of data valid signal
    * PCI, IDE, Local Bus
  - Transport demux supports 128 dedicated PIDs (aggregate)
  - Transport input bit rate: 243Mbps maximum (aggregate)
- Media formats
  - BD-ROM, BD-RE v2, BD-R, HD DVD-Video
  - DVD-Video, DVD-Audio, SVCD (IEC 62107-2000), VCD 1.x and 2.0
  - DVD-R, DVD-RW, DVD+R, DVD+RW (conditional)
  - HDV
  - Audio CD, CD-R, CD-RW, CompactFlash
  - WMA, JPEG, MP3 and MPEG-4.2 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-4.2
  - MPEG-2, MPEG-4
  - WM9 with DRM
  - MPEG-4.10 (H.264) and SMPTE 421M (VC-1) over MPEG-2 Transport
- 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)
- Operating system support
  - Linux
  - WinCE

1.3 Main Components of SMP8634

The SMP8634 is an advanced, single-chip audio/video decoder that provides highly-integrated solutions for HDTV, IPTV, DVD/BD/HD-DVD, MPEG-4.10 (H.264), WMV9 and SMPTE 421M (VC-1) decoding. It incorporates flexible, advanced audio/video processing, enabling cost-effective solutions for consumer products, such as digital media adapters, IPTV set-top boxes, networked DVD players, BD and HD-DVD players/recorders and digital televisions.

The SMP8634 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, DVD/BD/HD-DVD playback, Video-on-Demand (VOD), Personal Video Recording (PVR) and Picture-in-Picture (PIP) into their products.

In addition, the SMP8634 supports numerous popular media formats including DVD-Video, 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 SMP8634 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 MIPS CPU 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 1600MB/sec at a system clock frequency of 200MHz. 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 SMP8634 are briefly described below. Each module is discussed in detail under their respective chapters.

Host CPU Block

The host CPU block of the SMP8634 contains a 300MHz MIPS 4KEc CPU with its instruction and data cache to support the embedded operating system, middle ware and applications required for consumer appliances. The supported operating systems include Linux and WinCE. The software for the CPU is loaded from an external flash using a secure boot loader. Alternately, an external host CPU (such as x86, MIPS or ARM) may be used.

In addition to the RISC 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).

Security Processor Subsystem

A separate 200MHz MIPS 4KEc CPU is used to securely execute conditional access and DRM software. The software for this CPU resides on an internal flash (encrypted and digitally signed) to facilitate upgrades and is loaded using a secure boot loader.

DRAM Controllers

The SMP8634 contains two identical double-data rate synchronous DRAM (DDR-SDRAM) controllers operating at the memory clock rate. Because the DDR technology transfers data on both edges of the clock, the effective burst data bandwidth of each controller is 1.6GB/sec when using the full 32-bit interface (total peak bandwidth of 3.2GB/sec with 2 controllers). The DRAM controllers can interface up to 512MB of external DDR SDRAM, using a 32-bit wide data bus.

System Block

The SMP8634 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 32 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 SMP8634 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 SMP8634 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 300MHz) 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, Ethernet 10/100, USB 2.0, the peripheral bus and the IDE interface.

The SMP8634 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 SMP8634 includes a USB 2.0 Embedded Host Controller and PHY. It also includes an Ethernet 10/100 MAC with an MII (Media Independent Interface) interface to connect to an external PHY device.

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 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 (25 address/16 data) or a multiplexed address/data mode. In the IDE mode, the interface can support ATA/ATAPI-6 device attachments.

Video Decoder Subsystem

The SMP8634 contains two identical video decoders. The video decoder subsystem executes the video decoding algorithms supported by the SMP8634. 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 SMP8634.

Video Processing Subsystem

The SMP8634 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 main analog output. The digital output supports 8, 16 or 24-bit output, RGB or YPbPr data format and the analog output supports RGB, YPbPr, S-video and composite video. The digital and analog component outputs can each support output formats up to 1920x1080p.

The SMP8634 contains a HDMI Transmitter requiring an external PHY. The HDMI Transmitter block in the SMP8634 is HDMI v1.1 compliant, which transports consumer electronics standard digital video and digital audio over a TMDS interface.

Audio Processing Subsystem

The SMP8634 contains an integrated audio subsystem based on a custom-designed 32-bit digital signal processor (DSP). Audio decoding and processing algorithms are implemented on two identical DSPs. This firmware-based approach gives great flexibility for accommodating future audio standards or specialized audio requirements. The audio unit provides 5 I^2S output channels, one S/PDIF output channel, and one I^2S audio input channel.

Transport Demultiplexer

The SMP8634 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 81Mbps each, with an aggregate total of up to 243Mbps.

1.4 Application Examples

1.4.1 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 SMP8634 provides most of the features including:

- Progressive DVD-Video, DVD-Audio, MPEG4.10 (H.264), WMV9/SMPTE 421M (VC-1) playback
- Interlaced or progressive YPbPr or RGB video outputs
- NTSC/PAL composite and S-video outputs
- 7.1-channel and S/PDIF audio outputs
- I^2C bus master function for controlling other chips
- A MIPS CPU for operating system, middleware and applications

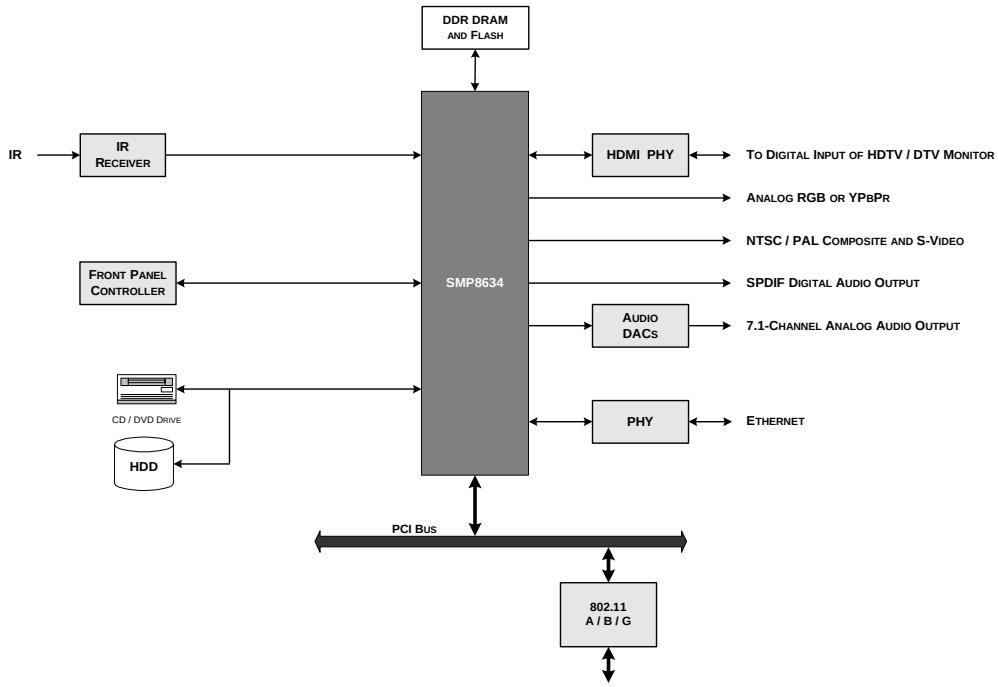


Figure 1-2 **Application example - networked DVD player**

1.4.2 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 SMP8634 provides most of the features including:

- Decoding of MPEG-1, MPEG-2, MPEG-4.10 (H.264) and WMV9/SMPTE 421M (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
- A MIPS CPU for operating system, middleware and applications

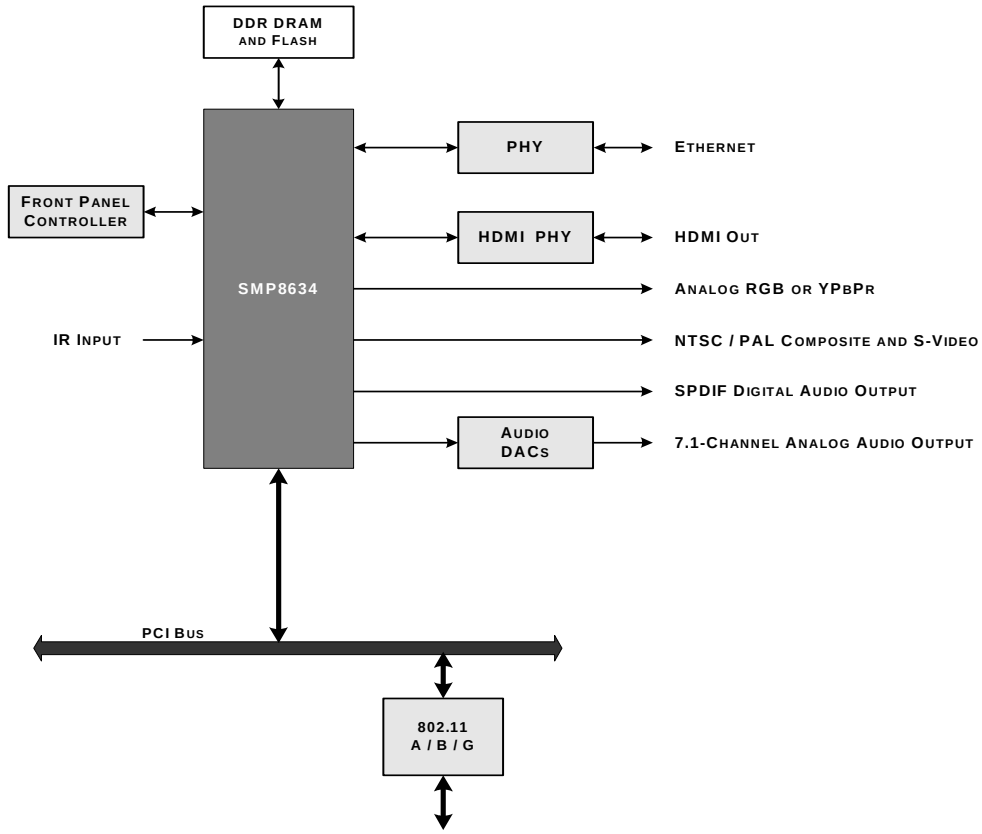


Figure 1-3 **Application example - digital media adapter or IPTV set-top box**

1.4.3 LCD/Plasma HDTV with Digital Cable Receiver

The HDTV receiver with digital cable receiver example below provides the typical functionality required for an digital cable compliant DTV. System integration requires very little external logic since the SMP8634 provides most of the features including:

- Wired and/or wireless networking support
- HDTV and SDTV decoding with 2D graphics, OSD and deinterlacing
- Analog and digital video outputs
- S/PDIF digital audio output for driving a surround sound receiver
- I^2C bus master function for controlling other chips
- A RISC CPU for operating system, middleware and applications
- Easily modified to support ATSC, DVB-T, DVB-S or DVB-C
- DVD playback may be easily added

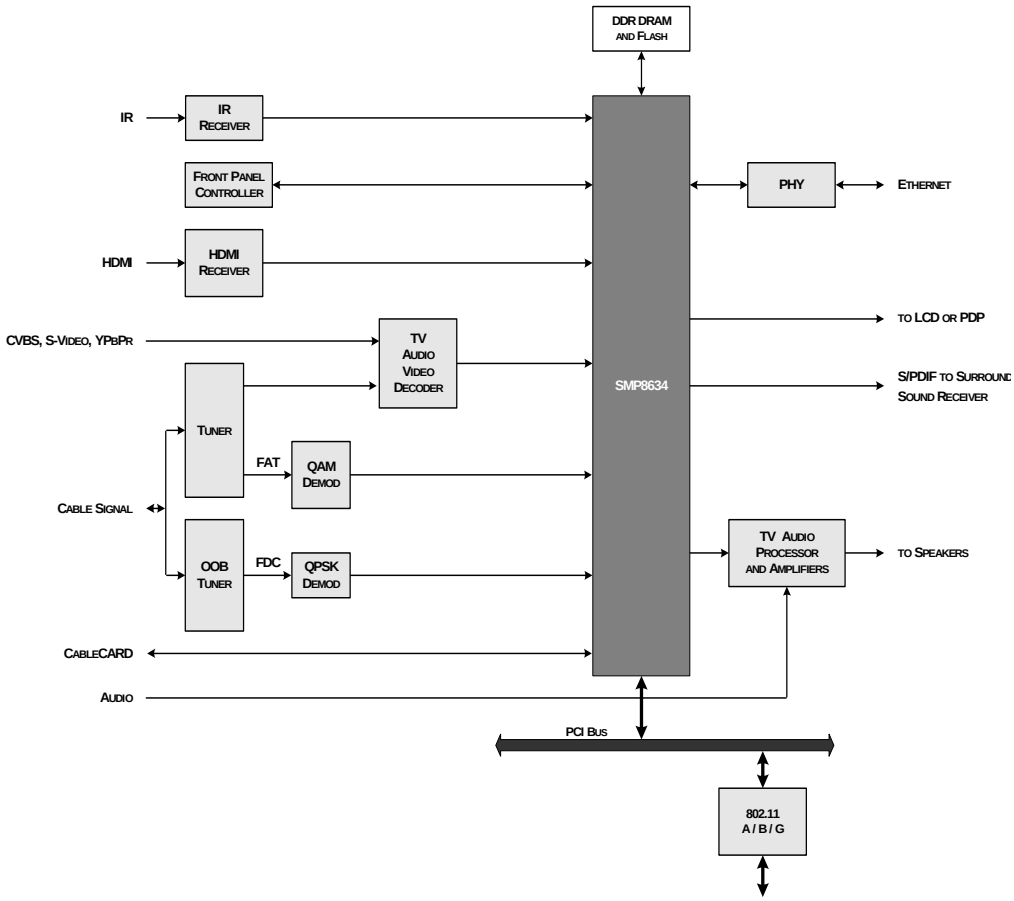


Figure 1-4 **Application example - LCD or plasma HDTV with digital cable receiver**

2 Host CPU Block

2.1 Block Diagram of Host CPU Block

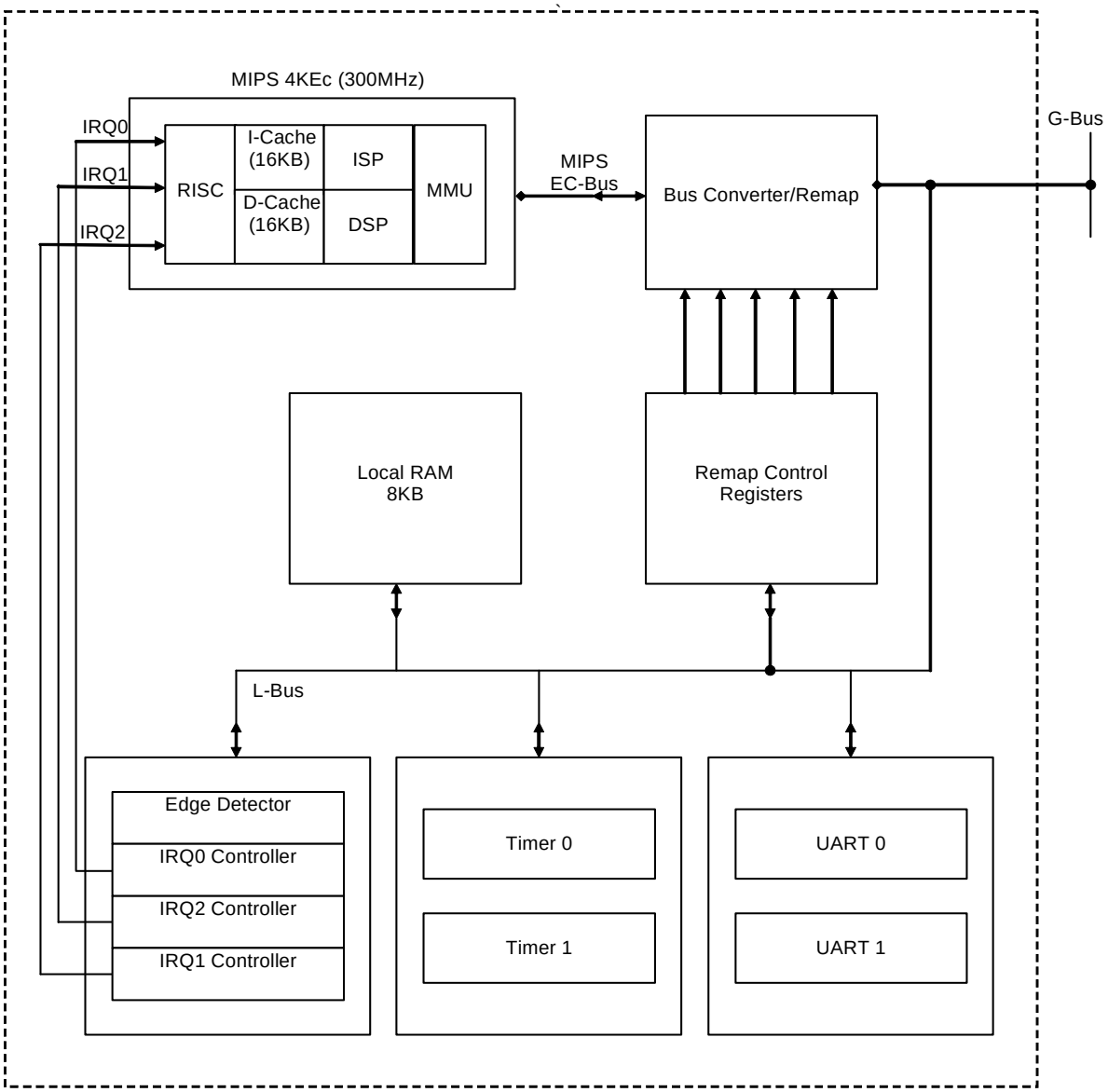


Figure 2-1 **Block diagram of host CPU block**

2.2 Introduction

The host CPU block of the SMP8634 contains a 300MHz MIPS 4KEc CPU with its instruction and data cache to support the embedded operating system, middleware and applications required for the consumer products. Supported operating systems include Linux and WinCE. Software for the CPU is loaded from an external flash using a secure boot loader. Alternately, an external host CPU (such as x86, MIPS or ARM) may be used.

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

- 2Kx32 (8KB) local memory
- Three interrupt controllers
- 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 through the L-Bus. For a MIPS 4KEc, a MIPS EC-Bus 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 the other G-Bus master. Both the bridges in the host CPU block perform a bus protocol adaptation and a configurable address translation (remap). The host CPU block connects to the rest of the chip via the G-Bus.

Table 2-1 **Remap registers**

Address[1]	Register Name	R/W/A[2]	Description
+F000	CPU_REMAP0	R/W	CPU Remap 0 Register
+F004	CPU_REMAP1	R/W	CPU Remap 1 Register
+F008	CPU_REMAP2	R/W	CPU Remap 2 Register
+F00C	CPU_REMAP3	R/W	CPU Remap 3 Register
+F010	CPU_REMAP4	R/W	CPU Remap 4 Register

1. G-Bus byte address relative to the host CPU block base.
2. Read/write/auto update.

Central Processor Unit (CPU)

2.3 Introduction

The SMP8634 uses a 300MHz MIPS 4KEc™ processor (~459 conforming DMIPS at 300MHz). The MIPS32™ 4KEc™ core is a high-performance, low-power, 32-bit MIPS RISC core. It is highly portable across processes. The 4KEc core 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 SMP8634 takes full advantage of all these features.

The 4KEc core is a 32-bit privileged resource architecture. A Memory Management Unit (MMU) contains 4-entry instruction and data Translation Lookaside Buffers (ITLB/ DTLB) and a 32 dual-entry joint TLB (JTLB) with variable page sizes. The 4KEc core includes a Multiply/Divide Unit (MDU) that implements single cycle MAC instructions, which enable the DSP algorithms to be performed efficiently. It allows 32-bit x 16-bit MAC instructions to be issued every cycle, while a 32-bit x 32-bit MAC instruction can be issued every 2 cycles.

It contains 2-way set associative 16KB instruction and 16KB data caches. The load and fetch cache misses only block until the critical word becomes available. The pipeline resumes execution while the remaining words are being written to the cache. Both caches are virtually indexed and physically tagged to allow them to be accessed in the same clock that the address is translated.

An Enhanced JTAG (EJTAG) block allows for single-stepping of the processor as well as instruction and data virtual address/value breakpoints.

2.4 Features

- 5-stage pipeline
- 32-bit address and data paths
- MIPS32-compatible instruction set
  - Multiply-Accumulate and Multiply-Subtract instructions (MADD, MADDU, MSUB, MSUBU)
  - Targeted Multiply instruction (MUL)
  - Zero/One Detect instructions (CLZ, CLO)
  - Wait instruction (WAIT)
  - Conditional Move instructions (MOVZ, MOVN)

  - Prefetch instruction (PREF)

- MIPS32 architecture features
  - Vectored interrupts and support for external interrupt controller
  - Programmable exception vector base
  - Atomic interrupt enable/disable
  - Bit field manipulation instructions
  - Improved virtual memory support (smaller page sizes and hooks for more extensive page table manipulation)
- Separate, 16KB, 2-way set associative instruction and data caches
  - Loads block only until the critical word is available
  - Write-back and write-through support
  - 16-byte cache line size
  - Virtually indexed, physically tagged
  - Cache line locking support
  - Non-blocking prefetches
- MIPS32 privileged resource architecture
  - Count/Compare registers for real-time timer interrupts
  - I and D watch registers for software breakpoints
- Programmable Memory Management Unit (MMU)
  - 32 dual-entry JTLB with variable page size
  - 4-entry ITLB
  - 4-entry DTLB
- Multiply/Divide unit
  - Maximum issue rate of one 32x16 multiply per clock
  - Maximum issue rate of one 32x32 multiply every other clock
  - Early-in iterative divide. Minimum 11 and maximum 34 clock latency (dividend (*rs*) sign extension-dependent)
- EJTAG debug
  - Support for single stepping
  - Virtual instruction and data address/value breakpoints

2.5 Block Diagram

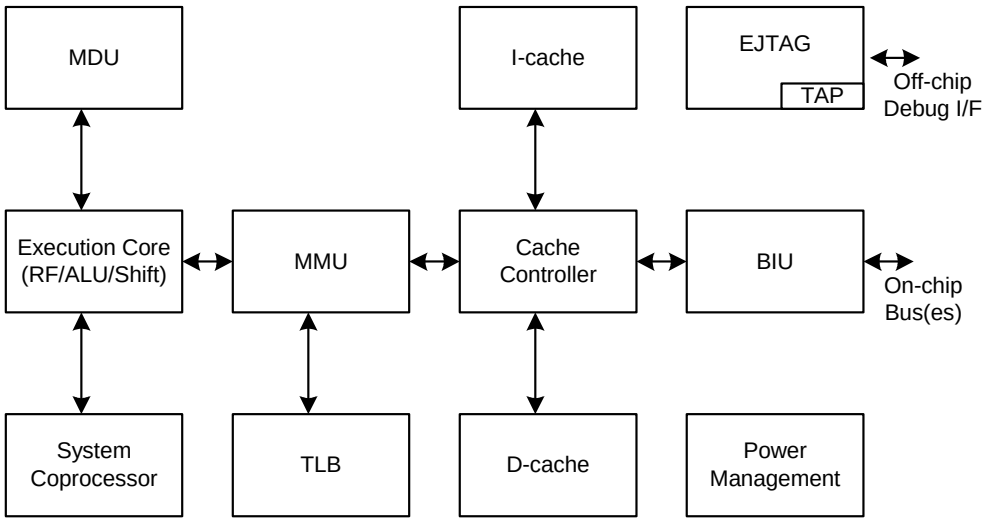


Figure 2-2 **4KEc core block diagram**

2.6 Functional Description

2.6.1 4KEc Core Logic Blocks

The 4KEc core contains the following blocks:

- Execution Unit
- Multiply/Divide Unit (MDU)
- System Control Coprocessor (CP0)
- Memory Management Unit (MMU)
- Transition Lookaside Buffer (TLB)
- Cache Controllers
- Bus Interface Unit (BIU)
- Power Management
- Instruction Cache
- Data Cache
- Enhanced JTAG (EJTAG) Controller

2.6.2 Pipeline Flow

The 4KEc core pipeline consists of five stages:

1. Instruction (I Stage)
2. Execution (E Stage)
3. Memory (M Stage)
4. Align (A Stage)
5. Writeback (W stage)

The 4KEc core implements a bypass mechanism that allows the result of an operation to be forwarded directly to the instruction that needs it without having to write the result to the register and then read it back. The following diagram shows a timing diagram of the 4KEc core pipeline.

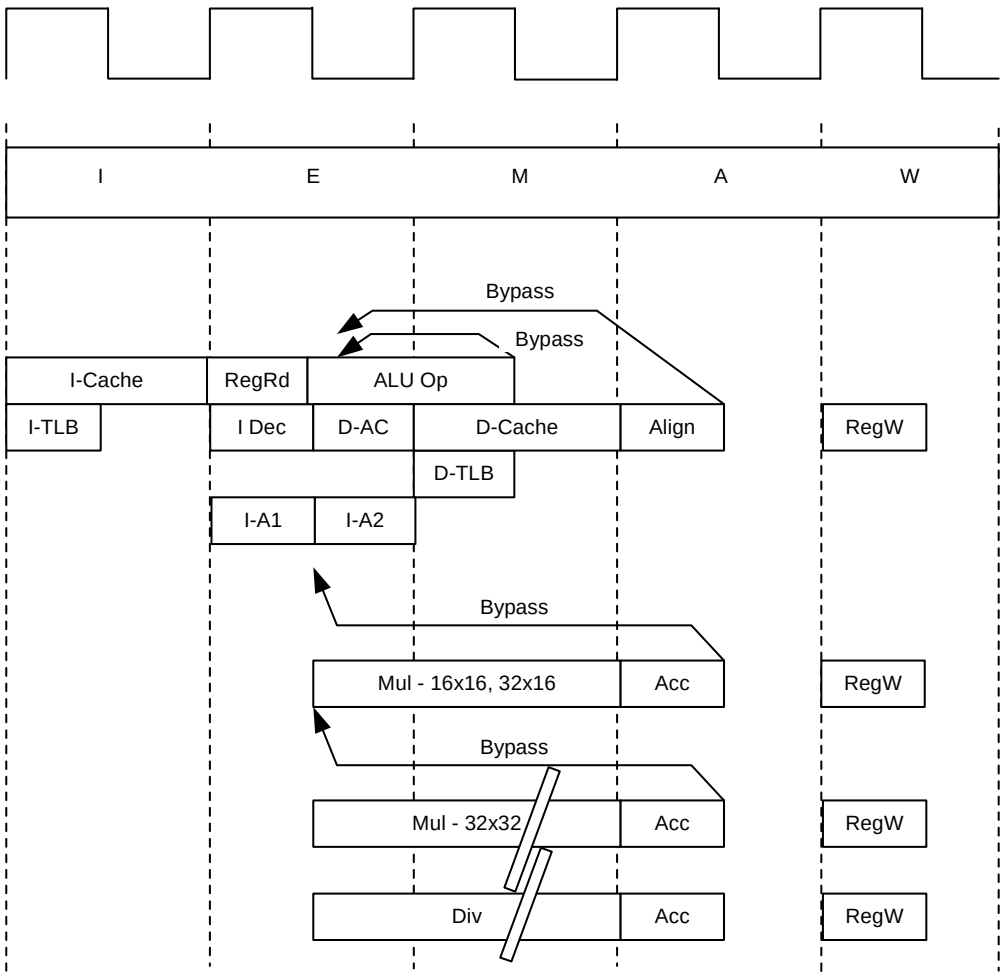


Figure 2-3 **4KEc core pipeline**

2.6.2.1 I Stage: Instruction Fetch

During the Instruction fetch stage:

- An instruction is fetched from the instruction cache
- MIPS16e instructions are expanded into MIPS32-like instructions

2.6.2.2 E Stage: Execution

During the Execution stage:

- Operands are fetched from the register file
- The Arithmetic Logic Unit (ALU) begins the arithmetic or the logical operation for register-to-register instructions
- The ALU calculates the data virtual address for load and store instructions
- The ALU determines whether the branch condition is true and calculates the virtual branch target address for branch instructions
- Instruction logic selects an instruction address
- All multiply and divide operations begin in this stage

2.6.2.3 M Stage: Memory Fetch

During the Memory fetch stage:

- The arithmetic ALU operation completes
- The data cache access and the data virtual-to-physical address translation are performed for load and store instructions
- Data cache look-up is performed and a hit/miss determination is made
- A 16x16 or 32x16 multiply calculation completes
- A 32x32 multiply operation stalls the MDU pipeline for one clock in the M stage
- A divide operation stalls the MDU pipeline for a maximum of 34 clocks in the M stage. Early-in sign extension detection on the dividend will skip 7, 15 or 23 stall clocks.

2.6.2.4 A Stage: Align

During the Align stage:

- Load data is aligned to its word boundary
- A 16x16 or 32x16 multiply operation performs the carry-propagate-add. The actual register writeback is performed in the W stage.
- A MUL operation makes the result available for writeback. The actual register write-back is performed in the W stage.

2.6.2.5 W Stage: Writeback

During the Writeback stage:

- For register-to-register or load instructions, the instruction result is written back to the register file.

2.6.3 Execution Unit

The 4KEc core execution unit implements a load/store architecture with single-cycle ALU operations (logical, shift, add and subtract) and an autonomous multiply/divide unit. The 4KEc core contains several 32-bit general purpose registers used for integer operations and address calculation.

The execution unit includes:

- A 32-bit adder used for calculating the data address
- An address unit for calculating the next instruction address
- A logic for branch determination and branch target address calculation
- A load aligner
- Bypass multiplexers used to avoid stalls when executing instructions streams where data producing instructions are followed closely by consumers of their results
- A leading zero/one detect unit for implementing the CLZ and CLO instructions
- An Arithmetic Logic Unit (ALU) for performing bitwise logical operations
- A shifter and store aligner

2.6.4 Multiply/Divide Unit (MDU)

The 4KEc core includes a multiply/divide unit (MDU) that contains a separate pipeline for multiply and divide operations. This pipeline operates in parallel with the integer unit (IU) pipeline and does not stall when the IU pipeline stalls. This setup allows long-running MDU operations, such as a divide, to be partially masked by system stalls and/or other integer unit instructions.

The MDU consists of a 32x16 booth recoded multiplier, result/accumulation registers (HI and LO), a divide state machine, and the necessary multiplexers and control logic. The first number shown (‘32’ of 32x16) represents the *rs* operand. The second number (‘16’ of 32x16) represents the *rt* operand. The 4KEc core only checks the value of the latter (*rt)* operand to determine how many times the operation must pass through the multiplier. The 16x16 and 32x16 operations pass through the multiplier once. A 32x32 operation passes through the multiplier twice. The MDU supports execution of one 16x16 or 32x16 multiply operation every clock cycle; 32x32 multiply operations can be issued every other clock cycle. Appropriate interlocks are implemented to stall the issuance of back-to-back 32x32 multiply operations. The multiply operand size is automatically determined by the logic built into the MDU.

Divide operations are implemented with a simple 1-bit per clock iterative algorithm. An early-in detection checks the sign extension of the dividend (*rs*) operand. If *rs* is 8 bits wide, then 23 iterations are skipped. For a 16-bit wide *rs*, 15 iterations are skipped, and for a 24-bit wide *rs*, 7 iterations are skipped. Any attempt to issue a subsequent MDU instruction while a divide is still active causes an IU pipeline stall until the divide operation is completed. The following table lists the repeat rate (peak issue rate of cycles until the operation can be reissued) and latency (number of cycles until a result is available) for the 4KEc core multiply and divide instructions. The approximate latency and repeat rates are listed in terms of pipeline clocks.

Table 2-2 **4KEc core latencies and repeat rates**

Opcode	Operand Size (mul rt) (div rs)	Latency	Repeat Rate
MULT/MULTU, MADD/MADDU, MSUB/MSUBU	16-bit	1	1
	32-bit	2	2
MUL	16-bit	2	1
	32-bit	3	2
DIV/DIVU	8-bit	12	11
	16-bit	19	18
	24-bit	26	25
	32-bit	32	33

The processor architecture defines that the result of a multiply or divide operation be placed in the HI and LO registers. Using the Move-From-HI (MFHI) and Move-From-LO (MFLO) instructions, these values can be transferred to the general purpose register file.

In addition to the HI/LO targeted operations, the processor architecture also defines a multiply instruction, MUL, which places the least significant results in the primary register file instead of the HI/LO register pair. By avoiding the explicit MFLO instruction, required when using the LO register, and by supporting multiple destination registers, the throughput of multiply-intensive operations is increased.

Two other instructions, multiply-add (MADD) and multiply-subtract (MSUB), are used to perform the multiply-accumulate and multiply-subtract operations. The MADD instruction multiplies two numbers and then adds the product to the current contents of the HI and LO registers. Similarly, the MSUB instruction multiplies two operands and then subtracts the product from the HI and LO registers. The MADD and MSUB operations are commonly used in the DSP algorithms.

2.6.5 System Control Coprocessor (CP0)

In the processor architecture, CP0 is responsible for the virtual-to-physical address translation and the cache protocols, the exception control system, the processor's diagnostics capability, the operating modes (kernel, user and debug) and whether interrupts are enabled or disabled. Configuration information such as, cache size and set associativity is also available by accessing the CP0 registers listed in the table below:

Table 2-3 **Coprocessor 0 registers in numerical order**

Register Number	Register Name	Register Usage	Function
0	INDEX	Memory management	Index into the TLB array
1	RANDOM	Memory management	Randomly generated index into the TLB array
2	ENTRYLO0	Memory management	Low-order portion of the TLB entry for odd-numbered virtual pages
3	ENTRYLO1	Memory management	Low-order portion of the TLB entry for odd-numbered virtual pages
4	CONTEXT	Exception processing	Pointer to page table entry in memory
4	CONTEXT-CONFIG	Exception processing	Controls the layout of the Context register
5	PAGEMASK	Memory management	Control for variable page sizes in TLB entries
5	PAGEGRAIN	Memory management	Controls the layout of the ENTRYLO, PAGEMASK and ENTRYHI registers

Table 2-3 **Coprocessor 0 registers in numerical order (Continued)**

Register Number	Register Name	Register Usage	Function
6	WIRED	Memory management	Controls the number of fixed (‘wired’) TLB entries
7	HWRENA		Enables access via the RDHWR instruction to selected hardware registers
8	BADVADDR	Exception processing	Reports the address for the most recent address-related exception
9	COUNT	Exception processing	Processor cycle count
10	ENTRYHI	Memory management	High-order portion of the TLB entry
11	COMPARE	Exception processing	Timer interrupt control
12	STATUS	Exception processing	Processor status and control
12	INTCTL	Exception processing	Interrupt system status and control
12	SRSCTL	Exception processing	Shadow register set status and control
12	SRSMA	Exception processing	Provides mapping from vectored interrupt to a shadow set
13	CAUSE	Exception processing	Cause of last general exception
14	EPC	Exception processing	Program counter at last exception
15	PRID		Processor identification and revision
15	EBASE		Exception vector base register
16	CONFIG		Configuration register
16	CONFIG1		Configuration register 1
16	CONFIG2		Configuration register 2
16	CONFIG3		Configuration register 3
17	LLADDR		Load linked address
18	WATCHLO	Exception processing	Low-order watchpoint address
19	WATCHHI	Exception processing	High-order watchpoint address
20-22	Reserved		
23	DEBUG	Debug	Debug control and exception status
23	TRACECONTROL	Debug	PC/Data trace control register
23	TRACECONTROL2	Debug	Additional PC/Data trace control
23	USE TRACEDATA	Debug	User Trace control register
23	TRACEBPC	Debug	Trace breakpoint control
24	DEPC	Debug	Program counter at last debug exception

Table 2-3 **Coprocessor 0 registers in numerical order (Continued)**

Register Number	Register Name	Register Usage	Function
25	Reserved		
26	ERRCTL		Used for software testing of cache arrays
27	Reserved		Reserved in the 4KEc core
28	TAGLO/DATALO		Low-order portion of cache tag interface
29	Reserved		
30	ERROREPC	Exception processing	Program counter at last error
31	DESAVE	Debug	Debug handler scratchpad register

The coprocessor 0 also contains the logic for identifying and managing exceptions. The exceptions can be caused by a variety of sources, including boundary cases in data, external events, or program errors. The following table shows the exception types in the order of priority:

Table 2-4 **4KEc core exception type**

Exception	Description
Reset	Assertion of SI_COLDRESET or SI_RESET signals
DSS	EJTAG debug single step
DINT	EJTAG debug interrupt. Caused by the assertion of the external EJ_DINT input, or by setting the EJTAGBRK bit in the ECR register.
NMI	Assertion of EB_NMI signal
Machine Check	TLB write that conflicts with an existing entry
Interrupt	Assertion of unmasked hardware or software interrupt signal
Deferred Watch	Deferred Watch (unmasked by K|DM->!(K|DM) transition)
DIB	EJTAG debug hardware instruction break matched
WATCH	A reference to an address in one of the watch registers (fetch)
AdEL	Fetch address alignment error. Fetch reference to protected address.
TLBL	Fetch TLB miss
TLBL	Fetch TLB hit to page with V=0
IBE	Instruction fetch bus error
DBp	EJTAG breakpoint (execution of SDBBP instruction)
Sys	Execution of SYSCALL instruction
Bp	Execution of BREAK instruction

Table 2-4 **4KEc core exception type (Continued)**

Exception	Description
RI	Execution of a reserved instruction
CpU	Execution of a coprocessor instruction for a coprocessor that is not enabled
Ov	Execution of an arithmetic instruction that overflowed
Tr	Execution of a trap (when trap condition is true)
DDBL/DDBS	EJTAG data address break (address only) or EJTAG data value break on store (address+value)
WATCH	A reference to an address in one of the watch registers (data)
AdEL	Load address alignment error. Load reference to protected address.
AdES	Store address alignment error. Store to protected address.
TLBL	Load TLB miss
TLBL	Load TLB hit to page with V=0
TLBS	Store TLB miss
TLBS	Store TLB hit to page with V=0
TLB Mod	Store to TLB page with D=0
DBE	Load or store bus error
DDBL	EJTAG data hardware breakpoint matched in load data compare

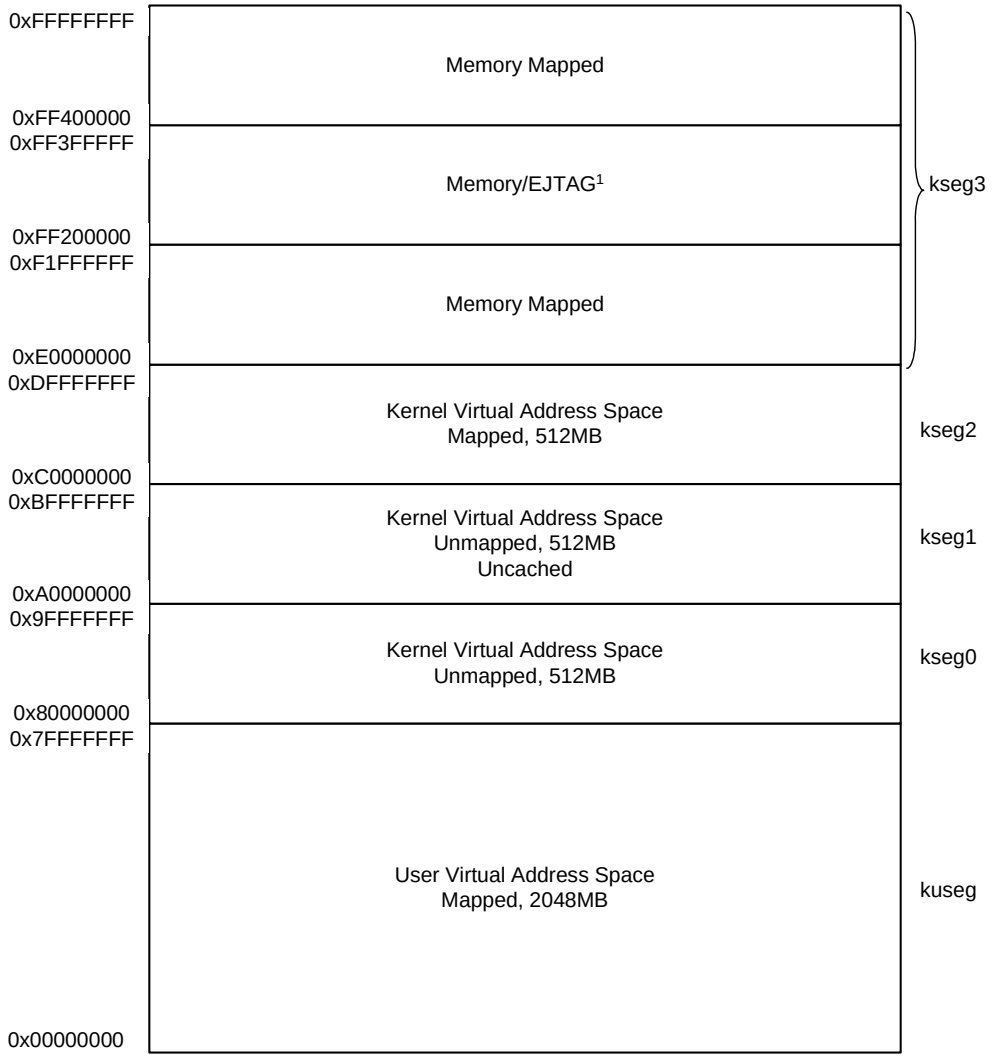
2.6.5.1 Interrupt Handling

The 4KEc core includes support for six hardware interrupt pins, two software interrupts and a timer interrupt. These interrupts can be used in any of the following two interrupt modes:

- Interrupt compatibility mode
- External Interrupt Controller (EIC) mode, which redefines the way in which interrupts are handled to provide full support for an external interrupt controller handling prioritization and vectoring of interrupts. The presence of this mode is denoted by the VEIC bit in the CONFIG3 register. On the 4KEc core, the VEIC bit is set externally by the static input, SI_EICPRESENT, to allow the system logic to indicate the presence of an external interrupt controller.

2.6.5.2 Modes of Operation

The 4KEc core supports three modes of operation: user mode, kernel mode and debug mode. The user mode is most often used for applications programs. The kernel mode is typically used for handling exceptions and operating system kernel functions, including CP0 management and I/O device accesses. An additional debug mode is used during system bring-up and software development. Refer to the EJTAG section outlined later in this chapter for more information on the debug mode.



1. This space is mapped to the memory in the user or the kernel mode, and by the EJTAG module in the debug mode.

Figure 2-4 **4KEc core virtual address map**

2.6.6 Memory Management Unit (MMU)

The 4KEc core contains a fully functional MMU that interfaces between the execution unit and the cache controller. Although the 4KEc core implements a 32-bit architecture, the MMU has a 64-bit architecture.

The 4KEc core implements a TLB-based MMU. The TLB consists of three translation buffers: a 32 dual-entry fully associative Joint TLB (JTLB), a 4-entry fully associative Instruction TLB (ITLB) and a 4-entry fully associative Data TLB (DTLB).

When an instruction address is calculated, the virtual address is compared to the contents of the 4-entry ITLB. If the address is not found in the ITLB, then the JTLB is accessed. If the entry is found in the JTLB, then that entry is then written into the ITLB. If the address is not found in the JTLB, then a TLB refill exception is taken.

When a data address is calculated, the virtual address is compared to both the 4-entry DTLB and the JTLB. If the address is not found in the DTLB, but is found in the JTLB, then that address is immediately written to the DTLB. If the address is not found in the JTLB, then a TLB refill exception is taken. The following figure shows how the ITLB, DTLB, and JTLB are implemented in the 4KEc core:

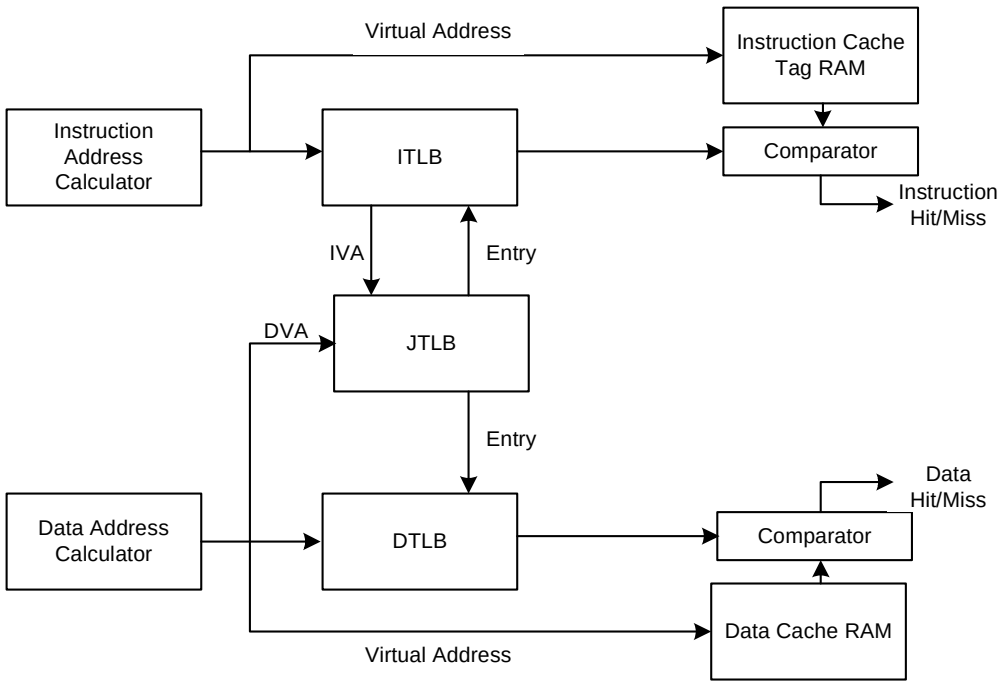


Figure 2-5 **Address translation during a cache access**

2.6.7 Translation Lookaside Buffer (TLB)

The TLB consists of three address translation buffers:

- 32 dual-entry fully associative Joint TLB (JTLB)
- 4-entry fully associative Instruction TLB (ITLB)
- 4-entry fully associative Data TLB (DTLB)

2.6.7.1 Joint TLB (JTLB)

The 4KEc core implements a 32-dual-entry, fully associative JTLB that maps 32 virtual pages to their corresponding physical addresses. The purpose of the TLB is to translate virtual addresses and their corresponding ASIDs into a physical memory address. The translation is performed by comparing the upper bits of the virtual address (along with the ASID) against each of the entries in the *tag* portion of the joint TLB structure.

The JTLB is organized as 32 pairs of even and odd entries containing pages that range in size from 4KB (or 1KB) to 256MB into the 4GB physical address space. By default, the minimum page size is normally 4KB on the 4KEc core; as a build time option, it is possible to specify a minimum page size of 1KB.

The JTLB is organized in page pairs to minimize the overall size. Each *tag* entry corresponds to 2 data entries: an even page entry and an odd page entry. The highest order virtual address bit not participating in the tag comparison is used to determine which of the data entries is used. Since the page size can vary on a page-pair basis, the determination of which address bits participate in the comparison and which bit is used to make the even-odd determination is decided dynamically during the TLB look-up.

2.6.7.2 Instruction TLB (ITLB)

The ITLB is a small 4-entry, fully associative TLB dedicated to performing the translations for the instruction stream. The ITLB only maps minimum sized pages/sub pages. The minimum page size is either 1KB or 4KB, depending on the PAGEGRAIN and CONFIG3 registers.

The ITLB is managed by the hardware and is transparent to the software. The larger JTLB is used as a backing store for the ITLB. If a fetch address cannot be translated by the ITLB, then the JTLB is used to attempt to translate it in the following clock cycle. If successful, then the translation information is copied into the ITLB for future use. There is a two cycle ITLB miss penalty.

2.6.7.3 Data TLB (DTLB)

The DTLB is a small 4-entry, fully associative TLB dedicated to performing the translations for loads and stores. Similar to the ITLB, the DTLB only maps either 1KB or 4KB pages/sub pages depending on the PAGEGRAIN and CONFIG3 registers.

The DTLB is managed by the hardware and is transparent to the software. The larger JTLB is used as a backing store for the DTLB. The JTLB is looked up in parallel with the DTLB to minimize the DTLB miss penalty. If the JTLB translation is successful, the translation information is copied into the DTLB for future use. There is a one cycle DTLB miss penalty.

2.6.7.4 Virtual-to-Physical Address Translation

Converting a virtual address to a physical address begins by comparing the virtual address from the processor with the virtual addresses in the TLB; there is a match when the virtual page number (VPN) of the address is the same as the VPN field of the entry, and either the Global (*G*) bit of the TLB entry is set, or the ASID field of the virtual address is the same as the ASID field of the TLB entry.

This match is referred to as a TLB *hit*. If there is no match, then a TLB *miss* exception is taken by the processor and the software is allowed to refill the TLB from a page table of virtual/physical addresses in memory. The following figure shows a flow diagram of the address translation process for two different page sizes:

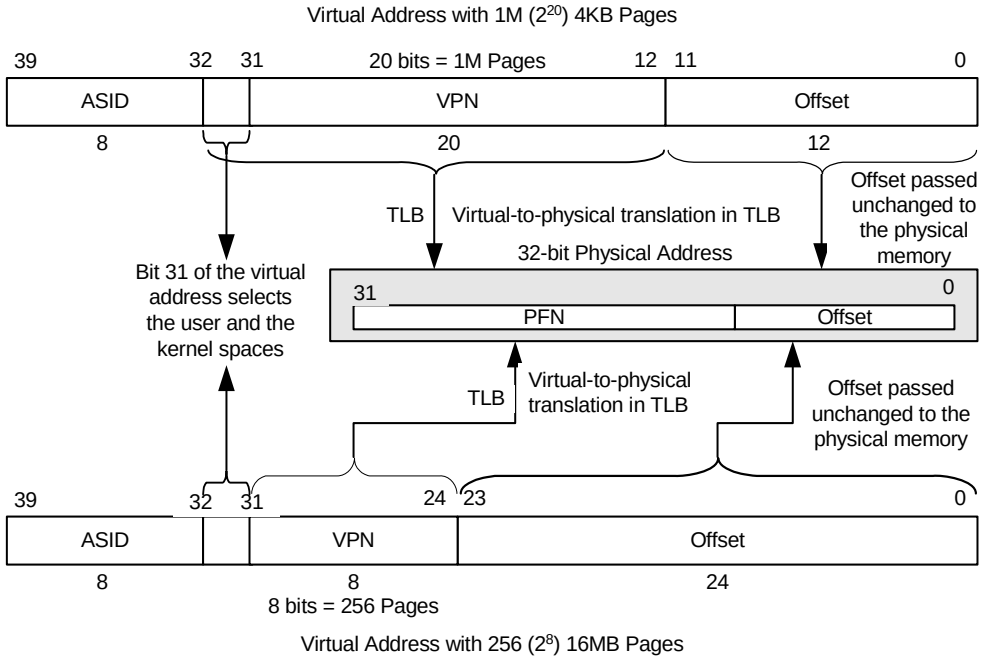


Figure 2-6 **32-bit virtual address translation**

The top portion of the above figure shows a virtual address for a 4KB page size. The width of the *Offset* in the above figure is defined by the page size. The remaining 20 bits of the address represent the virtual page number (VPN), and index the 1M-entry page table.

The bottom portion of the above figure shows the virtual address for a 16MB page size. The remaining 8 bits of the address represent the VPN, and index the 256-entry page table.

In the above figure, the virtual address is extended with an 8-bit address space identifier (ASID), which reduces the frequency of TLB flushing during a context switch. This 8-bit ASID contains the number assigned to that process and is stored in the CP0 ENTRYHI register.

2.6.7.5 Hits, Misses and Multiple Matches

Each JTLB entry contains a tag portion and a data portion. If a match is found, then the upper bits of the virtual address are replaced with the page frame number (PFN) stored in the corresponding entry in the data array of the joint TLB (JTLB). The granularity of JTLB mappings is defined in terms of TLB pages. The 4KEc core's JTLB supports pages of different sizes ranging from 1KB to 256MB in powers of 4.

If no match occurs (TLB miss), an exception is taken and software refills the TLB from the page table resident in memory. Software can write over a selected TLB entry or use a hardware mechanism to write into a random entry.

The 4KEc core implements a TLB write compare mechanism to ensure that multiple TLB matches do not occur. On the TLB write operation, the write value is compared with all other entries in the TLB. If a match occurs, then the 4KEc core takes a machine check exception, sets the TS bit in the CP0 STATUS register, and aborts the write operation.

The following table shows the address bits used for even/odd bank selection depending on page size and the relationship between the legal values in the mask register and the selected page size:

Table 2-5 **Mask and page size values**

Pagemask[28:11]	Page Size	Even/Odd Bank Select Bit
000000000000000000	1KB (if present)	VAddr[10]
000000000000000011	4KB	VAddr[12]
000000000000001111	16KB	VAddr[14]
000000000000111111	64KB	VAddr[16]
000000000011111111	25KB	VAddr[18]
000000001111111111	1MB	VAddr[20]

Table 2-5 **Mask and page size values (Continued)**

Pagemask[28:11]	Page Size	Even/Odd Bank Select Bit
000000111111111111	4MB	VAddr[22]
000011111111111111	16MB	VAddr[24]
001111111111111111	64MB	VAddr[26]
111111111111111111	256MB	VAddr[28]

2.6.7.6 TLB Tag and Data Formats

The entry is divided into the following fields:

- Global process indicator
- Address space identifier
- Virtual page number
- Compressed page mask

Setting the global process indicator (G bit) indicates that the entry is global to all the processes and/or threads in the system. In this case, the 8-bit address space identifier (ASID) value is ignored since the entry is not relative to a specific thread or process.

The ASID helps to reduce the frequency of the TLB flushing on a context switch. The existence of the ASID allows multiple processes to exist in both the TLB and instruction caches. The current ASID value is stored in the ENTRYHI register and is compared to the ASID value of each entry. The following figure shows the TLB tag entry format:

G	ASID[7:0]	VPN2[31:25]	VPN2[24:11]	CMASK[7:0]
1	8	7	14	8

Figure 2-7 **TLB tag entry format**

The following table shows the TLB tag entry fields:

Table 2-6 **TLB tag entry fields**

Field Name	Description
G	Global Bit. When set, indicates that this entry is global to all the processes and/or the threads and thus disables the inclusion of the ASID in the comparison.
ASID[7:0]	Address Space Identifier. Identifies with which process or thread this TLB entry is associated.
VPN2[31:25], VPN2[24:11]	Virtual Page Number divided by 2. This field contains the upper bits of the virtual page number. Because it represents a pair of TLB pages, it is divided by 2. Bits [31:25] are always included in the TLB lookup comparison. Bits [24:11] are included depending on the page size.
CMASK[7:0]	Compressed Page Mask Value. This field is a compressed version of the page mask. It defines the page size by masking the appropriate VPN2 bits from being involved comparison. It is also used to determine which address bit is used to make the even/ odd page determination.

The following figure shows the TLB data array entry format:

PFN([31:12] or [29:10])	C[2:0]	D	V
20	3	1	1

Figure 2-8 **TLB data array entry format**

The following table shows the TLB data array entry fields:

Table 2-7 **TLB data array entry fields**

Field Name	Description
PFN[31:12] or [29:10])	Physical Frame Number. Defines the upper bits of the physical address. The [29:10] range illustrates, that if 1KB page granularity is enabled in the PAGEGRAIN register, the PFN is shifted to the right, before being appended to the part of the virtual address that is not translated. In this mode the upper two physical address bits are not covered by PFN but forced tozero.For page sizes larger than the minimum configured page size, only a subset of these bits is actually used.
C[2:0]	Cacheability. Contains an encoded value of the cacheability attributes and determines whether the page should be placed in the cache or not.
D	'Dirty' or write-enable bit. Indicates that the page has been written and/or is writable. If this bit is set, stores to the page are permitted. If the bit is cleared, stores to the page cause a TLB Modified exception.
V	Valid bit. Indicates that the TLB entry, and thus the virtual page mapping, are valid. If this bit is set, accesses to the page are permitted. If the bit is cleared, accesses to the page cause a TLB Invalid exception.

2.6.7.7 Page Sizes and Replacement Algorithm

To assist in controlling both the amount of mapped space and the replacement characteristics of various memory regions, the 4KEc core provides two mechanisms. First, the page size can be configured, on a per-entry basis, to map a page size of 1KB to 256MB (in multiples of 4). The CP0 PAGEMASK register is loaded with the mapping page size, which is then entered into the TLB when a new entry is written. Thus, operating systems can provide special purpose maps. For example, a typical frame buffer can be memory mapped with only one TLB entry.

The second mechanism controls the replacement algorithm when a TLB miss occurs. To select a TLB entry to be written with a new mapping, the 4KEc core provides a random replacement algorithm. However, the processor also provides a mechanism where a programmable number of mappings can be locked into the TLB via the CP0 WIRED register, thus avoiding random replacement.

2.6.8 Cache Controllers

The 4KEc core contains separate, 16KB, 2-way set associative instruction and data cache controllers. Each cache can each be accessed in a single processor cycle. In addition, each cache has its own 32-bit data path and both caches can be accessed in the same pipeline clock cycle. Refer to the section on instruction and data cache outlined in this chapter for more information on instruction and data cache organization.

2.6.8.1 Hardware Reset

For historical reasons within the processor architecture, the 4KEc core has two types of reset input signals: SI_RESET and SI_COLDRESET.

Functionally, these two signals are ORed together within the core and then used to initialize critical hardware state. Both reset signals can be asserted either synchronously or asynchronously to the core clock, SI_CLKIN, and will trigger a Reset exception. The reset signals are active high, and must be asserted for a minimum of 5 SI_CLKIN cycles. The falling edge triggers the Reset exception. The primary difference between the two reset signals is that the signal SI_RESET sets a bit in the STATUS register. This bit could be used by the software to distinguish between the two reset signals if desired.

The reset behavior is summarized in the following table:

Table 2-8 **4KEc reset types**

SI_RESET	SI_COLDRESET	Action
0	0	Normal operation, no reset.
1	0	Reset exception; sets the STATUS.SR bit.
X	1	Reset exception

One (or both) of the reset signals must be asserted at power-on or whenever the hardware initialization of the core is desired. A power-on reset typically occurs when the machine is turned on initially. A hard reset usually occurs when the machine is already on and the system is rebooted. In the debug mode, EJTAG can request that a soft reset (via the SI_RESET pin) be masked. It is system dependent whether this functionality is supported. In normal mode, the SI_RESET pin cannot be masked. The SI_COLDRESET pin is never masked.

2.6.9 Power Management

The 4KEc core offers a number of power management features, including low-power design, active power management, and power-down modes of operation. The core is a static design that supports slowing or halting of the clocks, which reduces system power consumption during idle periods. The 4KEc core provides two mechanisms for system-level low power support:

- Register-controlled power management
- Instruction-controlled power management

2.6.9.1 Register-Controlled Power Management

The RP bit in the CP0 Status register provides a software mechanism for placing the system into a low power state. The state of the RP bit is available externally via the SI_RP signal. The external agent then decides whether to place the device in a low power mode, such as reducing the system clock frequency.

Three additional bits, $Status_{EXL}$, $Status_{ERL}$, and $Debug_{DM}$ support the power management function by allowing the user to change the power state if an exception or error occurs while the 4KEc core is in a low power state. Depending on what type of exception is taken, one of these three bits will be asserted and reflected on the SI_EXL, SI_ERL, OR EJ_DEBUGM outputs. The external agent can look at these signals and determine whether to leave the low power state to service the exception.

The following 4 power-down signals are part of the system interface and change state as the corresponding bits in the CP0 registers are set or cleared:

- The SI_RP signal represents the state of the RP bit (27) in the CP0 Status register
- The SI_EXL signal represents the state of the EXL bit (1) in the CP0 Status register
- The SI_ERL signal represents the state of the ERL bit (2) in the CP0 Status register
- The EJ_DEBUGM signal represents the state of the DM bit (30) in the CP0 Debug register

2.6.9.2 Instruction-Controlled Power Management

The second mechanism for invoking power-down mode is through the execution of the WAIT instruction. When the WAIT instruction is executed, the internal clock is suspended; however, the internal timer and some of the input pins (SI_INT[5:0], SI_NMI, SI_RESET and SI_COLDRESET) continue to run. Once the CPU is in instruction-controlled power management mode, any interrupt, NMI, or reset condition causes the CPU to exit this mode and resume normal operation.

The 4KEc core asserts the SI_SLEEP signal which is part of the system interface bus, whenever the WAIT instruction is executed. The assertion of SI_SLEEP indicates that the clock has stopped and the 4KEc core is waiting for an interrupt.

2.6.10 Instruction Cache

The instruction cache is an on-chip memory block of 16KB. Because the instruction cache is virtually indexed, the virtual-to-physical address translation occurs in parallel with the cache access rather than having to wait for the physical address translation. The tag holds 22 bits of physical address, a valid bit, and a lock bit. The LRU replacement bits (0-6b per set depending on associativity) are stored in a separate array.

The instruction cache block also contains and manages the instruction line fill buffer. Besides accumulating the data to be written to the cache, instruction fetches that reference data in the line fill buffer are serviced either by a bypass of that data, or the data coming from the external interface. The instruction cache control logic controls the bypass function.

The 4KEc core supports instruction-cache locking. Cache locking allows critical code or data segments to be locked into the cache on a “per-line” basis, enabling the system programmer to maximize the efficiency of the system cache.

The cache-locking function is always available on all instruction-cache entries. Entries can then be marked as locked or unlocked on a per entry basis using the cache instruction.

2.6.11 Data Cache

The data cache is an on-chip memory block of 16KB. This virtually indexed, physically tagged cache is protected. Because the data cache is virtually indexed, the virtual-to-physical address translation occurs in parallel with the cache access. The tag holds 22 bits of physical address, a valid bit and a lock bit. There is an additional array holding dirty bits and LRU replacement algorithm bits (0-6b depending on associativity) for each set of the cache.

In addition to instruction-cache locking, the 4KEc core also supports a data-cache locking mechanism identical to the instruction cache. Critical data segments are locked into the cache on a ‘per-line’ basis. The locked contents can be updated on a store hit, but cannot be selected for replacement on a cache miss. The cache-locking function is always available on all the data cache entries. Entries can then be marked as locked or unlocked on a per-entry basis using the cache instruction.

2.6.11.1 Cache Memory Configuration

The 4KEc core incorporates on-chip instruction and data caches that can each be accessed in a single processor cycle. Each cache has its own 32-bit data path and can be accessed in the same pipeline clock cycle. The following table lists the 4KEc core instruction and data cache attributes:

Table 2-9 **4KEc core instruction and data cache attributes**

Parameter	Instruction	Data
Size	16KB	16KB
Organization	2-way set associative	2-way set associative
Line size	16-byte	16-byte
Read unit	32-bit	32-bit
Write policies	NA	Write-through with write allocate, Write-through without write allocate, Write-back with write allocate.
Miss restart after transfer of	Miss Word	Miss Word
Cache locking	Per Line	Per Line

2.6.11.2 Cache Protocols

The 4KEc core supports the following cache protocols:

- Uncached: Addresses in a memory area indicated as uncached are not read from the cache. Stores to such addresses are written directly to the main memory, without changing the cache contents.
- Write-through, no write allocate: Loads and instruction fetches first search the cache, reading the main memory only if the desired data does not reside in the cache. On data store operations, the cache is first searched to see if the target address is cache resident. If it is resident, then the cache contents are updated and the main memory is also written. If the cache look-up misses, then only the main memory is written.
- Write-through, write allocate: Similar to above, but stores missing in the cache will cause a cache refill. The store data is then written to both the cache and the main memory.
- Write-back, write allocate: Stores that miss in the cache will cause a cache refill. Store data, however, is only written to the cache. Cache lines that are written by stores will be marked as dirty. If a dirty line is selected for replacement, then the cache line will be written back to the main memory.

2.6.12 EJTAG Debug Support

The 4KEc core provides for an optional Enhanced JTAG (EJTAG) interface for use in the software debug of application and the kernel code. In addition to the standard user mode and the kernel modes of operation, the 4KEc core provides a Debug mode that is entered after a debug exception (derived from a hardware breakpoint, single-step exception, etc.) is taken and continues until a debug exception return (DERET) instruction is executed. During this time, the processor executes the debug exception handler routine.

The EJTAG interface operates through the Test Access Port (TAP), a serial communication port used for transferring test data in and out of the 4KEc core. In addition to the standard JTAG instructions, special instructions defined in the EJTAG specification define what registers are selected and how they are used.

2.6.12.1 Debug Registers

Three debug registers (DEBUG, DEPC and DESAVE) have been added to the processor Coprocessor 0 (CP0) register set. The DEBUG register shows the cause of the debug exception and is used for setting up single-step operations.

The Debug Exception Program Counter Register (DEPC) holds the address on which the debug exception was taken. This is used to resume program execution after the debug operation finishes. Finally, the debug exception save register (DESAVE) enables the saving of general purpose registers used during execution of the debug exception handler.

To exit the debug mode, a debug exception return (DERET) instruction is executed. When this instruction is executed, the system exits the debug mode, allowing normal execution of application and system code to resume.

2.6.12.2 EJTAG Hardware Breakpoints

There are several types of simple hardware breakpoints defined in the EJTAG specification. These stop the normal operation of the CPU and force the system into the debug mode. There are two types of simple hardware breakpoints implemented in the 4KEc core: instruction breakpoints and data breakpoints. The 4KEc core implements two data and four instruction breakpoints.

The instruction breaks occur on instruction fetch operations, and the break is set on the virtual address. Instruction breaks can also be made on the ASID value used by the MMU. A mask can be applied to the virtual address to set the breakpoints on a range of instructions.

The data breakpoints occur on load/store transactions. The breakpoints are set on virtual address and ASID values, similar to the Instruction breakpoint. Data breakpoints can be set on a load, a store or both. The data breakpoints can also be set based on the value of the load/store operation. Finally, masks can be applied to both the virtual address and the load/store value.

2.6.12.3 EJTAG Pin Description

Table 2-10 **EJTAG pin descriptions**

Pin Name	Ball ID	Direction	Description
JTAG_UART#	D11	I	A high level on this pin enables the JTAG interface of certain UART1 pins (listed in this table). This pin includes an internal pull-down resistor.
UART1_DSR	B11	I	EJ_TRST_N. Active-low Test Reset Input (TRST) for the EJTAG TAP. At power-up, the assertion of EJ_TRST_N causes the TAP controller to be reset.
UART1_DCD	C11	I	EJ_TCK. Test clock input (TCK) for the EJTAG TAP.
UART1_CTS	C12	I	EJ_TMS. Test mode select input (TMS) for the EJTAG TAP.
UART1_RX	A12	I	EJ_TDI. Test data input (TDI) for the EJTAG TAP.
UART1_RTS	B10	O	EJ_TDO. Test data output (TDO) for the EJTAG TAP.

2.6.12.4 Electrical Characteristics

EJTAG AC Electrical Characteristics

Table 2-11 **EJTAG AC characteristics**

Symbol	Units	Min	Typ	Max
T_{TCK}	ns	25		
T_{TCKH}	ns	10		
T_{TCKL}	ns	10		
T_{SU}	ns	5		
T_{H}	ns	3		
T_{TDODLY}	ns			5
T_{TDOZ}	ns			5
T_{TRSTL}	ns	25		
T_{rf}	ns			3

2.6.12.5 Timing Diagram

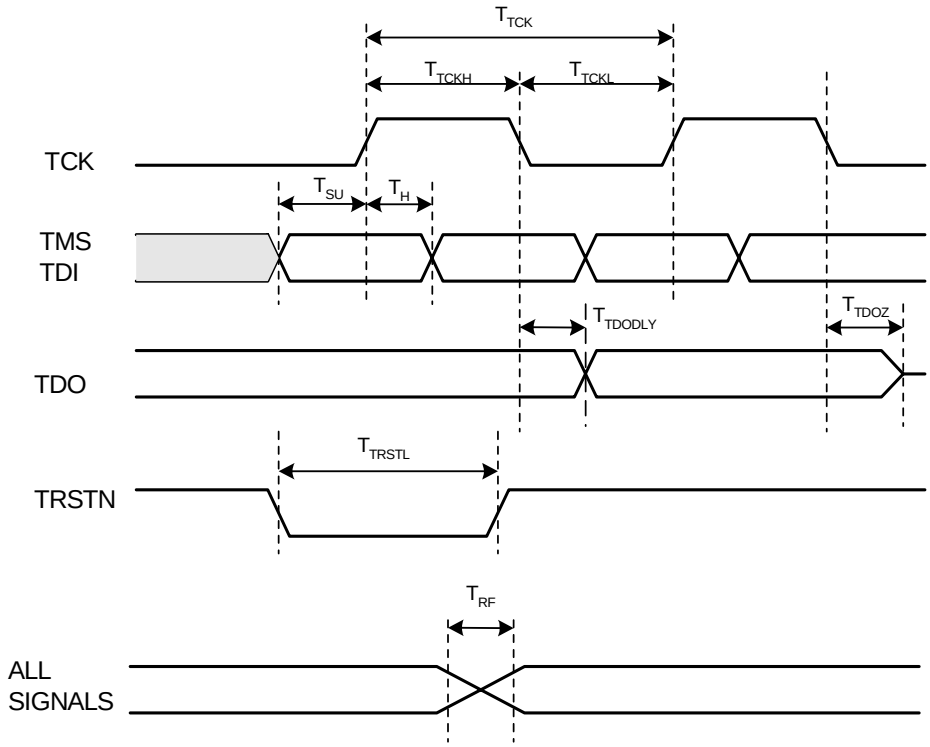


Figure 2-9 **EJTAG timing diagram**

Local Memory

2.7 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.

2.8 Features

- 2Kx32 memory
- Supports byte, word and dword accesses
- Supports 8, 16 and 32-bit reads and writes

2.9 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.

2.9.1 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_IRQ0_VEC, CPU_IRQ1_VEC and CPU_IRQ2_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.

2.10 Register Map

2.10.1 Local Memory Registers

Table 2-12 **Host CPU block local memory registers**

Address[1]	Register Name	R/W/A[2]	Description
+6000	CPU_RESET_VEC	R/W	CPU Reset Vector Register
+6004	CPU_UNDEF_VEC	R/W	CPU Undefined Vector Register
+6008	CPU_SWI_VEC	R/W	CPU Software Interrupt Vector Register
+600C	CPU_INSTR_ABORT	R/W	CPU Instruction Abort Register
+6010	CPU_DATA_ABORT	R/W	CPU Data Abort Register
+6014	Reserved		
+6018	CPU_IRQ0_VEC	R/W	CPU IRQ0 Vector Register
+601C	CPU_IRQ1_VEC	R/W	CPU IRQ1 Vector Register
+6020	CPU_IRQ2_VEC	R/W	CPU IRQ2 Vector Register

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

Interrupt Controller

2.11 Introduction

The interrupt controller is the central location for interrupt handling. It accepts the interrupts generated by the internal blocks and generates the processor interrupts. It enables and disables each interrupt individually or globally. A 2-level interrupt priority selection can be implemented.

2.12 Features

- Enabling/disabling of individual interrupts
- Global enable/disable
- 2-level interrupt priority selection

2.13 Block Diagram

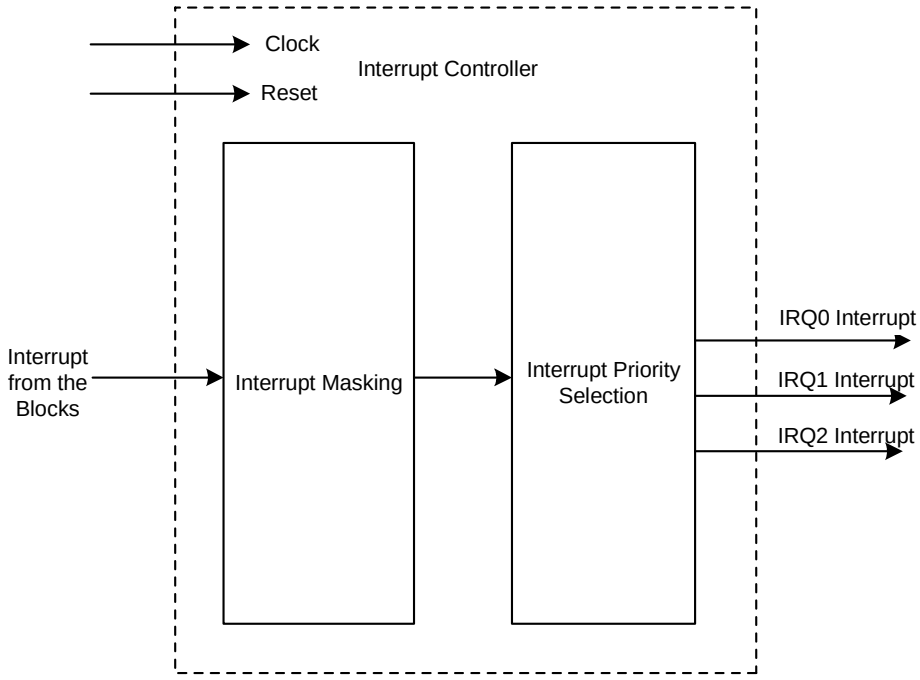


Figure 2-10 **Interrupt controller block diagram**

2.14 Functional Description

The interrupt controller allows each interrupt to be asserted as three interrupt levels, IRQ0, IRQ1 or IRQ2. 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.

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:

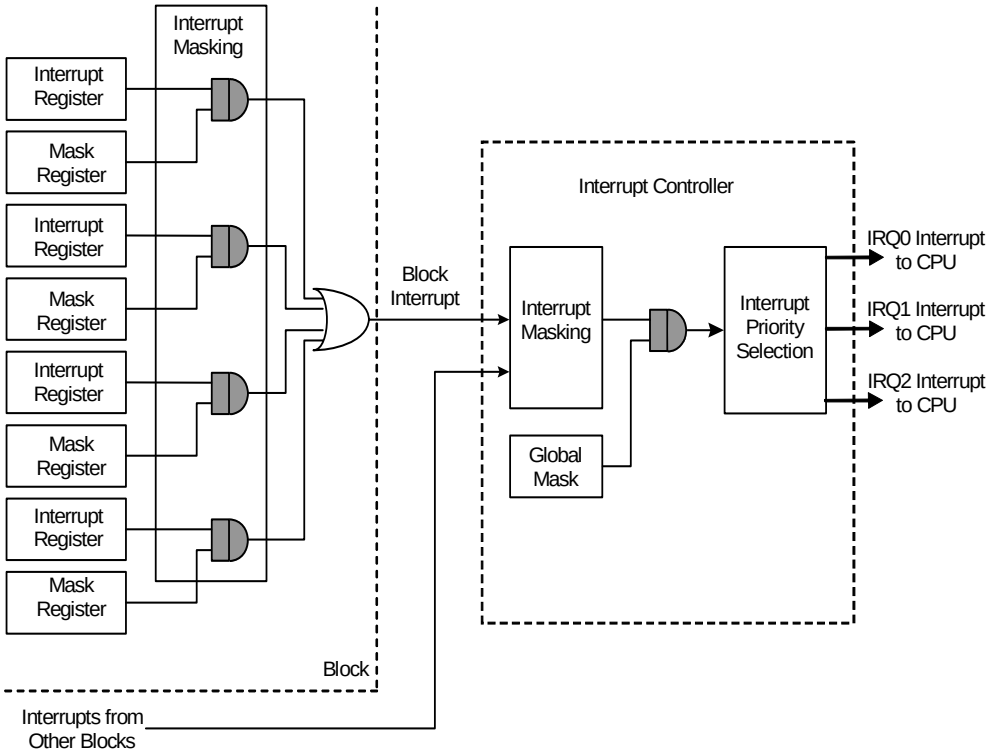


Figure 2-11 **Cascaded interrupt structure**

Interrupt Priority

Each interrupt may be assigned a priority of IRQ0, IRQ1 or IRQ2, 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.

2.14.1 Control (mask) Blocks

The control (masking) blocks allow the 63 (possibly edge-detected) interrupt lines to be masked, and routed to three CPU interrupt lines. They also contain registers to generate software interrupts.

2.14.2 Register Map - Control Block

Table 2-13 **Control block registers**

Address[1]					
IRQ0	**IRQ1**	**IRQ2**	**Register Name[2]**	**R/W/A[3]**	**Description**
+E000	+E100	+E300	CPU_XXX_STATUS_LO	R/A	CPU IRQ0/1/2 Status Low Register
+E004	+E104	+E304	CPU_XXX_RAWSTAT_LO	R/A	CPU IRQ0/1/2 Raw Status Low Register
+E008	+E108	+E308	CPU_XXX_ENABLESET_LO	R/W/A	CPU IRQ0/1/2 Enable Set Low Register
+E00C	+E10C	+E30C	CPU_XXX_ENABLECLR_LO	R/W	CPU IRQ0/1/2 Enable Clear Low Register
+E010	+E110	+E310	CPU_XXX_SOFTSET	R/W	CPU IRQ0/1/2 Soft Set Register
+E014	+E114	+E314	CPU_XXX_SOFTCLR	R/W	CPU IRQ0/1/2 Soft Clear Register
+E018	+E118	+E318	CPU_XXX_STATUS_HI	R/W/A	CPU IRQ0/1/2 Enable Set High Register
+E01C	+E11C	+E31C	CPU_XXX_RAWSTAT_HI	R/W	CPU IRQ0/1/2 Enable Clear High Register
+E020	+E120	+E320	CPU_XXX_ENABLESET_HI	R/W	CPU IRQ0/1/2 Soft Set High Register
+E02	+E124	+E324	CPU_XXX_ENABLECLR_HI	R/W	CPU IRQ0/1/2 Soft Clear High Register

1. Address refers to G-Bus byte address relative to the host CPU block base.
2. XXX is IRQ0/IRQ1/IRQ2 depending on the address.
3. Read/Write/Auto update

2.14.3 Processor Timer Interrupt

The CPU has six 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.

2.14.4 Edge Detector

The CPU interrupt controller block receives up to 40 hardware interrupt sources. An edge detector receives these 40 lines, and if needed, an edge detection is performed on selected lines. The output of the edge detector consists of 40 ‘latched’ interrupt lines, driven to three identical control (masking) blocks. Each control block can independently mask each of its 40 inputs, plus one software controlled interrupt.

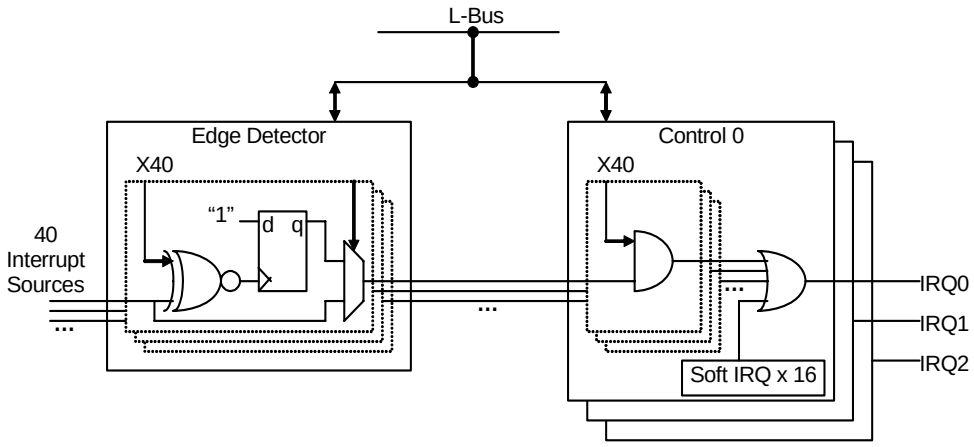


Figure 2-12 **Edge detector**

The 40 interrupts are assigned as shown in the following table:

Table 2-14 **Interrupt sources**

Bit	Source
0	Used for software interrupts
1	UART 0
2	UART 1
3	Unused
4	Unused
5	Timer 0

Table 2-14 **Interrupt sources (Continued)**

Bit	Source
6	Timer 1
7	DVD loader
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	IDE DMA
18	IDE
19	Front Panel Interface (FIP)
20	PCI local bus fault
21	Infrared receiver
22	I^2C
23	Graphics accelerator
24	VSYNC 0 (Composite analog output)
25	VSYNC 1 (Component analog output)
26	VSYNC 2 (Main analog output)
27	VSYNC 3 (Digital output)
28	VSYNC 4 (Gfxin-vsync: start of v-blanking)
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)
32	Smart card
33	HDMI
34	HDMI I^2C
35	V-Bus channel W0

Table 2-14 **Interrupt sources (Continued)**

Bit	Source
36	V-Bus channel W3
37	Ethernet PHY
38	Ethernet MAC
39	USB 1.1 Embedded Host
40	USB 2.0 Embedded Host

For each of the 40 interrupt lines, two bits called rise and fall select the operating mode according to the following table:

Table 2-15 **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

In order to set/clear the operating mode of some lines without changing the others, a read-modify-write operation can be performed on CPU_CONFIG_RISE_LO/HI and CPU_CONFIG_FALL_LO/HI registers. To make this operation simpler, the following 4 register pairs should be used: CPU_CONFIG_RISE_SET_LO/HI, CPU_CONFIG_RISE_CLR_LO/HI, CPU_CONFIG_FALL_SET_LO/HI and CPU_CONFIG_FALL_CLR_LO/HI. When writing to these 4 register pairs, the bits set to1 imply a modification of the operating mode of the corresponding interrupt, the bits set to 0 have no effect.

Reading the CPU_EDGE_STATUS_LO/HI register returns the individual interrupt lines values at the output of the edge detector.

Reading the CPU_EDGE_RAWSTAT_LO/HI register returns the individual interrupt lines values at the input of the edge detector. Note that for interrupts lines that are in “level sensitive active high” mode, these two registers will return the same value.

Each bit written to 1 in the CPU_EDGE_RAWSTAT_LO/HI register will clear the corresponding interrupt capture register; bits written to 0 have no effect. In the case of an edge sensitive interrupt, this write shall be done before exiting the interrupt subroutine in order to avoid automatic re-entering.

For debug purposes, each bit written to 1 in the CPU_EDGE_STATUS_LO/HI register will set the corresponding capture register. Bits written to 0 have no effects.

2.14.4.6 Register Map

Table 2-16 **Register map - Edge detector**

Address[1]	Register Name	R/W/A[2]	Description
+E200	CPU_EDGE_STATUS_LO	R/W	CPU Edge Detector Status Low Register
+E204	CPU_EDGE_RAWSTAT_LO	R/W/A	CPU Edge Detector Raw Status Low Register
+E208	CPU_EDGE_CONFIG_RISE_LO	R/W	CPU Edge Detector Configuration Rise Low Register
+E20C	CPU_EDGE_CONFIG_FALL_LO	R/W	CPU Edge Detector Configuration Fall Low Register
+E210	CPU_EDGE_CONFIG_RISE_SET_LO	R/W	CPU Edge Detector Configuration Rise Set Low Register
+E214	CPU_EDGE_CONFIG_RISE_CLR_LO	R/W	CPU Edge Detector Configuration Rise Clear Low Register
+E218	CPU_EDGE_CONFIG_FALL_SET_LO	R/W	CPU Edge Detector Configuration Fall Set Low Register
+E21C	CPU_EDGE_CONFIG_FALL_CLR_LO	R/W	CPU Edge Detector Configuration Fall Clear Low Register
+E220	CPU_EDGE_STATUS_HI	R/W	CPU Edge Detector Status High Register
+E224	CPU_EDGE_RAWSTAT_HI	R/W/A	CPU Edge Detector Raw Status High Register
+E228	CPU_EDGE_CONFIG_RISE_HI	R/W	CPU Edge Detector Configuration Rise High Register
+E22C	CPU_EDGE_CONFIG_FALL_HI	R/W	CPU Edge Detector Configuration Fall High Register
+E230	CPU_EDGE_CONFIG_RISE_SET_HI	R/W	CPU Edge Detector Configuration Rise Set High Register
+E234	CPU_EDGE_CONFIG_RISE_CLR_HI	R/W	CPU Edge Detector Configuration Rise Clear High Register
+E238	CPU_EDGE_CONFIG_FALL_SET_HI	R/W	CPU Edge Detector Configuration Fall Set High Register
+E23C	CPU_EDGE_CONFIG_FALL_CLR_HI	R/W	CPU Edge Detector Configuration Fall Clear High Register

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

Timers

2.15 Introduction

The SMP8634 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 prescaler is supported). When the counter reaches zero, an interrupt is generated and the initial count value is automatically reloaded.

2.16 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

2.17 Block Diagram

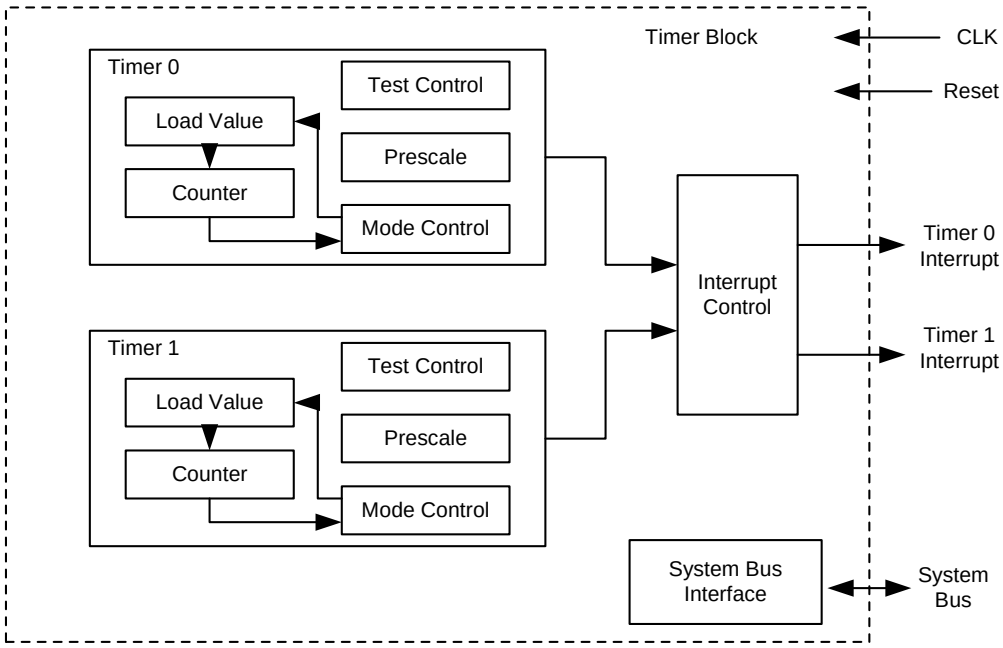


Figure 2-13 **Timer block diagram (system overview)**

2.18 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).

2.18.1 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.

2.18.1.1 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.

2.18.1.2 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.

2.18.1.3 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.

2.18.1.4 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 2-17 **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.

2.19 Register Map

2.19.1 Timer Registers

Table 2-18 **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

2.20 Introduction

The SMP8634 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.

2.21 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

2.22 Block Diagram

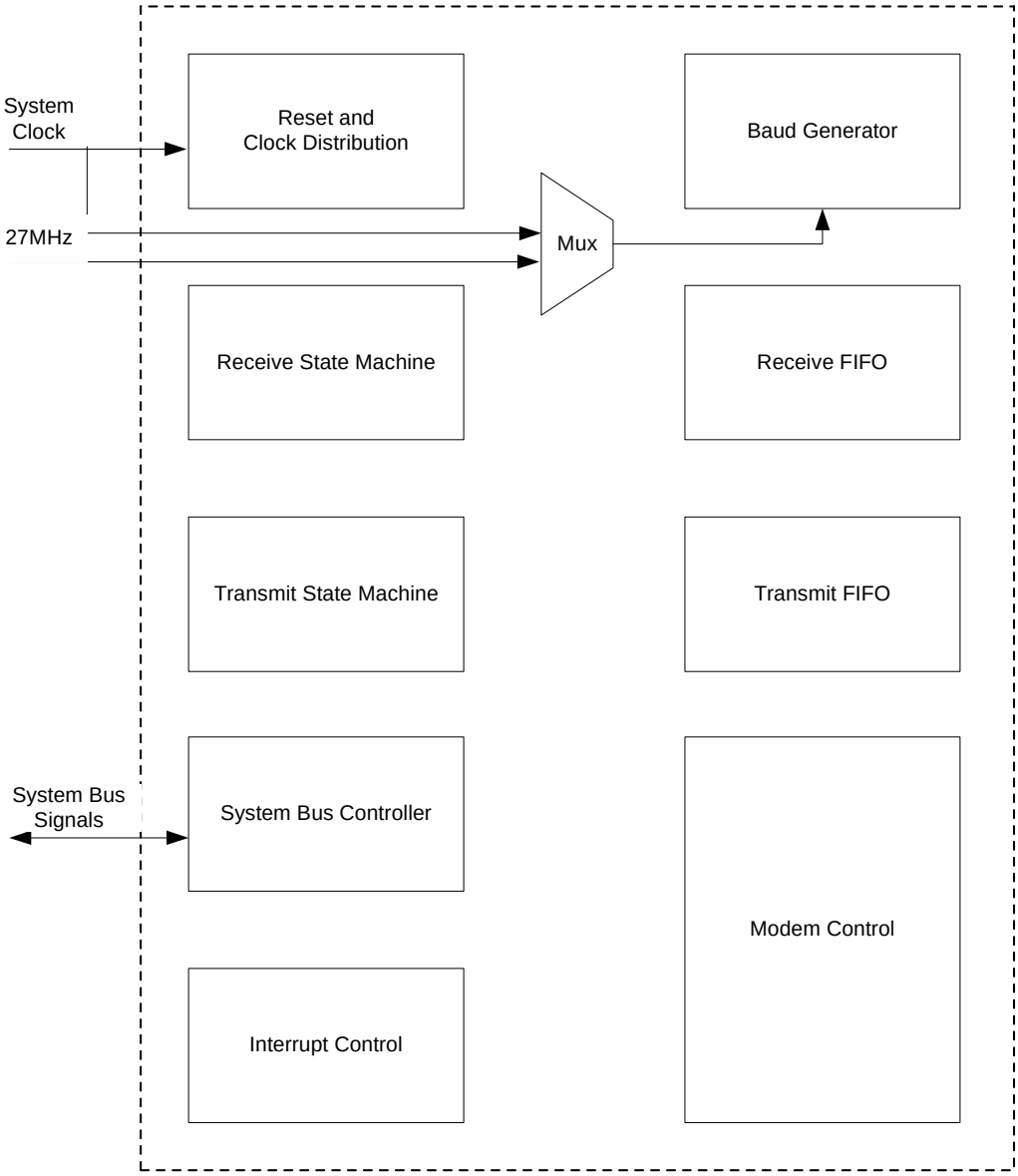


Figure 2-14 **UART block diagram**

2.23 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 SMP8634 are almost compatible with the industry standard 16550. The differences between the SMP8634 UARTs and the 16550 are listed below:

- Registers are 4-byte apart in the SMP8634 UART versus the 1-byte in the 16550
- The UART in the SMP8634 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 Multiplier 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.

2.23.1 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.

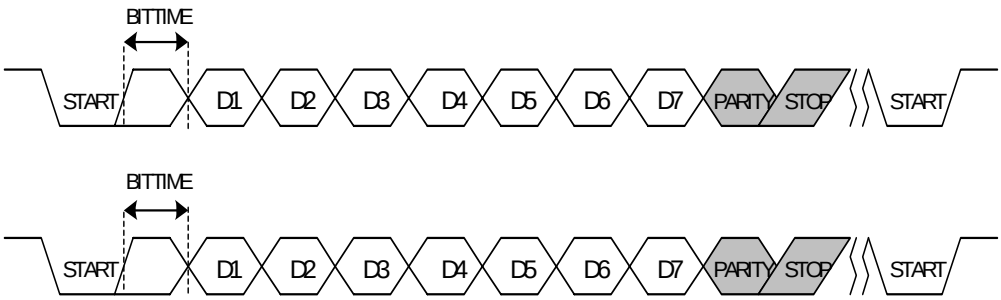
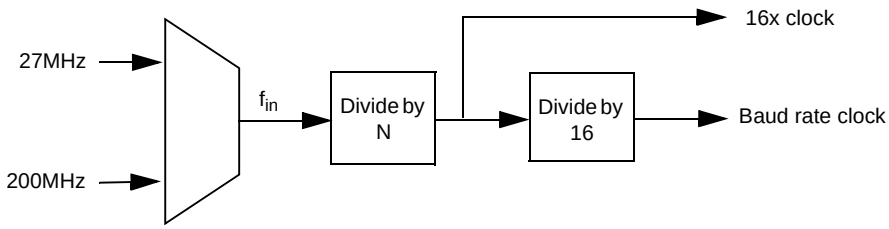


Figure 2-15 **UART data formats**



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 changing.

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.

2.23.2 UART Shared Function

2.23.2.1 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_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.

2.23.2.2 Smart Card Function

In the SMP8634 three extra GPIO pins, SCARD_CTL[2:0] are provided to control an external interface circuit associated with the smart card reader. These 3 extra pins are controlled by bits [18:16] of the CPU_UART_GPIO_DIR register (0=input, 1=output) and bits [18:16] of the CPU_UART_GPIO_DATA register for UART # 0.

2.24 Register Map

2.24.1 UART Registers

Table 2-19 **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

Table 2-19 **UART registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+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.

2.25 Pin Description

2.25.1 UART Pins

Table 2-20 **UART pin descriptions**

Pin Name	Ball ID	Direction	Description
UART0_CTS	B14	B	UART 0 clear to send. Flow control signal.
UART0_DCD	B12	B	UART 0 data carrier detect. Data set status signal.
UART0_DSR	C14	B	UART 0 data set ready. Data set status signal.
UART0_DTR	C13	B	UART 0 data terminal ready. Data terminal status signal.
UART0_RTS	B13	B	UART 0 request to send. Flow control signal.
UART0_RX	A14	B	UART 0 receive data input
UART0_TX	A13	B	UART 0 transmit data output
UART1_CTS	C12	B	UART 1 clear to send. Flow control signal.
UART1_DCD	C11	B	UART 1 data carrier detect. Data set status signal.
UART1_DSR	B11	B	UART 1 data set ready. Data set status signal.
UART1_DTR	C10	B	UART 1 data terminal ready. Data terminal status signal.
UART1_RTS	B10	B	UART 1 request to send. Flow control signal.
UART1_RX	A12	B	UART 1 receive data input
UART1_TX	A11	B	UART 1 transmit data output

2.26 Electrical Characteristics

2.26.1 UART DC characteristics

Table 2-21 **UART 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	2.0		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.

2.26.2 UART AC characteristics

Table 2-22 **UART AC characteristics**

Parameter	Minimum	Typical	Maximum	Units
T_{BIT}	8.68			µs
Baud Rate (1/ T_{BIT})			115.2	Kbps

2.27 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

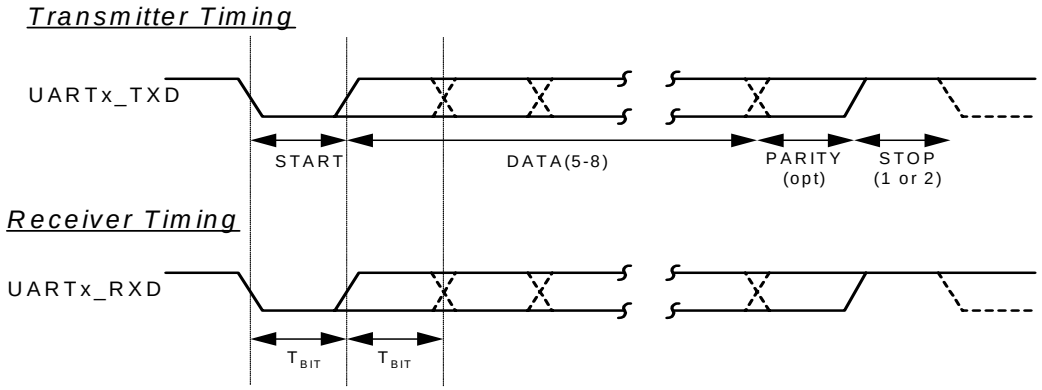


Figure 2-16 **UART timing diagram**

3 Security Subsystem

3.1 Block Diagram of Security Subsystem

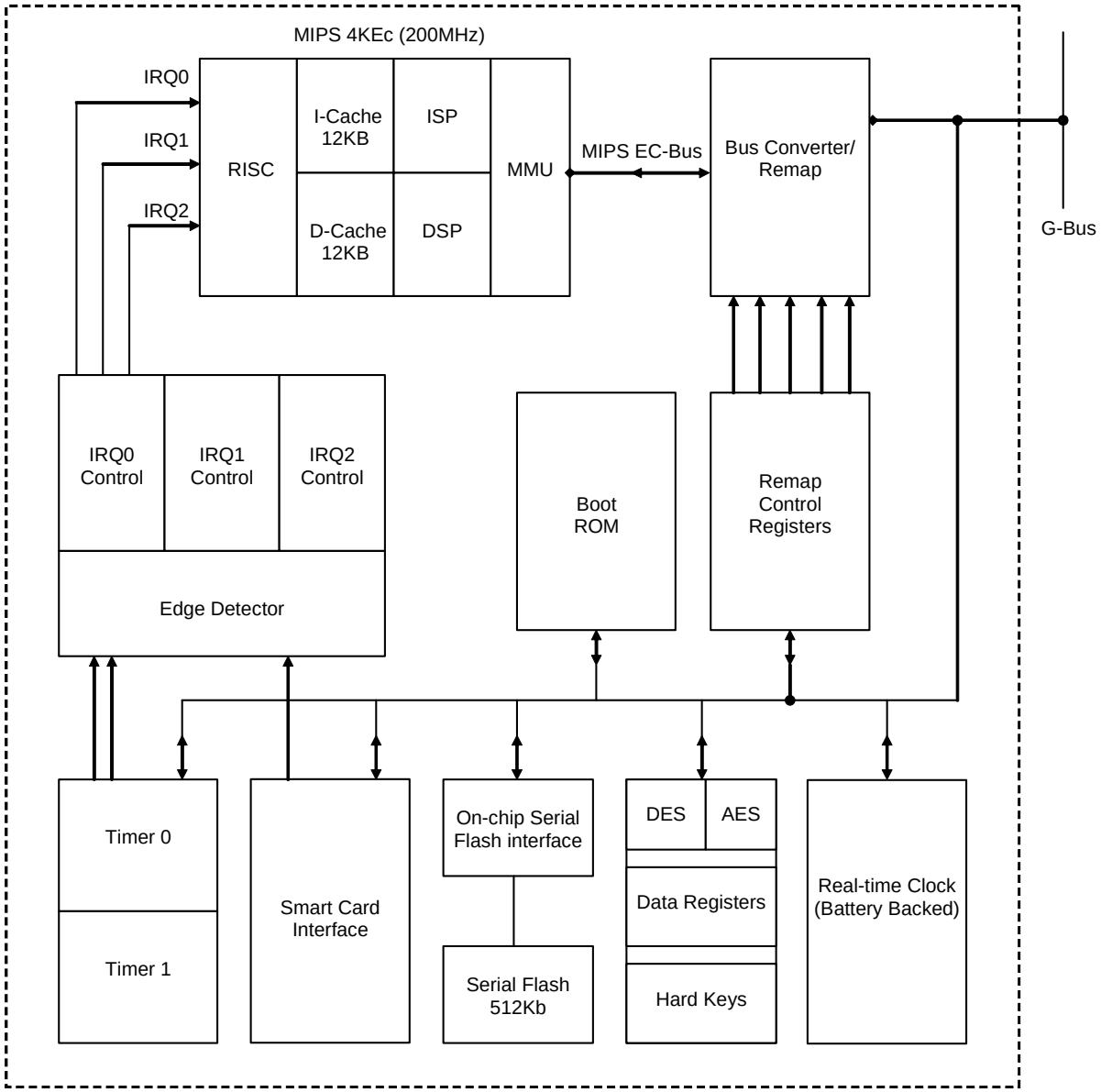


Figure 3-1 **Security subsystem block diagram**

3.2 Introduction

The SMP8634 includes a variety of features designed to ensure that the DRM/CA software runs with robust security:

- Security CPU (XPU)
- Secure boot loader - chip boots from the embedded flash only (trusted boot block)
- Embedded serial flash
- Parallel flash data can be encrypted and digitally signed
- DRAM data can be encrypted
- Secure PKI - on-chip key storage, keys are not externally accessible.
- Real-time clock with battery back-up. Detection of a disabled clock can be done.
- Unique ID - programmable at production (inside embedded flash)
- As a PCI device, access to internal resources (registers, DRAM etc.) can be restricted by the internal security software

The security subsystem block of the SMP8634 includes a dedicated 200MHz MIPS 4KEc CPU (~306 conforming DMIPS). The application software for the XPU is loaded from external flash memory. In addition to the XPU processor itself, the security subsystem block contains the following additional resources:

- Boot ROM: Upon hardware reset, the XPU processor executes from the boot ROM.
- RSA public keys: Another ROM (can be modified with on-metal change).
- Serial flash interface: Controls the embedded serial flash located in the chip package.
- Cipher block: For acceleration of AES and (3)DES functions.
- Smart card Interface: Supports ISO/IEC7816 standard asynchronous protocols.
- Real-time clock: Battery-backed real-time clock with tamper detection.
- Interrupt controller
- 2 programmable timers

The XPU accesses the G-Bus as a master. A bus converter allows the XPU to be a G-Bus master, and access all the G-Bus mapped local resources through the L-Bus. Other components of the security subsystem can be accessed by either the XPU or another G-Bus master. The bus converter in the security subsystem performs a bus protocol adaptation and a configurable address translation (remap). The security subsystem connects to the rest of the chip via the G-Bus.

3.3 Features

- Dedicated 200MHz MIPS 4KEc CPU to securely execute the DRM and the conditional access software
- 24KB program and 16KB data memory
- Encrypted DRAM and flash memory, secure boot loader, secure PKI, encrypted I/O interfaces.
- Supports up to 4 DRM solutions simultaneously
- Supports a wide variety of DRM/CA solutions
  - Available from Sigma Designs Inc.,
    * WM DRM-ND and WM DRM-PD
    * DTCP/IP
    * HDCP for DVI and HDMI
    * CPPM for DVD-Audio playback
    * CPRM for DVD-R and DVD-RW playback
    * CSS for DVD-Video playback
    * AACS for BD-ROM, BD-R, BD-RE v2 and HD DVD-Video playback
    * BD+ for BD-ROM playback
    * BD-CPS for BD-RE v1 playback
    * Verance VCMS/AV Detector for HD-DVD, BD
    * Verance Audio Watermark Detector for DVD-Audio
    * VCPS for DVD+R/+RW
    * SSL
    * Macrovision copy protection for analog video outputs
    * Widevine Cypher® Virtual SmartCard™
  - Third party solutions (requires the purchase of SMP8634 security SDK)
    * NDS VGS
    * Irdeto Access
    * SecureMedia
    * ARIB, ATSC, DVB-CI, DVB-CSA and OpenCable* conditional access

Note: The OEM is responsible for licensing and purchasing any production keys and programming them (in the encrypted form) into the on-chip serial flash during OEM product production. Sigma Designs provides the tools needed to do this.

Note: * Requires the use of a SCM Microsystem CIMax SP2 chip.

Note: DRM software and/or keys are not included with SDKs and Development Kits. Once your license(s) are verified, software and/or keys for required features will be available via our developer website.

Central Processor Unit (XPU)

3.4 Description

The SMP8634 uses a separate 200MHz MIPS 4KEc CPU (~306 conforming DMIPS) to securely execute conditional access and the DRM/CA software. The application software for this XPU resides in flash memory (encrypted and digitally signed) to facilitate upgrades.

The encryption and decryption keys are stored (in the encrypted form) in the on-chip serial flash. Additional keys may be encrypted and loaded into the serial flash during the OEM product production. To ensure security, keys are not available to the software running on the host CPU. The encryption and decryption keys are stored (in encrypted form) in the on-chip serial flash.

The XPU core is a high-performance, low-power, 32-bit MIPS RISC core. The 4KEc core is a 32-bit privileged resource architecture. A Memory Management Unit (MMU) contains 4-entry instruction and data Translation Lookaside Buffers (ITLB/DTLB) and a 32 dual-entry joint TLB (JTLB) with variable page sizes. The 4KEc core includes a Multiply/Divide Unit (MDU) that implements single cycle MAC instructions, which enable the DSP algorithms to be performed efficiently. It allows 32-bit x 16-bit MAC instructions to be issued every cycle, while a 32-bit x 32-bit MAC instruction can be issued every 2 cycles.

It contains 3-way set associative 12KB instruction and 12KB data caches. The load and fetch cache misses only block until the critical word becomes available. The pipeline resumes execution while the remaining words are being written to the cache. Both caches are virtually indexed and physically tagged to allow them to be accessed in the same clock that the address is translated.

For more details on the XPU processor refer to the section on CPU in the chapter *Host CPU Block*.

3.4.1 XPU MIPS Requirements

The following table lists the MIPS requirements for various functions performed by the XPU.

Table 3-1 **MIPS requirements for various functions performed by the XPU**

Function	Typical MIPS needed
XPU baseline	TBD
- WM DRM-ND	TBD
- WM DRM-PD	TBD
- DTCP/IP	TBD
- HDCP for DVI and HDMI	TBD
- CPPM for DVD-Audio playback	TBD
- CPRM for DVD-R and DVD-RW playback	TBD
- CSS for DVD-Video playback	TBD
- AACS for BD-ROM playback	TBD
- AACS for HD DVD-Video playback	TBD
- BD+ for BD-ROM playback	TBD
- BD-CPS for BD-RE playback	TBD
- Verance audio watermark detection	50
- VCPS for DVD+R/+RW	TBD

Hardware Cipher Block

3.5 Description

The cipher block for the XPU has eight 32-bit data registers, corresponding to 4 DES blocks. There is a second set of 8 registers, that is a copy of the first set with byte swap (for endianess adjustment). For example, writing 0x33221100 to DATA0 or writing 0x00112233 to DATA0R are identical (same for reading).

The encryption/decryption is performed by writing N = 1, 2, 3 or 4 data blocks into the data registers starting from DATA0. The DES cipher is started for N blocks. After a wait for the DES idle flag, N blocks from the data registers can be read.

3.5.1 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 3-2 **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.

3.5.2 AES Encryption/Decryption Unit

The AES encryption/decryption unit can either encrypt or decrypt a 128, 192 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, 192 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 3-3 **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)

Real-time Clock (RTC)

3.6 Description

The real-time clock (RTC) provides a battery powered time counter and 64-bit storage that remains when the main power is removed. This enables applications requiring time-limited access to content, such as 24-hour viewing of downloaded content, etc.

The RTC contains 3 internal registers. The time register value increments every second. The write only data register values, key0 and key1, can be used as AES keys.

The three registers are duplicated for main power and battery power. When a register is written to, both the copies take the same value. Once the main power is removed, the main power registers lose their value, while the battery powered registers keep the value. Upon power up, a command is issue to reload the main power registers from the battery power registers.

One additional register, valid, can be set to 0 or 1 and read back. This flag reads 0 when the battery power is removed, and stays at zero after the battery power is restored, until a ‘set valid’ command is given to the RTC. When the valid flag is 0, time, key0 and key1 take the value 0xFFFFFFFF. Therefore, a valid flag must be set before the RTC can be used for any function.

The commands are given to the RTC by optionally writing a value into the register XPU_RTC_DATA, and then writing a command code into the register XPU_RTC_CMD.

3.7 Register Map

3.7.1 Real-time Clock Registers

Table 3-4 **Real-time clock registers**

Address[1]	Register Name	R/W/A[2]	Description
CB00	XPU_RTC_DATA	W	XPU Real-time Clock Data Register
CB04	XPU_RTC_CMDSTAT	R/W	XPU Real-time Clock CMD Status Register
CB08	XPU_RTC_TIME	R	XPU Real-time Clock Time Register

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

3.8 Pin Description

3.8.1 Real-time Clock Pins

Table 3-5 **Real-time clock pin descriptions**

Pin Name	Ball ID	Direction	Description
RTC_CLK_IN	AN24	I	Reserved. Make no connection to this pin (internal pull-down).
RTC_CLK_OUT	AP23	O	RTC clock output
RTC_RING	AM24	O	RTC ring output
RTC_TEST	AM25	I	RTC test input
RTC_VDD_BAT_3V3	AL24	I	RTC crystal oscillator power (3.3V)
RTC_VDD_BAT_3V3	AL25	I	RTC battery power (3.3V)
RTC_VSS	AK24	I	RTC crystal oscillator ground
RTC_XTAL_DISC	AN25	I	Reserved. Make no connection to this pin (internal pull-down).
RTC_XTAL_IN	AP24	I	RTC 32KHz crystal oscillator input
RTC_XTAL_OUT	AP25	O	RTC 32KHz crystal oscillator output

Boot ROM

The boot ROM contains the initial boot code, and the RSA keys used to verify the secondary boot code. During system start-up, the device begins executing from the boot ROM. It then continues booting either from the serial flash or an external parallel flash, after authenticating the digital signature of the code.

Serial Flash

The SMP8634 contains an embedded 512Kb serial flash memory. The flash memory contains the system boot code (in conjunction with the boot ROM) and any necessary encryption/decryption keys for AACS, CPPM, CPRM, DTCP/IIP, etc. The serial flash image is encrypted and digitally signed using a 2048-bit RSA key. The boot ROM code verifies the signature, before executing. Only code and keys signed by Sigma Designs may reside in the serial flash.

Smart Card Interface

3.9 Introduction

The security subsystem in the SMP8634 contains a smart card interface. The smart card interface supports ISO/IEC7816 standard asynchronous protocols. It may be used for applications such as DVB-CSA conditional access and proprietary IPTV DRM solutions. 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.

3.10 Features

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

3.11 Block Diagram

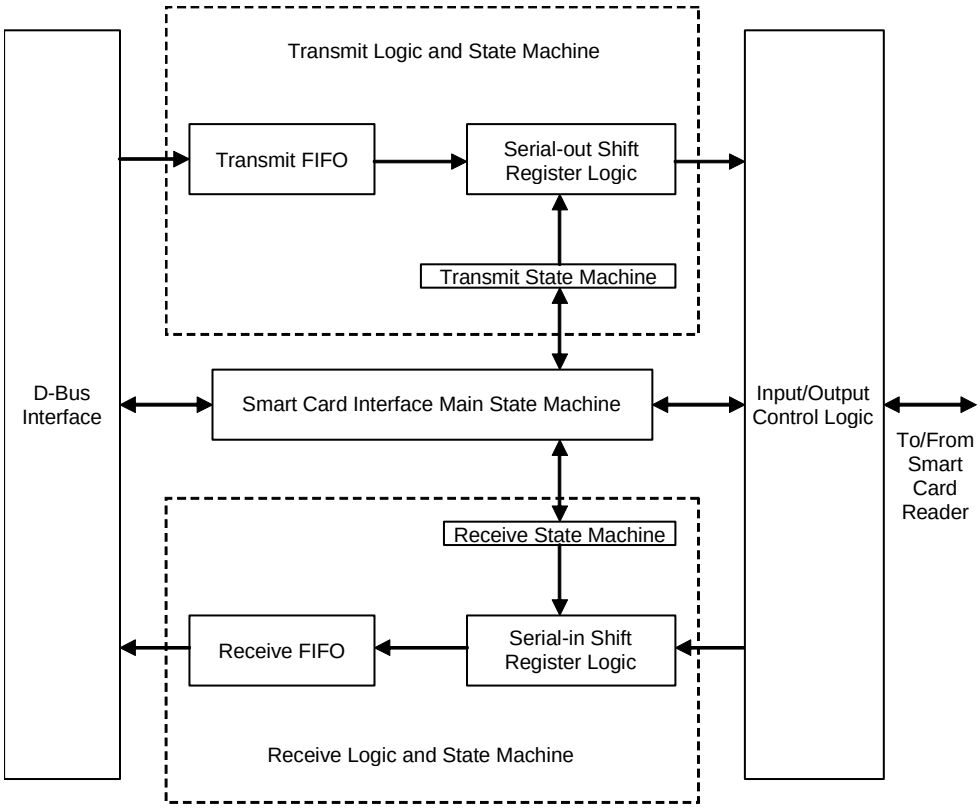


Figure 3-2 **Smart card interface block diagram**

3.12 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.

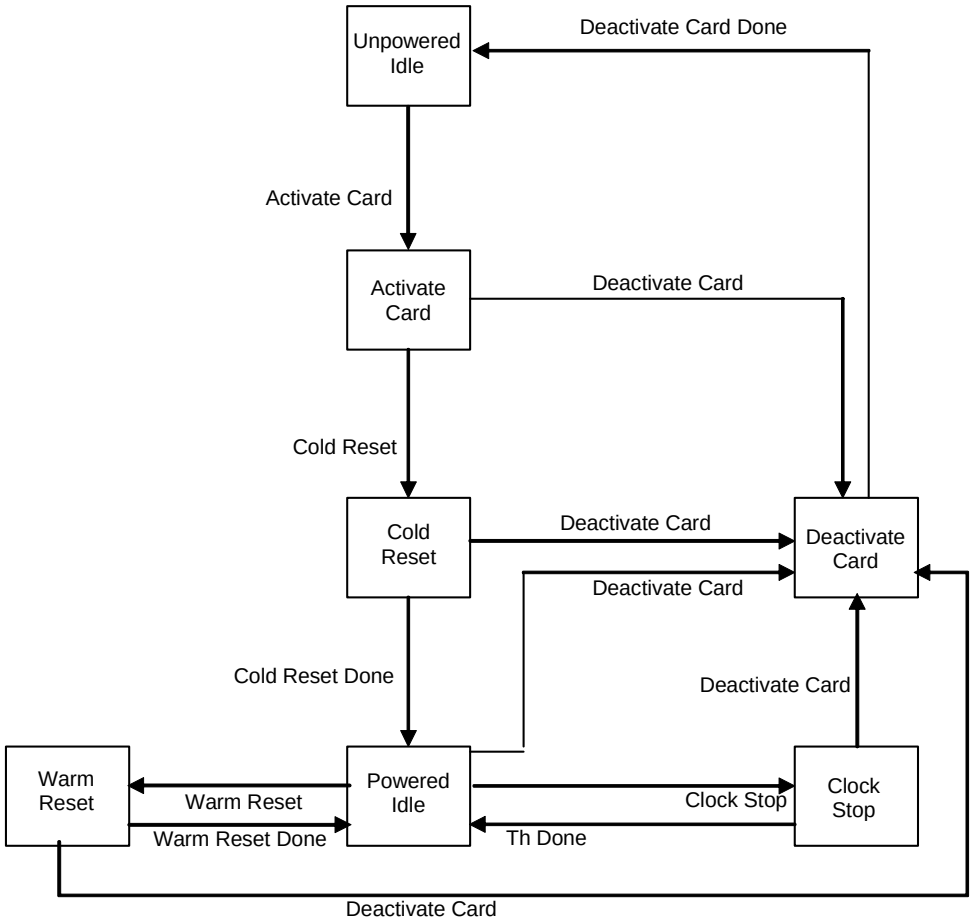


Figure 3-3 **Smart card interface state machine**

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.

3.13 Register Map

3.13.1 Smart Card Interface Registers

Table 3-6 **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 security subsystem register base.
2. Read/Write/Auto update.

3.14 Pin Description

3.14.1 Smart Card Interface Pins

Table 3-7 **Smart card interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SCARD_CLK	A16	O	Smart card clock
SCARD_CTL0	C15	B	Smart card control 1
SCARD_CTL1	D16	B	Smart card control 2
SCARD_CTL2	E16	B	Smart card control 3
SCARD_FC#	B16	O	Smart card function code
SCARD_IO	A15	B	Smart card IO
SCARD_RST	B15	O	Smart card reset

XPU Timers

3.15 Introduction

The SMP8634 has a timer block implemented in the security subsystem. 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.

3.16 Features

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

3.17 Block Diagram

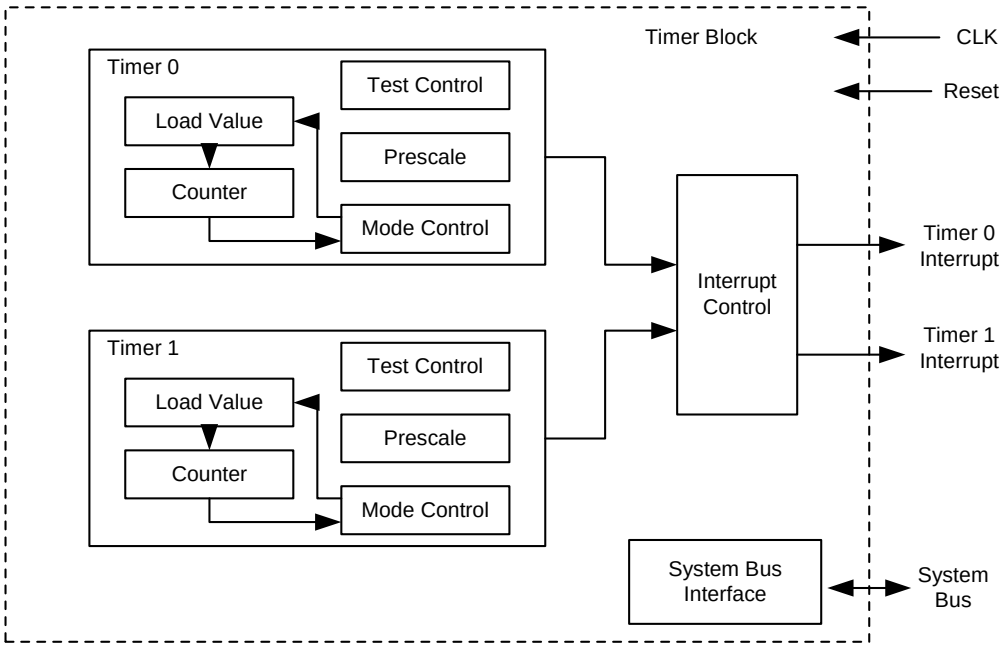


Figure 3-4 **Timer block diagram (system overview)**

3.18 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).

3.18.1 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.

3.18.1.1 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.

3.18.1.2 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.

3.18.1.3 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.

3.18.1.4 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 3-8 **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.

XPU Interrupt Controller

3.19 Introduction

The interrupt controller is the central location for interrupt handling. It accepts the interrupts generated by the internal blocks and generates the processor interrupts. It enables and disables each interrupt individually or globally. A 2-level interrupt priority selection can be implemented.

3.20 Features

- Enabling/disabling of individual interrupts
- Global enable/disable
- 2-level interrupt priority selection

3.21 Block Diagram

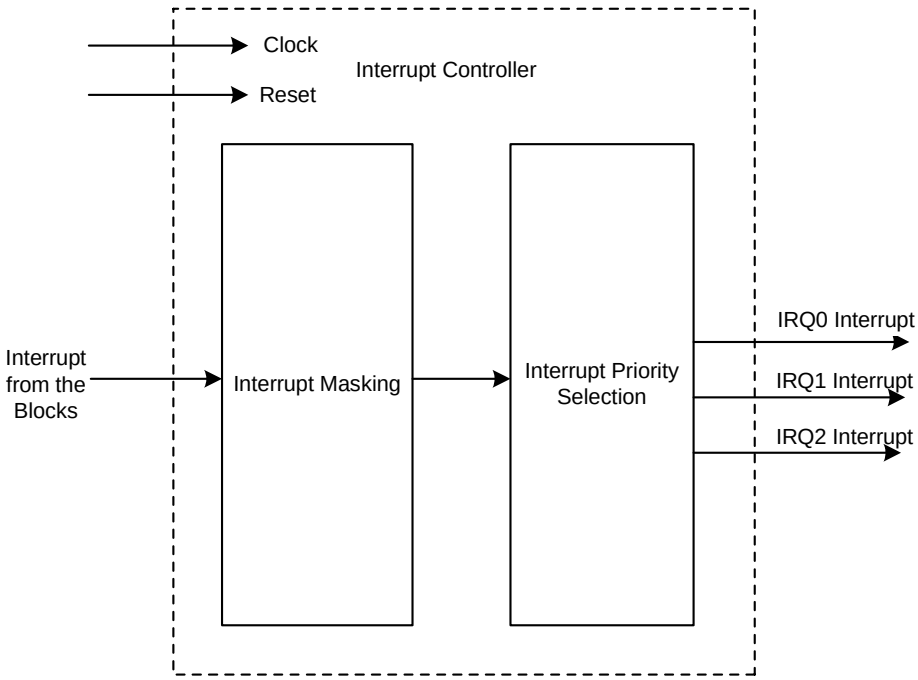


Figure 3-5 **Interrupt controller block diagram**

3.22 Functional Description

The interrupt controller allows each interrupt to be asserted as three interrupt levels, IRQ0, IRQ1 or IRQ2. 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:

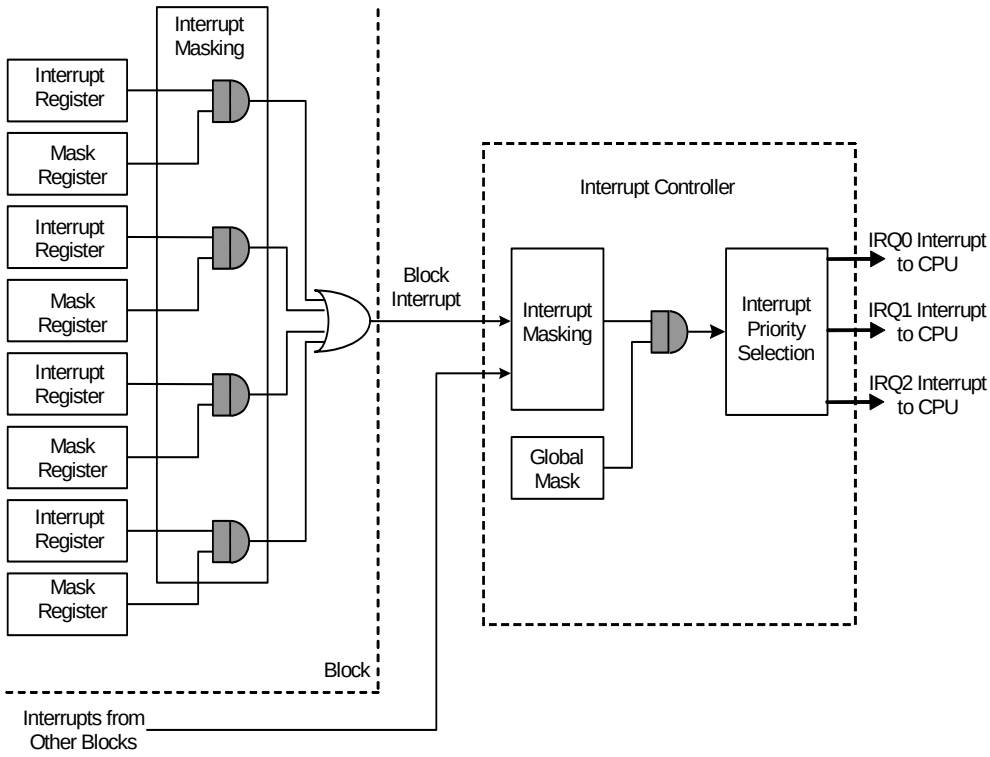


Figure 3-6 **Cascaded interrupt structure**

Interrupt Priority

Each interrupt may be assigned a priority of IRQ0, IRQ1 or IRQ2, 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.

3.22.1 Control (mask) Blocks

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

3.22.2 Processor Timer Interrupt

The XPU has six 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.

3.22.3 Edge Detector

The XPU 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 three identical control (masking) blocks. Each control block can independently mask each of its 31 inputs, plus one software controlled interrupt.

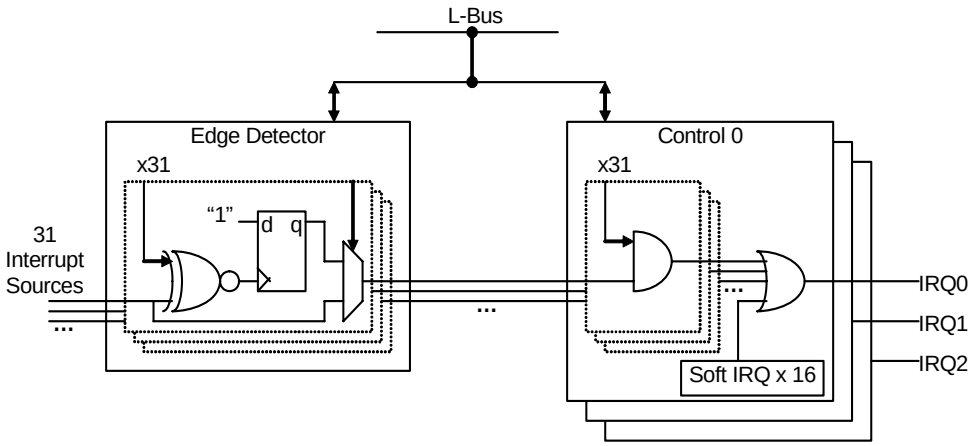


Figure 3-7 **Edge detector**

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

Table 3-9 **Interrupt sources**

Bit	Source
0	Unused
1	UART 0
2	UART 1
3	Unused
4	Unused
5	XPU Timer 0
6	XPU Timer 1
7	DVD loader
8	XPU 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
22	I^2C
23	Graphics accelerator
24	VSYNC 0 (Component 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 3-9 **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 3-10 **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.

Bus accessibility

3.23 Description

In the SMP8634 some restrictions are placed on the bus accesses. For example, the XPU will have access to certain areas of the DRAM that are inaccessible by the host CPU.

The following table shows the four top level busses that interconnect the top level blocks of the chip and their master/slave connections:

Table 3-11 **Top level bus connections**

Bus	Master Group[1]	Master	Slave
G-Bus	G0	GWD/GRD task during simulation	Security subsystem
	G0	Security subsystem	Video decoder
	G1	Video decoder	Audio block
	G1	Audio block	Transport demultiplexer
	G1	Transport demultiplexer	Video decoder
	G2	Host CPU block	DRAM controller
	G3	Host interface	System block
			Host CPU block
			Host interface
M-Bus	D1	Video decoder	DRAM controller
	D1	Audio block	
	D1	Transport demultiplexer	
	D3	Host interface	
V-Bus	D2	Video output	DRAM controller
Direct load port	D1	Video decoder	DRAM controller

1. G0…G3 are groups of G-Bus masters. D1…D3 are groups of DMA masters (i.e. M-Bus, V-Bus and the direct load port) and are therefore relevant to only one slave, the DRAM controller. Typically, a hierarchy G0>G1>G2>G3 is implemented, meaning that G0 has the maximum number of accesses and G3 the least.

The bus accessibility is handled at the slave level, and is similar (with minor differences) to all the slaves described above.

Every time a slave is accessed it will, first check which of the masters is initiating the transaction. The initiating master ID is converted into a group code. A check on address is being accessed is placed. Whether the master group is allowed to access the requested address is verified. Depending on the outcome the request is either serviced or ignored. Any number masters from the same group can have accesses to the same address.

The information about group accessibilities and address accessibilities is stored in the accessibility configuration registers, which are accessible to group G0 (security sub-system) only.

3.23.1 Memory Area Accessibility

The memory area (256MB) is configurable on a 4KB page basis using the memory access-registers. These registers can be used in the increasing order, depending on how many areas are required (up to five). This is particularly important when less than 128MB of memory is implemented on the DRAM controller. For example, if only 64MB of area is present, then an access to any address above 64MB aliases to the address minus 64MB, defeating the protection mechanism.

Reset Condition

Upon hardware reset, only the XPU (G0), and no other resource on the SMP8634 will have full access to the DRAM. A DRAM controller reset does not affect the access control registers.

Register Area Accessibility

The access registers are (64KB) configurable on a dword basis using the registers DC_REG_ACCESS_X. Unlike the DRAM itself, the DRAM controller registers are not accessible by the M-Bus, V-Bus or the direct load port. The group information contains four 2-bit fields (corresponding to 3 master groups). Within each 2-bit field,

- Bit 0 if set, allows write access to the corresponding group
- Bit 1 if set, allows read access to the corresponding group

The accessibility control for the transport demux, the video decoder and audio block is identical to the one from the DRAM controller, except that, in the DRAM controller, total memory space is 256MB, configurable with a granularity of 4KB. Transport demux memory space is 128KB, configurable with a granularity of 4-byte and only two configuration registers are there for the register space accessibility.

The accessibility control for the security subsystem is identical to the one from the DRAM controller, except that, in the DRAM controller, total memory space is 256MB, configurable with a granularity of 4KB.

The XPU memory space is 4MB, configurable with a granularity of 256-byte; there are 6 registers for memory space accessibility, and three registers for register space accessibility and power-up values are discussed below.

Power-up values when the PROTECTED_BUSSES directive is present:

XPU_MEM_ACCESS0 = ‘h03_00_0000

XPU_MEM_ACCESS1 = ‘h00_00_0000 // Overridden by XPU_MEM_ACCESS0 -> unused

XPU_MEM_ACCESS2 = ‘h00_00_0000 // Overridden by XPU_MEM_ACCESS0 -> unused

XPU_MEM_ACCESS3 = ‘h00_00_0000 // Overridden by XPU_MEM_ACCESS0 -> unused

XPU_REG_ACCESS0 = ‘h57_00_2000

XPU_REG_ACCESS1 = ‘h03_00_7FE0

XPU_REG_ACCESS2 = ‘hFF_00_0000

With these values,

1. The XPU ROM is writable by all the resources, but readable only by the XPU. This is done for quickturn emulation where the ROM is replaced by a RAM to download the contents before starting the XPU. On the real chip, writability by any other resource will have no effect since a ROM will be implemented.
2. The XPU bridge registers are read-write accessible for all the resources, allowing the PCI to start the XPU once the ROM is loaded. One of the first instructions in the XPU ROM should make this registers inaccessible to prevent the CPU or PCI from resetting the XPU. On the real chip, the XPU starts automatically.
3. Other resources within the XPU register space are read/write accessible to the XPU only.

4 DRAM Controllers

4.1 Block Diagram of DRAM Controller

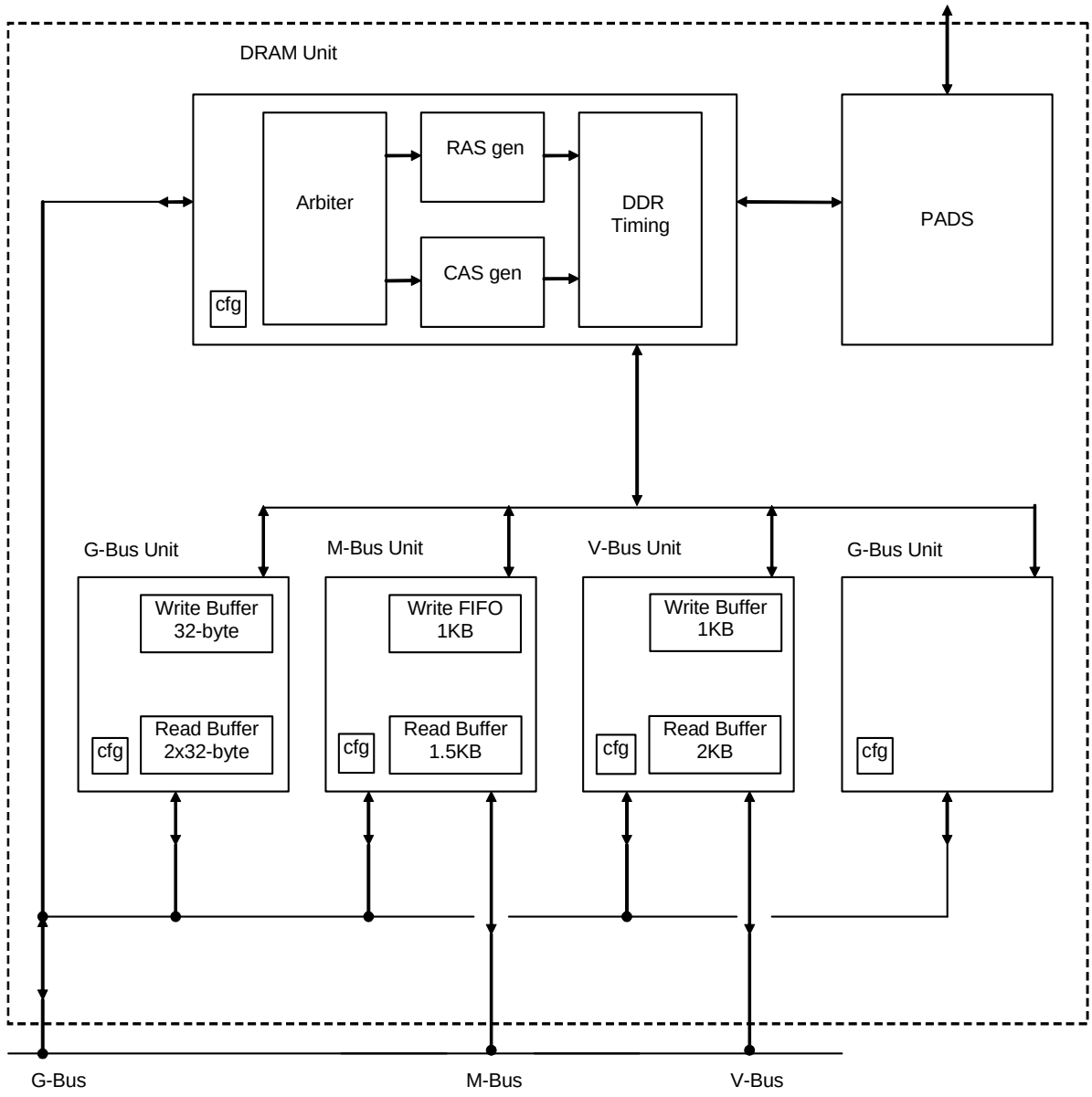


Figure 4-1 **DRAM controller block diagram**

4.2 Introduction

The SMP8634 contains two identical double-data rate synchronous DRAM (DDR-SDRAM) controllers operating at up to 200MHz. Each DRAM controller supports up to 256MB of DDR-1 memory (DDR-2 is not supported). Because the DDR technology transfers data on both edges of the clock, the maximum effective burst data bandwidth of each controller is 1.6GB/sec. No more than 80% of the memory bandwidth should be used (~1.3GB/sec for each 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 SMP8634. 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 controllers supports memory devices which conform to JEDEC standard JESD79. The electrical interface to the external memories supports JEDEC SSTL_2 compliant signaling.

4.3 Features

- Two 32-bit DRAM controllers
- Double-data rate synchronous DRAM (DDR-400)
- 32 or 64-bit DRAM interface
  - 32-bit: MPEG-2 HDTV applications
  - 64-bit: MPEG-4.10 (H.264) or SMPTE 421M (VC-1) HDTV applications
- Effective burst data bandwidth of the DRAM interface is up to 3.2GB/sec
- Interfaces up to 512MB of external DDR SDRAM

4.4 Functional Description

The two SMP8634 DRAM controllers (DRAM controller 0 and DRAM controller 1) can interface to up to 512MB of external DDR SDRAM, using two 32-bit wide data busses. The two DRAM controllers interface with the different modules of the SMP8634 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.

Each of the DRAM controllers 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 units, 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 units are designed to maximize utilization of the DRAMs, while simultaneously controlling read and write access latency.

The SMP8634 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.

4.4.1 Memory Configurations

The two 32-bit DRAM controller interfaces can connect to either 64, 128 or 512MB of DDR-1 SDRAM using two 32-bit wide data busses. Due to timing tolerances, capacitive load considerations and PCB trace length considerations, only the recommended DRAM vendors/products should be used with the SMP8634.

Table 4-1 **Supported memory configurations**

Total Memory (MB)	Bus Width (DRAM0/DRAM1)	Memory Chip Configuration	Rows	Columns
64	32/0	Two 16Mx16	8192	512
128	32/0	Two 32Mx16	8192	1024
256	32/0	Two 64Mx16	8192	1024
256	32/32	Four 32Mx16	8192	1024
384	32/32	DRAM0: Four 64Mx8 DRAM1: Two 32Mx16	8192 8192	2048 1024
512	32/32	Four 64Mx16	8192	1024
512	32/32	Eight 64Mx8	8192	2048

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 host 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.

4.4.2 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.

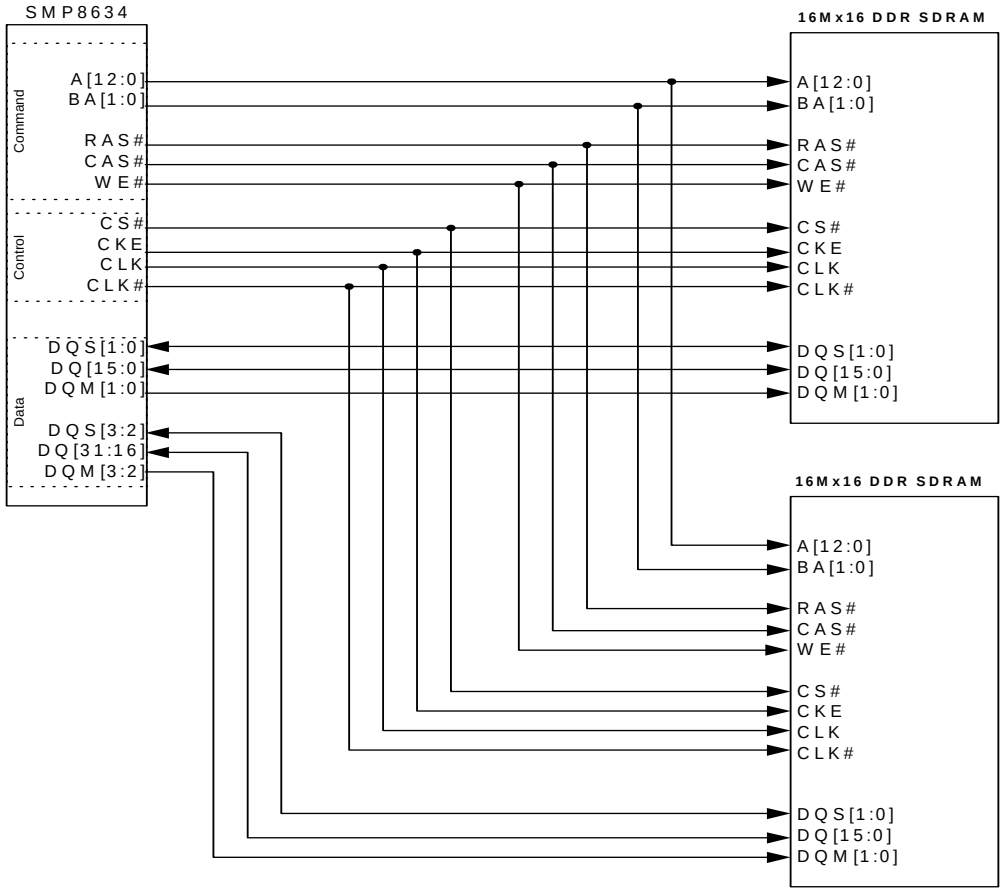


Figure 4-2 **64MB memory; 32-bit bus width**

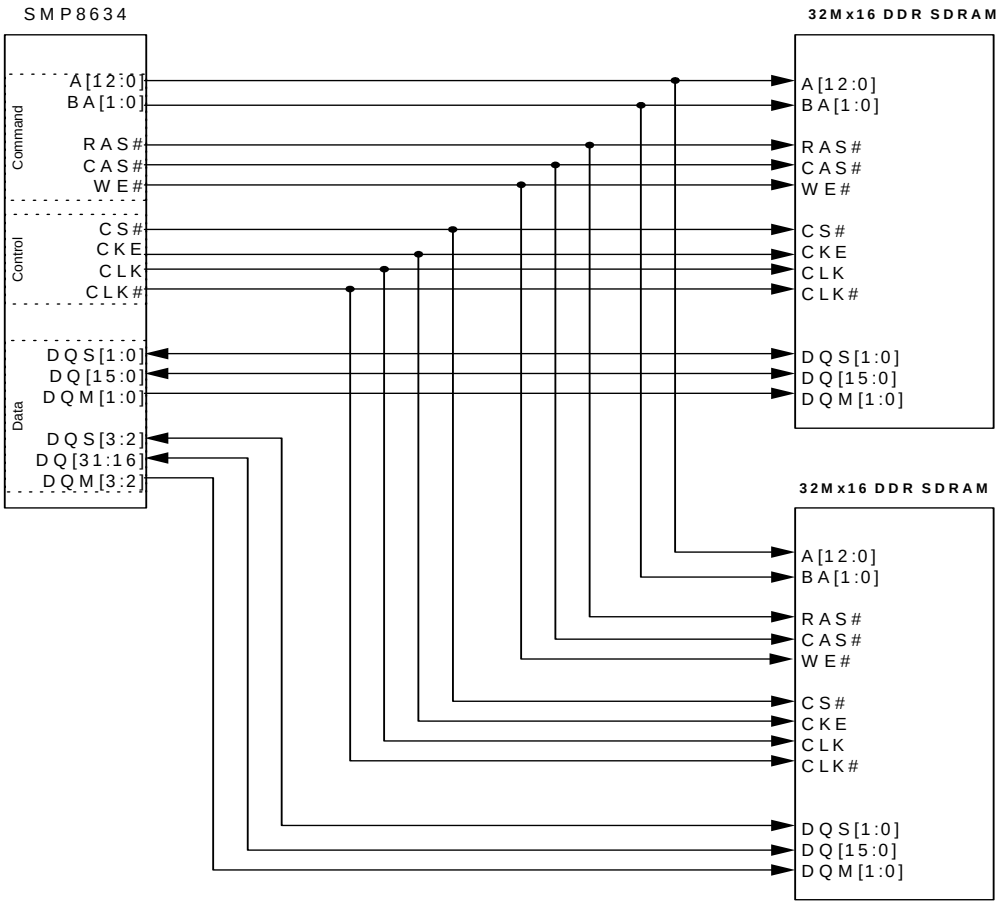


Figure 4-3 **128MB memory; 32-bit bus width**

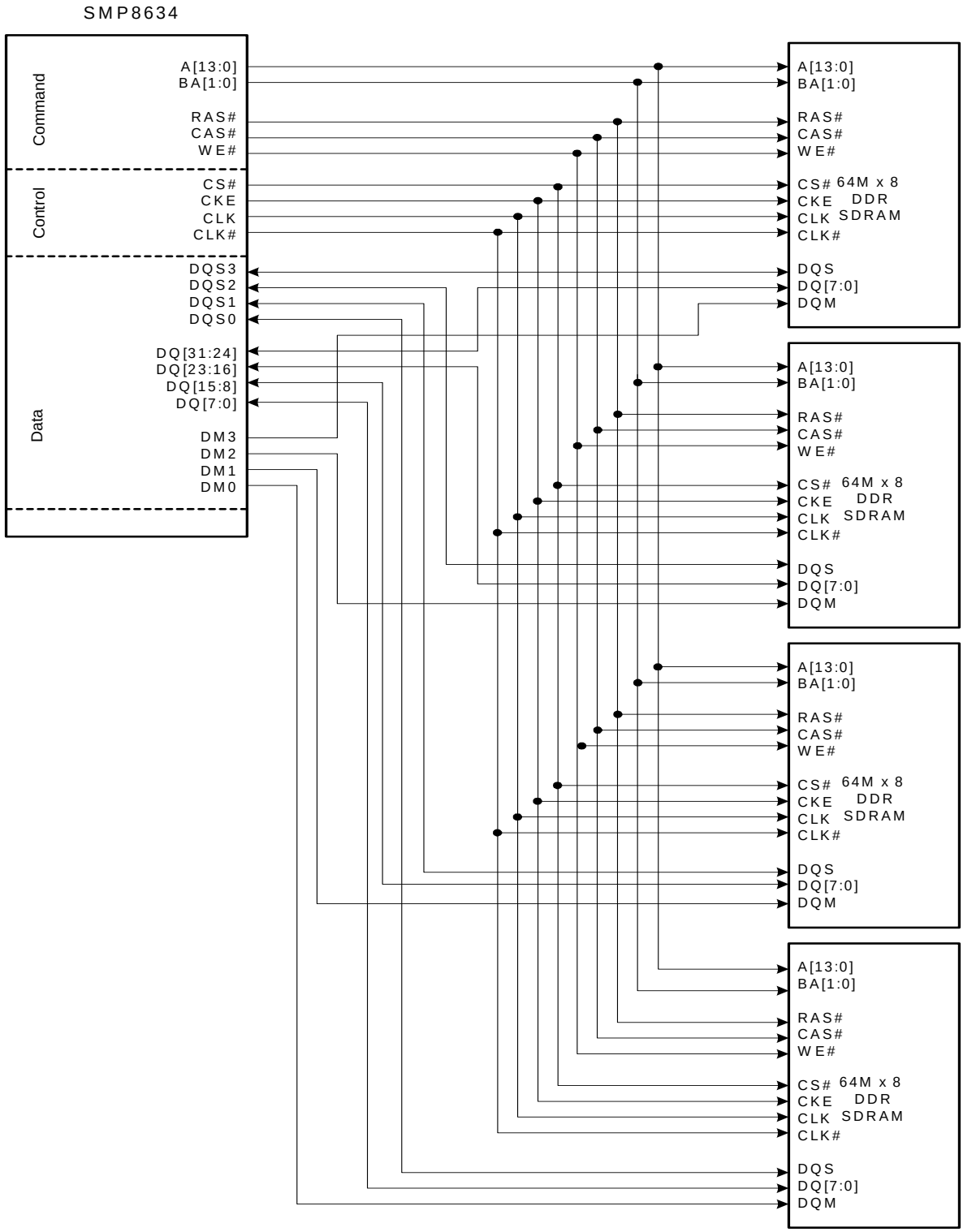


Figure 4-4 **256MB memory; 32-bit bus width**

4.4.3 Memory Bandwidth Considerations

For complex applications, the available memory bandwidth may limit what can be done with SMP8634. In these cases the total available memory bandwidth (assuming DDR-400 DRAM) is as follows:

- For each 32-bit interface, maximum memory bandwidth is 1.6GB/s (1.3GB/s for DDR-333)
- Should not exceed 80% of maximum, resulting in 1.3GB/s available memory bandwidth for each 32-bit interface (1GB/s for DDR-333)

Table 4-2 **Memory bandwidth requirements for video**

Function	Source Format (frames per second)	Memory bandwidth needed
MPEG-4.10 (H.264) video decode and display	1920 x 1080 x 30	1.1GB/s (0.95GB/s decode only)
	1280 x 720 x 60	1.0GB/s
	720 x 576 x 50	0.45GB/s
	720 x 576 x 25	0.22GB/s
	720 x 480 x 60	0.45GB/s
	720 x 480 x 30	0.22GB/s
SMPTE 421M (VC-1) video decode and display	1920 x 1080 x 30	0.97GB/s
	1280 x 720 x 60	0.86GB/s
	720 x 576 x 50	0.31GB/s
	720 x 576 x 25	0.15GB/s
	720 x 480 x 60	0.31GB/s
	720 x 480 x 30	0.15GB/s
MPEG-2 video decode and display	1920 x 1080 x 30	0.41GB/s
	1280 x 720 x 60	0.37GB/s
	720 x 576 x 50	0.14GB/s
	720 x 576 x 25	0.07GB/s
	720 x 480 x 60	0.14GB/s
	720 x 480 x 30	0.07GB/s
Displaying 32-bit graphics	1920 x 1080 x 60	498MB/s
	1920 x 1080 x 50	415MB/s
	1920 x 1080 x 30	249MB/s
	1920 x 1080 x 25	207MB/s
	1280 x 720 x 60	221MB/s
	1280 x 720 x 50	184MB/s
	720 x 576 x 50	83MB/s
	720 x 576 x 25	41MB/s
	720 x 480 x 60	83MB/s
	720 x 480 x 30	41MB/s

Table 4-2 **Memory bandwidth requirements for video (Continued)**

Function	Source Format (frames per second)	Memory bandwidth needed
Displaying 24-bit graphics (24-bit data is written and read as 32-bit data)	1920 x 1080 x 60	498MB/s
	1920 x 1080 x 50	415MB/s
	1920 x 1080 x 30	249MB/s
	1920 x 1080 x 25	207MB/s
	1280 x 720 x 60	221MB/s
	1280 x 720 x 50	184MB/s
	720 x 576 x 50	83MB/s
	720 x 576 x 25	41MB/s
	720 x 480 x 60	83MB/s
	720 x 480 x 30	41MB/s
Displaying 16-bit graphics	1920 x 1080 x 60	249MB/s
	1920 x 1080 x 50	207MB/s
	1920 x 1080 x 30	124MB/s
	1920 x 1080 x 25	104MB/s
	1280 x 720 x 60	111MB/s
	1280 x 720 x 50	92MB/s
	720 x 576 x 50	41MB/s
	720 x 576 x 25	21MB/s
	720 x 480 x 60	41MB/s
	720 x 480 x 30	21MB/s
Displaying 8-bit graphics	1920 x 1080 x 60	124MB/s
	1920 x 1080 x 50	104MB/s
	1920 x 1080 x 30	62MB/s
	1920 x 1080 x 25	52MB/s
	1280 x 720 x 60	55MB/s
	1280 x 720 x 50	46MB/s
	720 x 576 x 50	21MB/s
	720 x 576 x 25	10MB/s
	720 x 480 x 60	21MB/s
	720 x 480 x 30	10MB/s
Additional bandwidth for Type 1 deinterlacing of main video	1920 x 1080i x 30	93MB/s
	1920 x 1080i x 25	78MB/s
	720 x 576i x 25	16MB/s
	720 x 480i x 30	16MB/s
Additional bandwidth for Type 2 (motion adaptive) deinterlacing of main video	1920 x 1080i x 30	187MB/s
	1920 x 1080i x 25	156MB/s
	720 x 576i x 25	31MB/s
	720 x 480i x 30	31MB/s
Additional bandwidth for simultaneous HD+SD output		
- On-the-fly mode		0MB/s
- Buffered mode		42MB/s
Video decoder RISC	-	3MB/s (assuming 24Mbps bitstream)

Table 4-3 **Memory bandwidth requirements for audio**

Function	Typical memory bandwidth needed
Decoding, output formatting and routing	
- Dolby Digital 5.1	29MB/s
- Dolby Digital Plus 2.0	TBD
- Dolby Digital Plus 5.1	TBD
- Dolby Digital Plus 7.1	TBD
- Dolby TrueHD 5.1	TBD
- Dolby TrueHD 7.1	TBD
- DTS 5.1	24MB/s
- DTS-HD 7.1	TBD
- MPEG-1 Layers I, II 2.0	75MB/s
- MPEG -1 Layer III (mp3) 2.0	65MB/s
- MPEG-4 AAC-LC 2.0	TBD
- MPEG-4 AAC-LC 5.1	49MB/s
- MPEG-4 HE-AAC 5.1	41MB/s
- MPEG-4 BSAC 2.0	TBD
- LPCM 7.1	70MB/s
- WMA9 2.0	53MB/s
- WMA Pro 5.1	32MB/s
- ATRAC3 2.0	1MB/s
Encoding	
- Dolby Digital 5.1	TBD
- DTS 5.1	TBD
Other	
- Watermark detection	1MB/s
- Post processing	TBD

For example, displaying a HD MPEG-4.10 (H.264) video + SD MPEG-4.10 (H.264) + 32-bit HD graphics + 8-bit HD subtitle requires, 1.1 + 0.22 + 0.25 + 0.06 = 1.6GB/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 (25) frames or 60 (50) 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 4-4 **Typical memory bandwidth values**

Function	Typical memory bandwidth needed
XPU baseline	1MB/s
- WM DRM-ND and WM DRM-PD	6MB/s (16Mbps bitstream max.)
- DTCP/IP	3MB/s
- HDCP for DVI and HDMI	1MB/s
- CPPM for DVD-Audio playback	1MB/s
- CPRM for DVD-R and DVD-RW playback	1MB/s
- CSS for DVD-Video playback	1MB/s
- AACS for BD-ROM playback	7MB/s
- AACS for HD DVD-Video playback	7MB/s
- BD+ for BD-ROM playback	7MB/s
- BD-CPS for BD-RE playback	7MB/s
- Verance audio watermark detection	1MB/s
- VCPS for DVD+R/+RW	1MB/s
BD-ROM	
- BD-J graphics (read + write)	200MB/s single-buffered 300MB/s double-buffered
- Java bytecode interpreter code and data cache misses	50MB/s estimated 150MB/s estimated for bytecode accelerated Java
- Presentation graphics (PG) rendering	16MB/s
- Interactive graphics (IG) rendering	8MB/s
- PG decoded object buffer	32MB/s (read) 16MB/s (write)
- Dripfeed decode	1MB/s
HD-DVD cursor, graphics, subpicture and secondary video rendering	280Mpixels/s
Transport demultiplexer	10MB/s
- AACS	10MB/s
- AACS+	10MB/s
- BD+	10MB/s

4.4.3.1 BD-ROM Player Considerations

The following tables illustrate a basic configuration for BD-ROM players that balances the memory bandwidth requirements between the two DRAM interfaces. DRAM bank 1 uses 64MB of DDR-400 DRAM; DRAM bank 0 uses 256MB of DDR-400 DRAM. Linux and DirectFB must use DRAM bank 0.

With this configuration, when outputting 1080i30 video (Table 4-4), more than 64MB may be used in DRAM bank 1. This enables having additional memory storage with an available memory bandwidth of ~200MB/s.

When outputting 1080p60 video (Table 4-5), there is not sufficient memory bandwidth on DRAM bank 1 to access unused or additional memory.

Table 4-5 **Typical BD-ROM player memory bandwidth values - 1080i30 source, 1080i30 output**

	DRAM bank 1 (MB/sec)	DRAM bank 0 (MB/sec)	Note
H.264 decode (1920x1080i30) and display	1100		Based on 8634 data sheet
Display blended BD-J graphics and OSD (32-bit, 1920x1080x30)		249	Based on 8634 data sheet
Display PG (8-bit, 1920x1080x30)		62	Based on 8634 data sheet
Display background (16-bit, 1920x1080x30)		124 (0 if PIP)	Based on 8634 data sheet
Write PG		16	BR-ROM spec
PG decoded object buffer (read+write)		48	BD-ROM spec
BD-J graphics (read+write, includes FAA)		200	300MB/s for double-buffered
Blending BD-J graphics and OSD data		50	Typical. 100 MB/s for double buffered OSD data. OSD pixel format, object number/sizes and frame rate dependent.
Dripfeed (background)		1	
Deinterlacing (1920x1080x30)	NA		Motion adaptive. Based on 8634 data sheet
Simultaneous HD + SD output	42		Based on 8634 datasheet

Table 4-5 **Typical BD-ROM player memory bandwidth values - 1080i30 source, 1080i30 output (Continued)**

	DRAM bank 1 (MB/sec)	DRAM bank 0 (MB/sec)	Note
XPU (AACS, BD+)		14	Based on 8634 datasheet
Audio		29	Typical. Based on 8634 datasheet
Video RISC		4	Bitstream from DRAM to video decoder
Transport demux (1ch, 40Mbps)		30	Based on 8634 datasheet
Driver sw cache misses		10	Estimated
Application software cache misses		100	Estimated
Java bytecode interpreter code and data cache misses		50	Estimated. For bytecode-accelerated Java, 150MB/s.
Total - without PIP	**1142**	**846**	
H.264 video decode (720x480i30) and display		220	Based on 8634 datasheet
Audio		4	Typical. Based on 8634 datasheet.
Transport demux (1ch, 20Mbps)		15	Based on 8634 datasheet
No background display with PIP		-124	Based on 8634 datasheet
Total - with PIP	**1142**	**961**	Luma keying for PIP on rev. C and later

Table 4-6 **Typical BD-ROM player memory bandwidth values - 1080i30 source, 1080p60 output**

	DRAM bank 1 (MB/sec)	DRAM bank 0 (MB/sec)	Note
H.264 decode (1920x1080i30) and display	1100		Based on 8634 data sheet
Display blended BD-J graphics and OSD (32-bit, 1920x1080x60)		498	Based on 8634 data sheet
Display PG (8-bit, 1920x1080x60)		124	Based on 8634 data sheet
Display background (16-bit, 1920x1080x60)		249 (0 if PIP)	Based on 8634 data sheet
Write PG		16	BR-ROM spec
PG decoded object buffer (read+write)		48	BD-ROM spec
BD-J graphics (read+write, includes FAA)		200	300MB/s for double-buffered
Blending BD-J graphics and OSD data		50	Typical. 100 MB/s for double buffered OSD data. OSD pixel format, object number/sizes and frame rate dependent.
Dripfeed (background)		1	
Deinterlacing (1920x1080x30)	187		Motion adaptive. Based on 8634 data sheet
Simultaneous HD + SD output	42		Based on 8634 datasheet
XPU (AACS, BD+)		14	Based on 8634 datasheet
Audio		29	Typical. Based on 8634 datasheet
Video RISC		4	Bitstream from DRAM to video decoder
Transport demux (1ch, 40Mbps)		30	Based on 8634 datasheet
Driver sw cache misses		10	Estimated
Application software cache misses		100	Estimated

Table 4-6 **Typical BD-ROM player memory bandwidth values - 1080i30 source, 1080p60 output (Continued)**

	DRAM bank 1 (MB/sec)	DRAM bank 0 (MB/sec)	Note
Java bytecode interpreter code and data cache misses		50	Estimated. For bytecode-accelerated Java, 150MB/s.
Total - without PIP	**1329**	**1282**	
H.264 video decode (720x480i30), deinterlace and display		236	Based on 8634 datasheet
Audio		4	Estimated (in + DRAM delay)
Transport demux (1ch, 20Mbps)		15	Based on 8634 datasheet
No background display with PIP		-249	Based on 8634 datasheet
No motion adaptive (Type 2) deinterlacing of main video with PIP due to sharing of multi-format scaler 1	-187		Based on 8634 datasheet
Type 1 deinterlacing of main video	93		Based on 8634 datasheet
Total - with PIP	**1235**	**1288**	Luma keying for PIP on rev. C and later

Table 4-7 **Typical BD-ROM player memory bandwidth values - 1080p24 source, 1080p60 output**

	DRAM bank 1 (MB/sec)	DRAM bank 0 (MB/sec)	Note
H.264 decode (1920x1080p24) and display	950		Based on 8634 data sheet
Display blended BD-J graphics and OSD (32-bit, 1920x1080x60)		498	Based on 8634 data sheet
Display PG (8-bit, 1920x1080x60)		124	Based on 8634 data sheet
Display background (16-bit, 1920x1080x60)		249 (0 if PIP)	Based on 8634 data sheet
Write PG		16	BR-ROM spec
PG decoded object buffer (read+write)		48	BD-ROM spec
BD-J graphics (read+write, includes FAA)		200	300MB/s for double-buffered
Blending BD-J graphics and OSD data		50	Typical. 100 MB/s for double buffered OSD data. OSD pixel format, object number/sizes and frame rate dependent.
Dripfeed (background)		1	
Deinterlacing (1920x1080x30)	NA		Motion adaptive. Based on 8634 data sheet
Simultaneous HD + SD output	42		Based on 8634 datasheet
XPU (AACS, BD+)		14	Based on 8634 datasheet
Audio		29	Typical. Based on 8634 datasheet.
Video RISC		4	Bitstream from DRAM to video decoder
Transport demux (1ch, 40Mbps)		30	Based on 8634 datasheet
Driver sw cache misses		10	Estimated
Application software cache misses		100	Estimated

Table 4-7 **Typical BD-ROM player memory bandwidth values - 1080p24 source, 1080p60 output (Continued)**

	DRAM bank 1 (MB/sec)	DRAM bank 0 (MB/sec)	Note
Java bytecode interpreter code and data cache misses		50	Estimated. For bytecode-accelerated Java, 150MB/s.
Total - without PIP	**992**	**1282**	
H.264 video decode (720x480p24) and display		220	Based on 8634 datasheet
Audio		4	Typical. Based on 8634 datasheet.
Transport demux (1ch, 20Mbps)		15	Based on 8634 datasheet
No background display with PIP		-249	Based on 8634 datasheet
Total - with PIP	**992**	**1272**	Luma keying for PIP on rev. C and later

4.4.4 Memory Size Requirements

The following tables list some of the major software/firmware components and the typical amount of DRAM they require.

Table 4-8 **Video Decoding DRAM requirements (MB, includes all needed buffers, no deinterlacing or type 1 deinterlacing)**

	MPEG-2	MPEG-4.2	MPEG-4.10 (H.264)	SMPTE 421M (VC-1)
352x240 352x288	1	1	3	2
352x480i 352x576i	2	2	5	3
720x480i 720x576i	4	4	7	4
1280x720p	7	8	13	8
1920x1080i	16	17	36	18

Table 4-9 **Video Decoding DRAM requirements (MB, includes all needed buffers, type 2 deinterlacing used)**

	MPEG-2	MPEG-4.2	MPEG-4.10 (H.264)	SMPTE 421M (VC-1)
352x240 352x288	2	2	3	2
352x480i 352x576i	3	3	5	3
720x480i 720x576i	4	5	8	5
1280x720p	9	9	14	10
1920x1080i	19	20	39	21

Table 4-10 **Typical Firmware DRAM requirements (MB)**

Firmware	Total DRAM Required	DRAM Bank 1	DRAM Bank 0
Video decoding (HD MPEG-4.10)	39	39	
Audio decoding	5		5
Linux (11) + MRUA/DCC-HD/DirectFB (11) + kernel mode drivers (8)	30		30
XPU	4		4
2D graphics accelerator	1		1
Simultaneous HD + SD output	2	2	
Total	**81**	**41**	**40**

4.4.4.2 BD-ROM Player Considerations

The following table illustrates a basic configuration for a BD-ROM player. It applies to the DRAM memory bandwidth calculations in Tables 4-4, 4-5 and 4-6. DRAM bank 1 uses 64MB of DDR-400 DRAM; DRAM bank 0 uses 256MB of DDR-400 DRAM. Linux and DirectFB must use DRAM bank 0.

Table 4-11 **Typical BD-ROM player DRAM requirements (MB)**

	Total DRAM Required	DRAM Bank 1	DRAM Bank 0
Drive buffer	12		12
Explicit bdav-std buffers (not including video buffers)	16 (42 if no BD-J)		16
Implied bdav-std buffers for seamless playback, etc.	10		10
Background plane	4		4
BD-J plane (includes off-screen drawing area, OSD)	8		8
IG plane (2x)	0 (4 if no BD-J)		0
PG plane (2x)	4		4
BD-J	72		72
Firmware	81	41	40
HDMV + BD-J middleware	35	-	35
Other software + OSD bitmaps	10	-	10
Total - without PIP	**252**	**41**	**211**
Physical total due to available DRAM sizes	**-**	**64**	**256**
Secondary video decode (SD MPEG-4.10)	7		7
Secondary audio decode	4		4
Explicit bdav-std buffers (not including video buffers)	4		4
Implied bdav-std buffers	0		0
Total - with PIP	**267**	**41**	**226**
Physical total due to available DRAM sizes	**-**	**64**	**256**

4.5 Pin Description

4.5.1 DRAM Controller Pins

Table 4-12 **DRAM controller pin descriptions**

Pin Name	Ball Id	Direction	Description
DRAM0_A0	V34	O	Memory address bit 0
DRAM0_A1	V33	O	Memory address bit 1
DRAM0_A10	U32	O	Memory address bit 10
DRAM0_A11	T30	O	Memory address bit 11
DRAM0_A12	R31	O	Memory address bit 12
DRAM0_A13	R33	O	Memory address bit 13. 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.
DRAM0_A2	V32	O	Memory address bit 2
DRAM0_A3	W31	O	Memory address bit 3
DRAM0_A4	W30	O	Memory address bit 4
DRAM0_A5	V31	O	Memory address bit 5
DRAM0_A6	V30	O	Memory address bit 6
DRAM0_A7	U31	O	Memory address bit 7
DRAM0_A8	U30	O	Memory address bit 8
DRAM0_A9	T31	O	Memory address bit 9
DRAM0_BA0	U34	O	Bank address output 0
DRAM0_BA1	U33	O	Bank address output 1. BA[1:0] define to which SDRAM bank an active, read, write or precharge command is being applied.
DRAM0_CAS#	T34	O	Column address strobe. Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested (active-low).
DRAM0_CK	K34	O	Non-inverted clock output. CK and CK# are a differential clock signal pair. The SDRAM0 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.

Table 4-12 **DRAM controller pin descriptions (Continued)**

Pin Name	Ball Id	Direction	Description
DRAM0_CK#	L34	O	Inverted clock output. CK and CK# are a differential clock signal pair. The 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 (active-low).
DRAM0_CKE	R30	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.
DRAM0_DM0	H33	O	Write data mask for bits [07:00] of memory data
DRAM0_DM1	H32	O	Write data mask for bits [15:08] of memory data
DRAM0_DM2	N33	O	Write data mask for bits [23:16] of memory data
DRAM0_DM3	N34	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.
DRAM0_DQ0	K31	B	Memory data bus bit 0 (LSB)
DRAM0_DQ1	K30	B	Memory data bus bit 1
DRAM0_DQ10	G33	B	Memory data bus bit 10
DRAM0_DQ11	G32	B	Memory data bus bit 11
DRAM0_DQ12	G30	B	Memory data bus bit 12
DRAM0_DQ13	G31	B	Memory data bus bit 13
DRAM0_DQ14	F34	B	Memory data bus bit 14
DRAM0_DQ15	F33	B	Memory data bus bit 15
DRAM0_DQ16	R34	B	Memory data bus bit 16
DRAM0_DQ17	P31	B	Memory data bus bit 17
DRAM0_DQ18	P30	B	Memory data bus bit 18
DRAM0_DQ19	P32	B	Memory data bus bit 19
DRAM0_DQ2	K33	B	Memory data bus bit 2
DRAM0_DQ20	P33	B	Memory data bus bit 20
DRAM0_DQ21	P34	B	Memory data bus bit 21
DRAM0_DQ22	N31	B	Memory data bus bit 22
DRAM0_DQ23	N30	B	Memory data bus bit 23
DRAM0_DQ24	M30	B	Memory data bus bit 24
DRAM0_DQ25	M32	B	Memory data bus bit 25

Table 4-12 **DRAM controller pin descriptions (Continued)**

Pin Name	Ball Id	Direction	Description
DRAM0_DQ26	M33	B	Memory data bus bit 26
DRAM0_DQ27	M34	B	Memory data bus bit 27
DRAM0_DQ28	L31	B	Memory data bus bit 28
DRAM0_DQ29	L30	B	Memory data bus bit 29
DRAM0_DQ3	K32	B	Memory data bus bit 3
DRAM0_DQ30	L32	B	Memory data bus bit 30
DRAM0_DQ31	L33	B	Memory data bus bit 31 (MSB)
DRAM0_DQ4	J34	B	Memory data bus bit 4
DRAM0_DQ5	J33	B	Memory data bus bit 5
DRAM0_DQ6	J32	B	Memory data bus bit 6
DRAM0_DQ7	J31	B	Memory data bus bit 7
DRAM0_DQ8	H31	B	Memory data bus bit 8
DRAM0_DQ9	G34	B	Memory data bus bit 9
DRAM0_DQS0	H34	B	Data strobes for bits 07:00 of memory data
DRAM0_DQS1	H30	B	Data strobes for bits [15:08] of memory data
DRAM0_DQS2	N32	B	Data strobes for bits [23:16] of memory data
DRAM0_DQS3	M31	B	Data strobes for bits [31:24] of memory data bus. Output by the SMP8634 with the write data; output by the SDRAM with the read data.
DRAM0_RAS#	T33	O	Memory command output. In conjunction with WE#, this signal determines the memory operation requested.
DRAM0_CS#	T32	O	Chip select (active low)
DRAM0_VREFSSTL0	L29	I	SSTL-2 voltage reference input (1.25V)
DRAM0_VREFSSTL1	J30	I	SSTL-2 voltage reference input (1.25V)
DRAM0_VREFSSTL2	G29	I	SSTL-2 voltage reference input (1.25V)
DRAM0_WE#	R32	O	Write enable (active-low). Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested.
DRAM1_A0	B34	O	Memory address bit 0
DRAM1_A1	C34	O	Memory address bit 1
DRAM1_A10	C33	O	Memory address bit 10
DRAM1_A11	E31	O	Memory address bit 11

Table 4-12 **DRAM controller pin descriptions (Continued)**

Pin Name	Ball Id	Direction	Description
DRAM1_A12	F31	O	Memory address bit 12
DRAM1_A13	B31	O	Memory address bit 13. 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.
DRAM1_A2	D34	O	Memory address bit 2
DRAM1_A3	E34	O	Memory address bit 3
DRAM1_A4	D33	O	Memory address bit 4
DRAM1_A5	E33	O	Memory address bit 5
DRAM1_A6	D32	O	Memory address bit 6
DRAM1_A7	E32	O	Memory address bit 7
DRAM1_A8	F32	O	Memory address bit 8
DRAM1_A9	D31	O	Memory address bit 9
DRAM1_BA0	A33	O	Bank address output 0
DRAM1_BA1	B33	O	Bank address output 1. BA[1:0] define to which SDRAM bank an active, read, write or precharge command is being applied.
DRAM1_CAS#	A32	O	Column address strobe (active-low). Memory command output. In conjunction with WE#, this signal determines the memory operation requested.
DRAM1_CK	A26	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.
DRAM1_CK#	A27	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 (active-low).
DRAM1_CKE	F30	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.
DRAM1_DM0	B24	O	Write data mask for bits [07:00] of memory data
DRAM1_DM1	C24	O	Write data mask for bits [15:08] of memory data
DRAM1_DM2	B29	O	Write data mask for bits [23:16] of memory data

Table 4-12 **DRAM controller pin descriptions (Continued)**

Pin Name	Ball Id	Direction	Description
DRAM1_DM3	A29	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.
DRAM1_DQ0	D26	B	Memory data bus bit 0 (LSB)
DRAM1_DQ1	E26	B	Memory data bus bit 1
DRAM1_DQ10	B23	B	Memory data bus bit 10
DRAM1_DQ11	C23	B	Memory data bus bit 11
DRAM1_DQ12	D23	B	Memory data bus bit 12
DRAM1_DQ13	A22	B	Memory data bus bit 13
DRAM1_DQ14	B22	B	Memory data bus bit 14
DRAM1_DQ15	C22	B	Memory data bus bit 15
DRAM1_DQ16	A31	B	Memory data bus bit 16
DRAM1_DQ17	D30	B	Memory data bus bit 17
DRAM1_DQ18	E30	B	Memory data bus bit 18
DRAM1_DQ19	C30	B	Memory data bus bit 19
DRAM1_DQ2	B26	B	Memory data bus bit 2
DRAM1_DQ20	B30	B	Memory data bus bit 20
DRAM1_DQ21	A30	B	Memory data bus bit 21
DRAM1_DQ22	D29	B	Memory data bus bit 22
DRAM1_DQ23	E29	B	Memory data bus bit 23
DRAM1_DQ24	E28	B	Memory data bus bit 24
DRAM1_DQ25	C28	B	Memory data bus bit 25
DRAM1_DQ26	B28	B	Memory data bus bit 26
DRAM1_DQ27	A28	B	Memory data bus bit 27
DRAM1_DQ28	D27	B	Memory data bus bit 28
DRAM1_DQ29	E27	B	Memory data bus bit 29
DRAM1_DQ3	C26	B	Memory data bus bit 3
DRAM1_DQ30	C27	B	Memory data bus bit 30
DRAM1_DQ31	B27	B	Memory data bus bit 31 (MSB)
DRAM1_DQ4	A25	B	Memory data bus bit 4
DRAM1_DQ5	B25	B	Memory data bus bit 5

Table 4-12 **DRAM controller pin descriptions (Continued)**

Pin Name	Ball Id	Direction	Description
DRAM1_DQ6	C25	B	Memory data bus bit 6
DRAM1_DQ7	D25	B	Memory data bus bit 7
DRAM1_DQ8	D24	B	Memory data bus bit 8
DRAM1_DQ9	A23	B	Memory data bus bit 9
DRAM1_DQS0	A24	B	Data strobes for bits 07:00 of memory data
DRAM1_DQS1	E24	B	Data strobes for bits [15:08] of memory data
DRAM1_DQS2	C29	B	Data strobes for bits [23:16] of memory data
DRAM1_DQS3	D28	B	Data strobes for bits [31:24] of memory data bus. Output by the SMP8634 with the write data; output by the SDRAM with the read data.
DRAM1_RAS#	B32	O	DRAM1 row address strobe (active-low). Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested.
DRAM1_CS#	C32	O	DRAM1 chip select (active-low).
DRAM1_VREFSSTL0	F28	I	SSTL-2 voltage reference input (1.25V)
DRAM1_VREFSSTL1	E25	I	SSTL-2 voltage reference input (1.25V)
DRAM1_VREFSSTL2	E23	I	SSTL-2 voltage reference input (1.25V)
DRAM1_WE#	C31	O	DRAM1 write enable (active-low). Memory command output. In conjunction with CAS# and WE#, this signal determines the memory operation requested.

4.6 Electrical Characteristics

All DRAM I/O signals conform to the electrical characteristics specified for SSTL-2 Class II interface as specified in the JEDEC standard, JESD79D, dated January 2004. Refer to the JEDEC standard for details regarding switching levels, drive performance, etc.

4.7 Timing Diagrams

The DDR SDRAM controller conforms to the interface specifications defined by the JEDEC standard, JESD79D, dated January 2004. All generated and required timings conform to the requirements of the JESD79 specification for the ‘DDR400’ speed grade.

The critical timings below are generated by characterization. All other timings specified in the JESD79 specification are guaranteed by design for the ‘DDR400’ speed grade, and are not listed in this document.

Table 4-13 **DRAM interface timing parameters**

Parameter	Minimum	Typical	Maximum	Units
t_{CK}	5.00		12.0	ns
t_{CH}	0.45		0.55	t_{CK}
t_{CL}	0.45		0.55	t_{CK}
t_{IS}	1.50			ns
t_{IH}	1.50			ns
t_{DQDLY}			0.55	ns
t_{DQVLD}	0.63			ns
t_{DS}	0.70			ns
t_{DH}	0.70			ns
t_{DQSH}	0.42			t_{CK}
t_{DQSL}	0.42			t_{CK}

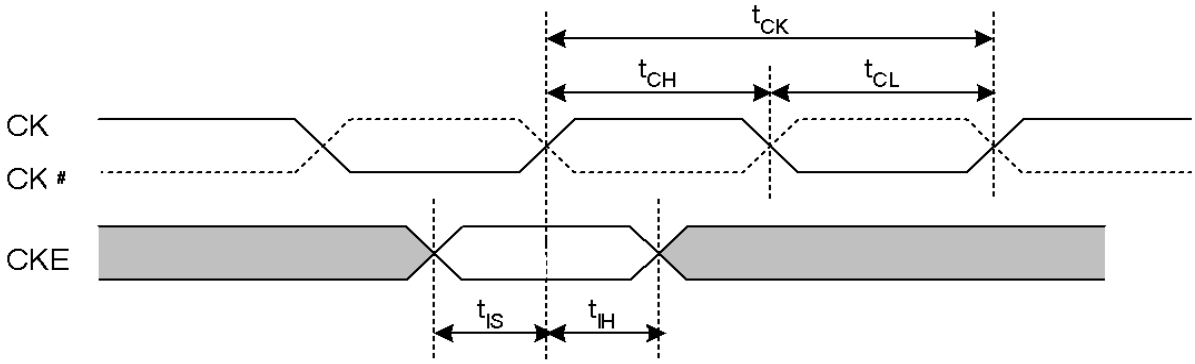


Figure 4-5 **CK/CK#/CKE timing diagram**

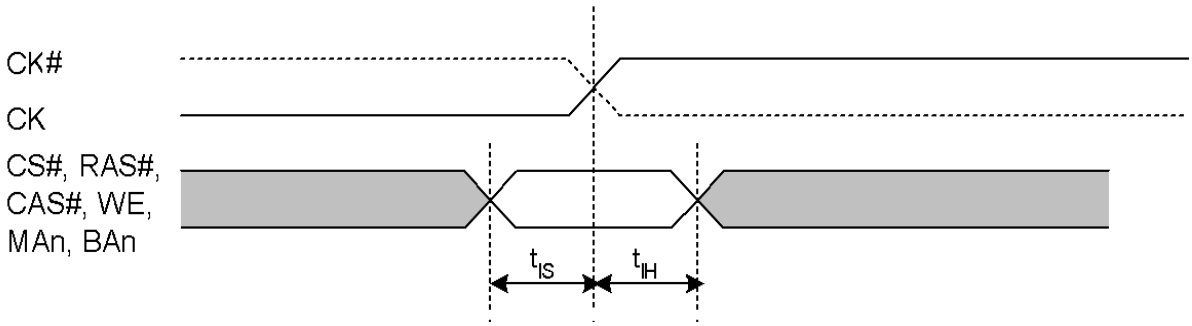


Figure 4-6 **Command timing diagram**

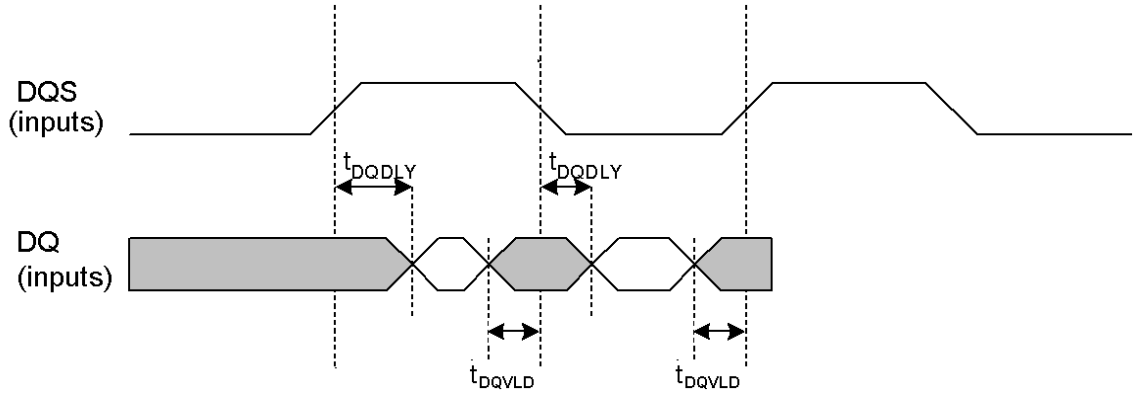


Figure 4-7 **DQ/DQS read timing diagram**

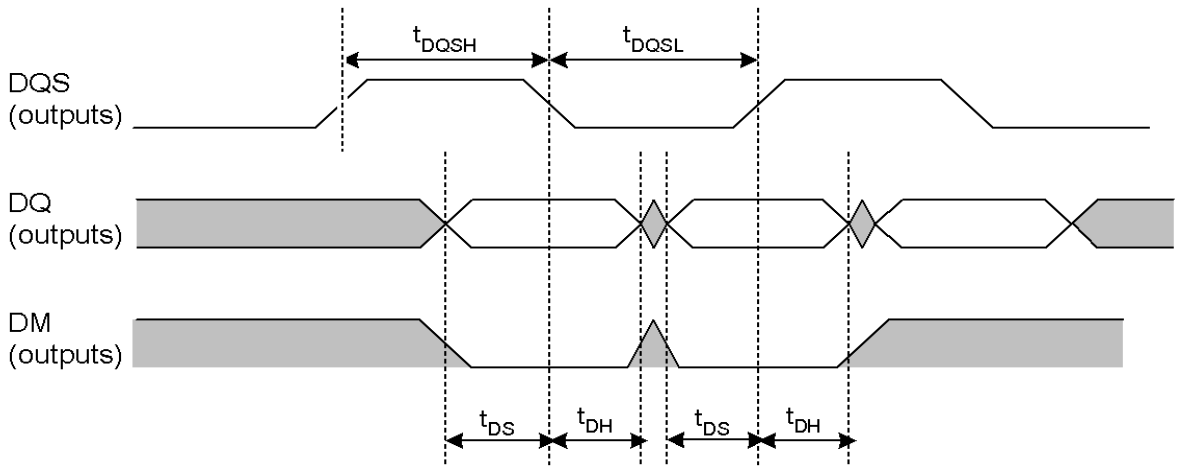


Figure 4-8 **DQ/DQS/DM write timing diagram**

5 System Block

5.1 Block Diagram of System Block

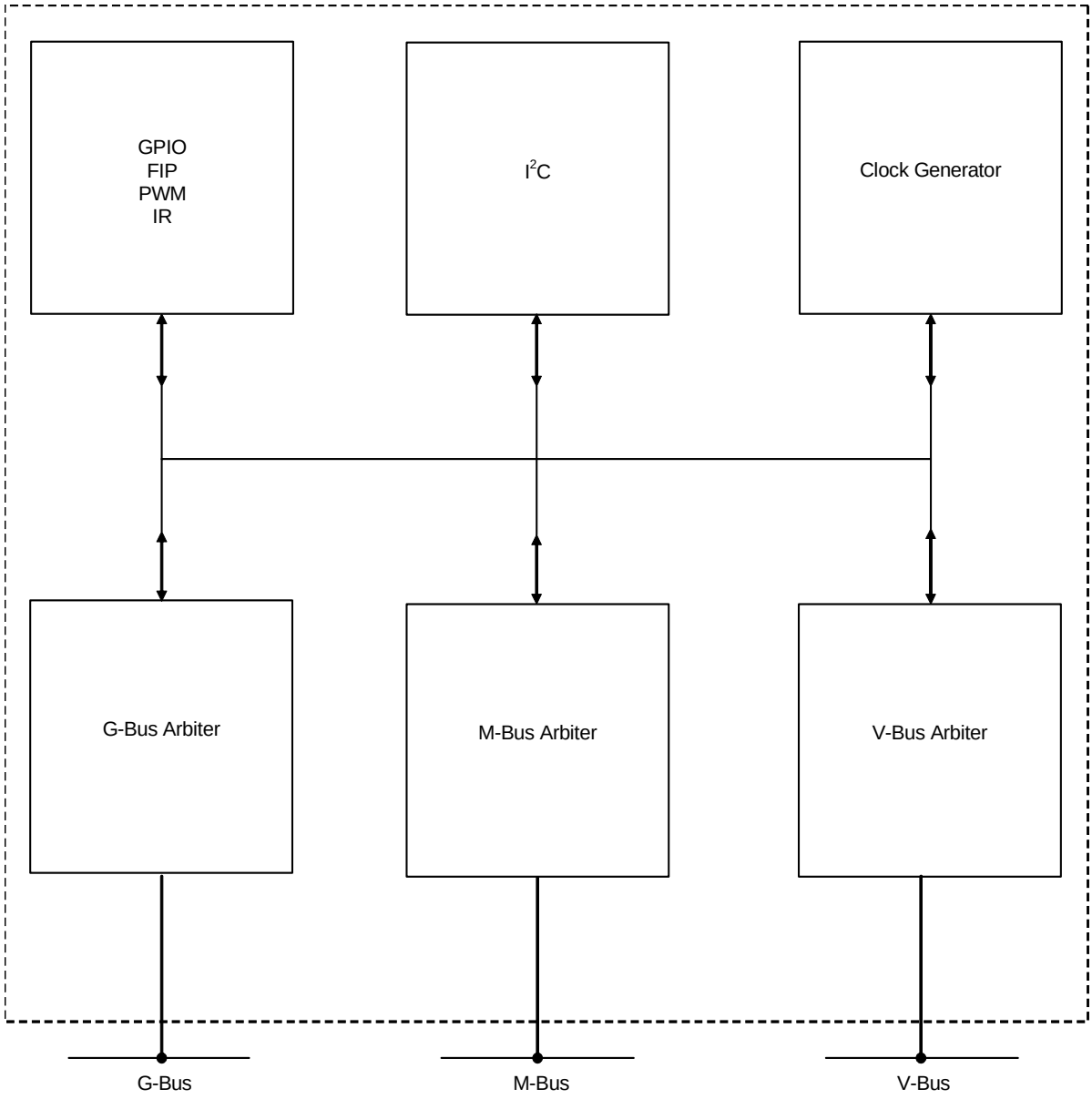


Figure 5-1 **Block diagram of system block**

5.2 Introduction

The SMP8634 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 five masters: MIPS CPU, PCI, MPEG0, MPEG1 and TDMX
- An M-Bus arbiter with 17 channels
- A V-Bus arbiter with 15 channels

Many SMP8634 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 SMP8634 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

5.3 General Purpose I/O (GPIO)

The SMP8634 provides 32 dedicated general-purpose input/output (GPIO) pins to help eliminate the glue logic necessary for system integration. Additional GPIO can be achieved if required, since certain other interfaces on the device have GPIO capability as an alternate function.

The 32 GPIO pins are individually configurable as inputs or outputs. Pins default to input mode after device reset. Registers 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 direct software controlled mode. When enabled, the PWM blocks produce 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.

Because the control of the GPIO configuration registers by the host CPU involves hardware elements (write buffers, shared buses) with variable “latency”, the GPIO pins should not be used for applications requiring precise control of timing. Generally, the GPIO pins should be use for low-speed or non-critical control and signaling applications.

5.3.1 Features

- Supports 32 independently controllable inputs/outputs
- Two pulse width modulators (PWM) as special functions
- Up to four pins can be used for host CPU interrupts

5.3.2 GPIO Controllers

The 32 dedicated GPIO signals are created from two separate instances of a GPIO controller block. The two controllers differ somewhat in their capabilities. The notation “High Block Controller” (HBC) refers to the controller managing GPIO(31:16), and “Low Block Controller” (LBC) refers to the controller managing GPIO(15:00).

5.3.2.1 Low Block Controller

Features of the LBC include the following:

- Independent direction control for each of the 16 GPIO pins
- Up to four of the 16 GPIO pins may be used as an interrupt to the system processor
- Special “alternate” function modules (two I^2C controllers, two pulse-width modulators, front-panel interface) optionally share pins with the GPIO pins
- A “masking” function is supported to allow only specified bits to be updated by a write operation to the data register

During a write to the Output Data Register, the 16 data bits to be written are placed in the lower 16 bits of the 32-bit value written by the CPU. The upper 16 bits function as a “mask” value. A logic zero in any of the upper 16 bits causes the corresponding data bit in the lower 16 bits to be NOT written (previous value is retained). As an example, a logic zero in bit position 24 causes Output Data Register bit 8 to be unchanged during a write, while a logic one in bit position 24 causes bit 8 to be written. This simplifies control of individual GPIO bits.

5.3.3 Block Diagram

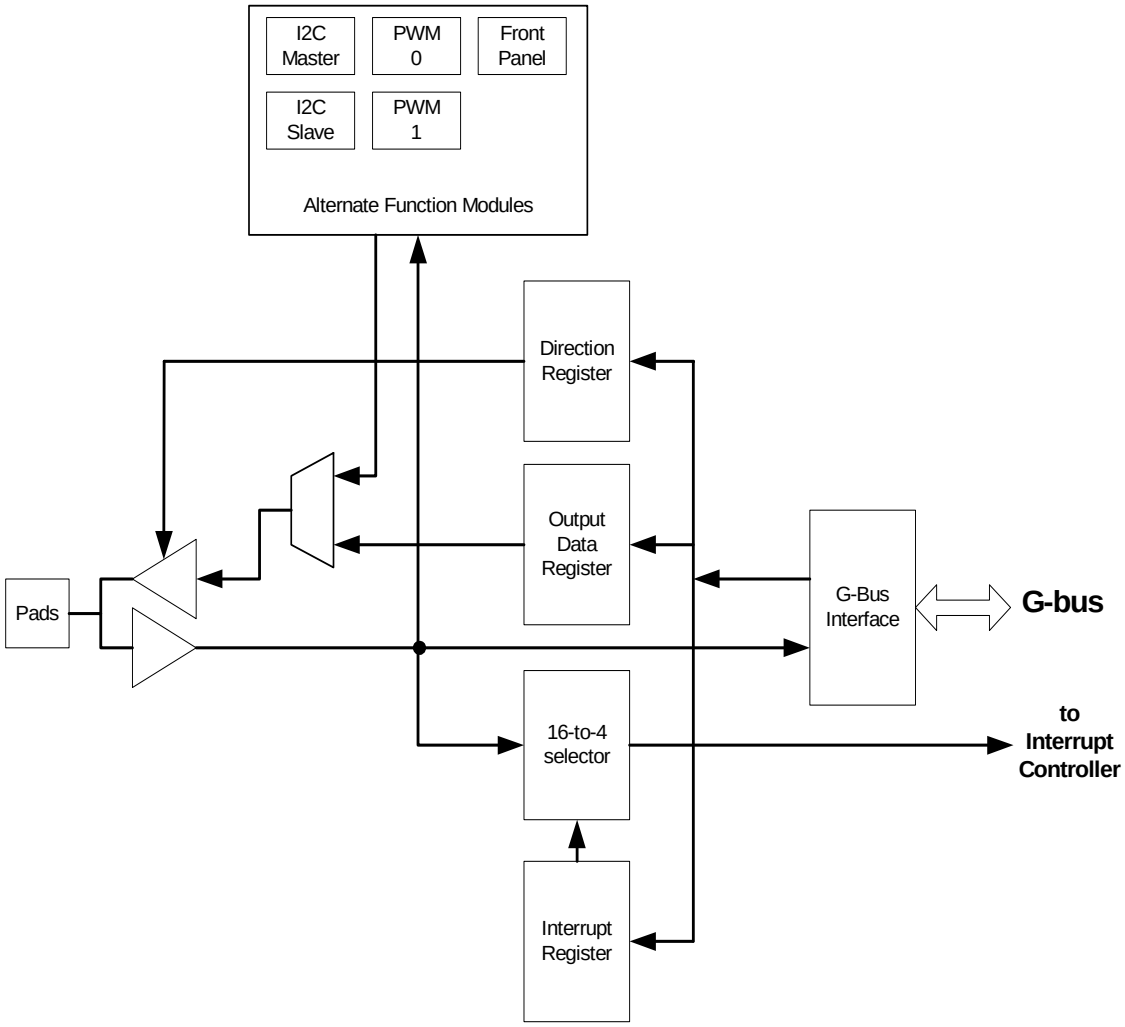


Figure 5-2 **LBC general block diagram**

The LBC includes both “alternate functions” and “selectable inputs” for greater flexibility.

Alternate functions include the two PWM modules, master and slave I^2C controllers, and serial front-panel interface (FIP) controller. When one of the alternate functions is enabled (e.g., I^2C master), the corresponding pins (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, I^2C master, I^2C slave, and FIP) which can be independently enabled. See Figure the following figure for pin assignments of the alternate functions.

Selectable inputs are input functions only (e.g., PCI Interrupt Input A). They are preassigned to a particular pin at reset, but can be reassigned to another pin using the control register. Pins used as the selectable inputs should be set up as GPIO pins and configured as inputs. The following figure shows the GPIO alternate function and selectable inputs in detail:

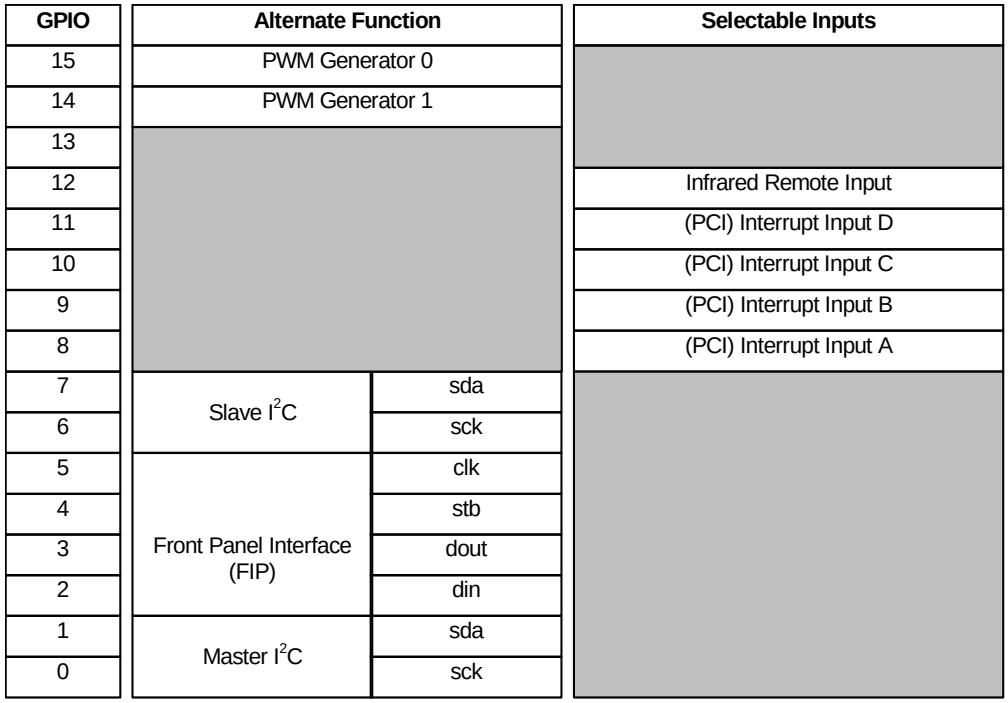


Figure 5-3 **GPIO[15:0] alternate and selectable inputs**

5.3.3.2 High Block Controller

The HBC is similar to the LBC, with the following simplifications:

- The HBC does not support the use of any of GPIO(31:16) as interrupt inputs. Therefore, the interrupt mapping register and 16:4 selector are not implemented.
- No “alternate functions” (I^2C, PWMs, or front-panel interface) are included. Only simple GPIO control is supported.

Output Data Register masking is supported in the same way described for the LBC.

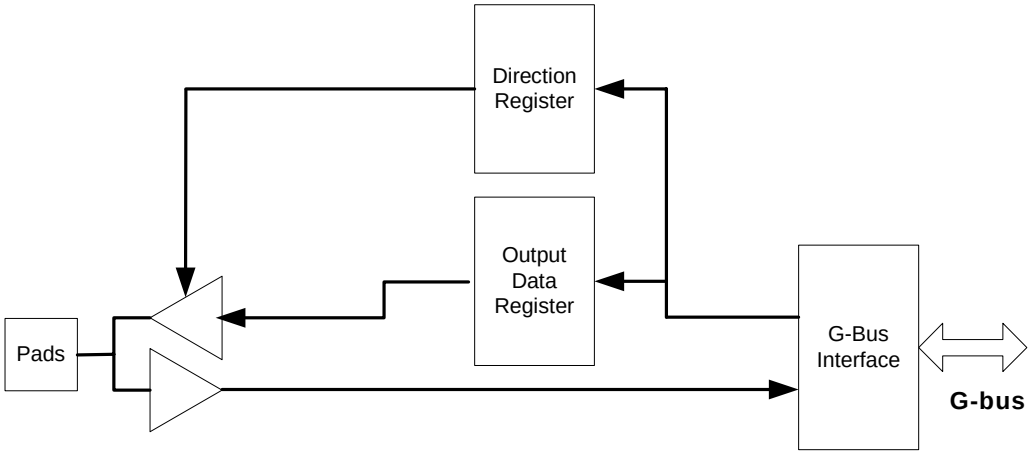


Figure 5-4 **HBC general block diagram**

5.3.3.3 Behavior of GPIO Pins during Chip Initialization

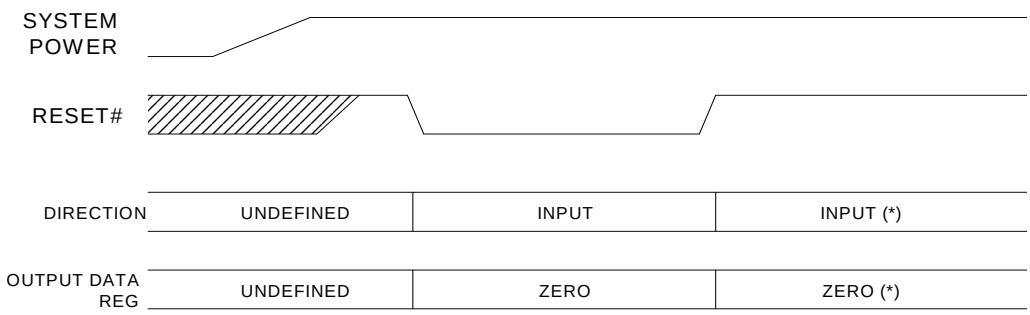


Figure 5-5 **Behavior of GPIO pins during chip initialization**

5.3.4 Register Map

5.3.4.1 GPIO Registers

Table 5-1 **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
+0508	SYS_GPIO_INT	R/W	GPIO Interrupt Register
+710C	GPIO_DIR2	R/W	GPIO Direction 2 Register
+7110	GPIO_DATA2	R/W	GPIO Data 2 Register

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

5.3.5 Pin Description

5.3.5.1 GPIO Pins

The SMP8634 provides 32 general-purpose I/O (GPIO) pins which may be individually configured as inputs or outputs and used for design-specific applications.

Table 5-2 **GPIO pin description**

Pin Name	Ball ID	Direction	Description
GPIO0	A5	B	General-purpose IO pin 0 Alternate function: Master I^2C.
GPIO1	B5	B	General-purpose IO pin 1 Alternate function: Master I^2C.
GPIO10	E6	B	General-purpose IO pin 10 Selectable input: (PCI) Interrupt Input C.
GPIO11	E8	B	General-purpose IO pin 11 Selectable input: (PCI) Interrupt Input D.
GPIO12	B7	B	General-purpose IO pin 12 Selectable input: Infrared Remote input.
GPIO13	C7	B	General-purpose IO pin 13
GPIO14	D7	B	General-purpose IO pin 14 Alternate function: PWM generator 1.
GPIO15	E7	B	General-purpose IO pin 15 Alternate function: PWM generator 0.
GPIO16	AF5	B	General-purpose IO pin 16

Table 5-2 **GPIO pin description (Continued)**

Pin Name	Ball ID	Direction	Description
GPIO17	AF4	B	General-purpose IO pin 17
GPIO18	AF3	B	General-purpose IO pin 18
GPIO19	AF2	B	General-purpose IO pin 19
GPIO2	C5	B	General-purpose IO pin 2 Alternate function: Front Panel Interface (FIP).
GPIO20	AE5	B	General-purpose IO pin 20
GPIO21	AE4	B	General-purpose IO pin 21
GPIO22	AE3	B	General-purpose IO pin 22
GPIO23	AE2	B	General-purpose IO pin 23
GPIO24	AD5	B	General-purpose IO pin 24
GPIO25	AD4	B	General-purpose IO pin 25
GPIO26	AD3	B	General-purpose IO pin 26
GPIO27	AD2	B	General-purpose IO pin 27
GPIO28	AC5	B	General-purpose IO pin 28
GPIO29	AC4	B	General-purpose IO pin 29
GPIO3	D5	B	General-purpose IO pin 3 Alternate function: Front Panel Interface (FIP).
GPIO30	AB5	B	General-purpose IO pin 30
GPIO31	AB4	B	General-purpose IO pin 31
GPIO4	D4	B	General-purpose IO pin 4 Alternate function: Front Panel Interface (FIP).
GPIO5	E4	B	General-purpose IO pin 5 Alternate function: Front Panel Interface (FIP).
GPIO6	E5	B	General-purpose IO pin 6 Alternate function: Slave I^2C.
GPIO7	B6	B	General-purpose IO pin 7 Alternate function: Slave I^2C.
GPIO8	C6	B	General-purpose IO pin 8 Selectable input: (PCI) Interrupt Input A.
GPIO9	D6	B	General-purpose IO pin 9 Selectable input: (PCI) Interrupt Input B.

5.3.6 Electrical Characteristics

5.3.6.1 GPIO DC Electrical Characteristics

Table 5-3 **GPIO 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.

5.4 Front Panel Controller Interface (FIP)

5.4.1 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.

5.4.2 Features

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

5.4.3 Block Diagram

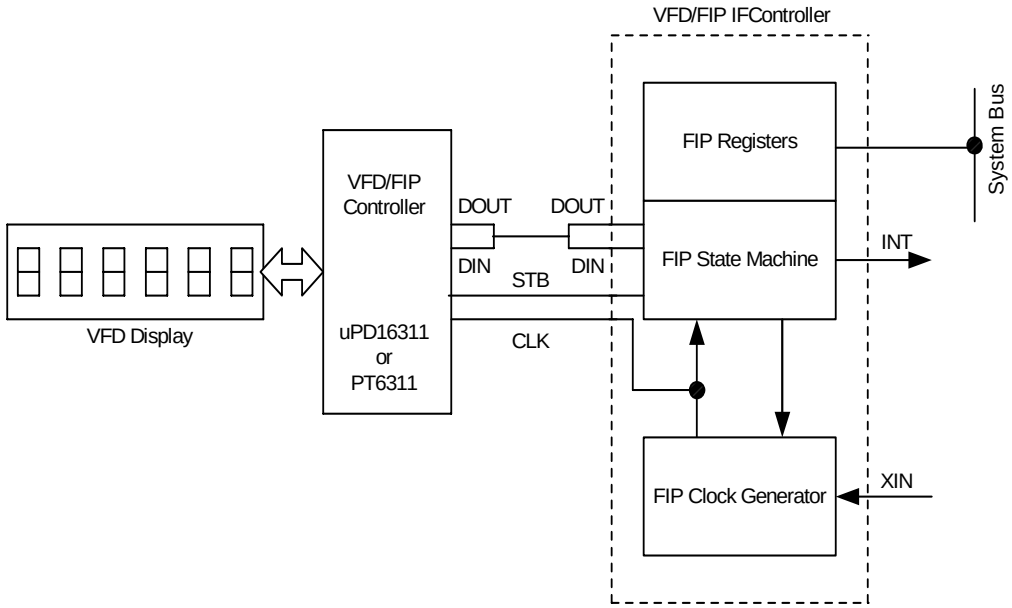


Figure 5-6 **FIP block diagram**

5.4.4 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 (SMP8634 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.

5.4.5 Register Map

5.4.5.1 FIP Registers

Table 5-4 **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.

5.4.6 Pin Description

5.4.6.1 FIP Pins

Table 5-5 **FIP pin description**

Pin Name	Ball ID	Direction	Description
GPIO2	C5	B	FIP serial data input
GPIO3	D5	B	FIP serial data output
GPIO4	D4	B	FIP data strobe
GPIO5	E4	B	FIP serial I/O clock

5.4.7 Electrical Characteristics

5.4.7.1 FIP Electrical Characteristics

Table 5-6 **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.

5.4.7.2 FIP AC Electrical Characteristics

Table 5-7 **FIP AC characteristics**

Symbol	Units	Min	Typ	Max
T_{CKL}	ns	500		
T_{CKH}	ns	500		
T_{DV}	ns	0		10
T_{SU}	ns	10		
T_{H}	ns	6		
T_{STBDLY}	ns	1250		
T_{PWSTB}	ns	1000		

5.4.8 Timing Diagram

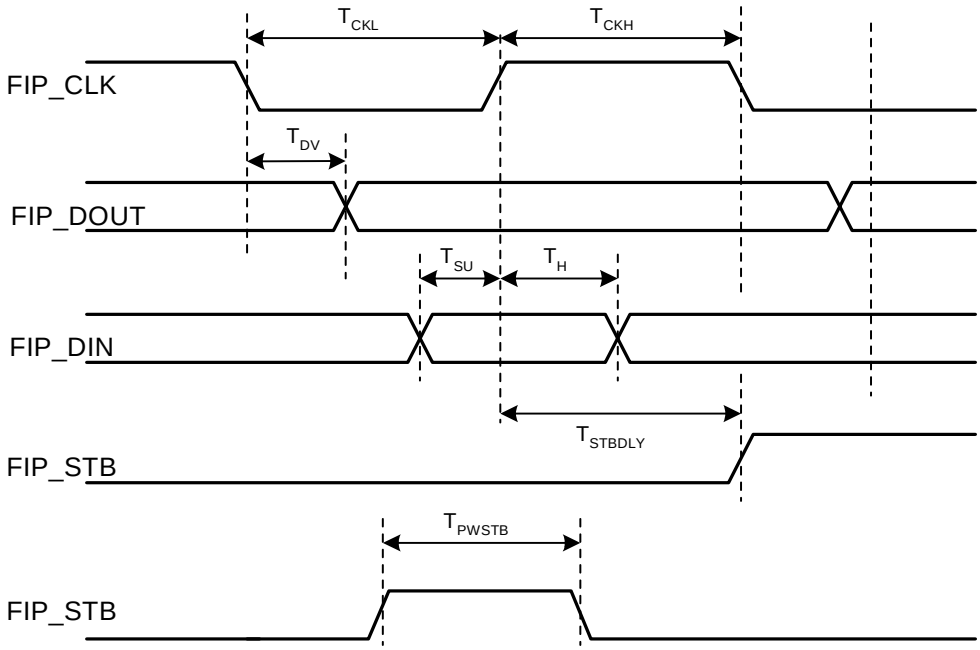


Figure 5-7 **FIP timing diagram**

5.5 Infrared Decoder

5.5.1 Introduction

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

5.5.2 Features

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

5.5.3 Block Diagram

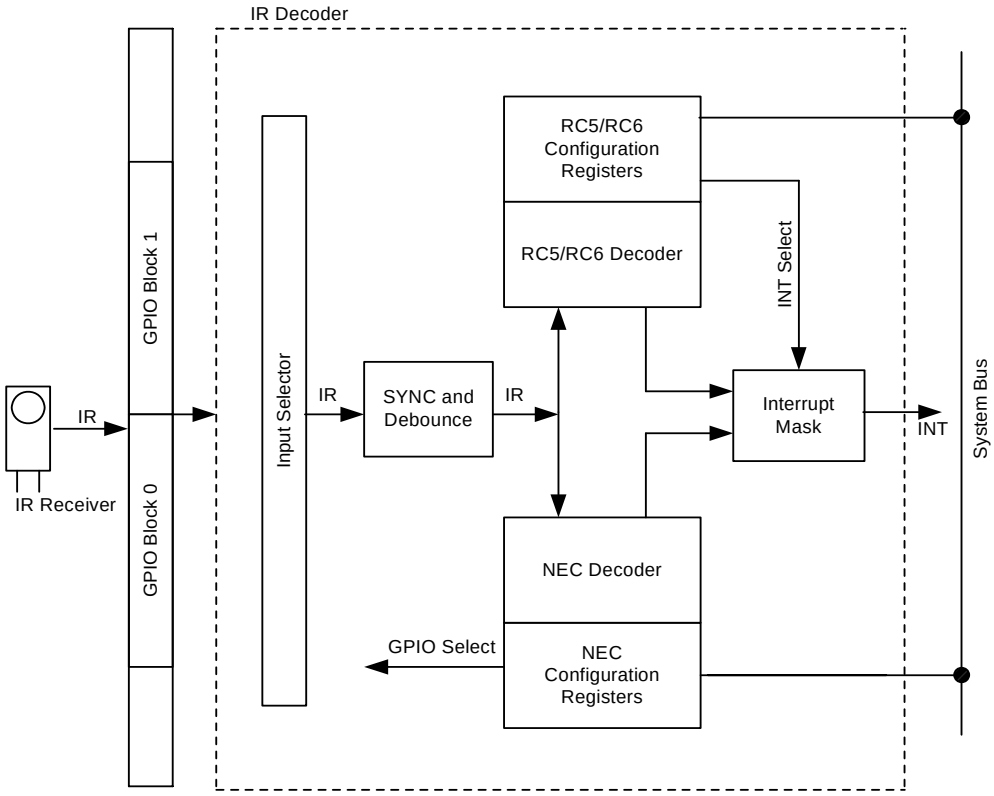


Figure 5-8 **Infrared decoder block diagram**

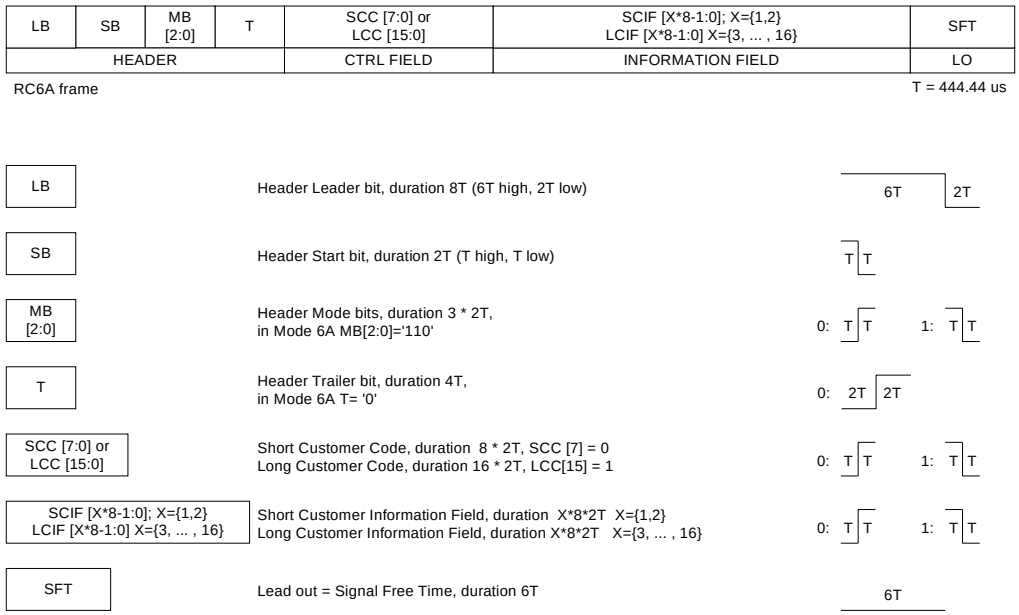
5.5.4 Functional Description

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.

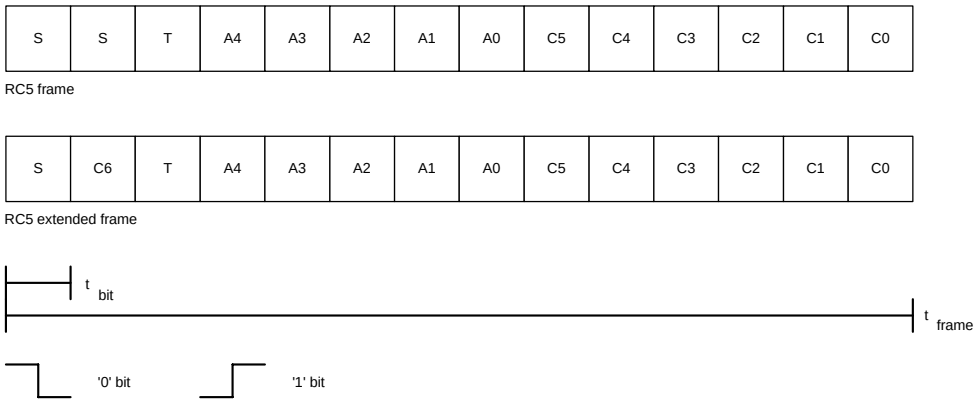
5.5.4.1 Supported RC6 Mode 6A Formats



5.5.4.2 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.



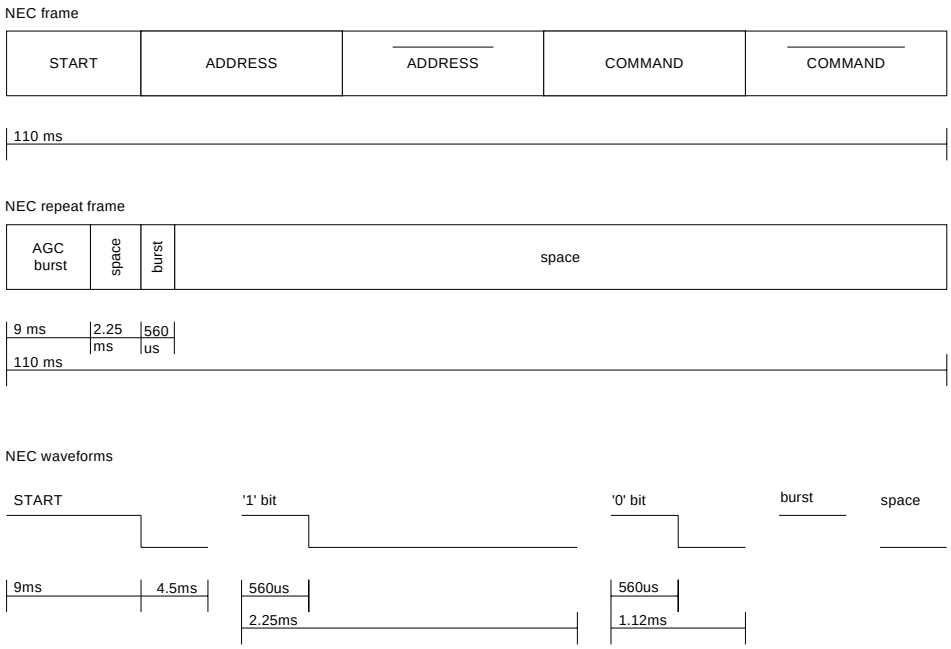
S: Start bit, T: Toggle bit, A[4:0]: 5 system address bits, C[5:0]: 6 system command bits (supported in the RC5 standard format), C[6:0]: 7 system command bits (supported only in the RC5 extended format)

Table 5-8 **RC5 timing parameters**

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

5.5.4.3 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 5-9 **NEC timing parameters**

	Minimum	Typical	Maximum	Units
t_{frame}		110		ms

5.5.5 Register Map

5.5.5.1 Infrared Decoder Registers

Table 5-10 **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.

5.5.6 Pin Description

5.5.6.1 Infrared Decoder Pin

Table 5-11 **Infrared decoder pin description**

Pin Name	Ball ID	Direction	Description
GPIO12	B7	B	Default pin assigned to the IR decoder input function (may be mapped to any other GPIO pin under the software control).

5.5.7 Electrical Characteristics

5.5.7.1 Infrared Decoder DC Characteristics

Table 5-12 **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.

Inter Integrated Circuits (I^2C)

5.6 Introduction

The I^2C master and slave interfaces enable the SMP8634 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 SMP8634 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.

5.7 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

5.8 Block Diagram

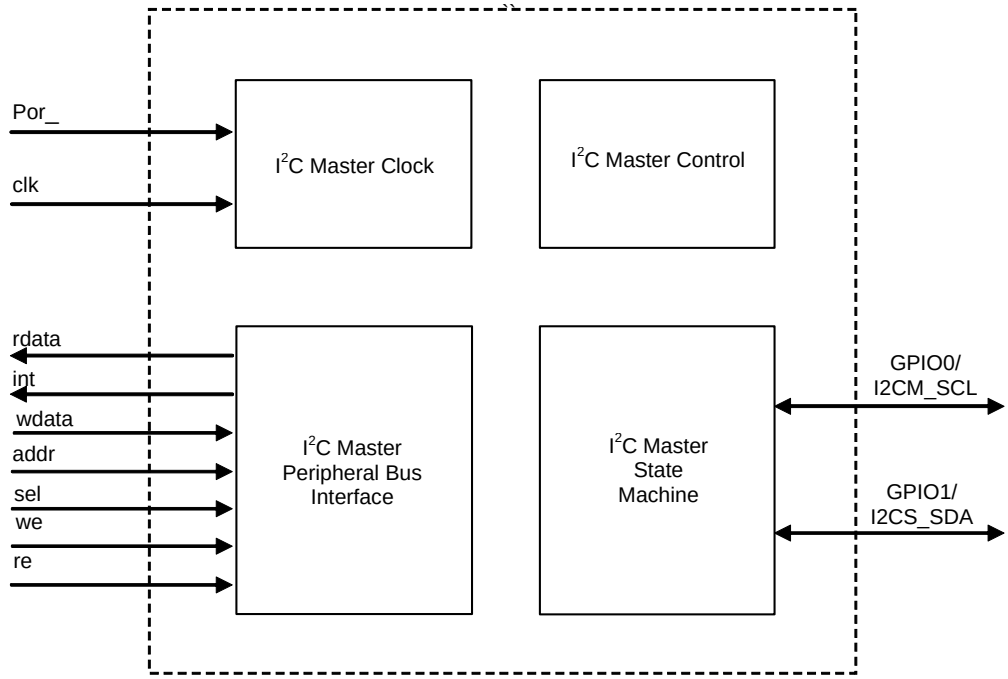


Figure 5-9 **I²C master block diagram**

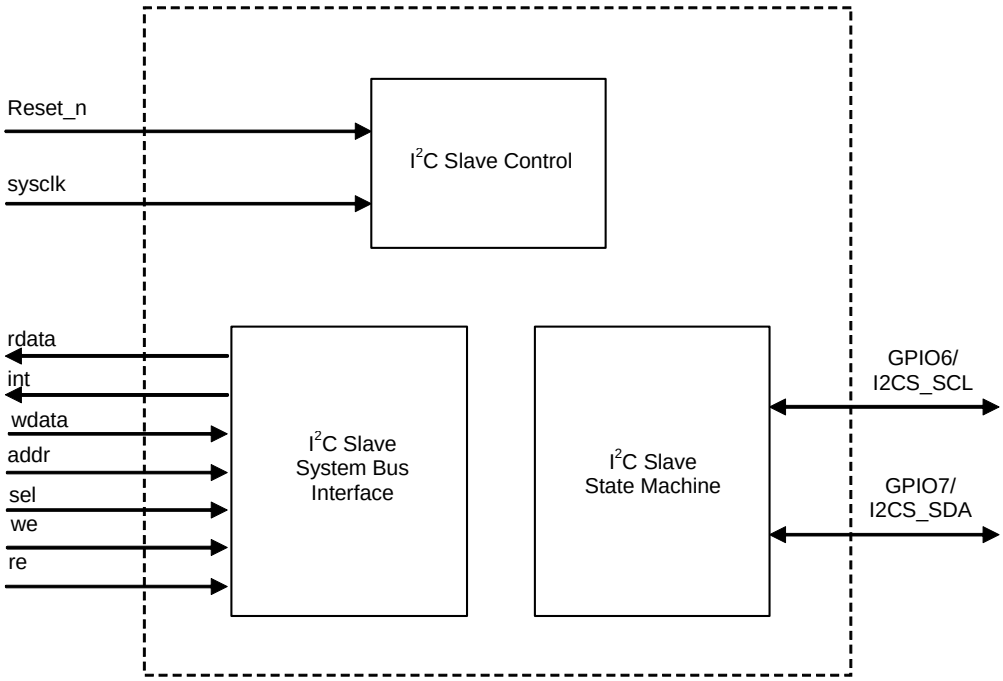


Figure 5-10 **I²C slave block diagram**

5.9 Functional Description

5.9.1 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 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.

5.9.2 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 SMP8634 acts as a slave device and the device address register is used to select the 7-bit slave address of the SMP8634. 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 SMP8634 acknowledges being selected by bringing the SDA low on the ninth pulse of the SCL clock. Otherwise, the SMP8634 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 SMP8634, 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 SMP8634. An interrupt will be generated and the SMP8634 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.

5.10 Register Maps

5.10.1 I^2C Master Registers

Table 5-13 **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.

5.10.2 I²C Slave Registers

Table 5-14 **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.

5.11 Pin Description

5.11.1 I²C Master and Slave Interface Pins

Table 5-15 **I²C master and slave interface pin descriptions**

Pin Name	Ball ID	Direction	Description
GPIO0	A5	B	I2CM_SCL. I²C master interface serial clock
GPIO1	B5	B	I2CM_SDA. I²C master interface serial data
GPIO6	E5	B	I2CS_SCL. I²C slave interface serial clock
GPIO7	B6	B	I2CS_SDA. I²C slave interface serial data

5.12 Electrical Characteristics

5.12.1 I²C Master and Slave Interface DC Characteristics

The following DC electrical characteristics apply to both SCL and SDA signals. These signals are bidirectional pins with open-drain output stages. Proper external resistive termination (pull-up resistors) must be used to achieve correct signaling levels.

Table 5-16 **I²C master and slave interface DC characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OL}	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
V_{IH}	Input high voltage	V	2		5.5
V_{IL}	Input low voltage	V	-0.3		0.8
I_{IN}	Input current (@V_{IN} = 0.2V to VDD_3V3)	µA	-10		10

5.12.2 I²C Master and Slave Interface AC Characteristics

Table 5-17 **I²C master and slave interface AC characteristics**

Parameter	Description	Minimum	Typical	Maximum	Units
F_{SCL}	SCL clock frequency			400	KHz
T_{HIGH}	SCL clock high time	1.3			µs
T_{LOW}	SCL clock low time	0.6			µs
T_{BUF}	Bus free time	1.3			µs
$T_{SU;DAT}$	Data setup time	250			ns
$T_{HD;DAT}$	Data hold time	0			ns
$T_{SU;STA}$	Setup time, START condition	0.6			µs
$T_{HD;STA}$	Hold time, START condition	0.6			µs
$T_{SU;STO}$	Setup time, STOP condition	0.6			µs

5.13 Timing Diagrams

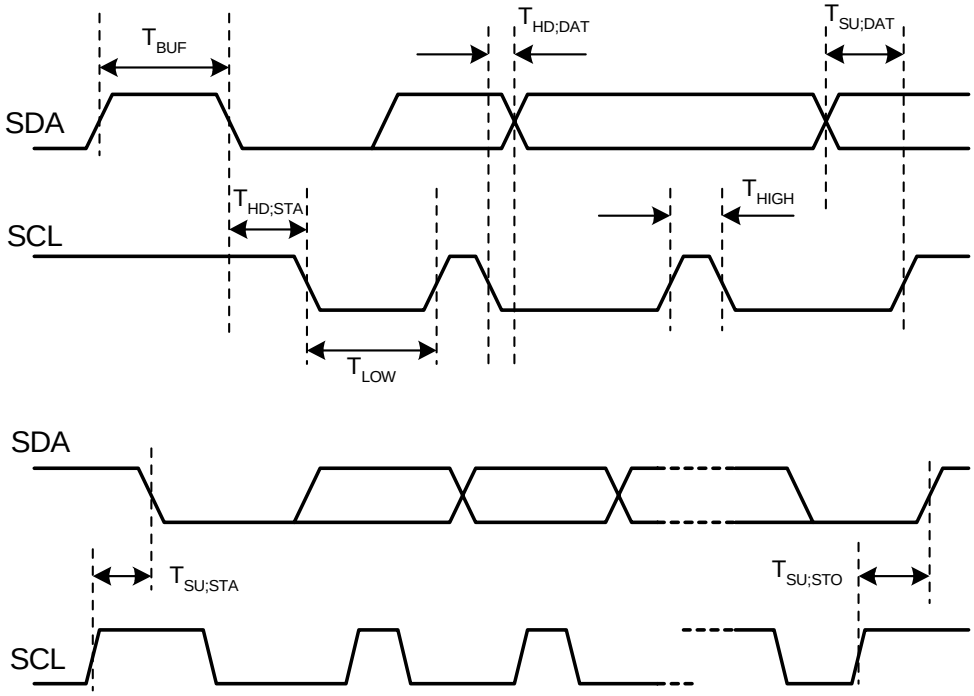


Figure 5-11 **I²C master transfer timing diagram**

Clock Generator

5.14 Introduction

The clock generator contains audio clocks, video clocks, system clock and a CPU clock. The clock generator creates two high speed (up to 300MHz) 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.

5.15 Features

- Creates two high speed clocks (up to 300MHz)
- Contains audio clocks, video clocks, system clock and a CPU clock
- Supports programmable PLLs

5.16 Block Diagram

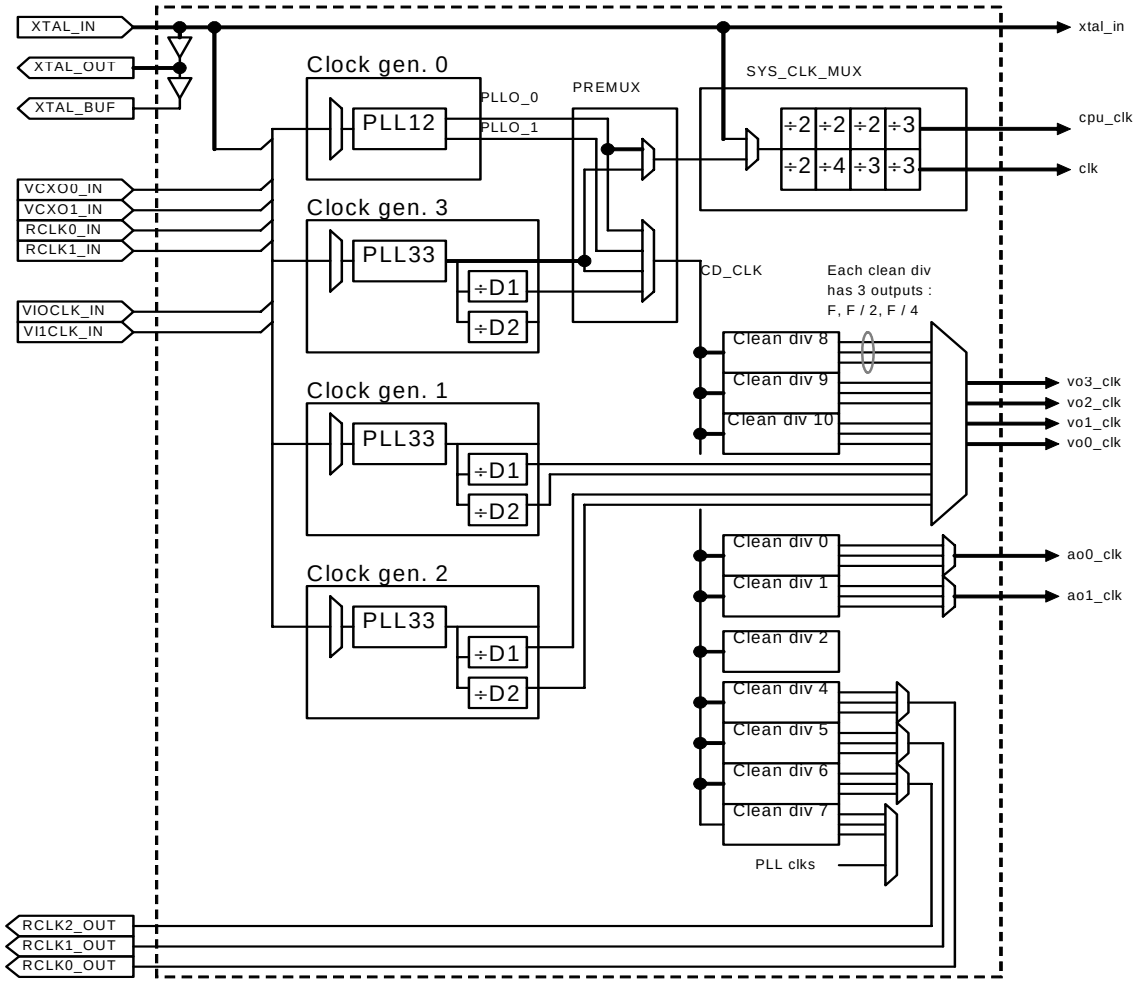


Figure 5-12 **Clock generator block diagram**

5.17 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, $F_{Out} = SysClk / (2 + divider \times 2^{-27})$.

5.18 Register Map

5.18.1 Clock Generation Registers

Table 5-18 **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
+0030	SYS_HOSTCLK_MUX	R/W	System Host Clock Mux 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.

5.18.2 Clean Dividers Registers

Table 5-19 **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

Table 5-19 **Clean dividers registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+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		
+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.

5.18.3 Clock Counter Registers

Table 5-20 **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
+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.

5.19 Pin Description

5.19.1 Clock Generator Pins

Table 5-21 **Clock generator pin description**

Pin Name	Ball ID	Direction	Description
XTAL_IN	A10	I	27MHz crystal oscillator input
XTAL_OUT	A9	O	Crystal oscillator output
XTAL_BUF	B9	O	Buffered crystal oscillator output
VCXO0_IN	C9	I	Input from external VCXO #0
VCXO1_IN	C8	I	Input from external VCXO #1
RCLK0_IN	B8	I	PLL reference clock input #0
RCLK0_OUT	D8	O	PLL reference clock output #0
RCLK1_OUT	D9	O	Auxiliary clock output #1
RCLK1_XTAL_IN	A7	I	Crystal oscillator input

Table 5-21 **Clock generator pin description**

Pin Name	Ball ID	Direction	Description
RCLK1_XTAL_OUT	A6	O	Crystal oscillator output

5.20 Electrical Characteristics

5.20.1 Clock Generator DC Characteristics

Table 5-22 **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.

5.20.2 System Clock Characteristics

Table 5-23 **System clock characteristics**

Symbol	Description	Min	Typ	Max	Units
F_{REF}	Reference clock frequency		27.000		MHz
	Reference clock tolerance			25	ppm
DERIVED CLOCKS					
F_{MEMCK}	DDR clock frequency			201	MHz
F_{VIDCLK}	Video output pixel clock frequency			148.5	MHz
F_{PCICLK}	PCI clock frequency			66	MHz
F_{AUDCLK}	Audio clock frequency			50	MHz
Delta (F_{MEMCK})	DDR clock long-term jitter			350	ps
Delta (F_{VIDCLK})	Video clock long-term jitter			500	ps
Delta (F_{PCICLK})	PCI clock long-term jitter			950	ps
Delta (F_{AUDCLK})	Audio clock long-term jitter			950	ps

6 Host Interface

6.1 Block Diagram of Host Interface

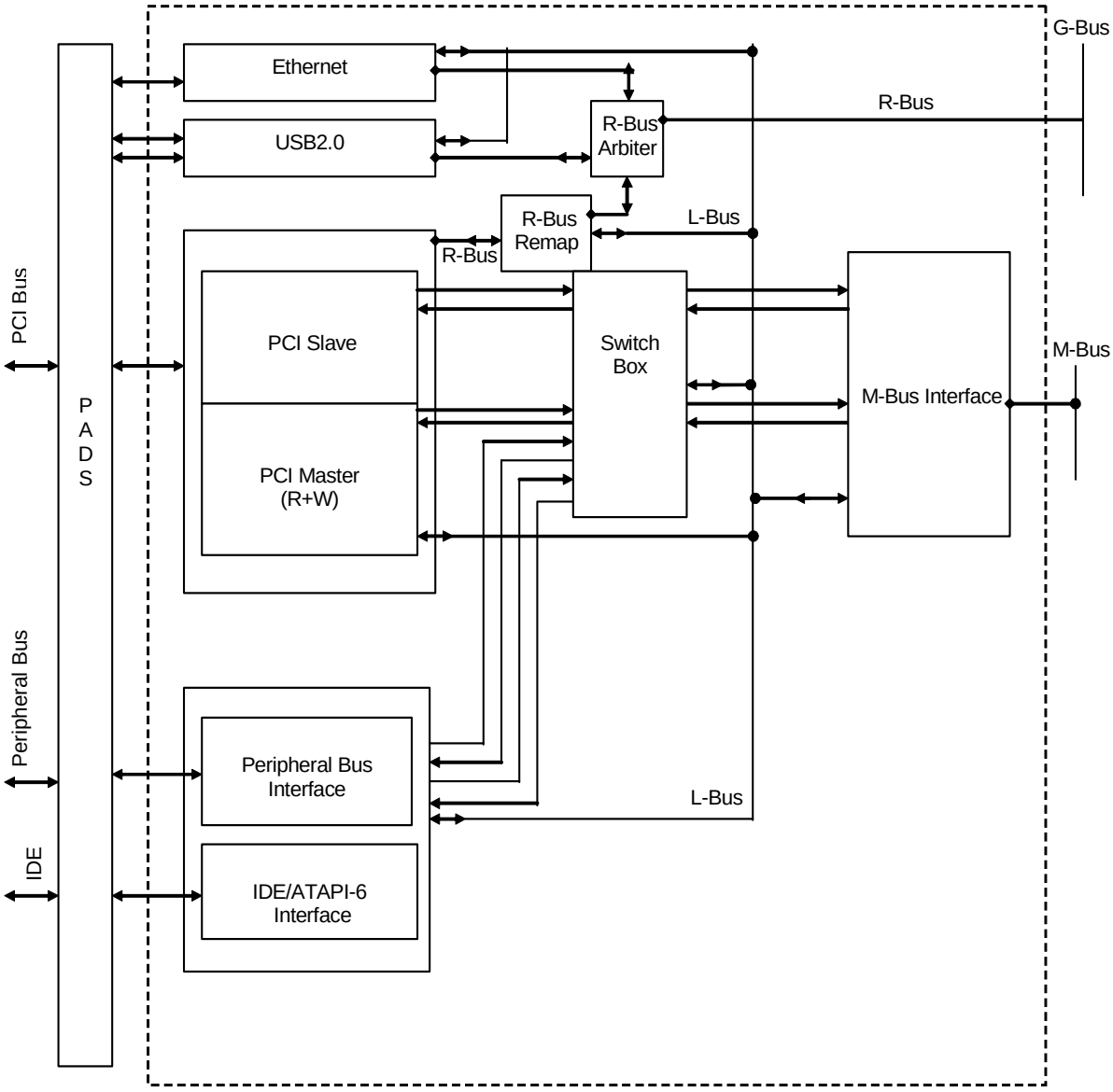


Figure 6-1 **Host interface block diagram**

6.2 Introduction

The host interface unit provides the interface between the primary internal buses (G-bus and M-bus) and the PCI, the peripheral bus and the IDE interfaces. 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 64MB section for linear access to an external flash memory.

The SMP8634 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 SMP8634 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 (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 (25 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-6 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.

PCI Master/Slave/Host Interface

6.3 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 PCI power management per the PCI Bus Power Management Specification Revision 1.2. Only D0 and D3 power states are supported; PME#'s are not supported. Up to four external PCI devices may be controlled by the SMP8634 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.

6.4 Features

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

6.5 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 internal serial flash memory. The boot code establishes the following nominal register values:

Table 6-1 **Default values for PCI register fields**

PCI Configuration Register Field	Default Value (Hex)
VendorID	0x1105
DeviceID	0x8634
Class Code	0x048000
RevisionID	0x01
Subsystem Vendor ID	0x1105
Subsystem Device ID	0x0000

The SMP8634 can function as the PCI host. To support this, it includes the PCI bus arbitration logic and four REQ#/GNT# pairs. In addition, it adds four IDSEL pins to support agent configuration cycle selection. These added pins and capabilities allow up to four external PCI masters and/or slave devices to be connected to the SMP8634 PCI bus without requiring an external host.

The following figures shows the general connection topology for the arbitration pins and IDSEL pins in the SMP8634:

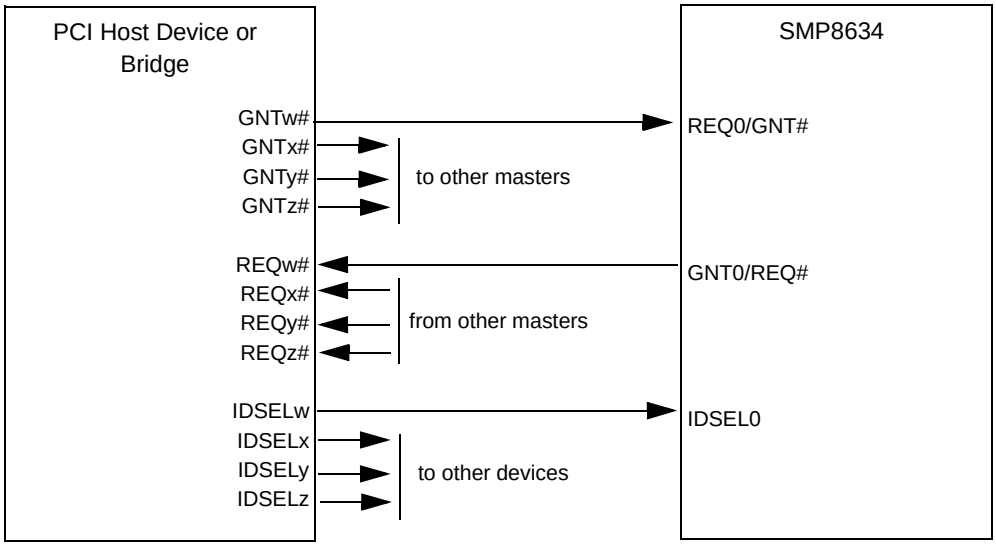


Figure 6-2 **PCI arbitration pin connection diagram (PCI device)**

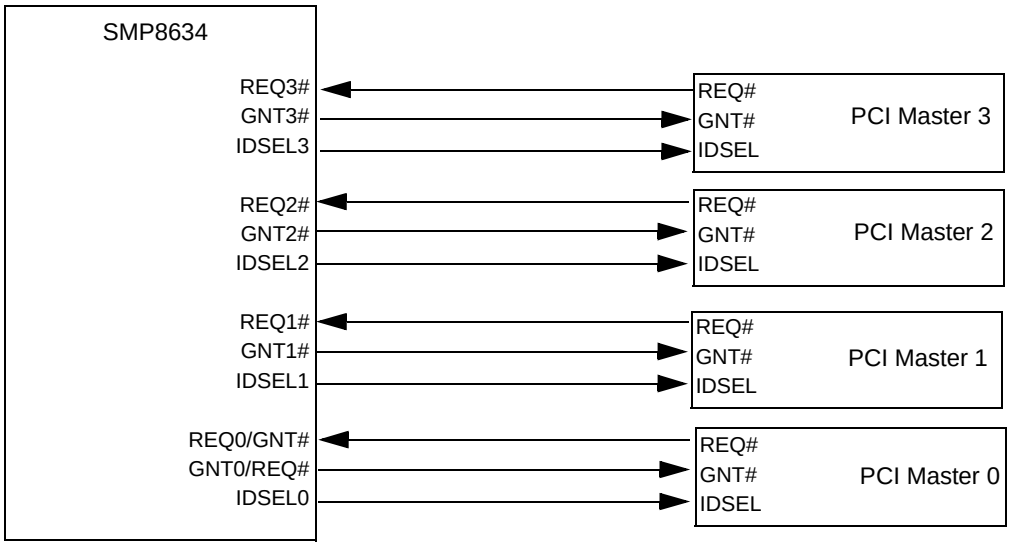


Figure 6-3 **PCI arbitration pin connection diagram (PCI host)**

6.5.1 PCI Slave Access

6.5.1.1 Direct Access

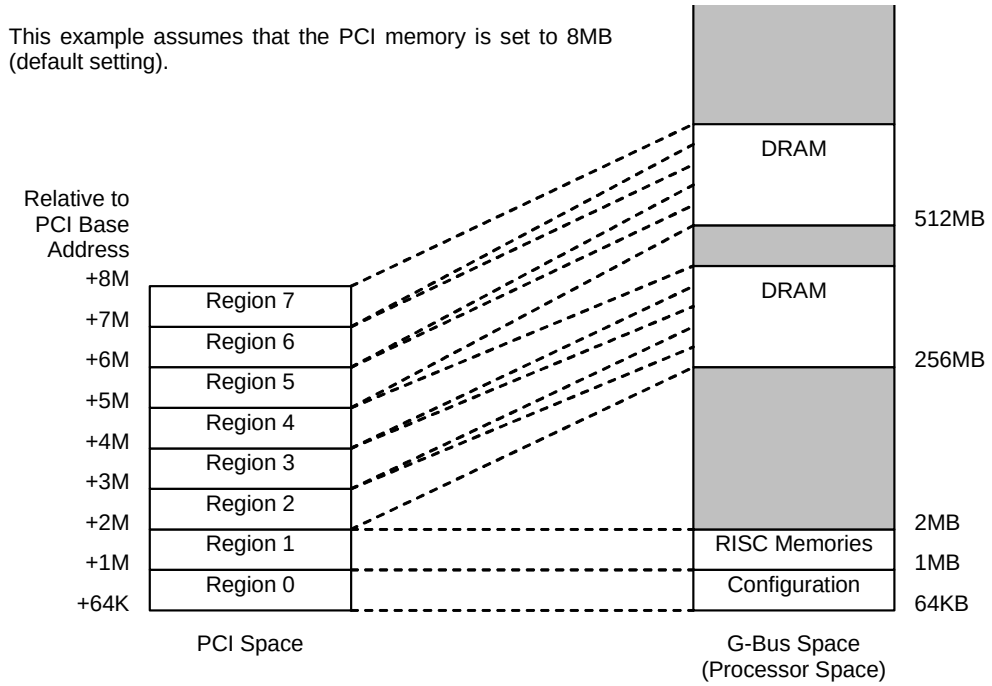


Figure 6-4 **PCI slave direct access (8MB default setting)**

When the SMP8634 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 SMP8634 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 SMP8634 is used as a host, and master-capable devices attached to the SMP8634 need access to particular regions of the SMP8634 memory.

6.5.1.2 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.

6.5.1.3 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.

6.5.1.4 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 internal 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 SMP8634. 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.

6.5.2 PCI Master Access

Transferring Data from the Host to the SMP8634 (Read Transaction)

To transfer a block of data from the host to the SMP8634, the switchbox and the M-Bus interface are programmed with the destination address and size in the SMP8634 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 SMP8634 to the Host (Write Transaction)

To transfer a block of data from the SMP8634 to the host, the switchbox and the M-Bus interface are programmed with the source address and size in the SMP8634 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.

6.5.3 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.

6.5.3.1 PCI Arbiter

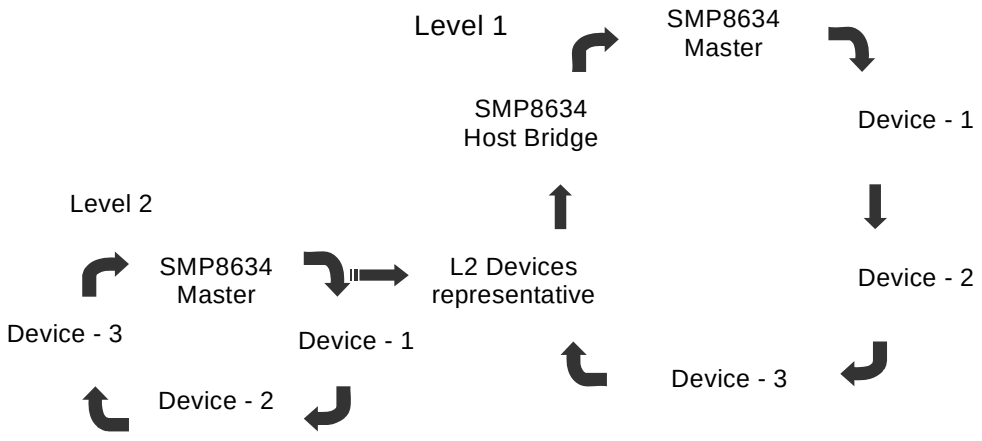


Figure 6-5 **PCI arbiter**

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, SMP8634 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.

6.5.3.2 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.

6.5.3.3 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].

6.5.3.4 Agent Detection

Upon power up to detect the PCI agents in the system the SMP8634 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 SMP8634 can only use 512MB on the PCI bus.

6.6 Register Map

6.6.1 PCI Slave Registers

6.6.1.1 PCI Slave Direct Access Registers

Table 6-2 **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
+9020	PCI_IRQ_STATUS	R/A	PCI Interrupt Status Register
+9024	PCI_IRQ_SET	R/W	PCI Interrupt Set Register
+9028	PCI_IRQ_CLEAR	R/W	PCI Interrupt Clear Register

Table 6-2 **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.

6.6.1.2 PCI Slave Time-out Control Registers

Table 6-3 **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.

6.6.1.3 PCI Slave Device Configuration Registers

Table 6-4 **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.

6.6.2 PCI Master Access Registers

Table 6-5 **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.

6.6.3 PCI Host Registers

Table 6-6 **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.

6.7 Pin Descriptions

6.7.1 PCI Pins

Table 6-7 **System bus pin descriptions**

Pin name	Ball ID	Direction	Description
PCI_AD0	AL19	B	PCI address/data pin 0 (LSB)
PCI_AD1	AN19	B	PCI address/data pin 1
PCI_AD10	AN17	B	PCI address/data pin 10
PCI_AD11	AL17	B	PCI address/data pin 11
PCI_AD12	AM17	B	PCI address/data pin 12
PCI_AD13	AL16	B	PCI address/data pin 13
PCI_AD14	AN16	B	PCI address/data pin 14
PCI_AD15	AK16	B	PCI address/data pin 15
PCI_AD16	AK15	B	PCI address/data pin 16
PCI_AD17	AP14	B	PCI address/data pin 17
PCI_AD18	AK14	B	PCI address/data pin 18
PCI_AD19	AP13	B	PCI address/data pin 19
PCI_AD2	AK19	B	PCI address/data pin 2
PCI_AD20	AL13	B	PCI address/data pin 20
PCI_AD21	AN13	B	PCI address/data pin 21
PCI_AD22	AM14	B	PCI address/data pin 22
PCI_AD23	AN12	B	PCI address/data pin 23
PCI_AD24	AL11	B	PCI address/data pin 24
PCI_AD25	AN11	B	PCI address/data pin 25
PCI_AD26	AM12	B	PCI address/data pin 26
PCI_AD27	AP11	B	PCI address/data pin 27
PCI_AD28	AL10	B	PCI address/data pin 28
PCI_AD29	AN10	B	PCI address/data pin 29
PCI_AD3	AM19	B	PCI address/data pin 3
PCI_AD30	AM11	B	PCI address/data pin 30
PCI_AD31	AP10	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.

Table 6-7 **System bus pin descriptions (Continued)**

Pin name	Ball ID	Direction	Description
PCI_AD4	AK17	B	PCI address/data pin 4
PCI_AD5	AP18	B	PCI address/data pin 5
PCI_AD6	AK18	B	PCI address/data pin 6
PCI_AD7	AN18	B	PCI address/data pin 7
PCI_AD8	AP17	B	PCI address/data pin 8
PCI_AD9	AM18	B	PCI address/data pin 9
PCI_CBE0#	AL18	B	Command/byte enable pin 0. Applies to PCI_AD(7:0).
PCI_CBE1#	AP16	B	Command/byte enable pin 1. Applies to PCI_AD(15:8).
PCI_CBE2#	AN14	B	Command/byte enable pin 2. Applies to PCI_AD(23:16).
PCI_CBE3#	AP12	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_CLK	AP8	I	Clock input for PCI interface section. Either 33MHz or a 66MHz clock signal.
PCI_DEVSEL#	AP15	B	Device select pin. A target asserts DEVSEL# when it decodes its address.
PCI_FRAME#	AL14	B	Cycle frame pin. Current initiator asserts the FRAME# pin to indicate the start and duration of a transaction.
PCI_GNTB0	AP9	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_GNTB1	AN9	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_GNTB2	AM10	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_GNTB3	AM9	O	PCI host mode: Bus grant for master #3. Asserted low to grant the PCI ownership to an external master. PCI device mode: Unused, make no connection.

Table 6-7 **System bus pin descriptions (Continued)**

Pin name	Ball ID	Direction	Description
PCI_IDSEL0	AM13	B	PCI host mode: ID select for PCI device #0. Asserted low by the SMP8634 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 SMP8634.
PCI_IDSEL1	AL12	O	PCI host mode: ID select for PCI device #1. Asserted low by the SMP8634 to indicate a configuration cycle to a PCI device. PCI Device Mode: Unused, make no connection.
PCI_IDSEL2	AK12	O	PCI host mode: ID Select for the PCI device #2. Asserted low by the SMP8634 to indicate a configuration cycle to a PCI device. PCI device mode: Unused, make no connection.
PCI_IDSEL3	AK13	O	PCI host mode: ID Select for the PCI device #3. Asserted low by the SMP8634 to indicate a configuration cycle to a PCI device. PCI device mode: Unused, make no connection.
PCI_INTA#	AN8	B	Interrupt A pin. Asserted by a PCI agent to request an interrupt.
PCI_IRDY#	AM15	B	Initiator ready pin. The current bus master asserts IRDY# to indicate that it is ready to complete a transaction.
PCI_PAR	AM16	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_REQ0#	AL9	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_REQ1#	AK9	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_REQ2#	AK10	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.
PCI_REQ3#	AK11	I	PCI host mode: Bus Request for master #3. Asserted low by external master requesting PCI bus transaction. PCI Device Mode: Unused, make no connection.
PCI_STOP#	AN15	O	Stop pin. Asserted by an addressed target to request the bus master to terminate the current transaction in progress.
PCI_TRDY#	AL15	B	Target ready pin. The currently addressed target asserts TRDY# to indicate that it is ready to complete a transaction.

6.8 Electrical Characteristics

6.8.1 PCI DC Characteristics

Table 6-8 **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

6.8.2 PCI AC Characteristics

The SMP8634 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 SMP8634 provide compliance with these specifications by design and characterization.

6.9 Timing Diagrams

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 6-9 **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	

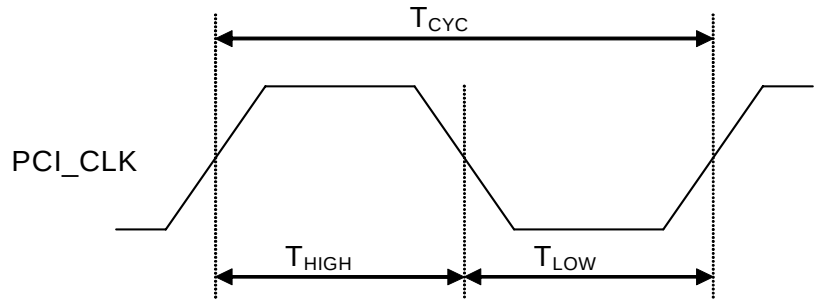


Figure 6-6 **PCI CLK timing diagram**

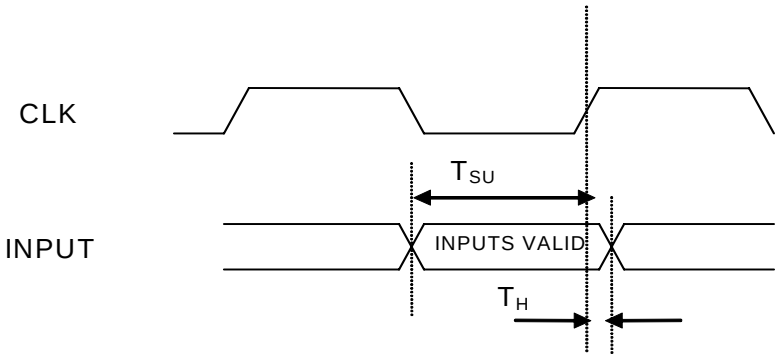


Figure 6-7 **PCI input timing diagram**

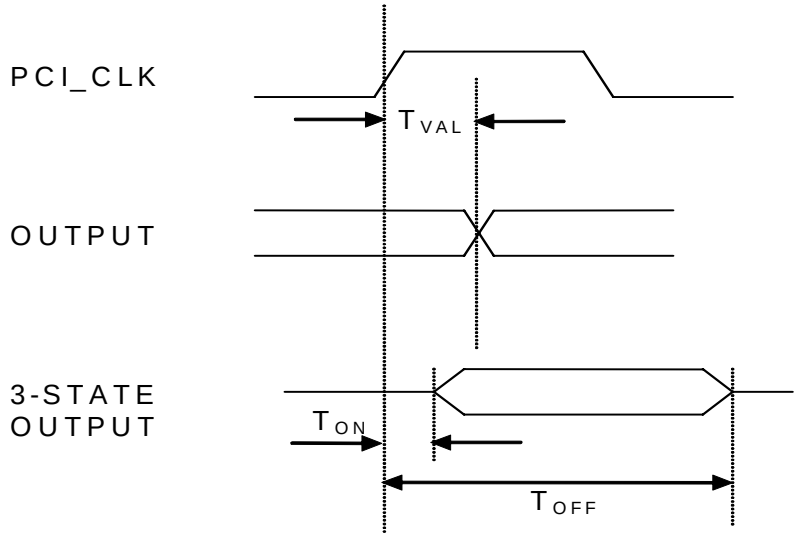


Figure 6-8 **PCI output timing diagram**

USB 2.0 Embedded Host Interface

The SMP8634 includes a USB 2.0 Embedded Host Controller and PHY. Two USB 2.0 Embedded Host ports are available.

Although a Host Controller is included on-chip, the SMP8634 is designed to be used in a closed system where the end-user cannot install additional software and drivers. Due to the wide variety of USB peripherals and applications, manufacturers must implement their own peripheral-specific drivers and application software to be able to use the USB interface. Each manufacturer is also required to have a Targeted Peripheral List (TPL) in their manuals that tells the end-users which peripherals are supported. Entries in a TPL may refer to specific products (manufacturer and model) or, may refer to a class of products (HDD, wheel mouse, flash card, etc.). To receive the USB compliance certification, manufacturers must submit their product with a fully function USB implementation.

6.10 Functional Description

6.10.1 USB 2.0 Embedded Host Controller

The USB 2.0 Embedded Host Controller is fully compliant with the USB 2.0 specification and the Enhanced Host Controller Interface (EHCI) specification revision 1.0. The controller supports high-speed (480Mbps) transfers. The controller comprehends the high-speed, full-speed and low-speed data ranges providing compatibility with a full range of USB devices. The full-speed and low-speed operations are supported via a USB 1.1 OHCI Host Companion Controller. The controller operates independently of the system bus of the SMP8634, shielding the complexities of the USB 2.0 Embedded Host Controller native protocol.

6.10.2 USB 2.0 Embedded Host Transceiver PHY

USB 2.0 Embedded Host interface includes a fully integrated PHY core supporting High Speed (HS), Full-Speed (FS) and Low-Speed Transceivers and which is compliant with the USB 2.0 specification. It includes clock/data recovery, on-chip PLL, integrated and calibrated termination and pull-up/down resistors with full analog transceiver functionality for the complete USB 2.0 PHY.

The USB 2.0 Embedded Host Transceiver transmits the data onto the line, receives the data and recovers the clock correctly from the receive data. The analog front end includes a high-speed differential driver and an envelope detection/disconnection differential receiver. The high-speed receiver primarily consists of three functional elements; a differential data receiver, a transmission envelope detector and a disconnection envelope detector.

The differential data receiver receives the high-speed (480Mbps) differential data from the USB and converts it to a single-ended NRZI. A transmission envelope detector monitors the amplitude of the high-speed differential data signal from the USB. A disconnection envelope detector also monitors the amplitude of the differential data signal from the USB, but with a different operation and different signal levels of interest.

A high-speed current driver is used for high-speed data transmission. When the driver is not transmitting, a standby mode allows the current to be directed internally to ground and places the driver in a high-output-impedance state. The transition between standby and non-standby modes does not exhibit much delay, and use of the stand-by/non-standby modes ensures that the driver meets the required accuracy starting with the first symbol of a packet.

6.11 Pin Description

6.11.1 USB 2.0 Pins

Table 6-10 **USB 2.0 pin descriptions**

Pin Name	Ball ID	Direction	Description
USB20_ATEST	AN23	B	USB 2.0 analog test I/O
USB20_DM_0	AP21	B	USB 2.0 line interface - port 0 -DM
USB20_DM_1	AP22	B	USB 2.0 line interface - port 1 -DM
USB20_DP_0	AN21	B	USB 2.0 line interface - port 0 -DP
USB20_DP_1	AN22	B	USB 2.0 line interface - port 1 -DP
USB20_REXT	AN20	B	USB 2.0 bias resistor input
USB20_VSST_0	AL20	I	USB 2.0 ground
USB20_VSSC	AL21	I	USB 2.0 ground
USB20_VDD33C	AL22	I	USB 2.0 analog power supply (3.3V)
USB20_VSST_1	AL23	I	USB 2.0 ground
USB20_VDD33T_0	AM20	I	USB 2.0 analog power supply (3.3V)

Table 6-10 **USB 2.0 pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
USB20_VDD33T_0	AM21	I	USB 2.0 analog power supply (3.3V)
USB20_VDD33T_0	AM22	I	USB 2.0 analog power supply (3.3V)
USB20_VDD33T_1	AM23	I	USB 2.0 analog power supply (3.3V)
USB20_VSST_0	AK20	I	USB 2.0 ground
USB20_VSST_0	AK21	I	USB 2.0 ground
USB20_VSST_1	AK22	I	USB 2.0 ground
USB20_VSST_1	AK23	I	USB 2.0 ground
USB20_XI	AP20	I	USB 2.0 Xtal oscillator input (optional)
USB20_XO	AP19	O	USB 2.0 Xtal oscillator output (optional)

6.12 Electrical Characteristics

6.12.1 USB 2.0 DC Characteristics – FS Operation

Table 6-11 **USB 2.0 DC characteristics – FS operation**

Parameter	Description	Conditions	Minimum	Typical	Maximum	Units
V_{DIFS}	Differential rcvr input sensitivity		0.2			V
V_{CMFS}	Differential rcvr common-mode voltage		0.8		2.5	V
V_{ILSE}	Single-ended rcvr low-level input voltage				0.8	V
V_{IHSE}	Single-ended rcvr high-level input voltage		2.0			V
V_{HYSSE}	Single-ended rcvr hysteresis		50		150	mV
V_{FSOL}	Low-level output voltage				0.3	V
V_{FSOH}	High-level output voltage		2.8		3.6	V
Z_{FSDRV}	Driver output impedance		40.5	45	49.5	ohms
V_{TERM}	DP pullup resistor termination voltage		3.0		3.6	V

6.12.2 USB 2.0 DC Characteristics – HS Operation

Table 6-12 **USB 2.0 DC characteristics – HS operation**

Parameter	Description	Conditions	Minimum	Typical	Maximum	Units
V_{DIHS}	Differential rcvr input sensitivity		100			mV
V_{CMHS}	Common-mode voltage range		-50		500	mV
V_{HSSQ}	Squelch detection threshold (diff)	Squelch threshold			100	mV
V_{HSSQ}	Single-ended rcvr high-level input voltage	Unsquelch threshold	150			mV
V_{HSOL}	Low-level output voltage	Ref. GND	-10		10	mV
V_{HSOH}	High-level output voltage	Ref. GND	360		440	mV
V_{OLHS}	Idle-level output voltage	Ref. GND	-10		10	mV
V_{CHIRPJ}	Chirp-J output voltage (diff)		700		1100	mV
V_{CHIRPK}	Chirp-K output voltage (diff)		-900		-500	mV
I_{LZ}	Off-state leakage current				+/-1	uA
C_{IN}	Transceiver input capacitance	Pin to GND		5	10	pF

6.12.3 Dynamic Characteristics – FS Operation

Table 6-13 **Dynamic Characteristics – FS Operation**

Parameter	Description	Conditions	Minimum	Typical	Maximum	Units
T_{FSR}	Rise time	CL= 50pF; 10 to 90% of |VOH – VOL|	4		20	nS
T_{FSF}	Fall time	CL= 50pF; 10 to 90% of |VOH – VOL|	4		20	nS
V_{CRS}	Output crossover voltage	Excluding first transition from IDLE	1.3		2.0	V

6.12.4 Dynamic Characteristics – HS Operation

Table 6-14 **Dynamic Characteristics – HS Operation**

Parameter	Description	Conditions	Minimum	Typical	Maximum	Units
T_{HSR}	Rise time		500			pS
T_{HSF}	Fall time		500			pS

Additional dynamic specifications:

1. The USB2.0 PHY output drivers conform to the eye pattern requirements as specified in Template 1 of the USB2.0 Revision 2.0 specification.
2. The USB2.0 PHY receivers conform to the eye pattern (including source and receiver jitter tolerance requirements) as specified in Template 4 of the USB2.0 Revision 2.0 specification.

Ethernet Interface

The SMP8634 includes an Ethernet 10/100 MAC (an external PHY chip is required). A single ethernet port is provided. The Ethernet MAC module allows the SMP8634 to communicate data over the Ethernet protocol (IEEE. 802.3). It consists of three modules: a DMA layer, a Transaction Layer Interface (TLI), and a Media Access Controller layer (MAC).

The Direct Memory Access (DMA) controller is the first layer of the Ethernet subsystem. It is responsible for exchanging the data between the TLI layer and the system memory. The SMP8634 controls the DMA operation through a set of registers.

The DMA is a generic 32-bit DMA for a variety of block data transfer applications. The data transfers are executed between the SMP8634 data buffers and a application interface The layer is optimized for packet-oriented DMA transfers with frame delimiters. It supports dual-buffer and linked descriptor chaining. The descriptor architecture allows for large blocks of data transfer with minimum CPU intervention. Each descriptor can transfer up to 2KB of data. The DMA layer provides comprehensive status reporting for normal operation and transfers with errors. A programmable burst transfer size provides optimal system bus utilization. The layer also supports start/stop modes of operation and separate VCI ports for host buffer, source, destination and CSR access. Single channel transmit and receive engines support simple round-robin arbitration between engines.

The 32-bit TLI block provides a bridge between the DMA controller and the Ethernet MAC. The layer is optimized for packet-oriented transfers. A programmable burst transfer size provides optimal system bus utilization. The TLI consists of two sets of FIFOs; a Transmit (TX) FIFO with dynamic threshold, and a Receive (RX) FIFO with a fixed threshold. The FIFOs support zero wait state bursts between internal FIFO buffers and the DMA/MAC. The receive FIFO sends only filtered packets to the DMA. The transmit FIFO supports store and forward mechanism. The frame data is held in the TLI FIFO until the MAC retransmits the packets with collision.

The MAC can operate in either 100Mbps or 10Mbps data transfer modes depending on the programming of the internal PHY. It handles all the requirements of the Ethernet protocol for both full-duplex and half-duplex modes. It supports CSMA/CD protocol for the half-duplex mode. Also supported in the half-duplex mode are, collision detection and auto retransmission on collisions. Flow control for the full-duplex operation and backpressure for the half-duplex operation are provided. The MAC contains a set of registers for controlling and checking the status of the transmissions. It supports Virtual Local Area Network (VLAN) operations. A variety of receive and transmit counters provide statistics management support. The 32-bit CRC generation and checking are done automatically.

There are options to insert PAD/CRC32 on transmit and to set automatic pad stripping in receive packets.

The MAC contains a variety of flexible address modes on the Ethernet side; one 48-bit perfect address, 64 hash-filtered multicast addresses, passes to all multicast addresses, promiscuous mode, inverse filtering passes to all incoming packets with status reports. It contains a programmable 32-bit, 16-bit or 8-bit wide data bus on the system bus.

6.13 Pin Description

6.13.1 MII Interface pins

Table 6-15 **MII interface pin descriptions**

Pin Name	Ball ID	Direction	Description
ETH_COL	AC3	B	Ethernet collision detect/GPIO14
ETH_CRS	AA3	B	Ethernet carrier sense/GPIO13
ETH_MDC	AD1	B	Ethernet management data clock/GPIO15
ETH_MDINT#	AA5	B	Ethernet management interrupt/GPIO17
ETH_MDIO	Y4	B	Ethernet management data I/O/GPIO16
ETH_RX_DV	AE1	B	Ethernet receive data valid /GPIO7
ETH_RX_ER	AF1	B	Ethernet receive error/GPIO8
ETH_RXCLK	AC2	B	Ethernet receive clock/GPIO6
ETH_RXD0	AC1	B	Ethernet receive data 0/GPIO9
ETH_RXD1	AB3	B	Ethernet receive data 1/GPIO10
ETH_RXD2	AB2	B	Ethernet receive data 2/GPIO11
ETH_RXD3	AB1	B	Ethernet receive data 3/GPIO12
ETH_TX_EN	AA2	B	Ethernet transmit enable/GPIO1
ETH_TXCLK	AA1	B	Ethernet transmit clock/GPIO0
ETH_TXD0	Y3	B	Ethernet transmit data 0/GPIO2
ETH_TXD1	Y2	B	Ethernet transmit data 1/GPIO3
ETH_TXD2	Y1	B	Ethernet transmit data 2/GPIO4
ETH_TXD3	W1	B	Ethernet transmit data 3/GPIO5

6.14 Electrical Characteristics

6.14.1 MII Interface DC Characteristics

Table 6-16 **MII 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	7.0	13	19
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.

6.14.2 MII Interface AC Electrical Characteristics

Table 6-17 **MII interface AC characteristics**

Symbol	Units	Min	Typ	Max
T_{RXCLK}	ns		40	
T_{RXDSU}	ns	3		
T_{RXDH}	ns	3		
T_{TXCLK}	ns		40	
T_{TXDVLD}	ns	0		8

6.15 Timing Diagram

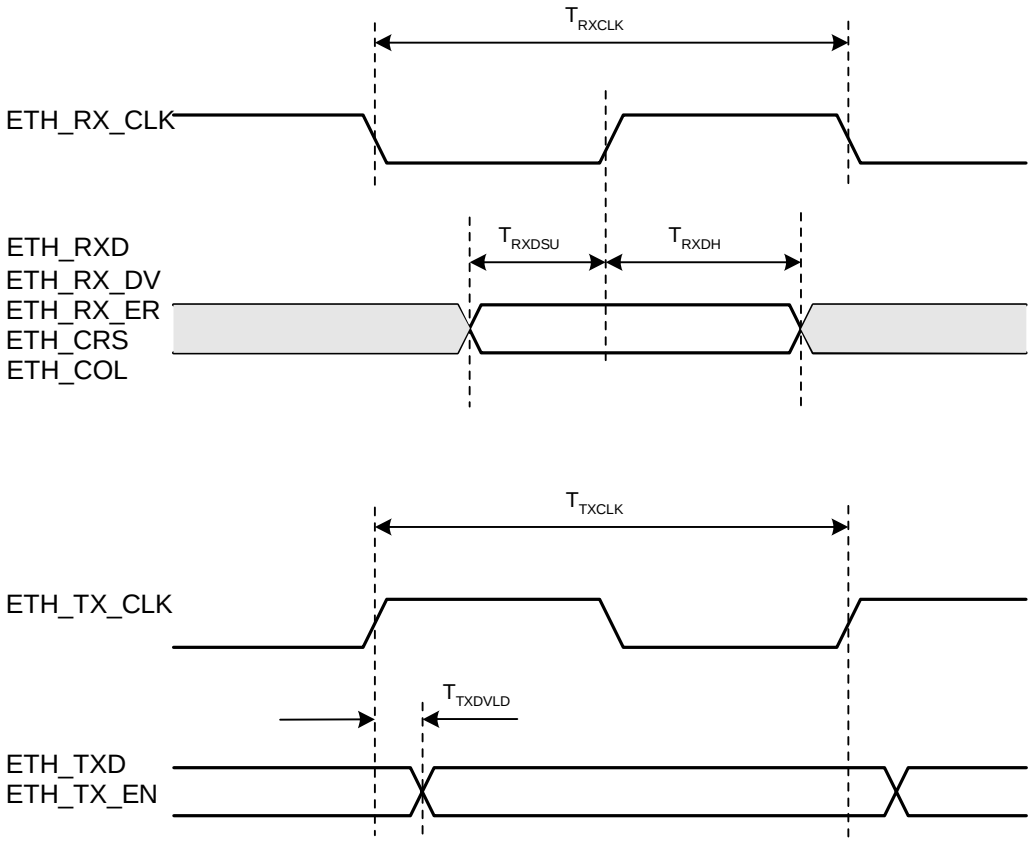


Figure 6-9 **MII Interface timing diagram**

Peripheral Bus Interface (PBI)

6.16 Introduction

The SMP8634 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 (i.e. CD-ROM, DVD drives)
- NOR-type parallel flash memory (up to 64MB per chip select)
- ISA compatible chips (MPEG2 encoder, etc.)

The PBI interface 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.

6.17 Features

- Supports a wide range of external devices like MPEG encoders, 802.11 chips and other external devices
- Supports up to four 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

6.18 Block Diagram

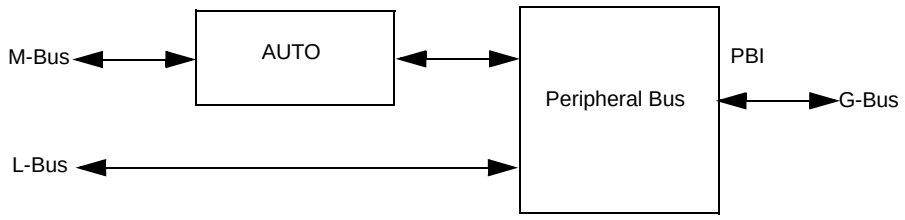


Figure 6-10 **Peripheral Bus Interface (PBI) block diagram**

6.19 Functional Description

The Peripheral Bus Interface (PBI) in the SMP8634 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 (MPEG encoder, etc.)

When being used as a general-purpose interface, the PBI can operate in either a separate address and data mode (25 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 SMP8634 provides a direct interface to these drives. 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 directly connected.

When used to connect external asynchronous memory such as NOR-type parallel flash memory, the PBI supports an 8-bit data bus width and 26-bit addressing (64MB per chip select), or a 16-bit data bus width and 25-bit addressing (64MB per chip select). Since there are total four chip select signals for the PBI, up to 256MB of NOR-type parallel flash memory many used. For applications requiring more flash memory, a device such as M System's 1GB NAND-type flash module with NOR-type interface may be used.

Note: Parallel flash device attached to store main CPU bootloader code must be a 16-bit device (not 8-bit) and must be connected to either chip select #2 or chip select #3.

Table 6-18 **Typical parallel flash memory requirements for various applications**

Networked DVD Player	Digital Media Adapter	Blu-ray Disc Player	HD-DVD Player
16MB	16MB	256MB (standard) 1GB (full-featured)	128MB (standard) 256MB (full-featured)

6.19.1 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#B[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 6-19 **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

Table 6-19 **PBI data transactions (Continued)**

G-Bus Transaction Type	PB Data Width = 8	PB Data Width = 16
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.

6.19.2 PBI Registers

The PBI 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 SMP8634 DRAM. Some applications using 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.

6.19.3 PBI Timing

6.19.3.1 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_CSx# 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.

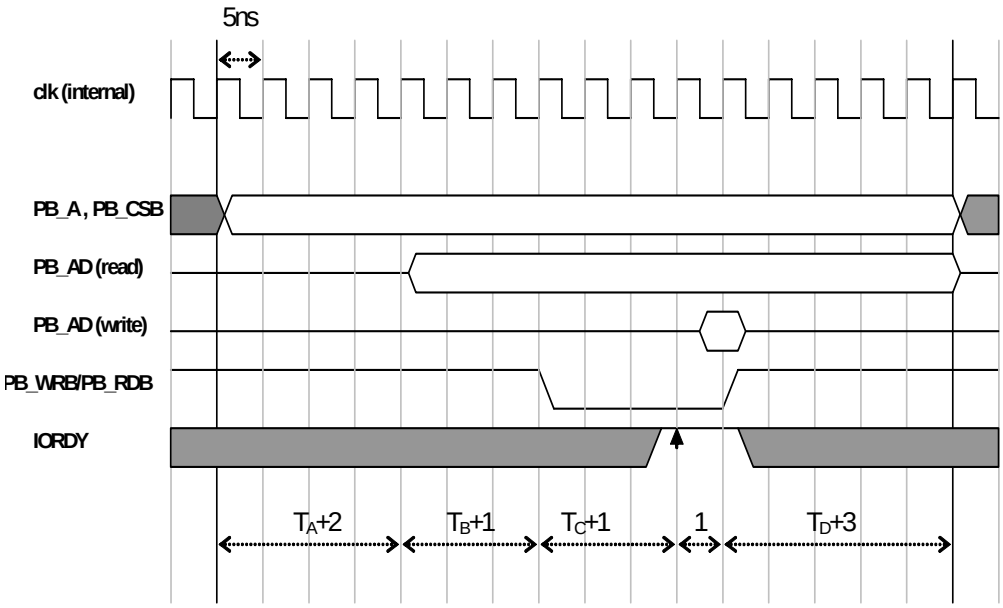


Figure 6-11 **Functional timing diagram: non-multiplexed modes (default parameters)**

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.

6.19.3.2 Multiplexed Mode

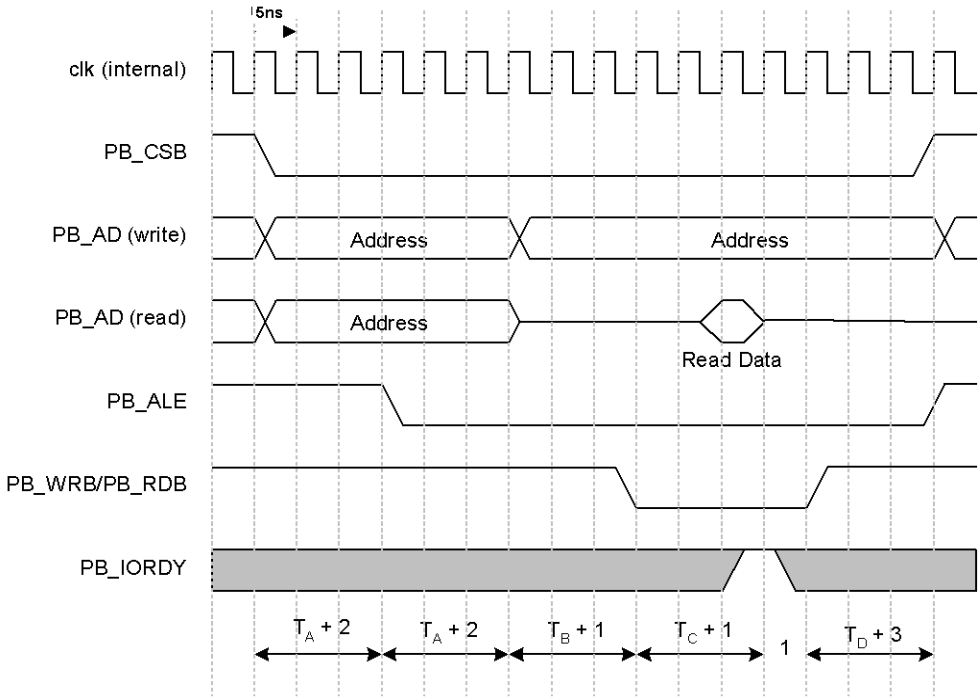


Figure 6-12 **Functional timing diagram: multiplexed modes**

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.

6.19.4 PBI Signals

The PBI supports four address/data modes:

- Mode 8/26: 8-bit data/26-bit address, non-multiplexed. Typically used with an 8-bit flash memory (up to 64MB)
- Mode 16/25: 16-bit data/25-bit address, non-multiplexed. Typically used with a 16-bit flash memory (up to 64MB) or IDE device (IDE devices require only three address bits)
- Mode 16/16mux: 16-bit data/16-bit address, multiplexed. Typically used with an external MPEG-2 encoder (up to 128KB).
- Mode 16/25mux: 16-bit data/25-bit address, multiplexed (up to 64MB).

Note: Parallel flash device attached to store main CPU bootloader code must be a 16-bit device (not 8-bit) and must be connected to either chip select #2 or chip select #3.

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 6-20 **PBI address/data pin descriptions**

Mode	8/26		16/25		16/16 mux		16/25 mux	
Signal	**Dir**	**Function**	**Dir**	**Function**	**Dir**	**Function**	**Dir**	**Function**
PB_A[24:18]	O	Not used	O	addr[24:18]	O	Not used	O	Not Used
PB_A[17:0]	O	addr[25:8]	O	addr[17:0]	O	Not used	O	addr [24:8][1]
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]

1. The signal PB_A17 is not used in this mode.

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

Table 6-21 **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

6.20 Register Map

6.20.1 PBI Configuration Registers

Table 6-22 **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
+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.

6.20.2 PBI Automode Registers

Table 6-23 **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.

6.20.3 IDE/Flash Interface Registers

Table 6-24 **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		
+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.

6.21 Pin Description

6.21.1 PBI Pins

Table 6-25 **PBI pin descriptions**

Pin Name	Ball ID	Direction	Description
PB_A0	AM32	O	PBI address bit 0 (LSB)
PB_A1	AN32	O	PBI address bit 1
PB_A10	AN30	O	PBI address bit 10
PB_A11	AP30	O	PBI address bit 11
PB_A12	AK29	O	PBI address bit 12
PB_A13	AL29	O	PBI address bit 13
PB_A14	AM29	O	PBI address bit 14
PB_A15	AN29	O	PBI address bit 15 (MSB). Provides address for non-multiplexed bus cycles.
PB_A16	AP29	O	PBI address bit 16
PB_A17	AL28	O	PBI address bit 17
PB_A18	AM28	O	PBI address bit 18
PB_A19	AN28	O	PBI address bit 19
PB_A2	AP32	O	PBI address bit 2
PB_A20	AP28	O	PBI address bit 20
PB_A21	AL27	O	PBI address bit 21
PB_A22	AM27	O	PBI address bit 22
PB_A23	AN27	O	PBI address bit 23
PB_A24	AP27	O	PBI address bit 24
PB_A3	AM31	O	PBI address bit 3
PB_A4	AN31	O	PBI address bit 4
PB_A5	AP31	O	PBI address bit 5
PB_A6	AJ30	O	PBI address bit 6
PB_A7	AK30	O	PBI address bit 7
PB_A8	AL30	O	PBI address bit 8
PB_A9	AM30	O	PBI address bit 9
PB_AD0	AK34	B	Multiplexed address/data bit 0 (LSB)
PB_AD1	AK33	B	Multiplexed address/data bit 1

Table 6-25 **PBI pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
PB_AD10	AH32	B	Multiplexed address/data bit 10
PB_AD11	AH31	B	Multiplexed address/data bit 11
PB_AD12	AJ34	B	Multiplexed address/data bit 12
PB_AD13	AJ33	B	Multiplexed address/data bit 13
PB_AD14	AJ32	B	Multiplexed address/data bit 14
PB_AD15	AJ31	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_AD2	AK32	B	Multiplexed address/data bit 2
PB_AD3	AK31	B	Multiplexed address/data bit 3
PB_AD4	AL34	B	Multiplexed address/data bit 4
PB_AD5	AL33	B	Multiplexed address/data bit 5
PB_AD6	AL32	B	Multiplexed address/data bit 6
PB_AD7	AL31	B	Multiplexed address/data bit 7
PB_AD8	AH34	B	Multiplexed address/data bit 8
PB_AD9	AH33	B	Multiplexed address/data bit 9
PB_ALE	AG30	O	Address latch enable
PB_CS0#	AG34	O	Chip select #0 (active low)
PB_CS1#	AG33	O	Chip select #1 (active low)
PB_CS2#	AG32	O	Chip select #2 (active low)
PB_CS3#	AG31	O	Chip select #3 (active low). Pins PB_CSB3:0 become active for accesses to specified regions of the PB memory space.
PB_DIR#	AH30	O	Transfer direction (low = read, high = write)
PB_DMACK#	AM33	O	DMA acknowledge
PB_DMARQ	AN33	I	DMA request
PB_IORDY	AP33	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#	AM34	O	Read command (active low)
PB_WR#	AN34	O	Write command (active low)

6.22 Electrical Characteristics

6.22.1 PBI AC Characteristics

Table 6-26 **PBI AC characteristics**

Symbol	Description	Units	Minimum	Maximum
T_{RDSU}	Read data setup time to PB_RD# deasserted	ns	$t_{SYSCLK}+8$	
T_{RDH}	Read data hold time from PB_RD# deasserted	ns	0	
T_{RDYSU}	IORDY setup time to RD#/WR# deasserted	ns	$t_{SYSCLK}+4$	
T_{RDYH}	IORDY hold time from RD#/WR# deasserted	ns	$-(t_{SYSCLK}-4)$	
T_1	Internal clock to earliest output transition		Note[1]	Note[1]
T_2	Internal clock to latest output transition		Note[1]	Note[1]
T_{OPSKEW}	Skew range for synchronous output signals	ns	0	4

1. No specifications supplied since parameter not externally measurable. Parameters used to explain T_{OPSKEW}.

6.23 Timing Diagrams

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 (200MHz 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.

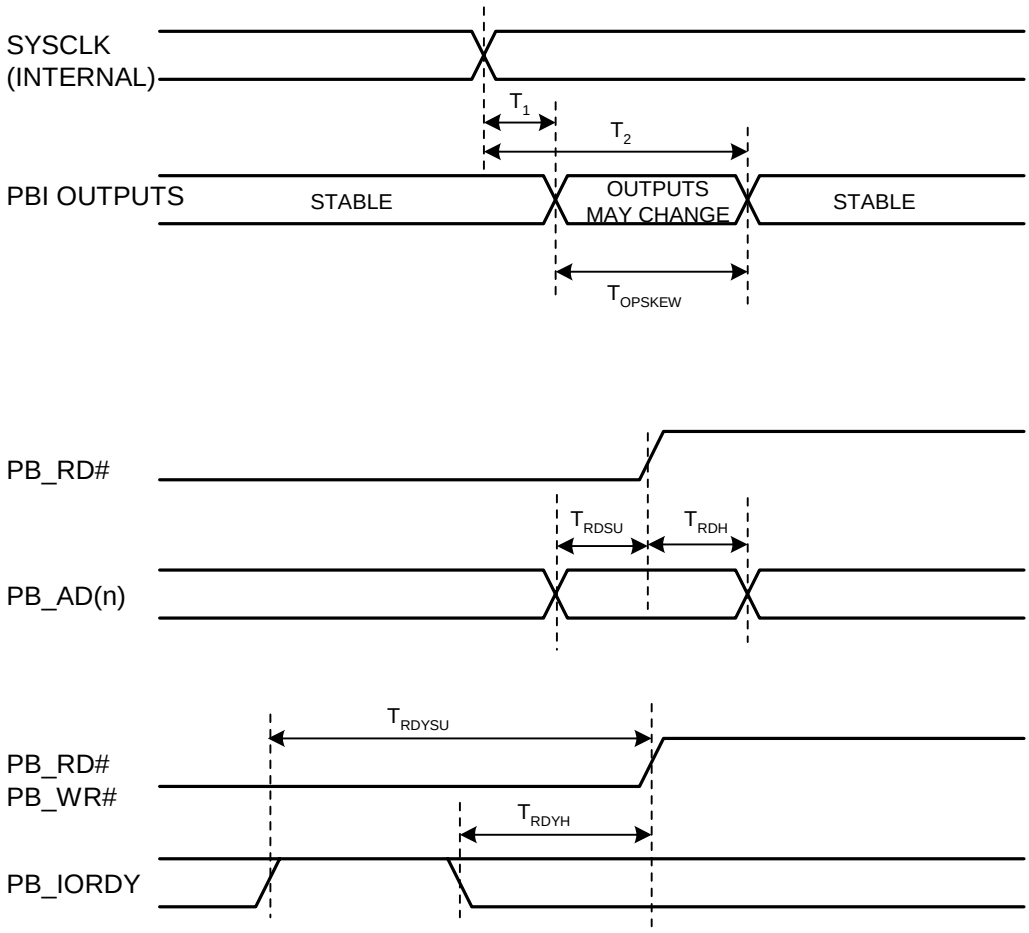


Figure 6-13 **PBI input timing diagram**

IDE/ATAPI-6 Interface

6.24 Introduction

The IDE/ATAPI-6 interface can support the ATA/ATAPI-6 device attachments. Two devices (master and slave) may be connected but only one can be used at any given time. Both the PIO and the DMA transfer modes are supported. The IDE interface allows CD, DVD, BD, HD-DVD or hard disk drives to be directly attached.

6.25 Features

- Dedicated IDE/ATAPI-6 interface
- Supports ATA/ATAPI-6 device attachments
- Supports transfer rates of up to 100MB/sec (UltraDMA mode 5)
- Supports both PIO and DMA transfer modes
- Allows direct attachment of CD, DVD, BD, HD-DVD or hard disk drives

6.26 Block Diagram

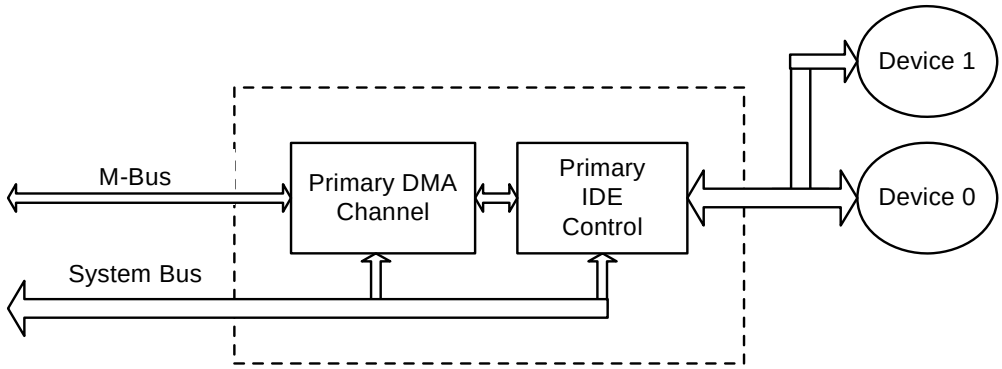


Figure 6-14 **IDE/ATAPI-6 interface block diagram**

6.27 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 SMP8634 provides a direct interface to these drives.

The SMP8634 IDE controller supports the ATAPI-6 specification, also known as UltraDMA/100. A conformance to this specification requires support for the following transfer modes and command sets:

- PIO modes: Modes 0, 1, 2, 3 and 4
- Multi word DMA modes: Modes 0, 1 and 2
- UltraDMA modes: Modes 2, 3, 4 and 5
- ATAPI command set

With ATAPI-6 compatibility, the SMP8634 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 ATAPI-6 interface is a standard addition to consumer-level DVD drives. The SMP8634 provides a direct interface to these IDE/ATAPI-6 drives. The ATA drives are memory-mapped devices.

6.27.1 IDE Interface

The pre-fetch and post write sub-module controls the data read/write of the external ATA/ATAPI 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.

6.27.2 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).

6.27.3 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/ATAPI 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.

6.28 Register Maps

6.28.1 IDE/ATAPI-6 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 6-27 **IDE/ATAPI-6 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.

6.28.2 IDE/ATAPI-6 Device Registers

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

Table 6-28 **IDE/ATAPI-6 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 Registers

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

6.28.3 IDE/ATAPI-6 DMA Channel Registers

Table 6-29 **IDE/ATAPI-6 DMA channel registers**

Address[1]	Register Name	R/W/A[2]	Description
+0900	BMIC	R/W	Bus Master IDE Command Register
+0904	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.

6.29 Pin Description

6.29.1 IDE/ATAPI-6 Pins

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

Table 6-30 **IDE/ATAPI-6 interface pin descriptions**

Pin Name	Ball ID	Direction	Description
IDE_CS0#	D22	O	Chip select #0
IDE_CS1#	E22	O	Chip select #1
IDE_DACK#	C20	O	DMA acknowledge
IDE_DMARQ	D20	I	DMA request
IDE_A0	B21	O	IDE bus address bit 0
IDE_A1	E21	O	IDE bus address bit 1
IDE_A2	C21	O	IDE bus address bit 2
IDE_D0	C19	B	IDE data bit 0 (LSB)
IDE_D1	A19	B	IDE data bit 1
IDE_D10	B17	B	IDE data bit 10
IDE_D11	D18	B	IDE data bit 11
IDE_D12	A18	B	IDE data bit 12
IDE_D13	C18	B	IDE data bit 13
IDE_D14	E19	B	IDE data bit 14
IDE_D15	B19	B	IDE data bit 15 (MSB)
IDE_D2	D19	B	IDE data bit 2
IDE_D3	B18	B	IDE data bit 3
IDE_D4	E18	B	IDE data bit 4
IDE_D5	C17	B	IDE data bit 5
IDE_D6	A17	B	IDE data bit 6
IDE_D7	D17	B	IDE data bit 7
IDE_D8	C16	B	IDE data bit 8
IDE_D9	E17	B	IDE data bit 9
IDE_IOR#	A20	O	Read command (active-low)
IDE_IOW#	E20	O	Write command (active-low)

Table 6-30 **IDE/ATAPI-6 interface pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
IDE_INTRQ	D21	I	IDE device interrupt request
IDE_IORDY	B20	I	IDE cycle extension input
IDE_NPCBLID	A21	I	Cable ID input. Used to detect the type of cable assembly in use.

6.30 Electrical Characteristics

6.30.1 IDE/ATAPI-6 Interface DC Characteristics

Table 6-31 **IDE/ATAPI-6 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}[a]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
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 IDE_D(15:0) when driving. Also applies to IDE_CS(1:0)#, IDE_DACK#, IDE_A(2:0), IDE_IOR#, and IDE_IOW#.
2. Parameter applies to IDE_D(15:0) when receiving. Also applies to IDE_DMARQ, IDE_INTRQ, IDE_IORDY, and IDE_NPCBLID.

6.30.1 IDE/ATAPI-6 Interface AC Characteristics

The SMP8634 contains an ATA/ATAPI (IDE) interface capable of supporting the following modes:

- PIO operation; modes 0 through 4
- Multi-word DMA operation; modes 0 through 2
- Ultra DMA operation; modes 0 through 5

The AC characteristics for the supported modes are described in this section. Many of the settings are dependent on values programmed into various configuration registers as well as the system clock frequency of the SMP8634. The values given in the tables in this section assume a system clock frequency of 200MHz.

Due to the number of waveform diagrams which are needed to depict all of the parameters for the various modes of operation, the reader is instead referred to the official specifications which define the ATA/IDE modes. The tables contained herein use the same terminology for the various parameters. The specification document which the SMP8634 ATA/IDE interface reflects is: INCITS 361:2002 “Information Technology - AT Attachment with Packet Interface-6 (ATA/ATAPI-6)”. This document is available from the International Committee for Information Technology Standards (www.incits.org).

6.30.1.1 PIO Mode Characteristics

Table 6-32 **PIO mode characteristics**

Parameter	Description	Mode	Minimum	Typical	Maximum	Units
T_0	Cycle time[1]	0	600			ns
		1	385			ns
		2	240			ns
		3	180			ns
		4	120			ns
T_1	Address valid to RD/WR Command assertion[2]	0	70			ns
		1	50			ns
		2	30			ns
		3	30			ns
		4	25			ns
$T_{2\text{-}8B}$	RD/WR Command asserted pulse width (8-bit)[3]	0	290			ns
		1	290			ns
		2	290			ns
		3	80			ns
		4	70			ns
$T_{2\text{-}16B}$	RD/WR Command asserted pulse width (16-bit)[3]	0	165			ns
		1	125			ns
		2	100			ns
		3	80			ns
		4	70			ns

Table 6-32 **PIO mode characteristics (Continued)**

Parameter	Description	Mode	Minimum	Typical	Maximum	Units
T_{2i}	RD/WR Command deasserted interval[4]	0	-			ns
		1	-			ns
		2	-			ns
		3	70			ns
		4	25			ns
T_3	Data setup time to WR Command deasserted	0-4	T_2			ns
T_4	WR Command deasserted to data invalid[5]	0	30			ns
		1	20			ns
		2	15			ns
		3	10			ns
		4	10			ns
T_5	Data valid to RD Command deasserted[6]	0	50			ns
		1	35			ns
		2	20			ns
		3	20			ns
		4	20			ns
T_6	RD Command deasserted to data invalid[6]	0-4	5			ns
T_{6Z}	RD Command deasserted to output data hi-Z[6]	0-4			30	ns
T_9	RD/WR Command deasserted to address invalid	0-4	T_{2i}			ns
T_{RD}	Read data valid to IORDY active[6]	0-4	0			ns
T_A	IORDY setup[6]	0-4			35	ns
T_B	IORDY pulse width[6]	0-4			1250	ns
T_C	IORDY assertion to release[6]	0-4			5	ns

1. Programmed as 1 – 128 system clock periods (PRI_DRVxTIM register).
2. Programmed as 1 – 16 system clock periods (PRI_DRVxTIM register).
3. Programmed as 1 – 64 system clock periods (PRI_DRVxTIM register).
4. Programmed as 1 – 64 system clock periods (PRI_DRVxTIM register).
5. Programmed as 1 –8 system clock periods (PRI_DRVxTIM register).
6. Device-controlled parameters.

6.30.1.2 Multi-word DMA Mode Characteristics

Table 6-33 **Multi-word DMA mode characteristics**

Parameter	Description	Mode	Minimum	Typical	Maximum	Units
T_0	Cycle time[1]	0	480			ns
		1	150			ns
		2	120			ns
T_D	RD/WR Command asserted pulse width[2]	0	215			ns
		1	80			ns
		2	70			ns
T_E	RD Command asserted to data valid[3]	0			195	ns
		1			60	ns
		2			50	ns
T_F	RD Command deasserted to data invalid[3]	0	5			ns
		1	5			ns
		2	5			ns
T_G	Data setup time to WR Command deasserted	0	185			ns
		1	50			ns
		2	40			ns
T_H	WR Command deasserted to data invalid[4]	0	20			ns
		1	15			ns
		2	10			ns
T_{IR}	DMACK to RD Command assertion delay	0-2	T_{KR}			ns
T_{IW}	DMACK to WR Command assertion delay	0-2	T_{KW}			ns
T_J	Command deassertion to DMACK delay	0	25			ns
T_{KR}	RD Command deasserted interval[5]	0	50			ns
		1	50			ns
		2	25			ns
T_{KW}	WR Command deasserted interval[5]	0	215			ns
		1	50			ns
		2	25			ns

Table 6-33 **Multi-word DMA mode characteristics (Continued)**

Parameter	Description	Mode	Minimum	Typical	Maximum	Units
T_{LR}	RD Command asserted to DMAREQ deasserted[3]	0			120	ns
		1			45	ns
		2			35	ns
T_{LW}	WR Command asserted to DMAREQ deasserted[3]	0			40	ns
		1			40	ns
		2			35	ns
T_{MR}	CSx assertion to RD Command assertion	0-2	T_{KR}			ns
T_{MW}	CSx assertion to WR Command assertion	0-2	T_{KR}			ns
T_N	RD/WR Command deassertion to CSx deassertion	0	15			ns
		1	10			ns
		2	10			ns
T_Z	DMACK to read data hi-Z[3]	0	20			ns
		1	25			ns
		2	25			ns

1. Programmed as 1–128 system clock periods (PRI_DRVxTIM register).
2. Programmed as –64 system clock periods (PRI_DRVxTIM register).
3. Device-controlled parameters.
4. Programmed as 1–8 system clock periods (PRI_DRVxTIM register).
5. Programmed as 1–64 system clock periods (PRI_DRVxTIM register).

6.30.1.3 Ultra DMA Mode Characteristics

Table 6-34 **Ultra DMA mode characteristics**

Parameter	Description/Comments	Mode	Minimum	Maximum	Units
$T_{2CYCTYP}$	Sustained average two cycle time	0	240		ns
		1	160		ns
		2	120		ns
		3	90		ns
		4	60		ns
		5	40		ns
T_{CYC}	Cycle time, strobe edge to strobe edge	0	120		ns
		1	80		ns
		2	60		ns
		3	45		ns
		4	30		ns
		5	20		ns
T_{2CYC}	Cycle time, strobe edge to strobe edge	0	240		ns
		1	160		ns
		2	120		ns
		3	90		ns
		4	60		ns
		5	40		ns
T_{DS}	Data valid before strobe edge at recipient	0-5	4		ns
T_{DH}	Data invalid after strobe edge at recipient	0-5	4		ns
T_{DVS}	Data valid before strobe edge at sender	0-5	T_{CYC-10}		ns
T_{DVH}	Data invalid after strobe edge at sender	0-4	10		ns
		5	5		ns

Table 6-34 **Ultra DMA mode characteristics (Continued)**

Parameter	Description/Comments	Mode	Minimum	Maximum	Units
T_{CVS}	CRC word valid setup time at host (relative to DMACK negation)	0	70		ns
		1	50		ns
		2	35		ns
		3	20		ns
		4	10		ns
		5	10		ns
T_{CVH}	CRC word valid hold time at sender (relative to DMACK negation)	0	10		ns
		1	10		ns
		2	10		ns
		3	10		ns
		4	10		ns
		5	10		ns
T_{ZFS}	Time from STROBE output released-to-driving until the first transition of critical timing	0-5	Device-controlled parameter		
T_{DZFS}	Time from data output released-to-driving until the first transition of critical timing	0	120		ns
		1	80		ns
		2	60		ns
		3	45		ns
		4	30		ns
		5	25		ns
T_{FS}	First strobe time	0-5	Device-controlled parameter		
T_{LI}	Limited interlock time	0-5	2	3	T_{SYSCLK}
T_{MLI}	Interlock time with minimum	0-5	20		ns
T_{UI}	Unlimited interlock time	0-5	2		T_{SYSCLK}
T_{AZ}	Outputs released to hi-Z	0-5		10	ns
T_{ZAH}	Minimum delay time required for output drivers to assert or negate from released (host)	0-5	20		ns

Table 6-34 **Ultra DMA mode characteristics (Continued)**

Parameter	Description/Comments	Mode	Minimum	Maximum	Units
T_{ZAD}	Minimum delay time required for output drivers to assert or negate from released (device)	0-5	Device-controlled parameter		
T_{ENV}	Envelope time [DMACK to STOP and DMACK to HDMARDY- during in-burst initiation and from DMACK to STOP during data out burst initiation]	0	20	70	ns
		1	20	70	ns
		2	20	70	ns
		3	20	55	ns
		4	20	55	ns
		5	20	50	ns
T_{RFS}	Ready-to-final-STROBE time	0-5		15	ns
T_{RP}	Ready-to-pause time	0	160		ns
		1	125		ns
		2	100		ns
		3	100		ns
		4	100		ns
		5	85		ns
T_{IORDYZ}	Maximum time to release IORDY	0-5	Device-controlled parameter		
T_{ZIORDY}	Minimum time before driving IORDY	0-5	Device-controlled parameter		
T_{ACK}	Setup and hold time for DMACK (before assertion or negation)	0-5	20		ns
T_{SS}	STROBE edge to negation of DMARQ or assertion of STOP (when sender terminates a burst)	0-5	50		ns

7 Video Decoder Subsystem

7.1 Block Diagram of Video Decoder Subsystem

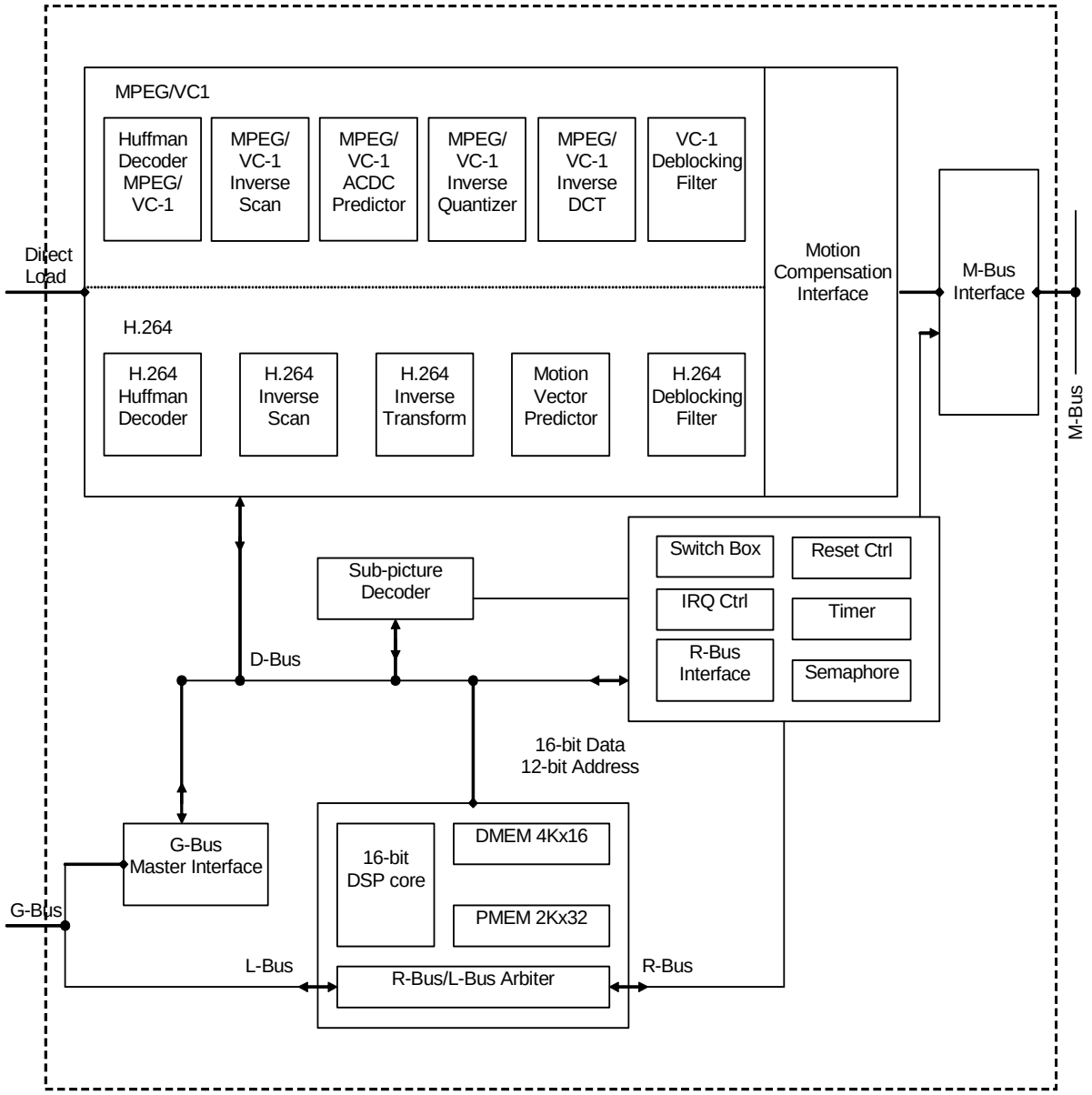


Figure 7-1 **Video decoder subsystem block diagram**

7.2 Introduction

The SMP8634 contains two identical video decoders each capable of decoding HD MPEG-4.10 (H.264), HD SMPTE, SMPTE 421M (VC-1), HD WMV9, HD MPEG-2 and MPEG-1. The video decoder subsystem executes the video decoding algorithms supported by the SMP8634. Its architecture is a hybrid of both processor-based and hardwired logic approaches. The proprietary RISC CPU runs at 300MHz, and the hardware-based data path processing runs at 200MHz.

The SMP8634 supports dual HD decoding, and so is capable of decoding, for example, two MPEG-2 streams, two MPEG-4.10 (H.264) streams, or one MPEG-4.10 (H.264) stream and one SMPTE 421M (VC-1) stream. The maximum number of simultaneous programs that can be decoded and displayed depends on the source format and resolution. The video decoder subsystem can also support the video decoding requirements for MSTV IPTV, BD, HD-DVD, FVD, DVD, SVCD, VCD, ARIB, ATSC, DMB, DVB and OpenCable applications.

Table 7-1 **Number of programs that can be decoded simultaneously by SMP8634**

	MPEG-2	MPEG-4.2	WMV9	SMPTE 421M (VC-1)		MPEG-4.10 (H.264)	
Max profile and level	MP@HL	ASP@L5	MP@HL	MP@HL	AP@L3	BP@L3	MP@L4.0[1] HP@L4.0[1]
1080i30	2		N/A	N/A	2	N/A	2
1080p25	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	4	4	N/A	N/A	4	4	4

1. L4.1 for BD and HD-DVD applications.

The following minimum and maximum resolutions are supported by each video decoder. The source resolution information is extracted from the AVI/ASF bitstream or from the video elementary bitstream.

Table 7-2 **Resolutions supported by each video decoder**

	Minimum	Maximum
MPEG-2	16x32	4096x4096
MPEG-4.2	64x64	4096x4096
MPEG-4.10 (H.264)	64x64	4096x4096
SMPTE 421M (VC-1)	64x64	2048x2048

Although each video decoder is capable of these maximum resolutions, a maximum resolution of 1920x1080 is more realistic due to the many other functions required to be supported in a typical design.

Common source program resolutions and frame refresh rates include,

- 480, 544 and 640 x 480i @ 25, 29.97 and 30Hz
- 480, 544 and 640 x 480p @ 23.976, 24, 29.97, 30, 59.94 and 60Hz
- 704 and 720 x 480i @ 29.97 and 30Hz
- 704 and 720 x 480p @ 23.976, 24, 29.97, 30, 59.94 and 60Hz
- 704 and 720 x 576i @ 25Hz
- 704 and 720 x 576p @ 23.976, 24, 25 and 50Hz
- 1280 x 720p @ 23.976, 24, 25, 29.97, 30, 50, 59.94 and 60Hz
- 960, 1280, 1440 and 1920 x 1080i @ 25, 29.97 and 30Hz
- 960, 1280, 1440 and 1920 x 1080p @ 23.976, 24, 25, 29.97 and 30Hz

Each of the 2 video decoder engines 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 SMP8634. 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 CPU 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 video codecs is addressed by the ‘Direct Load Port’ feature of the engine. This port provides a dedicated, prioritized, high-bandwidth port to the system memory.

7.3 Features

- MPEG-1
- MPEG-2 MP@HL
- MPEG-4.2 ASP@L5
- WMV9 MP@HL
- SMPTE 421M (VC-1) MP@HL and AP@L3
- MPEG-4.10 (H.264) MP@L4.0 and HP@L4.0 up to 1920x1080p25, 1920x1080i30 or 1280x720p60 resolution. L4.1 for BD and HD-DVD applications.
- Microsoft MPEG-4 v3 for maximum DivX compatibility
- Hardware accelerated Baseline JPEG decoding
- In-loop deblocking filter for MPEG-4.10 (H.264) and SMPTE 421M (VC-1)
- Error concealment for MPEG-2 and MPEG-4.2
- 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/SMPTE 421M (VC-1) SDTV (HDTV): 20 (40) Mbps maximum

7.4 Functional Description

7.4.1 MPEG-2 Decoding

Features and capabilities of MPEG-2 decoding include,

- MP@HL up to 1920x1080p25, 1920x1080i30 or 1280x720p60 resolution
- Supports low delay mode (no B pictures, no frame reordering delay, handling of VBV underflow)

An error detected during the texture decoding triggers the resynchronization to the next start code. The area corresponding to the macroblocks skipped (not decoded) during this resynchronization process keeps the content of the previously decoded picture in that buffer. Once an error is detected at the texture level, the decoded picture is marked as corrupted. The application decides if the corrupted pictures are displayed or not by setting the DisplayErrorThreshold (the default value is zero which means that only good pictures are displayed).

7.4.2 MPEG-4.2 Decoding

Features and capabilities of MPEG-4.2 decoding include,

- 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)

7.4.3 MPEG-4.10 (H.264) Decoding

Features and capabilities of MPEG-4.10 (H.264) decoding include,

- BP@L3 up to 720x480p30 or 720x576p25 resolution, including FMO and ASO
- MP@L4.0 and HP@L4.0 up to 1920x1080p25, 1920x1080i30 or 1280x720p60 resolution. L4.1 for BD and HD-DVD applications.
- The High 10, High 4:2:2, High 4:4:4 and Extended profiles are not supported
- Decoding can start at any recovery point
- Handling of SPS and PPS between slices of a same picture
- DPB output delay information is used to output pictures as soon as possible (reduces latency)
- Almost all display meta data from the elementary video stream is output, including. but not limited to. aspect ratio, pan-scan, chromaticity, picture structure and cropping information
- Key-frame (aka I-frame) playback supported
- Supports up to 180 Mbin/s for CABAC streams
- Supports up to 135 slices per frame
- An error detected during the texture decoding triggers:
    - The discarding of the current picture

- The display of the remaining anchor(s) and
- The resynchronization to the next intra-picture

7.4.3.1 SEI Messages and VUI

Table 7-3 **Supported MPEG-4.10 (H.264) SEI (Supplemental enhancement information) messages and VUI (video usability information)**

	Supported	Not Supported at this Time	Passed to Application
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		

7.4.4 SMPTE 421M (VC-1) Decoding

Features and capabilities of SMPTE 421M (VC-1) decoding include,

- MP@HL up to 1280x720p60 (<90% of the P-picture macroblocks have 4 motion vectors) or 1920x1080p25 resolution. Up to 1280x720p30 or 1920x1080p25 resolution if unconstrained 4MV bitstream.
- AP@L3 up to 1920x1080i30 (<60% of the B-picture macroblocks have 4 motion vectors), 1920x1080p25 or 1280x720p60 resolution. Up to 1920x1080i25, 1920x1080p25 or 1280x720p60 resolution if unconstrained 4MV bitstream.
- Almost all display metadata from the elementary video stream is output, including. but not limited to. aspect ratio, pan-scan, chromaticity, picture structure and cropping information
- Key-frame (aka I-frame) playback supported
- Pan-scan playback supported
- An error detected during the texture decoding triggers:
  - The discarding of the current picture
  - The display of the remaining anchor(s) and,
  - The resynchronization to the next intra-picture

7.4.5 WMV9 Decoding

Features and capabilities of WMV9 decoding include,

- MP@HL up to 1280x720p60 (<90% of the P-picture macroblocks have 4 motion vectors) or 1920x1080p25 resolution. Up to 1280x720p30 or 1920x1080p25 resolution if unconstrained 4MV bitstream.
- Does not support WMV7 or WMV8 formats
- Does not support Screen, Image and Image Version 2 profiles

7.4.6 H.263 Decoding

Decoding of MPEG-4.2 ‘short_video_header’ bitstreams, which are technically identical to baseline H.263 bitstreams, is supported. Overlapped motion compensation and PB pictures are not supported.

7.4.7 User Data Extraction

If user data (such as closed captioning, teletext, etc.) is present in the picture layer of the video stream, the video decoder extracts and stores it in a DRAM-based FIFO and provides an interrupt to the host CPU. The application software on the host CPU may then access the user data for further processing.

7.4.8 JPEG Decode Acceleration

Each video decoder supports the decoding of 4:2:0 and 4:2:2 YCbCr baseline JPEG images (from 1x1 to 4096x4096 resolution) using hardware acceleration. The resulting YCbCr image may be rendered into an appropriate display plane or off-screen memory (for additional processing). When rotating a 4:2:2 image, it is converted to 4:2:0, resulting in a minor degradation of video quality. 4:4:4 YCbCr or RGB images may be decoded using software on the host CPU.

BD-ROM and HD DVD-Video use JFIF v1.02 as the container format for JPEG images. BD-ROM and HD DVD-Video performance requirements are:

Table 7-4 **BD-ROM and HD DVD-Video JPEG decode performance requirements**

	BD-ROM	HD DVD-Video
JPEG	600K pixels/sec (baseline) 300K pixels/sec (progressive)	1.03M pixels/sec

For HD DVD-Video, one 1920x1080 JPEG image, one 1920x1080 PNG-32 image and one 1920x1080 PNG-8 image must be able to be decoded within six seconds without affecting playback. JPEG decoding performance is:

Table 7-5 **JPEG decode performance measurements**

JPEG Source	Decoding Performance (Mpixels/sec)
4:2:0 baseline	13
4:2:2 baseline	10

7.4.9 PNG/MNG Decode Acceleration

PNG/MNG decoding is done by software on the host CPU. No hardware acceleration is available. BD-ROM and HD DVD-Video performance requirements are:

Table 7-6 **BD-ROM and HD DVD-Video PNG decode performance requirements**

	BD-ROM	HD DVD-Video
PNG-32	500K pixels/sec	690K pixels/sec
PNG-8	700K pixels/sec	2.07M pixels/sec

Current PNG-32 and PNG-8 decoding performances are ~1.26M and ~2.76Mpixels/second, respectively.

7.4.10 GIF Decode Acceleration

GIF decoding is done by software on the host CPU. No hardware acceleration is available. BD-ROM and HD DVD-Video performance requirements are:

Table 7-7 **BD-ROM and HD DVD-Video GIF decode performance requirements**

	BD-ROM	HD DVD-Video
GIF	700K pixels/sec	TBD

7.4.11 Subtitle Decode

Each video decoder supports decoding of the following subtitle formats,

- DVD-Video bit-mapped subpicture
- 2-bit HD DVD-Video bit-mapped subpicture
- BD-ROM bit-mapped subtitle

The resulting image may be rendered into an appropriate display plane or off-screen memory (for additional processing).

To improve subtitle rendering performance, the video decoder includes hardware acceleration for run-length decoding of,

- DVD-Video bit-mapped subpictures
- 2-bit HD DVD-Video bit-mapped subpictures (DVD-Video compatible except HD resolution)

- BD-ROM bit-mapped subtitles

Hardware acceleration for 8-bit HD DVD-Video run-length decoding is not supported.

8 Video Processing Subsystem

8.1 Block Diagram of Video Processing Subsystem

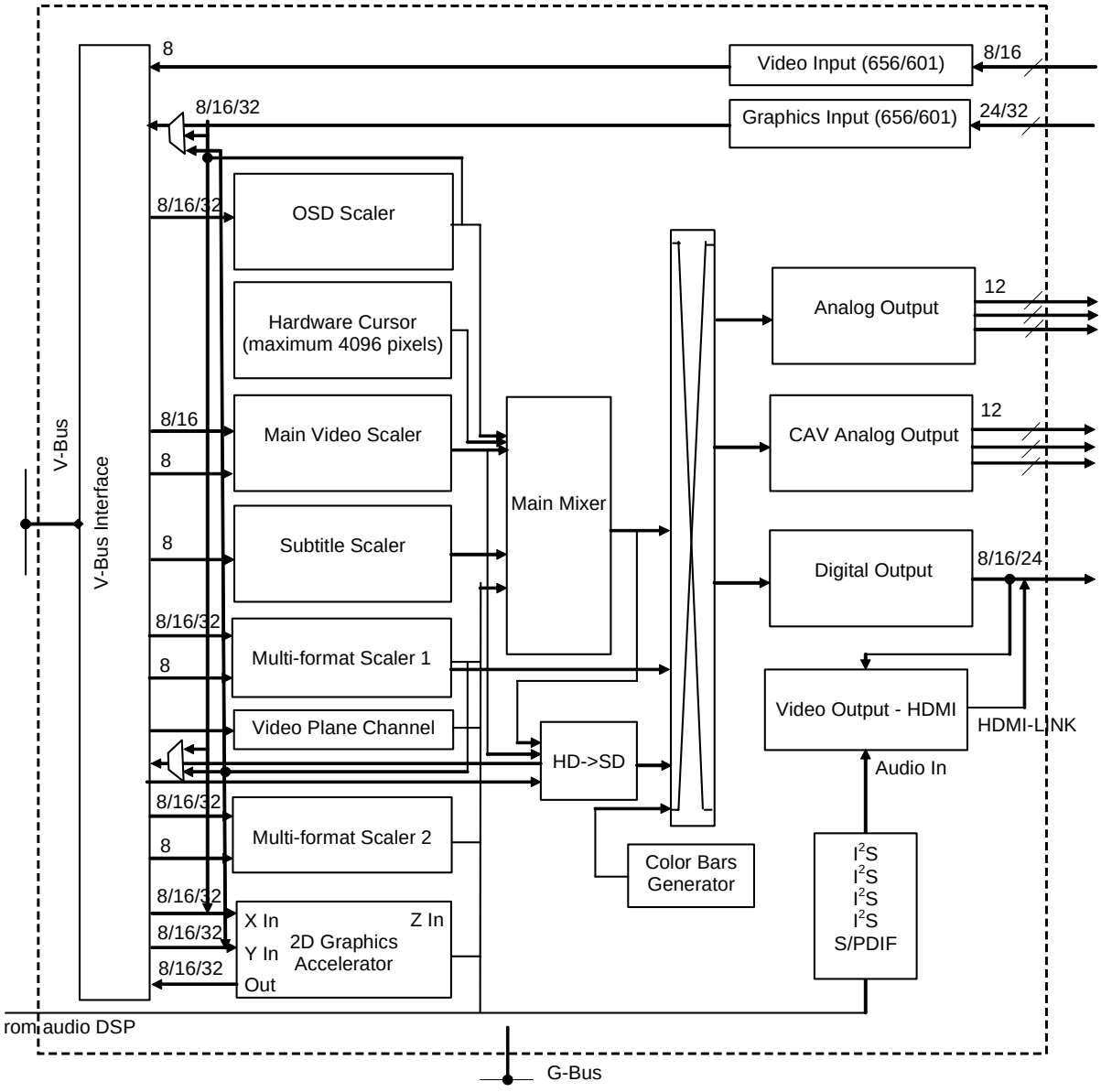


Figure 8-1 **Video processing subsystem block diagram**

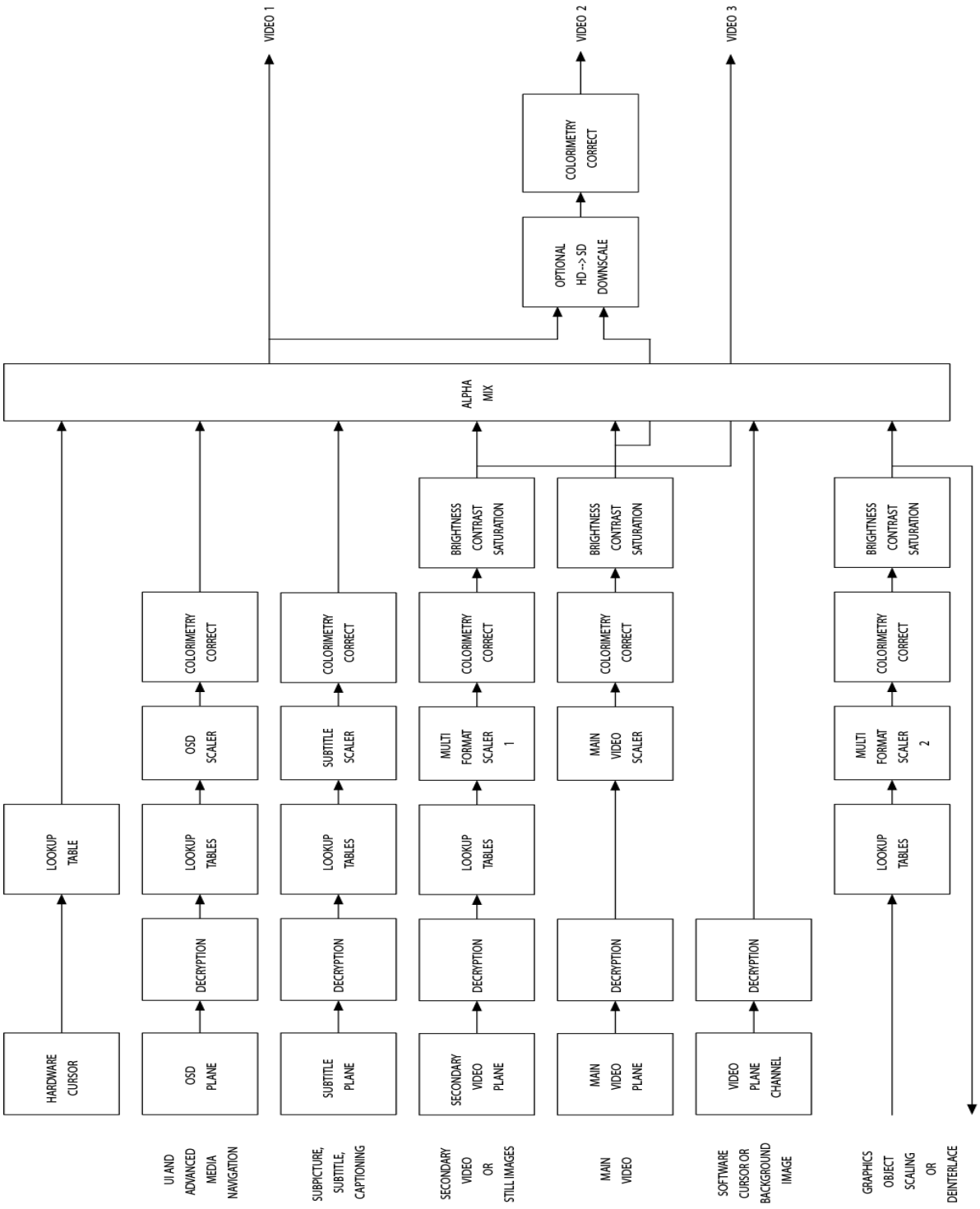


Figure 8-2 **Video processing pipeline**

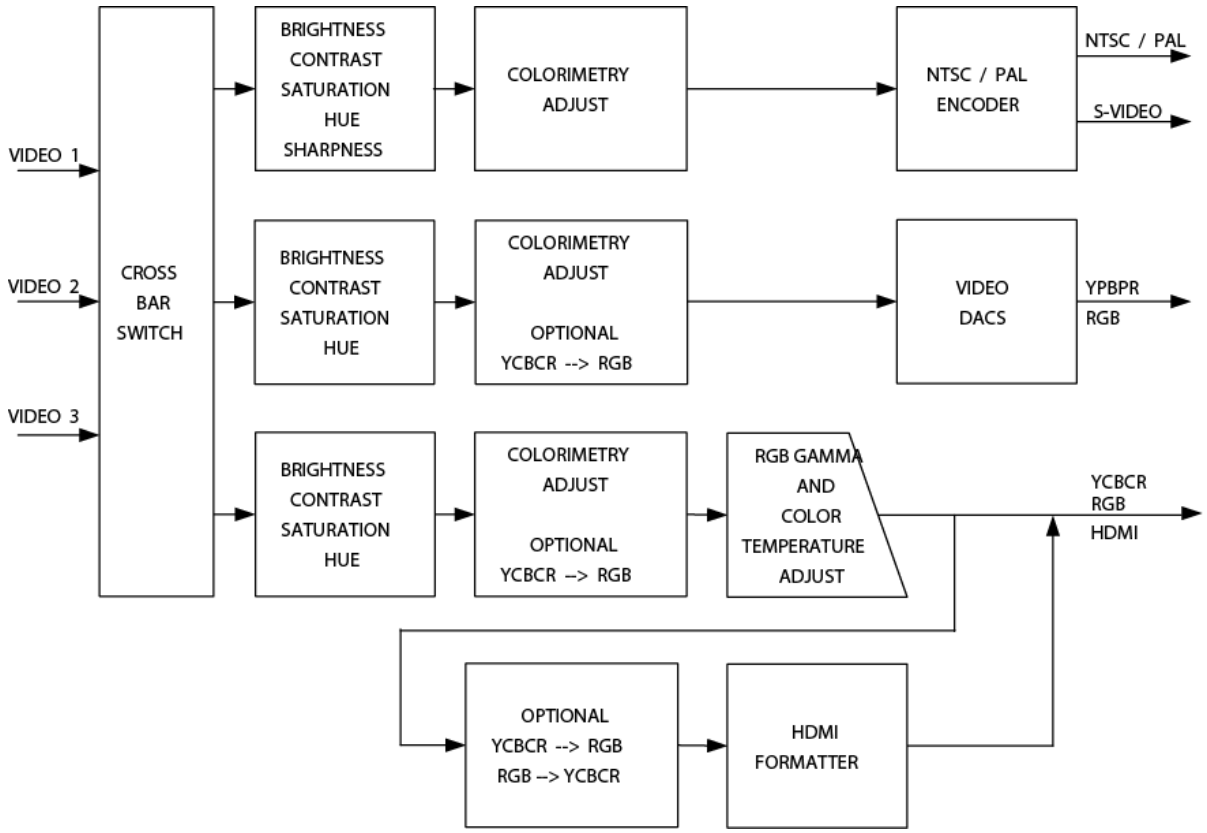


Figure 8-3 **Video processing pipeline**

8.2 Introduction

The video processing subsystem 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 include hardware-assisted 2D graphics acceleration, and support for video and graphics input ports. The primary functions performed include:

- Up- or down-scaling a video or graphics images 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 up to eight video/graphics sources
- Brightness, contrast and saturation controls
- TV encoding to NTSC or PAL standards (composite and s-video)

- Selection of analog out mode (various RGB and YPbPr formats)
- Video digital-to-analog conversion
- Digital video output formatting
- Video and graphics input interfaces
- 2D graphics acceleration

Each video and graphics picture retrieved from the memory can be converted to other pixel formats as necessary, and scaled independently. The results are then alpha-blended on a pixel-by-pixel basis using a programmable priority of the plane order.

The video processing block 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.

The 3 available video outputs consist of,

1. Digital video output: Supports 8, 16 or 24-bit RGB/YCbCr or HDMI data format.
2. Analog component output: Supports RGB or YPbPr.
3. Analog CVBS/S-video outputs: Supports NTSC/PAL composite and s-video.

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

8.3 Features

Video Processing

- Hardware cursor (4096 pixels, 4-bpp, up to 255 pixels horizontally and vertically)
- 2D graphics accelerator with OpenType/TrueType 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 main video, including support for motion-adaptive deinterlacing
- Supports arbitrary scaling of video and graphics of up to 1920x1080 pixels, with all common HDTV set-top box and HDTV scaling modes supported.
- Supports pixel-based alpha mixing of video, OSD, subtitles, graphics and hardware cursor with programmable global plane priority

Video Input and Output Interfaces

- Supports a flexible 40-bit video/graphics input interface
- Supports brightness, saturation and contrast controls for each output port
- 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 optional Macrovision v7.1.L1 protection (54MHz, 12-bit DACs)
- Supports 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)
- Supports a 150MHz RGB/YCbCr digital video output interface with option of gamma and color temperature control
  - 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
  - Supports BT.601, BT.656, or VIP 2.0 timing with optional ‘video valid’ output signal
  - Supports master or slave timing
- Supports HDMI v1.1 output (external PHY chip required)

8.4 Functional Description

8.4.1 Multi-plane Memory Architecture

Conceptually, the SMP8634 uses a multi-plane memory architecture. For many applications typical planes include:

- Hardware cursor
- OSD
- Subtitles (includes subpictures, teletext and closed captions)
- Secondary video (picture-in-picture or PIP)
- Main video

8.4.1.1 OSD Plane

The OSD plane contains menus for setting user preferences and displaying playback status, program guides, user messages, etc.

Application software on the host CPU is typically responsible for rendering data into the OSD plane. The following sample software is available from Sigma:

- Navigation for,
  - DVD-Video, DVD-VR, DVD+VR
  - DVD-Audio
  - VideoCD, SVCD
  - CD-DA
  - File Player (from HDD for example)
  - For additional information on sample navigation software, please refer to the appendix.
- Picture CD (JPEG files using ISO 9660 format)
- ATSC/DVB program guide

To improve OSD rendering performance, the SMP8634 includes hardware acceleration for,

- OpenType/TrueType font rendering
- 2D graphics acceleration with optional simultaneous up/down scaling

8.4.1.2 Subtitle Plane

The subtitle plane contains decoded bitmapped subtitles, text subtitles, animated graphics, subpictures, teletext and closed captioning data that is to be rendered over the main video. Whether or not it is rendered over the secondary video (PIP) is determined by the programmable global plane priority order.

Application software on the host CPU is typically responsible for rendering into the subtitle plane any subtitle formats not supported by the video decoder block. The following sample software are available from Sigma:

- Closed captioning for,
  - CEA-608
  - EIA-708
- Subtitles for,
  - BD-ROM text subtitles
  - HD DVD-Video advanced subtitles
- Navigation for,
  - DVD-Video
  - DVD-VR, DVD+VR
  - DVD-Audio
  - For additional information on sample navigation software, please refer to the appendix.

While the video decoder block uses hardware to improve bit-mapped subtitle performance, text based subtitle performance is enhanced by using hardware acceleration for,

- OpenType/TrueType font rendering
- 2D graphics acceleration with optional simultaneous up/down scaling

8.4.1.3 Secondary Video Plane

The secondary video plane contains decompressed video data from the secondary video decoder for PIP or dual TV output applications. It may alternately contain foreground graphics, background graphics and still images.

8.4.1.4 Main Video Plane

The main video plane contains decompressed video data from the main video decoder.

8.4.1.5 Typical Memory Plane and Scaler Usage

Each memory plane is typically capable of supporting a wide variety of picture buffer formats. In addition to support for double-buffered memory planes, off-screen graphics composition can be done, with the result rendered into any memory plane that supports the graphics picture buffer formats. Each memory plane is also typically associated with a scaler, capable of color format conversion and independent x-y scaling. After scaling, pixel-based alpha mixing is done. The priority display order of the memory planes is programmable, except for the on-chip hardware cursor which is always on top. How these memory planes and their associated scalers are typically used is shown below:

Table 8-1 **Memory planes and scaler usage for a typical application**

Memory Plane	Scaler Used	Function[1]
OSD	OSD scaler	Device OSD
Subtitle	Subtitle scaler	Subtitles Subpictures Closed Captions Teletext
Secondary Video[2]	Multi-format scaler 1 or 2	Secondary Video (PIP) Photos
Main Video[3]	Main video scaler	Main Video
	Multi-format scaler 1	Motion adaptive deinterlacing of Main Video Scaling of Secondary Video for PIP or still images
	Multi-format scaler 2	Scaling during 2D Graphics Acceleration Motion adaptive deinterlacing of Main Video Scaling of Secondary Video for PIP or still images

1. Each scaler may only be used for one of the indicated functions.
2. When both the main video and secondary video are high-definition MPEG-4.10 (H.264) or SMPTE 421M (VC-1), the OSD should be limited to 8bpp HD or upscaled 32bpp SD due to DRAM memory bandwidth considerations.
3. Motion adaptive (Type 2) deinterlacing of main video is not available when both Multi-format scaler 1 and 2 are used for other purposes.

8.4.1.6 Blu-ray and HD-DVD Applications

For more advanced applications, such as Blu-ray (BD) and HD-DVD, additional memory planes and scalers are available to enable the support of:

- Hardware cursor
- OSD
- Subtitles
- Secondary video
- Main video
- Background graphics or foreground software cursor

How these memory planes and their associated scalers can be used for the Blu-ray Disc (BD) and HD-DVD applications is shown below:

Table 8-2 **Memory planes and scaler usage for BD and HD-DVD applications (supports PIP or Type 2 motion adaptive deinterlacing)**

Memory Plane	Scaler Used	BD	HD-DVD
FG/BG Graphics	Video plane channel	Background Graphics	Cursor
OSD[1]	OSD scaler	Device OSD BD-J Graphics	Device OSD iHD Graphics
Subtitle	Subtitle scaler	Presentation Graphics Closed Captioning	Subpictures Advanced Subtitles Closed Captioning
Secondary Video[2], [3]	Multi-format scaler 1	Secondary Video Motion Adaptive Deinterlacing	Secondary Video Motion Adaptive Deinterlacing
Main Video	Main video scaler	Main Video	Main Video
BD IG[4]	Multi-format scaler 2	BD-J Graphics Acceleration HDMV Interactive Graphics	Scaling during 2D Graphics Acceleration

1. For BD, DirectFB software is used to merge the Device OSD and BD-J Graphics planes into a single OSD plane. For HD-DVD, DirectFB software is used to merge the Device OSD and iHD Graphics planes into a single OSD plane. DirectFB may only be used with DRAM bank 0.
2. The source resolution and frame rate of Secondary Video is limited to720x480p24, 720x576p24, 720x480i30, 720x480p30, 720x576i25 or 720x576p25.
3. Motion adaptive (Type 2) deinterlacing of Main Video is not available when Secondary Video is displayed using PIP (picture-in-picture) since Multi-format scaler 1 is used for both features. For Blu-ray players:
   - 1080p60 (Type 2 deinterlaced) output is possible when no secondary (PIP) video is displayed
   - 1080p60 (3-2 pulldown) or 1080p24 output is possible when both main and secondary (PIP) video are p24 sources. This limitation is because multi-format scaler 2 is needed for graphics performance in case the user calls up any menus.
   - 1080p60 (Type 1 deinterlaced) output is possible when both main and secondary (PIP) video are i30 sources.
4. BD-J Graphics acceleration and HDMV Interactive Graphics are mutually exclusive.

Table 8-3 **Alternate memory planes and scaler usage for BD applications (no support for PIP or Type 2 motion adaptive deinterlacing)**

Memory Plane	Scaler Used	BD
FG/BG Graphics	Video plane channel	Background Graphics
OSD	OSD scaler	Device OSD
Subtitle	Subtitle scaler	Presentation Graphics Closed Captioning
BD-J Graphics	Multi-format scaler 1	BD-J Graphics
Main Video	Main video scaler	Main Video
BD IG	Multi-format scaler 2	BD-J Graphics Acceleration HDMV Interactive Graphics

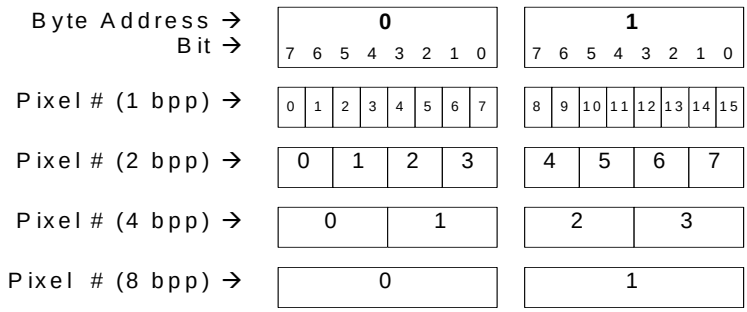
The FG/BG graphics plane contains still images that are used for BD background graphics (visible when no video or graphics is present for a region of the display), or a HD-DVD cursor image. Typically, a JPEG image, an MPEG-2 I-frame or, an SD MPEG-2 I-/P-frame video drip (1 every 30 seconds) is used to provide BD background graphics data. A PNG image (up to 256 x 256 8-bit pixels) is used to provide HD-DVD cursor data. Since the video plane channel does not have scaling capability, Multi-format scaler 2 and the 2D graphics accelerator are used to read the decoded source image, scale it to the desired output resolution, and write it to the FG/BG Graphics plane.

8.4.2 Picture Buffer Formats

8.4.2.1 Indexed Graphics – 1/2/4/8-bpp, using LUT

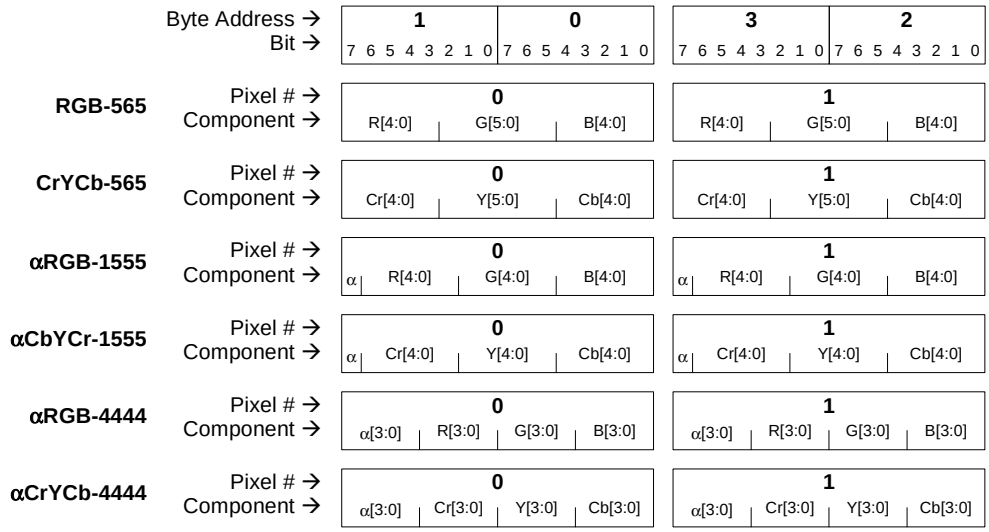
Indexed graphics LUT formats with 1/2/4-bpp are supported by the 2D graphics accelerator (Y input only), OSD scaler and multi-format scalers.

Indexed graphics LUT formats with 8-bpp is supported by the 2D graphics accelerator (X and Y inputs, output) as well as the OSD scaler, subtitle and multi-format scalers.



8.4.2.2 Graphics - 16-bpp

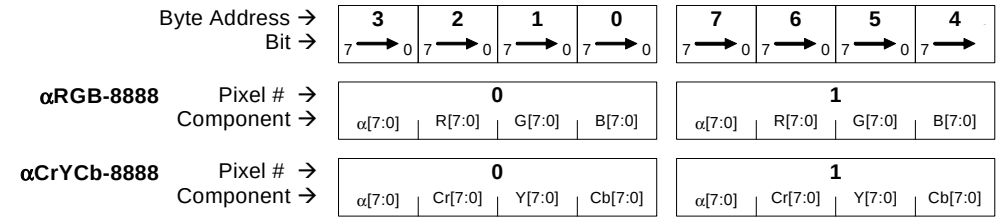
There are six formats using 16-bpp. All are supported by the 2D graphics accelerator (X and Y inputs, output) and by the OSD and multi-format scalers.



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.

8.4.2.3 Graphics - 32-bpp

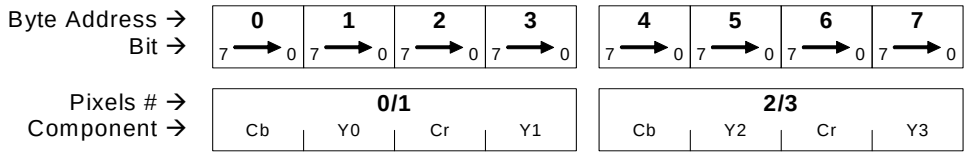
These modes are supported by the 2D graphics accelerator (source input, destination input and destination output) and by the OSD and multi-format scalers.



8.4.2.4 Video-Luminance

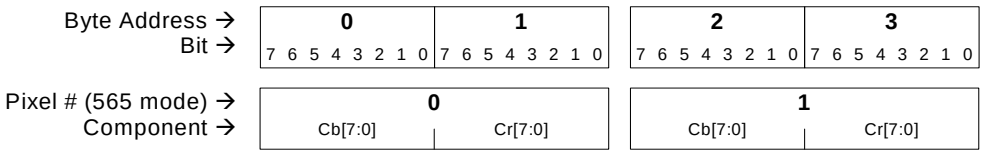
When an MPEG bit stream is decoded, the resulting frames are in the 4:2:0 or the 4:2:2 format. In either case, the luminance and the chrominance are stored in separate buffers (one buffer for luminance and one buffer for Cb and Cr).

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.



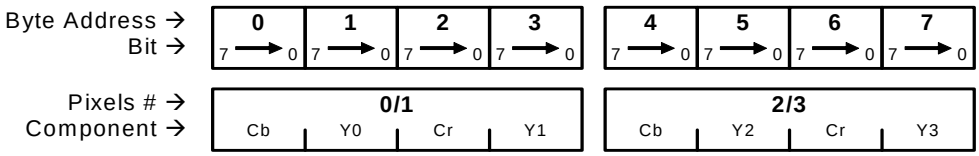
8.4.2.5 Video-Chrominance

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.



8.4.2.6 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:



8.4.3 OSD Scaler

The OSD enables full-screen menus, images and text to be blended over the video, graphics and subtitles. The OSD scaler accepts most graphics data formats, and performs programmable edge adaptive scaling with directional filtering. Input resolutions of up to 2048x1080 are supported.

Supported input formats to the OSD scaler are:

- Indexed graphics - 1, 2, 4 and 8-bpp (no subtitle/subpicture support)
- Graphics - 16-bpp
  - RGB-565
  - CrYCb-565
  - αRGB-1555
  - αYCbCr-1555
  - αRGB-4444
  - αYCbCr-4444
- Graphics - 24-bpp
  - RGB-888
  - YCbCr-888
- Graphics - 32-bpp
  - αRGB-8888
  - αYCbCr-8888

Before scaling, four 256x8 LUTs are available to perform αRGB gamma correction, 0-255 to 16-235 RGB range conversion or Indexed Graphics to 16/24/32-bpp RGB/YCbCr conversion.

The individually programmable H and V-scalers can operate in either 2-tap mode (2-tap horizontally and 4-tap vertically so processing is done on a 2x4 matrix of data) or 4-tap mode (4-tap horizontally and 4-tap vertically so processing is done on a 4x4 matrix of data). 2-tap scaling is implemented as two 63-coefficient, 16 times interpolating filters - for each of the 16 interpolating positions, 2 coefficients (taps) are used. In the 2-tap mode, the input line resolution of up to 2048 pixels is supported for all the input data formats. 4-tap scaling is implemented as four 63-coefficient, 16 times interpolating filters - for each of the 16 interpolating positions, 4 coefficients (taps) are used. In the 4-tap mode, the input line buffer size restricts the input line resolution to 1440 or 1024 pixels in 24- and 32-bpp data formats.

Table 8-4 **OSD line resolutions**

Tap	Pixel Resolution	Vertical Scaling Ratio	Maximum Line Resolution
2-tap	16-bpp	Any	2048
	24/32-bpp	Any	2048
4-tap	16-bpp	Any	2048
	24/32-bpp	(X1, Infinity)	1920
		[/2, X1]	1440
		[/3, /2]	1024

The OSD scaler also implements a 3-tap flicker reduction filter with edge detection (scaler must be in 4-tap mode). Three sets of coefficients provide light, moderate, and strong flicker reduction options, in addition to no filtering.

After scaling, BT.601 or BT.709 RGB to YCbCr conversion and colorimetry correction between BT.601 and BT.709 may be performed. No contrast (or ‘picture’ or ‘white level’), saturation (or ‘color’), brightness (or ‘black level’), hue (or ‘tint’) or sharpness control is provided. The data (typically 32-bpp αYCbCr) is then sent to the mixer, 2D graphics accelerator (X input) or DRAM.

8.4.3.1 Alpha Support

For the 1/2/4/8-bpp, RGB-656, RGB-888, CrYCb-565 and CrYCb-888 source formats, 8 bits of alpha data can be generated via the α0 value in the VO_OSD_ALPHA_ROUTING register. By changing the value of α0, the entire OSD may be faded in and out.

For the αRGB-1555 and αYCbCr-1555 formats (either source or from the LUTs), two bytes are used to generate 8 bits of alpha information, the α0 and α1 values in the VO_OSD_ALPHA_ROUTING register. Which alpha value is used is determined by the 1-bit source alpha value. By changing the value of these bytes, the entire OSD may be faded in and out.

For the αRGB-4444, αYCbCr-4444, αRGB-8888 and αYCbCr-8888 formats (either source or from the LUTs), the entire OSD may be faded in or out by generating new alpha information as follows:

AO = (AS * α1) + ([1-AS] * α0) where,

AO = Output alpha value

AS = Source alpha value

α0, α1 = Values of the α0 and α1 in the VO_OSD_ALPHA_ROUTING register.

8.4.3.2 Color Keying

Color keying forces the output alpha data to a value of 0 (transparent) whenever the input color falls within the color key value and 4-bit range. The color key value is set according to the input source color space (which may different from the output color space).

It is not possible to use the color key feature with Indexed Graphics data.

8.4.4 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. Each 4-bit pixel is fed to a 16x32 look-up table which outputs 32-bit (8-8-8-8 format) αYCbCr pixels.The data is then sent to the 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

8.4.5 Multi-format Scaler 1 and Multi-format Scaler 2

The Multi-format scaler 1 and Multi-format scaler 2 can be used for a variety of functions such as,

- Enabling a secondary video to be displayed over the main video to support picture-in-picture (PIP) applications
- Enabling the output of a second video stream for dual TV applications (Multi-format scaler 1 only)
- Supporting Type 2 deinterlacing (motion adaptive deinterlacing) of the main video
- Scaling during 2D graphics processing (Multi-format scaler 2 only)
- Scaling and displaying of high-resolution JPEG images (photos)

Note: Only two of the above features may be used at a time since there are only two multi-scalers.

Input resolutions of up to 2048x1080 are supported. Supported input formats to the Multi-format scaler 1 and Multi-format scaler 2 are,

- Video (4:2:0 or 4:2:2 YCbCr)
- Indexed graphics - 1, 2, 4 and 8-bpp (no subtitle/subpicture support)
- Graphics - 16-bpp
  - RGB-565
  - YCbCr-565
  - αRGB-1555
  - αYCbCr-1555
  - αRGB-4444
  - αYCbCr-4444
- Graphics - 24-bpp
  - RGB-888
  - YCbCr-888
- Graphics - 32-bpp
  - αRGB-8888
  - αYCbCr-8888

Before scaling, four 256x8 LUTs are available to perform αRGB gamma correction, 0-255 to 16-235 RGB range conversion or Indexed Graphics to 16/24/32-bpp RGB/YCbCr conversion.

The individually programmable H and V-scalers operate in a 2-tap mode (2-tap horizontally and 4-tap vertically so processing is done on a 2x4 matrix of data) with a scaling range of 0.25 to infinity. The 2-tap scaling is implemented as two 63-coefficient, 16 times interpolating filters - for each of the 16 interpolating positions, 2 coefficients (taps) are used. Additional downscaling is possible by employing the pre-downscaler before the 2-tap scalers. The scalers can support the differing chrominance sample alignment of MPEG-1 versus MPEG-2. Memory limitations on rev. A and B constrain the pixel/line input (line buffer level) resolutions, rev. C and later do not have these constraints.

Table 8-5 **Multi format scaler line resolutions**

Pixel Resolution	Vertical Scaling Ratio	Maximum Line Resolution rev. A and B	Maximum Line Resolution rev. C and later
16-bpp	Any	2048	2048
24/32-bpp	[X1, Infinity]	2048	2048
	[/2, X1]	1280	2048
	[/3, /2]	1024	2048

A horizontal downscaler before the H scaler is available to support larger downscaling factors. It decreases the number of pixels per line (using pixel skipping) by a fractional ratio. Vertical downscaling before the V scaler (by skipping scan lines) is also possible. Combining these two capabilities enables support for source resolutions up to 4096x4096 or scaling factors less than 0.25x.

For displaying 4:3 content on a 16:9 display, the commonly used ‘pillar’, ‘wide’, ‘zoom’ and ‘panorama’ modes can be implemented. For the ‘pillar’ mode, the 4:3 source is displayed without distortion; black bars are added to the left and right sides of the 4:3 image to fill the 16:9 screen. For the ‘wide’ mode, the 4:3 source is linearly scaled to fill the 16:9 display, resulting in a distorted image unless anamorphic content is used. For the ‘zoom’ mode, the center 16:9 portion of the 4:3 source is displayed without distortion. For ‘panorama’ mode, the center portion of the image is linearly scaled, while the left and right sides are nonlinearly scaled; the left and right locations where nonlinear scaling stops/starts is programmable.

For displaying 16:9 content on a 4:3 display, the commonly used ‘crop’, ‘letterbox’ and ‘squeeze’ modes can be implemented. For the ‘crop’ mode, the center 4:3 portion of the 16:9 source is displayed without distortion. For the ‘letterbox’ mode, the entire 16:9 source is displayed without distortion; black bars are added to the top and bottom of the 16:9 image to fill the 4:3 screen. For the ‘squeeze’ mode, the 16:9 source is linearly scaled to fill the 4:3 screen, resulting in a distorted image. The appendix illustrates the various scaling modes, and is also available on our web site.

After scaling, a 3x3 matrix multiplier is used to implement,

- RGB to YCbCr conversion (BT.601 or BT.709)
- Contrast (or ‘picture’ or ‘white level’) control, range of 0 to 4x in 1/128 steps
- Saturation (or ‘color’) control, range of 0 to 4x in 1/128 steps
- BT.709 (HDTV) to BT.601 (SDTV) and BT.601 (SDTV) to BT.709 (HDTV) colorimetry conversion

The matrix multiplier is followed by three programmable DC offsets to provide optional brightness (or ‘black level’) control. The DC offsets have a range of -512 to +511.7/8, in steps of 1/8. No sharpness or hue (or ‘tint’) control is provided. The appendix illustrates differences of BT.601 and BT.709 colorimetry, and is also available on our web site.

Note that content being displayed via a Multi-format scaler must be authored for flicker-free display on an interlaced TV; there is no anti-flicker filtering filter available.

The output of the Multi-format scaler 1 (typically 32-bpp αYCbCr) is input to the main mixer (when used for PIP or motion adaptive deinterlacing), display routing block (when used for dual TV output), or 2D graphics accelerator (Y input), or may be stored in DRAM. The output of the Multi-format scaler 2 is input to the main mixer (when used for PIP or motion adaptive deinterlacing) or 2D graphics accelerator (Z input).

8.4.5.1 Alpha Support

For the video, 1/2/4/8-bpp, RGB-656, RGB-888, CrYCb-565 and CrYCb-888 source formats, 8 bits of alpha data can be generated via the α0 value in the VO_VCR_ALPHA_ROUTING and VO_GFX_ALPHA_ROUTING registers. By changing the value of α0, the entire picture may be faded in and out.

For the αRGB-1555 and αYCbCr-1555 formats (either source or from the LUTs), two bytes are used to generate 8 bits of alpha information, the α0 and α1 values in the VO_VCR_ALPHA_ROUTING and VO_GFX_ALPHA_ROUTING registers. Which alpha value is used is determined by the 1-bit source alpha value. By changing the value of these bytes, the entire picture may be faded in and out.

For the αRGB-4444, αYCbCr-4444, αRGB-8888 and αYCbCr-8888 formats (either source or from the LUTs), the entire picture may be faded in or out by generating new alpha information as follows:

AO = (AS * α1) + ([1-AS] * α0) where,

AO = Output alpha value

AS = Source alpha value

α0, α1 = Values of the α0 and α1 in the VO_VCR_ALPHA_ROUTING register and VO_GFX_ALPHA_ROUTING registers.

8.4.5.2 Color Keying

Color keying forces the output alpha data to a value of 0 (transparent) whenever the input color falls within the color key value and 4-bit range. The color key value is set according to the input source color space (which may different from the output color space).

It is not possible to use the color key feature with Indexed Graphics data or video data.

8.4.5.3 Luma Keying of Secondary Video - HD DVD-Video

Although secondary video for HD DVD-Video can be done, the SMP8634 cannot support the programmable range of luma keying that HD-DVD specifies. This prevents the main video being shown for below-black secondary video pixels. Therefore, the PIP will always be rectangular.

8.4.5.4 Luma Keying of Secondary Video - BD-ROM

On rev. C and later, luma keying of BD-ROM secondary video is supported. Additional memory bandwidth is not required for perform luma keying since it is a real-time hardware-based process within the video processing pipeline.

8.4.6 Main Video Scaler

The main video scaler is used to scale the main video. Input resolutions up to 2048x1080 are supported.

Supported input formats to the video scaler are:

- Video (4:2:0 or 4:2:2 YCbCr)

The individually programmable H and V-scaler operates in a 4-tap mode (4-tap horizontally and 4-tap vertically so processing is done on a 4x4 matrix of data). 4-tap scaling is implemented as four 63-coefficient, 16 times interpolating filters - for each of the 16 interpolating positions, 4 coefficients (taps) are used.

For displaying 4:3 content on a 16:9 display, the commonly used ‘pillar’, ‘wide’, ‘zoom’ and ‘panorama’ modes can be implemented. For the ‘pillar’ mode, the 4:3 source is displayed without distortion; black bars are added to the left and right sides of the 4:3 image to fill the 16:9 screen. For the ‘wide’ mode, the 4:3 source is linearly scaled to fill the 16:9 display, resulting in a distorted image unless anamorphic content is used. For the ‘zoom’ mode, the center 16:9 portion of the 4:3 source is displayed without distortion. For ‘panorama’ mode, the center portion of the image is linearly scaled, while the left and right sides are nonlinearly scaled; the left and right locations where nonlinear scaling stops/ starts is programmable.

For displaying 16:9 content on a 4:3 display, the commonly used ‘crop’, ‘letterbox’ and ‘squeeze’ modes can be implemented. For the ‘crop’ mode, the center 4:3 portion of the 16:9 source is displayed without distortion. For the ‘letterbox’ mode, the entire 16:9 source is displayed without distortion; black bars are added to the top and bottom of the 16:9 image to fill the 4:3 screen. For the ‘squeeze’ mode, the 16:9 source is linearly scaled to fill the 4:3 screen, resulting in a distorted image. The appendix illustrates the various scaling modes, and is also available on our web site.

After scaling, a 3x3 matrix multiplier is used to implement,

- Contrast (or ‘picture’ or ‘white level’) control, range of 0 to 4x in 1/128 steps
- Saturation (or ‘color’) control, range of 0 to 4x in 1/128 steps
- BT.709 (HDTV) to BT.601 (SDTV) and BT.601 (SDTV) to BT.709 (HDTV) colorimetry conversion

The matrix multiplier is followed by three programmable DC offsets to provide optional brightness (or ‘black level’) control. The DC offsets have a range of -512 to +511.7/8, in steps of 1/8. No sharpness or hue (or ‘tint’) control is provided. The appendix illustrates differences of BT.601 and BT.709 colorimetry, and is also available on our web site.

The output of the video scaler (32-bpp αYCbCr) is input to the main mixer (for viewing) and the HD to SD downscaler (for downscaling to SD resolution when HD+SD outputs are used).

8.4.6.1 Deinterlacing

Two deinterlacing modes are available, Type 1 and Type 2.

De-interlacing 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.

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.

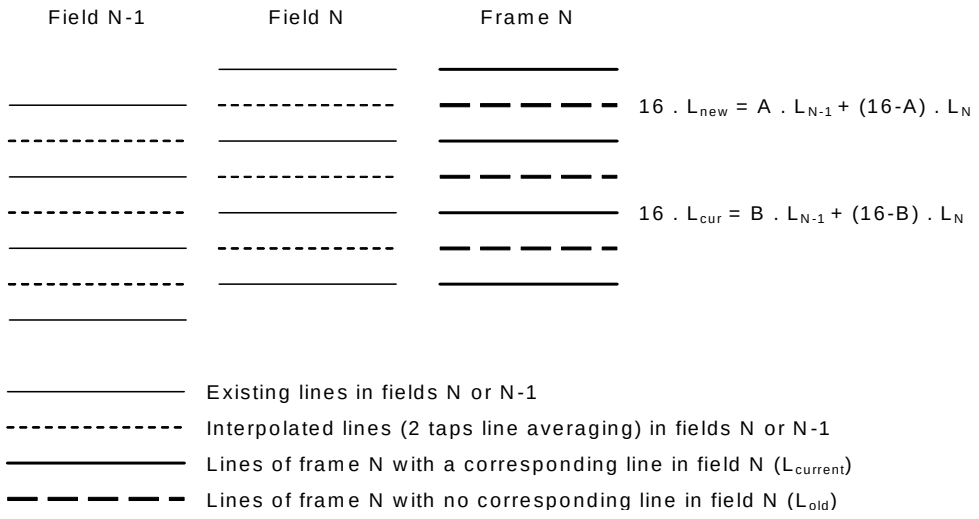


Figure 8-4 **De-interlacing type 1**

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

This mode is motion and edge adaptive and 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 (‘weave’) when there is little motion, and intra-field deinterlacing (‘bob’) 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 the Multi-format scaler 1 or Multi-format scaler 2. 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-format scaler 1 or Multi-format scaler 2 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 480i and the display is 720P. It could also be used to scale from one interlaced format to another interlaced format: for instance 480i to 1080i, or 480i to 576i.

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-format scaler 1 or Multi-format scaler 2 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).

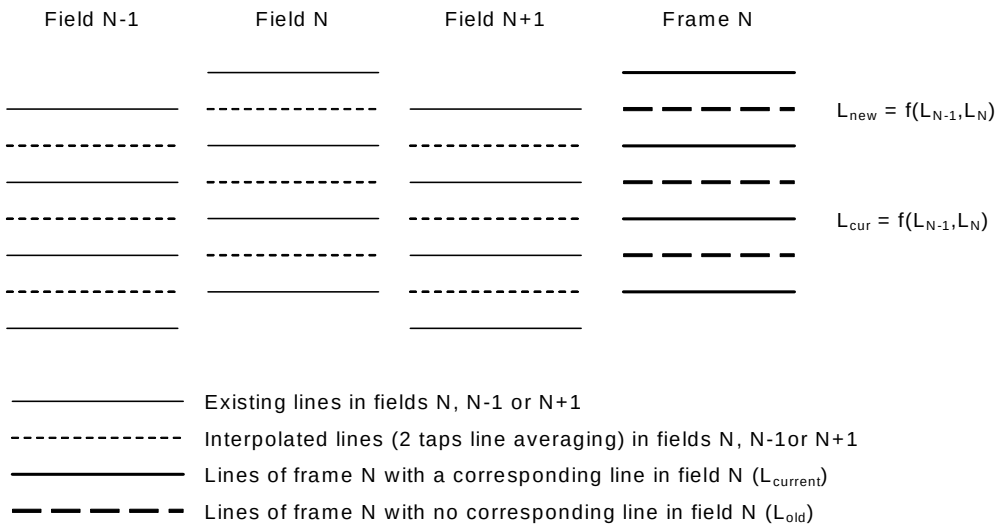


Figure 8-5 **De-interlacing type 2**

8.4.6.2 Film Grain Technology: HD DVD-Video

Thomson's Film Grain Technology (FGT) for HD DVD-Video is not supported. As a result, the main video cannot have artificial film grain added to it during playback.

8.4.7 Subtitle Scaler

This scaler enables closed captioning, teletext and various types of subtitles/subpictures to be blended over the video. Input resolutions of up to 2048x1080 are supported.

Supported input formats to the subtitle scaler are:

- Indexed graphics - 1, 2, 4 and 8-bpp
- Indexed graphics - 8-bpp + 8-bpp of alpha

Before scaling, a 256x32 LUT performs Indexed Graphics to 32-bpp αYCbCr conversion. In the 8-bpp + 8-bpp alpha mode, the 8-bpp of indexed graphic data addresses the 256x32 LUT to generate 24-bpp YCbCr; the 8-bpp of source alpha data is directly used.

The individually programmable H and V-scalers operate in a 2-tap mode (2-tap horizontally and 4-tap vertically so processing is done on a 2x4 matrix of data), with a scaling range of 0.25 to infinity. 2-tap scaling is implemented as two 63-coefficient, 16 times interpolating filters - for each of the 16 interpolating positions, 2 coefficients (taps) are used.

A horizontal downscaler before the H scaler is available to support larger downscaling factors. It decreases the number of pixels per line (using pixel skipping) by a fractional ratio. Vertical downscaling before the V scaler (by skipping scan lines) is also possible. Combining these two capabilities enables support for source resolutions up to 4096x4096 or scaling factors less than 0.25x.

The subtitle scaler also implements a 3-tap flicker reduction filter. Three sets of coefficients provide light, moderate, and strong flicker reduction options, in addition to no filtering.

After scaling, colorimetry correction from BT.601 to BT.709 may be performed. No contrast (or 'picture' or 'white level'), saturation (or 'color'), brightness (or 'black level'), hue (or 'tint') or sharpness control is provided. The output of the subtitle scaler (32-bpp αYCbCr) is input to the main mixer.

The 8-bit HD DVD-Video subpicture format is not supported since it requires each of the three button states to support 256 colors, requiring three 256x32 LUTs and more than 8bpp memory plane support.

8.4.8 HD to SD Scaler

This scaler is used to downscale HD content to SD resolution to enable simultaneous HD and SD video outputs. Possible sources to the scaler are either the main mixer output (used for simultaneous SD and HD outputs) or, main video scaler output (no OSD available, so useful for recording).

The HD image from the main mixer or main video scaler is first horizontally downscaled, with a range of 1/8x to 1x in increments of 1/2048, using a bilinear filter. The first 0-1023 pixels and the first 0-1023 lines of the HD image can be ignored, and 3 different 3-tap low-pass filters can be enabled prior to the horizontal downscaling. 4:4:4 to 4:2:2 YCbCr or RGB-888 to RGB-565 conversion is then performed to reduce the DRAM memory bandwidth and the on-chip memory buffer sizes. Vertical downscaling, with a range of 1/8x to 1x in increments of 1/2048, is then performed using a bilinear filter. Up to 511 pixels or lines of black stripes can then be added on the left/right sides or top/bottom sides of the SD image.

For displaying 16:9 content on a 4:3 display, the commonly used 'crop', 'letterbox' and 'squeeze' modes can be implemented. For the 'crop' mode, the center 4:3 portion of the 16:9 source is displayed without distortion. For the 'letterbox' mode, the entire 16:9 source is displayed without distortion; black bars are added to the top and bottom of the 16:9 image to fill the 4:3 screen. For the 'squeeze' mode, the 16:9 source is linearly scaled to fill the 4:3 screen, resulting in a distorted image. The appendix illustrates the various scaling modes, and is also available on the Sigma web site.

The down-scaled content is sent either directly to the display routing block (on-the-fly mode) or stored in DRAM and read back before being displayed (buffered mode).

8.4.8.1 On-the-fly Mode

In on-the-fly mode, the 4:2:2 YCbCr or RGB-565 SD image is converted to 4:4:4 YCbCr or RGB-888 and sent directly to the display routing block. This mode does not use any DRAM memory bandwidth, but there are two considerations.

VSYNC frequency: The SD and HD outputs must be running at exactly the same VSYNC frequency.

Vertical scaling mismatch: The scaling ratio between the HD source and the SD output is determined by the ratio of total lines (for example 525 from 1125 = 0.467, when scaling from 1080i to 480i), which differs from the ratio of active lines (480 from 1080 = 0.444 using the same example). The first ratio is typically higher and as a consequence the on-the-fly mode will usually result in an active picture that is slightly stretched vertically. In the example, 480 / 0.467 * 0.444 = 457 active lines that will be stretched (~5%) and displayed using 480 lines.

8.4.8.2 Buffered Mode

Although this implementation requires DRAM memory bandwidth, it makes it possible to output HD at 60Hz and SD at 59.94Hz (for example) and/or avoid the vertical scaling mismatch issue.

The 4:2:2 YCbCr or RGB-565 SD image is written to DRAM at the HD display vsync frequency. The SD image is then read back from the DRAM at the SD display vsync frequency. Synchronization between the vsync frequencies is done in the software by using the multiple buffers in the DRAM. The SD image is converted back to 4:4:4 YCbCr or RGB-888 and the result delivered to the display routing block.

8.4.9 Video Plane Channel

The video plane channel, useful for BD-ROM background graphics or HD DVD-Video software cursor, reads YCbCr data from the DRAM, converts it to 4:4:4 YCbCr data (if required) and provides it to the main mixer. No scaling, color space conversion or user adjustments (brightness, contrast, etc.) are available. Input resolutions of up to 2047x2047 are supported.

Supported input formats are:

- 16-bpp 4:2:2 YCbCr (CbCr aligned with Y or half-way between Y samples)
- 24-bpp 4:4:4 YCbCr
- 32-bpp 4:4:4:4 αYCbCr

Typical sources for background images are MPEG-2 I-frames and JPEG images. Typical sources for foreground software cursors are PNG images. These would need to be decoded, converted to the appropriate YCbCr format and scaled to the desired resolution before being used as background graphics.

8.4.9.1 Alpha Support

For the 16-bpp 4:2:2 YCbCr and 24-bpp 4:4:4 YCbCr formats, 8 bits of alpha data can be generated via the alpha value in the register VO_VP_CONF. By changing the value of alpha, the entire image may be faded in and out.

For the 32-bit 4:4:4 αYCbCr format, the 8 bits of source alpha data are passed through.

8.4.9.2 Color Keying

Color keying forces the output alpha data to a value of 0 (transparent) whenever the input Y data has a value of 0 and/or 255 (selectable).

8.4.10 2D Graphics and OpenType/TrueType Font Accelerator

The 2D graphics 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. It handles 2D bitmap basic operations (moves, blends, replaces, merge, alpha modulation, raster operations etc.).
2. A vectorial accelerator to generate a two colors per pixel mask from a vectorial definition (e.g. true type fonts). It renders quadratic and cubic vectorial streams (fonts).
3. The gradient generator to generate linear or radial fills.

The accelerator can be combined with the OSD scaler, the VCR multi scaler, the GFX multi scaler to support scaling and tiling. The GFX multi scaler is connected to the Z input, the VCR multi scaler can be connected to the Y input, the OSD scaler can be connected to the X input.

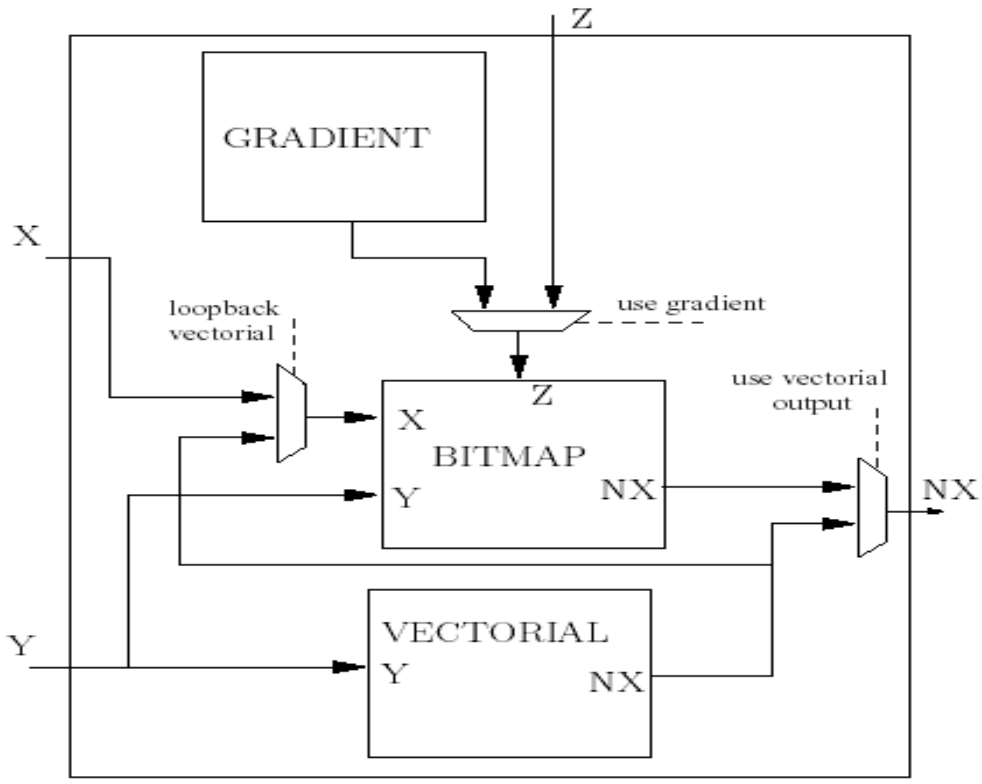


Figure 8-6 **Graphics accelerator block diagram**

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. The graphics accelerator also includes a vectorial mode unit which greatly accelerates the rendering of vector mode fonts such as OpenType/TrueType.

8.4.10.3 Graphics Accelerator (Bitmap Mode)

The graphic accelerator has 3 inputs and 1 output to DRAM. 2 inputs are DRAM channels while the 3rd one allows to couple the accelerator with a multi format scaler, thus giving scaling and format conversion capacities to the accelerator.

The graphics accelerator provides 5 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: Combines graphics in the memory.
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.

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 as $\alpha_{out} = \{\alpha_{in}, \alpha_{in}\}$
   - 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 whether the pixel is transparent or not, in case of a replace command. For example,

Y_for_mode = ‘b 010 (LUT 4-bpp)

Y_sub_mode = ‘b 001 (16-bpp 1555)

Y_alpha_0 = ‘h 00

Y_alpha_1 = ‘h FF

Output_format = ‘b 00 (8-bpp)

1. Input: {P_{n+1}[3], P_{n+1}[2], P_{n+1}[1], P_{n+1}[0], P_n[3], P_n[2], P_n[1], P_n[0]} (two 4-bit -pixels per byte).
2. Pixels are serialized (P_n and P_{n+1} are separated).
3. For each n, P_n addresses the 256x32 LUT. The true color output is TC(P_n). In that case, TC(P_n) is in 16-bpp 1555.
4. The alpha associated to P_n is determined as follows: $\alpha(P_n)$ = [TC(P_n)[15] = 1] ? Y_alpha_1: Y_alpha_0
5. Output of the Y channel: {$\alpha(P_n)$[7:0], 20’d 0, P_n[3:0]}

In case the current command is replace, {4’d 0, P_n[3:0]} will be written in the DRAM, using a flag determined by $\alpha(P_n)$[7:0].

The Z channel is the 32-bpp output of the Multi-format scaler 2.

Commands

Fill: Fills a rectangle with a font. The V-Bus interface handles DRAM addressing to shape the rectangle. The active channels are X, Y/Z and X'.

Blend: Blends any of X,Y or Z with any of X,Y or Z. The X' resolution is 8-bit (no LUT encoding). Blend order is selected by BLEND_ORDER. The active channels are X/Y/Z, X/Y/Z and X'. The 'blend' command requires 2 channels, data read through X, Y or Z.

If the full blending option is not enabled then the whether top layer is read through X,Y or Z is set by the scaling requirements. The top layer provides the alpha information. The output layer's alpha is calculated as follows: $\alpha_{out} = \alpha_{\tau op} + \alpha_{bottom}$ (saturated to FF).If the alpha saturation bit in VO_GRAPH_ACC_CONTROL is enabled, α_{out} = FF.

Otherwise the blending takes place as follows (with a normalized between 0 and 1):

$$\alpha_{out} = \alpha_{\tau op} + \alpha_{bottom} - \alpha_{\tau op} * \alpha_{bottom}$$

$$D_{out} = D_{\tau op} * \alpha_{\tau op}/\alpha_{out} + D_{bottom} * (1-\alpha_{\tau op}/\alpha_{out})$$

The layer generated by this logic can be used for further blend operations in the accelerator or the mixer.

Move: Moves a graphic in the DRAM. The V-Bus interface handles the DRAM addressing to move the graphic.The channel Y/Z is selected by SOURCE_CTRL. If MERGE_CTRL is enabled, then the X channel provides the accelerator with alpha information which is merged with the color information provided by Y/Z.The active channels are X, Y/Z or X'.

The main purpose of this command is to move data from an area of DRAM to another (the frame buffer for instance), using channels Y or Z. While being moved, the data may be scaled, its pixel resolution may be increased, and it can be merged with the alpha data. In the latter case, a color layer can be read from the DRAM through Y or Z as well as an alpha layer through X. The merging operation is controlled by MERGE_CTRL (DATA_CTRL[1]). The alpha layer can be coded in 1/2-bpp (2/4 bytes LUT modes) or in 4/8-bpp (true alpha modes).

Replace: Replaces data without using channel X. Similar to Move, but generates a 1-bpp flag (active high) to enable DRAM writing. For each pixel the flag is set to 1 if its alpha information is not nil. No source is needed, saves bandwidth (compared to Blend no need to read the destination).The channel Y/Z is selected by SOURCE_CTRL. If MERGE_CTRL is enabled, then the X channel provides the accelerator with alpha information which is merged with the color information provided by Y/Z.The active channels are X, Y/Z or X'

The main purpose of this command is to move data from an area of DRAM to another (the frame buffer for instance), using channels Y or Z. While being moved, the data may be scaled, its pixel resolution may be increased, and it can be merged with the alpha data. In the latter case, a color layer can be read from the DRAM through Y or Z as well as an alpha layer through X. The merging operation is controlled by MERGE_CTRL (DATA_CTRL[1]). The alpha layer can be coded in 1/2-bpp (2/4 bytes LUT modes) or in 4/8-bpp (true alpha modes).

"Replace" generates a 1-bpp flag sent to the V-Bus write channel, that enables the writing of the pixel in the DRAM. The flag is generated as follows: flag = (alpha!= 0), where alpha is the transparency information of the output pixel.

Raster operations: (see Appendix for a description of the operations). The Destination is mapped on X, Source is mapped on Z and Pattern is mapped on Y. All 3 operands should therefore be stored in the DRAM and padded if necessary to fit the image dimensions. To avoid using the Z channel, a 2 operands operation code may be altered in order to use a Source instead of a Pattern.

Raster operations can be made in 16/24/32-bpp. They can also be performed in 8-bpp. The V-Bus channels must then be programmed in an 8-bpp mode and the graphic accelerator in a 32-bpp mode.

8.4.10.4 Graphic Accelerator (Vectorial Mode)

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

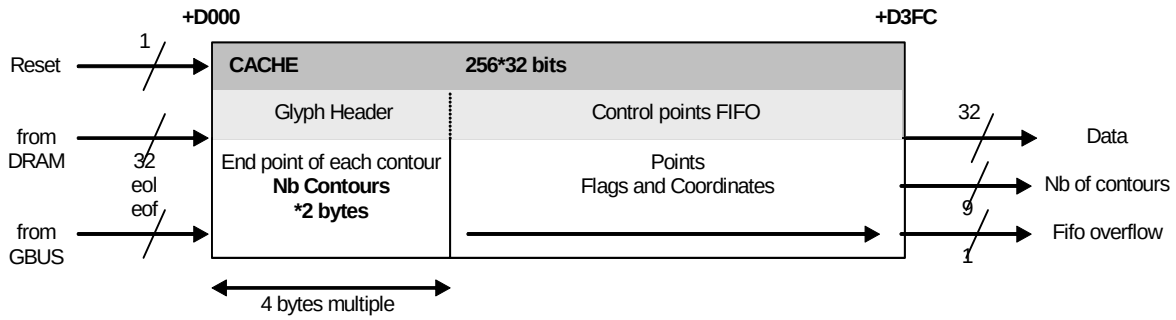
1. Reads a pre-parsed glyph from the DRAM (this data can also be provided directly by the G-Bus) and pre-processes the points stream.
2. Renders the glyph outline and fills the character.
3. A 1-bpp (black and white) bitmap is then generated in the DRAM.

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.

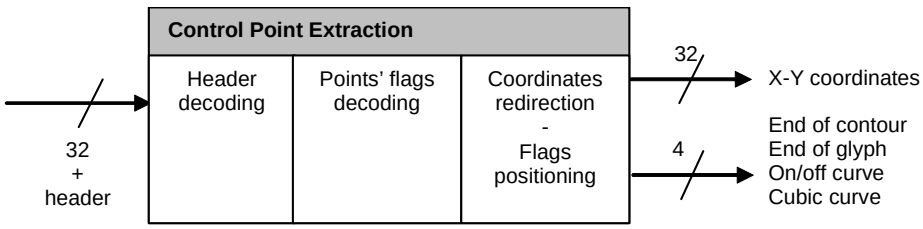
Cache

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 multiple of 4 bytes in order to be easily addressable. The outline control points information (flags and coordinates) is stored in a FIFO way in the remaining free part of the cache.



If the FIFO overflows with the points information, then the cache must signal it to the font rendering state machine so that the data can be read again from the DRAM for each working area.

Control Point Extraction



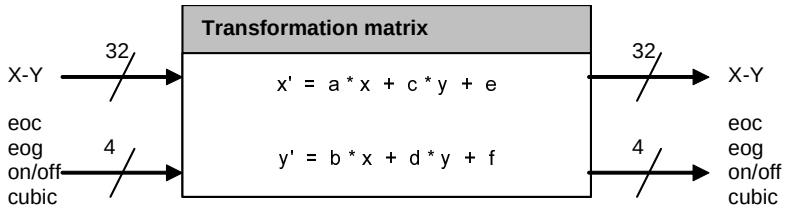
The ‘control point extraction’ module inputs the data stream read in the cache. The purpose is to separate the data into different buses relative to:

- 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.

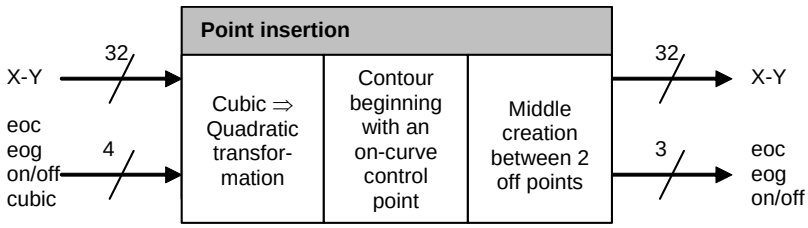
Transformation Matrix

In the case of a composite glyph, the points’ coordinates 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 scales (a, b, c, d) are 2.14 fixed point numbers (2-bit integer part, 14-bit decimal part). (e, f) offsets are 16-bit integers in FUnits, and are applied only for the first point of the glyph as the coordinates are still relatives at this point. Default values for a non-composite glyph must be 1, 0, 0, 1, 0, 0 (x’ = x and y’ = y).

Point Insertion



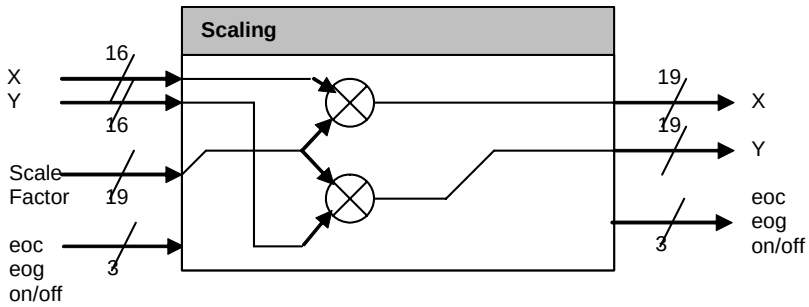
The module ‘point insertion’ inserts:

- 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).

Scaling

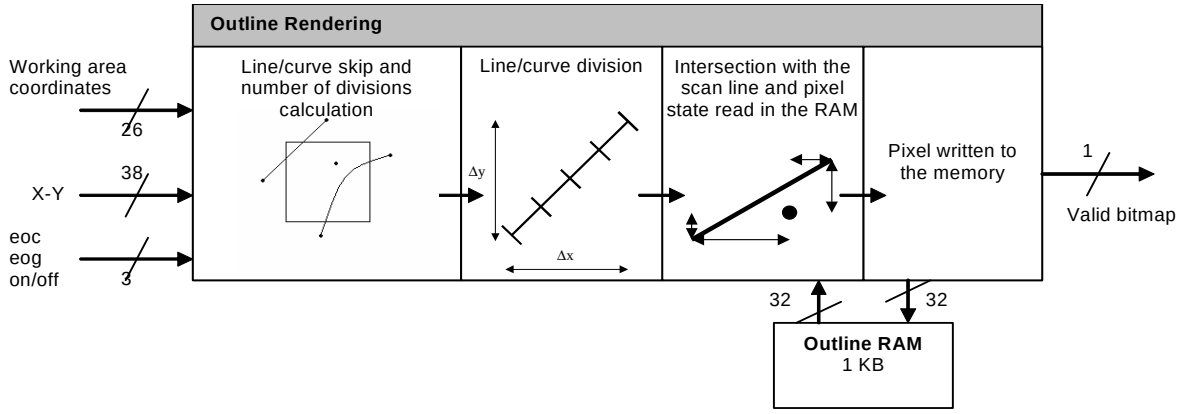
When scaling, the original coordinates in FUnits are converted into pixel units. This step is done by multiplying the coordinates by the scale factor.



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.

Outline Rendering



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.

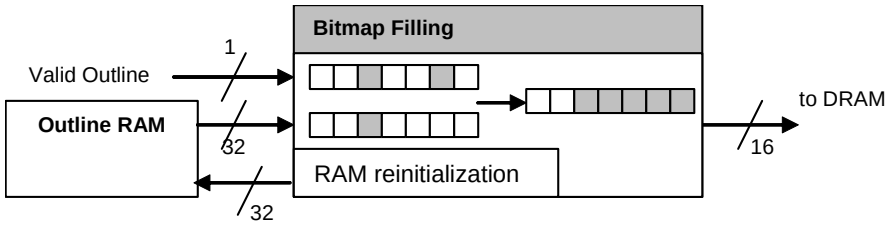
After a step of the lines or curves division into elementary segments, each intersection of these segments forming the outline with the working area scan lines is calculated. In the outline memory a 2-bit information for each touched pixel is stored.

- 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).

Bitmap Filling

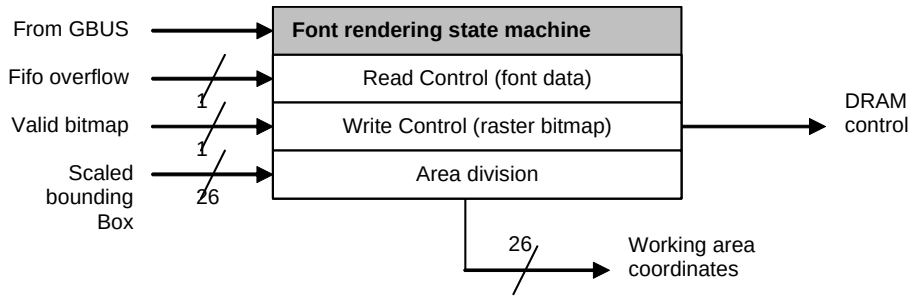
When the outline drawing is completed in the area, the final bitmap is filled and written to the DRAM through the W2 channel (1-bpp, black and white).



The filling is processed line by line from left to right and top to bottom in each 64x64 working area. The status of the end of lines is memorized for the next working area.

Font Rendering State Machine

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

8.4.10.5 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.

8.4.10.6 Porter-Duff Compatibility

The paper "Compositing Digital Images" (1984) by Thomas Porter & Tom Duff sets various rules for properly blending graphics objects, and is referenced by many specifications. While their rules use RGB data that is premultiplied by alpha (r/α, g/α, b/α), the SMP8634 uses non-premultiplied RGB data plus alpha (r,g,b,α).

CLEAR Rule

Figure 8-7 **CLEAR example**

CLEAR is described by the following equations:

- If source_alpha = 0,
  - destination_color = destination_color
  - destination_alpha = destination_alpha
- If source_alpha! = 0,
  - destination_color = destination_color
  - destination_alpha = 0

The alpha of the destination is cleared. When source_alpha!= 0, the CLEAR rule can be viewed as simply overwriting existing data with the source data having its alpha set to zero. This can be achieved by setting the accelerator to the “replace” mode and forcing the output alpha to 0.

SRC Rule

Figure 8-8 **SRC example**

SRC is described by the following equations:

- If source_alpha = 0,
  - destination_color = destination_color
  - destination_alpha = destination_alpha
- If source_alpha! = 0,
  - destination_color = source_color
  - destination_alpha = source_alpha

The source is copied to the destination. When source_alpha!= 0, the SRC rule can be viewed as simply overwriting existing data with the source data. This can be achieved by setting the accelerator to the “replace” mode.

DST_ IN and SRC_IN Rules

Figure 8-9 **DST_IN and SRC_IN examples**

DST_IN is described by the following equations:

- If source_alpha = 0,
  - destination_alpha = destination_alpha
  - destination_color = destination_color
- If source_alpha! = 0,
  - destination_alpha = destination_alpha * source_alpha
  - destination_color = destination_color

SRC_IN is described by the following equations:

- If source_alpha = 0,
  - destination_alpha = destination_alpha
  - destination_color = destination_color
- If source_alpha! = 0,
  - destination_alpha = destination_alpha * source_alpha
  - destination_color = source_color

For SRC_IN, the part of the source lying inside of the destination replaces the destination. For DST_IN, the part of the destination lying inside of the source replaces the destination. When source_alpha!= 0, the DST_IN and SRC_IN rules can be viewed as a modulation of the alpha of object A by the alpha of object B, with the color of A remaining the same. This can be achieved by reading object A through the Y channel, object B through the X channel and by setting the accelerator to the “move+merge+modulate” mode. The output is then determined as follows:

- color = color from Y channel (object A)
- alpha = alpha from X channel (object B) multiplied by alpha of Y channel (object A)

Y channel can be replaced with Z channel and it is possible to input {alpha,color} or {alpha}-only information in the X channel.

DST_OUT and SRC_OUT Rules

Figure 8-10 **DST_OUT and SRC_OUT examples**

DST_OUT is described by the following equations:

- If source_alpha = 0,
  - destination_alpha = destination_alpha
  - destination_color = destination_color
- If source_alpha! = 0,
  - destination_alpha = destination_alpha * (1-source_alpha)
  - destination_color = destination_color

SRC_OUT is described by the following equations:

- If source_alpha = 0,
  - destination_alpha = destination_alpha
  - destination_color = destination_color
- If source_alpha! = 0,
  - destination_alpha = source_alpha * (1-destination_alpha)
  - destination_color = source_color

For SRC_OUT, the part of the source lying outside of the destination replaces the destination. For DST_OUT, the part of the destination lying outside of the source replaces the destination. When source_alpha!= 0, the DST_OUT and SRC_OUT rules are similar to the “IN” rules, except that the complement of the alpha of object B is used to modulate the alpha of object A. Two passes are necessary to implement these rules using the accelerator:

Pass 1: Extract object B alpha and invert it.

a. Input the object B via the Y channel.
b. Fill the Y channel alpha LUT in the reverse order [LUT(0)=255,LUT(1)=254, etc.].
c. Fill the Y channel color LUT to FF for each component.
d. Set the Z channel to use the gradient generator. Configure the latter to generate a rectangle of dimensions similar to object B and with the color set to 00 per component.
e. Blend Y and Z using the basic blending mode. The color output by the blender is, $(!\alpha_B,!\alpha_B,!\alpha_B\}$.
f. Set the V-Bus write channel in 8-bpp mode. It will build a 8-bpp alpha map in DRAM that contains$!\alpha_B$.

Pass 2: Read object A through Y or Z channel and the alpha map extracted in pass 1, and configure the accelerator to move+merge+modulate mode.

DST_OVER and SRC_OVER Rules

Figure 8-11 **DST_OVER and SRC_OVER examples**

SRC_OVER is described by the following equations:

- If source_alpha = 0,
  - destination_color = destination_color
  - destination_alpha = destination_alpha

- If source_alpha = 0,
  - destination_color = source_color + [destination_color * (1 - source_alpha)]
  - destination_alpha = source_alpha + destination_alpha - (source_alpha * destination_alpha)

DST_OVER is described by the following equations:

- If source_alpha = 0,
  - destination_color = destination_color
  - destination_alpha = destination_alpha
- If source_alpha = 0,
  - destination_color = destination_color + [source_color * (1 - destination_alpha)]
  - destination_alpha = source_alpha + destination_alpha - (source_alpha * destination_alpha)

For SRC_OVER, the source is composited over the destination. For DST_OVER, the destination is composited over the source and the result replaces the destination. The DST_OVER and SRC_OVER rules are implemented using the “full” blend command of the accelerator. The hardware does not differentiate between the DST_OVER and SRC_OVER, the command is symmetric from that point of view.

8.4.11 Main Mixer

The main mixer receives the picture streams from each of the following 8 sources. The global plane order (layer priority) is programmable, except for the hardware cursor which is always the top layer (highest priority).

1. Hardware cursor
2. OSD scaler
3. Subtitle scaler
4. Multi-format scaler 1
5. Multi-format scaler 2
6. Main video scaler
7. Video plane channel
8. Video/graphics input port

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.

8.4.11.1 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:

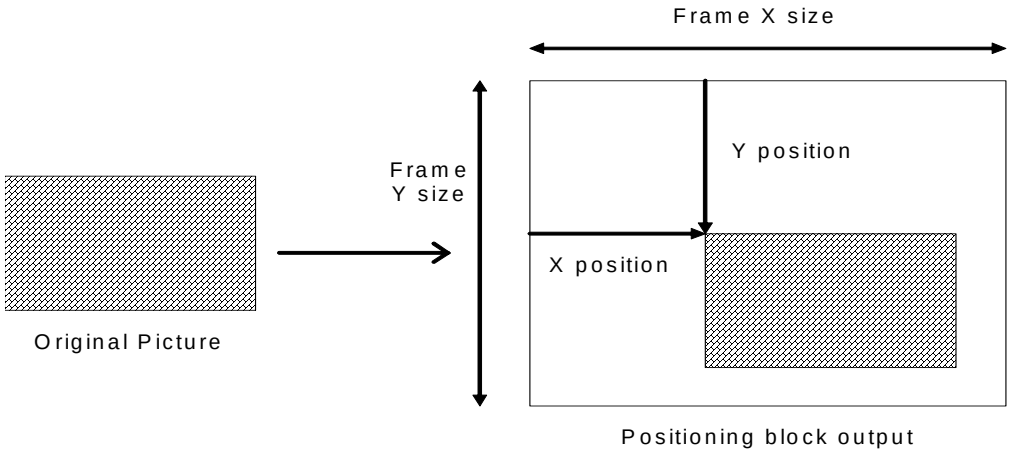


Figure 8-12 **Positioning blocks**

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 bandwidth problems will appear. In this case, the input picture should be cut in the V-Bus interface.

8.4.11.2 Mixing

Within the mixer, the positioned streams can be assigned to any of the layers 0-6. The highest priority (layer 7, the top layer) is always assigned to the hardware cursor.

For each output pixel, the 8 input pixels are layered and mixed according to the programmed layer (global priority) order. Each output pixel from the main mixer ultimately consists of the alpha-blended combination of the 4 highest-priority non-transparent input pixels.

At any pixel location where there's no data provided by the sources, or where all the sources are transparent, a programmable background color (typically black) is output. The output of the main mixer is then sent to the display routing block.

8.4.11.3 Clear Rectangle - HD DVD-Video

Due to the additional memory bandwidth requirements of using the 2D graphics accelerator to implement Clear Rectangle on active video, the SMP8634 cannot support the HD-DVD Clear Rectangle feature on the secondary video plane.

8.4.12 Display Routing Block

The display router is essentially a crossbar switch, which couples 4 input video streams to any of the 3 video output ports. The 4 input sources to the display router are the main mixer output, the Multi-format scaler 1 output, the HD to SD downscaler output and the color bars generator.

Each video output port (digital video output /component video output/CVBS and s-video video output) is either disabled or assigned one source. Each source can thus have from 0 to 3 associated video outputs. It should be made sure that the video outputs sharing the same source are timing compatible. This arrangement enables simultaneous HD and SD outputs or the ability to output two programs simultaneously (HD+HD, HD+SD or SD+SD).

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

8.4.13 Video Output Interfaces

8.4.13.1 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 HSYNC and VSYNC from an internal video clock. In the slave mode, the display controller receives HSYNC and VSYNC from another video input/output port or an external device.

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 @ 23.976, 24, 50, 59.94 and 60Hz

On rev. B and later, the output video may be optionally rotated by 90, 180 or 270 degrees, with symmetry with respect to the X axis, 1st bisector, Y axis or 2nd bisector.

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.

8.4.13.2 Frame Rate Conversion

The programmable output video timing, in addition to the 2:2 and 3:2 pulldown processing and deinterlacing, enables the following source-to-output frame rate conversions to be supported:

Table 8-6 **Supported source and output frame rate combinations**

Source/Output	23.976p	24p	25i	29.97i 30i	50p	59.94p 60p
23.976p	Yes	Yes	Yes	Yes	Yes	Yes
24p	Yes	Yes	Yes	Yes	Yes	Yes
25i[1]	-	-	Yes	Yes	Yes	Yes
25p	-	-	Yes	Yes	Yes	Yes
29.97i[1]	-	-	Yes	Yes	Yes	Yes
29.97p	-	-	Yes	Yes	Yes	Yes
30i[1]	-	-	Yes	Yes	Yes	Yes
30p	-	-	Yes	Yes	Yes	Yes
50p	-	-	Yes	Yes	Yes	Yes
59.94p	-	-	Yes	Yes	Yes	Yes
60p	-	-	Yes	Yes	Yes	Yes

1. 25i, 29.97i and 30i are also commonly called 50i, 59.94i and 60i, respectively. The difference is whether frame rate or field rate is specified.

Each field/frame from the video decoder has a PTS and is compared with the STC to decide what is the best field/frame to display.

When frame rate converting interlaced sources, the output is either the source field closer in time (resulting in occasional dropped or duplicated fields) or has a matching phase (resulting in occasional dropped or duplicated frames).

For 24p and 23.976p sources, the 3:2 pulldown info is used as a hint to determine if a frame should be displayed again or not; the PTS and STC have the final say. For example, if displaying a frame 3 times leads to a large gap between the PTS and the STC, then the frame will be displayed twice. If there is no repeat_first_field flag, the display is completely PTS/STC driven - the display sequence is the same as if there is the flag but the first repeated frame is random, instead of being the one set in the stream.

Table 8-7 **Combinations of supported output frame rates - OSD, subtitles and graphics available on all outputs**

	Supported Output Frame Rates							
HDMI	23.976p	24p	25i	29.97i	30i	50p	59.94p	60p
Analog Component[1]			25i	29.97i	30i	50p	59.94p	60p
Analog Component[2]	29.97i	25i 29.97i	25i	29.97i	29.97i	25i	29.97i	29.97i
S-Video	29.97i	25i 29.97i	25i	29.97i	29.97i	25i	29.97i	29.97i
Composite	29.97i	25i 29.97i	25i	29.97i	29.97i	25i	29.97i	29.97i

1. Resolution of HDMI and analog component video is the same.
2. Resolution and frame rate of analog component video and composite video is the same.

Table 8-8 **Combinations of supported output frame rates - no OSD, subtitles or graphics on analog component outputs**

	Supported Output Frame Rates							
HDMI	23.976p	24p	25i	29.97i	30i	50p	59.94p	60p
Analog Component[1,2]	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i	60p 59.94p 50p 30i 29.97i 25i
S-Video	29.97i	25i 29.97i	25i	29.97i	29.97i	25i	29.97i	29.97i

Table 8-8 **Combinations of supported output frame rates - no OSD, subtitles or graphics on analog component outputs (Continued)**

	Supported Output Frame Rates							
Composite	29.97i	25i 29.97i	25i	29.97i	29.97i	25i	29.97i	29.97i

1. Resolution of HDMI and analog component video may be different.
2. Requires additional memory bandwidth to read the main video plane a second time.

8.4.13.3 VBI Support on Video Outputs

Sigma's hardware library provides for the extraction, parsing, and output of various user data types onto the video outputs during the vertical blanking interval (VBI).

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

480i closed captioning (including extended data services, XDS such as parental control, v-chip, etc.) on lines 21 and 284, as defined by CEA-608-C is supported. There is no standard for transmitting closed captioning over the HDMI or 480p, 576i, 576p, 720p, 1080i or 1080p analog component interfaces.

The registers VO_MAIN_ANALOG_CC_AGC (CC_data[15:0]), VO_COMPONENT_OUT_TV_CONFIG (bits 17 and 18), VO_COMPOSITE_OUT_TV_CONFIG (bits 17 and 18) and VO_MAIN_ANALOG_TV_CONFIG (bits 17 and 18) control closed captioning. The bit 18 determines the extended closed captioning.

Closed captioning data (including XDS data) to be output during VBI may be provided by application software on the host CPU or by Sigma's hardware library.

The hardware library supports the following closed captioning formats,

- DVD
  - MPEG-2
    * User_data_start_code = 0x000001B2
    * User data type = 0x4343

- HD-DVD
  - MPEG-2
    * User_data_start_code = 0x000001B2
    * Closed_caption_indicator = 0x4343
  - MPEG-4.10 (H.264)
    * Uuid_iso_iec_11578 = 0xFE80A41042A911D996690008DFF5B00
    * Closed_caption_indicator = 0x4343
  - SMPTE 421M (VC-1)
    * User_data_start_code = 0x0000011E
    * Closed_caption_indicator = 0x4343
- Blu-ray Disc (BD)
  - MPEG-2
    * User_data_start_code = 0x000001B2
    * User_data_type_code = 0x03
    * Cc_type = 0x0 or 0x1
  - MPEG-4.10 (H.264)
    * Uuid_iso_iec_11578 = 0x17ee8c60-f84d-11d9-8cd6-0800200c9a66
    * Type_indicator = 0x47413934
  - SMPTE 421M (VC-1)
    * User_data_identifier = 0x48444D56
    * Type_indicator = 0x47413934
- ATSC
  - MPEG-2
    * User_data_start_code = 0x000001B2
    * User_data_type_code = 0x03
    * Cc_type = 0x0 or 0x1
- SCTE 20 and 21
  - MPEG-2
    * User_data_start_code = 0x000001B2
    * User_data_type_code = 0x03
    * Cc_type = 0x0 or 0x1
- Minerva

Nielsen AMOL I and II

480i Nielson AMOL I and II data on lines 20 and 22, as defined by CEA-805-C, is not supported.

Multi-level Pulse-Amplitude Modulation (PAM)

Multi-level pulse-amplitude modulation (PAM), used to reconstruct a VBI analog waveform, is not supported on the SMP8634.

Wide Screen Signaling and Copy Protection Control

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

480p wide screen signaling (WSS, copy generation management (CGMS-A) and APS data on line 41, as defined by EIAJ CPR1204-1, CEA-805-C Type A and IEC 61880-2, is supported. On rev. C and later, CGMS-A, RCI and APS data as defined by CEA-805-C Type B on line 40 is also supported.

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

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

720p copy generation management (CGMS-A) and APS data on line 24, as defined by CEA-805-C Type A and EIAJ CPR1204-2, is supported. On rev. C and later, CGMS-A, RCI and APS data as defined by CEA-805-C Type B on line 23 is also supported.

1080i copy generation management (CGMS-A) and APS data on lines 19 and 582, as defined by CEA-805-C Type A and EIAJ CPR1204-2, is supported. On rev. C and later, CGMS-A, RCI and APS data as defined by CEA-805-C Type B on lines 18 and 581 is also supported.

The register VO_MAIN_ANALOG_CGMS (R/W) controls Wide Screen Signaling and CGMS.

Table 8-9 **Wide Screen Signaling and CGMS - register VO_MAIN_ANALOG_CGMS**

Bit #	1	0
23	CGMS/WSS odd field (NTSC) / WSS (PAL)	CGMS/WSS disabled on odd fields / WSS disabled
22	CGMS/WSS on even fields (NTSC)	CGMS/WSS disabled on even fields
21	Software CRC generation for CGMS/ WSS(NTSC)	Hardware CRC generation for CGMS/ WSS(NTSC)
20	x	x

Table 8-9 **Wide Screen Signaling and CGMS - register VO_MAIN_ANALOG_CGMS**

Bit #	1	0
19:0	Copy Generation Management System data and WSS (NTSC)	
13:0	Wide Screen Signaling Data (PAL)	

WSS and CGMS data to be output during VBI must be provided by application software on the host CPU.

Teletext

The SMP8634 supports 3 systems:

- System B-625: for 625-line PAL systems in Europe
- System C-525: for 525-line NTSC systems in USA
- System D-525: for 525-line NTSC systems in Japan

The active teletext lines are located in the vertical blanking interval:

- For system B-625 (625 lines/frame):
  - Top field line 6 to line 22 (17 lines max)
  - Bottom field line 318 to line 335 (18 lines max)
- For systems C-525 and D-525 (525 lines/frame):
  - Top field line 7 to line 17 (11 lines max)
  - Bottom field line 269 to line 280 (12 lines max)

The teletext output is configured by the register VO_MAIN_ANALAOG_TV_TELETEXT_CONFIG. There are 2 parameters, CONFIG_LINE[17:0] and CONFIG_CODE[7:0]

CONFIG_LINE[17:0] enables the active teletext lines within the blanking: this is an enable (high) register. Bit 0 corresponds to the first line of teletext as defined by the system and the polarity of the field (see previous paragraph), and so on. For instance in system B, in a top field, programming bit 0 at 1 will enable teletext on line 6.

CONFIG_CODE[7:0] (or framing code) identifies the system. It is,

- 8’b 0010_0111 for system B-625
- 8’b 1110_0111 for system C-525
- 8’b 1010_0111 for system D-525

Bit 0 is output first. Each system outputs a different number of bits per line. Each line starts with a hardware generated clock run-in of 16 bits (alternating 0s & 1s), followed by the framing code byte. After these 3 bytes are output, the content of the teletext memory is output.

The maximum data lengths for each system are:

- System B-625: 42 bytes (+2 run-in +1 framing code=45)
- System C-525: 33 bytes (+2 run-in +1 framing code=36)
- System D-525: 34 bytes (+2 run-in +1 framing code=37)

Teletext data to be output during VBI must be provided by application software on the host CPU. Sample software is available from Sigma that extracts DVB teletext data (EN 300 472) and outputs it onto the 576i analog video outputs during VBI.

8.4.13.4 Sync Timing Generators

The programmable sync timing generators enable support for a wide variety of video timing standards, including:

- EIA-770.2 and 770.3
  - 480i30, 480p60, 720p60 and 1080i30
- SMPTE 170M
  - NTSC-M
- SMPTE 267M
- SMPTE 274M
  - 1080i25, 1080i30, 1080p24, 1080p25, 1080p30, 1080p50 and 1080p60
- SMPTE 296M
  - 720p24, 720p25, 720p30, 720p50 and 720p60
- ITU-R BT.470
  - 480i30 and 576i25

- ITU-R BT.601
  - 480i30 and 576i25
- ITU-R BT.656
  - 480i30 and 576i25
- ITU-R BT.709
  - 1080i25, 1080i30, 1080p24, 1080p25, 1080p30, 1080p50 and 1080p60
- ITU-R BT.1358
  - 480p60 and 576p50

Custom timing to support various flat panel display resolutions, such as 1366x768, is also possible.

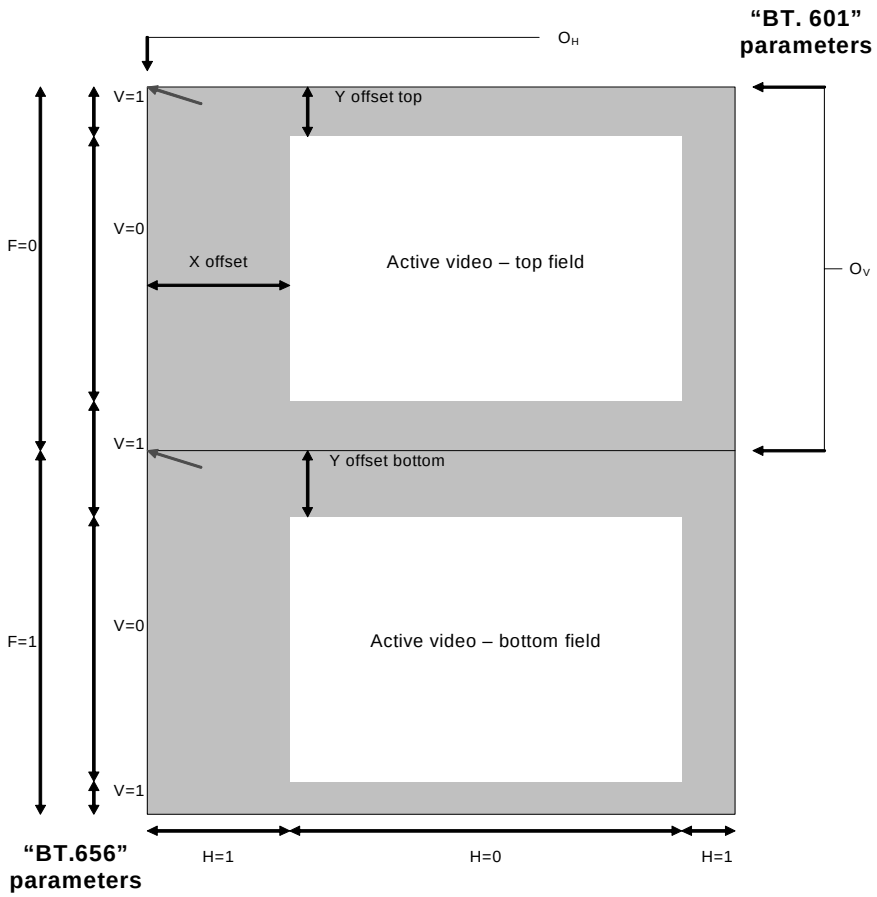


Figure 8-13 **Video output timing parameters**

The range of programmability for each video timing parameter for the digital video output port is as follows:

Table 8-10 **Range of programmability for digital video output timing**

Parameter	Range
Vt_total_size (vertical total per fame)	0 - 4095 lines
Hz_total_size (horizontal total per line)	0 - 4095 pixels
X_offset (Hsync to active video)	0 - 4095 pixels
Y_offset_top (Vsync to active video, top field)	0 - 4095 lines
Y_offset_bottom (Vsync to active video, bottom field)	0 - 4095 lines
Top_field_height (total size of top field)	0 - 8191 (half lines)
Hz_sync_width (Hsync width)	0 - 4095 pixels
Vt_sync_width (Vsync width)	0 - 8191 half lines
Vt_sync_fine_adjust, lower byte (Vsync to Hsync relative position adjust, top field)	0 to +/-127pixels
Vt_sync_fine_adjust, upper byte (Vsync to Hsync relative position adjust, bottom field)	0 to +/-127 pixels

The range of programmability for each video timing parameter for the component and composite/s-video output ports is as follows:

Table 8-11 **Range of programmability for component, composite and s-video output timing**

Parameter	Range
Height (total line per frame)	0 - 4095 lines
Width (total pixels per line)	0 - 4095 pixels
Hsync0 (horizontal coordinate where Hsync goes low)	0 - 4095 pixels
Hsync1 (horizontal coordinate where Hsync goes high)	0 - 4095 pixels
Hsync_width (total Hsync width)	0 - 4095 pixels
Vsync_O_0_line (vertical coordinate where Vsync goes low for odd field)	0 - 4095 lines
Vsync_O_1_line (vertical coordinate where Vsync goes high for odd field)	0 - 4095 lines
Vsync_E_0_line (vertical coordinate where Vsync goes low for even field)	0 - 4095 lines
Vsync_E_1_line (vertical coordinate where Vsync goes high for even field)	0 - 4095 lines
Vsync_O_0_pixel (horizontal coordinate where Vsync goes low for odd field)	0 - 4095 pixels

Table 8-11 **Range of programmability for component, composite and s-video output timing (Continued)**

Parameter	Range
Vsync_O_1_pixel (horizontal coordinate where Vsync goes high for odd field)	0 - 4095 pixels
Vsync_E_0_pixel (horizontal coordinate where Vsync goes low for even field)	0 - 4095 pixels
Vsync_E_1_pixel (horizontal coordinate where Vsync goes high for even field)	0 - 4095 pixels
Vsync_start (horizontal coordinate where Vsync starts)	0 - 4095 lines
Vsync_width (total Vsync width)	0 - 4095 lines

Note: The appendix includes a listing of parameter values for standard video timings.

8.4.13.5 Analog Video Output

The analog video output block controls six 12-bit video DACs to provide video in the composite, S-video (Y/C) and component (RGB or YPbPr) formats. The block contains 2 independent video timing generators, one for the component outputs and one for the composite/s-video outputs. Each timing generator can operate in a master or slave timing mode.

One set of 3 video DACs (the VO2 block) provides the component video output (selectable as either RGB or YPbPr). The other set of 3 video 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 video DAC can be independently disabled when not used for significant power savings.

For SD sources, the SMP8634 offers the ability to output composite, s-video, SD or upscaled HD analog component, and SD or upscaled HD HDMI outputs simultaneously. For HD sources, downscaled composite, downscaled s-video, HD or downscaled SD analog component outputs, and HD or downscaled SD HDMI outputs are simultaneously available. For two SD or HD sources, the SMP8634 also offers the ability to output both sources simultaneously.

The DRM software running on the XPU may restrict the ability to output upscaled HD analog component outputs from SD sources or the ability to output HD analog component outputs from HD sources.

Component Video Outputs

The component video outputs support optional Macrovision v7.1.L1 and v1.2 protection and DAC attenuation compensation. When generating 480i or 576i video, the DACs operate at 54MHz; for 480p or 576p video, the DACs operate at 108MHz; for 1920x1080i or 1280x720p video, the DACs operate at 74.25MHz. For 1920x1080p video, the DACs operate at 148.5MHz. The analog video outputs are capable of driving a doubly-terminated 75-ohm load.

The Macrovision v7.1.L1 and v1.2 certifications that Sigma receive applies only to chip manufacturers. Other certifications (such as v7.1.S1, etc.) are implementation-dependent (i.e. they are dependent on the type of box being produced) and therefore are up to the OEM/ODM to obtain.

Component Video Output Timing

The block contains a complete programmable sync timing generator for operating as a sync master, or the timing generator may be slaved to H/V sync timing from,

- Video input
- Graphics input
- Composite/s-video analog output
- Digital video output
- External device connected to component video output port

H/V sync signals output onto this port may be obtained from,

- Sync timing generator

Either bi-level or tri-level sync pulses may be generated when outputting high-definition RGB or YPbPr signals.

Component Video Output Processing

A 3x3 matrix multiplier with programmable coefficients is used to implement:

- YCbCr/RGB color space conversion
- Contrast (or ‘picture’ or ‘white level’) control
- Saturation (or ‘color’) control
- Hue (or ‘tint’) control

- BT.709 (HDTV) to BT.601 (SDTV) and BT.601 (SDTV) to BT.709 (HDTV) colorimetry conversion

The inputs of the matrix multiplier are 24-bit YCbCr from the display routing block; the outputs are 36-bit YCbCr or RGB which drives the three 12-bit video DACs. The matrix coefficients have a range of -4 to +3.4095/4096, in steps of 1/4096. The matrix multiplier is followed by three programmable DC offsets to provide optional brightness (or 'black level') control. The DC offsets have a range of -512 to + 511.15/16, in steps of 1/16. No sharpness control is provided for the component video outputs.

Component Video Output Formats

Supported component output formats are:

- RGB SCART (IEC 60933)
  - RGB + CVBS (composite)
  - RGB: 0.700v range
  - 0 IRE blanking setup
  - No sync on R, G and B
  - No support for fast blanking signal
- RGB sync-on-green
  - 0.300v bi-level sync (SD) or +/-0.300v tri-level sync (HD)
  - RGB: 0.700v range
  - 0 IRE blanking setup
- RGB SMPTE
  - 0.300v bi-level sync (SD) or +/-0.300v tri-level sync (HD)
  - RGB: 0.700v range
  - 0 IRE blanking setup
  - sync on R, G and B
- YPbPr Betacam
- YPbPr MII
- YPbPr SMPTE
  - 0.300v bi-level sync (SD) or +/-0.300v tri-level sync (HD)
  - Y: 0.700v range
  - Pb, Pr: +/-0.350v range
  - 0 IRE blanking setup
  - sync on Y, Pb and Pr

- YPbPr EIA-770.2 and EIA-770.3
  - 0.300v bi-level sync (SD) or +/-0.300v tri-level sync (HD)
  - Y: 0.700v range
  - Pb, Pr: +/-0.350v range
  - 0 IRE blanking setup
  - no sync on Pb and Pr

Component Video Output Configuration Options

Configuration register bits can be used to select a number of features and characteristics:

- DAC enable or power-down
- H- and V-sync source
- H- and V-sync polarity
- Bi-level or tri-level sync pulses for HD output
- SD or HD output
- Progressive or interlaced output
- Closed caption and extended closed caption enable/disable
  - CEA-608-C data is present on analog RGB outputs instead of only the G output
- Y and RGB filter options
  - 9.00 MHz
  - 13.0 MHz
- PbPr filter options
  - 6.50 MHz
  - 13.0 MHz
- YPbPr/RGB output mode

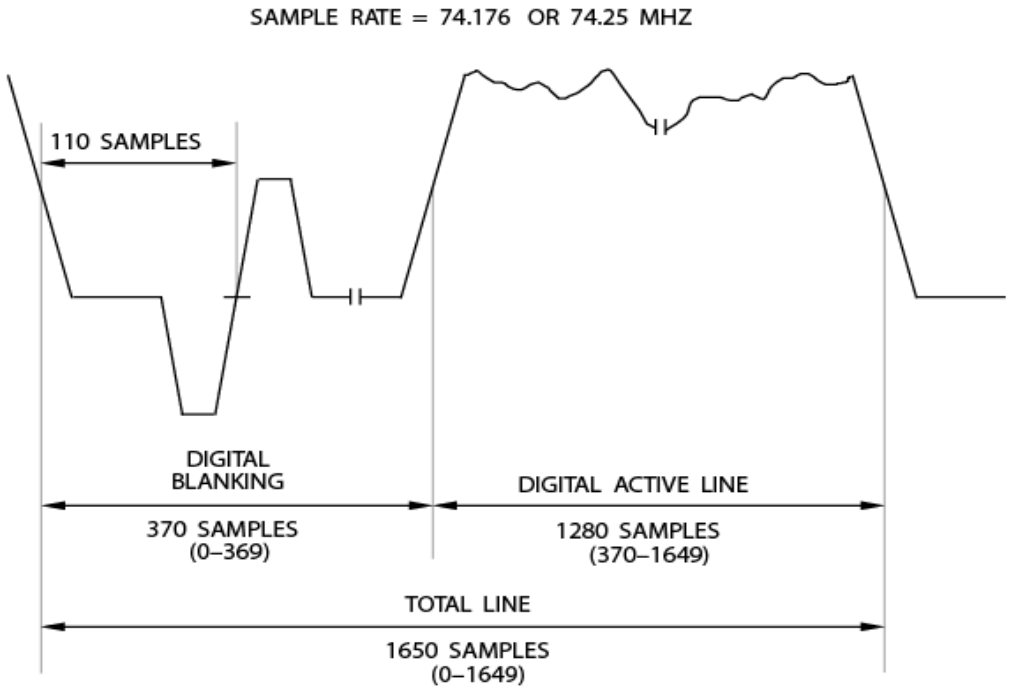


Figure 8-14 **Video waveform - 720p**

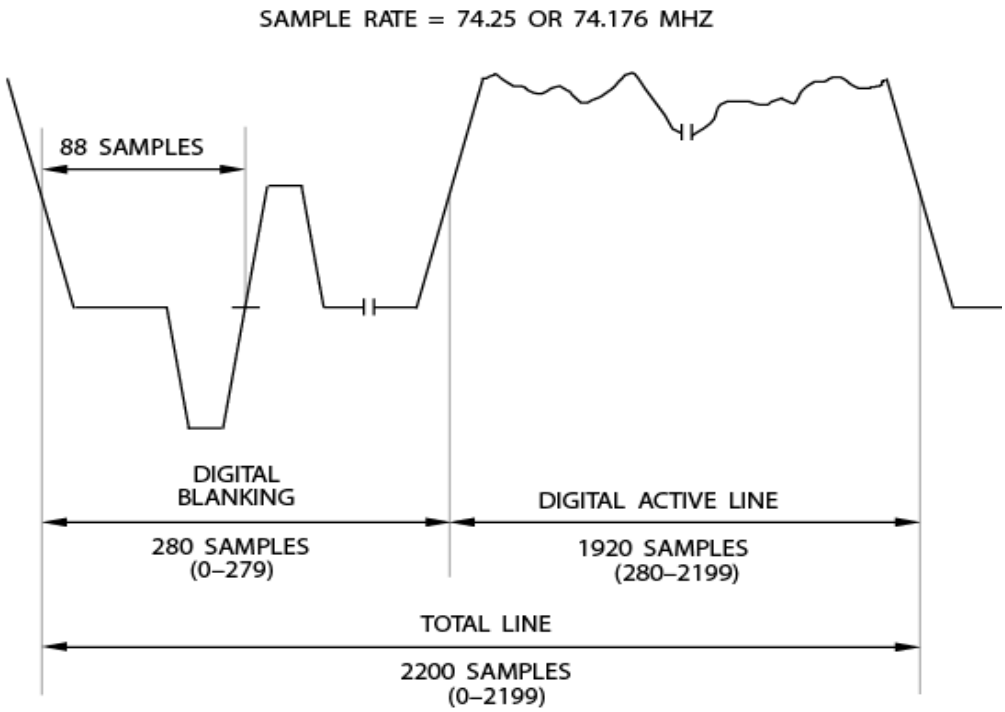


Figure 8-15 **Video waveform - 1080i**

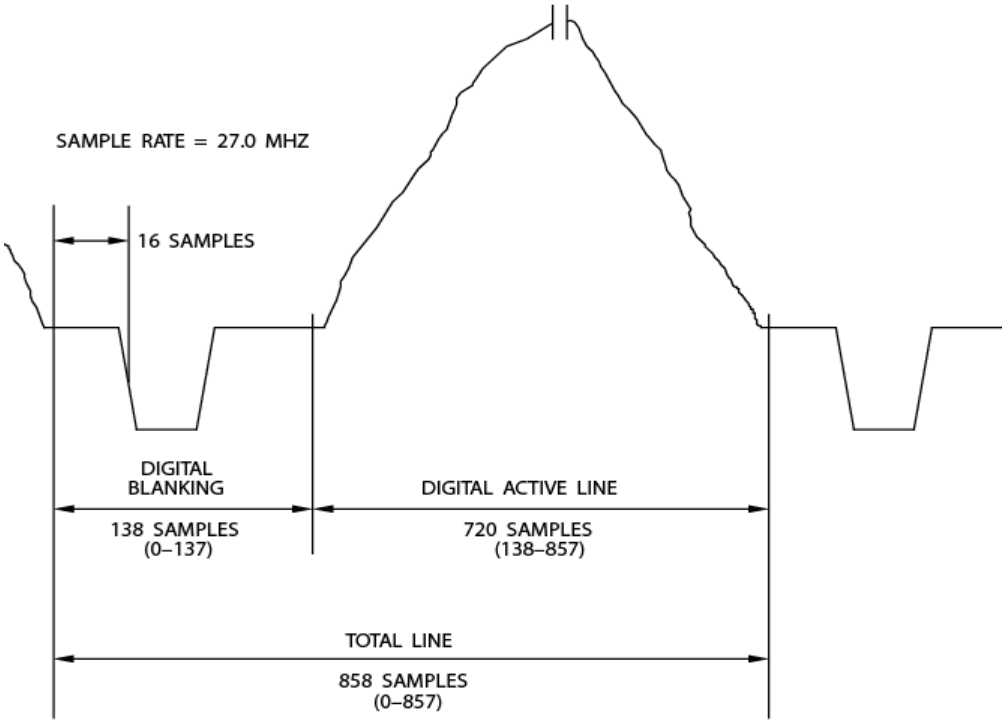


Figure 8-16 **Video waveform - 480i**

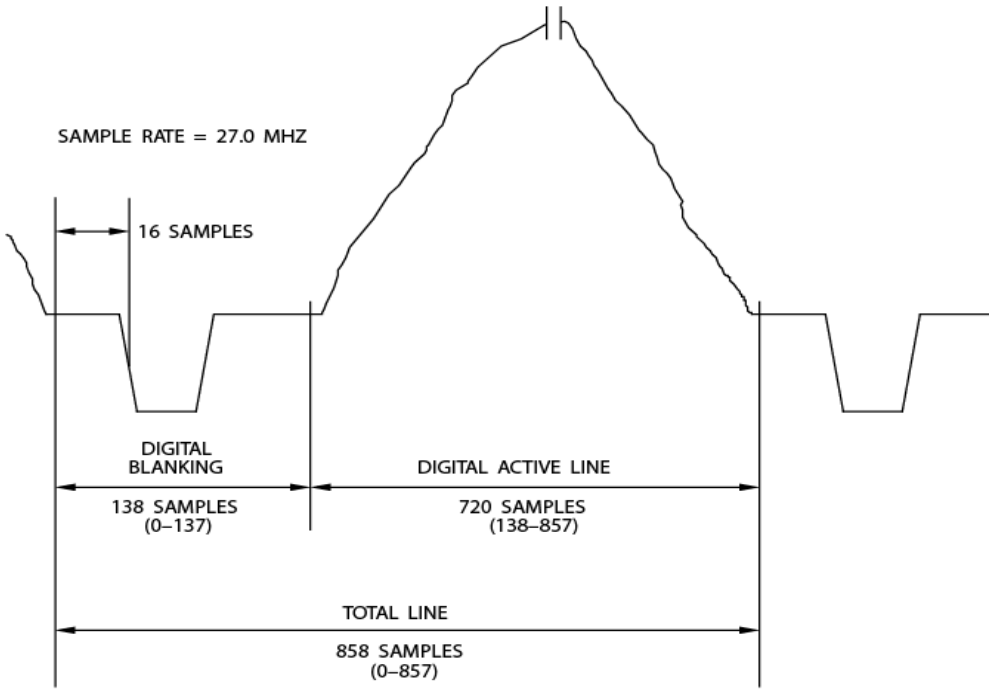


Figure 8-17 **Video waveform - 480p**

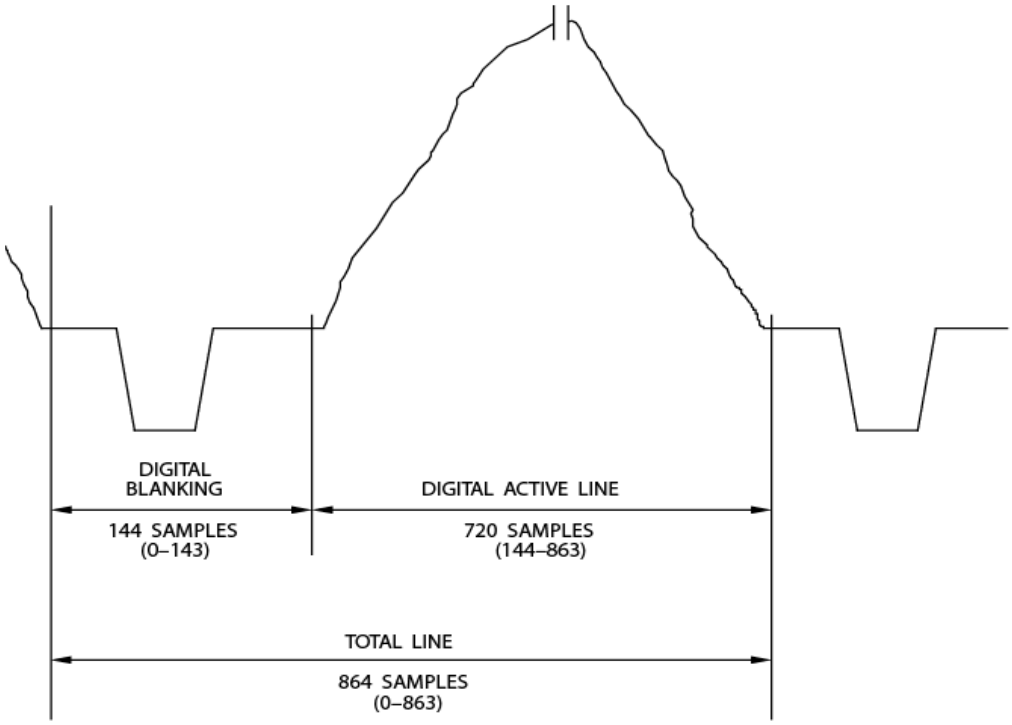


Figure 8-18 **Video waveform - 576i**

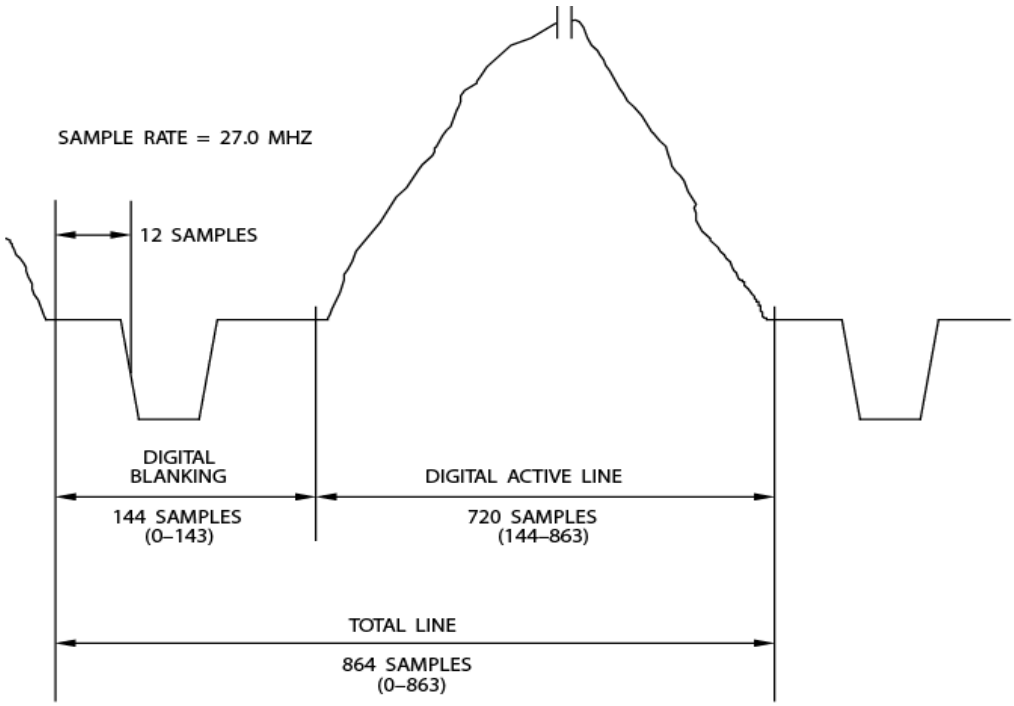


Figure 8-19 **Video waveform - 576p**

Composite (CVBS) and S-Video Outputs

The NTSC/PAL encoder (with optional Macrovision v7.1.L1 protection and DAC attenuation compensation) supports the NTSC-M, NTSC-J, PAL-B/D/G/H/I, PAL-60 and PAL-M baseband video standards. It uses three 54MHz 12-bit video DACs to generate simultaneous composite and s-video outputs. The analog video outputs are capable of driving a doubly-terminated 75-ohm load.

The Macrovision v7.1.L1 certification that Sigma receives applies only to chip manufacturers. Other certifications (such as v7.1.S1, etc.) are implementation-dependent (i.e. they are dependent on the type of box being produced) and therefore are up to the OEM/ODM to obtain.

Composite and S-Video Output Timing

The block contains a complete programmable sync timing generator for operating as a sync master, or the timing generator may be slaved to H/V sync timing from,

- Component video analog output
- Digital video output
- External device connected to composite and S-video output port

H/V sync signals output onto this port may be obtained from,

- Sync timing generator

Composite and S-Video Output Processing

A 3x3 matrix multiplier with programmable coefficients is used to implement:

- Contrast (or ‘picture’ or ‘white level’) control
- Saturation (or ‘color’) control
- Hue (or ‘tint’) control
- BT.709 (HDTV) to BT.601 (SDTV) colorimetry conversion

The inputs of the matrix multiplier are 24-bit YCbCr from the display routing block; the outputs are 30-bit YCbCr which drives the NTSC/PAL encoder. The matrix coefficients have a range of -4 to +3.4095/4096, in steps of 1/4096. The matrix multiplier is followed by three programmable DC offsets to provide optional brightness (or ‘black level’) control. The DC offsets have a range of -512 to + 511.15/16, in steps of 1/16. The sharpness can be adjusted from +/-12.5% to 100% in 8 steps. A color sub carrier notch filter is also available.

The composite and s-video digital outputs from the NTSC/PAL encoder are 10-bit data. Oversampling filters output 12-bit data, which drives the three 12-bit video DACs.

When outputting 720 samples/line, either all 720 active samples may be output (does not meet the NTSC/PAL timing) or, 1-16 samples may be cropped from the left and right sides resulting in the center 688-720 active samples being output (in order to meet the NTSC/PAL timing). Typically, to meet the NTSC timing, 716 samples should be output and to meet the PAL timing, 702 samples should be output.

Composite and S-Video Output Formats

Supported composite and s-video output formats are:

- Baseband NTSC-M
- Baseband NTSC-J (NTSC-M for Japan)
- Baseband PAL-B/D/G/H/I
- Baseband PAL-M
- Baseband PAL-60

Composite and S-Video Output Configuration

Configuration register bits can be used to select a number of features and characteristics:

- DAC enable or power-down
- H- and V-sync source
- H- and V-sync polarity
- Closed caption and extended closed caption enable/disable
- CGMS, WSS and Teletext enable/disable
- Sharpness/notch filter enable/disable and gain/attenuation control
- Luma filter options
  - 4.5 MHz (driving RF modulator)
  - 6.5 MHz
- Chroma filter options
  - 0.65 MHz (480i out, driving RF modulator)
  - 1.00 MHz (576i out, driving RF modulator)
  - 1.30 MHz (480i out)
  - 2.00 MHz (576i out)

- Y-C relative delay
  - No delay
  - Delay C by 1, 2 or 3 sample clocks
  - Delay Y by 1, 2 or 3 sample clocks
- Composite and s-video output format
- Output three composite video signals instead of one composite and one s-video

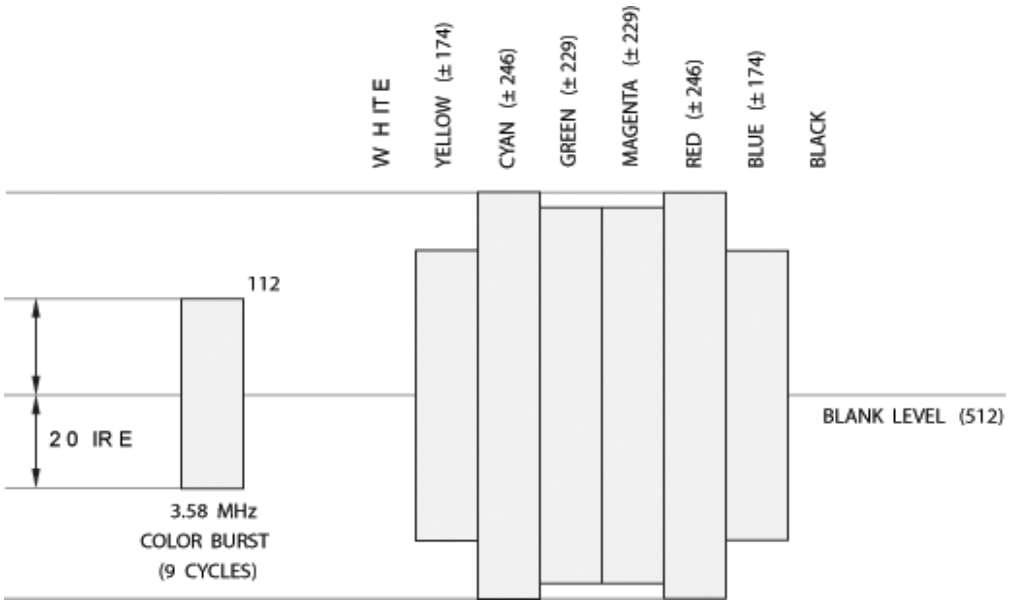


Figure 8-20 **NTSC c for s-video**

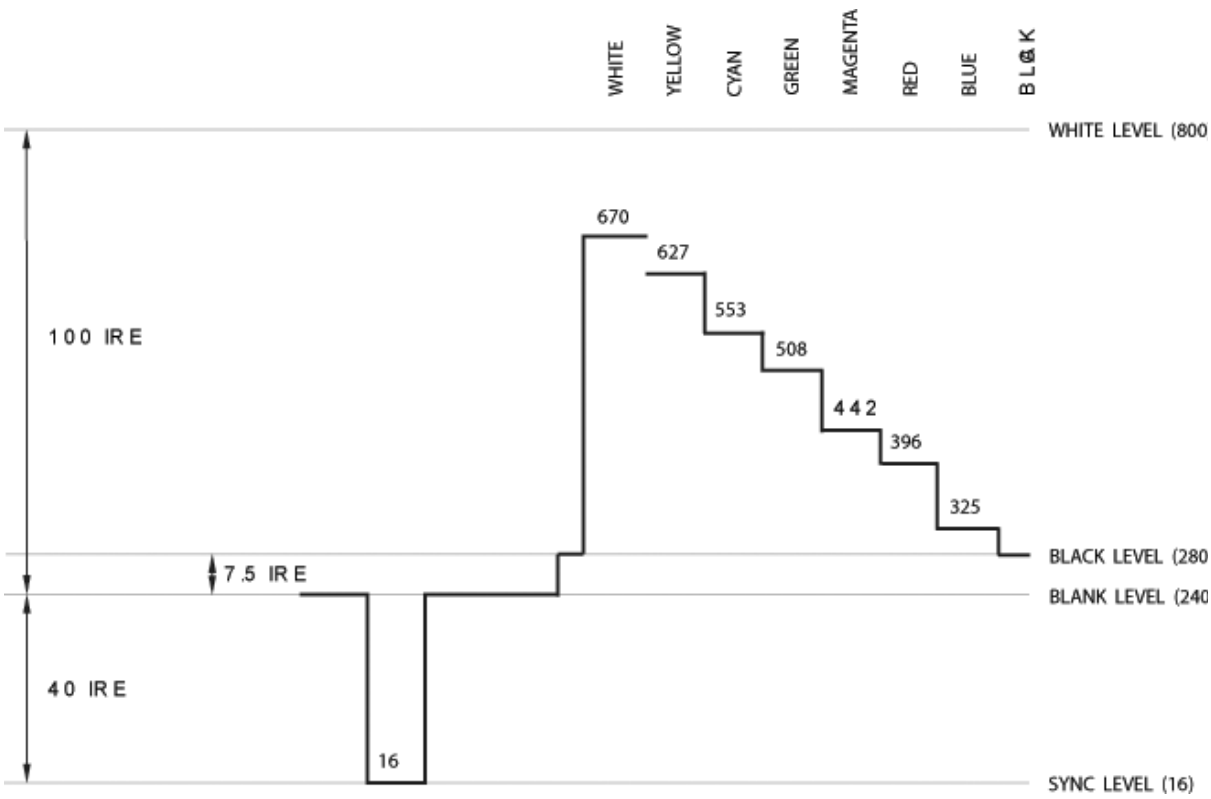


Figure 8-21 **NTSC y for s-video**

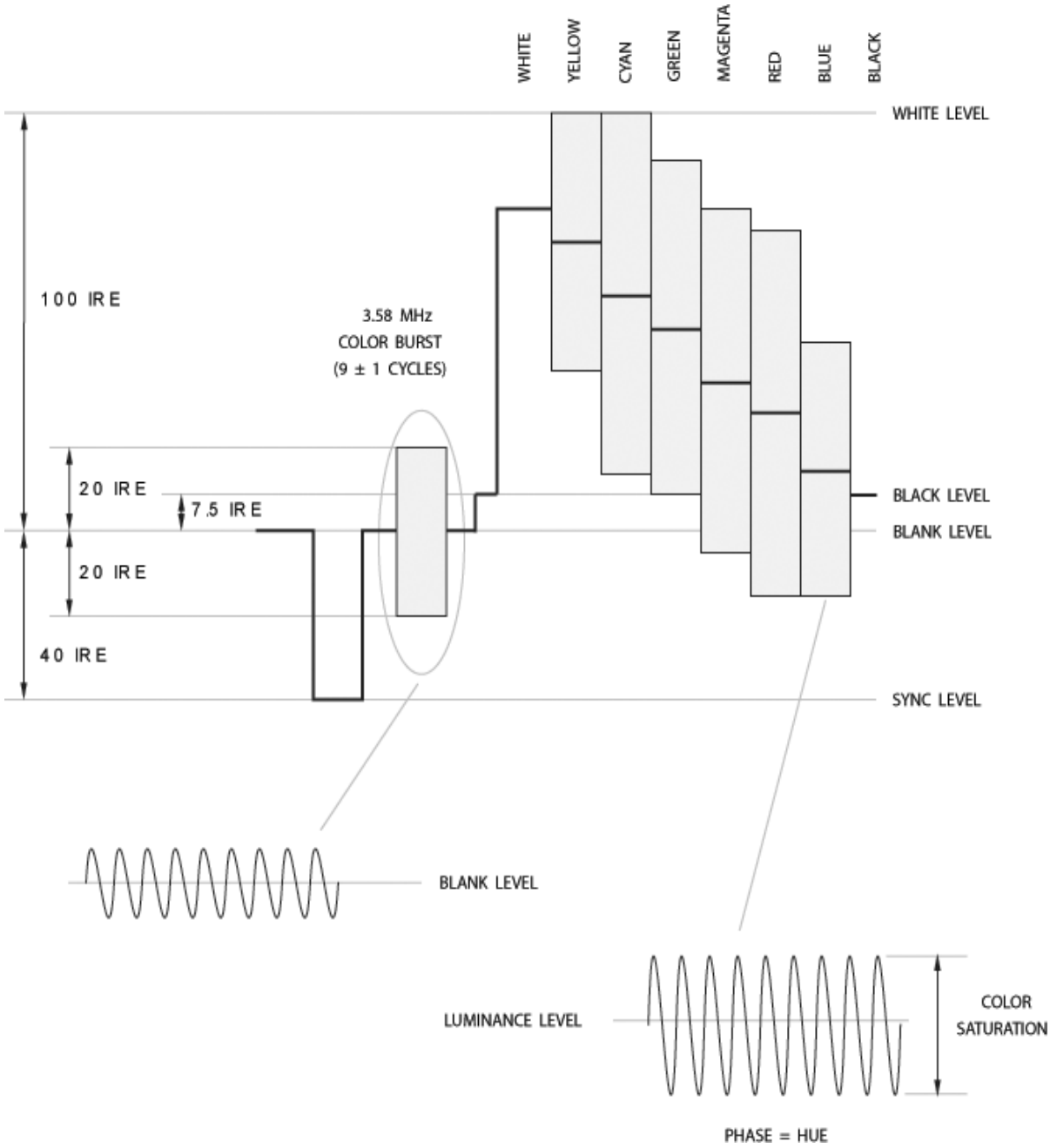


Figure 8-22 **CVBS (NTSC)**

8.4.13.6 Digital Video Output

The digital video output has the following features:

- 8/16-bit (4:2:2 YCbCr) or 24-bit (RGB or 4:4:4 YCbCr) digital output with optional DDR output (2x12-bit)
- 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

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/hue adjustment, 4:4:4 to 4:2:2 conversion, timing generation, and output data formatting.

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 maximum pixel clock frequency for the digital video output is 148.5MHz, enabling output resolutions up to 1920x1080p60 to be supported. The 1080p output is not available when the digital video output port is operated in a double-clocked output mode (data transferred on both edges of the clock).

VBI data is not present on this output port due to the variety of proprietary formats used.

This port may also be used to interface to, and drive progressively, LCD and PDP panels within a digital television.

Digital Video Output Timing

The block contains a complete programmable sync timing generator for operating as a sync master, or the timing generator may be slaved to H/V sync timing from,

- Video input port 1 or 2
- Graphics input port 1 or 2
- Component analog output
- Composite/s-video analog output
- External device connected to digital video output port

H/V sync signals output onto this port may be obtained from,

- Sync timing generator
- Component analog output

- Composite/s-video analog output

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. 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.

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.

Digital Video Output Processing

A 3x3 matrix multiplier with programmable coefficients is used to implement,

- YCbCr/RGB color space conversion
- Contrast (or ‘picture’ or ‘white level’) control
- Saturation (or ‘color’) control
- Hue (or ‘tint’) control
- BT.709 (HDTV) to BT.601 (SDTV) and BT.601 (SDTV) to BT.709 (HDTV) colorimetry conversion

The inputs of the matrix multiplier are 24-bit YCbCr from the display routing block; the outputs are 24-bit YCbCr or RGB. The matrix coefficients have a range of -4 to +3.2047/2048, in steps of 1/2048. The matrix multiplier is followed by three programmable DC offsets to provide optional brightness (or ‘black level’) control. The DC offsets have a range of -512 to + 511.7/8, in steps of 1/8. No sharpness control is provided for the digital video outputs.

Video Gamma Correction LUT (24-bpp RGB output mode only)

RGB video gamma correction is achieved through three 256x8 Look Up Tables (1 for each R/G/B component). The 3 LUTs are accessed simultaneously by the GBUS master, but are addressed separately by the data flow. They may be used to adjust the gamma of the video when driving LCD panels (since the LCDs have a different gamma curve than what the video uses) or black level adjustment.

Color Temperature Control LUT (24-bpp RGB output mode only)

After video gamma correction, a luminance component Y’ is derived for each (R, G, B) pixel. For SD outputs, the following equation is used to calculate Y':

Y' = 0.299R + 0.587G + 0.114B

For HD outputs, the following equation is used to calculate Y':

Y' = 0.2126R + 0.7152G + 0.0722B

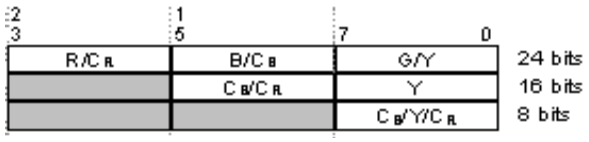
Y’ then addresses a 256x24 LUT containing delta (d) adjustments per RGB component. The delta (d) adjustments are 2’s complement signed and have 8 bits each. Their range is therefore [-128,+127].

The RGB output after color temperature control is defined as follows:

- R_{out} -> R_{in} + dR
- G_{out} -> G_{in} + dG
- B_{out} -> B_{in} + dB

Digital Video Output Formats

In the single data rate, depending on the selection made by the software either one of the 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.

Table 8-12 **Digital video output formats**

	Single Edge Transfer: Component Order 0				Single Edge Transfer: Component Order 1				Double Edge Transfer			
Video Output Pin	**24-bit RGB**	**24-bit YCbCr**	**16-bit YCbCr**	**8-bit YCbCr**	**24-bit RGB**	**24-bit YCbCr**	**16-bit YCbCr**	**8-bit YCbCr**	**12-bit RGB**			**12-bit YPrPb**
VO0_P23	R[7]	Cr[7]			R[7]	Cr[7]						
VO0_P22	R[6]	Cr[6]			R[6]	Cr[6]						
VO0_P21	R[5]	Cr[5]			R[5]	Cr[5]						
VO0_P20	R[4]	Cr[4]			R[4]	Cr[4]						
VO0_P19	R[3]	Cr[3]			R[3]	Cr[3]						
VO0_P18	R[2]	Cr[2]			R[2]	Cr[2]						
VO0_P17	R[1]	Cr[1]			R[1]	Cr[1]						
VO0_P16	R[0]	Cr[0]			R[0]	Cr[0]						
VO0_P15	G[7]	Y[7]	Y[7]		B[7]	Cb[7]	Cb/ Cr[7]					
VO0_P14	G[6]	Y[6]	Y[6]		B[6]	Cb[6]	Cb/ Cr[6]					
VO0_P13	G[5]	Y[5]	Y[5]		B[5]	Cb[5]	Cb/ Cr[5]					
VO0_P12	G[4]	Y[4]	Y[4]		B[4]	Cb[4]	Cb/ Cr[4]					
VO0_P11	G[3]	Y[3]	Y[3]		B[3]	Cb[3]	Cb/ Cr[3]		G[3]	R[7]	Y[3]	Cr[7]
VO0_P10	G[2]	Y[2]	Y[2]		B[2]	Cb[2]	Cb/ Cr[2]		G[2]	R[6]	Y[2]	Cr[6]
VO0_P9	G[1]	Y[1]	Y[1]		B[1]	Cb[1]	Cb/ Cr[1]		G[1]	R[5]	Y[1]	Cr[5]
VO0_P8	G[0]	Y[0]	Y[0]		B[0]	Cb[0]	Cb/ Cr[0]		G[0]	R[4]	Y[0]	Cr[4]
VO0_P7	B[7]	Cb[7]	Cb/ Cr[7]	Y/Cb/ Cr[7]	G[7]	Y[7]	Y[7]	Y/Cb/ Cr[7]	B[7]	R[3]	Cb[7]	Cr[3]
VO0_P6	B[6]	Cb[6]	Cb/ Cr[6]	Y/Cb/ Cr[6]	G[6]	Y[6]	Y[6]	Y/Cb/ Cr[6]	B[6]	R[2]	Cb[6]	Cr[2]
VO0_P5	B[5]	Cb[5]	Cb/ Cr[5]	Y/Cb/ Cr[5]	G[5]	Y[5]	Y[5]	Y/Cb/ Cr[5]	B[5]	R[1]	Cb[5]	Cr[1]
VO0_P4	B[4]	Cb[4]	Cb/ Cr[4]	Y/Cb/ Cr[4]	G[4]	Y[4]	Y[4]	Y/Cb/ Cr[4]	B[4]	R[0]	Cb[4]	Cr[0]

Table 8-12 **Digital video output formats (Continued)**

	Single Edge Transfer: Component Order 0				Single Edge Transfer: Component Order 1				Double Edge Transfer			
Video Output Pin	**24-bit RGB**	**24-bit YCbCr**	**16-bit YCbCr**	**8-bit YCbCr**	**24-bit RGB**	**24-bit YCbCr**	**16-bit YCbCr**	**8-bit YCbCr**	**12-bit RGB**			**12-bit YPrPb**
VO0_P3	B[3]	Cb[3]	Cb/ Cr[3]	Y/Cb/ Cr[3]	G[3]	Y[3]	Y[3]	Y/Cb/ Cr[3]	B[3]	G[7]	Cb[3]	Y[7]
VO0_P2	B[2]	Cb[2]	Cb/ Cr[2]	Y/Cb/ Cr[2]	G[2]	Y[2]	Y[2]	Y/Cb/ Cr[2]	B[2]	G[6]	Cb[2]	Y[6]
VO0_P1	B[1]	Cb[1]	Cb/ Cr[1]	Y/Cb/ Cr[1]	G[1]	Y[1]	Y[1]	Y/Cb/ Cr[1]	B[1]	G[5]	Cb[1]	Y[5]
VO0_P0	B[0]	Cb[0]	Cb/ Cr[0]	Y/Cb/ Cr[0]	G[0]	Y[0]	Y[0]	Y/Cb/ Cr[0]	B[0]	G[4]	Cb[0]	Y[4]
Output Order									1st	2nd	1st	2nd

The RGB output data may have a range of 0 (black) to 255 (white) or 16 (black) to 235 (white) by adjusting the matrix coefficient values. Data may be optionally clipped to only generate values of 1-254 or 16-235.

The Y output data has a range of 16 (black) to 235 (white). Data may be optionally clipped to only generate values of 1-254 or 16-235. CbCr output data has a range of 16-240. Data may be optionally clipped to only generate values of 1-254 or 16-240.

When generating BT.656 16-bit YCbCr data, EAV/SAV codes can be embedded in just the Y channel or both the Y and CbCr channels.

8.4.13.7 HDMI v1.1 Output

The SMP8634 contains a HDMI Transmitter. A complete HDMI system implementation requires an external Silicon Image Sil9002 PHY transmitter chip. The HDMI Transmitter block in the SMP8634 is fully HDMI compliant, and is used to transport consumer electronics standard digital video and digital audio over a TMDS interface. It supports the HDMI 1.1 (High Definition Multimedia Interface) and HDCP 1.1 (High-Bandwidth Digital Content Protection) specifications. The block provides a simple method for sending protected digital audio and video, providing end users with a truly all-digital experience. At the same time, built in backward compatibility with DVI 1.0 allows HDMI host systems to connect to any DVI 1.0 display (DTV, plasma display, LCD TV, PC monitor).

Note that 1080p video cannot be output using on-chip HDMI. Support for 1080p video via HDMI requires using an external HDMI transmitter chip, interfaced to the SMP8634's digital video output interface operating in a single-clock-edge mode.

The SMP8634 has the following features:

- HDMI 1.1, HDCP 1.1 and DVI 1.0 compliant transmitter (when using the Sil9002 external PHY chip)
- Integrated HDCP encryption engine for transmitting protected audio and video content
- Programmable data enable generator
- Power management
- Brightness, contrast (or ‘picture’ or ‘white level’), hue (or ‘tint’) and saturation (or ‘color’) controls

The HDMI transmitter block takes in uncompressed digital video data from the video output block, compressed or uncompressed digital audio from audio DSP 0, formats and encrypts it, then drives the digital video output pins. Note that when HDMI is used, the digital video output port can not be used for any other application.

HDMI Output Video Processing

The registers are configured to describe which format of video is being supplied to the HDMI transmitter. This information is passed over the HDMI link in the CEA-861C Active Video Information (AVI) packets.

The optional YCbCr/RGB color space converter (CSC) within the HDMI transmitter block interfaces to the video output block with YCbCr-only outputs, and provides full DVI 1.0 backwards compatibility. The CSC performs the standard-definition conversions according to BT.601, or high-definition conversions (BT.709). An up-converter allows 4:2:2 YCbCr data input to be converted to 4:4:4 YCbCr for transmitting over the HDMI link.

The HDCP encryption engine encrypts the incoming audio and video data. The encryption process is entirely controlled by the security CPU (XPU) through a set sequence of register reads and writes. The HDCP keys and Key Selector Value (KSV) are used in the encryption process.

HDMI Output Audio Processing

The HDMI block accepts digital audio from internal I^2S and S/PDIF channels and forms the audio data into packets in accordance with the HDMI specification. The I^2S channels used to drive the HDMI block are the same as those providing the audio DSP 0 I^2S outputs. Therefore, LPCM data over HDMI will have the same timing and format as the audio DSP 0 I^2S output pins. The HDMI v1.1 interface supports up to 2.0-channel 192KHz or 7.1-channel 96KHz LPCM audio. The S/PDIF channel used to drive the HDMI block is the same as that providing the audio DSP 0 S/PDIF output. Therefore, S/PDIF-based audio data over HDMI will have the same timing and format as the audio DSP 0 S/PDIF output pin.

Table 8-13 **Supported audio output combinations**

I^2S Outputs	S/PDIF Output	HDMI Output	Supported
LPCM	Bitstream	Bitstream	Yes
LPCM	Bitstream	LPCM	Yes
LPCM	LPCM	Bitstream	Yes. Requires audio DSP 1 I^2S and S/PDIF outputs be used.
LPCM	LPCM	LPCM	Yes

Note that it is not possible to transfer some compressed DTS-HD and Dolby TrueHD bitstreams over HDMI v1.1. HDMI v1.3 and CEA-861D are required to support these.

Within the HDMI block, the audio data may be down-sampled by one-half or one-fourth with register control. This allows for the sharing of the I^2S outputs with a high sample-rate audio DAC, while down-sampling HDMI audio for an attached display which supports only lower rates. Conversions from 192 to 48KHz, 176.4 to 44.1KHz, 96 to 48KHz and 88.2 to 44.1KHz are supported. The appropriate registers are configured to describe the format of audio being input. This information is passed over the HDMI link in the CEA-861C Audio Info (AI) packets.

The MCLK frequencies support various audio sample rates as shown in the table below:

Table 8-14 **Supported MCLK frequencies**

Multiple of Fs	Audio Sample Rate, Fs						
	32KHz	44.1KHz	48KHz	88.2KHz	96KHz	176.4KHz	192KHz
128	4.096MHz	5.645MHz	6.144MHz	11.29MHz	12.288MHz	22.579MHz	24.576MHz
192	6.144MHz	8.467MHz	9.216MHz	16.934MHz	18.432MHz	33.869MHz	36.864MHz
256	8.192MHz	11.290MHz	12.288MHz	22.579MHz	24.576MHz	45.158MHz	49.152MHz
384	12.288MHz	16.934MHz	18.432MHz	33.869MHz	33.864MHz	67.738MHz	73.728MHz
512	16.384MHz	22.579MHz	24.576MHz	45.158MHz	49.152MHz		
768	18.432MHz	16.934MHz	12.288MHz	67.738MHz	73.728MHz		
1024	24.576MHz	45.158MHz	49.152MHz				
1152	36.864MHz	50.803MHz	55.296MHz				

HDMI Output DCC Control

A dedicated I^2C master interface is used to control the DDC signals.

8.4.13.8 HDMI v1.3 Support

For applications requiring HDMI v1.3 support, an external HDMI v1.3 transmitter chip, such as the Silicon Image Sil9034 or Sil9134, may be interfaced to the SMP8634. Using the Sil9134 enables sending DTS-HD and Dolby TrueHD compressed audio bitstreams over HMDI. HDMI v1.3 deep-color (10- or 12-bit video data) cannot be supported using the SMP8634.

8.4.13.9 Simultaneous SD and HD Outputs

The SMP8634 is capable of simultaneous SD and HD outputs. The HD content used for downscaling can come from the output of the main video scaler (no OSD) or from the output of the main mixer. The first case is for the recording of the HD video, and the later case is for simultaneous output of the HD content at both HD and SD resolutions.

8.4.13.10 Dual TV Outputs

The SMP8634 is capable of decoding two programs and driving two TVs simultaneously. Possible modes of operation include SD+SD, HD+SD and HD+HD.

The source of the second TV output is the Multi-format scaler 1. No OSD is available for the second TV output.

Audio for the main video is output onto the audio DSP 0 I^2S and S/PDIF outputs. Audio for the secondary video is output onto the audio DSP 1 I^2S and S/PDIF outputs.

8.4.14 Video Input Interfaces

Two video input interfaces, one 8-bit (Video Input interface) and one 32-bit (Video/Graphics Input interface), are designed to capture a variety of digital video formats. They enable a design to support 2 NTSC/PAL video decoder chips plus an HDMI receiver chip, or one NTSC/PAL video decoder chip plus a 3D graphics chip. Example configurations for these inputs include:

Example 1:

- Two 8-bit 4:2:2 YCbCr inputs (NTSC/PAL decoders)
- One 16-bit 4:2:2 YCbCr input (HDMI receiver)

Example 2:

- Two 8-bit 4:2:2 YCbCr inputs (NTSC/PAL decoders)
- One 24-bit RGB input (DVI receiver)

Example 3:

- One 8-bit 4:2:2 YCbCr input (NTSC/PAL decoder)
- One 32-bit RGBA input (3D graphics chip)

Only two input ports can be captured to the DRAM (via the V-Bus) simultaneously. This allows the video data to then be read by the main video scaler, Multi-format scaler 1 or Multi-format scaler 2 for processing and display.

8.4.14.1 Video Input Interface

The digital video input interface is designed to capture 8-bit 4:2:2 408i/576i/480p/576p YCbCr digital video data in the BT.601 or BT.656 format. It supports clock rates up to 54MHz and resolutions of up to 720x576p. Video data is stored in the 16bpp 4:2:2 YCbCr format.

This input port may be combined with the pins from the video/graphics input interface to accept 16-bit 4:2:2 YCbCr digital video data in either the BT.601 or BT.656 format at clock rates up to 75MHz. This enables capturing 16-bit 4:2:2 YCbCr video for 720p or 1080i HDTV applications. The video data is stored in the 16bpp 4:2:2 YCbCr format.

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).

Digital Video Input Formats

Table 8-15 **Digital video input formats**

	Use Video/graphics Input 2 = 0		Use Video/graphics Input 2 = 1
	Single Edge Transfer		Single Edge Transfer
Video Input Pin	16-bit YCbCr	8-bit YCbCr	8-bit YCbCr
VI0_31	Y[7]		Y/Cb/Cr[7]
VI0_30	Y[6]		Y/Cb/Cr[6]
VI0_29	Y[5]		Y/Cb/Cr[7]
VI0_28	Y[4]		Y/Cb/Cr[6]
VI0_27	Y[3]		Y/Cb/Cr[7]
VI0_26	Y[2]		Y/Cb/Cr[6]
VI0_25	Y[1]		Y/Cb/Cr[7]
VI0_24	Y[0]		Y/Cb/Cr[6]
VI1_7	Cb/Cr[7]	Y/Cb/Cr[7]	
VI1_6	Cb/Cr[6]	Y/Cb/Cr[6]	
VI1_5	Cb/Cr[5]	Y/Cb/Cr[5]	
VI1_4	Cb/Cr[4]	Y/Cb/Cr[4]	
VI1_3	Cb/Cr[3]	Y/Cb/Cr[3]	
VI1_2	Cb/Cr[2]	Y/Cb/Cr[2]	

Table 8-15 **Digital video input formats (Continued)**

	Use Video/graphics Input 2 = 0		Use Video/graphics Input 2 = 1
	Single Edge Transfer		Single Edge Transfer
Video Input Pin	16-bit YCbCr	8-bit YCbCr	8-bit YCbCr
VI1_1	Cb/Cr[1]	Y/Cb/Cr[1]	
VI1_0	Cb/Cr[0]	Y/Cb/Cr[0]	

Note: [VI0_P31 to VI0_P24], these pins are available to the video input channel only if, the graphic input is not in single edge 32-bit mode and the video/graphic input 2 = 1 is used.

Digital video Input Timing

The video input block contains a complete programmable sync timing generator for operating as a sync master, or the timing generator may be slaved to H/V sync timing from,

- External device connected to video input port
- Graphics input
- Component analog output
- Composite/s-video analog output
- Digital video output

H/V sync signals output onto this port may be obtained from,

- Sync timing generator

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.

In 16-bit BT.656 mode, EAV/SAV codes present in the CbCr in stream are used for timing information. EAV/SAV codes in the Y stream are ignored.

The range of programmability for each video timing parameter for the digital video input port is as follows:

Table 8-16 **Range of programmability for digital video input timing**

Parameter	Range
Vt_total_size (vertical total per fame)	0 - 4095 lines
Hz_total_size (horizontal total per line)	0 - 4095 pixels
X_size (horizontal active per line)	0 - 4095 pixels
Y_size (vertical active per frame)	0 - 4095 lines
X_offset (Hsync to active video)	0 - 4095 pixels
Y_offset_top (Vsync to active video, top field)	0 - 4095 lines
Y_offset_bottom (Vsync to active video, bottom field)	0 - 4095 lines
Top_field_height (total size of top field)	0 - 8191 (half lines)
Hz_sync_width (Hsync width)	0 - 4095 pixels
Vt_sync_width (Vsync width)	0 - 8191 half lines
Vt_sync_fine_adjust, lower byte (Vsync to Hsync relative position adjust, top field)	0 to +/-127 pixels
Vt_sync_fine_adjust, upper byte (Vsync to Hsync relative position adjust, bottom field)	0 to +/-127 pixels

Table 8-17 **Range of programmability for VBI capture**

Parameter	Range
Top_VBI_line_enable (VBI capture for top field or frame)[1]	-
Top_VBI_offset (reference line for starting top field or frame VBI capture)[2]	0 - 31
Bot_VBI_line_enable (VBI capture for bottom field)[3]	-
Bot_VBI_offset (reference line for starting bottom field VBI capture[2])	0 - 31
H_size (width of VBI data, used only for VBI data during horizontal blanking)	0 - 4095 bytes
V_size (height of VBI data, used only for VBI data during horizontal blanking)	0 - 4095 lines
V_offset_top (top field line number where VBI window starts, used only for VBI data during horizontal blanking)	0 - 4095
V_offset_bottom (bottom field line number where VBI window starts, used only for VBI data during horizontal blanking)	0 - 4095

1. 24-bit value. Each bit enables (1) or disables (0) VBI capture for 24 lines, starting with the line specified by top_VBI_offset.
2. BT.656 progressive: line 0 = first line of vertical blanking
   BT.601 progressive: line 0 = first line of Vsync active edge
   BT.656 interlaced: line 0 = first line after field ID change
   BT.601 interlaced: line 0 = first line of Vsync active edge for top field and the line after for bottom field
3. 24-bit value. Each bit enables (1) or disables (0) VBI capture for 24 lines, starting with the line specified by bot_VBI_offset.

Digital Video Input Configuration

Configuration register bits can be used to select a number of features and characteristics:

- H- and V-sync source
- H- and V-sync polarity
- VBI capture enable/disable
- VBI capture timing
- Start of active video interrupt enable/disable
- 8-bit or 16-bit input format
- BT.601 or BT.656 input timing

Table 8-18 **Common video input formats**

Color Space	Video Format	Input Pixel Clock (MHz)							
		480i	480p	XGA	720p	1080i	SXGA	1080p	UXGA
RGB	4:4:4	27	27	65	74.25	74.25	108	148.5	165
		27	27	65	74.25	74.25			
YCbCr	4:4:4	27	27		74.25	74.25		148.5	165
		27	27	65	74.25	74.25			
	4:2:2	27	27		74.25	74.25		148.5	165
		27	27		74.25	74.25		148.5	165
		27	27	65	74.25	74.25			
		27	54		74.25	74.25			
		27	54						
			54						

8.4.14.2 Video/Graphics Input Interface

The SMP8634 also contains a video/graphics input port. This block can input 16/24/32-bpp data using either BT.656 or BT.601 protocols. The maximum input frequency is 148.5MHz. The block includes:

- A 128 pixels buffer (useful when the graphic is directly sent to the display)
- A bpp format converter (with key-color support)
- A color space converter (RGB-> YCbCr (HD,SD) and YCbCr HD/SD)

When the output format of the block is set to 16-bit and the color space converter output is YCbCr, the content is converted from YCbCr 4:4:4 (24-bit) to YCbCr 4:2:2 (16-bit). This layer can be displayed by the main video scaler or the Multi-format scalers 1 or 2. In 24/32-bpp modes, the input port supports DDR transfer (using either 12 or 16 pins).

Video/Graphics Input Formats

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 αRGB (4444, 1555, or 565)
- 24-bit αRGB (0888, 8565, or 5676)
- 24-bit RGB data (12-bit 2x multiplexed, 888 over two words, ‘C’ and ‘I’ version), Intel compatible
- 32-bit αRGB (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. Though both modes can be captured, only the 16-235 mode is converted to YCbCr.

In dual edge transfer, the 1st chunk can be either on posedge on negedge (and vice versa for the 2nd chunk).

Table 8-19 **Video/graphics input formats**

Use Graphic Input 2 = 0											Use Graphic Input 2 = 1
	Single Edge transfer: component order 0				Single Edge transfer: component order 1				Dual edge transfer		Single Edge Transfer
Video/ graphic Input Pin	24/32-bit RGB	24/32-bit YCbCr	16-bit YCbCr	8-bit YCbCr	24/32-bit RGB	24/32-bit YCbCr	16-bit YCbCr	8-bit YCbCr	12-bit RGB	12-bit YCbCr	8 bit YCbCr
VI0_P31	A[7]	A[7]			A[7]	A[7]					Y/Cb/Cr[7]
VI0_P30	A[6]	A[6]			A[6]	A[6]					Y/Cb/Cr[6]
VI0_P29	A[5]	A[5]			A[5]	A[5]					Y/Cb/Cr[5]
VI0_P28	A[4]	A[4]			A[4]	A[4]					Y/Cb/Cr[4]
VI0_P27	A[3]	A[3]			A[3]	A[3]					Y/Cb/Cr[3]
VI0_P26	A[2]	A[2]			A[2]	A[2]					Y/Cb/Cr[2]
VI0_P25	A[1]	A[1]			A[1]	A[1]					Y/Cb/Cr[1]
VI0_P24	A[0]	A[0]			A[0]	A[0]					Y/Cb/Cr[0]
VI0_P23	R[7]	Cr[7]			R[7]	Cr[7]					
VI0_P22	R[6]	Cr[6]			R[6]	Cr[6]					
VI0_P21	R[5]	Cr[5]			R[5]	Cr[5]					
VI0_P20	R[4]	Cr[4]			R[4]	Cr[4]					
VI0_P19	R[3]	Cr[3]			R[3]	Cr[3]					
VI0_P18	R[2]	Cr[2]			R[2]	Cr[2]					
VI0_P17	R[1]	Cr[1]			R[1]	Cr[1]					
VI0_P16	R[0]	Cr[0]			R[0]	Cr[0]					
VI0_P15	G[7]	Y[7]	Y[7]		B[7]	Cb[7]	Cb/ Cr[7]				
VI0_P14	G[6]	Y[6]	Y[6]		B[6]	Cb[6]	Cb/ Cr[6]				
VI0_P13	G[5]	Y[5]	Y[5]		B[5]	Cb[5]	Cb/ Cr[5]				
VI0_P12	G[4]	Y[4]	Y[4]		B[4]	Cb[4]	Cb/ Cr[4]				

Table 8-19 **Video/graphics input formats (Continued)**

Use Graphic Input 2 = 0													**Use Graphic Input 2 = 1**
	Single Edge transfer: component order 0				**Single Edge transfer: component order 1**				**Dual edge transfer**				**Single Edge Transfer**
Video/ graphic Input Pin	**24/32-bit RGB**	**24/32-bit YCbCr**	**16-bit YCbCr**	**8-bit YCbCr**	**24/32-bit RGB**	**24/32-bit YCbCr**	**16-bit YCbCr**	**8-bit YCbCr**	**12-bit RGB**		**12-bit YCbCr**		**8 bit YCbCr**
VI0_P11	G[3]	Y[3]	Y[3]		B[3]	Cb[3]	Cb/ Cr[3]		G[3]	R[7]	Y[3]	Cr[7]	
VI0_P10	G[2]	Y[2]	Y[2]		B[2]	Cb[2]	Cb/ Cr[2]		G[2]	R[6]	Y[2]	Cr[6]	
VI0_P9	G[1]	Y[1]	Y[1]		B[1]	Cb[1]	Cb/ Cr[1]		G[1]	R[5]	Y[1]	Cr[5]	
VI0_P8	G[0]	Y[0]	Y[0]		B[0]	Cb[0]	Cb/ Cr[0]		G[0]	R[4]	Y[0]	Cr[4]	
VI0_P7	B[7]	Cb[7]	Cb/ Cr[7]	Y/ Cb/ Cr[7]	G[7]	Y[7]	Y[7]	Y/Cb/ Cr[7]	B[7]	R[3]	Cb[7]	Cr[3]	
VI0_P6	B[6]	Cb[6]	Cb/ Cr[6]	Y/ Cb/ Cr[6]	G[6]	Y[6]	Y[6]	Y/Cb/ Cr[6]	B[6]	R[2]	Cb[6]	Cr[2]	
VI0_P5	B[5]	Cb[5]	Cb/ Cr[5]	Y/ Cb/ Cr[5]	G[5]	Y[5]	Y[5]	Y/Cb/ Cr[5]	B[5]	R[1]	Cb[5]	Cr[1]	
VI0_P4	B[4]	Cb[4]	Cb/ Cr[4]	Y/ Cb/ Cr[4]	G[4]	Y[4]	Y[4]	Y/Cb/ Cr[4]	B[4]	R[0]	Cb[4]	Cr[0]	
VI0_P3	B[3]	Cb[3]	Cb/ Cr[3]	Y/ Cb/ Cr[3]	G[3]	Y[3]	Y[3]	Y/Cb/ Cr[3]	B[3]	G[7]	Cb[3]	Y[7]	
VI0_P2	B[2]	Cb[2]	Cb/ Cr[2]	Y/ Cb/ Cr[2]	G[2]	Y[2]	Y[2]	Y/Cb/ Cr[2]	B[2]	G[6]	Cb[2]	Y[6]	
VI0_P1	B[1]	Cb[1]	Cb/ Cr[1]	Y/ Cb/ Cr[1]	G[1]	Y[1]	Y[1]	Y/Cb/ Cr[1]	B[1]	G[5]	Cb[1]	Y[5]	
VI0_P0	B[0]	Cb[0]	Cb/ Cr[0]	Y/ Cb/ Cr[0]	G[0]	Y[0]	Y[0]	Y/Cb/ Cr[0]	B[0]	G[4]	Cb[0]	Y[4]	
Input Order									1st on pose dge	2nd on neg edge	1st on pose dge	2nd on neg edge	

Note: The pins VI0_P31 to VI0_P24 are available to the graphic input channel only if, the video input is in the 8-bit mode and video/graphics = 0 is used.

Single Edge Data Pins Assignment

The “Input Format” bits in the VO_GRAPH_IN_FORMAT register describe the different pixel formats that are supported. The following table illustrates the data bits/pins assignment of the graphic input port in each of those modes.

Notice that the indices of the component bits indicate the position they would have, in an 8-bit description of the component.

Also, for simplification R, G and B are used to identify the color components. The color space could be different. In particular, the display can handle Cr, Y, Cb.

Table 8-20 **Single edge data pins assignment**

Bit	24bpp	24bpp-8565	24bpp-5676	32bpp	16bpp-565	16bpp-1555	16bpp-4444	31bpp-7888
31				A7				A7
30				A6				A6
29				A5				A5
28				A4				A4
27				A3				A3
26				A2				A2
25				A1				A1
24				A0				
23	R7	A7	A7	R7				R7
22	R6	A6	A6	R6				R6
21	R5	A5	A5	R5				R5
20	R4	A4	A4	R4				R4
19	R3	A3	A3	R3				R3
18	R2	A2	R7	R2				R2
17	R1	A1	R6	R1				R1
16	R0	A0	R5	R0				R0
15	G7	R7	R4	G7	R7	A	A7	G7
14	G6	R6	R3	G6	R6	R7	A6	G6

Table 8-20 **Single edge data pins assignment (Continued)**

Bit	24bpp	24bpp-8565	24bpp-5676	32bpp	16bpp-565	16bpp-1555	16bpp-4444	31bpp-7888
13	G5	R5	R2	G5	R5	R6	A5	G5
12	G4	R4	G7	G4	R4	R5	A4	G4
11	G3	R3	G6	G3	R3	R4	R7	G3
10	G2	G7	G5	G2	G7	R3	R6	G2
9	G1	G6	G4	G1	G6	G7	R5	G1
8	G0	G5	G3	G0	G5	G6	R4	G0
7	B7	G4	G2	B7	G4	G5	G7	B7
6	B6	G3	G1	B6	G3	G4	G6	B6
5	B5	G2	B7	B5	G2	G3	G5	B5
4	B4	B7	B6	B4	B7	B7	G4	B4
3	B3	B6	B5	B3	B6	B6	B7	B3
2	B2	B5	B4	B2	B5	B5	B6	B2
1	B1	B4	B3	B1	B4	B4	B5	B1
0	B0	B3	B2	B0	B3	B3	B4	B0

8.5 Register Maps

8.5.1 Video Processing Registers

8.5.1.1 Multi-format Scaler Registers

Table 8-2. **Register map - Multi-format scaler**

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_SCALE_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
+072C	+062C	+052C	VO_XXX_BCS2	R/W	Video Output GFX/CRT/VCR BCS 2 Register
+C000	+B000	+A000	VO_XXX_LUT0	R/W	Video Output GFX/CRT/VCR LUT 0 Register
+C004	+B004	+A004	VO_XXX_LUT1	R/W	Video Output GFX/CRT/VCR LUT 1 Register
+C3FC	+B3FC	+A3FC	VO_XXX_LUT255	R/W	Video Output GFX/CRT/VCR LUT 255 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

8.5.2.1 Main Video Scaling Registers

Table 8-1 **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.

8.5.2.2 OSD Scaler Registers

Table 8-2 **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 8-2 **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.

8.5.2.3 Hardware Cursor Registers

Table 8-3 **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.

8.5.2.4 Main Mixer Registers

Table 8-4 **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	Reserved		
+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 8-4 **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.

8.5.2.5 Graphic Accelerator (Bitmap Mode) Registers

Table 8-5 **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.

8.5.2.6 Graphic Accelerator (Vectorial Mode) Registers

Table 8-6 **Graphic accelerator (vectorial mode) registers**

Address[1]	Register Name	R/W	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.

8.5.2.7 Display Routing Register

Table 8-7 **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.

8.5.2.8 Color Bars Generator Registers

Table 8-8 **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.

8.5.2.9 HD to SD Scaler Registers

Table 8-9 **Color bars generator registers**

Address[1]	Register Name	R/W	Description
+1300	VO_HDSD_CONFIG	R/W	Video Output HD to SD Configuration Register
+1304	VO_HDSD_XSCALE	R/W	Video Output HD to SD X-scale Register
+1308	VO_HDSD_YSCALE	R/W	Video Output HD to SD Y-scale Register
+130C	VO_HDSD_DUMP	R/W	Video Output HD to SD Dump Register

1. Address refers to G-Bus byte address relative to the video output block base.

8.5.3 Video Input Registers

Table 8-10 **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
+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.

8.5.4 Graphic Input Registers

Table 8-11 **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
+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.

8.5.5 Video Output Registers

8.5.5.1 Digital Output Registers

Table 8-12 **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.

8.5.5.2 Analog Output Registers

Table 8-13 **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
+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

Table 8-13 **Analog output registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+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.

8.5.5.3 Component Analog Output Registers

Table 8-14 **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
+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

Table 8-14 **Component analog output registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+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.

8.5.5.4 Composite Analog Output Registers

Table 8-15 **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.

8.6 Pin Description

8.6.1 Video Input Pins

8.6.1.1 Digital Video Input 0 Pins - Graphic Input

Table 8-16 **Digital video input 0 (graphic input) pin descriptions**

Pin Name	Ball ID	Direction	Description
VI0_CLK	K1	I	Video graphic input port 0 clock
VI0_HS	J2	B	Video graphic input port 0 Hsync
VI0_P0	D2	I	Video graphic input port 0 bit 0
VI0_P1	D1	I	Video graphic input port 0 bit 1
VI0_P10	G5	I	Video graphic input port 0 bit 10
VI0_P11	G4	I	Video graphic input port 0 bit 11
VI0_P12	G3	I	Video graphic input port 0 bit 12
VI0_P13	G2	I	Video graphic input port 0 bit 13
VI0_P14	G1	I	Video graphic input port 0 bit 14
VI0_P15	H5	I	Video graphic input port 0 bit 15
VI0_P16	H4	I	Video graphic input port 0 bit 16
VI0_P17	H3	I	Video graphic input port 0 bit 17
VI0_P18	H2	I	Video graphic input port 0 bit 18
VI0_P19	H1	I	Video graphic input port 0 bit 19
VI0_P2	E3	I	Video graphic input port 0 bit 2
VI0_P20	J5	I	Video graphic input port 0 bit 20
VI0_P21	J4	I	Video graphic input port 0 bit 21
VI0_P22	J3	I	Video graphic input port 0 bit 22
VI0_P23	K5	I	Video graphic input port 0 bit 23
VI0_P24	K4	I	Video graphic input port 0 bit 24
VI0_P25	K3	I	Video graphic input port 0 bit 25
VI0_P26	L5	I	Video graphic input port 0 bit 26
VI0_P27	L4	I	Video graphic input port 0 bit 27
VI0_P28	L3	I	Video graphic input port 0 bit 28
VI0_P29	M5	I	Video graphic input port 0 bit 29
VI0_P3	E2	I	Video graphic input port 0 bit 3

Table 8-16 **Digital video input 0 (graphic input) pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
VI0_P30	M4	I	Video graphic input port 0 bit 30
VI0_P31	M3	I	Video graphic input port 0 bit 31
VI0_P4	E1	I	Video graphic input port 0 bit 4
VI0_P5	F5	I	Video graphic input port 0 bit 5
VI0_P6	F4	I	Video graphic input port 0 bit 6
VI0_P7	F3	I	Video graphic input port 0 bit 7
VI0_P8	F2	I	Video graphic input port 0 bit 8
VI0_P9	F1	I	Video graphic input port 0 bit 9
VI0_VLD	K2	I	Video graphic input port 0 data valid
VI0_VS	J1	B	Video graphic input port 0 Vsync

8.6.1.2 Digital Video Input 1 Pins

Table 8-17 **Digital video input 1 pin descriptions**

Pin Name	Ball ID	Direction	Description
VI1_CLK	A3	I	Video input port 1 clock signal. Active edge programmable.
VI1_HS	B3	B	Video input port 1 Hsync input or output
VI1_P0	A2	I	Video input port 1 pixel bus bit 0
VI1_P1	B2	I	Video input port 1 pixel bus bit 1
VI1_P2	C3	I	Video input port 1 pixel bus bit 2
VI1_P3	C4	I	Video input port 1 pixel bus bit 3
VI1_P4	B1	I	Video input port 1 pixel bus bit 4
VI1_P5	C1	I	Video input port 1 pixel bus bit 5
VI1_P6	C2	I	Video input port 1 pixel bus bit 6
VI1_P7	D3	I	Video input port 1 pixel bus bit 7
VI1_VLD	B4	I	Video input port 1 data valid
VI1_VS	A4	B	Video input port 1 Vsync input or output

8.6.1.3 Digital Video Input 2 Pins

Table 8-18 **Digital video input 2 pin descriptions**

Pin Name	Ball ID	Direction	Description
VI2_CLK	M1	I	Video input port 2 clock signal. Active edge programmable.
VI2_HS	L2	B	Video input port 2 Hsync input or output
VI2_VLD	M2	I	Video input port 2 data valid
VI2_VS	L1	B	Video input port 2 Vsync input or output

8.6.1 Video Output Pins

8.6.1.1 Digital Video Output Pins

Table 8-19 **Digital video output pin descriptions**

Pin Name	Ball ID	Direction	Description
VO0_CLK	U1	O	Video output pixel clock signal
VO0_HS	V1	B	Video output Hsync input or output. Polarity programmable.
VO0_P0	W3	O	Video output pixel bus bit 0 (YPbPr/RGB)
VO0_P1	V4	O	Video output pixel bus bit 1 (YPbPr/RGB)
VO0_P10	R4	O	Video output pixel bus bit 10 (YPbPr/RGB)
VO0_P11	R3	O	Video output pixel bus bit 11 (YPbPr/RGB)
VO0_P12	R2	O	Video output pixel bus bit 12 (YPbPr/RGB)
VO0_P13	R1	O	Video output pixel bus bit 13 (YPbPr/RGB)
VO0_P14	P5	O	Video output pixel bus bit 14 (YPbPr/RGB)
VO0_P15	P4	O	Video output pixel bus bit 15 (YPbPr/RGB)
VO0_P16	P3	O	Video output pixel bus bit 16 (YPbPr/RGB)
VO0_P17	P2	O	Video output pixel bus bit 17 (YPbPr/RGB)
VO0_P18	P1	O	Video output pixel bus bit 18 (YPbPr/RGB)
VO0_P19	N5	O	Video output pixel bus bit 19 (YPbPr/RGB)
VO0_P2	V3	O	Video output pixel bus bit 2 (YPbPr/RGB)
VO0_P20	N4	O	Video output pixel bus bit 20 (YPbPr/RGB)
VO0_P21	N3	O	Video output pixel bus bit 21 (YPbPr/RGB)
VO0_P22	N2	O	Video output pixel bus bit 22 (YPbPr/RGB)

Table 8-19 **Digital video output pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
VO0_P23	N1	O	Video output pixel bus bit 23 (YPbPr/RGB)
VO0_P3	U4	O	Video output pixel bus bit 3 (YPbPr/RGB)
VO0_P4	U3	O	Video output pixel bus bit 4 (YPbPr/RGB)
VO0_P5	T4	O	Video output pixel bus bit 5 (YPbPr/RGB)
VO0_P6	T3	O	Video output pixel bus bit 6 (YPbPr/RGB)
VO0_P7	T2	O	Video output pixel bus bit 7 (YPbPr/RGB)
VO0_P8	T1	O	Video output pixel bus bit 8 (YPbPr/RGB)
VO0_P9	R5	O	Video output pixel bus bit 9 (YPbPr/RGB)
VO0_VLD	U2	O	Video output port data valid signal. Active high.
VO0_VS	V2	B	Video output port Vsync input or output. Polarity programmable.

8.6.1.2 Analog Video Output Pins - Composite and S-video

Table 8-20 **Composite and S-video pin descriptions**

Pin Name	Ball ID	Direction	Description
VO1_AVDD	AG4	I	Video output analog block 3.3V power supply connection
VO1_AVDD_U	AJ2	I	Pb-channel DAC power supply. Connect to a 3.3V nominal supply.
VO1_AVDD_V	AH2	I	Pr-channel DAC power supply. Connect to a 3.3V nominal supply.
VO1_AVDD_Y	AG2	I	Y-channel DAC power supply. Connect to a 3.3V nominal supply.
VO1_AVSS	AG5	I	Video output analog block ground connection
VO1_AVSS_U	AJ3	I	Pb-channel DAC ground connection
VO1_AVSS_V	AH3	I	Pr-channel DAC ground connection
VO1_AVSS_Y	AG3	I	Y-channel DAC ground connection
VO1_RSET	AH4	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_U	AJ1	O	Analog video output. Outputs Pb signal in the component YPbPr 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.

Table 8-20 **Composite and S-video pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
VO1_V	AH1	O	Analog video output. Outputs Pr signal in component YPbPr 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_VREF	AH5	B	Video DAC current source reference voltage (1.25V nominal)
VO1_Y	AG1	O	Analog video output. Output Y (luminance) signals in the component YPbPr or S-video mode, or G signal in the component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.

8.6.1.3 Analog Video Output Pins - Component Analog Output

Table 8-21 **Component analog output pin descriptions**

Pin Name	Ball ID	Direction	Description
VO2_AVDD	AK4	I	Video output analog block 3.3V power supply connection.
VO2_AVDD_U	AM2	I	Pb-channel DAC power supply. Connect to a 3.3V nominal supply.
VO2_AVDD_V	AL2	I	Pr-channel DAC power supply. Connect to a 3.3V nominal supply.
VO2_AVDD_Y	AK2	I	Y-channel DAC power supply. Connect to a 3.3V nominal supply.
VO2_AVSS	AK5	I	Video output analog block ground connection
VO2_AVSS_U	AM3	I	Pb-channel DAC ground connection
VO2_AVSS_V	AL3	I	Pr-channel DAC ground connection
VO2_AVSS_Y	AK3	I	Y-channel DAC ground connection
VO2_RSET	AJ4	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_U	AM1	O	Analog video output. Outputs Pb signal in component YPbPr mode, or B signal in component the RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO2_V	AL1	O	Analog video output. Outputs Pr signal in component YPbPr mode, or R signal in the component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.
VO2_VREF	AJ5	B	Video DAC current source reference voltage (1.25V nominal)

Table 8-21 **Component analog output pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
VO2_Y	AK1	O	Analog video output. Outputs Y (luminance) signal in component YPbPr mode, or G signal in the component RGB mode. Current output, intended to drive 75 ohm doubly-terminated load.

8.6.1.4 HDMI Pins

Table 8-22 **HDMI pin descriptions**

Pin Name	Ball ID	Direction	Description
HDMI_DSCL	U5	B	HDMI DDC clock
HDMI_DSDA	V5	B	HDMI DDC data (open drain output)
HDMI_HPD	W4	I	HDMI hot plug detect input
HDMI_MSEN	W5	I	HDMI monitor sense
HDMI_PD#	T5	O	HDMI power down (active low)
VO0_P0	W3	O	Bit 0 (LSB) of double-data rate bus to external HDMI PHY device
VO0_P1	V4	O	Bit 1 of double-data rate bus to external HDMI PHY device
VO0_P2	V3	O	Bit 2 of double-data rate bus to external HDMI PHY device
VO0_P3	U4	O	Bit 3 of double-data rate bus to external HDMI PHY device
VO0_P4	U3	O	Bit 4 of double-data rate bus to external HDMI PHY device
VO0_P5	T4	O	Bit 5 of double-data rate bus to external HDMI PHY device
VO0_P6	T3	O	Bit 6 of double-data rate bus to external HDMI PHY device
VO0_P7	T2	O	Bit 0 of double-data rate bus to external HDMI PHY device
VO0_P8	T1	O	Bit 8 of double-data rate bus to external HDMI PHY device
VO0_P9	R5	O	Bit 9 of double-data rate bus to external HDMI PHY device
VO0_P10	R4	O	Bit 10 of double-data rate bus to external HDMI PHY device
VO0_P11	R3	O	Bit 11 (MSB) of double-data rate bus to external HDMI PHY device
VO0_P12	R2	O	‘CTL2’ signal to external HDMI PHY device

Table 8-22 **HDMI pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
VO0_P13	R1	O	'CTL3' signal to external HDMI PHY device

8.7 Electrical Characteristics

8.7.2 Digital Video Input DC Characteristics

Table 8-23 **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	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
R_{PU}[3]	Pulldown resistor value	Kohms	38	54	83
R_{PD}[4]	Pullup resistor value	Kohms	63	114	259

1. Parameter applies to VI0_HS, VI0_VS, VI1_HS, VI1_VS, VI2_HS, VI2_VS when configured as outputs.
2. Parameter applies to VI0_CLK, VI0_VLD, VI0_P(31:0), VI1_CLK, VI1_VLD, VI2_CLK, VI2_VLD. Also applies to VI0_HS, VI0_VS, VI1_HS, VI1_VS, VI2_HS, VI2_VS when configured as inputs.
3. Parameter applies to VI0_CLK, VI0_VLD, VI0_P(31:0), VI1_CLK, VI1_VLD, VI1_P(7:0), VI2_CLK, VI2_VLD.
4. Parameter applies to VI0_HS, VI0_VS, VI1_HS, VI1_VS, VI2_HS, VI2_VS.

8.7.3 Digital Video Output DC Characteristics

Table 8-24 **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	27	46
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
I_{oH}[2]	High level output current (@V_{OH} = 2.4V)	mA	17	36	61
I_{oL}[2]	Low level output current (@V_{OL} = 0.4V)	mA	11	19	29
V_{IH}[3]	Input high voltage	V	2		5.5
V_{IL}[2]	Input low voltage	V	-0.3		0.8

1. Parameter applies to HDMI_PD#; also HDMI_DSCL and HDMI_DSDA when in output mode.
2. Parameter applies to VO0_CLK, VO0_VLD, VO0_P(23:0; also VO0_HS and VO0_VS when configured as outputs.

3. Parameter applies to HDMI_HPD, HDMI_MSEN; also HDMI_DSCL and HDMI_DSDA when in input mode. Also applies to VO0_HS and VO0_VS when configured as inputs.

8.7.4 Digital Video Input AC Characteristics

Table 8-25 **Digital video input AC electrical characteristics**

Parameter	Conditions	Minimum	Typical	Maximum	Units
T_{VICLK}		6.7			ns
F_{VICLK}				148.5	MHz
T_{VICLKL}		0.45		0.55	T_{VICLK}
T_{VICLKH}		0.45		0.55	T_{VICLK}
T_{SU}		1.0			NS
T_{H}		0			ns

8.7.5 Digital Video Output AC Characteristics

Table 8-26 **Digital video output AC electrical characteristics**

Parameter	Conditions	Minimum	Typical	Maximum	Units
T_{PXCLK}	SDR mode	6.7			ns
F_{PXCLK}	SDR mode			148.5	MHz
T_{PXCLK}	DDR mode	13.4			ns
F_{PXCLK}	DDR mode			74.25	MHz
T_{PXCLKL}		0.3		0.7	T_{PXCLK}
T_{PXCLKH}		0.3		0.7	T_{PXCLK}
T_{DLY}		0		3.0	ns
T_{OD}	SDR mode	0		3.5	ns
T_{OD}	DDR mode	0		2.5	ns
$T_{DLYSTEP}$	DDR mode only	600	700	800	ps

8.7.6 Analog Video Output DC Characteristics

Table 8-27 **Analog video output electrical characteristics**

Symbol	Description	Minimum	Typical	Maximum	Units
	DAC resolution		12		bits
RSET	Current set resistor (1% tolerance)		140		ohms
IFS	Full scale output current	35		37	mA
VFS	Full scale output voltage (R_{LOAD} = 37.5)		1.35		V
NL_{DIFF}	Differential nonlinearity			2	LSB
NL_{INT}	Integral nonlinearity			3	LSB
	Offset error			1	LSB
	Gain error			5	%FS
ZOUT	Output impedance		1800		ohms
FPIXEL	Update rate			148.5	MHz

8.8 Timing Diagrams

8.8.1 Digital Video Timing

8.8.1.1 Digital Video Input timing

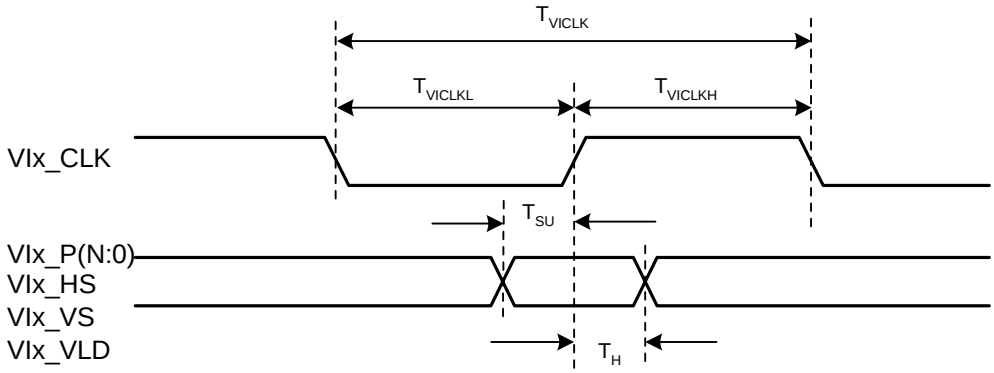


Figure 8-1 **Digital video input timing diagram**

Ts = 1 ns (minimum), Th = 0 ns (minimum).

8.8.1 Digital Video Output Timing

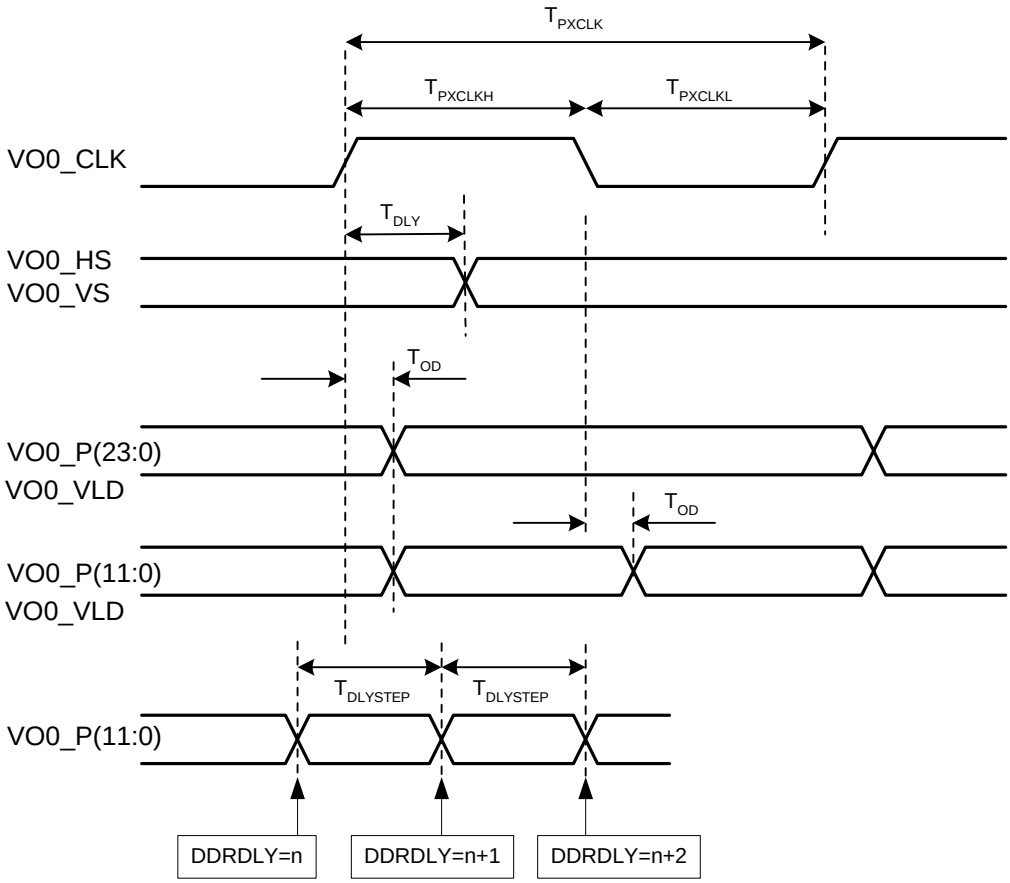


Figure 8-2 **Digital video output timing diagram**

8.8.1.2 HSYNC/VSYNC Timing

Interlaced Mode, Top/Bottom Signalization

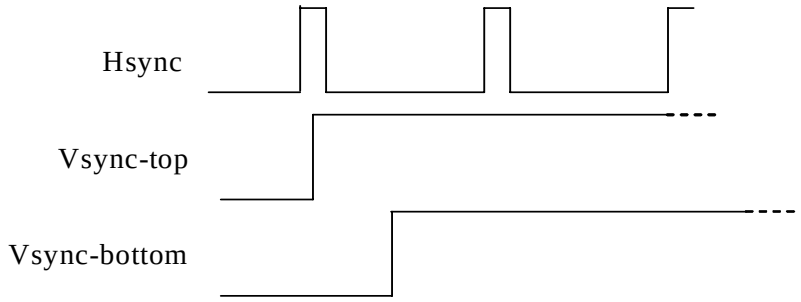


Figure 8-3 **HSYNC/VSYNC timing diagram (interlaced mode, top/bottom signalization)**

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.

8.8.1.3 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.

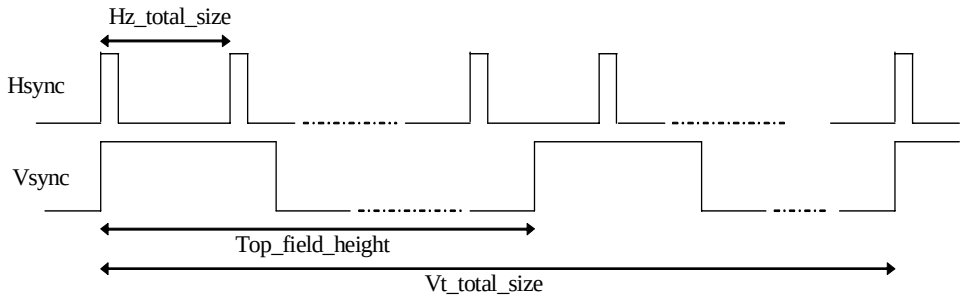


Figure 8-4 **Active high Hsync and Vsync in the interlaced mode**

8.8.1.4 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 below:

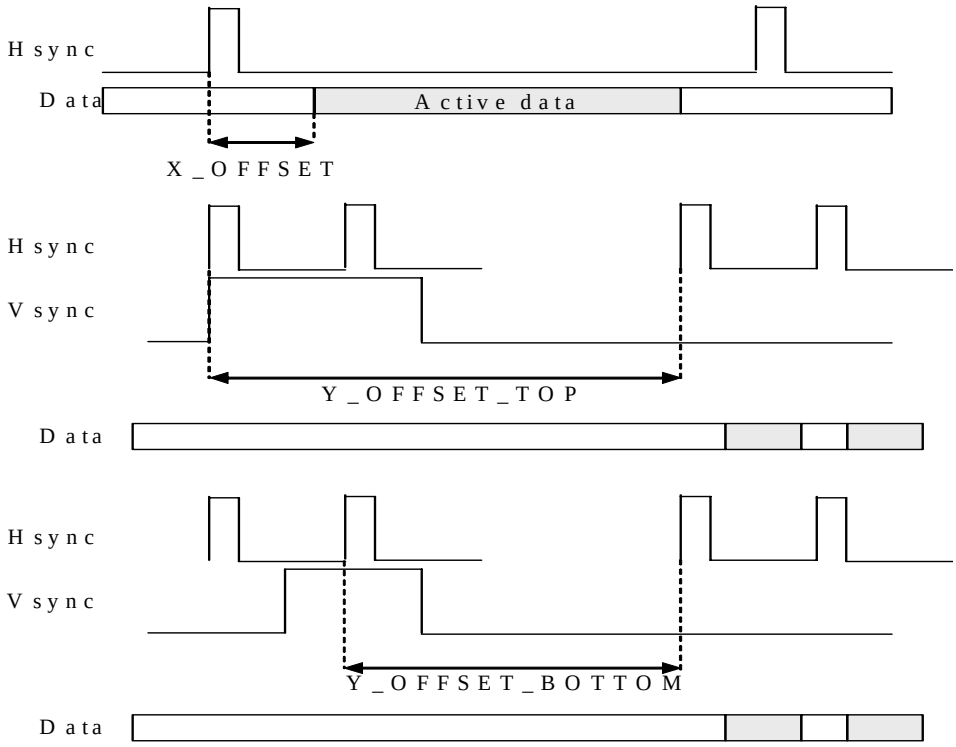


Figure 8-5 **Active picture positioning timing diagram**

8.8.1.5 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).

9 Audio Processing Subsystem

9.1 Block Diagram of Audio Processing Subsystem

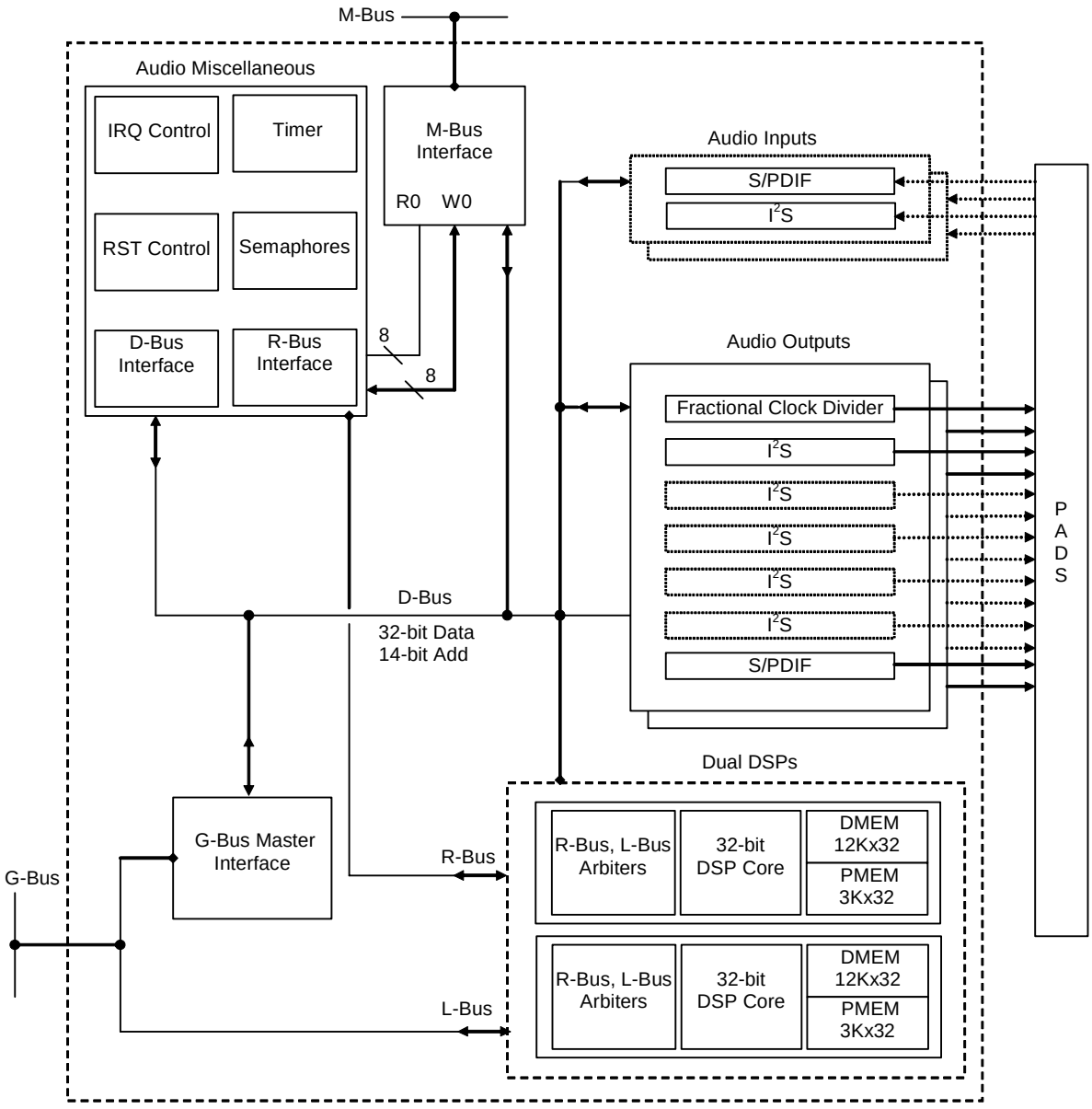


Figure 9-1 **Audio processing subsystem block diagram**

9.2 Introduction

The SMP8634 contains an integrated audio subsystem based on two identical custom-designed 32-bit digital signal processors (DSP), each operating at up to 300MHz (rev. C or later, 200MHz for previous revisions). The audio decoding and processing algorithms are implemented on the proprietary DSPs. This firmware-based approach gives great flexibility for accommodating future audio standards or specialized audio requirements.

Each DSP is capable of decoding 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.

- Dolby Digital 5.1, Dolby Digital Plus 7.1, Dolby TrueHD 7.1
- DTS 5.1, DTS-HD 7.1
- MPEG-1 Layers I, II and III (MP3) 2.0
- MPEG-4 AAC-LC 5.1
- MPEG-4 HE-AAC 5.1
- MPEG-4 BSAC 2.0
- WMA9 2.0, WMA9 Pro 5.1
- DVD-Audio with MLP option
- Apple iTunes 2.0 (no DRM)
- ATRAC3 2.0
- 24-bit linear PCM 7.1

Each DSP is capable of encoding the following audio formats. In order to receive code for a specific audio codec, the user must be a codec licensee.

- Dolby Digital 5.1
- DTS 5.1

The audio subsystem can also support the audio requirements for MSTV IPTV, BD, HD-DVD, FVD, DVD, SVCD, VCD, ARIB, ATSC, DMB, DVB and OpenCable applications.

9.3 Features

- 2 audio DSPs to support BD and HD-DVD audio requirements
- Proprietary DSP-based audio processors
- Dual 9.1-ch I^2S digital audio outputs (one for each audio DSP, typically configured as 7.1-ch + 2.0-ch downmix)
- Dual S/PDIF digital audio outputs (one for each audio DSP)
  - IEC 60958
  - IEC 61937-1
  - IEC 61937-2
  - IEC 61937-3 (Dolby Digital)
  - IEC 61937-4 (MPEG)
  - IEC 61937-5 (DTS)
  - IEC 61937-6 (MPEG-2 AAC format)
- Dual 2-ch I^2S or S/PDIF digital audio inputs (one for each audio DSP)
- Audio Over HDMI

9.4 Functional Description

The audio block of the SMP8634 contains two identical proprietary 32-bit DSPs, 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 DSPs to become G-Bus masters 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 DSPs.
- Audio output: Sends an ten/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 DSPs.

The audio block connects to other SMP8634 on-chip components via the G-Bus and the M-Bus. Several local buses connect the modules contained within the audio block.

9.4.1 Audio Processing Capabilities

The audio unit provides two 9.1-ch I^2S outputs (one for each audio DSP), two S/PDIF outputs (one for each audio DSP), and two I^2S or S/PDIF audio inputs (one for each audio DSP).

9.4.1.1 Audio Codec Support

Table 9-1 **MPEG audio decode support**

Codec	MPEG-1 Layer I, II	MPEG-1 Layer III	MPEG-4 AAC-LC	MPEG-4 HE-AAC (aacPlus)	MPEG-4 BSAC
Maximum bit rate	I: 448kbps II:384kbps	320kbps	384kbps	384kbps	64kbps
Maximum channels	2.0	2.0	5.1 2.0	5.1	2.0
Bits/sample	16 24	16 24	16 24	16 24	16 24
Sampling frequency (kHz)	32 44.1 48	16 22.05 24 32 44.1 48	8 11.025 12 16 22.05 24 32 44.1 48 64 (2.0) 88.2 (2.0) 96 (2.0)	8 11.025 12 16 22.05 24 32 44.1 48	8 11.025 12 16 22.05 24 32 44.1 48
Downmix support	-	-	2.0	2.0	-
Percentage of one audio DSP (300MHz) used	12%	15%	27% (5.1) 7% (2.0)	36%	TBD
MIPS used on host CPU	-	-	-	-	TBD
BD-ROM main audio	-	-	-	-	-
BD-ROM secondary audio	-	-	-	-	-
HD-DVD main audio	Mandatory - 2.0 (48kHz) Optional - 5.1 (48kHz) - 7.1 (48kHz)	-	-	-	-

Table 9-1 **MPEG audio decode support (Continued)**

Codec	MPEG-1 Layer I, II	MPEG-1 Layer III	MPEG-4 AAC-LC	MPEG-4 HE-AAC (aacPlus)	MPEG-4 BSAC
HD-DVD secondary audio	-	Optional - 2.0 (24kHz) - 2.0 (48kHz)	-	Optional - HE-AAC v2 - 2.0 (48kHz)	-

Table 9-2 **Dolby audio decode support**

Codec	Dolby Digital	Dolby Digital Plus	Dolby TrueHD	MLP Lossless (for DVD-A)
Maximum bit rate	640kbps	4.736Mbps	18Mbps	10Mbps
Maximum channels	5.1	7.1	7.1 5.1	5.1 2.0
Bits/sample	16 20 24	16 20 24	16 20 24	16 20 24
Sampling frequency (kHz)	32 44.1 48	32 44.1 48	48 96 192 (5.1)	48 96 192 (2.0)
Downmix support	2.0	5.1 2.0	5.1 2.0	2.0
Percentage of one audio DSP (300MHz) used	33%	53% (7.1) 33% (5.1) 23% (2.0)	65%	47%
MIPS used on host CPU	-	-	-	-
BD-ROM main audio	Mandatory - 5.1 (48kHz)	Optional - 7.1 (48kHz)	Optional - 5.1 (192kHz) - 7.1 (96kHz)	-
BD-ROM secondary audio	-	Mandatory - 2.0 (48kHz) Optional - 5.1 (48kHz)	-	-
HD-DVD main audio	Mandatory - 5.1 (48kHz)	Mandatory - 5.1 (48kHz) Optional - 7.1 (48kHz)	Mandatory - 2.0 (192kHz) Optional - 5.1(96kHz) - 7.1 (96kHz)	-
HD-DVD secondary audio	-	Mandatory - 2.0 (48kHz)	-	-

Note: For Blu-ray, Dolby Lossless incorporates a Dolby Digital stream (2.0- to 5.1-ch) and a Dolby TrueHD stream (2.0- to 7.1-ch). Dolby TrueHD is a version of MLP Lossless for Blu-ray Disc and HD-DVD applications.

Note: For Blu-ray, the Dolby Digital Plus 7.1 stream contains a Dolby Digital 5.1 stream plus data to extend it to 7.1 channels.

Note: For Blu-ray, the Dolby TrueHD stream contains a Dolby Digital stream (up to 5.1-ch) and one or two TrueHD extension streams to add lossless and up to 2 additional channels.

Table 9-3 **DTS audio decode support**

Codec	DTS	DTS-HD-HRA DTS-HD-MA
Maximum bit rate	1536kbps	6.0Mbps 24.5Mbps
Maximum channels	5.1	5.1 (192kHz) 7.1 (96kHz)
Bits/sample	16 20 24	16 20 24
Sampling frequency (kHz)	8 11.05 12 16 22.05 24 32 44.1 48 64 88.2 96 128 176.4 192	8 11.05 12 16 22.05 24 32 44.1 48 64 88.2 96 128 176.4 192
Downmix support	2.0	5.1 2.0
Percentage of one audio DSP (300MHz) used	44%	TBD
MIPS used on host CPU	-	-

Table 9-3 **DTS audio decode support (Continued)**

Codec	DTS	DTS-HD-HRA DTS-HD-MA
BD-ROM main video	Mandatory - Core data Optional - Extended data	Mandatory - Core data of core substream Optional - Extended data of core substream - Extension substream
BD-ROM secondary audio	-	-
HD-DVD main video		Mandatory - 5.1 (96kHz) Optional - 7.1 (96kHz) - 2.0 lossless (192kHz)
HD-DVD secondary audio	-	Mandatory - 2.0 (48kHz)

Table 9-4 **LPCM, WMA9 and ATRAC3 audio decode support**

Codec	LPCM	WMA9	WMA9 Pro	ATRAC3
Maximum bit rate	36.864Mbps	192kbps (CBR) 384kbps (VBR)	768kbps (CBR) 768kbps (VBR)	768kbps
Maximum channels	7.1	2.0	5.1	2.0
Bits/sample	16 20 24	16 20 24	16 20 24	16 20 24
Sampling frequency (kHz)	32 44.1 48 88.2 96 192	8 11.025 12 16 22.05 24 32 44.1 48 96	32 44.1 48	44.1 48
Downmix support	5.1 2.0	-	2.0	-
Percentage of one audio DSP (300MHz) used		20%	53%	TBD
MIPS used on host CPU	-	-	-	TBD

Table 9-4 **LPCM, WMA9 and ATRAC3 audio decode support (Continued)**

Codec	LPCM	WMA9	WMA9 Pro	ATRAC3
BD-ROM main audio	Mandatory - 2.0 (96kHz) Optional - 5.1 (192kHz) - 7.1 (96kHz)	-	-	-
BD-ROM secondary audio	-	-	-	-
HD-DVD main audio	Mandatory - 5.1 (96kHz) - 2.0 (192kHz)	-		-
HD-DVD secondary audio	-	-	Optional - 2.0 (24kHz) - 2.0 (48kHz)	-

Table 9-5 **Dolby and DTS audio encode support**

Codec	Dolby Digital	DTS
Maximum bit rate	640kbps	1536kbps
Maximum channels	5.1	5.1
Bits/sample	16 20 24	16 20 24
Sampling frequency (kHz)	48	48
Percentage of one audio DSP (300MHz) used	50%	TBD
MIPS used on host CPU	-	-

Table 9-6 **Blu-ray audio processing requirements**

Codec				
Process	M1 mixer	M2 mixer	Parse interac-tive sound	Upsampling Downsampling Postprocessing
Percentage of one audio DSP (300MHz) used	5%	5%	13%	30%
MIPS used on host CPU	-	-	-	-

9.4.1.2 BD-ROM Audio Support

Table 9-7 **BD-ROM audio support matrix - rev. C and later**

Feature	Audio DSP Used	Feature
Main Audio Decode	DSP 0	Dolby Digital 5.1 Dolby Digital Plus 7.1 DTS 5.1 Dolby TrueHD 5.1/7.1 (192kHz/ 96kHz) LPCM 5.1/7.1 (192kHz/96kHz)
Secondary Audio Decode	DSP 1	Dolby Digital Plus 2.0
Interactive Audio Parsing and Mixing	DSP 1	8-channel
Main Audio Mixing	DSP 0	Yes
Audio Encode for SPDIF Output	DSP 1	Dolby Digital 5.1
Upsampling, Downsampling and Postprocessing	DSP 0	Yes

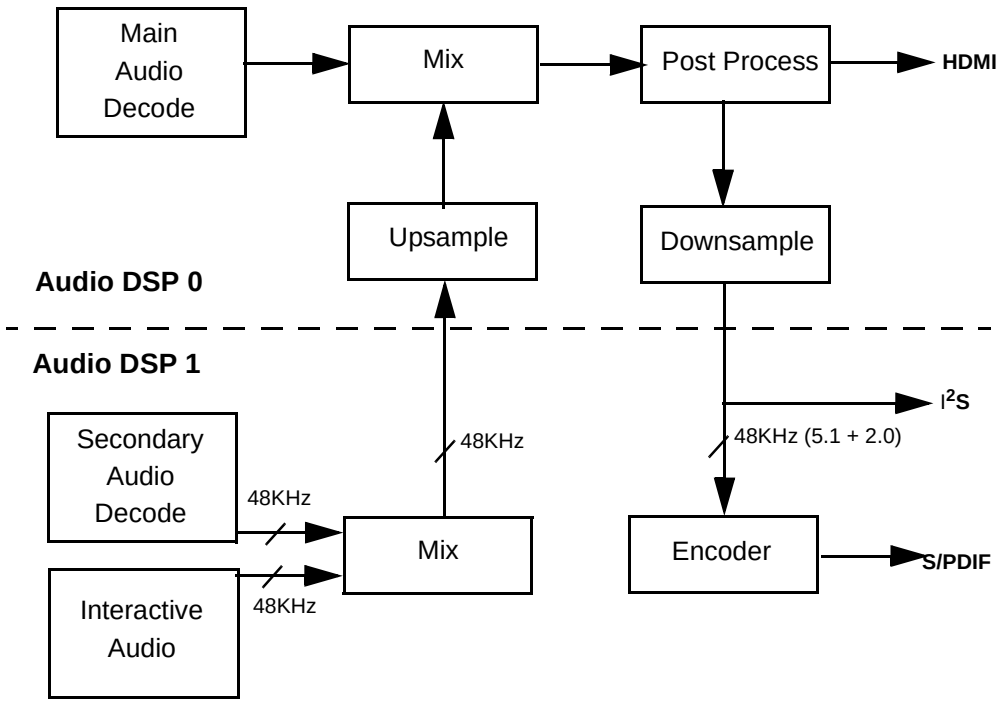


Figure 9-2 **Blu-ray audio processing - rev. C and later**

Table 9-8 **BD-ROM audio support matrix - rev. A and B**

Feature	Audio DSP Used	Feature
Main Audio Decode	DSP 0	Dolby Digital 5.1 Dolby Digital Plus 7.1 DTS 5.1 Dolby TrueHD 5.1/7.1 (192kHz/ 96kHz) LPCM 5.1/7.1 (192kHz/96kHz)
Secondary Audio Decode	-	No
Interactive Audio Parsing and Mixing	DSP 1	8-channel
Main Audio Mixing	DSP 1	Yes
Audio Encode for SPDIF Output	-	No
Upsampling, Downsampling and Postprocessing	DSP 1	Yes

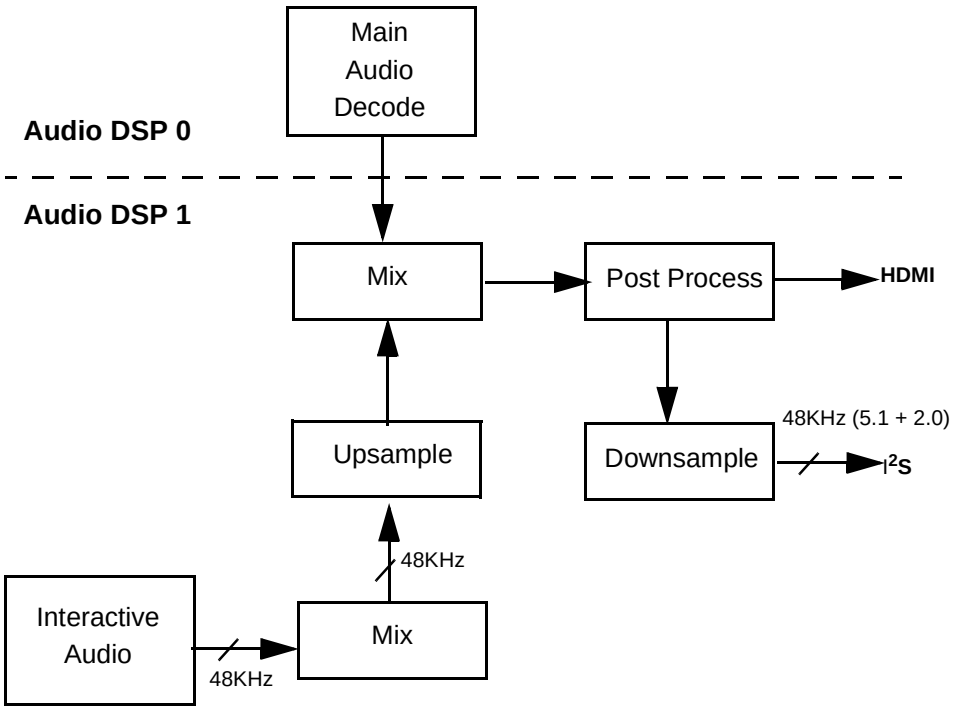


Figure 9-3 **Blu-ray audio processing - rev. A and B**

9.4.1.3 Additional Processing Capabilities

Additional capabilities include:

- Downloading of new firmware
- Mixing I^2S or S/PDIF input with audio output
- Mixing and panning of 8 streams of LPCM 1.0 or 6 streams of LPCM 1.0 + 1 stream of LPCM 2.0
- Upsampling (44.1 to 48 kHz, 2x and 4x)
- De-emphasis (48 and 44.1 kHz audio only)
- Soft fade-in and fade-out
- Audio delay adjust, channel delay adjust, channel level adjust, test tone generation.

9.4.2 Audio DSPs

The SMP8634 audio subsystem contains two identical proprietary DSPs. Each DSP block is built around a 32-bit DSP core. It has the following features:

- 300MHz (rev. C or later, 200MHz for previous revisions) cycle time (system clock). All instructions execute in 1 cycle, except for the branching instructions (bra, call, trap, ret etc. - 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.

9.4.3 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.

9.4.4 Audio Input Interfaces

The audio input interfaces allow each DSP to receive and process the audio 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. For a given audio DSP, 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 has two channels organized as one pair: (L0,R0). Once the input modes are set (once at the beginning of each session), the 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 five 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 input 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 mode is 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.

9.4.5 Audio Output Interfaces

The audio output interfaces provide the DSPs 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 for each audio DSP has 12 channels organized as 6 pairs: L0/R0, L1/R1, L2/R2, L3/R3, L4/R4 and LS/RS. The first 10 channels (5 pairs) drive the 5 I^2S output blocks while the 6^{th} pair drives the S/PDIF output block. The I^2S outputs operate only as a master; they cannot operate in a slave mode. The audio data on the audio DSP 0 I^2S and S/PDIF outputs may also be output via the HDMI output port. All the I^2S and S/PDIF output channels for a given DSP work with one LRCLK, limiting to one sample rate for all the channels.

Table 9-9 **Supported audio output combinations**

I^2S Outputs	S/PDIF Output	HDMI Output	Supported
LPCM	Bitstream	Bitstream	Yes
LPCM	Bitstream	LPCM	Yes
LPCM	LPCM	Bitstream	Yes. Requires audio DSP 1 I^2S and S/PDIF outputs be used.
LPCM	LPCM	LPCM	Yes

The configuration register bits for each DSP 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, 96KHz or 192KHz). The interrupt service routine must write 8 samples into 10 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 12 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.

9.4.5.1 I²S Output Format

The I²S interfaces operate 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).

To control external audio DACs that do not use I²C for the control interface, GPIO pins may be used.

Table 9-10 **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
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

Table 9-10 **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
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.

9.4.5.2 S/PDIF (IEC 60958) Output Format

The S/PDIF outputs 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.

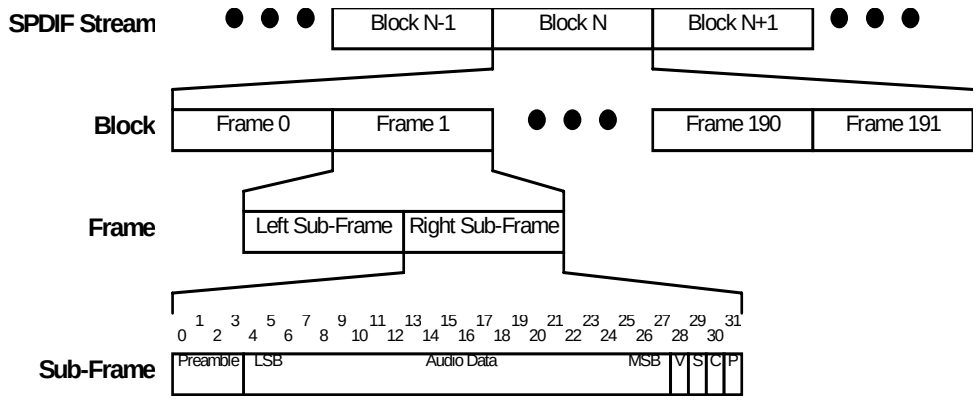


Figure 9-4 **S/PDIF data format**

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). Subcode data is used to convey user-specific data.

Channel Status Information (Subframe Bit 30)

This bit is used to carry side information on the S/PDIF stream, including SCMS information. 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 subframe contain an even number of 1's (and 0's).

9.4.5.3 HDMI v1.1 Output Format

The HDMI v1.1 output supports LPCM and compressed audio. The audio data from the audio DSP 0 I^2S outputs or the S/PDIF output may also be transferred over HDMI. Note that the audio capabilities are dependent on the video format being transmitted over HDMI. For additional information, please see CEA-861D and the HDMI video output section.

9.5 Register Maps

9.5.1 Audio Input Interface Registers

Table 9-11 **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.

9.5.2 Audio Output Interface Registers

Table 9-12 **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)

9.5.3 Audio Miscellaneous Registers

Table 9-13 **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

Table 9-13 **Audio miscellaneous registers (Continued)**

Address[1]	Register Name	R/W/A[2]	Description
+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.

9.6 Pin Description

9.6.1 Audio Input Interface Pins

Table 9-14 **Audio input interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SI0_BCLK	AN7	I	Audio input 0 serial bit clock = LRCLKx32 / 64. Maximum frequency is 12.288MHz.
SI0_DATA	AP7	I	Audio input 0 serial data
SI0_LRCLK	AL7	I	Audio input 0 serial left/right clock
SI0_SPDIF	AM7	I	Audio input 0 serial data S/PDIF
SI1_BCLK	AL8	I	Audio input 1 serial bit clock = LRCLKx32 / 64. Maximum frequency is 12.288MHz.
SI1_DATA	AM8	I	Audio input 1 serial data
SI1_LRCLK	AK7	I	Audio input 1 serial left/right clock
SI1_SPDIF	AK8	I	Audio input 1 serial data S/PDIF

9.6.2 Audio Output Interface Pins

Table 9-15 **Audio output interface pin descriptions**

Pin Name	Ball ID	Direction	Description
SO0_BCLK	AP3	O	Audio output 0 I^2S audio bit clock output. Maximum frequency is 12.288MHz.
SO0_DATA1	AP4	O	Audio output 0 I^2S channel 1 audio data output
SO0_DATA2	AN3	O	Audio output 0 I^2S channel 2 audio data output
SO0_DATA3	AN4	O	Audio output 0 I^2S channel 3 audio data output
SO0_DATA4	AM4	O	Audio output 0 I^2S channel 4 audio data output
SO0_DATA5	AL4	O	Audio output 0 I^2S channel 5 audio data output
SO0_LRCLK	AN2	O	Audio output 0 I^2S audio frame clock output
SO0_MACLK	AP2	O	Audio output 0 audio clock generator master clock output. Maximum frequency is 50MHz.
SO0_SPDIF	AN1	O	Audio output 0 S/PDIF audio data output
SO1_BCLK	AN5	O	Audio output 1 I^2S audio bit clock output. Maximum frequency is 12.288MHz.
SO1_DATA1	AP6	O	Audio output 1 I^2S channel 1 audio data output
SO1_DATA2	AM6	O	Audio output 1 I^2S channel 2 audio data output
SO1_DATA3	AN6	O	Audio output 1 I^2S channel 3 audio data output
SO1_DATA4	AL6	O	Audio output 1 I^2S channel 4 audio data output
SO1_DATA5	AL5	O	Audio output 1 I^2S channel 5 audio data output
SO1_LRCLK	AK6	O	Audio output 1 I^2S audio frame clock output
SO1_MACLK	AP5	O	Audio output 1 audio clock generator master clock output. Maximum frequency is 50MHz.
SO1_SPDIF	AM5	O	Audio output 1 S/PDIF audio data output

9.7 Electrical Characteristics

9.7.1 Audio Input Interface DC Characteristics

Table 9-16 **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	35	54	85

9.7.2 Audio Output Interface DC Characteristics

Table 9-17 **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	27	46
I_{OL}[1]	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
I_{OH}[2]	High level output current (@V_{OH} = 2.4V)	mA	25	54	91
I_{OL}[2]	Low level output current (@V_{OL} = 0.4V)	mA	15	26	38

1. Parameter applies to all SO0 pins except SO0_SPDIF.
2. Parameter applies to SO0_SPDIF.

9.7.3 I^2S Input AC Electrical Characteristics

Table 9-18 **I^2S Input AC characteristics**

Symbol	Units	Minimum	Typical	Maximum
F_{BCLK}	MHz		12.288	20
T_{BCLK}	ns	50		
T_{BCLKL}	T_{BCLK}	0.35		0.65
T_{BCLKH}	T_{BCLK}	0.35		0.65
T_{SR}	ns	10		
T_{HTR}	ns	0		

9.7.4 I²S Output AC Electrical Characteristics

Table 9-19 **I²S Output AC characteristics**

Symbol	Units	Minimum	Typical	Maximum
F_{BCLK}	MHz		12.288	20
T_{BCLK}	ns	50		
T_{BCLKL}	T_{BCLK}	0.48		0.52
T_{BCLKH}	T_{BCLK}	0.48		0.52
T_{VLD}	ns	$T_{BCLKH - 10}$		$T_{BCLKH + 10}$

9.8 Timing Diagram

9.8.1 I²S Audio Input Timing Diagram

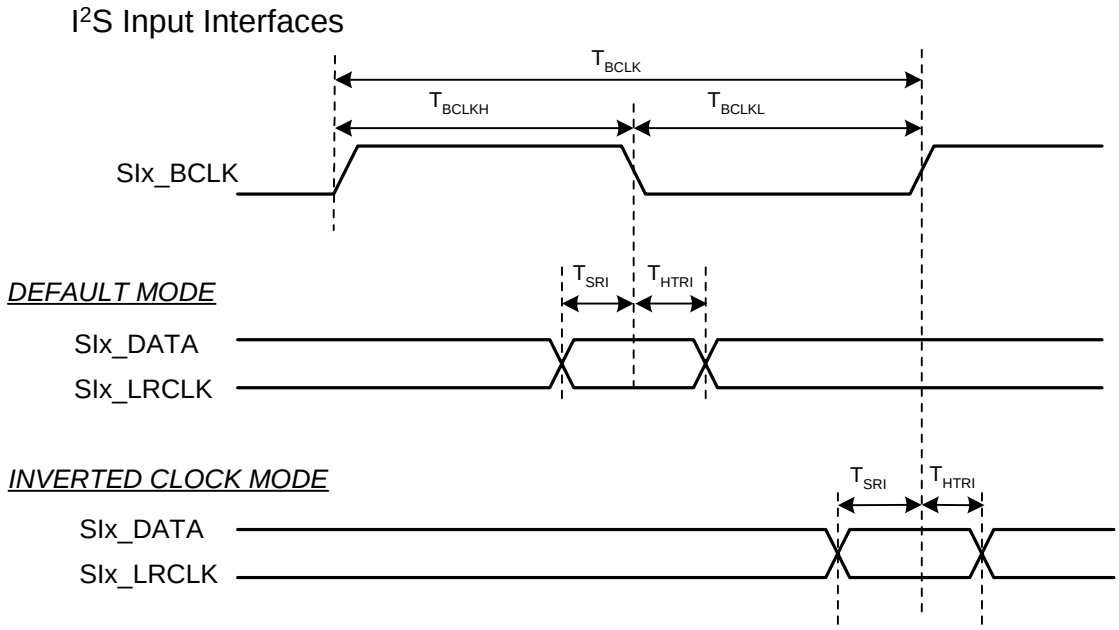


Figure 9-5 **I²S audio input timing diagram**

9.8.2 I²S Audio Output Timing Diagram

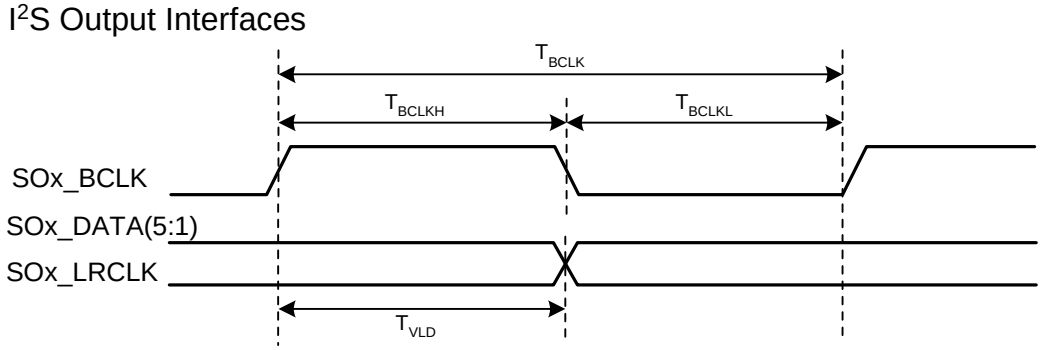


Figure 9-6 **I²S audio output timing diagram**

10 Transport Demultiplexer

10.1 Block Diagram of Transport Demultiplexer

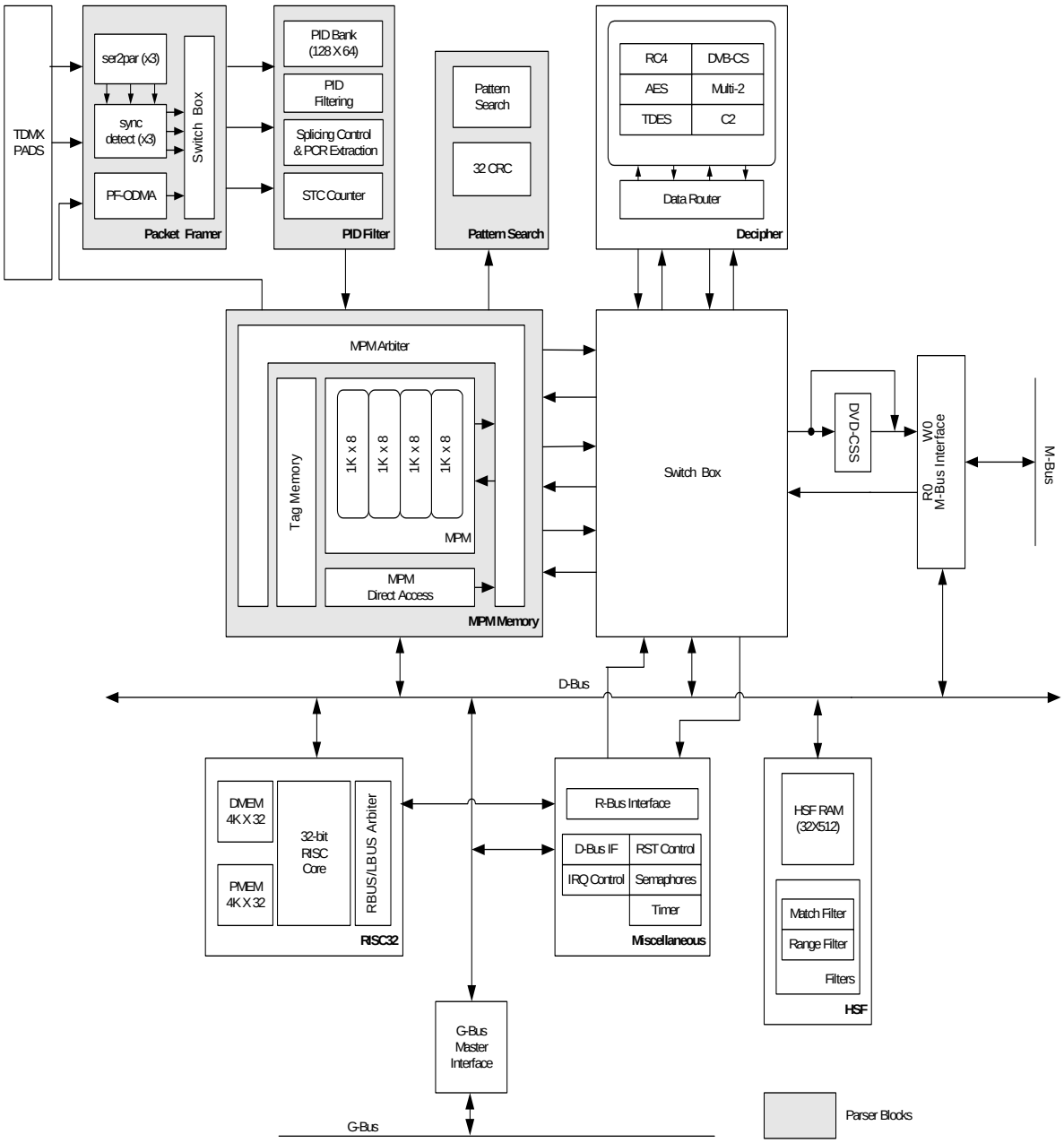


Figure 10-1 **Transport demultiplexer block diagram**

10.2 Introduction

The SMP8634 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 81Mbps each, with an aggregate total of up to 243Mbps. On-chip demultiplexing supports the following formats:

- DVD-Video, DVD-Audio, SVCD and VCD bitstreams
- HD DVD-Video, BD-ROM, BD-RE and BD-R 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, SMPTE 421M (VC-1) over MPEG-2 transport

The TDMX unit consists primarily of a dedicated 32-bit RISC processor and parser blocks along with specialized support hardware. The parser block contains a multiport memory (MPM), organized as 1K by 32-bit (4 KB) with 6 independent access ports. 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 bit stream data may be input via the transport stream inputs, 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 SMP8634 (such as EPG, etc.) are written to the DRAM and passed on to the middleware for processing.

10.3 Features

- Dedicated demultiplexer RISC
  - 32-bit 300MHz (rev. C or later, 200MHz for previous revisions) RISC CPU core (proprietary)
  - 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 three transport or program streams, up to 243 Mbps total aggregate
  - Hardware/software configurable two 8-bit parallel and three serial input interfaces
  - Programmable clock and data valid polarity
  - Transport streams may also be input via PCI, IDE, etc.
  - Programmable packet length to remove FEC and other data
- 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 and OpenCable® CableCARD
- 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

10.4 Functional Description

The demultiplexer contains 3 independent transport stream processing channels. This provides the capability to handle up to 3 independent transport streams simultaneously. In addition, a program or PES bitstreams from external DRAM can be processed for DVD, DVR, etc.applications.

At any given time, only 2 of the 3 data processing pipelines can be used for the system clock recovery (by tracking the encoder clock rate). The decoder clock frequency is adjusted according to the error terms computed, based on the differences in the value of the extracted PCR and the value of the STC running on the decoder system. The encoder samples its system clock counter periodically, and inserts the value in the bit stream as PCR.

As the decoder receives the sampled value, the current value of the decode system clock counter is compared against the extracted PCR. The difference produces the error term, that controls the clock rate change of the external VCXO. All the AV display clocks in the chip will be generated and synchronized to this adjusted clock. Without this clock recovery scheme, ‘push’ type applications suffer from buffer underflow or over flow as there is limited control over the input data.

The packet framer block provides a seamless interface to various channel decoders (e.g. QAM, QPSK, 8-PSK, FEC) and accepts direct inputs from multiple detachable external conditional access modules, such as the common interface (CI) and the point of deployment (POD) data ports. Pre-recorded data (program stream from a DVD ROM, or HDD etc.) can also be accepted from the external DRAM.

In general, the inputs from the external channel decoders are sourced from the ‘push’ type models in which the application data is transported in unidirectional manner. The receiving device accepts all the data transmitted from the source stream. Therefore, no or minimal flow-control is provided.

The applications whose inputs are coming from the external DRAM are grouped as ‘pull’ type models, where the data is initially stored at a storage media and is supplied as needed. Hence, a full flow-control is employed.

The input to other pipelines can come from both ‘push’ and ‘pull’ types. As a result, the output video can suffer from occasional frame dropping or repeating at the display device due to the processing data rate mismatch if the data is sourced from ‘push’ type applications.

As the transport stream is ‘locked’ (i.e. the sync byte is found at every 188 valid byte interval), the packet framer starts to output the packets to the PID filtering block in 32-bit packet aligned fashion. Therefore the TS sync byte (0x47) is always located at the MS byte of the 32-bit output word if the word contains the sync byte. For systems using 192-byte transport streams, the 4-byte header is ignored.

During this process if the sync signal is provided by the interface protocol, then the transport sync input can be used to construct the packet-aligned transport stream data. Otherwise, the hardware will search for the transport stream sync byte for packet alignment. In the case of the demultiplexer configured to accept the serial transport stream inputs, the packet framer undergoes a bit-serial to 32-bit word data format conversion.

According to the PID settings, the input data packets are filtered, so that only the selected packets are kept in the MPM. For these selected packets, the PID filter also creates a corresponding packet attribute as TAG. This information is stored in the MPM at the end of each packet filtering process.

By saving the tag information at the end of each packet filtering process, the number of entries in the tag memory is always the same as the number of complete packets in the MPM. The tag information contains information like the packet types (e.g. audio, video or system), the payload destinations, the data parsing information and the sync byte location in the MPM etc. This tag data supplies sufficient information about the packet, so that the processing time for the RISC is reduced.

The MPM is comprised of four 1K by 8-bit single port memories for storing the TS packets and another memory (32-bit wide, 64 words deep) for storing the tag information. These four 8-bit, 1KB MPM memories are stacked side-by-side in order to form one 32-bit by 1K memory. This 32-bit by 1K memory is then segmented into four equal 32x256 memories, each serving as 1KB stream buffers. The first three segments are used to store the transport packets from each TS channel, and the last segment processes of the data streams from the external memory.

This structure allows not only for 32-bit access in a single clock cycle, but also the access data need not be word-aligned, so that the data can be accessed from any byte location of the MPM.

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 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 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.

10.4.1 Packet Framer

The primary purpose of the packet framer block is to accept asynchronous transport streams in order to produce 32-bit packet-aligned data output, along with other packet control signals that are synchronous to the internal system clock.

The main features of this module are,

- This module can receive MPEG TS stream data with the following three configurations
  - Three external serial input ports with data, valid, clock, and/or packet sync signals. The input data clocks can operate at any frequency up to 87.9MHz (81MHz x 204/188). Each parallel port can be configured to accept two serial transport streams; and the host can select any three serial inputs to process.
  - One external 8-bit parallel input port and two serials with its own data, valid and/or packet sync signals. When the input port is configured in the parallel mode, the maximum clock rate should be 10.99MHz (10.125MHz x 204/188) which is equivalent to 81Mbps in input data rate.
  - One internal 32-bit OF-DMA input at a maximum average input data rate of 2.53125Mword/sec and two additional streams from external SPI PAD inputs.
- 32-bit word aligned output data with unified data flow control signals
- Programmable sync byte detection either by the hardware or assisted by the input sync signal
- Automatic sync-lock and sync-loss detection based on software controlled hysterisis mechanism
- Data synchronization to internal system clock

Up to 2 independent channel demodulators can be connected and simultaneously processed. The interface protocol of one input channel may not be the same as the others. The SMP8634 provides 2 or 3 digital tuner input ports. Tuners from different vendors can be connected at the same time. Each of these tuners can have its own protocol.

Four major logical partitions make up the packet framer: input switchbox, packet alignment, sync detect and the packet framer output DMA (PF-ODMA) blocks. In order to support concurrent processing of three independent channel interfaces, packet alignment and the sync detect blocks are instantiated three times, while the switchbox and the PF-ODMA are instantiated only once.

The PF-ODMA provides the filtering capability of the TS packets from external the DRAM, such as packets transported over the IP network or recorded in the hard disk. The unfiltered transport stream is temporarily transferred to the MPM from the external DRAM through the IDMA channels. This transport stream is then transferred to the input switchbox via PF-ODMA.

The input switchbox accepts the data traffic from either the input pads or the PF-ODMA and directs it to the assigned processing channels. The signal activities on the input ports are monitored by the input switchbox to see if channel demodulators are connected. When connected, the logic block reports the status to the RISC to communicate with the host CPU in order to obtain the channel routing instructions.

The packet synchronization is performed by the sync detect block. This is done by observing the input data patterns and searching for the sync byte at the fixed packet interval. To prevent any false detection, this block keeps track of the sync state hysterisis by counting the number of consecutive valid packets. Once the packet synchronization is established, complete transport packets are transferred to the PID filtering block after the byte-to-word conversion by the packet alignment block.

10.4.2 PID Filter Block

The PID filter block extracts the PID and compares it against the entries in the PID bank to see if there is any match. This is done by examining the 4-byte transport header of each packet.

The PID bank is a 128x64 memory. It is constructed with four single port 32x64 bit SRAM memories, whose entry represents PID with associated packet attributes. This SRAM can hold up to 128 PID entries that are accessed by all three the PID filtering channels.

The main features of the PID filter block are,

- Automatic PID switch for splicing control
- Hardware PCR extraction with dedicated PID filter register for each channel
- 128 general PID entries that can be accessed by all the 3 channels

- 4 additional hardwired PID filters for NULL, PAT, CAT and MGT for each channel
- Software controlled TS-level error handling
- Packet continuity check by monitoring the CC field in the TS header
- TAG entry creation for reducing the packet parsing processing load of the RISC
- PID filtering logic can be bypassed, allowing software PID filtering
- Optional hardware adaptation field processing and user data extraction

No content modification is made to the packet during the PID filtering, meaning. If a match is found, then the whole packet is stored in the MPM along with the packet attributes in the TAG memory. When no match is found, the packet is discarded by resetting the write pointer to the previous packet start address.

The packet PID is compared against the PCR PID value before it undergoes the general PID comparisons with the PID bank entries. Each filtering channel contains a dedicated register-based PCR PID whose value can be set by either the host CPU or the RISC. If the PID of a packet matches with that of PCR, then the PCR value in the packet adaptation field is copied to a register for the RISC to initiate the clock recovery process.

The result from PCR PID comparison will not affect the result of the general PID comparison. Regardless of the result in PCR PID comparison, the packet PID will be compared against the PID entries in the PID bank as this packet may contain either audio or video payload.

During the PID filtering process, packet errors are checked in two ways: by examining the state of the transport error indicator (TEI) bit, and by observing the continuity counter (CC) 4-bit field. Depending on the attribute settings for a PID, the packets with are either discarded or stored in the MPM.

The splicing control is also handled by this block by monitoring the splicing counter in the adaptation field. The splicing requires two PID values (these two PID values may be identical). When the splicing is enabled, the splicing logic monitors the splicing counter in the adaptation field of the packets associated with previous PID value, looking for the splicing point. Meanwhile, this logic block continues storing packets with the previous PID value. As the splicing counter reaches zero, the old PID is replaced with the new PID.

The time-base of the new packets need not be the same as that of the old packets. This causes PCR discontinuity. In this case, the host CPU is notified of the nature of PCR discontinuity, so that the AV display units are handled properly.

Once the PID matching is done successfully, the attributes for the packet is prepared and stored in the TAG memory as the last word of the packet enters the MPM. Therefore, the number of entries in the TAG memory equals to the number of TS packets stored in MPM. The contents of TAG memory include the pre-parsed information such as the type of each packet (i.e. audio, video or system), routing information (i.e. playback or recording) and the packet start address in the MPM. These packet attributes come from both the host and the hardware logic states, in order to facilitate the packet parsing effort by the RISC.

The PID filtering for each channel can be disabled by setting the corresponding bit in the control register. When this bit is set, the packet filtering is disabled and all the packet data will be stored in the MPM. The associated TAG information is be created.

10.4.3 Cipher Block

The ciphering module contains several modules to encrypt/decrypt the stream data. It supports the following algorithms:

1. DES/Triple DES – (ECB, OFB, CBC)
2. AES – (ECB, OFB, CTR, CBC, CFB)
3. RC4
4. DVD
5. DVB-CSA
6. Multi-2 (ECB, OFB, CBC, CFB)
7. C2 block (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.

10.4.3.1 DES Encryption/Decryption Unit

The DES encryption/decryption unit can either encrypt or decrypt a 64-bit data block using the DES or 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.

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 10-1 **DES encryption/decryption 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.

10.4.3.2 AES Encryption/Decryption Unit

The AES encryption/decryption unit can either encrypt or decrypt a 128, 192 or 256-bit data block using the AES algorithm. The supported modes are ECB, OFB and CTR, both in encryption and decryption. The supported key lengths are 128, 192 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 can be up to 255.

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 10-2 **AES encryption/decryption 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)

10.4.3.3 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 10-3 **RC4 encryption/decryption 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

10.4.3.4 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.

10.4.3.5 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.

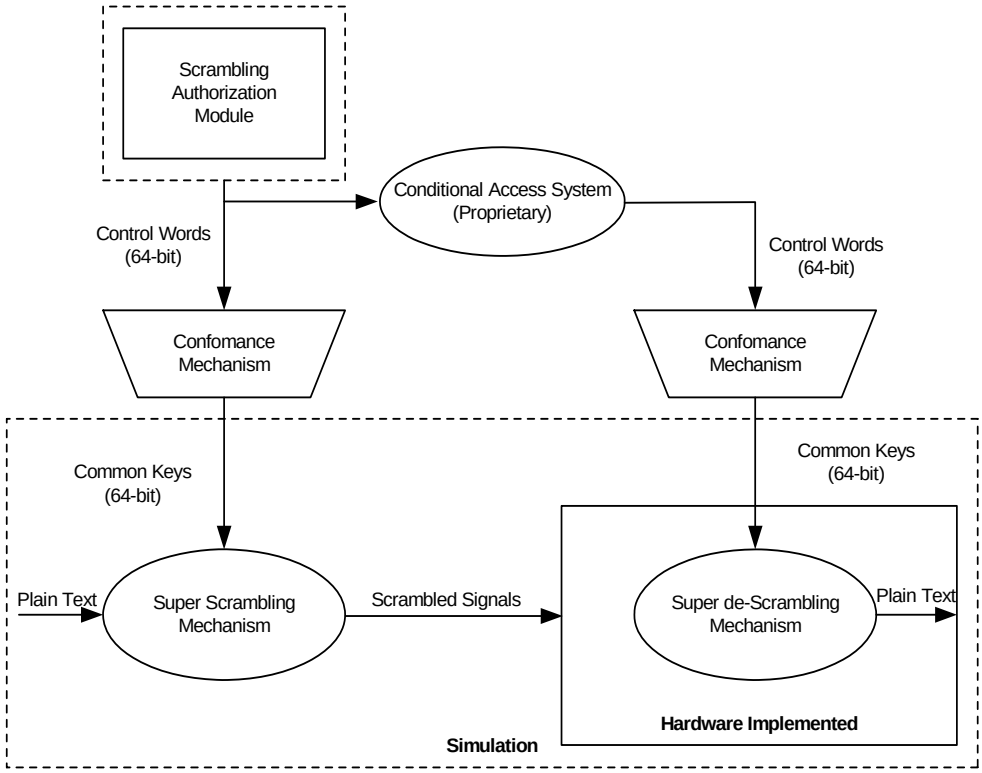


Figure 10-2 **Overview of DVB-CSA**

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.

10.4.3.6 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.

In the following performance table N is the number of 64-bit size blocks, R is the number of residual bytes, Twork is the number of cycles for the working keys and Tbase is the number of cycles for the base submodule.

Table 10-4 **Performance index for various cases**

Round	Size	N	R	T_{Work}	T_{Base}	Cycles	Bits/Cycle
1	1	0	1	14	2	23	0.34
1	8	1	0	6	2	29	2.20
16	8	1	0	6	21	48	1.30
32	1	0	1	14	21	62	0.12
32	8	1	0	6	21	68	0.94
32	128	16	0	6	21	698	1.40
32	255	31	7	6	21	1369	1.40
128	255	31	7	6	161	5209	0.39

10.4.3.7 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 SMP8634 is used as the decoder.

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.

10.4.4 Clock Recovery Process

In the MPEG system it is assumed that the network has a constant end-to-end delay, and the encoded bit stream is delivered to the decoder over the constant delay network. The elapsed time between two time instances; capturing of a frame by the encoder and the displaying the frame by the decoder, is constant. This delay is independent of both the coding type of a frame (i.e. I, P and B), and of the network types the data is transmitted over (e.g. ATM, SONNET, DBS etc.).

Due to the packet based processing nature of the network, the instantaneous bit rate of the network is not always constant. Rather, in most of the cases, the bit rate is burst and introduces jitter. Consequently, the constant network delay model fails in most real time applications. Typically, a well-behaved system accommodates the network jitter by putting a large amount of smoothing buffer at the decoder front end so that the output of the smoothing buffer is constant.

The clock recovery scheme at the decoder end has been developed based on the constant network delay assumption, by transporting the periodically sampled encoder system time to the decoder as part of the coded bit stream. These sampled time stamps are used to adjust the decoder clock frequency by comparing against its local time stamps.

The difference between the two time stamps (one from encoder and the other from the decoder) is fed back to the clock controller for frequency adjustment. When the decoder clock continuously runs faster than the encoder clock, the decoder does not get enough data, causing the same frame to be displayed repeatedly to fill the gap. When the frequency of the decoder clock is lower than that of the encoder, the amount of data from encoder is more than the decoder can handle, causing the dropping of data (frames) in order to keep up with the encoder.

Therefore, to avoid losing or repeating the frames, the encoder and decoder share a common time-base that synchronizes the display rate of decoded audio, video and other associated data. This also reduces the number of buffers needed for maintaining the appropriate frame rates, regardless of the amount of the data required for encoding a frame.

The clock reference resides within the MPEG packet. Although, the decoder can use other means to synchronize to the encoder clock, the use of a digital phase locked loop circuitry (DPLL) is discussed in this section.

For the decoder to decode and to display an MPEG bit stream, it should be able to generate a clock whose characteristics are the same as the encoder clock that produces the bit stream. Normally, a PLL is composed of the following parts: a reference clock, a system clock, a frequency multiplier and divider, a charge pump and control, a low pass filter and a voltage controlled oscillator (VCO). In a typical MPEG system, the PLL circuit consists of the following with all frequency multiplier and divider constants being 1:

1. Periodically sampled encoder system time stamps (SCR or PCR) as the clock reference.
2. Decoder system clock as the system clock.
3. PCR and STC compare as phase and frequency detectors.
4. Digital low pass filter and scaler and 12-bit DAC as the charge pump and controllers.
5. Analog low pass filter as the low pass filter to the VCO.
6. VCXO as the VCO.

When the demultiplexer detects the PCR information from the bit stream, it samples the current system STC value as it extracts the last bit of the PCR value from the bit stream. The local STC value is subtracted from the extracted PCR and the difference is forwarded to the CPU for further computation. This value is used to provide fine adjustment to the 27MHz system clock frequency. The local STC up-counters are driven by this system clock, and have the same relationship as PCR or SCR. The counter values are sampled and compared whenever the availability of a PCR is detected from the bit stream.

Both SCR and PCR are sampled at the 27MHz encoder clock and transmitted in two pieces, a 33-bit PCR base, sampled at 90KHz and a 9-bit PCR extension, sampled at 27MHz.

Since the PCR is sampled at the time when the last bit of the 33-bit base leaves the encoder, the MPEG spec requires this value to be compared against the value of the STC counter when LSB of the PCR base enters the decoder.

10.4.5 Section/ECM Filtering

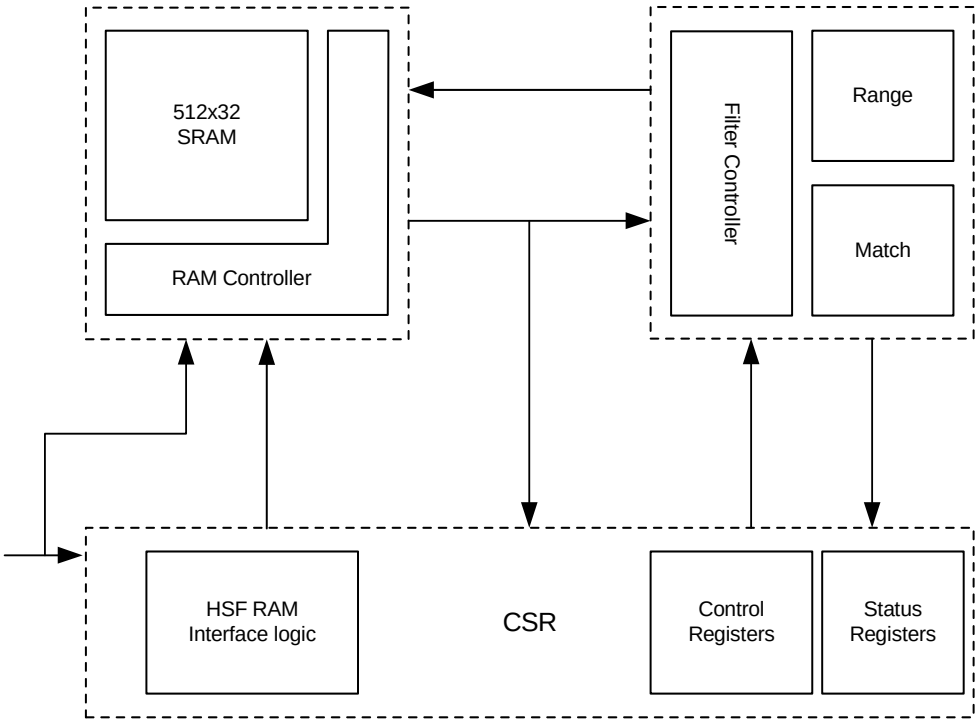


Figure 10-3 **Hardware section filter**

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.

10.4.5.1 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.

10.4.5.2 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.

10.5 Pin Description

10.5.1 Transport Demultiplexer Pins

Table 10-5 **SPI transport stream interface pin descriptions**

Pin Name	Ball ID	Direction	Description
TDMX_GPIO0	AB30	B	Transport demultiplexer GPIO 0
TDMX_GPIO1	AB31	B	Transport demultiplexer GPIO 1
TS0_CLK	AF34	I	SPI port clock input data is transferred on the positive-going edge of this clock. When the port is operated in the SSI mode, this pin is not used.
TS0_IN_D0	AE33	I	SPI input data bit 0 (LSB). When the port is operated in the SSI mode, this pin is the SSI0_DATA input.
TS0_IN_D1	AF33	I	SPI input data bit 1. When the port is operated in the SSI mode, this pin is the SSI0_CLK input.
TS0_IN_D2	AE32	I	SPI input data bit 2. When the port is operated in the SSI mode, this pin is the SSI0_SYNC input.
TS0_IN_D3	AF32	I	SPI input data bit 3. When the port is operated in the SSI mode, this pin is the SSI0_VLD input.
TS0_IN_D4	AE31	I	SPI input data bit 4. When the port is operated in the SSI mode, this pin is the SSI1_DATA input.
TS0_IN_D5	AF31	I	SPI input data bit 5. When the port is operated in the SSI mode, this pin is the SSI1_CLK input.
TS0_IN_D6	AE30	I	SPI input data bit 6. When the port is operated in the SSI mode, this pin is the SSI1_SYNC input.
TS0_IN_D7	AF30	I	SPI input data bit 7 (MSB). When the port is operated in the SSI mode, this pin is the SSI1_VLD input.
TS0_IN_SYNC	AE34	I	SPI sync (selectable polarity, active high/low). Identifies the first byte of a packet. When the port is operated in the SSI mode, this pin is not used.
TS0_IN_VLD	AD30	I	SPI data valid (active high). Indicates valid transport packet bytes. When the port is operated in the SSI mode, this pin is not used.
TS1_CLK	AC34	I	SPI port clock input data is transferred on the positive-going edge of this clock. When the port is operated in the SSI mode, this pin is not used.
TS1_IN_D0	AD34	I	SPI input data bit 0 (LSB). When the port is operated in the SSI mode, this pin is the SSI2_DATA input.
TS1_IN_D1	AB33	I	SPI input data bit 1. When the port is operated in the SSI mode, this pin is the SSI2_CLK input.
TS1_IN_D2	AC33	I	SPI input data bit 2. When the port is operated in the SSI mode, this pin is the SSI2_SYNC input.

Table 10-5 **SPI transport stream interface pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
TS1_IN_D3	AD33	I	SPI input data bit 3. When the port is operated in the SSI mode, this pin is the SSI2_VLD input.
TS1_IN_D4	AC32	I	SPI input data bit 4.
TS1_IN_D5	AD32	I	SPI input data bit 5.
TS1_IN_D6	AC31	I	SPI input data bit 6.
TS1_IN_D7	AD31	I	SPI input data bit 7 (MSB).
TS1_IN_SYNC	AB32	I	SPI sync (selectable polarity, active high/low). Identifies the first byte of a packet. When the port is operated in the SSI mode, this pin is not used.
TS1_IN_VLD	AC30	I	SPI data valid (active high). Indicates valid transport packet bytes. When the port is operated in the SSI mode, this pin is not used.

10.6 Electrical Characteristics

10.6.1 Transport Demultiplexer DC Electrical Characteristics

Table 10-6 **Transport demultiplexer DC electrical characteristics**

Symbol	Description	Units	Min	Typ	Max
I_{OH}[1]	High level output current (@V_{OH} = 2.4V)	mA	12	27	46
I_{OL}	Low level output current (@V_{OL} = 0.4V)	mA	7	13	19
V_{IH}[2]	Input high voltage	V	2		5.5
V_{IL}	Input low voltage	V	-0.3		0.8
R_{PU}[3]	Pull-up resistor value	Kohms	38	54	83

1. Parameter applies to TDMX_GPIO[1:0] when configured as outputs.
2. Parameter applies to TDMX_GPIO[1:0] when configured as inputs, and all other TS input interface signals.
3. Applies to TDMX_GPIO[1:0] pins.

10.6.2 Transport Demultiplexer AC Electrical Characteristics

10.6.2.1 8-bit Parallel (SPI) Input Mode

Table 10-7 **Transport demultiplexer AC characteristics**

Symbol	Units	Min	Typ	Max
T_{TSCLK}	ns	98.77		
T_{TSCLKH}	ns	$3xt_{SYSCLK}$		
T_{TSCLKL}	ns	$3xt_{SYSCLK}$		
T_{TSDSU}	ns	0		
T_{TSDH}	ns	$4xt_{SYSCLK}$		

10.6.2.2 Serial Input (SSI) Mode

Table 10-8 **Transport demultiplexer AC characteristics**

Symbol	Units	Min	Typ	Max
T_{TSCLK}	ns	12.35		
T_{TSCLKH}	ns	4.9		
T_{TSCLKL}	ns	4.9		
T_{TSDSU}	ns	4		
T_{TSDH}	ns	1		

10.7 Timing Diagrams

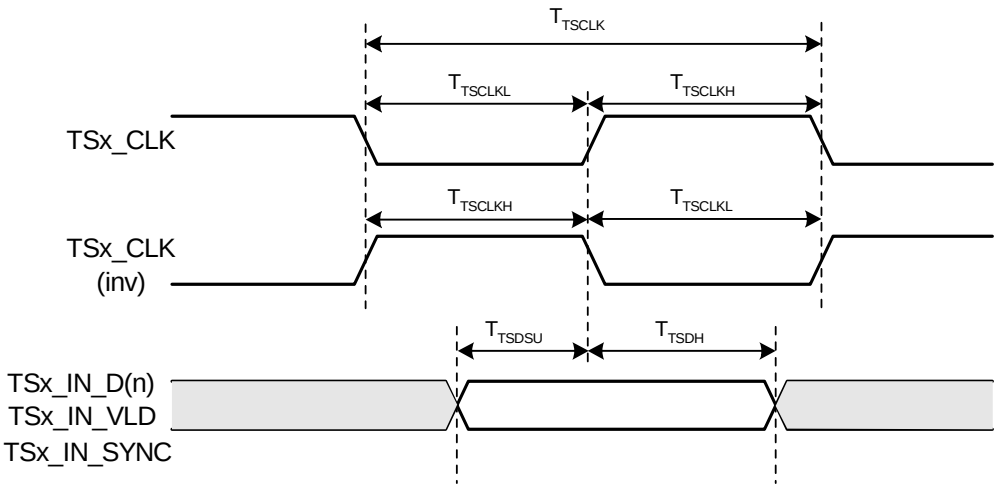


Figure 10-4 **TS input timing diagram**

10.7.1 Supported Transport Stream Input Interfaces

Serial Data Input Protocol

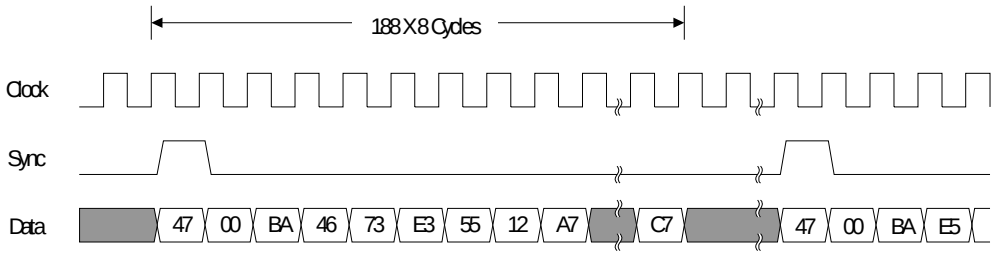


Figure 10-5 **Clock, Data, Sync Protocol**

The above figure shows the 3-pin interface protocol, where the sync signal indicates the start of a packet. With this protocol, the data is assumed to be valid for next 188 X 8 clock cycles, and any data after 188th clock cycle will be discarded until a new sync is asserted

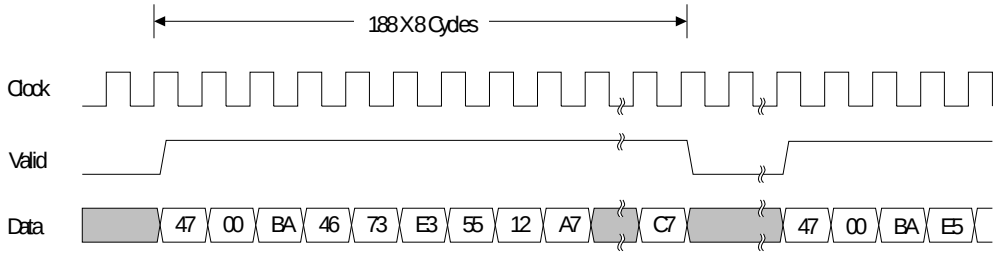


Figure 10-6 **Clock, Data, Valid Protocol**

Another possible 3-pin interface is shown in the above figure. In this protocol, the valid signal qualifies the data at the corresponding clock cycle. Since the channel decoder processes packet-wise data transfer in general, the valid signal is asserted during the entire packet time frame. However, with this protocol, intra-packet gaps are allowed, i.e. the channel decoder can de-assert the valid signal at any clock cycle.

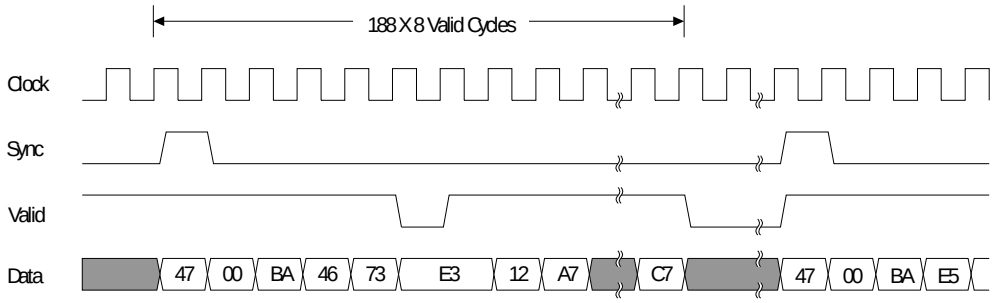


Figure 10-7 **Clock, Data, Sync, Valid Protocol**

The above figure shows a 4-pin input interface that is composed of clock, data, sync and valid signals. The assertion of sync indicates the start of a packet, while the valid signal qualifies the validity of the data at the clock cycle. Although, a de-assertion of the valid signal from the channel decoder during each packet time frame (i.e. 1503 subsequent cycles after sync is asserted) is unlikely due to the nature of FEC coder, this demultiplexer allows inter-packet gaps.

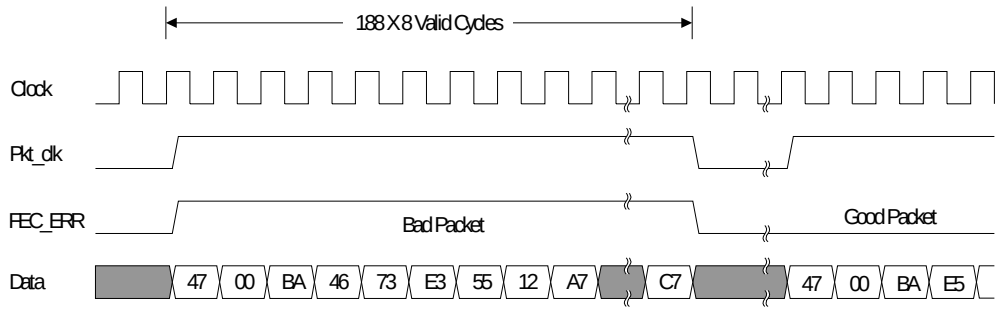


Figure 10-8 **Traditional TS Input Protocol**

The above figure depicts the traditional serial input interface where the interface is composed of four signals: channel clock (clock), clock valid (PKT_CLK), FEC error (FEC_ERR) and serial data (data). When the valid signal in the previous protocol is replaced with the inverted FEC error signal, this protocol can be treated the same as the clock, valid, data 3-pin interface protocol.

Parallel Data Input Protocol

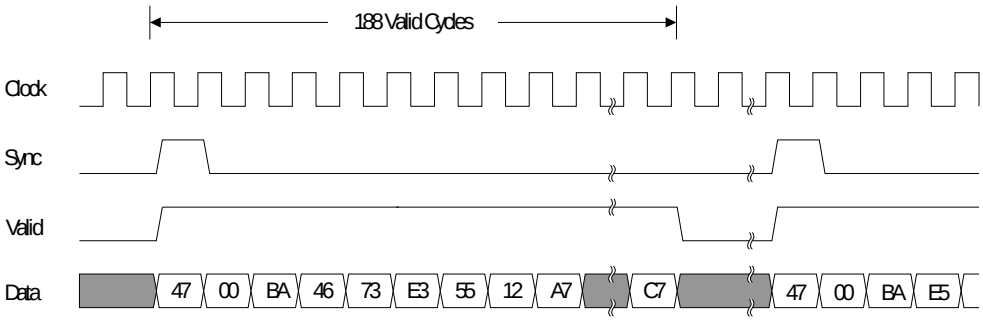


Figure 10-9 **Parallel Input Protocol**

The 11-pin parallel input protocol is shown in the above figure. This protocol consists of the same signals as in the clock, data, sync, data protocol except for the 8-bit data interface. Unlike the other serial input protocols, the 8-bit data is latched at each rising edge of the channel clock.

11 Pin Information

11.1 Pin Layout

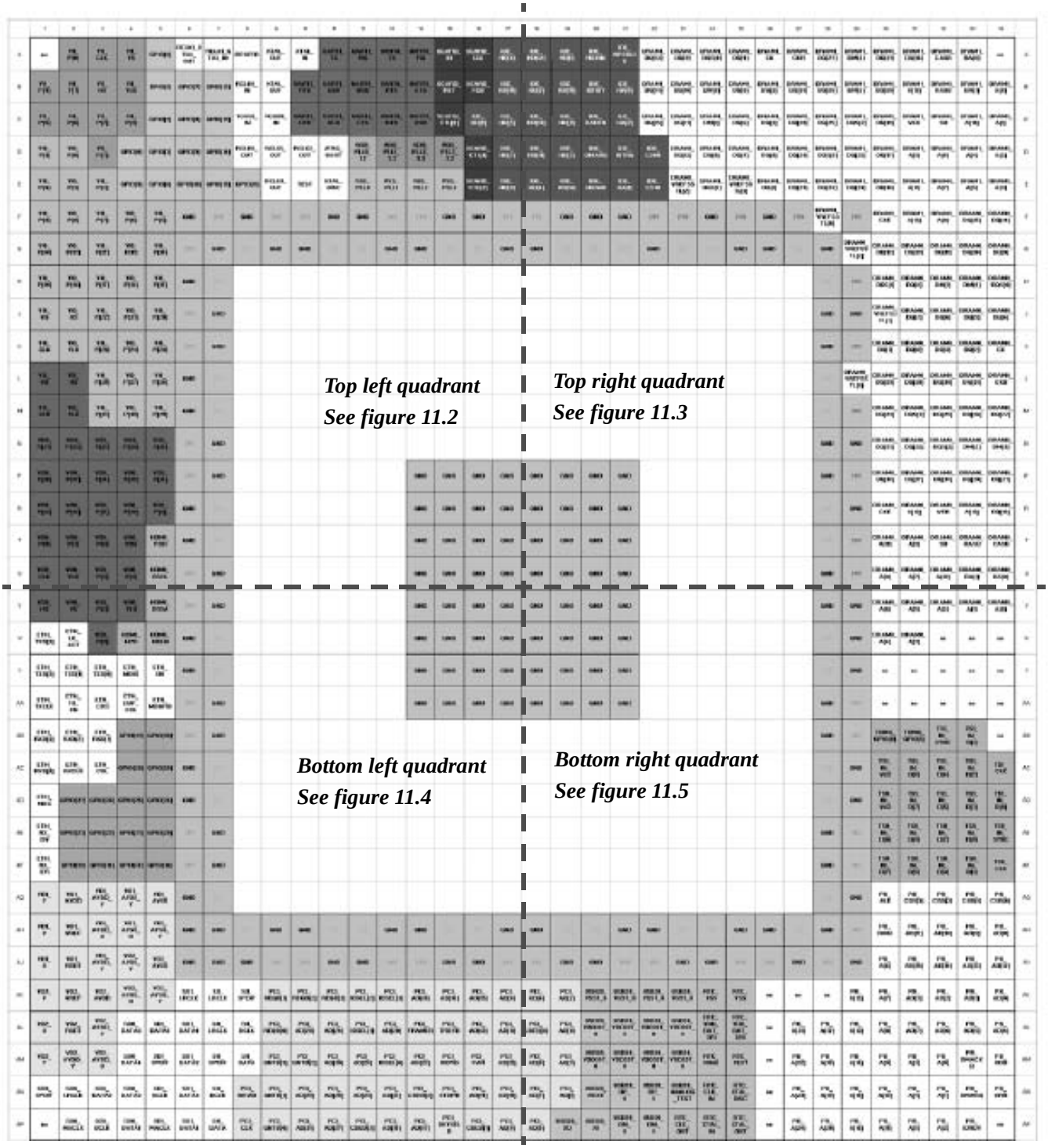


Figure 11-1 **Overview of SMP8634 BGA**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	NC	VI1_P0	VI1_CLK	VI1_VS	GPIO0	RCLK1_XTAL_OUT	RCLK1_XTAL_IN	RESET#	XTAL_OUT	XTAL_IN	UART1_TX	UART1_RX	UART0_TX	UART0_RX	SCARD_IO	SCARD_CLK	IDE_D6
B	VI1_P4	VI1_P1	VI1_HS	VI1_VLD	GPIO1	GPIO7	GPIO12	RCLK0_IN	XTAL_BUF	UART1_RTS	UART1_DSR	UART0_DCD	UART0_RTS	UART0_CTS	SCARD_RST	SCARD_FC#	IDE_D10
C	VI1_P5	VI1_P6	VI1_P2	VI1_P3	GPIO2	GPIO8	GPIO13	VCXO1_IN	VCXO0_IN	UART1_DTR	UART1_DCD	UART1_CTS	UART0_DTR	UART0_DSR	SCARD_CTL0	IDE_D8	IDE_D5
D	VI0_P1	VI0_P0	VI1_P7	GPIO4	GPIO3	GPIO9	GPIO14	RCLK0_OUT	RCLK1_OUT	RCLK2_OUT	JTAG_UART#	VDD_PLL0_1V2	VDD_PLL1_3V3	VDD_PLL2_3V3	VDD_PLL3_3V3	SCARD_CTL1	IDE_D7
E	VI0_P4	VI0_P3	VI0_P2	GPIO5	GPIO6	GPIO10	GPIO15	GPIO11	RCLK3_OUT	TEST	XTAL_DISC	VSS_PLL0	VSS_PLL1	VSS_PLL2	VSS_PLL3	SCARD_CTL2	IDE_D9
F	VI0_P9	VI0_P8	VI0_P7	VI0_P6	VI0_P5	VSS	VDD_3V3	VSS	VDD_3V3	VDD_3V3	VSS	VSS	VDD_3V3	VDD_3V3	VSS	VSS	VDD_3V3
G	VI0_P14	VI0_P13	VI0_P12	VI0_P11	VI0_P10	VDD_3V3	VSS	VDD_1V2	VSS	VSS	VDD_1V2	VDD_1V2	VSS	VSS	VDD_1V2	VDD_1V2	VSS
H	VI0_P19	VI0_P18	VI0_P17	VI0_P16	VI0_P15	VSS	VDD_1V2										
J	VI0_VS	VI0_HS	VI0_P22	VI0_P21	VI0_P20	VDD_3V3	VSS										
K	VI0_CLK	VI0_VLD	VI0_P25	VI0_P24	VI0_P23	VDD_3V3	VSS										
L	VI2_VS	VI2_HS	VI0_P28	VI0_P27	VI0_P26	VSS	VDD_1V2										
M	VI2_CLK	VI2_VLD	VI0_P31	VI0_P30	VI0_P29	VSS	VDD_1V2										
N	VO0_P23	VO0_P22	VO0_P21	VO0_P20	VO0_P19	VDD_3V3	VSS										
P	VO0_P18	VO0_P17	VO0_P16	VO0_P15	VO0_P14	VDD_3V3	VSS							VSS	VSS	VSS	VSS
R	VO0_P13	VO0_P12	VO0_P11	VO0_P10	VO0_P9	VSS	VDD_1V2							VSS	VSS	VSS	VSS
T	VO0_P8	VO0_P7	VO0_P6	VO0_P5	HDMI_PD#	VSS	VDD_1V2							VSS	VSS	VSS	VSS
U	VO0_CLK	VO0_VLD	VO0_P4	VO0_P3	HDMI_DSCL	VDD_3V3	VSS							VSS	VSS	VSS	VSS

Figure 11-2 **SMP8634 pin diagram (top left quadrant)**

18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	
IDE_D12	IDE_D1	IDE_IOR#	IDE_NPCBLID	DRAM1_DQ13	DRAM1_DQ9	DRAM1_DQS0	DRAM1_DQ4	DRAM1_CK	DRAM1_CK#	DRAM1_DQ27	DRAM1_DM3	DRAM1_DQ21	DRAM1_DQ16	DRAM1_CAS#	DRAM1_BA0	NC	A
IDE_D3	IDE_D15	IDE_IORDY	IDE_A0	DRAM1_DQ14	DRAM1_DQ10	DRAM1_DM0	DRAM1_DQ5	DRAM1_DQ2	DRAM1_DQ31	DRAM1_DQ26	DRAM1_DM2	DRAM1_DQ20	DRAM1_A13	DRAM1_RAS#	DRAM1_BA1	DRAM1_A0	B
IDE_D13	IDE_D0	IDE_ACK#	IDE_A2	DRAM1_DQ15	DRAM1_DQ11	DRAM1_DM1	DRAM1_DQ6	DRAM1_DQ3	DRAM1_DQ30	DRAM1_DQ25	DRAM1_DQS2	DRAM1_DQ19	DRAM1_WE#	DRAM1_CS#	DRAM1_A10	DRAM1_A1	C
IDE_D11	IDE_D2	IDE_DMARQ	IDE_IRQ	IDE_CS0#	DRAM1_DQ12	DRAM1_DQ8	DRAM1_DQ7	DRAM1_DQ0	DRAM1_DQ28	DRAM1_DQS3	DRAM1_DQ22	DRAM1_DQ17	DRAM1_A9	DRAM1_A6	DRAM1_A4	DRAM1_A2	D
IDE_D4	IDE_D14	IDE_IOW#	IDE_A1	IDE_CS1#	DRAM1_VREFSSTL2	DRAM1_DQS1	DRAM1_VREFSSTL1	DRAM1_DQ1	DRAM1_DQ29	DRAM1_DQ24	DRAM1_DQ23	DRAM1_DQ18	DRAM1_A11	DRAM1_A7	DRAM1_A5	DRAM1_A3	E
VDD_3V3	VSS	VSS	VSS	VDD_2V5	VDD_2V5	VSS	VDD_2V5	VSS	VDD_2V5	DRAM1_VREFSSTL0	VDD_2V5	DRAM1_CKE	DRAM1_A12	DRAM1_A8	DRAM0_DQ15	DRAM0_DQ14	F
VSS	VDD_1V2	VDD_1V2	VDD_1V2	VSS	VDD_1V2	VDD_1V2	VSS	VSS	VDD_1V2	VSS	DRAM0_VREFSSTL2	DRAM0_DQ12	DRAM0_DQ13	DRAM0_DQ11	DRAM0_DQ10	DRAM0_DQ9	G
										VDD_1V2	VDD_2V5	DRAM0_DQS1	DRAM0_DQ8	DRAM0_DM1	DRAM0_DM0	DRAM0_DQS0	H
										VSS	VSS	DRAM0_VREFSSTL1	DRAM0_DQ7	DRAM0_DQ6	DRAM0_DQ5	DRAM0_DQ4	J
										VSS	VDD_2V5	DRAM0_DQ1	DRAM0_DQ0	DRAM0_DQ3	DRAM0_DQ2	DRAM0_CK	K
										VDD_1V2	DRAM0_VREFSSTL0	DRAM0_DQ29	DRAM0_DQ28	DRAM0_DQ30	DRAM0_DQ31	DRAM0_CK#	L
										VDD_1V2	VDD_2V5	DRAM0_DQ24	DRAM0_DQS3	DRAM0_DQ25	DRAM0_DQ26	DRAM0_DQ27	M
										VSS	VSS	DRAM0_DQ23	DRAM0_DQ22	DRAM0_DQS2	DRAM0_DM2	DRAM0_DM3	N
VSS	VSS	VSS	VSS							VSS	VDD_2V5	DRAM0_DQ18	DRAM0_DQ17	DRAM0_DQ19	DRAM0_DQ20	DRAM0_DQ21	P
VSS	VSS	VSS	VSS							VDD_1V2	VSS	DRAM0_CKE	DRAM0_A12	DRAM0_WE#	DRAM0_A13	DRAM0_DQ16	R
VSS	VSS	VSS	VSS							VDD_1V2	VDD_2V5	DRAM0_A11	DRAM0_A9	DRAM0_CS#	DRAM0_RAS#	DRAM0_CAS#	T
VSS	VSS	VSS	VSS							VSS	VDD_2V5	DRAM0_A8	DRAM0_A7	DRAM0_A10	DRAM0_BA1	DRAM0_BA0	U

Figure 11-3 **SMP8634 pin diagram (top right quadrant)**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
V	VO0_HS	VO0_VS	VO0_P2	VO0_P1	HDMI_DSDA	VDD_3V3	VSS							VSS	VSS	VSS	VSS
W	ETH_TXD3	NC	VO0_P0	HDMI_HPD	HDMI_MSEN	VSS	VDD_1V2							VSS	VSS	VSS	VSS
Y	ETH_TXD2	ETH_TXD1	ETH_TXD0	ETH_MDIO	NC	VSS	VDD_1V2							VSS	VSS	VSS	VSS
AA	ETH_TXCLK	ETH_TX_EN	ETH_CRS	NC	ETH_MDINT#	VDD_3V3	VSS							VSS	VSS	VSS	VSS
AB	ETH_RXD3	ETH_RXD2	ETH_RXD1	GPIO31	GPIO30	VDD_3V3	VSS										
AC	ETH_RXD0	ETH_RXCLK	ETH_COL	GPIO29	GPIO28	VSS	VDD_1V2										
AD	ETH_MDC	GPIO27	GPIO26	GPIO25	GPIO24	VSS	VDD_1V2										
AE	ETH_RX_DV	GPIO23	GPIO22	GPIO21	GPIO20	VDD_3V3	VSS										
AF	ETH_RX_ER	GPIO19	GPIO18	GPIO17	GPIO16	VDD_3V3	VSS										
AG	VO1_Y	VO1_AVDD_Y	VO1_AVSS_Y	VO1_AVDD	VO1_AVSS	VSS	VDD_1V2										
AH	VO1_V	VO1_AVDD_V	VO1_AVSS_V	VO1_RSET	VO1_VREF	VSS	VSS	VDD_1V2	VSS	VSS	VDD_1V2	VDD_1V2	VSS	VSS	VDD_1V2	VDD_1V2	VSS
AJ	VO1_U	VO1_AVDD_U	VO1_AVSS_U	VO2_RSET	VO2_VREF	VSS	VSS	VSS	VDD_3V3	VDD_3V3	VSS	VSS	VDD_3V3	VDD_3V3	VSS	VSS	VDD_3V3
AK	VO2_Y	VO2_AVDD_Y	VO2_AVSS_Y	VO2_AVDD	VO2_AVSS	SO1_LRCLK	SI1_LRCLK	SI1_SPDIF	PCI_REQ1#	PCI_REQ2#	PCI_REQ3#	PCI_IDSEL2	PCI_IDSEL3	PCI_AD18	PCI_AD16	PCI_AD15	PCI_AD4
AL	VO2_V	VO2_AVDD_V	VO2_AVSS_V	SO0_DATA5	SO1_DATA5	SO1_DATA4	SI0_LRCLK	SI1_BCLK	PCI_REQ0#/GNT#	PCI_AD28	PCI_AD24	PCI_IDSEL1	PCI_AD20	PCI_FRAME#	PCI_TRDY#	PCI_AD13	PCI_AD11
AM	VO2_U	VO2_AVDD_U	VO2_AVSS_U	SO0_DATA4	SO1_SPDIF	SO1_DATA2	SI0_SPDIF	SI1_DATA	PCI_GNT3#	PCI_GNT2#	PCI_AD30	PCI_AD26	PCI_IDSEL0	PCI_AD22	PCI_IRDY#	PCI_PAR	PCI_AD12
AN	SO0_SPDIF	SO0_LRCLK	SO0_DATA2	SO0_DATA3	SO1_BCLK	SO1_DATA3	SI0_BCLK	PCI_INTA#	PCI_GNT1#	PCI_AD29	PCI_AD25	PCI_AD23	PCI_AD21	PCI_CBE2#	PCI_STOP#	PCI_AD14	PCI_AD10
AP	NC	SO0_MCLK	SO0_BCLK	SO0_DATA1	SO1_MCLK	SO1_DATA1	SI0_DATA	PCI_CLK	PCI_GNT0#/REQ#	PCI_AD31	PCI_AD27	PCI_CBE3#	PCI_AD19	PCI_AD17	PCI_DEVSEL#	PCI_CBE1#	PCI_AD8

Figure 11-4 **SMP8634 pin diagram (bottom left quadrant)**

VSS	VSS	VSS	VSS							VSS	VSS	DRAM0_A6	DRAM0_A5	DRAM0_A2	DRAM0_A1	DRAM0_A0	V
VSS	VSS	VSS	VSS							VDD_1V2	VSS	DRAM0_A4	DRAM0_A3	NC	NC	NC	W
VSS	VSS	VSS	VSS							VDD_1V2	VSS	NC	NC	NC	NC	NC	Y
VSS	VSS	VSS	VSS							VSS	VDD_3V3	NC	NC	NC	NC	NC	AA
										VSS	VDD_3V3	TDMX_GPIO0	TDMX_GPIO1	TS1_IN_SYNC	TS1_IN_D1	NC	AB
										VDD_1V2	VSS	TS1_IN_VLD	TS1_IN_D6	TS1_IN_D4	TS1_IN_D2	TS1_CLK	AC
										VDD_1V2	VSS	TS0_IN_VLD	TS1_IN_D7	TS1_IN_D5	TS1_IN_D3	TS1_IN_D0	AD
										VSS	VDD_3V3	TS0_IN_D6	TS0_IN_D4	TS0_IN_D2	TS0_IN_D0	TS0_IN_SYNC	AE
										VSS	VDD_3V3	TS0_IN_D7	TS0_IN_D5	TS0_IN_D3	TS0_IN_D1	TS0_CLK	AF
										VDD_1V2	VSS	PB_ALE	PB_CS3#	PB_CS2#	PB_CS1#	PB_CS0#	AG
VSS	VDD_1V2	VDD_1V2	VSS	VSS	VDD_1V2	VDD_1V2	VSS	VSS	VDD_1V2	VSS	VDD_3V3	PB_DIR#	PB_AD11	PB_AD10	PB_AD9	PB_AD8	AH
VDD_3V3	VSS	VSS	VDD_3V3	VDD_3V3	VSS	VSS	VDD_3V3	VDD_3V3	VSS	VDD_3V3	VSS	PB_A6	PB_AD15	PB_AD14	PB_AD13	PB_AD12	AJ
PCI_AD6	PCI_AD2	USB20_VSSAT_0	USB20_VSSAT_0	USB20_VSSAT_1	USB20_VSSAT_1	RTC_VSS	NC	NC	NC	NC	PB_A12	PB_A7	PB_AD3	PB_AD2	PB_AD1	PB_AD0	AK
PCI_CBE0#	PCI_AD0	USB20_VSSAT_0	USB20_VSSAC	USB20_VDDAC_3V3	USB20_VSSAT_1	RTC_VDD_BAT_3V3	RTC_VDD_BAT_3V3	NC	PB_A21	PB_A17	PB_A13	PB_A8	PB_AD7	PB_AD6	PB_AD5	PB_AD4	AL
PCI_AD9	PCI_AD3	USB20_VDDAT_0_3V3	USB20_VDDAT_0_3V3	USB20_VDDAT_1_3V3	USB20_VDDAT_1_3V3	RTC_RING	RTC_TEST	NC	PB_A22	PB_A18	PB_A14	PB_A9	PB_A3	PB_A0	PB_DMAACK#	PB_RD#	AM
PCI_AD7	PCI_AD1	USB20_REXT	USB20_DP_0	USB20_DP_1	USB20_ATEST	RTC_CLK_IN	RTC_XTAL_DISC	NC	PB_A23	PB_A19	PB_A15	PB_A10	PB_A4	PB_A1	PB_DMARQ	PB_WR#	AN
PCI_AD5	USB20_XO	USB20_XI	USB20_DM_0	USB20_DM_1	RTC_CLK_OUT	RTC_XTAL_IN	RTC_XTAL_OUT	NC	PB_A24	PB_A20	PB_A16	PB_A11	PB_A5	PB_A2	IORDY	NC	AP
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	

Figure 11-5 **SMP8634 pin diagram (bottom right quadrant)**

11.2 Pin Listing by Ball Id

Table 11-1. **SMP8634 pin listing by ball id**

Ball Id	Pin Name
A1	NC
A2	VI1_P0
A3	VI1_CLK
A4	VI1_VS
A5	GPIO0
A6	RCLK1_XTAL_OUT
A7	RCLK1_XTAL_IN
A8	RESET#
A9	XTAL_OUT
A10	XTAL_IN
A11	UART1_TX
A12	UART1_RX
A13	UART0_TX
A14	UART0_RX
A15	SCARD_IO
A16	SCARD_CLK
A17	IDE_D6
A18	IDE_D12
A19	IDE_D1
A20	IDE_IOR#
A21	IDE_NPCBLID
A22	DRAM1_DQ13
A23	DRAM1_DQ9
A24	DRAM1_DQS0
A25	DRAM1_DQ4
A26	DRAM1_CK
A27	DRAM1_CK#
A28	DRAM1_DQ27
A29	DRAM1_DM3
A30	DRAM1_DQ21
A31	DRAM1_DQ16
A32	DRAM1_CAS#
A33	DRAM1_BA0
A34	NC
B1	VI1_P4
B2	VI1_P1
B3	VI1_HS
B4	VI1_VLD
B5	GPIO1
B6	GPIO7
B7	GPIO12
B8	RCLK0_IN
B9	XTAL_BUF
B10	UART1_RTS
B11	UART1_DSR
B12	UART0_DCD
B13	UART0_RTS
B14	UART0_CTS
B15	SCARD_RST
B16	SCARD_FC#
B17	IDE_D10
B18	IDE_D3
B19	IDE_D15
B20	IDE_IORDY
B21	IDE_A0
B22	DRAM1_DQ14
B23	DRAM1_DQ10
B24	DRAM1_DM0
B25	DRAM1_DQ5
B26	DRAM1_DQ2
B27	DRAM1_DQ31
B28	DRAM1_DQ26
B29	DRAM1_DM2
B30	DRAM1_DQ20
B31	DRAM1_A13
B32	DRAM1_RAS#
B33	DRAM1_BA1
B34	DRAM1_A0
C1	VI1_P5
C2	VI1_P6
C3	VI1_P2
C4	VI1_P3
C5	GPIO2
C6	GPIO8
C7	GPIO13
C8	VCXO1_IN
C9	VCXO0_IN
C10	UART1_DTR
C11	UART1_DCD
C12	UART1_CTS
C13	UART0_DTR
C14	UART0_DSR
C15	SCARD_CTL0

Ball Id	Pin Name
C16	IDE_D8
C17	IDE_D5
C18	IDE_D13
C19	IDE_D0
C20	IDE_ACK#
C21	IDE_A2
C22	DRAM1_DQ15
C23	DRAM1_DQ11
C24	DRAM1_DM1
C25	DRAM1_DQ6
C26	DRAM1_DQ3
C27	DRAM1_DQ30
C28	DRAM1_DQ25
C29	DRAM1_DQS2
C30	DRAM1_DQ19
C31	DRAM1_WE#
C32	DRAM1_CS#
C33	DRAM1_A10
C34	DRAM1_A1
D1	VI0_P1
D2	VI0_P0
D3	VI1_P7
D4	GPIO4
D5	GPIO3
D6	GPIO9
D7	GPIO14
D8	RCLK0_OUT
D9	RCLK1_OUT
D10	RCLK2_OUT
D11	JTAG_UART#
D12	VDD_PLL0_1V2

Ball Id	Pin Name
D13	VDD_PLL1_3V3
D14	VDD_PLL2_3V3
D15	VDD_PLL3_3V3
D16	SCARD_CTL1
D17	IDE_D7
D18	IDE_D11
D19	IDE_D2
D20	IDE_DMARQ
D21	IDE_IRQ
D22	IDE_CS0#
D23	DRAM1_DQ12
D24	DRAM1_DQ8
D25	DRAM1_DQ7
D26	DRAM1_DQ0
D27	DRAM1_DQ28
D28	DRAM1_DQS3
D29	DRAM1_DQ22
D30	DRAM1_DQ17
D31	DRAM1_A9
D32	DRAM1_A6
D33	DRAM1_A4
D34	DRAM1_A2
E1	VI0_P4
E2	VI0_P3
E3	VI0_P2
E4	GPIO5
E5	GPIO6
E6	GPIO10
E7	GPIO15
E8	GPIO11
E9	RCLK3_OUT

Ball Id	Pin Name
E10	TEST
E11	XTAL_DISC
E12	VSS_PLL0
E13	VSS_PLL1
E14	VSS_PLL2
E15	VSS_PLL3
E16	SCARD_CTL2
E17	IDE_D9
E18	IDE_D4
E19	IDE_D14
E20	IDE_IOW#
E21	IDE_A1
E22	IDE_CS1#
E23	DRAM1_VREFSSTL2
E24	DRAM1_DQS1
E25	DRAM1_VREFSSTL1
E26	DRAM1_DQ1
E27	DRAM1_DQ29
E28	DRAM1_DQ24
E29	DRAM1_DQ23
E30	DRAM1_DQ18
E31	DRAM1_A11
E32	DRAM1_A7
E33	DRAM1_A5
E34	DRAM1_A3
F1	VI0_P9
F2	VI0_P8
F3	VI0_P7
F4	VI0_P6
F5	VI0_P5
F6	VSS

Ball Id	Pin Name
F7	VDD_3V3
F8	VSS
F9	VDD_3V3
F10	VDD_3V3
F11	VSS
F12	VSS
F13	VDD_3V3
F14	VDD_3V3
F15	VSS
F16	VSS
F17	VDD_3V3
F18	VDD_3V3
F19	VSS
F20	VSS
F21	VSS
F22	VDD_2V5
F23	VDD_2V5
F24	VSS
F25	VDD_2V5
F26	VSS
F27	VDD_2V5
F28	DRAM1_VREFSSTL0
F29	VDD_2V5
F30	DRAM1_CKE
F31	DRAM1_A12
F32	DRAM1_A8
F33	DRAM0_DQ15
F34	DRAM0_DQ14
G1	VI0_P14
G2	VI0_P13
G3	VI0_P12
G4	VI0_P11
G5	VI0_P10
G6	VDD_3V3
G7	VSS
G8	VDD_1V2
G9	VSS
G10	VSS
G11	VDD_1V2
G12	VDD_1V2
G13	VSS
G14	VSS
G15	VDD_1V2
G16	VDD_1V2
G17	VSS
G18	VSS
G19	VDD_1V2
G20	VDD_1V2
G21	VDD_1V2
G22	VSS
G23	VDD_1V2
G24	VDD_1V2
G25	VSS
G26	VSS
G27	VDD_1V2
G28	VSS
G29	DRAM0_VREFSSTL2
G30	DRAM0_DQ12
G31	DRAM0_DQ13
G32	DRAM0_DQ11
G33	DRAM0_DQ10
G34	DRAM0_DQ9
H1	VI0_P19
H2	VI0_P18
H3	VI0_P17
H4	VI0_P16
H5	VI0_P15
H6	VSS
H7	VDD_1V2
H28	VDD_1V2
H29	VDD_2V5
H30	DRAM0_DQS1
H31	DRAM0_DQ8
H32	DRAM0_DM1
H33	DRAM0_DM0
H34	DRAM0_DQS0
J1	VI0_VS
J2	VI0_HS
J3	VI0_P22
J4	VI0_P21
J5	VI0_P20
J6	VDD_3V3
J7	VSS
J28	VSS
J29	VSS
J30	DRAM0_VREFSSTL1
J31	DRAM0_DQ7
J32	DRAM0_DQ6
J33	DRAM0_DQ5
J34	DRAM0_DQ4
K1	VI0_CLK
K2	VI0_VLD
K3	VI0_P25

Ball Id	Pin Name
K4	VI0_P24
K5	VI0_P23
K6	VDD_3V3
K7	VSS
K28	VSS
K29	VDD_2V5
K30	DRAM0_DQ1
K31	DRAM0_DQ0
K32	DRAM0_DQ3
K33	DRAM0_DQ2
K34	DRAM0_CK
L1	VI2_VS
L2	VI2_HS
L3	VI0_P28
L4	VI0_P27
L5	VI0_P26
L6	VSS
L7	VDD_1V2
L28	VDD_1V2
L29	DRAM0_VREFSSTL0
L30	DRAM0_DQ29
L31	DRAM0_DQ28
L32	DRAM0_DQ30
L33	DRAM0_DQ31
L34	DRAM0_CK#
M1	VI2_CLK
M2	VI2_VLD
M3	VI0_P31
M4	VI0_P30
M5	VI0_P29
M6	VSS

Ball Id	Pin Name
M7	VDD_1V2
M28	VDD_1V2
M29	VDD_2V5
M30	DRAM0_DQ24
M31	DRAM0_DQS3
M32	DRAM0_DQ25
M33	DRAM0_DQ26
M34	DRAM0_DQ27
N1	VO0_P23
N2	VO0_P22
N3	VO0_P21
N4	VO0_P20
N5	VO0_P19
N6	VDD_3V3
N7	VSS
N28	VSS
N29	VSS
N30	DRAM0_DQ23
N31	DRAM0_DQ22
N32	DRAM0_DQS2
N33	DRAM0_DM2
N34	DRAM0_DM3
P1	VO0_P18
P2	VO0_P17
P3	VO0_P16
P4	VO0_P15
P5	VO0_P14
P6	VDD_3V3
P7	VSS
P14	VSS
P15	VSS

Ball Id	Pin Name
P16	VSS
P17	VSS
P18	VSS
P19	VSS
P20	VSS
P21	VSS
P28	VSS
P29	VDD_2V5
P30	DRAM0_DQ18
P31	DRAM0_DQ17
P32	DRAM0_DQ19
P33	DRAM0_DQ20
P34	DRAM0_DQ21
R1	VO0_P13
R2	VO0_P12
R3	VO0_P11
R4	VO0_P10
R5	VO0_P9
R6	VSS
R7	VDD_1V2
R14	VSS
R15	VSS
R16	VSS
R17	VSS
R18	VSS
R19	VSS
R20	VSS
R21	VSS
R28	VDD_1V2
R29	VSS
R30	DRAM0_CKE

Ball Id	Pin Name
R31	DRAM0_A12
R32	DRAM0_WE#
R33	DRAM0_A13
R34	DRAM0_DQ16
T1	VO0_P8
T2	VO0_P7
T3	VO0_P6
T4	VO0_P5
T5	HDMI_PD#
T6	VSS
T7	VDD_1V2
T14	VSS
T15	VSS
T16	VSS
T17	VSS
T18	VSS
T19	VSS
T20	VSS
T21	VSS
T28	VDD_1V2
T29	VDD_2V5
T30	DRAM0_A11
T31	DRAM0_A9
T32	DRAM0_CS#
T33	DRAM0_RAS#
T34	DRAM0_CAS#
U1	VO0_CLK
U2	VO0_VLD
U3	VO0_P4
U4	VO0_P3
U5	HDMI_DSCL

Ball Id	Pin Name
U6	VDD_3V3
U7	VSS
U14	VSS
U15	VSS
U16	VSS
U17	VSS
U18	VSS
U19	VSS
U20	VSS
U21	VSS
U28	VSS
U29	VDD_2V5
U30	DRAM0_A8
U31	DRAM0_A7
U32	DRAM0_A10
U33	DRAM0_BA1
U34	DRAM0_BA0
V1	VO0_HS
V2	VO0_VS
V3	VO0_P2
V4	VO0_P1
V5	HDMI_DSDA
V6	VDD_3V3
V7	VSS
V14	VSS
V15	VSS
V16	VSS
V17	VSS
V18	VSS
V19	VSS
V20	VSS

Ball Id	Pin Name
V21	VSS
V28	VSS
V29	VSS
V30	DRAM0_A6
V31	DRAM0_A5
V32	DRAM0_A2
V33	DRAM0_A1
V34	DRAM0_A0
W1	ETH_TXD3
W2	NC
W3	VO0_P0
W4	HDMI_HPD
W5	HDMI_MSE#
W6	VSS
W7	VDD_1V2
W14	VSS
W15	VSS
W16	VSS
W17	VSS
W18	VSS
W19	VSS
W20	VSS
W21	VSS
W28	VDD_1V2
W29	VSS
W30	DRAM0_A4
W31	DRAM0_A3
W32	NC
W33	NC
W34	NC
Y1	ETH_TXD2

Ball Id	Pin Name
Y2	ETH_TXD1
Y3	ETH_TXD0
Y4	ETH_MDIO
Y5	NC
Y6	VSS
Y7	VDD_1V2
Y14	VSS
Y15	VSS
Y16	VSS
Y17	VSS
Y18	VSS
Y19	VSS
Y20	VSS
Y21	VSS
Y28	VDD_1V2
Y29	VSS
Y30	NC
Y31	NC
Y32	NC
Y33	NC
Y34	NC
AA1	ETH_TXCLK
AA2	ETH_TX_EN
AA3	ETH_CRS
AA4	NC
AA5	ETH_MDINT#
AA6	VDD_3V3
AA7	VSS
AA14	VSS
AA15	VSS
AA16	VSS

Ball Id	Pin Name
AA17	VSS
AA18	VSS
AA19	VSS
AA20	VSS
AA21	VSS
AA28	VSS
AA29	VDD_3V3
AA30	NC
AA31	NC
AA32	NC
AA33	NC
AA34	NC
AB1	ETH_RXD3
AB2	ETH_RXD2
AB3	ETH_RXD1
AB4	GPIO31
AB5	GPIO30
AB6	VDD_3V3
AB7	VSS
AB28	VSS
AB29	VDD_3V3
AB30	TDMX_GPIO0
AB31	TDMX_GPIO1
AB32	TS1_IN_SYNC
AB33	TS1_IN_D1
AB34	NC
AC1	ETH_RXD0
AC2	ETH_RXCLK
AC3	ETH_COL
AC4	GPIO29
AC5	GPIO28

Ball Id	Pin Name
AC6	VSS
AC7	VDD_1V2
AC28	VDD_1V2
AC29	VSS
AC30	TS1_IN_VLD
AC31	TS1_IN_D6
AC32	TS1_IN_D4
AC33	TS1_IN_D2
AC34	TS1_CLK
AD1	ETH_MDC
AD2	GPIO27
AD3	GPIO26
AD4	GPIO25
AD5	GPIO24
AD6	VSS
AD7	VDD_1V2
AD28	VDD_1V2
AD29	VSS
AD30	TS0_IN_VLD
AD31	TS1_IN_D7
AD32	TS1_IN_D5
AD33	TS1_IN_D3
AD34	TS1_IN_D0
AE1	ETH_RX_DV
AE2	GPIO23
AE3	GPIO22
AE4	GPIO21
AE5	GPIO20
AE6	VDD_3V3
AE7	VSS
AE28	VSS

Ball Id	Pin Name
AE29	VDD_3V3
AE30	TS0_IN_D6
AE31	TS0_IN_D4
AE32	TS0_IN_D2
AE33	TS0_IN_D0
AE34	TS0_IN_SYNC
AF1	ETH_RX_ER
AF2	GPIO19
AF3	GPIO18
AF4	GPIO17
AF5	GPIO16
AF6	VDD_3V3
AF7	VSS
AF28	VSS
AF29	VDD_3V3
AF30	TS0_IN_D7
AF31	TS0_IN_D5
AF32	TS0_IN_D3
AF33	TS0_IN_D1
AF34	TS0_CLK
AG1	VO1_Y
AG2	VO1_AVDD_Y
AG3	VO1_AVSS_Y
AG4	VO1_AVDD
AG5	VO1_AVSS
AG6	VSS
AG7	VDD_1V2
AG28	VDD_1V2
AG29	VSS
AG30	PB_ALE
AG31	PB_CS3#
AG32	PB_CS2#
AG33	PB_CS1#
AG34	PB_CS0#
AH1	VO1_V
AH2	VO1_AVDD_V
AH3	VO1_AVSS_V
AH4	VO1_RSET
AH5	VO1_VREF
AH6	VSS
AH7	VSS
AH8	VDD_1V2
AH9	VSS
AH10	VSS
AH11	VDD_1V2
AH12	VDD_1V2
AH13	VSS
AH14	VSS
AH15	VDD_1V2
AH16	VDD_1V2
AH17	VSS
AH18	VSS
AH19	VDD_1V2
AH20	VDD_1V2
AH21	VSS
AH22	VSS
AH23	VDD_1V2
AH24	VDD_1V2
AH25	VSS
AH26	VSS
AH27	VDD_1V2
AH28	VSS
AH29	VDD_3V3
AH30	PB_DIR#
AH31	PB_AD11
AH32	PB_AD10
AH33	PB_AD9
AH34	PB_AD8
AJ1	VO1_U
AJ2	VO1_AVDD_U
AJ3	VO1_AVSS_U
AJ4	VO2_RSET
AJ5	VO2_VREF
AJ6	VSS
AJ7	VSS
AJ8	VSS
AJ9	VDD_3V3
AJ10	VDD_3V3
AJ11	VSS
AJ12	VSS
AJ13	VDD_3V3
AJ14	VDD_3V3
AJ15	VSS
AJ16	VSS
AJ17	VDD_3V3
AJ18	VDD_3V3
AJ19	VSS
AJ20	VSS
AJ21	VDD_3V3
AJ22	VDD_3V3
AJ23	VSS
AJ24	VSS
AJ25	VDD_3V3

Ball Id	Pin Name
AJ26	VDD_3V3
AJ27	VSS
AJ28	VDD_3V3
AJ29	VSS
AJ30	PB_A6
AJ31	PB_AD15
AJ32	PB_AD14
AJ33	PB_AD13
AJ34	PB_AD12
AK1	VO2_Y
AK2	VO2_AVDD_Y
AK3	VO2_AVSS_Y
AK4	VO2_AVDD
AK5	VO2_AVSS
AK6	SO1_LRCLK
AK7	SI1_LRCLK
AK8	SI1_SPDIF
AK9	PCI_REQ1#
AK10	PCI_REQ2#
AK11	PCI_REQ3#
AK12	PCI_IDSEL2
AK13	PCI_IDSEL3
AK14	PCI_AD18
AK15	PCI_AD16
AK16	PCI_AD15
AK17	PCI_AD4
AK18	PCI_AD6
AK19	PCI_AD2
AK20	USB20_VSSAT_0
AK21	USB20_VSSAT_0
AK22	USB20_VSSAT_1

Ball Id	Pin Name
AK23	USB20_VSSAT_1
AK24	RTC_VSS
AK25	NC
AK26	NC
AK27	NC
AK28	NC
AK29	PB_A12
AK30	PB_A7
AK31	PB_AD3
AK32	PB_AD2
AK33	PB_AD1
AK34	PB_AD0
AL1	VO2_V
AL2	VO2_AVDD_V
AL3	VO2_AVSS_V
AL4	SO0_DATA5
AL5	SO1_DATA5
AL6	SO1_DATA4
AL7	SI0_LRCLK
AL8	SI1_BCLK
AL9	PCI_REQ0#/GNT#
AL10	PCI_AD28
AL11	PCI_AD24
AL12	PCI_IDSEL1
AL13	PCI_AD20
AL14	PCI_FRAME#
AL15	PCI_TRDY#
AL16	PCI_AD13
AL17	PCI_AD11
AL18	PCI_CBE0#
AL19	PCI_AD0

Ball Id	Pin Name
AL20	USB20_VSSAT_0
AL21	USB20_VSSAC
AL22	USB20_VDDAC_3V3
AL23	USB20_VSSAT_1
AL24	RTC_VDD_BAT_3V3
AL25	RTC_VDD_BAT_3V3
AL26	NC
AL27	PB_A21
AL28	PB_A17
AL29	PB_A13
AL30	PB_A8
AL31	PB_AD7
AL32	PB_AD6
AL33	PB_AD5
AL34	PB_AD4
AM1	VO2_U
AM2	VO2_AVDD_U
AM3	VO2_AVSS_U
AM4	SO0_DATA4
AM5	SO1_SPDIF
AM6	SO1_DATA2
AM7	SI0_SPDIF
AM8	SI1_DATA
AM9	PCI_GNT3#
AM10	PCI_GNT2#
AM11	PCI_AD30
AM12	PCI_AD26
AM13	PCI_IDSEL0
AM14	PCI_AD22
AM15	PCI_IRDY#
AM16	PCI_PAR

Ball Id	Pin Name
AM17	PCI_AD12
AM18	PCI_AD9
AM19	PCI_AD3
AM20	USB20_VDDAT_0_3V3
AM21	USB20_VDDAT_0_3V3
AM22	USB20_VDDAT_1_3V3
AM23	USB20_VDDAT_1_3V3
AM24	RTC_RING
AM25	RTC_TEST
AM26	NC
AM27	PB_A22
AM28	PB_A18
AM29	PB_A14
AM30	PB_A9
AM31	PB_A3
AM32	PB_A0
AM33	PB_DMAACK#
AM34	PB_RD#
AN1	SO0_SPDIF
AN2	SO0_LRCLK
AN3	SO0_DATA2
AN4	SO0_DATA3
AN5	SO1_BCLK
AN6	SO1_DATA3
AN7	SI0_BCLK
AN8	PCI_INTA#
AN9	PCI_GNT1#
AN10	PCI_AD29
AN11	PCI_AD25

Ball Id	Pin Name
AN12	PCI_AD23
AN13	PCI_AD21
AN14	PCI_CBE2#
AN15	PCI_STOP#
AN16	PCI_AD14
AN17	PCI_AD10
AN18	PCI_AD7
AN19	PCI_AD1
AN20	USB20_REXT
AN21	USB20_DP_0
AN22	USB20_DP_1
AN23	USB20_ATEST
AN24	RTC_CLK_IN
AN25	RTC_XTAL_DISC
AN26	NC
AN27	PB_A23
AN28	PB_A19
AN29	PB_A15
AN30	PB_A10
AN31	PB_A4
AN32	PB_A1
AN33	PB_DMARQ
AN34	PB_WR#
AP1	NC
AP2	SO0_MCLK
AP3	SO0_BCLK
AP4	SO0_DATA1
AP5	SO1_MCLK
AP6	SO1_DATA1
AP7	SI0_DATA
AP8	PCI_CLK

Ball Id	Pin Name
AP9	PCI_GNT0#/REQ#
AP10	PCI_AD31
AP11	PCI_AD27
AP12	PCI_CBE3#
AP13	PCI_AD19
AP14	PCI_AD17
AP15	PCI_DEVSEL#
AP16	PCI_CBE1#
AP17	PCI_AD8
AP18	PCI_AD5
AP19	USB20_XO
AP20	USB20_XI
AP21	USB20_DM_0
AP22	USB20_DM_1
AP23	RTC_CLK_OUT
AP24	RTC_XTAL_IN
AP25	RTC_XTAL_OUT
AP26	NC
AP27	PB_A24
AP28	PB_A20
AP29	PB_A16
AP30	PB_A11
AP31	PB_A5
AP32	PB_A2
AP33	PB_IORDY
AP34	NC

11.3 Pin Listing by Pin Name

Table 11-2. **SMP8634 pin listing by pin name**

Pin Name	Ball Id
DRAM0_A0	V34
DRAM0_A1	V33
DRAM0_A10	U32
DRAM0_A11	T30
DRAM0_A12	R31
DRAM0_A13	R33
DRAM0_A2	V32
DRAM0_A3	W31
DRAM0_A4	W30
DRAM0_A5	V31
DRAM0_A6	V30
DRAM0_A7	U31
DRAM0_A8	U30
DRAM0_A9	T31
DRAM0_BA0	U34
DRAM0_BA1	U33
DRAM0_CAS#	T34
DRAM0_CK	K34
DRAM0_CK#	L34
DRAM0_CKE	R30
DRAM0_CS#	T32
DRAM0_DM0	H33
DRAM0_DM1	H32
DRAM0_DM2	N33
DRAM0_DM3	N34
DRAM0_DQ0	K31
DRAM0_DQ1	K30

Pin Name	Ball Id
DRAM0_DQ10	G33
DRAM0_DQ11	G32
DRAM0_DQ12	G30
DRAM0_DQ13	G31
DRAM0_DQ14	F34
DRAM0_DQ15	F33
DRAM0_DQ16	R34
DRAM0_DQ17	P31
DRAM0_DQ18	P30
DRAM0_DQ19	P32
DRAM0_DQ2	K33
DRAM0_DQ20	P33
DRAM0_DQ21	P34
DRAM0_DQ22	N31
DRAM0_DQ23	N30
DRAM0_DQ24	M30
DRAM0_DQ25	M32
DRAM0_DQ26	M33
DRAM0_DQ27	M34
DRAM0_DQ28	L31
DRAM0_DQ29	L30
DRAM0_DQ3	K32
DRAM0_DQ30	L32
DRAM0_DQ31	L33
DRAM0_DQ4	J34
DRAM0_DQ5	J33
DRAM0_DQ6	J32
DRAM0_DQ7	J31

Pin Name	Ball Id
DRAM0_DQ8	H31
DRAM0_DQ9	G34
DRAM0_DQS0	H34
DRAM0_DQS1	H30
DRAM0_DQS2	N32
DRAM0_DQS3	M31
DRAM0_RAS#	T33
DRAM0_VREFSSTL0	L29
DRAM0_VREFSSTL1	J30
DRAM0_VREFSSTL2	G29
DRAM0_WE#	R32
DRAM1_A0	B34
DRAM1_A1	C34
DRAM1_A10	C33
DRAM1_A11	E31
DRAM1_A12	F31
DRAM1_A13	B31
DRAM1_A2	D34
DRAM1_A3	E34
DRAM1_A4	D33
DRAM1_A5	E33
DRAM1_A6	D32
DRAM1_A7	E32
DRAM1_A8	F32
DRAM1_A9	D31
DRAM1_BA0	A33
DRAM1_BA1	B33
DRAM1_CAS#	A32

Pin Name	Ball Id
DRAM1_CK	A26
DRAM1_CK#	A27
DRAM1_CKE	F30
DRAM1_CS#	C32
DRAM1_DM0	B24
DRAM1_DM1	C24
DRAM1_DM2	B29
DRAM1_DM3	A29
DRAM1_DQ0	D26
DRAM1_DQ1	E26
DRAM1_DQ10	B23
DRAM1_DQ11	C23
DRAM1_DQ12	D23
DRAM1_DQ13	A22
DRAM1_DQ14	B22
DRAM1_DQ15	C22
DRAM1_DQ16	A31
DRAM1_DQ17	D30
DRAM1_DQ18	E30
DRAM1_DQ19	C30
DRAM1_DQ2	B26
DRAM1_DQ20	B30
DRAM1_DQ21	A30
DRAM1_DQ22	D29
DRAM1_DQ23	E29
DRAM1_DQ24	E28
DRAM1_DQ25	C28
DRAM1_DQ26	B28
DRAM1_DQ27	A28
DRAM1_DQ28	D27
DRAM1_DQ29	E27

Pin Name	Ball Id
DRAM1_DQ3	C26
DRAM1_DQ30	C27
DRAM1_DQ31	B27
DRAM1_DQ4	A25
DRAM1_DQ5	B25
DRAM1_DQ6	C25
DRAM1_DQ7	D25
DRAM1_DQ8	D24
DRAM1_DQ9	A23
DRAM1_DQS0	A24
DRAM1_DQS1	E24
DRAM1_DQS2	C29
DRAM1_DQS3	D28
DRAM1_RAS#	B32
DRAM1_VREFSSTL0	F28
DRAM1_VREFSSTL1	E25
DRAM1_VREFSSTL2	E23
DRAM1_WE#	C31
ETH_COL	AC3
ETH_CRS	AA3
ETH_MDC	AD1
ETH_MDINT#	AA5
ETH_MDIO	Y4
ETH_RX_DV	AE1
ETH_RX_ER	AF1
ETH_RXCLK	AC2
ETH_RXD0	AC1
ETH_RXD1	AB3
ETH_RXD2	AB2
ETH_RXD3	AB1
ETH_TX_EN	AA2

Pin Name	Ball Id
ETH_TXCLK	AA1
ETH_TXD0	Y3
ETH_TXD1	Y2
ETH_TXD2	Y1
ETH_TXD3	W1
GPIO0	A5
GPIO1	B5
GPIO10	E6
GPIO11	E8
GPIO12	B7
GPIO13	C7
GPIO14	D7
GPIO15	E7
GPIO16	AF5
GPIO17	AF4
GPIO18	AF3
GPIO19	AF2
GPIO2	C5
GPIO20	AE5
GPIO21	AE4
GPIO22	AE3
GPIO23	AE2
GPIO24	AD5
GPIO25	AD4
GPIO26	AD3
GPIO27	AD2
GPIO28	AC5
GPIO29	AC4
GPIO3	D5
GPIO30	AB5
GPIO31	AB4

Pin Name	Ball Id
GPIO4	D4
GPIO5	E4
GPIO6	E5
GPIO7	B6
GPIO8	C6
GPIO9	D6
HDMI_DSCL	U5
HDMI_DSDA	V5
HDMI_HPD	W4
HDMI_MSE#	W5
HDMI_PD#	T5
IDE_A0	B21
IDE_A1	E21
IDE_A2	C21
IDE_CS0#	D22
IDE_CS1#	E22
IDE_D0	C19
IDE_D1	A19
IDE_D10	B17
IDE_D11	D18
IDE_D12	A18
IDE_D13	C18
IDE_D14	E19
IDE_D15	B19
IDE_D2	D19
IDE_D3	B18
IDE_D4	E18
IDE_D5	C17
IDE_D6	A17
IDE_D7	D17
IDE_D8	C16

Pin Name	Ball Id
IDE_D9	E17
IDE_ACK#	C20
IDE_DMARQ	D20
IDE_IRQ	D21
IDE_IOR#	A20
IDE_IORDY	B20
IDE_IOW#	E20
IDE_NPCBLID	A21
JTAG_UART#	D11
NC	A1
NC	A34
NC	AA30
NC	AA31
NC	AA32
NC	AA33
NC	AA34
NC	AA4
NC	AB34
NC	AK25
NC	AK26
NC	AK27
NC	AK28
NC	AL26
NC	AM26
NC	AN26
NC	AP1
NC	AP26
NC	AP34
NC	W2
NC	W32
NC	W33

Pin Name	Ball Id
NC	W34
NC	Y30
NC	Y31
NC	Y32
NC	Y33
NC	Y34
NC	Y5
PB_A0	AM32
PB_A1	AN32
PB_A10	AN30
PB_A11	AP30
PB_A12	AK29
PB_A13	AL29
PB_A14	AM29
PB_A15	AN29
PB_A16	AP29
PB_A17	AL28
PB_A18	AM28
PB_A19	AN28
PB_A2	AP32
PB_A20	AP28
PB_A21	AL27
PB_A22	AM27
PB_A23	AN27
PB_A24	AP27
PB_A3	AM31
PB_A4	AN31
PB_A5	AP31
PB_A6	AJ30
PB_A7	AK30
PB_A8	AL30

Pin Name	Ball Id
PB_A9	AM30
PB_AD0	AK34
PB_AD1	AK33
PB_AD10	AH32
PB_AD11	AH31
PB_AD12	AJ34
PB_AD13	AJ33
PB_AD14	AJ32
PB_AD15	AJ31
PB_AD2	AK32
PB_AD3	AK31
PB_AD4	AL34
PB_AD5	AL33
PB_AD6	AL32
PB_AD7	AL31
PB_AD8	AH34
PB_AD9	AH33
PB_ALE	AG30
PB_CS0#	AG34
PB_CS1#	AG33
PB_CS2#	AG32
PB_CS3#	AG31
PB_DIR#	AH30
PB_DMAACK#	AM33
PB_DMARQ	AN33
PB_IORDY	AP33
PB_RD#	AM34
PB_WR#	AN34
PCI_AD0	AL19
PCI_AD1	AN19
PCI_AD10	AN17

Pin Name	Ball Id
PCI_AD11	AL17
PCI_AD12	AM17
PCI_AD13	AL16
PCI_AD14	AN16
PCI_AD15	AK16
PCI_AD16	AK15
PCI_AD17	AP14
PCI_AD18	AK14
PCI_AD19	AP13
PCI_AD2	AK19
PCI_AD20	AL13
PCI_AD21	AN13
PCI_AD22	AM14
PCI_AD23	AN12
PCI_AD24	AL11
PCI_AD25	AN11
PCI_AD26	AM12
PCI_AD27	AP11
PCI_AD28	AL10
PCI_AD29	AN10
PCI_AD3	AM19
PCI_AD30	AM11
PCI_AD31	AP10
PCI_AD4	AK17
PCI_AD5	AP18
PCI_AD6	AK18
PCI_AD7	AN18
PCI_AD8	AP17
PCI_AD9	AM18
PCI_CBE0#	AL18
PCI_CBE1#	AP16

Pin Name	Ball Id
PCI_CBE2#	AN14
PCI_CBE3#	AP12
PCI_CLK	AP8
PCI_DEVSEL#	AP15
PCI_FRAME#	AL14
PCI_GNT0#/REQ#	AP9
PCI_GNT1#	AN9
PCI_GNT2#	AM10
PCI_GNT3#	AM9
PCI_IDSEL0	AM13
PCI_IDSEL1	AL12
PCI_IDSEL2	AK12
PCI_IDSEL3	AK13
PCI_INTA#	AN8
PCI_IRDY#	AM15
PCI_PAR	AM16
PCI_REQ0#/GNT#	AL9
PCI_REQ1#	AK9
PCI_REQ2#	AK10
PCI_REQ3#	AK11
PCI_STOP#	AN15
PCI_TRDY#	AL15
RCLK0_IN	B8
RCLK0_OUT	D8
RCLK1_OUT	D9
RCLK1_XTAL_IN	A7
RCLK1_XTAL_OUT	A6
RCLK2_OUT	D10
RCLK3_OUT	E9
RESET#	A8
RTC_CLK_IN	AN24

Pin Name	Ball Id
RTC_CLK_OUT	AP23
RTC_RING	AM24
RTC_TEST	AM25
RTC_VDD_BAT_3V3	AL24
RTC_VDD_BAT_3V3	AL25
RTC_VSS	AK24
RTC_XTAL_DISC	AN25
RTC_XTAL_IN	AP24
RTC_XTAL_OUT	AP25
SCARD_CLK	A16
SCARD_CTL0	C15
SCARD_CTL1	D16
SCARD_CTL2	E16
SCARD_FC#	B16
SCARD_IO	A15
SCARD_RST	B15
SI0_BCLK	AN7
SI0_DATA	AP7
SI0_LRCLK	AL7
SI0_SPDIF	AM7
SI1_BCLK	AL8
SI1_DATA	AM8
SI1_LRCLK	AK7
SI1_SPDIF	AK8
SO0_BCLK	AP3
SO0_DATA1	AP4
SO0_DATA2	AN3
SO0_DATA3	AN4
SO0_DATA4	AM4
SO0_DATA5	AL4
SO0_LRCLK	AN2

Pin Name	Ball Id
SO0_MCLK	AP2
SO0_SPDIF	AN1
SO1_BCLK	AN5
SO1_DATA1	AP6
SO1_DATA2	AM6
SO1_DATA3	AN6
SO1_DATA4	AL6
SO1_DATA5	AL5
SO1_LRCLK	AK6
SO1_MCLK	AP5
SO1_SPDIF	AM5
TDMX_GPIO0	AB30
TDMX_GPIO1	AB31
TEST	E10
TS0_CLK	AF34
TS0_IN_D0	AE33
TS0_IN_D1	AF33
TS0_IN_D2	AE32
TS0_IN_D3	AF32
TS0_IN_D4	AE31
TS0_IN_D5	AF31
TS0_IN_D6	AE30
TS0_IN_D7	AF30
TS0_IN_SYNC	AE34
TS0_IN_VLD	AD30
TS1_CLK	AC34
TS1_IN_D0	AD34
TS1_IN_D1	AB33
TS1_IN_D2	AC33
TS1_IN_D3	AD33
TS1_IN_D4	AC32

Pin Name	Ball Id
TS1_IN_D5	AD32
TS1_IN_D6	AC31
TS1_IN_D7	AD31
TS1_IN_SYNC	AB32
TS1_IN_VLD	AC30
UART0_CTS	B14
UART0_DCD	B12
UART0_DSR	C14
UART0_DTR	C13
UART0_RTS	B13
UART0_RX	A14
UART0_TX	A13
UART1_CTS	C12
UART1_DCD	C11
UART1_DSR	B11
UART1_DTR	C10
UART1_RTS	B10
UART1_RX	A12
UART1_TX	A11
USB20_ATEST	AN23
USB20_DM_0	AP21
USB20_DM_1	AP22
USB20_DP_0	AN21
USB20_DP_1	AN22
USB20_REXT	AN20
USB20_VDDAC_3V3	AL22
USB20_VDDAT_0_3V3	AM20
USB20_VDDAT_0_3V3	AM21
USB20_VDDAT_1_3V3	AM22

Pin Name	Ball Id
USB20_VDDAT_1_3V3	AM23
USB20_VSSAC	AL21
USB20_VSSAT_0	AK20
USB20_VSSAT_0	AK21
USB20_VSSAT_0	AL20
USB20_VSSAT_1	AK22
USB20_VSSAT_1	AK23
USB20_VSSAT_1	AL23
USB20_XI	AP20
USB20_XO	AP19
VCXO0_IN	C9
VCXO1_IN	C8
VDD_1V2	AC28
VDD_1V2	AC7
VDD_1V2	AD28
VDD_1V2	AD7
VDD_1V2	AG28
VDD_1V2	AG7
VDD_1V2	AH11
VDD_1V2	AH12
VDD_1V2	AH15
VDD_1V2	AH16
VDD_1V2	AH19
VDD_1V2	AH20
VDD_1V2	AH23
VDD_1V2	AH24
VDD_1V2	AH27
VDD_1V2	AH8
VDD_1V2	G11
VDD_1V2	G12
VDD_1V2	G15

Pin Name	Ball Id
VDD_1V2	G16
VDD_1V2	G19
VDD_1V2	G20
VDD_1V2	G21
VDD_1V2	G23
VDD_1V2	G24
VDD_1V2	G27
VDD_1V2	G8
VDD_1V2	H28
VDD_1V2	H7
VDD_1V2	L28
VDD_1V2	L7
VDD_1V2	M28
VDD_1V2	M7
VDD_1V2	R28
VDD_1V2	R7
VDD_1V2	T28
VDD_1V2	T7
VDD_1V2	W28
VDD_1V2	W7
VDD_1V2	Y28
VDD_1V2	Y7
VDD_2V5	F22
VDD_2V5	F23
VDD_2V5	F25
VDD_2V5	F27
VDD_2V5	F29
VDD_2V5	H29
VDD_2V5	K29
VDD_2V5	M29
VDD_2V5	P29

Pin Name	Ball Id
VDD_2V5	T29
VDD_2V5	U29
VDD_3V3	AA29
VDD_3V3	AA6
VDD_3V3	AB29
VDD_3V3	AB6
VDD_3V3	AE29
VDD_3V3	AE6
VDD_3V3	AF29
VDD_3V3	AF6
VDD_3V3	AH29
VDD_3V3	AJ10
VDD_3V3	AJ13
VDD_3V3	AJ14
VDD_3V3	AJ17
VDD_3V3	AJ18
VDD_3V3	AJ21
VDD_3V3	AJ22
VDD_3V3	AJ25
VDD_3V3	AJ26
VDD_3V3	AJ28
VDD_3V3	AJ9
VDD_3V3	F10
VDD_3V3	F13
VDD_3V3	F14
VDD_3V3	F17
VDD_3V3	F18
VDD_3V3	F7
VDD_3V3	F9
VDD_3V3	G6
VDD_3V3	J6

Pin Name	Ball Id
VDD_3V3	K6
VDD_3V3	N6
VDD_3V3	P6
VDD_3V3	U6
VDD_3V3	V6
VDD_PLL0_1V2	D12
VDD_PLL1_3V3	D13
VDD_PLL2_3V3	D14
VDD_PLL3_3V3	D15
VI0_CLK	K1
VI0_HS	J2
VI0_P0	D2
VI0_P1	D1
VI0_P10	G5
VI0_P11	G4
VI0_P12	G3
VI0_P13	G2
VI0_P14	G1
VI0_P15	H5
VI0_P16	H4
VI0_P17	H3
VI0_P18	H2
VI0_P19	H1
VI0_P2	E3
VI0_P20	J5
VI0_P21	J4
VI0_P22	J3
VI0_P23	K5
VI0_P24	K4
VI0_P25	K3
VI0_P26	L5

Pin Name	Ball Id
VI0_P27	L4
VI0_P28	L3
VI0_P29	M5
VI0_P3	E2
VI0_P30	M4
VI0_P31	M3
VI0_P4	E1
VI0_P5	F5
VI0_P6	F4
VI0_P7	F3
VI0_P8	F2
VI0_P9	F1
VI0_VLD	K2
VI0_VS	J1
VI1_CLK	A3
VI1_HS	B3
VI1_P0	A2
VI1_P1	B2
VI1_P2	C3
VI1_P3	C4
VI1_P4	B1
VI1_P5	C1
VI1_P6	C2
VI1_P7	D3
VI1_VLD	B4
VI1_VS	A4
VI2_CLK	M1
VI2_HS	L2
VI2_VLD	M2
VI2_VS	L1
VO0_CLK	U1

Pin Name	Ball Id
VO0_HS	V1
VO0_P0	W3
VO0_P1	V4
VO0_P10	R4
VO0_P11	R3
VO0_P12	R2
VO0_P13	R1
VO0_P14	P5
VO0_P15	P4
VO0_P16	P3
VO0_P17	P2
VO0_P18	P1
VO0_P19	N5
VO0_P2	V3
VO0_P20	N4
VO0_P21	N3
VO0_P22	N2
VO0_P23	N1
VO0_P3	U4
VO0_P4	U3
VO0_P5	T4
VO0_P6	T3
VO0_P7	T2
VO0_P8	T1
VO0_P9	R5
VO0_VLD	U2
VO0_VS	V2
VO1_AVDD	AG4
VO1_AVDD_U	AJ2
VO1_AVDD_V	AH2
VO1_AVDD_Y	AG2

Pin Name	Ball Id
VO1_AVSS	AG5
VO1_AVSS_U	AJ3
VO1_AVSS_V	AH3
VO1_AVSS_Y	AG3
VO1_RSET	AH4
VO1_U	AJ1
VO1_V	AH1
VO1_VREF	AH5
VO1_Y	AG1
VO2_AVDD	AK4
VO2_AVDD_U	AM2
VO2_AVDD_V	AL2
VO2_AVDD_Y	AK2
VO2_AVSS	AK5
VO2_AVSS_U	AM3
VO2_AVSS_V	AL3
VO2_AVSS_Y	AK3
VO2_RSET	AJ4
VO2_U	AM1
VO2_V	AL1
VO2_VREF	AJ5
VO2_Y	AK1
VSS	AA14
VSS	AA15
VSS	AA16
VSS	AA17
VSS	AA18
VSS	AA19
VSS	AA20
VSS	AA21
VSS	AA28

Pin Name	Ball Id
VSS	AA7
VSS	AB28
VSS	AB7
VSS	AC29
VSS	AC6
VSS	AD29
VSS	AD6
VSS	AE28
VSS	AE7
VSS	AF28
VSS	AF7
VSS	AG29
VSS	AG6
VSS	AH10
VSS	AH13
VSS	AH14
VSS	AH17
VSS	AH18
VSS	AH21
VSS	AH22
VSS	AH25
VSS	AH26
VSS	AH28
VSS	AH6
VSS	AH7
VSS	AH9
VSS	AJ11
VSS	AJ12
VSS	AJ15
VSS	AJ16
VSS	AJ19

Pin Name	Ball Id
VSS	AJ20
VSS	AJ23
VSS	AJ24
VSS	AJ27
VSS	AJ29
VSS	AJ6
VSS	AJ7
VSS	AJ8
VSS	F11
VSS	F12
VSS	F15
VSS	F16
VSS	F19
VSS	F20
VSS	F21
VSS	F24
VSS	F26
VSS	F6
VSS	F8
VSS	G10
VSS	G13
VSS	G14
VSS	G17
VSS	G18
VSS	G22
VSS	G25
VSS	G26
VSS	G28
VSS	G7
VSS	G9
VSS	H6

Pin Name	Ball Id
VSS	J28
VSS	J29
VSS	J7
VSS	K28
VSS	K7
VSS	L6
VSS	M6
VSS	N28
VSS	N29
VSS	N7
VSS	P14
VSS	P15
VSS	P16
VSS	P17
VSS	P18
VSS	P19
VSS	P20
VSS	P21
VSS	P28
VSS	P7
VSS	R14
VSS	R15
VSS	R16
VSS	R17
VSS	R18
VSS	R19
VSS	R20
VSS	R21
VSS	R29
VSS	R6
VSS	T14

Pin Name	Ball Id
VSS	T15
VSS	T16
VSS	T17
VSS	T18
VSS	T19
VSS	T20
VSS	T21
VSS	T6
VSS	U14
VSS	U15
VSS	U16
VSS	U17
VSS	U18
VSS	U19
VSS	U20
VSS	U21
VSS	U28
VSS	U7
VSS	V14
VSS	V15
VSS	V16
VSS	V17
VSS	V18
VSS	V19
VSS	V20
VSS	V21
VSS	V28
VSS	V29
VSS	V7
VSS	W14
VSS	W15

Pin Name	Ball Id
VSS	W16
VSS	W17
VSS	W18
VSS	W19
VSS	W20
VSS	W21
VSS	W29
VSS	W6
VSS	Y14
VSS	Y15
VSS	Y16
VSS	Y17
VSS	Y18
VSS	Y19
VSS	Y20
VSS	Y21
VSS	Y29
VSS	Y6
VSS_PLL0	E12
VSS_PLL1	E13
VSS_PLL2	E14
VSS_PLL3	E15
XTAL_BUF	B9
XTAL_DISC	E11
XTAL_IN	A10
XTAL_OUT	A9

11.4 Shared Pins

11.4.1 UART Shared Pins

Table 11-1 **Shared pin descriptions**

Pin Name	Ball ID	Direction	Description	Alternate Function	Alternate Function
UART1_RX	A12	B	UART 1 receive data input	EJ_TDI. Test data input (TDI) for the EJTAG TAP.	GPIO0
UART1_CTS	C12	B	UART 1 clear to send. Flow control signal.	EJ_TMS. Test mode select input (TMS) for the EJTAG TAP.	GPIO1
UART1_DSR	B11	B	UART 1 data set ready. Data set status signal.	EJ_TRST_N. Active-low Test Reset Input (TRST) for the EJTAG TAP. At power-up, the assertion of EJ_TRST_N causes the TAP controller to be reset.	GPIO2
UART1_DCD	C11	B	UART 1 data carrier detect. Data set status signal.	EJ_TCK. Test clock input (TCK) for the EJTAG TAP.	GPIO3
UART1_TX	A11	B	UART 1 transmit data output		GPIO4
UART1_RTS	B10	B	UART 1 request to send. Flow control signal.	EJ_TDO. Test data output (TDO) for the EJTAG TAP.	GPIO5
UART1_DTR	C10	B	UART 1 data terminal ready. Data terminal status signal.		GPIO6
UART0_RX	A14	B	UART 0 receive data input		GPIO0
UART0_CTS	B14	B	UART 0 clear to send. Flow control signal.		GPIO1
UART0_DSR	C14	B	UART 0 data set ready. Data set status signal.		GPIO2
UART0_DCD	B12	B	UART 0 data carrier detect. Data set status signal.		GPIO3
UART0_TX	A13	B	UART 0 transmit data output		GPIO4

Table 11-1 **Shared pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description	Alternate Function	Alternate Function
UART0_RTS	B13	B	UART 0 request to send. Flow control signal.		GPIO5
UART0_DTR	C13	B	UART 0 data terminal ready. Data terminal status signal.		GPIO6

11.4.2 GPIO Shared Pins

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:

GPIO	Alternate Function		Selectable Inputs
15	PWM Generator 0		
14	PWM Generator 1		
13			
12			Infrared Remote Input
11			(PCI) Interrupt Input D
10			(PCI) Interrupt Input C
9			(PCI) Interrupt Input B
8			(PCI) Interrupt Input A
7	Slave I^2C	sda	
6		sck	
5	Front Panel Interface	clk	
4		stb	
3		dout	
2		din	
1	Master I^2C	sda	
0		sck	

11.4.2.1 GPIO and FIP, IR and I²C Shared Pins

Table 11-2 **GPIO shared pin description**

Pin Name	Ball ID	Direction	Alternate Function
GPIO2	C5	B	FIP serial data input
GPIO3	D5	B	FIP serial data output
GPIO4	D4	B	FIP data strobe
GPIO5	E4	B	FIP serial I/O clock
GPIO12	B7	B	Infrared decoder. Default pin assigned to the IR decoder input function (may be mapped to any other GPIO pin under the software control).
GPIO0	A5	B	I2CM_SCL. I²C master interface serial clock.
GPIO1	B5	B	I2CM_SDA. I²C master interface serial data.
GPIO6	E5	B	I2CS_SCL. I²C slave interface serial clock.
GPIO7	B6	B	I2CS_SDA. I²C slave interface serial data.

11.4.2.2 GPIO and Ethernet Shared Pins

Table 11-3 **Ethernet pin descriptions**

Pin name	Ball ID	Direction	Alternate Function
GPIO0	AA1	B	ETH_TXCLK. Ethernet transmit clock.
GPIO1	AA2	B	ETH_TX_EN. Ethernet transmit enable.
GPIO2	Y3	B	ETH_TXD0. Ethernet transmit data 0.
GPIO3	Y2	B	ETH_TXD1. Ethernet transmit data 1.
GPIO4	Y1	B	ETH_TXD2. Ethernet transmit data 2.
GPIO5	W1	B	ETH_TXD3. Ethernet transmit data 3.
GPIO6	AC2	B	ETH_RXCLK. Ethernet receive clock.
GPIO7	AE1	B	ETH_RX_DV. Ethernet receive data valid.
GPIO8	AF1	B	ETH_RX_ER. Ethernet receive error.
GPIO9	AC1	B	ETH_RXD0. Ethernet receive data 0.
GPIO10	AB3	B	ETH_RXD1. Ethernet receive data 1.
GPIO11	AB2	B	ETH_RXD2. Ethernet receive data 2.
GPIO12	AB1	B	ETH_RXD3. Ethernet receive data 3.
GPIO13	AA3	B	ETH_CRS. Ethernet carrier sense.

Table 11-3 **Ethernet pin descriptions (Continued)**

Pin name	Ball ID	Direction	Alternate Function
GPIO14	AC3	B	ETH_COL. Ethernet collision detect.
GPIO15	AD1	B	ETH_MDC. Ethernet management data clock.
GPIO16	Y4	B	ETH_MDIO. Ethernet management data I/O.
GPIO17	AA5	B	ETH_MDINT#. Ethernet management interrupt.

11.5 Miscellaneous Pins

11.5.3 Miscellaneous Pins

Table 11-4 **Miscellaneous pin descriptions**

Pin Name	Ball ID	Direction	Description
NC	A1	-	No connect
NC	A34	-	No connect
NC	AA4	-	No connect
NC	AK25	-	No connect
NC	W2	-	No connect
NC	W32	-	No connect
NC	W33	-	No connect
NC	W34	-	No connect
NC	Y5	-	No connect
NC	Y30	-	No connect
NC	Y31	-	No connect
NC	Y32	-	No connect
NC	Y33	-	No connect
NC	Y34	-	No connect
NC	AA30	-	No connect
NC	AA31	-	No connect
NC	AA32	-	No connect
NC	AA33	-	No connect
NC	AA34	-	No connect
NC	AB34	-	No connect

Table 11-4 **Miscellaneous pin descriptions (Continued)**

Pin Name	Ball ID	Direction	Description
NC	AK26	-	No connect
NC	AK27	-	No connect
NC	AK28	-	No connect
NC	AL26	-	No connect
NC	AM26	-	No connect
NC	AN26	-	No connect
NC	AP1	-	No connect
NC	AP26	-	No connect
NC	AP34	-	No connect
RESET#	A8	I	Device reset input. Active low.
RTC_TEST	W3	I	Used for manufacturing purposes only
TEST	E10	I	Test mode input. Tie to VSS for normal operation.
XTAL_DISC	E11	I	Used for manufacturing purposes only

12 System Specifications

12.1 Electrical Specifications

12.1.1 Absolute Maximum Ratings

Table 12-1 **Absolute maximum ratings**

Parameter	Symbol	Unit	Minimum	Maximum
DC supply voltage	VDD_1V2	V		1.50
DC supply voltage	VDD_2V5	V		5.50
DC supply voltage	VDD_3V3	V		5.50
DC input voltage	V_{Imax}	V	-0.30	5.50
Storage temperature	T_{STG}	^0C	-40	125

12.1.2 Recommended Operating Conditions

Table 12-2 **Recommended operating conditions**

Parameter	Symbol	Unit	Minimum	Nominal	Maximum
DC supply voltage	VDD_1V2	V	1.08	1.2	1.26
DC supply voltage[1]	VDD_2V5	V	2.5	2.6	2.7
DC supply voltage	VDD_3V3	V	3	3.3	3.6
Operating ambient temperature	T_{AMB}	^0C	0		70

1. The voltages listed represent values when using DDR-400 DRAMs. When using DDR-333 DRAM devices the minimum, nominal and maximum voltage values are 2.3, 2.5 and 2.7 volts respectively.

12.2 Power Consumption

The table below provides consolidated test and simulation results for the typical current requirements for the various SMP8634 power supply rails under typical operating conditions. These figures are not represented as worst-case, since system profiles with higher overall activity levels could be created. Also, the measurements were obtained with nominal supply voltage on 'typical' process chips at room temperature. When designing system power supplies and performing thermal analysis, a reasonable margin should be added to these values.

Table 12-3 **SMP8634 power consumption data**

Power Input	Typical Value	Typical Value	Notes
1.2V (Logic Core)	1400	mA	1, 2
2.6V (SSTL2 DDR)	170	mA	1, 3, 6
3.3V (Digital I/O)	300	mA	1, 4
3.3V (Video DACs)	80/160	mA	4, 7
3.3V (USB MAC + PHY)	65	mA	4, 8
3.0V (RTC)	20	uA	5, 9

Notes

1. The SMP8634 decoding dual HD (1080i) MPEG-2, single AC-3 audio, transport demux active, main CPU and XPU active, DDR clock frequency are = 200MHz, and the CPU and the system clock frequency are = 300MHz.
2. Nominal voltage = 1.20V.
3. Nominal voltage = 2.60V
4. Nominal voltage = 3.30V.
5. Nominal voltage = 3.00V.
6. The SMP8634 driving 4 16Mx16 DDR400 DRAMs (power consumed by DDR chips not included).
7. 105mA for one 3-DAC video output block enabled; 210mA for both 3-DAC analog video output blocks enabled. Typical average picture levels assumed with all outputs double-terminated.
8. Two transceivers configured as host and device. Result represents average power consumption with 1µs-test packets transmitted and an inter packet delay (idle time) of 8.5µs. Nominal process and voltage, temperature = 75^0C.
9. Simulated over WC process, voltage and temperature.

12.3 Power Supply Sequencing

In order to minimize excessive current flow through the device IO cells during power-up, and to eliminate the possibility of latch-up, the power supply sequencing guidelines shown in the figure below should be observed.

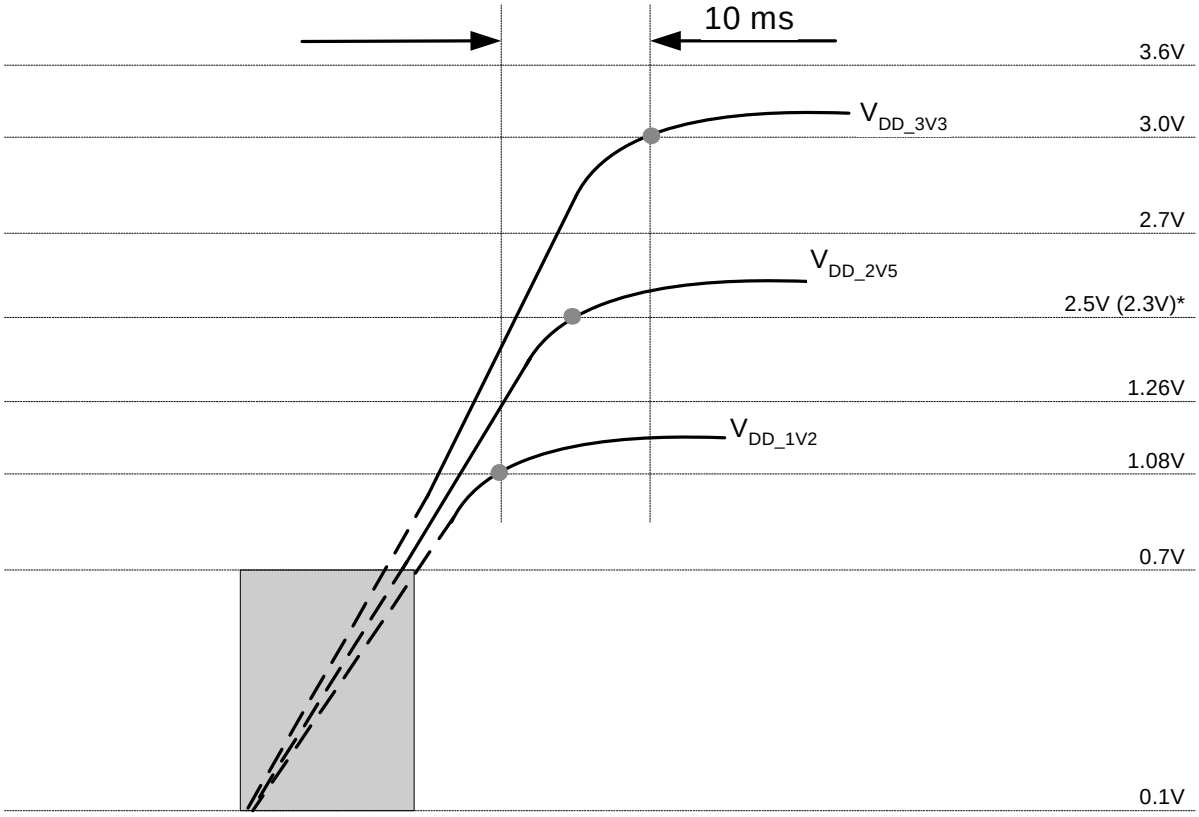


Figure 12-1 **Power supply sequencing**

1. Ideally, VDD_3V3 and VDD_2V5 should exceed VDD_1V2 at all times as the supplies are ramping up, and be within the range of 0.1V-0.7V (gray region in the above figure).
2. The rule VDD_3V3 > VDD_1V2 and VDD_2V5 > VDD_1V2 may be relaxed as long as both supplies ramp up through the 0.1V to 0.7V region in 250 microseconds or less, and VDD_1V2 at no time exceeds VDD_3V3 or VDD_2V5 by more than 0.5V.
3. VDD_3V3 and VDD_1V2 should each reach their specified operating ranges within 10 millisecond of each other.

Note: 2.5V is minimum VDD spec for DRAM devices with nominal VDD_2V5 voltage of 2.6V and 2.3V is minimum VDD spec for DRAM devices with nominal VDD_2V5 voltage of 2.5V.

12.3.3 Listing of Ground Pins (VSS)

The following table lists the SMP8634 ground pins:

Table 12-4 **Ground pins (VSS)**

Pin Name	Ball ID	Direction	Description
VSS	F6	I	Ground. Zero volt reference for 1.2, 2.5 and 3.3V supplies
VSS	F8	I	Ground
VSS	F11	I	Ground
VSS	F12	I	Ground
VSS	F15	I	Ground
VSS	F16	I	Ground
VSS	F19	I	Ground
VSS	F20	I	Ground
VSS	F21	I	Ground
VSS	F24	I	Ground
VSS	F26	I	Ground
VSS	G7	I	Ground
VSS	G9	I	Ground
VSS	G10	I	Ground
VSS	G13	I	Ground
VSS	G14	I	Ground
VSS	G17	I	Ground
VSS	G18	I	Ground
VSS	G22	I	Ground
VSS	G25	I	Ground
VSS	G26	I	Ground
VSS	G28	I	Ground
VSS	H6	I	Ground
VSS	J7	I	Ground
VSS	J28	I	Ground
VSS	J29	I	Ground
VSS	K7	I	Ground

Table 12-4 **Ground pins (VSS) (Continued)**

VSS	K28	I	Ground
VSS	L6	I	Ground
VSS	M6	I	Ground
VSS	N7	I	Ground
VSS	N28	I	Ground
VSS	N29	I	Ground
VSS	P7	I	Ground
VSS	P14	I	Ground
VSS	P15	I	Ground
VSS	P16	I	Ground
VSS	P17	I	Ground
VSS	P18	I	Ground
VSS	P19	I	Ground
VSS	P20	I	Ground
VSS	P21	I	Ground
VSS	P28	I	Ground
VSS	R6	I	Ground
VSS	R14	I	Ground
VSS	R15	I	Ground
VSS	R16	I	Ground
VSS	R17	I	Ground
VSS	R18	I	Ground
VSS	R19	I	Ground
VSS	R20	I	Ground
VSS	R21	I	Ground
VSS	R29	I	Ground
VSS	T6	I	Ground
VSS	T14	I	Ground
VSS	T15	I	Ground
VSS	T16	I	Ground
VSS	T17	I	Ground

Table 12-4 **Ground pins (VSS) (Continued)**

VSS	T18	I	Ground
VSS	T19	I	Ground
VSS	T20	I	Ground
VSS	T21	I	Ground
VSS	U7	I	Ground
VSS	U14	I	Ground
VSS	U15	I	Ground
VSS	U16	I	Ground
VSS	U17	I	Ground
VSS	U18	I	Ground
VSS	U19	I	Ground
VSS	U20	I	Ground
VSS	U21	I	Ground
VSS	U28	I	Ground
VSS	V7	I	Ground
VSS	V14	I	Ground
VSS	V15	I	Ground
VSS	V16	I	Ground
VSS	V17	I	Ground
VSS	V18	I	Ground
VSS	V19	I	Ground
VSS	V20	I	Ground
VSS	V21	I	Ground
VSS	V28	I	Ground
VSS	V29	I	Ground
VSS	W6	I	Ground
VSS	W14	I	Ground
VSS	W15	I	Ground
VSS	W16	I	Ground
VSS	W17	I	Ground
VSS	W18	I	Ground

Table 12-4 **Ground pins (VSS) (Continued)**

VSS	W19	I	Ground
VSS	W20	I	Ground
VSS	W21	I	Ground
VSS	W29	I	Ground
VSS	Y6	I	Ground
VSS	Y14	I	Ground
VSS	Y15	I	Ground
VSS	Y16	I	Ground
VSS	Y17	I	Ground
VSS	Y18	I	Ground
VSS	Y19	I	Ground
VSS	Y20	I	Ground
VSS	Y21	I	Ground
VSS	Y29	I	Ground
VSS	AA7	I	Ground
VSS	AA14	I	Ground
VSS	AA15	I	Ground
VSS	AA16	I	Ground
VSS	AA17	I	Ground
VSS	AA18	I	Ground
VSS	AA19	I	Ground
VSS	AA20	I	Ground
VSS	AA21	I	Ground
VSS	AA28	I	Ground
VSS	AB7	I	Ground
VSS	AB28	I	Ground
VSS	AC6	I	Ground
VSS	AC29	I	Ground
VSS	AD6	I	Ground
VSS	AD29	I	Ground
VSS	AE7	I	Ground

Table 12-4 **Ground pins (VSS) (Continued)**

VSS	AE28	I	Ground
VSS	AF7	I	Ground
VSS	AF28	I	Ground
VSS	AG6	I	Ground
VSS	AG29	I	Ground
VSS	AH6	I	Ground
VSS	AH7	I	Ground
VSS	AH9	I	Ground
VSS	AH10	I	Ground
VSS	AH13	I	Ground
VSS	AH14	I	Ground
VSS	AH17	I	Ground
VSS	AH18	I	Ground
VSS	AH21	I	Ground
VSS	AH22	I	Ground
VSS	AH25	I	Ground
VSS	AH26	I	Ground
VSS	AH28	I	Ground
VSS	AJ6	I	Ground
VSS	AJ7	I	Ground
VSS	AJ8	I	Ground
VSS	AJ11	I	Ground
VSS	AJ12	I	Ground
VSS	AJ15	I	Ground
VSS	AJ16	I	Ground
VSS	AJ19	I	Ground
VSS	AJ20	I	Ground
VSS	AJ23	I	Ground
VSS	AJ24	I	Ground
VSS	AJ27	I	Ground
VSS	AJ29	I	Ground

Table 12-4 **Ground pins (VSS) (Continued)**

VSS_PLL3	E12	I	Ground. Dedicated connection for PLL#3
VSS_PLL2	E13	I	Ground. Dedicated connection for PLL#2
VSS_PLL1	E14	I	Ground. Dedicated connection for PLL#1
VSS_PLL0	E15	I	Ground. Dedicated connection for PLL#0

12.3.4 Volt Power Rail

12.3.4.1 1.2V Power Rail

Table 12-5 **1.2V power rail (VDD_1V2)**

Pin Name	Ball ID	Direction	Description
VDD_1V2	G8	I	1.2V (nominal) power supply
VDD_1V2	G11	I	1.2V (nominal) power supply
VDD_1V2	G12	I	1.2V (nominal) power supply
VDD_1V2	G15	I	1.2V (nominal) power supply
VDD_1V2	G16	I	1.2V (nominal) power supply
VDD_1V2	G19	I	1.2V (nominal) power supply
VDD_1V2	G20	I	1.2V (nominal) power supply
VDD_1V2	G21	I	1.2V (nominal) power supply
VDD_1V2	G23	I	1.2V (nominal) power supply
VDD_1V2	G24	I	1.2V (nominal) power supply
VDD_1V2	G27	I	1.2V (nominal) power supply
VDD_1V2	H7	I	1.2V (nominal) power supply
VDD_1V2	H28	I	1.2V (nominal) power supply
VDD_1V2	L7	I	1.2V (nominal) power supply
VDD_1V2	L28	I	1.2V (nominal) power supply
VDD_1V2	M7	I	1.2V (nominal) power supply
VDD_1V2	M28	I	1.2V (nominal) power supply
VDD_1V2	R7	I	1.2V (nominal) power supply
VDD_1V2	R28	I	1.2V (nominal) power supply
VDD_1V2	T7	I	1.2V (nominal) power supply
VDD_1V2	T28	I	1.2V (nominal) power supply
VDD_1V2	W7	I	1.2V (nominal) power supply

Table 12-5 **1.2V power rail (VDD_1V2) (Continued)**

Pin Name	Ball ID	Direction	Description
VDD_1V2	W28	I	1.2V (nominal) power supply
VDD_1V2	Y7	I	1.2V (nominal) power supply
VDD_1V2	Y28	I	1.2V (nominal) power supply
VDD_1V2	AC7	I	1.2V (nominal) power supply
VDD_1V2	AC28	I	1.2V (nominal) power supply
VDD_1V2	AD7	I	1.2V (nominal) power supply
VDD_1V2	AD28	I	1.2V (nominal) power supply
VDD_1V2	AG7	I	1.2V (nominal) power supply
VDD_1V2	AG28	I	1.2V (nominal) power supply
VDD_1V2	AH8	I	1.2V (nominal) power supply
VDD_1V2	AH11	I	1.2V (nominal) power supply
VDD_1V2	AH12	I	1.2V (nominal) power supply
VDD_1V2	AH15	I	1.2V (nominal) power supply
VDD_1V2	AH16	I	1.2V (nominal) power supply
VDD_1V2	AH19	I	1.2V (nominal) power supply
VDD_1V2	AH20	I	1.2V (nominal) power supply
VDD_1V2	AH23	I	1.2V (nominal) power supply
VDD_1V2	AH24	I	1.2V (nominal) power supply
VDD_1V2	AH27	I	1.2V (nominal) power supply
VDD_PLL0_1V2	D12	I	1.2V (nominal) power supply. Dedicated power connection for PLL#0.

12.3.4.2 2.5V Power Rail

The VDD_2V5 powers the SSTL-2 I/O signals which form the DDR DRAM interfaces. Nominal voltage for this supply is 2.6V when using DDR400 DRAMs (system clock frequency = 200MHz).

Table 12-6 **2.5V power rail (VDD_2V5)**

Pin Name	Ball ID	Direction	Description
VDD_2V5	F22	I	2.5/2.6V (nominal) power supply
VDD_2V5	F23	I	2.5/2.6V (nominal) power supply
VDD_2V5	F25	I	2.5/2.6V (nominal) power supply
VDD_2V5	F27	I	2.5/2.6V (nominal) power supply
VDD_2V5	F29	I	2.5/2.6V (nominal) power supply
VDD_2V5	H29	I	2.5/2.6V (nominal) power supply
VDD_2V5	K29	I	2.5/2.6V (nominal) power supply
VDD_2V5	M29	I	2.5/2.6V (nominal) power supply
VDD_2V5	P29	I	2.5/2.6V (nominal) power supply
VDD_2V5	T29	I	2.5/2.6V (nominal) power supply
VDD_2V5	U29	I	2.5/2.6V (nominal) power supply

12.3.4.3 3.3V Power Rail

Table 12-7 **3.3V power rail (VDD_3V3)**

Pin Name	Ball ID	Direction	Description
VDD_3V3	F7	I	3.3V (nominal) power supply
VDD_3V3	F9	I	3.3V (nominal) power supply
VDD_3V3	F10	I	3.3V (nominal) power supply
VDD_3V3	F13	I	3.3V (nominal) power supply
VDD_3V3	F14	I	3.3V (nominal) power supply
VDD_3V3	F17	I	3.3V (nominal) power supply
VDD_3V3	F18	I	3.3V (nominal) power supply
VDD_3V3	G6	I	3.3V (nominal) power supply
VDD_3V3	J6	I	3.3V (nominal) power supply
VDD_3V3	K6	I	3.3V (nominal) power supply

Table 12-7 **3.3V power rail (VDD_3V3) (Continued)**

Pin Name	Ball ID	Direction	Description
VDD_3V3	N6	I	3.3V (nominal) power supply
VDD_3V3	P6	I	3.3V (nominal) power supply
VDD_3V3	U6	I	3.3V (nominal) power supply
VDD_3V3	V6	I	3.3V (nominal) power supply
VDD_3V3	AA6	I	3.3V (nominal) power supply
VDD_3V3	AA29	I	3.3V (nominal) power supply
VDD_3V3	AB6	I	3.3V (nominal) power supply
VDD_3V3	AB29	I	3.3V (nominal) power supply
VDD_3V3	AE6	I	3.3V (nominal) power supply
VDD_3V3	AE29	I	3.3V (nominal) power supply
VDD_3V3	AF6	I	3.3V (nominal) power supply
VDD_3V3	AF29	I	3.3V (nominal) power supply
VDD_3V3	AH29	I	3.3V (nominal) power supply
VDD_3V3	AJ9	I	3.3V (nominal) power supply
VDD_3V3	AJ10	I	3.3V (nominal) power supply
VDD_3V3	AJ13	I	3.3V (nominal) power supply
VDD_3V3	AJ14	I	3.3V (nominal) power supply
VDD_3V3	AJ17	I	3.3V (nominal) power supply
VDD_3V3	AJ18	I	3.3V (nominal) power supply
VDD_3V3	AJ21	I	3.3V (nominal) power supply
VDD_3V3	AJ22	I	3.3V (nominal) power supply
VDD_3V3	AJ25	I	3.3V (nominal) power supply
VDD_3V3	AJ26	I	3.3V (nominal) power supply
VDD_3V3	AJ28	I	3.3V (nominal) power supply
VDD_PLL1_3V3	D13	I	3.3V (nominal) power supply Dedicated power connection for PLL#1.
VDD_PLL2_3V3	D14	I	3.3V (nominal) power supply. Dedicated power connection for PLL#2.
VDD_PLL3_3V3	D15	I	3.3V (nominal) power supply. Dedicated power connection for PLL#3.

12.4 Thermal Specifications

12.4.1 Package Thermal Characteristics

Table 12-8 **Package thermal characteristics**

Parameter	Symbol	Unit	Typical	Condition
Thermal resistance, junction to ambient	Theta J-A	^{0}C/W	16.6^{0}C/W	0m/s airflow
Thermal resistance, junction to ambient	Theta J-A	^{0}C/W	14^{0}C/W	1m/s airflow
Thermal resistance, junction to ambient	Theta J-A	^{0}C/W	13.1^{0}C/W	2m/s airflow
Thermal resistance, junction to case	Theta J-C	^{0}C/W	5.1^{0}C/W	0m/s airflow

12.5 Mechanical Specifications

12.5.1 Package Specifications

		Symbol	Common Dimensions
Package :			PBGA
Body Size:	X	D	35.000
	Y	E	35.000
Ball Pitch :	X	eD	1.000
	Y	eE	1.000
Total Thickness :		A	2.280 ±0.130
Mold Thickness :		A3	1.170 Ref.
Substrate Thickness :		A2	0.610 Ref.
Ball Diameter :			0.600
Stand Off :		A1	0.400 ~ 0.600
Ball Width :		b	0.500 ~ 0.700
Mold Area :	X	M	30.000 REF.
	Y	N	30.000 REF.
Chamfer		CA	4.000 Ref.
Package Edge Tolerance :		aaa	0.200
Substrate Flatness :		bbb	0.250
Mold Flatness :		ccc	0.350
Coplanarity:		ddd	0.200
Ball Offset (Package) :		eee	0.250
Ball Offset (Ball) :		fff	0.100
Ball Count :		n	820
Edge Ball Center to Center :	X	D1	33.000
	Y	E1	33.000

12.5.2 Package Drawings

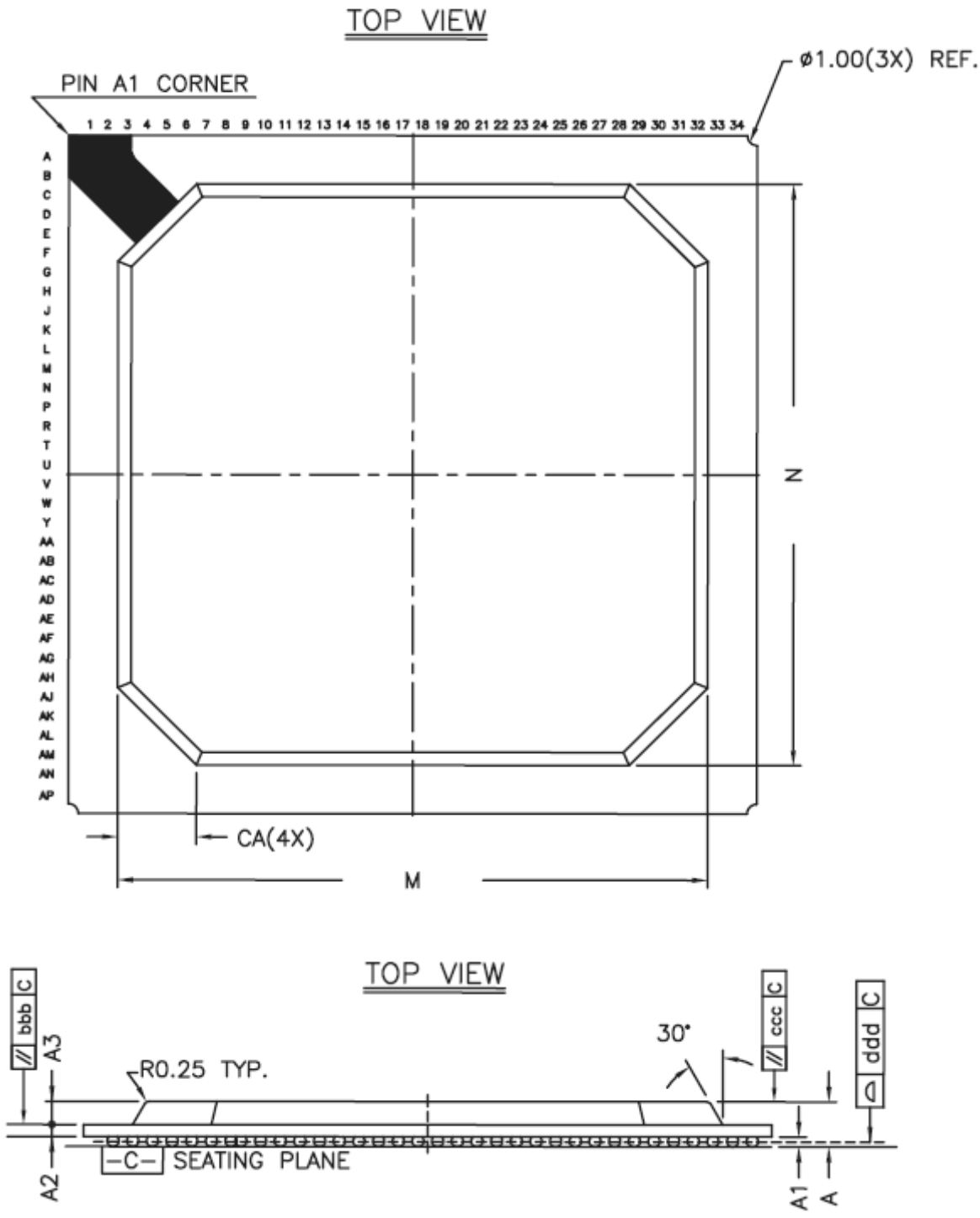


Figure 12-2 **Package diagram - top view (820 BGA)**

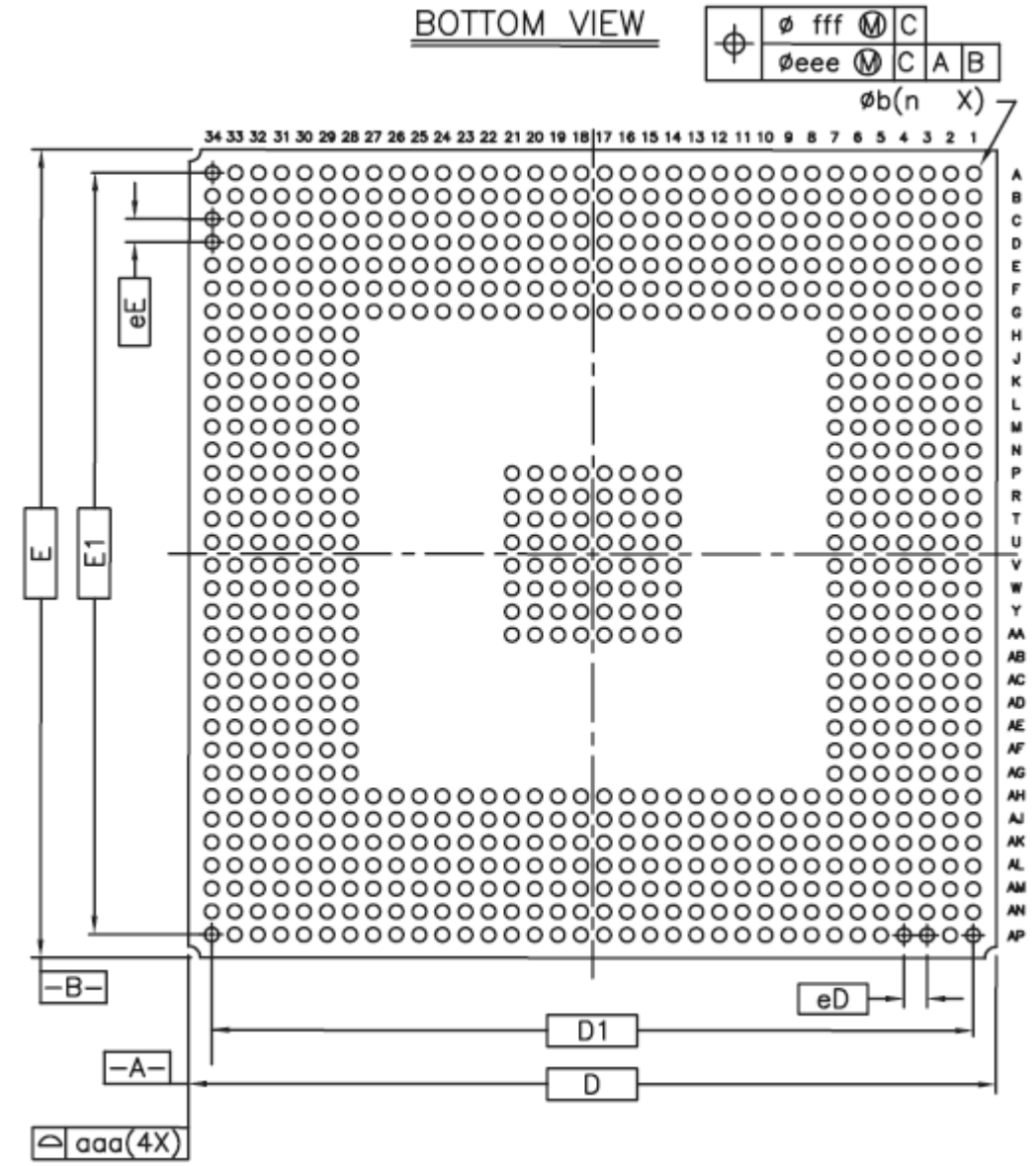


Figure 12-3 **Package diagram - bottom view (820 BGA)**

13 Appendix

13.1 SMP8634 DDR System Implementation Guidelines

The SMP8634 Secure Media Processor utilizes state-of-the art technology to integrate virtually all of the processing functions required for typical set-top box (STB) applications into a single system-on-chip semiconductor device. The SMP8634 incorporates seven independent processors and numerous hardware functional blocks, all requiring high-bandwidth low-latency access to memory areas of various size and function. In the SMP8634, these memory requirements are aggregated into a “unified” memory architecture based on commodity DRAM devices external to the chip. To provide adequate total memory bandwidth, the unified memory is implemented as two separate 32-bit DDR subsystems, each having an identical but separate controller on the chip.

Since DDR devices using a 200MHz clock (“DDR400”) are used, the memory architecture provides a theoretical peak bandwidth of:

*2 (controllers) * 4 (bytes/controller) * 400 (Mtransfers/sec) = 3.2 Gbytes/sec*

This architecture presents several challenges to reliable operation of the complete memory system.

First, DDR400 memory technology is characterized by very tight timing requirements. The timing “budgets” allow very little margin for error. As a result, every aspect of memory system design MUST be carefully planned and implemented, or the memory system may have insufficient margin and be prone to errors.

Secondly, with two independent controllers on the SOC, the patterns of overall DRAM activity become more complex. For example, read and write operations can occur simultaneously, creating possibilities for channel-to-channel coupling and similar effects which cannot occur in a single DRAM architecture.

Sigma Designs has invested considerable effort in understanding the various factors which bear on memory system integrity, and learning techniques to maximize margin. It is the goal of this document to share our findings on these topics.

Note: In this document, the SSTL-2 power supply is named VDD_2V5, even though its nominal voltage with DDR400 speed DRAMs is 2.60V.

- Whenever DDR400 DRAMs are used, the VDD_2V5 supply should have a nominal value of 2.60V, and the Vref should be 50% of VDD_2V5, or 1.3V nominal.
- When DDR333 speed DRAMs are used (FMEMCK = 166MHz), the VDD_2V5 supply should be set to 2.50V, and the Vref should be 50% or 1.25V nominal.

13.1.1 Layout Topology

The physical placement of the DDR DRAMS and decoupling components in relation to the media processor has been shown to have a significant impact on the level of performance and margin of the memory system. Sigma Designs recommends that the layout of the DDR system as embodied in the “Vantage” reference design be followed as much as possible. The general configuration shown in the following figure is recommended:

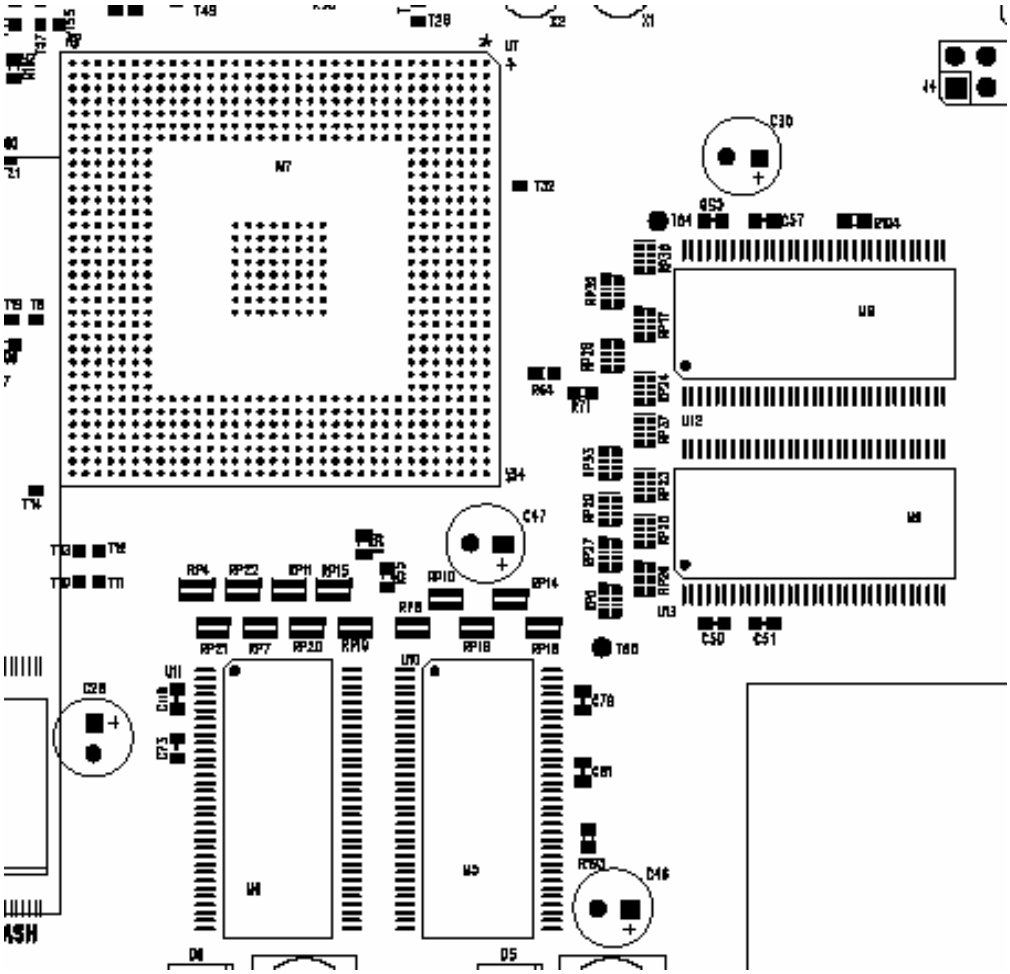


Figure 13-1 **General recommended DRAM/SOC placement**

Key features of this topology include the following:

- All data signals (DQ, DQS, DM) can be routed with a maximum trace length of 1500 mils or less.
- All address, command, and clock signals can be routed with a maximum trace length of 2000 mils or less.
- Coupling between DRAM banks 0 and 1 is minimized.

13.1.2 General Layout Rules

Each bank of memory is comprised of two memory devices. The address and command signals are routed in a daisy chain topology, while the data group signals are routed point-to-point. The clock pair routing is discussed in the next section.

Power (VDD_2V5) and ground planes should be continuous, without cuts, separations or embedded traces, throughout the area containing the DDR DRAMS, associated discretes, and the DRAM interface section of the SMP8634 ball pattern. It is preferred that the internal planes use the “flood-over-via” technique for connecting to the vias, rather than thermal relief patterns.

The memory interface signals should not be routed on any layer that is not directly adjacent to a common reference plane. To help reduce crosstalk, the data and data strobes should be isolated from the address and command signals. This is accomplished by routing different signal types on separate layers of the PCB.

For each DRAM bank there are four data signal groups. For all PCB tracks, shorter lengths are always to be preferred. Within each data signal group, the following signal length matching should be followed:

DRAMx_D[31:24] and DRAMx_DQM3 must be matched as closely as possible to DRAMx_DQS3, and must not exceed 1500 mils total length.

DRAMx_D[23:16] and DRAMx_DQM2 must be matched as closely as possible to DRAMx_DQS2, and must not exceed 1500 mils total length.

DRAMx_D[15:8] and DRAMx_DQM1 must be matched as closely as possible to DRAMx_DQS1, and must not exceed 1500 mils total length.

DRAMx_D[7:0] and DRAMx_DQM0 must be matched as closely as possible to DRAMx_DQS0, and must not exceed 1500 mils total length.

Designs should use the following PCB track width and clearance rules (all dimensions in mils = 0.001 inch):

Table 13-1 **PCB design rules**

Signal Group	General Routing		Within BGA Footprint	
	Track Width	Clearance	Track Width	Clearance
DQSx	5	10	4	4
CK/CK#	Target to 100ohm differential impedance	Target to 100ohm differential impedance	N/A	N/A
All others	5	5	4	4

13.1.3 Clock and Signal Termination

The Vantage reference design includes series termination resistors on all DRAM interface signals. All series resistors have a value of 33 ohms, although the optimal value for any given PCB design may vary somewhat. Resistor packs may be used for the address and command signals, and also for the data (DQ) signals. Individual resistors should be used for the differential clock (CK, CK#) as well as the DQS and DM signals. The Vantage design places the resistors approximately near the midpoint of the signal traces.

The differential clock signal can be routed either as two distinct point-to-point connections from BGA to DRAMs (“Y” connection) or as a daisy-chained connection. The clock signals should be routed as a differential pair, i.e., as parallel traces dimensioned to result in a nominal 100 ohm impedance. A parallel termination resistor is placed between CK and CK# to match the clock pair impedance. If the “Y” topology is used, each branch should be individually terminated with a 220 ohm resistor as near as possible to the receiving end (DRAM). In the daisy-chain topology, a single 100 ohm resistor is used, again very near the receiving end.

Checklist for clock and signal termination,

- Series termination – nominal 33 ohms – is recommended for all DRAM signals
- Optimal termination resistor values can be determined by experiment
- Differential clock signals should be routed as a differential pair, targeted for a differential impedance of 100 ohms (each clock trace 50 ohms)
- Parallel termination of each clock differential pair should be used

13.1.4 Power Supply Distribution

The power supply distribution system (PDS, for short) is responsible for providing power to the system components. It consists of the power supplies themselves, plus the PCB routing and decoupling components. The high speeds involved in the DDR system make the PDS design critical to achieving reliable operation. This section will include specific recommendations for the SSTL-2 supply rail (VDD_2V5), the ground rail (VSS) and the SSTL-2 reference voltage.

13.1.4.1 VDD_2V5 Rail

The SSTL-2 supply rail (VDD_2V5) should be well decoupled using proper high frequency bypassing techniques. Sigma recommends the use of at least 8 high quality ceramic SMT capacitors in close proximity to the BGA package. Size 0603 capacitors of value 0.1uF should be adequate, but size 0402 is even better. To the degree routing will permit, they should be placed directly underneath the BGA (the Vantage design has four directly underneath, and several more near the perimeter of the BGA). Via-in-pad connections to the internal planes will yield the best performance, but in any case stub lengths to vias must be kept to a minimum. Use of multiple vias per capacitor lead is desirable.

The Vantage reference design uses 2 vias per VDD_2V5 ball to connect to the power supply plane of the PCB stack. This is recommended to reduce parasitic inductance in the power distribution path.

Each DRAM device must also be well decoupled. Use ceramic 0603 SMT capacitors placed very close to the DRAM pins, directly connected through short layer-1 traces.

Checklist for VDD_2V5 PDS,

- Use at least 8 high quality 0603 or 0402 ceramic decoupling caps for SMP8634
- Mount at least four directly underneath the BGA in the SSTL- interface section
- Decouple each DDR DRAM according to DRAM vendors recommendations
- If possible, use 2 vias per BGA VDD_2V5 ball
- Use trace width of 20 mils or greater between ball pads and vias for all VDD_2V5 and VSS connections

13.1.4.2 Vref Reference Voltage

Vref is a reference voltage for the SSTL-2 signaling scheme. It has a nominal value equal to half of the SSTL-2 supply voltage, or 1.3V nominal. It is usually created using a resistive voltage divider connected between VDD_2V5 and VSS. It is very important that Vref be kept as clean as possible. Since high frequency signal coupling is proportional to trace length, the Vref trace should be kept short and clear of adjacent fast switching signals. In addition, the Vref signal should be effectively decoupled to both VDD_2V5 and to VSS.

In order to keep Vref traces short, the Vantage reference design uses three separate voltage dividers. One divider provides Vref for both DRAMs of bank 0, another supplies the DRAMs of bank 1, and the third provides Vref to the SMP8634. Place the Vref generator for the SMP8634 directly underneath the Vref balls of the BGA.

Each Vref generator is constructed as shown below:

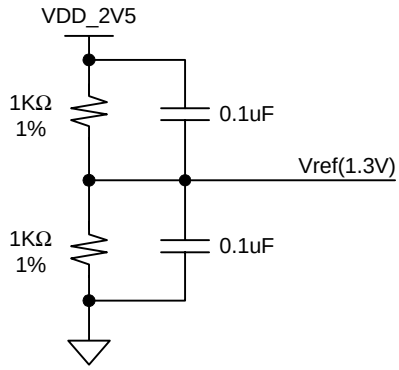


Figure 13-2 **Vref generator circuit**

13.2 Licensing

13.2.1 General Notifications

"Dolby" and the "double-D" symbol are trademarks of Dolby Laboratories. Manufactured under license from Dolby Laboratories. This product contains one or more programs protected under international and U.S. copyright laws as unpublished works. They are confidential and proprietary to Dolby Laboratories. Their reproduction or disclosure, in whole or in part, or the production of derivative works therefrom without the express permission of Dolby Laboratories is prohibited. Copyright 1993-2005 by Dolby Laboratories. All rights reserved. Supply of this Implementation of Dolby technology does not convey a license nor imply a right under any patent, or any other industrial or intellectual property right of Dolby Laboratories, to use this Implementation in any finished end-user or ready-to-use final product. It is hereby notified that a license for such use is required from Dolby Laboratories.

13.2.2 3rd-party Intellectual Property Licenses

Many 3rd party intellectual property licenses are typically required before a product can go into production. These licenses are required regardless of the chosen chip supplier. For your convenience, we have indicated various 3rd party intellectual property licenses you may need.

13.2.2.1 Macrovision

Prior to ordering Macrovision-enabled chips, you will need to have a Macrovision (http://www.macrovision.com) license. Before shipping each order, Sigma Designs will verify your license with Macrovision.

13.2.2.2 Licenses that Sigma will Verify

If your product supports any of the following features, you will need to have the corresponding license(s) before ordering production chips. Before providing you with a secure "authorization token" that enables the desired feature(s), Sigma Designs will ask for a copy of the appropriate license(s) and/or verify your license(s) with licensee(s). We will again verify your license(s) before shipping each order.

Audio

- Dolby audio decoding and /or encoding feature(s): appropriate license(s) from Dolby (http://www.dolby.com).
- DTS audio decoding and/or decoding feature(s): appropriate license(s) from DTS (http://www.dtsonline.com).
- WMA and/or WMA Pro audio decoding feature(s): "Windows Media Components Final Product Agreement" from Microsoft (http://www.microsoft.com/windows/windowsmedia/licensing/default.aspx).
- ATRAC3 audio decoding feature: ATRAC3 license from Sony (http://www.sony.com).

Video

- WMV9 decoding feature: "Windows Media Components Final Product Agreement" from Microsoft (http://www.microsoft.com/windows/windowsmedia/licensing/default.aspx).

Optical Media Playback

- DVD-Video playback feature: CSS license from DVD CCA (http://www.dvdcca.org). Note that you must get a DVD format/logo license from DVD FLLC before a CSS license will be issued. You will also need to sign our CSS Microcode License Agreement.
- DVD-R and DVD-RW playback feature: CPRM for DVD license from 4C (http://www.4centity.com)
- DVD+R and DVD+RW playback feature: VCPS license from Philips (http://www.licensing.philips.com)
- DVD-Audio playback feature: CPPM for DVD license from 4C (http://www.4centity.com); Audio Watermark Detector license from Verance (http://www.verance.com)
- BD-ROM playback feature: AACS license from AACS LA; BD+ license from Blu-ray Disc Association (http://www.blu-raydisc.info/); VCMS/AV Detector license from Verance (http://www.verance.com). Note that you must get a BD format/logo license from BDA before a BD+ license will be issued.
- BD-REv2 and/or BD-R playback feature: AACS license from AACS LA (http://www.aacsla.com)
- HD DVD-Video playback feature: AACS license from AACS LA (http://www.aacsla.com); VCMS/AV Detector license from Verance (http://www.verance.com)

DRM

- WM DRM 10 feature: "Windows Media DRM 10 for Devices Final Product Distribution Agreement" from Microsoft (http://www.microsoft.com/windows/windowsmedia/licensing/default.aspx)
- DTCP/IP feature: DTCP/IP license from DTLA (http://www.dtcp.com)
- HDMI output feature: HDCP license from Digital Content Protection (http://www.digital-cp.com); HDMI license from HDMI LLC (http://www.hdmi.org)

Note that since Sigma Designs only provides low-level software tools, additional application-level software development is required to make any desired feature fully functional.

13.2.2.3 Licenses that Sigma does not Verify

For your convenience, we also indicate other various 3rd party intellectual property licenses you may need to obtain prior to going into production. Sigma Designs does not verify these licenses and does not guarantee that all required licenses are listed.

- ATSC feature: MPEG LA (http://www.mpegla.com)
- BD feature(s): Blu-ray Disc Association (http://www.blu-raydisc.info/)
- DVB-T feature: MPEG LA (http://www.mpegla.com)
- DVD-Video feature: DVD FLLC (http://www.dvdfllc.com); DVD 6C (http://www.dvd6cla.com); Philips (4C) (http://www.licensing.philips.com); Nissim (http://www.nissim.com)
- HD DVD-Video feature: DVD FLLC (http://www.dvdfllc.com)
- JPEG feature: Forgent Networks (http://www.forgent.com)
- MPEG audio feature(s): Via Licensing (http://www.vialicensing.com) for MPEG-2 AAC and MPEG-4, including HE-AAC. Philips (http://www.licensing.philips.com) for Layers I and II for DVD players. For Layer III (mp3): Sisvel (http://www.sisvel.com) (Philips) for non-US patents; Audio MPEG (http://www.audiompeg.com) (Philips) for US patents; MP3 Licensing (http://www.mp3licensing.com) (Thomson)
- MPEG-2 system and/or video feature(s): MPEG LA (http://www.mpegla.com)
- MPEG-4.2 video feature: MPEG LA (http://www.mpegla.com); AT&T (http://www.att.com)
- MPEG-4.10 (H.264) video feature: MPEG LA (http://www.mpegla.com); Via Licensing (http://www.vialicensing.com); AT&T (http://www.mpegla.com)
- MPEG-4.1 systems feature: MPEG LA (http://www.mpegla.com)
- SMPTE 421M (VC-1) video feature: MPEG LA (http://www.mpegla.com)
- WMA, WMV9, WM DRM and/or HighMAT feature(s): "Logo Agreement" from Microsoft (http://www.microsoft.com/windows/windowsmedia/licensing/default.aspx)

13.2.3 Licenses and Royalty Responsibility

Licensing and any associated royalty payments for 3rd party intellectual property are the responsibility of the customer.

13.3 Basic Analog Video Parameters

Table 13-2 **Basic analog video parameters**

	NTSC-M NTSC-J	PAL-B/D/ G/H/I	480P	576P	720P	1080i	1080P
Frame Rate	29.97	29.97	59.94	25	59.94	29.97 (25)	59.94
HEIGHT	0x020D 525	0x0271 625	0x020D 525	0x0271 625	0x02EE 750	0x0465 1125	0x0465 1125
WIDTH	0x0D68 3432	0x0D80 3456	0x0D68 3432	0x0D80 3456	0x0672 1650	0x0898 (0x0A50) 2200 (2640)	0x0898 (0x0A50) 2200 (2640)
HSYNC1	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
HSYNC0	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001
VSYNC_O_0_PIXEL	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001
VSYNC_O_0_LINE	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001
VSYNC_O_1_PIXEL	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000	0x0000
VSYNC_O_1_LINE	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001	0x0001
VSYNC_E_0_PIXEL	0x06B5 WIDTH/2+1 1717	0x06C1 WIDTH/2+1 1729	-	-	-	0x044D (0x0529) WIDTH/2+1 1101 (1321)	-
VSYNC_E_0_LINE	0x0107 263	0x0139 313	-	-	-	0x0233 563	-
VSYNC_E_1_PIXEL	0x06B4 WIDTH/2 1716	0x06C0 WIDTH/2 1728	-	-	-	0x044C (0x0528) WIDTH/2 1100 (1320)	-
VSYNC_E_1_LINE	0x0107 263	0x0139 313	-	-	-	0x0233 563	-
HSYNC_WIDTH	-	-	-	-	0x0050 80	0x0058 88	0x0058 58
VSYNC_START	-	-	-	-	0x0104 260	0x0084 132	0x0084 132
VSYNC_WIDTH	-	-	-	-	0x0500 1280	0x0370 880	0x07BC 1980
X Offset	130*4	132*4	130*4	132*4	305	240	240
Y Offset (Top)	18	22	36	44	25	20	40
Y Offset (Bottom)	18	22	-	-	-	21	-

13.4 Main Video Scaler Filters

Four ‘4-taps’ filters are available in the main video scaler for luma/chroma horizontal/vertical filtering. They are four 63-coefficient, 16 times interpolating filters. For each of the 16 interpolating positions 4 coefficients (taps) are used (64 = 4 x 16).

13.4.1 Frequency Response

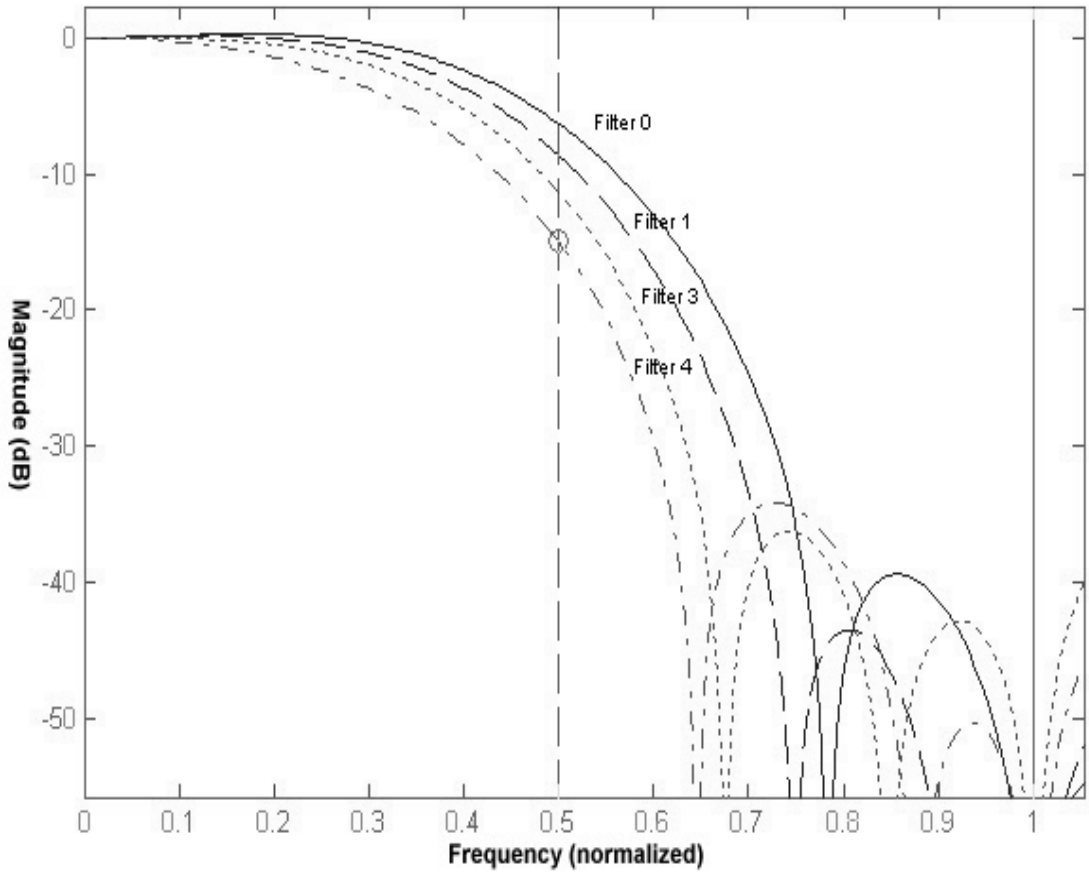


Figure 13-3 **Frequency response**

13.4.2 Impulse Response (x 1024)

Filter 0	64	64	62	60	57	53	48	44	38	33	28	22	17	12	8	4	0	-2	-4	-6	-7	-7	-7	-7	-6	-6	-5	-4	-3	-2	-2	-2
Filter 1	58	58	57	55	52	49	46	42	37	33	28	24	20	15	11	8	3	2	-1	-3	-4	-5	-5	-6	-5	-5	-5	-4	-4	-3	-3	-4
Filter 2	52	52	51	50	48	46	43	40	36	33	29	25	21	17	14	10	6	5	2	1	-1	-2	-3	-4	-4	-5	-5	-5	-4	-4	-3	-3
Filter 3	46	46	44	43	42	40	38	36	34	31	28	25	22	19	16	12	9	8	6	4	3	1	0	-1	-2	-2	-2	-2	-3	-2	-2	-2

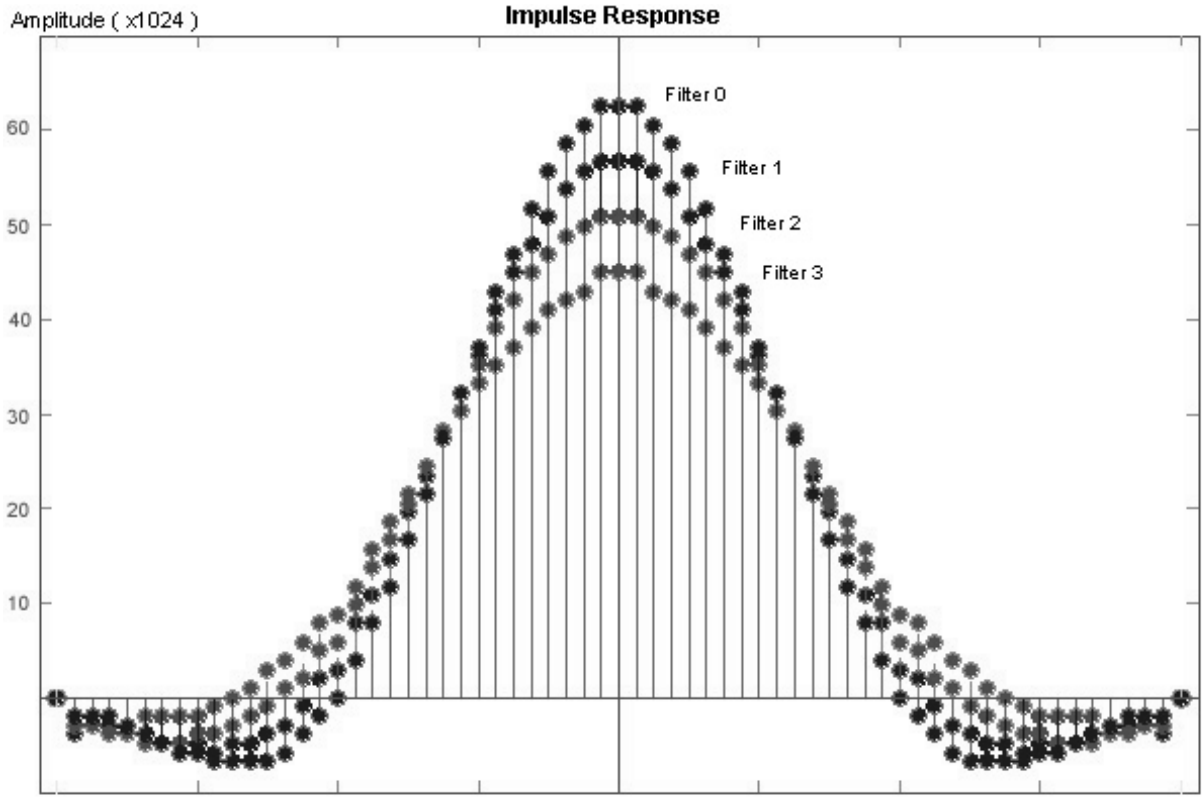


Figure 13-4 **Impulse response**

13.5 Color Space Conversion Basic Matrices

$$\begin{bmatrix} Y/G \\ Cb/B \\ Cr/R \end{bmatrix} = \begin{bmatrix} \mathbf{m00} & \mathbf{m01} & \mathbf{m02} \\ \mathbf{m10} & \mathbf{m11} & \mathbf{m12} \\ \mathbf{m20} & \mathbf{m21} & \mathbf{m22} \end{bmatrix} \times \begin{bmatrix} Y/G \\ Cb/B \\ Cr/R \end{bmatrix} + \begin{bmatrix} \mathbf{Cst0} \\ \mathbf{Cst1} \\ \mathbf{Cst2} \end{bmatrix}$$

	Real	*Analog*	*Digital*
601→ RGB 0-255	$\begin{bmatrix} 1.164 & -0.391 & -0.813 & 135488 \\ 1.164 & 2.018 & 0 & -276928 \\ 1.164 & 0 & 1.596 & -222912 \end{bmatrix}$	$\begin{bmatrix} 4768 & 31166 & 29438 & 2168 \\ 4768 & 8266 & 0 & 11953 \\ 4768 & 0 & 6537 & 12817 \end{bmatrix}$	$\begin{bmatrix} 2384 & 15583 & 14719 & 1084 \\ 2384 & 4133 & 0 & 5977 \\ 2384 & 0 & 3268 & 6409 \end{bmatrix}$
601→ RGB 16-235	$\begin{bmatrix} 1 & -0.336 & -0.698 & 132352 \\ 1 & 1.732 & 0 & -221696 \\ 1 & 0 & 1.371 & -175488 \end{bmatrix}$	$\begin{bmatrix} 4096 & 31392 & 29909 & 2118 \\ 4096 & 7094 & 0 & 12837 \\ 4096 & 0 & 5616 & 13578 \end{bmatrix}$	$\begin{bmatrix} 2048 & 15696 & 14954 & 1059 \\ 2048 & 3547 & 0 & 6418 \\ 2048 & 0 & 2808 & 6788 \end{bmatrix}$
709 → RGB 0-255	$\begin{bmatrix} 1.164 & -0.213 & -0.534 & 76.992 \\ 1.164 & 2.115 & 0 & -289.344 \\ 1.164 & 0 & 1.793 & -248.128 \end{bmatrix}$	$\begin{bmatrix} 4768 & 31896 & 30581 & 1232 \\ 4768 & 8663 & 0 & 11754 \\ 4768 & 0 & 7344 & 12414 \end{bmatrix}$	$\begin{bmatrix} 2384 & 15948 & 15290 & 616 \\ 2384 & 4332 & 0 & 5877 \\ 2384 & 0 & 3672 & 6207 \end{bmatrix}$
709 → RGB 16-235	$\begin{bmatrix} 1 & -0.183 & -0.459 & 82.176 \\ 1 & 1.816 & 0 & -232.448 \\ 1 & 0 & 1.540 & -197.12 \end{bmatrix}$	$\begin{bmatrix} 4096 & 32018 & 30888 & 1315 \\ 4096 & 7438 & 0 & 12665 \\ 4096 & 0 & 6308 & 13230 \end{bmatrix}$	$\begin{bmatrix} 2048 & 16009 & 15444 & 657 \\ 2048 & 3719 & 0 & 6332 \\ 2048 & 0 & 3154 & 6615 \end{bmatrix}$
RGB 0-255 → 601	$\begin{bmatrix} 0.504 & 0.098 & 0.257 & 16 \\ -0.291 & 0.439 & -0.148 & 128 \\ -0.368 & -0.071 & 0.439 & 128 \end{bmatrix}$	$\begin{bmatrix} 2064 & 401 & 1053 & 256 \\ 31576 & 1798 & 32162 & 2048 \\ 31261 & 32477 & 1798 & 2048 \end{bmatrix}$	$\begin{bmatrix} 1032 & 201 & 526 & 128 \\ 15788 & 899 & 16081 & 1024 \\ 15630 & 16239 & 899 & 1024 \end{bmatrix}$
RGB 16-235 → 601	$\begin{bmatrix} 0.587 & 0.114 & 0.299 & 0 \\ -0.339 & 0.511 & -0.172 & 128 \\ -0.428 & -0.083 & 0.511 & 128 \end{bmatrix}$	$\begin{bmatrix} 2404 & 467 & 1225 & 0 \\ 31379 & 2093 & 32063 & 2048 \\ 31015 & 32428 & 2093 & 2048 \end{bmatrix}$	$\begin{bmatrix} 1202 & 233 & 612 & 0 \\ 15690 & 1047 & 16032 & 1024 \\ 15507 & 16214 & 1047 & 1024 \end{bmatrix}$
RGB 0-255 → 709	$\begin{bmatrix} 0.614 & 0.062 & 0.183 & 16 \\ -0.338 & 0.439 & -0.101 & 128 \\ -0.399 & -0.040 & 0.439 & 128 \end{bmatrix}$	$\begin{bmatrix} 2515 & 254 & 750 & 256 \\ 31384 & 1798 & 32354 & 2048 \\ 31134 & 32604 & 1798 & 2048 \end{bmatrix}$	$\begin{bmatrix} 1257 & 127 & 375 & 128 \\ 15692 & 899 & 16177 & 1024 \\ 15567 & 16302 & 899 & 1024 \end{bmatrix}$
RGB 16-235 → 709	$\begin{bmatrix} 0.715 & 0.072 & 0.213 & 0 \\ -0.394 & 0.511 & -0.117 & 128 \\ -0.464 & -0.047 & 0.511 & 128 \end{bmatrix}$	$\begin{bmatrix} 2929 & 295 & 872 & 0 \\ 31154 & 2093 & 32289 & 128 \\ 30867 & 32575 & 2093 & 128 \end{bmatrix}$	$\begin{bmatrix} 1464 & 147 & 436 & 0 \\ 15577 & 1047 & 16144 & 1024 \\ 15434 & 16288 & 1047 & 1024 \end{bmatrix}$
601 → 709	$\begin{bmatrix} 1 & -0.116 & -0.207 & 41.344 \\ 0 & 1.017 & 0.115 & -16.896 \\ 0 & 0.075 & 1.024 & -12.672 \end{bmatrix}$	$\begin{bmatrix} 4096 & 32293 & 31920 & 662 \\ 0 & 4166 & 471 & 16114 \\ 0 & 307 & 4194 & 16181 \end{bmatrix}$	$\begin{bmatrix} 2048 & 16146 & 15960 & 331 \\ 0 & 2083 & 236 & 8057 \\ 0 & 154 & 2097 & 8091 \end{bmatrix}$
709 → 601	$\begin{bmatrix} 1 & 0.1 & 0.191 & -37.248 \\ 0 & 0.99 & -0.109 & 15.232 \\ 0 & -0.072 & 0.983 & 11.392 \end{bmatrix}$	$\begin{bmatrix} 4096 & 410 & 782 & 15788 \\ 0 & 4055 & 32322 & 244 \\ 0 & 32473 & 4026 & 182 \end{bmatrix}$	$\begin{bmatrix} 2048 & 205 & 391 & 7894 \\ 0 & 2028 & 16161 & 122 \\ 0 & 16237 & 2013 & 91 \end{bmatrix}$

sign(x) = 1 if x < 0; 0 otherwise
round m_a,cst_a,m_d and cst_d to the closest integer

$m_a = 32768 * sign(m_v) + m_v * 4096$
$cst_a = 16384 * sign(cst_v) + cst_v * 16$

$m_d = 16384*sign(m_v) + m_v * 2048$
$cst_d = 8192*sign(cst_v) + cst_v * 8$

13.6 Video Output Scaling Examples

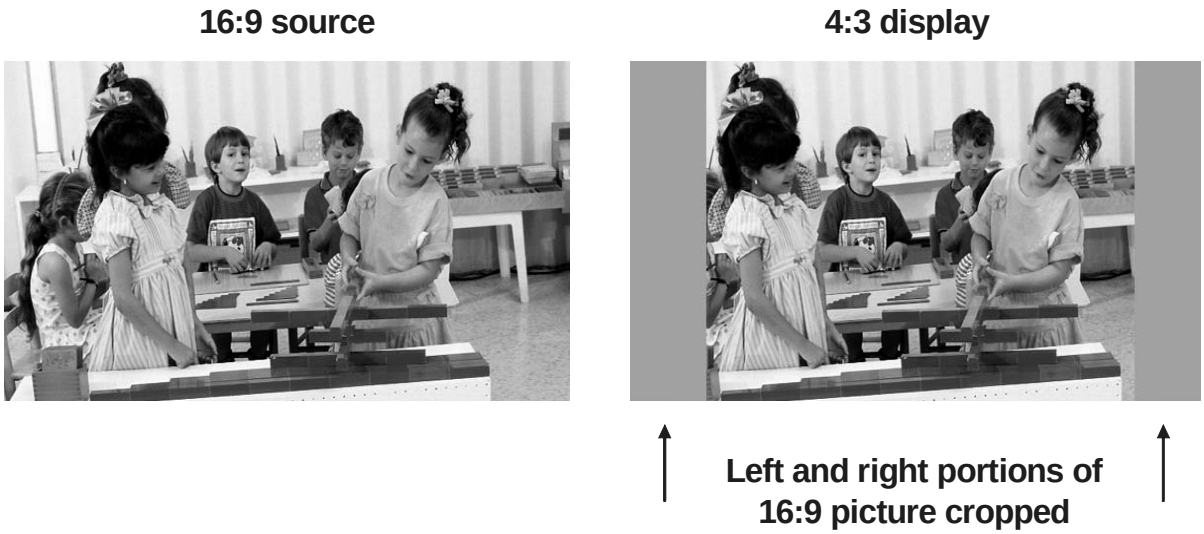


Figure 13-5 **Scaling 16:9 content for a 4:3 display – “Crop” mode**

Figure 13-6 **Scaling 16:9 content for a 4:3 display – “Letterbox” mode**

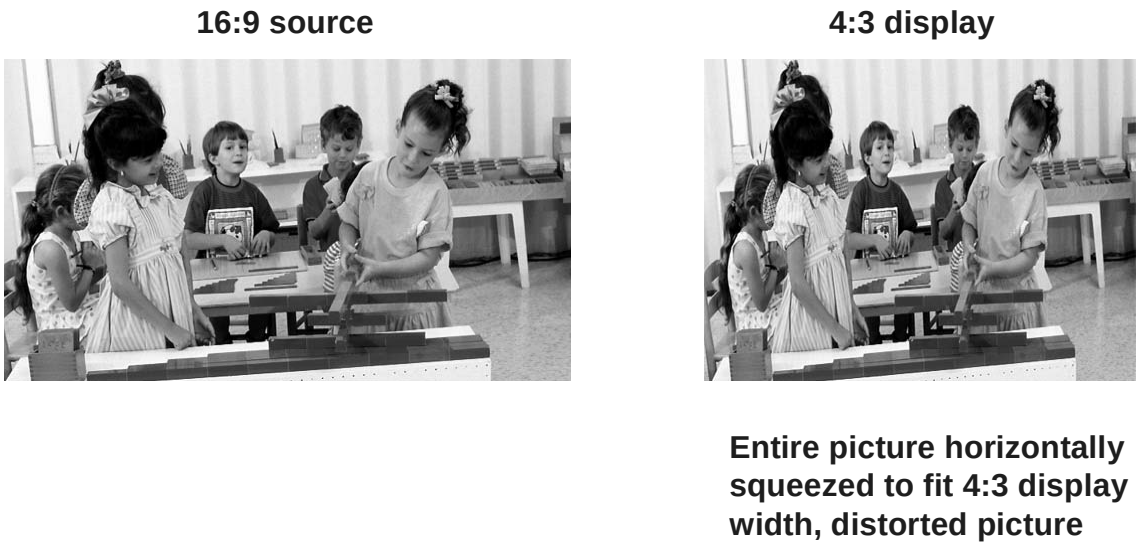


Figure 13-7 **Scaling 16:9 content for a 4:3 display – "Squeezed" mode**

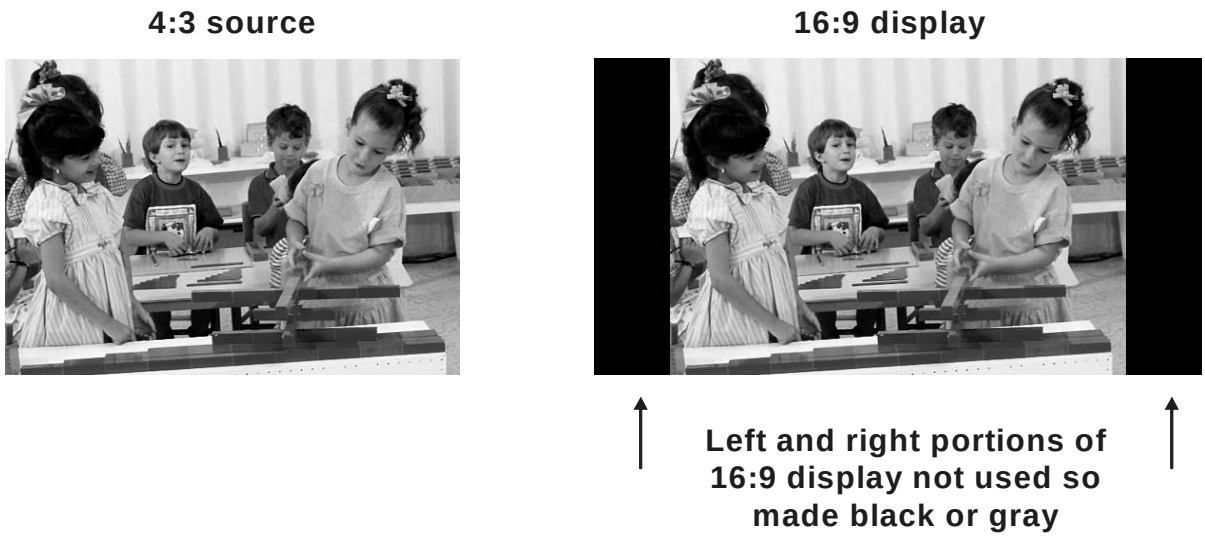


Figure 13-8 **Scaling 4:3 content for a 16:9 display – "Pillar" mode**

4:3 source

16:9 display

Entire picture linearly scaled horizontally to fill 16:9 display, distorted picture unless used with anamorphic content

Figure 13-9 **Scaling 4:3 content for a 16:9 display – "Wide" mode**

4:3 source

16:9 display

Top and bottom portion of picture deleted, then scaled to fill 16:9 display

Figure 13-10 **Scaling 4:3 content for a 16:9 display – "Zoom" mode**

4:3 source

16:9 display

Edges of picture nonlinearly scaled horizontally to fill 16:9 display, distorted picture on left and right sides

Figure 13-11 **Scaling 4:3 content for a 16:9 display – “Panorama” mode**

13.7 SDTV vs. HDTV Colorimetry Example

Figure 13-12 **SDTV Vs. SDTV Colorimetry Example**

13.8 Supported Media Formats

Table 13-3 **Supported Media Formats**

Format		Feature	Supported
Blu-ray	Media	BD-ROM	x
		BD-RE v1, v2	x
	Trick Play	Pause / resume	x
		Stop	x
		Next / Previous	x
		Fast forward (2X, … 32X)	x
		Fast reverse (2X, … 32X)	x
		Audio playback during FF or RW	
		Slow playback (1/2X, … 1/32X)	x
		Reverse playback	
		Frame advance	x
		Title Search	x
		Chapter Search	x
		Time search	x
		Time advance (user defined)	x
		Angle	x
		Subtitle	x
		Audio Select	x
		Changeable Subpicture position	
		Brightness control - subpicture	x
		‘Scan’ of (chapter, title)	x
		Playback speed control (x0.6 - x1.4), with audio	
		Instant replay	x
		Advance x seconds and resume playback	x
		Bookmark	x
		Playlist	x
		Shuffle playback	x
		Repeat (chapter, title, disc)	x
		A-B Repeat	x

Table 13-3 **Supported Media Formats (Continued)**

Format		Feature	Supported
DVD-Video	Media	DVD-R	x
		DVD-RW	x
		DVD-R Dual Layer	x
	Trick Play	Pause / resume	x
		Stop	x
		Next / Previous	x
		Fast forward (2X, … 32X)	x
		Fast reverse (2X, … 32X)	x
		Audio playback during FF or RW	x
		Slow playback (1/2X, … 1/32X)	x
		Reverse playback	
		Frame advance	x
		Title Search	x
		Chapter Search	x
		Time search	x
		Time advance (user defined)	x
		Angle	x
		Subtitle	x
		Audio Select	x
		Changeable Subpicture position	
		Brightness control - subpicture	x
		‘Scan’ of (chapter, title)	x
		Playback speed control (x0.6 - x1.4), with audio	
		Instant replay	x
		Advance x seconds and resume playback	x
		Bookmark	x
		Playlist	x
		Shuffle playback	x
		Repeat (chapter, title, disc)	x
		A-B Repeat	x

Table 13-3 **Supported Media Formats (Continued)**

Format		Feature	Supported
		Jacket picture support	x
DVD-VR	Media	DVD-R	x
		DVD-RW	x
		DVD-RAM	x
		DVD-R Dual Layer	
	Trick Play	Pause / resume	x
		Stop	x
		Next / Previous	x
		Fast forward (2X, … 32X)	x
		Fast reverse (2X, … 32X)	x
		Audio playback during FF or RW	x
		Slow playback (1/2X, … 1/32X)	x
		Reverse playback	
		Frame advance	x
		Title Search	x
		Chapter Search	x
		Time search	x
		Time advance (user defined)	x
		Angle	x
		Subtitle	x
		Audio Select	x
		Changeable Subpicture position	
		Brightness control - subpicture	x
		‘Scan’ of (chapter, title) -	x
		Playback speed control (x0.6 - x1.4), with audio	x
		Instant replay	x
		Advance x seconds and resume playback.	x
		Bookmark	x
		Playlist	x
		Shuffle playback	x

Table 13-3 **Supported Media Formats (Continued)**

Format		Feature	Supported
		Repeat (chapter, title, disc)	x
		A-B Repeat	x
		Thumbnails of playlist	
		Thumbnail of playlist	
DVD+VR	Media	+R	x
		+RW	x
		+R Dual Layer	x
CD-DA	Media	CD-R	x
		CD-R with HighMAT	x
		CD-RW	x
		CD-RW with HighMAT	x
		CD-TEXT	x
		dtsCD	x
		HDCD	
	Trick Play	Pause / resume	x
		Stop	x
		Skip	x
		Fast forward	x
		Fast reverse	x
		Audio output during search	x
		Track search	x
		Time search	x
		Instant replay	x
		Bookmark	x
		Playlist support	x
		Shuffle	x
		Repeat (track, disc)	x
		A-B Repeat	x
		Jacket picture support	
VideoCD v1.x and v2.0	Media	CD-R	x

Table 13-3 **Supported Media Formats (Continued)**

Format		Feature	Supported
		CD-RW	x
	Trick Play	Pause / resume	x
		Stop	x
		Next / previous	x
		Fast forward	x
		Fast reverse	x
		Audio output during search	x
		Slow playback	x
		Reverse playback	
		Frame advance	x
		Time search	x
		Track search	x
		Audio Select (same audio to both left and right)	x
		Instant replay	x
		Advance x seconds and resume playback.	x
		Bookmark	x
		Playlist support	x
		Shuffle	x
		Repeat	x
		A-B Repeat	x
		PBC select	x
		PBC select during Play Back	x
		Zoom	x
		Karaoke CD	x
		VCD with CDDA	x
SVCD IEC 62107-2000	Media	CD-R	x
		CD-RW	x
	Trick Play	Pause / resume	x
		Stop	x
		Next / previous	x

Table 13-3 **Supported Media Formats (Continued)**

Format	Feature	Supported
	Fast forward	x
	Fast reverse	x
	Audio output during search	x
	Slow playback	x
	Reverse playback	
	Frame advance	x
	Time search	x
	Track search	x
	Audio Select (same audio to both left and right)	x
	Instant replay	x
	Advance x seconds and resume playback.	x
	Bookmark	x
	Playlist support	x
	Shuffle	x
	Repeat	x
	A-B Repeat	x
	PBC select	x
	PBC select during Play Back	x
	Zoom	x
	Karaoke CD	x
	VCD with CDDA	x
	OGT display & select	x

13.9 Supported File Formats

Table 13-4 **File formats supported by SMP8634**

Format		Feature	Supported
File Player		ISO9660 level 2	x
		UDF 1.02	x
		UDF 1.5	x
		UDF 2.01	x
		UDF 2.5	x
		UDF 2.6	
		Number of folders	x
		Number of files	x
		Display by filename	x
		Display font	x
		Slideshow mode	x
AVI	Video	MPEG-2	x
		MPEG-4.2	x
		MPEG-4.10 (H.264)	
		SMPTE 421M (VC-1)	
		WMV9	x
	Audio	LPCM	
		Dolby Digital	x
		WMA	
		WMA Pro	
ASF	Video	MPEG-2	x
		MPEG-4.2	x
		MPEG-4.10 (H.264)	
		SMPTE 421M (VC-1)	
		WMV9	x
	Audio	LPCM	
		Dolby Digital	x
		WMA	
		WMA Pro	

Table 13-4 **File formats supported by SMP8634**

Format		Feature	Supported
MP3	Trick Play	Pause/resume	x
		Stop	x
		Next / previous	x
		Fast forward	x
		Fast reverse	
		Audio output during search	x
		Elapsed time	x
WMA	Trick Play	Pause/resume	x
		Stop	x
		Next/previous	x
		Fast forward	x
		Fast reverse	
		Audio output during search	x
		Elapsed time	x
		Time search	x
		Instant replay	x
		Bookmark	x
		Playlist	x
		Shuffle	x
		Repeat	x
		A-B Repeat	x
		Time search	x
		Instant replay	x
		Bookmark	x
		Playlist	x
		Shuffle	x
		Repeat	x
		A-B Repeat	x
WMA Pro	Trick Play	Pause/resume	x
		Stop	x

Table 13-4 **File formats supported by SMP8634**

Format	Feature	Supported
	Next/previous	x
	Fast forward	x
	Fast reverse	
	Audio output during search	x
	Elapsed time	x
	Time search	x
	Instant replay	x
	Bookmark	x
	Playlist	x
	Shuffle	x
	Repeat	x
	A-B Repeat	x
DivX	v3.11	x
	v4.x	x
	v5.x	x
Trick Play	Pause/resume	x
	Stop	x
	Next/previous	x
	Fast forward	x
	Fast reverse	x
	Audio output during search	x
	Slow playback	x
	Reverse playback	x
	Frame advance	x
	Time search	x
	Time advance (user defined)	x
	Subtitle	x
	Audio Select	x
	Changeable subpicture position	x
	Brightness control of subpicture	x

Table 13-4 **File formats supported by SMP8634**

Format		Feature	Supported
		Instant replay	x
		Advance x seconds and resume playback	x
		Bookmark	x
		Playlist	x
		Shuffle	x
		Repeat	x
		A-B Repeat	x
		DivX DRM	
JPEG	Trick Play	Pause/resume	x
		Stop	x
		Next/previous	x
		Bookmark	x
		Playlist	x
		Shuffle	x
		Repeat	x
		Zoom	x
		Rotate CW/CCW 90 degrees	x
		Thumbnail	x
		File information	x
		Slideshow	x
		Set up a time of slide show	x

13.10 Supported Broadcast Formats

Table 13-5 **Broadcast Formats supported by SMP8634**

Format		Feature	Supported
ATSC	Video	MPEG-2	x
	Audio	Dolby Digital	x
DVB	Video	MPEG-2	x
	Audio	Dolby Digital	x

13.11 Raster Operations

Ternary (256) raster-operation codes define how GDI combines the bits in a source bitmap with the bits in the destination bitmap. Each raster-operation code represents a Boolean operation in which the values of the pixels in the source, the selected brush, and the destination are combined. Following are the three operands used in these operations:

Table 13-6 **Operands used in raster operations**

Operand	Meaning
D	Destination bitmap
P	Selected brush (also called pattern)
S	Source bitmap

Boolean operators used in these operations follow:

Table 13-7 **Boolean operator meanings**

Operator	Meaning
a	Bitwise AND
n	Bitwise NOT (inverse)
o	Bitwise OR
X	Bitwise exclusive OR (XOR)

All Boolean operations are presented in reverse Polish notation. For example, the following operation replaces the values of the pixels in the destination bitmap with a combination of the pixel values of the source and brush: PSo

The following operation combines the values of the pixels in the source and brush with the pixel values of the destination bitmap (there are alternative spellings of the same function, so although a particular spelling may not be in the list, an equivalent form would be): DPSoo

Each raster-operation code is a 32-bit integer whose high-order word is a Boolean operation index and whose low-order word is the operation code. The 16-bit operation index is a zero-extended, 8-bit value that represents the result of the Boolean operation on predefined brush, source, and destination values. For example, the operation indexes for the PSo and DPSoo operations are shown in the following list:

Table 13-8 **Operation indexes for the PSo and DPSoo operations**

P	S	D	PSo	DPSoo
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1
Operation index:			00FCh	00FEh

In this case, PSo has the operation index 00FC (read from the bottom up); DPSoo has the operation index 00FE. These values define the location of the corresponding raster-operation codes, as shown in the "Raster-Operation Codes" table. The PSo operation is in line 252 (00FCh) of the table; DPSoo is in line 254 (00FEh).

The most commonly used raster operations have been given special names in the SDK header file, WINDOWS.H. You should use these names whenever possible in your applications.

When the source and destination bitmaps are monochrome, a bit value of zero represents a black pixel and a bit value of 1 represents a white pixel. When the source and the destination bitmaps are color, those colors are represented with RGB values.

13.11.1 Raster Operation Codes

Table 13-9 **Raster operation codes**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
00	00000042	0	BLACKNESS
01	00010289	DPSoon	–
02	00020C89	DPSona	–
03	000300AA	PSon	–
04	00040C88	SDPona	–
05	000500A9	DPon	–
06	00060865	PDSxnon	–
07	000702C5	PDSaon	–
08	00080F08	SDPnaa	–
09	00090245	PDSxon	–
0A	000A0329	DPna	–
0B	000B0B2A	PSDnaon	–
0C	000C0324	SPna	–
0D	000D0B25	PDSnaon	–
0E	000E08A5	PDSonon	–
0F	000F0001	Pn	–
10	00100C85	PDSona	–
11	001100A6	DSon	NOTSRCERASE
12	00120868	SDPxnon	–
13	001302C8	SDPaon	–
14	00140869	DPSxnon	–
15	001502C9	DPSaon	–
16	00165CCA	PSDPSanaxx	–
17	00171D54	SSPxDSxaxn	–
18	00180D59	SPxPDxa	–
19	00191CC8	SDPSanaxn	–
1A	001A06C5	PDSPaox	–
1B	001B0768	SDPSxaxn	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
1C	001C06CA	PSDPaox	–
1D	001D0766	DSPDxaxn	–
1E	001E01A5	PDSox	–
1F	001F0385	PDSoan	–
20	00200F09	DPSnaa	–
21	00210248	SDPxon	–
22	00220326	DSna	–
23	00230B24	SPDnaon	–
24	00240D55	SPxDSxa	–
25	00251CC5	PDSPanaxn	–
26	002606C8	SDPSaox	–
27	00271868	SDPSxnox	–
28	00280369	DPSxa	–
29	002916CA	PSDPSaoxxn	–
2A	002A0CC9	DPSana	–
2B	002B1D58	SSPxPDxaxn	–
2C	002C0784	SPDSoax	–
2D	002D060A	PSDnox	–
2E	002E064A	PSDPxox	–
2F	002F0E2A	PSDnoan	–
30	0030032A	PSna	–
31	00310B28	SDPnaon	–
32	00320688	SDPSoox	–
33	00330008	Sn	NOTSRCCOPY
34	003406C4	SPDSaox	–
35	00351864	SPDSxnox	–
36	003601A8	SDPox	–
37	00370388	SDPoan	–
38	0038078A	PSDPoax	–
39	00390604	SPDnox	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
3A	003A0644	SPDSxox	–
3B	003B0E24	SPDnoan	–
3C	003C004A	PSx	–
3D	003D18A4	SPDSonox	–
3E	003E1B24	SPDSnaox	–
3F	003F00EA	PSan	–
40	00400F0A	PSDnaa	–
41	00410249	DPSxon	–
42	00420D5D	SDxPDxa	–
43	00431CC4	SPDSanaxn	–
44	00440328	SDna	SRCERASE
45	00450B29	DPSnaon	–
46	004606C6	DSPDaox	–
47	0047076A	PSDPxaxn	–
48	00480368	SDPxa	–
49	004916C5	PDSPDaoxxn	–
4A	004A0789	DPSDoax	–
4B	004B0605	PDSnox	–
4C	004C0CC8	SDPana	–
4D	004D1954	SSPxDSxoxn	–
4E	004E0645	PDSPxox	–
4F	004F0E25	PDSnoan	–
50	00500325	PDna	–
51	00510B26	DSPnaon	–
52	005206C9	DPSDaox	–
53	00530764	SPDSxaxn	–
54	005408A9	DPSonon	–
55	00550009	Dn	DSTINVERT
56	005601A9	DPSox	–
57	00570389	DPSoan	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
58	00580785	PDSPoax	–
59	00590609	DPSnox	–
5A	005A0049	DPx	PATINVERT
5B	005B18A9	DPSDonox	–
5C	005C0649	DPSDxox	–
5D	005D0E29	DPSnoan	–
5E	005E1B29	DPSDnaox	–
5F	005F00E9	DPan	–
60	00600365	PDSxa	–
61	006116C6	DSPDSaoxxn	–
62	00620786	DSPDoax	–
63	00630608	SDPnox	–
64	00640788	SDPSoax	–
65	00650606	DSPnox	–
66	00660046	DSx	SRCINVERT
67	006718A8	SDPSonox	–
68	006858A6	DSPDSonoxxn	–
69	00690145	PDSxxn	–
6A	006A01E9	DPSax	–
6B	006B178A	PSDPSoaxxn	–
6C	006C01E8	SDPax	–
6D	006D1785	PDSPDoaxxn	–
6E	006E1E28	SDPSnoax	–
6F	006F0C65	PDSxnan	–
70	00700CC5	PDSana	–
71	00711D5C	SSDxPDxaxn	–
72	00720648	SDPSxox	–
73	00730E28	SDPnoan	–
74	00740646	DSPDxox	–
75	00750E26	DSPnoan	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
76	00761B28	SDPSnaox	–
77	007700E6	DSan	–
78	007801E5	PDSax	–
79	00791786	DSPDSoaxxn	–
7A	007A1E29	DPSDnoax	–
7B	007B0C68	SDPxnan	–
7C	007C1E24	SPDSnoax	–
7D	007D0C69	DPSxnan	–
7E	007E0955	SPxDSxo	–
7F	007F03C9	DPSaan	–
80	008003E9	DPSaa	–
81	00810975	SPxDSxon	–
82	00820C49	DPSxna	–
83	00831E04	SPDSnoaxn	–
84	00840C48	SDPxna	–
85	00851E05	PDSPnoaxn	–
86	008617A6	DSPDSoaxx	–
87	008701C5	PDSaxn	–
88	008800C6	DSa	SRCAND
89	00891B08	SDPSnaoxn	–
8A	008A0E06	DSPnoa	–
8B	008B0666	DSPDxoxn	–
8C	008C0E08	SDPnoa	–
8D	008D0668	SDPSxoxn	–
8E	008E1D7C	SSDxPDxax	–
8F	008F0CE5	PDSanan	–
90	00900C45	PDSxna	–
91	00911E08	SDPSnoaxn	–
92	009217A9	DPSDPoaxx	–
93	009301C4	SPDaxn	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
94	009417AA	PSDPSoaxx	–
95	009501C9	DPSaxn	–
96	00960169	DPSxx	–
97	0097588A	PSDPSonoxx	–
98	00981888	SDPSonoxn	–
99	00990066	DSxn	–
9A	009A0709	DPSnax	–
9B	009B07A8	SDPSoaxn	–
9C	009C0704	SPDnax	–
9D	009D07A6	DSPDoaxn	–
9E	009E16E6	DSPDSaoxx	–
9F	009F0345	PDSxan	–
A0	00A000C9	DPa	–
A1	00A11B05	PDSPnaoxn	–
A2	00A20E09	DPSnoa	–
A3	00A30669	DPSDxoxn	–
A4	00A41885	PDSPonoxn	–
A5	00A50065	PDxn	–
A6	00A60706	DSPnax	–
A7	00A707A5	PDSPoaxn	–
A8	00A803A9	DPSoa	–
A9	00A90189	DPSoxn	–
AA	00AA0029	D	–
AB	00AB0889	DPSono	–
AC	00AC0744	SPDSxax	–
AD	00AD06E9	DPSDaoxn	–
AE	00AE0B06	DSPnao	–
AF	00AF0229	DPno	–
B0	00B00E05	PDSnoa	–
B1	00B10665	PDSPxoxn	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
B2	00B21974	SSPxDSxox	–
B3	00B30CE8	SDPanan	–
B4	00B4070A	PSDnax	–
B5	00B507A9	DPSDoaxn	–
B6	00B616E9	DPSDPaoxx	–
B7	00B70348	SDPxan	–
B8	00B8074A	PSDPxax	–
B9	00B906E6	DSPDaoxn	–
BA	00BA0B09	DPSnao	–
BB	00BB0226	DSno	MERGEPAINT
BC	00BC1CE4	SPDSanax	–
BD	00BD0D7D	SDxPDxan	–
BE	00BE0269	DPSxo	–
BF	00BF08C9	DPSano	–
C0	00C000CA	PSa	MERGECOPY
C1	00C11B04	SPDSnaoxn	–
C2	00C21884	SPDSonoxn	–
C3	00C3006A	PSxn	–
C4	00C40E04	SPDnoa	–
C5	00C50664	SPDSxoxn	–
C6	00C60708	SDPnax	–
C7	00C707AA	PSDPoaxn	–
C8	00C803A8	SDPoa	–
C9	00C90184	SPDoxn	–
CA	00CA0749	DPSDxax	–
CB	00CB06E4	SPDSaoxn	–
CC	00CC0020	S	SRCCOPY
CD	00CD0888	SDPono	–
CE	00CE0B08	SDPnao	–
CF	00CF0224	SPno	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
D0	00D00E0A	PSDnoa	–
D1	00D1066A	PSDPxoxn	–
D2	00D20705	PDSnax	–
D3	00D307A4	SPDSoaxn	–
D4	00D41D78	SSPxPDxax	–
D5	00D50CE9	DPSanan	–
D6	00D616EA	PSDPSaoxx	–
D7	00D70349	DPSxan	–
D8	00D80745	PDSPxax	–
D9	00D906E8	SDPSaoxn	–
DA	00DA1CE9	DPSDanax	–
DB	00DB0D75	SPxDSxan	–
DC	00DC0B04	SPDnao	–
DD	00DD0228	SDno	–
DE	00DE0268	SDPxo	–
DF	00DF08C8	SDPano	–
E0	00E003A5	PDSoa	–
E1	00E10185	PDSoxn	–
E2	00E20746	DSPDxax	–
E3	00E306EA	PSDPaoxn	–
E4	00E40748	SDPSxax	–
E5	00E506E5	PDSPaoxn	–
E6	00E61CE8	SDPSanax	–
E7	00E70D79	SPxPDxan	–
E8	00E81D74	SSPxDSxax	–
E9	00E95CE6	DSPDSanaxxn	–
EA	00EA02E9	DPSao	–
EB	00EB0849	DPSxno	–
EC	00EC02E8	SDPao	–
ED	00ED0848	SDPxno	–

Table 13-9 **Raster operation codes (Continued)**

Boolean function (hexadecimal)	Raster operation (hexadecimal)	Boolean function in reverse Polish	Common name
EE	00EE0086	DSo	SRCPAINT
EF	00EF0A08	SDPnoo	–
F0	00F00021	P	PATCOPY
F1	00F10885	PDSono	–
F2	00F20B05	PDSnao	–
F3	00F3022A	PSno	–
F4	00F40B0A	PSDnao	–
F5	00F50225	PDno	–
F6	00F60265	PDSxo	–
F7	00F708C5	PDSano	–
F8	00F802E5	PDSao	–
F9	00F90845	PDSxno	–
FA	00FA0089	DPo	–
FB	00FB0A09	DPSnoo	PATPAINT
FC	00FC008A	PSo	–
FD	00FD0A0A	PSDnoo	–
FE	00FE02A9	DPSoo	–
FF	00FF0062	1	WHITENESS

14 Ordering Information

The SMP8634 Secure Media Processors can be ordered in 2 distinct variations, referred to as ‘Production’ or ‘Development’ versions. The versions differ in the internal keys which are used to authenticate and decrypt the software during the bootup process. Development versions are intended to be used only during the product development process, since they do not offer the high level of security of the Production devices.

The SMP8634 devices can be distinguished as either production or development versions by looking at the Revision field on the package. If the revision level consists of a single alphabetic character (e.g. Revision A), then the device is a production version chip. If the revision level is of the form ESn, where ‘n’ is a numeric value (e.g., ES6) then the device is a development version chip.

Evaluation boards from Sigma Designs are always assembled using the Development version of the SMP8634.

Ordering Information

Table 14-1. **SMP8634 ordering information**

Part Number	Revision	Keyset	Solder Type	Package Type	Package Size	Ball Pitch	Macrovision	Temperature Range
SMP8634	Alpha[1]	Production	Pb	820BGA	35x35mm	1mm	Yes	0-70^{0}C
SMP8634-LF	Alpha[1]	Production	Pb-free	820BGA	35x35mm	1mm	Yes	0-70^{0}C
SMP8635	Alpha[1]	Production	Pb	820BGA	35x35mm	1mm	No	0-70^{0}C
SMP8635-LF	Alpha[1]	Production	Pb-free	820BGA	35x35mm	1mm	No	0-70^{0}C
SMP8634	ESn	Development	Pb	820BGA	35x35mm	1mm	Yes	0-70^{0}C
SMP8634-LF	ESn	Development	Pb-free	820BGA	35x35mm	1mm	Yes	0-70^{0}C

1. ‘Alpha’ refers to the revision level code consisting of a single alphabetic character (e.g., ‘A’) representing the current production revision.

List of Figures

5 System Block

6 Host Interface

7 Video Decoder Subsystem

8 Video Processing Subsystem

9 Audio Processing Subsystem

10 Transport Demultiplexer

11 Pin Information

12 System Specifications

13 Appendix

List of Tables

Conventions

2 Host CPU Block

3 Security Subsystem

4 DRAM Controllers

5 System Block

6 Host Interface

7 Video Decoder Subsystem

8 Video Processing Subsystem

9 Audio Processing Subsystem

10 Transport Demultiplexer

11 Pin Information

12 System Specifications

13 Appendix

14 Ordering Information

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

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

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

Europe
Sigma Designs, Inc.
+32 496 501234
+32 234 72260 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

Indonesia, Malaysia, Singapore
Convergent Systems (S) Pte. Ltd.
60 Albert Street
#11-01 Albert Complex
Singapore 189969
+65 6337-0177
+65 6336-2247 FAX
www.convergent.com.sg

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