To all our customers

Information regarding change of names mentioned within this document, to Renesas Technology Corp.

On April 1st 2003 the following semiconductor operations were transferred to Renesas Technology Corporation: operations covering microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.).
Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have all been changed to Renesas Technology Corporation.
Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.
Thank you for your understanding.

Renesas Technology Home Page: www.renesas.com

Renesas Technology Corp.
April 1, 2003

RENESAS

Renesas Technology Corp.

SuperH™ RISC engine Peripheral LSI

HD64404

Hardware Manual

HITACHI

ADE-607-042

Rev. 1.0
09/13/02
Hitachi, Ltd.

Shaping intelligence
HITACHI SEMICONDUCTOR

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

General Precautions on Handling of Product

1. Treatment of NC Pins

Note: Do not connect anything to the NC pins.
The NC (not connected) pins are either not connected to any of the internal circuitry or are they are used as test pins or to reduce noise. If something is connected to the NC pins, the operation of the LSI is not guaranteed.

2. Treatment of Unused Input Pins

Note: Fix all unused input pins to high or low level.
Generally, the input pins of CMOS products are high-impedance input pins. If unused pins are in their open states, intermediate levels are induced by noise in the vicinity, a pass-through current flows internally, and a malfunction may occur.

3. Processing before Initialization

Note: When power is first supplied, the product's state is undefined.
The states of internal circuits are undefined until full power is supplied throughout the chip and a low level is input on the reset pin. During the period where the states are undefined, the register settings and the output state of each pin are also undefined. Design your system so that it does not malfunction because of processing while it is in this undefined state. For those products which have a reset function, reset the LSI immediately after the power supply has been turned on.

4. Prohibition of Access to Undefined or Reserved Addresses

Note: Access to undefined or reserved addresses is prohibited.
The undefined or reserved addresses may be used to expand functions, or test registers may have been be allocated to these addresses. Do not access these registers; the system's operation is not guaranteed if they are accessed.

Configuration of This Manual

This manual comprises the following items:

1. General Precautions on Handling of Product
2. Configuration of This Manual
3. Preface
4. Contents
5. Overview
6. Description of Functional Modules

   System-Control Modules

   On-Chip Peripheral Modules

   The configuration of the functional description of each module differs according to the module. However, the generic style includes the following items:

   i) Feature
   ii) Input/Output Pin
   iii) Register Description
   iv) Operation
   v) Usage Note

When designing an application system that includes this LSI, take notes into account. Each section includes notes in relation to the descriptions given, and usage notes are given, as required, as the final part of each section.

7. Electrical Characteristics
8. Appendix
9. Main Revisions and Additions in this Edition (only for revised versions)

The list of revisions is a summary of points that have been revised or added to earlier versions. This does not include all of the revised contents. For details, see the actual locations in this manual.

10. Index

Preface

The HD64404 is a companion chip for the SuperH family SH-4 CPU core. It incorporates a graphic processing engine and an interface function with various network and multimedia devices that are required to configure Car Information Systems (CIS).

Target Users: This manual was written for users who will be using this LSI in the design of application systems. Users of this manual are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of this LSI to the above users.

Notes on reading this manual:

- Product names

  The following products are covered in this manual.

Product Code
HD64404BT

- In order to understand the overall functions of the chip

  Read the manual according to the contents. This manual can be roughly categorized into parts on system control functions, peripheral functions, and electrical characteristics.
- In order to understand the details of the CPU's functions of SH-4

  Read the SH-4 Programming Manual.

Rules:	Register name:	The following notation is used for cases when the same or a similar function, e.g. serial communication, is implemented on more than one channel: XXX_N (XXX is the register name and N is the channel number)
	Bit order:	The MSB (most significant bit) is on the left and the LSB (least significant bit) is on the right.
	Number notation:	Binary is B'xxxx, hexadecimal is H'xxxx, decimal is xxxx.
	Signal notation:	An overbar is added to a low-active signal: $\overline{\text{xxxx}}$

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require. http://www.hitachisemiconductor.com/

HD64404 manuals:

Manual Title	ADE No.
HD64404 Hardware Manual	This manual

Users manuals for related LSI:

Manual Title	ADE No.
SH7750 Series Hardware Manual	ADE-602-124E
SH7751 Series Hardware Manual	ADE-602-201B
SH-4 Programming Manual	ADE-602-156D

Abbreviations

ATA	AT Attachiment
ATAPI	AT Attachment Packet Interface
bpp	bit per pixcel
bps	bit per second
BMC	biphase-mark-code
CPG	Clock Pulse Generator
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
DRAM	Dynamic Random Access Memory
DSP	Digital Signal Processor
FIFO	First-In First-Out
GE	Graphic Engine
HCAN-2	Hitachi Control Area Network-2
Hi-Z	High Impedance
HSPI	Hitachi Serial Peripheral Interface
IEEE	Institute Electronic and Electronics Engineers
I^2C	Inter IC bus
INTC	Interrupt Controller
IrDA	Infrared Data Association
ISO	International Organization for Standardization
JTAG	Joint Test Action Group
LSB	Least Significant Bit
MOST	Media Oriented Systems Transport
MSB	Most Significant Bit
PCIC	Peripheral Component Interconnect Controller
PLL	Phase Locked Loop
Q2SD	Quick 2D Graphics Renderer with Synchronous DRAM Interface
RAM	Random Access Memory
RGB	Red Green Blue
SPDIF	Sony Philips Digital Interface Format
SPI	Serial Peripheral Interface
SSI	Serial Sound Interface
SRAM	Static Random Access Memory

T.B.D	To Be Determined
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VIO	Video I/O
YUV	Y-signal, U-signal, V-signal

Contents

Figures

Section 10 Display Out Module

Section 11 GE for HD64404

Section 12 Color Space Converter

Section 13 Audio Codec Interface

Section 14 Serial Sound Interface (SSI) Module

Section 15 Hitachi SPDIF Interface

Section 16 Hitachi I^2C Interface

Section 17 Hitachi Serial Peripheral Interface

Section 18 ATAPI

Section 19 HCAN-2 Module

Section 20 Most Interface Module

Section 21 UART

Section 22 IrDA

Section 23 USB Function

Section 24 USB HOST

Section 25 Interrupt Input

HITACHI

Tables

Section 1 Overview

1.1 General Description

This device is a companion to the Super H RISC family of processors from Hitachi. It is aimed at the Car Information System market and combines multiple functions needed in this type of system. These features include a high performance 2D-graphics engine, video input and communication peripherals.

The device will support automotive environmental conditions e.g. –40 → + 85°C and will be designed to minimize power consumption to reduce system costs.

This device will be supported with WinCE, VXWorks and QNX drivers among other operating systems.

1.2 Architecture Overview

The architecture consists of a dual bus structure, the register bus and the pixel bus. The majority of the modules will be connected to the register bus for control and mid speed data transfer. Devices responsible for pixel data or high bandwidth data will be connected to the pixel bus, which will allow DMA transfer of data to and from the graphics memory. The register bus will support DMA between any two peripherals and to the system memory or graphics memory.

A number of pins on this device are multi-functioned and that is shown on the diagram. In addition to having a major different function, some pins can also be configured as GPIO.

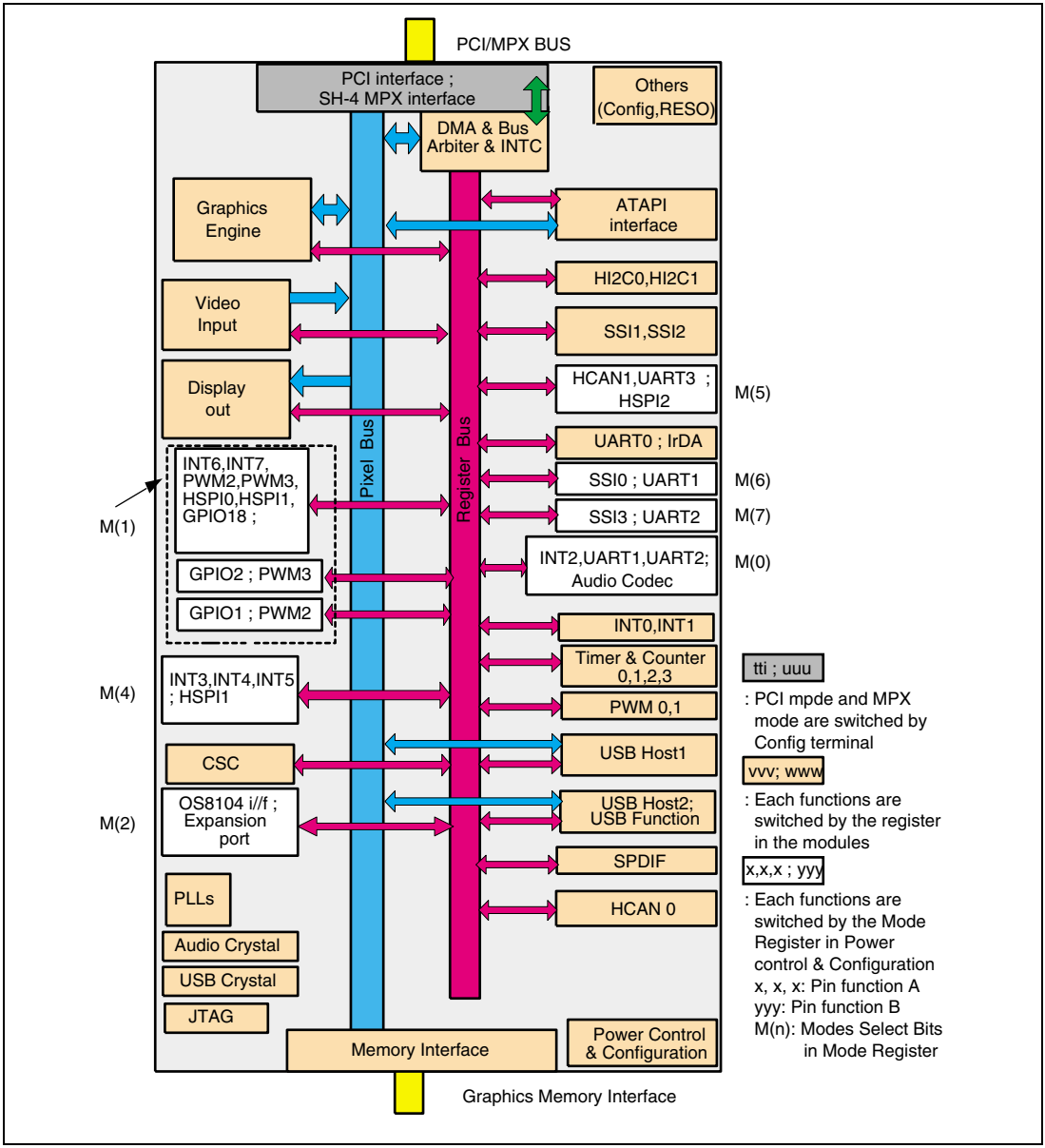


Figure 1.1 HD64404 Block Diagram

1.2.1 Mode Register Configuration

Mode Register	Group 1			Group 2		Group 3	Group 4
Mode Bit	M(0)	M(6)	M(7)	M(1)	M(4)	M(2)	M(5)
0: Function A	INT2	SSI0	SSI3	INT6	INT3	OS8104 I/F	HCAN1
	UART1			INT7	INT4		UART3
	UART2			HSPI0	INT5		
				HSPI1			
				PWM2			
				PWM3			
				GPIO1			
				GPIO2			
				GPIO18			
1: Function B	AC	UART1	UART2	PWM2	HSPI1	Expansion Port	HSPI2
				PWM3			

- Group 1 Combinations

Group 1	M(0)	M(6)	M(7)
Mode Bit	0	0	0
	1	0	0
	1	0	1
	1	1	0
	1	1	1

- Group 2 Combinations

Group 2	M(1)	M(4)
Mode Bit	0	0
	1	0
	1	1

1.2.2 Peripheral Module Register Configuration

Peripheral Module Register Bit	Group 5	Group 6	Group 7
	UART Module	USB Module	SSI Module
Function 1	UART0	USB Host 2	SSI2
Function 2	IrDA	USB Function	SSI2

1.2.3 Config Pin Configuration

Config Pin	Group 8
1: CPU I/F 1	PCI Bus
0: CPU I/F 2	SH-4 Multiplex(MPX) Bus

HD64404 has $5 \times 3 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 960$ Configuration Combinations in total.

1.3 Features

1.3.1 System Interface

- Open architecture connection to the PCI or SH-4 MPX interface
- Multiplexed address/data interface (Support included for SH7750 & SH7751)
- MPX operates in slave mode
- Initiator/Target PCI configuration
- Complete memory-mapped system
- The 32-bit MPX bus will operate up to 100 MHz
- MPX bus supports two chip selects which in turn supports up to 128-MBytes of memory mapped address space.
- PCI interface supports 2 channels of DMAC transferring data between System Memory and Graphics Memory.
- SH-4 MPX interface support 1 channel of DMA transfer using SH-4 DMAC from System Memory to Graphics Memory.
- System clock generated from SH-4 clock (CKIO) or PCI clock (PCI_CLK).

1.3.2 Pixel Bus

- Synchronous high speed bus
- Synchronized to PCI or SH-4 MPX bus to reduce latency where possible
- Up to 100 MHz operation at 32 bits
- Sideband arbitration (i.e. arbitration happens in parallel with transfers)
- Combined fixed and round robin arbitration

1.3.3 Register Bus

- Synchronous mid speed bus
- Synchronized to PCI or SH-4 MPX bus where possible
- Up to 50-MHz operation (2 cycle access)
- Hitachi IP standard 32-bit bus
- Sideband arbitration

1.3.4 DMAC

- DMA transfers between peripherals on the register bus
- DMA transfer from/to peripherals across PCI bus
- DMA transfer from/to peripherals to/from SDRAM
- 16 channels of DMA
- End of transfer interrupt
- Continuous Data Transfer Mode Support.
- Local RAM storage for each peripheral to allow burst transfers on pixel bus and PCI/MPX bus.

1.3.5 Graphics Engine

- Up to 100 MHz operation
- Pre/Post hardware clipping
- Linear addressing: Supported through translation in the MPX I/F module, PCI I/F module, and ATAPI module. The architecture for the graphics memory is tiled for increased performance, but this is transparent to the software.

Q2SD Rendering Unit (Q2SD/RU):

- Software backward compatibility with Q2SD
- Drawing commands: 4-vertex surface drawing, line drawing, work surface drawing, and work line drawing

- Color representation: source: 1/8/16 bit/pixel; destination: 8/16 bit/pixel; work: 1 bit/pixel
- Zooming: Supported through the arbitrary polygon copy.
- Patterns: Source image can be copied to destination image that is larger. The source image is copied multiple times replicating in the horizontal and vertical directions.
- Rotation: The relationship between the source vertices and the destination vertices should be the same but with the destination vertices rotated. This does not include filtering of the image.
- Binary to color expansion: Converts an image described as 1 bit/pixel into a color image where the color is foreground color and background color.
- Bresenham line drawing: accurate 8-direction line drawing

2D Graphics Engine (2DGE):

- BitBLT (Source + Destination => Destination): Allows the logical combination of the source area with the destination.
- Raster operations: 16 logical operations of source + destination
- Anti-aliased fonts using 16 levels of alpha blending
- Alpha value to color expansion: Converts an image described as a 4 bit/pixel intensity value into a color image where the resultant pixels are a proportions of the foreground color and background image as where the proportion is defined by the 4-bit alpha value.

1.3.6 Video Input

- ITU-R BT.656 interface at 27-MHz video input clock
- Color space conversion and dithering from 4:2:2YCrCb to RGB 5:6:5
- Interlace mode: Odd field capture, Even field capture, Odd field and Even filed capture, Full Interlace (Both the odd and even field are processed as single frame.)
- Scaling down to any resolution based on sub-pixel interpolator
- 9-tap horizontal programmable multi-rate decimation filter
- 2-tap vertical interpolator for arbitrary scaling down or scaling up by up to a factor of 3
- Triple frames support for frame rate conversion
- Single capture and continuous capture are supported.

1.3.7 Display Output

- R (6 bits), G (6 bits), B (6 bits) digital interface
- Dual planes with additional PIP on background plane: Two planes can be combined in display output. In addition part of the background plane can be replaced by a third plane, normally video, for creating an alternative method for PIP.
  PIP: Picture in Picture

- Alpha blending of foreground (FG) and background (BG) planes
  FG/BG = 16bpp/16bpp, 16bp/8bpp and 8bpp/16bpp
- Chroma-keying on foreground plane: This allows a foreground image to be overlaid on a background image on display output, where a color is defined for the foreground that will be transparent and allow the background to be shown. All other colours will be alpha blended with constant alpha value. This allows for menu systems.
- 8/16 bits per pixel on FIG, BG and PIP planes.
- Dual 64 × 64 or 32 × 32 hardware cursors with 8 bpp.
- Scrolling on Background plane: A window for the background plane can be defined as a subset of a greater canvas to allow the moving of this window over a larger scene.
- Wrapping: When scrolling and window moves outside background then data is fetched from opposite side of background
- Automatic double buffering switching on graphics planes and triple buffering on Videoplanes.
- Support for up to 854 × 480-display size (Note: Refer to section 1.8.2 Pixel Bus). The counters will support images up to 1024 × 768 but the bandwidth to a 32-bit memory system will restrict the screen size support to this resolution, but this is also dependent on the screen refresh rate.
- Refresh rates programmable for performance and screen size optimization.
- VSYNC, HSYNC, Display Enable
- Programmable display DOT clock (DOT_CLK)

1.3.8 CSC (Color Space Converter)

- Input: YUV data format or DELTA YUV data format
- Output: RGB data (R: 5bit, G: 6bit, B: 5bit) format
- The Color Space Converter is used to convert YUV data into RGB format line by line.
- This function is available only for DMA transfer.

1.3.9 SDRAM Interface

- 32-bit interface running at up to 100 MHz dependent on system configuration.
- Multi-bank activation for reduced pre-charge and activation delays.
- Overlapping SDRAM command access. This can be disabled to improve latency at the expense of bandwidth in UMA systems.
- UMA supported (depending on system requirements)
- Support up to 128 Mbytes of SDRAM
- Operates synchronously to the system clock (= Pixel bus clock)

1.3.10 Interrupt Priority

Takes a single interrupt from each block and based on each assigned priority, an interrupt is generated and the highest priority interrupt number value stored.

- Support up to 28 interrupts
- Programmable priorities
- Programmable masks
- Each block can have multiple sources generating input to interrupt controller
- Standby mode support via Interrupt Input to raise system processor interrupt even when HD64404 clock stops.

1.3.11 Serial Sound Interface (SSI)

- 4-channel bi-directional SSI (maximum)
- All support multi-channel and compressed data
- Programmable frame size
- Two SSI channels may be configured as GPIO
- Supports the Philips format

1.3.12 Hitachi I^2C Interface

- 2-channel (maximum)
- Master/Slave
- 7-or-10 bit compatible master
- Fast I^2C up to 400 Kbits/sec
- Supports the Philips I^2C bus interface
- Programmable clock derived from system clock(= Register bus clock)

1.3.13 Hitachi Serial Peripheral Interface (HSPI)

- 3-channels (maximum)
- Configurable in either Master mode or Slave mode
- Programmable data rate

1.3.14 Hitachi S/PDIF Interface

- Separate transmitter and receiver
- Supports the IEC 60958 communications standard (Stereo and Consumer use modes only).
- Receiver automatically detects IEC 61937 compressed mode data

1.3.15 Audio Codec

- Digital interface to a single AC97 version 2.1 Audio Codec.
- PIO from status slots 1 and 2 of the Rx frame.
- PIO to command slots 1 and 2 of the Tx frame.
- PIO from data slots 3 and 4 of the Rx frame.
- PIO to data slots 3 and 4 of the Tx frame.
- Selectable 16-or-20 bit DMA from data slots 3 and 4 of the Rx frame.
- Selectable 16-or-20 bit DMA to data slots 3 and 4 of the Tx frame.
- Supports variable sample rates by qualifying slot data with Tag bits and responding to Rx frame slot request bits for the Tx frame.
- Interrupts can be generated for data ready/required and overrun/underrun.
- 12.288 MHz data clock input

1.3.16 USB Host and Function Interface

- Dual channel
- Support for either 2 Host ports or a combination of 1 Host port and 1 Function port.
- Supports 1.5Mbits/s and 12Mbits/s data transfer rates
- USB version 1.1. for Host and Function
- OHCI version 1.0 support
- 48-MHz clock via either external clock input or X'tal oscillation
- Transmit and receive buffers are in SDRAM memory connected to HD64404
- USB transceiver on-chip

1.3.17 HCAN2

- Dual channels (Maximum)
- Supports CAN Specification 2.0A and 2.0B
- Standard Data and Remote Frames (11-bit identifier).
- Extended Data and Remote Frames (29-bit identifier).
- 32 independent message buffers, using standard (11 bits) or extended (29 bits) identifier format.

- 31 mailboxes, programmable for the direction transmit or receive.
- 1 receive-only mailbox.
- Acceptance filtering by identifier:
  - Standard Message Identifier.
  - Extended Message Identifier.
- Sleep mode for low power consumption.
- Programmable Local Acceptance Filter Mask (standard and extended identifier) supported by all Mailboxes.
- Programmable CAN data rate up to 1 Mbit/s.
- Transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications.
- Data buffer access without handshake requirement.
- 16-bit free running timer with flexible clock sources and pre-scaler, 3 Timer Compare Match Registers, CAN-ID Compare Match, 2 Input Capture Registers, Drift Correction Registers, Local Offset Register
- 4-bit Basic Cycle Counter for Time Trigger Transmission
- Timer Compare Match Registers with interrupt generation + Timer counter clear/set capability to support schedule-monitoring of transmit/receive, one-shot transmission at a specific time, etc
- CAN-ID Compare Match with Timer Clear/Set + Input Capture Register Disable when receiving a specific CAN Frame
- Input Capture Registers used for TimeStamp and Global Synchronisation on a CAN system, interacting with SOF/EOF of CAN Frame and CAN-ID Compare Match
- Flexible TimeStamp for both transmission and reception (stamp-timing programmable) supported
- Time-Trigger Transmission, Periodic Transmission supported on top of Event Trigger Transmission
- Timer Counter and Basic Cycle value can be embedded into a CAN frame and transmitted

1.3.18 UART

- 4 channels (maximum)
- Asynchronous serial controller.
- Programmable baud rate.
- Programmable start/stop and parity bits.
- One UART is multiplexed with IrDA port

1.3.19 IrDA

- Supported by configuring one channel of the UART.
- SIR (Slow IrDA: 115.2 Kbps) compatible.
- Independent transmit and receive unit.

1.3.20 OS8104 Interface or Expansion Bus

- Can be configured for MOST or SRAM type interface
- Direct connection to the OS8104
- Unsupervised hardware flow control to the OS8104
- Allows the connection of additional peripherals in SRAM mode

1.3.21 ATAPI

- One channel support ATA/ATAPI-4
- Support up to 2 devices (master/slave)
- 3.3V I/O interface
- PIO mode 0 to 4, Multiword DMA mode 0 to 2, Ultra DMA mode 0 to 2 support

1.3.22 GPIO

- Maximum of 60 GPIOs
- 3 dedicated GPIOs

1.3.23 Interrupt Input

Converts input signals to a single interrupt to the central interrupt controller by detecting edges or levels (8 inputs)

- Level or edge sensitive
- High/Low active level
- Positive/negative active edge
- Programmable mask
- Standby mode support to deliver interrupt even when HD64404 clock stops

1.3.24 Timer/Counter

- 32-bit free running timer (FRT)
- 4 input captures/output compares
- Positive and negative edge configurable
- Programmable FRT clock
- I/O pins for all timers can be used as GPIO

1.3.25 PWM

- 4 channels of PWM
- Programmable source clock frequency giving cycle time from 30ns with PCI bus or 20ns for MPX, to 2 minutes.
- Programmable high value and programmable cycle duration (8 bits)

1.3.26 PLL Clock Generation

- System clock generated from SH-4 clock (CKIO) or PCI clock (PCI_CLK)
- Programmable display DOT clock (DOT_CLK)

1.3.27 Crystal Oscillators

- 512 times Audio sampling frequency (24.576 MHz, 22.5792 MHz) that is used for SSI and SPDIF.
- USB clock (48 MHz)

1.3.28 Power Management

- Individual power down of each module via software.
- There are two methods: Memory disabled, Self refresh, Self refresh can be entered by setting the SR bit of Memory control register

1.4 Pin Modes

Table 1.1 Pin Modes

No.	Package No.	Function A		Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
1	AB1	SDRAM I/f (32 bit)	SD_DATA(31)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
2	AA2		SD_DATA(30)	B					B		Input (A)	Low	Input(A)	Low	Input	Low			
3	Y2		SD_DATA(29)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
4	W4		SD_DATA(28)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
5	W2		SD_DATA(27)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
6	V4		SD_DATA(26)	B					B		Input (A)	Low	Input(A)	Low	Input	Low			
7	U4		SD_DATA(25)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
8	U2		SD_DATA(24)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
9	U3		SD_DATA(23)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
10	V1		SD_DATA(22)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
11	W1		SD_DATA(21)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
12	W3		SD_DATA(20)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
13	Y1		SD_DATA(19)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
14	Y3		SD_DATA(18)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
15	AA3		SD_DATA(17)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
16	AB2		SD_DATA(16)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
17	C2		SD_DATA(15)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
18	D2		SD_DATA(14)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
19	E4		SD_DATA(13)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
20	E1		SD_DATA(12)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
21	F2		SD_DATA(11)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
22	G4		SD_DATA(10)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
23	G1		SD_DATA(09)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
24	H2		SD_DATA(08)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
25	H1		SD_DATA(07)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
26	H3	SDRAM I/f (32 bit) SD_DATA(06)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
27	G3	SD_DATA(05)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
28	F1	SD_DATA(04)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
29	F3	SD_DATA(03)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
30	E2	SD_DATA(02)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
31	D1	SD_DATA(01)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
32	C1	SD_DATA(00)	B					B		Input (A)	Low	Input (A)	Low	Input	Low			
33	K1	SD_ADDR(12)	O					B		Output (A)	0	Output (A)	0	Output	0			
34	L1	SD_ADDR(11)	O					B		Output (A)	0	Output (A)	0	Output	0			
35	L2	SD_ADDR(10)	O					B		Output (A)	0	Output (A)	0	Output	0			
36	L3	SD_ADDR(09)	O					B		Output (A)	0	Output (A)	0	Output	0			
37	L4	SD_ADDR(08)	O					B		Output (A)	0	Output (A)	0	Output	0			
38	M2	SD_ADDR(07)	O					B		Output (A)	0	Output (A)	0	Output	0			
39	M3	SD_ADDR(06)	O					B		Output (A)	0	Output (A)	0	Output	0			
40	M4	SD_ADDR(05)	O					B		Output (A)	0	Output (A)	0	Output	0			
41	N2	SD_ADDR(04)	O					B		Output (A)	0	Output (A)	0	Output	0			
42	P1	SD_ADDR(03)	O					B		Output (A)	0	Output (A)	0	Output	0			
43	P2	SD_ADDR(02)	O					B		Output (A)	0	Output (A)	0	Output	0			
44	P3	SD_ADDR(01)	O					B		Output (A)	0	Output(A)	0	Output	0			
45	P4	SD_ADDR(00)	O					B		Output (A)	0	Output(A)	0	Output	0			
46	R2	$\overline{RAS}$	O					O		Output (A)	1	Output(A)	1	Output	1			
47	T2	$\overline{CAS}$	O					O		Output (A)	1	Output(A)	1	Output	1			
48	R1	$\overline{WE}$	O					O		Output (A)	1	Output(A)	1	Output	1			
49	T3	$\overline{CS}$	O					O		Output (A)	1	Output(A)	1	Output	1			
50	K2	$\overline{BA0}$	O					O		Output (A)	0	Output(A)	0	Output	0			
51	J1	$\overline{BA1}$	O					O		Output (A)	0	Output (A)	0	Output	0			
52	T1	SD_CLK	O					B		Output (A)	Clock	Output (A)	Clock	Output	Clock			
53	J2	SD_CKE	O					O		Output (A)	1	Output (A)	1	Output	1			
54	U1	DQM(3)	O					B		Output (A)	1	Output (A)	1	Output	1			
55	T4	DQM(2)	O					B		Output (A)	1	Output (A)	1	Output	1			
56	J4	DQM(1)	O					B		Output (A)	1	Output (A)	1	Output	1			
57	J3	DQM(0)	O					B		Output (A)	1	Output (A)	1	Output	1			

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
58	AE2	Video input VI_Data(7)	I				GPIO (63)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
59	AE1	VI_Data(6)	I				GPIO (62)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
60	AC4	VI_Data(5)	I				GPIO (61)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
61	AC3	VI_Data(4)	I				GPIO (60)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
62	AC2	VI_Data(3)	I				GPIO (59)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
63	AC1	VI_Data(2)	I				GPIO (58)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
64	AB3	VI_Data(1)	I				GPIO (57)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
65	AA4	VI_Data(0)	I				GPIO (56)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
66	AD2	VI_Clk	I					I		Input (A)	Clock/Low	Input (A)	Clock/Low	Input	Clock/Low			
67	AD20	Display out digital DO_Data(17) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
68	AE20	DO_Data(16) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
69	AF20	DO_Data(15) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
70	AC21	DO_Data(14) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
71	AD21	DO_Data(13) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
72	AE21	DO_Data(12) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
73	AF21	DO_Data(11) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
74	AD22	DO_Data(10) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
75	AE22	DO_Data(09) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
76	AF22	DO_Data(08) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
77	AF23	DO_Data(07) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
78	AC23	DO_Data(06) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				

No.	Package No.		Function A	Signal Type: A		Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
79	AE24	Display out digital	DO_Data(05) (Tri)	O						O		High-Z (A)		High-Z (A)		High-Z				
80	AE25		DO_Data(04) (Tri)	O						O		High-Z (A)		High-Z (A)		High-Z				
81	AF25		DO_Data(03) (Tri)	O						O		High-Z (A)		High-Z (A)		High-Z				
82	AF26		DO_Data(02) (Tri)	O						O		High-Z (A)		High-Z (A)		High-Z				
83	AE26		DO_Data(01) (Tri)	O						O		High-Z (A)		High-Z (A)		High-Z				
84	AD24		DO_Data(00) (Tri)	O						O		High-Z (A)		High-Z (A)		High-Z				
85	AD25		$\overline{\text{DO\_VSYNC}}$	B						B		Input (A)	High	Input (A)	High	Input	High			
86	AB24		$\overline{\text{DO\_HSYNC}}$	B						B		Input (A)	High	Input (A)	High	Input	High			
87	AC25		DO_DEN (Tri)	O						O		High-Z (A)		High-Z (A)						
88	AD26		DOT_CLK	B						B		Input (A)	Clock	Input (A)	Clock	Input	Clock			
89	F26	PCI	AD(31)	B	SH4 I/f	D(00)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
90	G26		AD(30)	B		D(15)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
91	H23		AD(29)	B				Config		B		Input (A)	Low			Input	Low			
92	H24		AD(28)	B		D(01)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
93	H25		AD(27)	B		D(14)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
94	H26		AD(26)	B		D(02)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
95	J23		AD(25)	B		D(13)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
96	J26		AD(24)	B		D(03)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
97	K25		AD(23)	B		D(11)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
98	K24		AD(22)	B		D(04)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
99	K26		AD(21)	B		D(05)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
100	L25		AD(20)	B		D(10)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
101	L26		AD(19)	B		D(06)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
102	M23		AD(18)	B				Config		B		Input (A)	Low			Input	Low			
103	M24		AD(17)	B		D(09)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
104	M25		AD(16)	B		D(07)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
105	T23		AD(15)	B		D(21)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
106	U26		AD(14)	B		D(26)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	

No.	Package No.		Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
107	U25	PCI	AD(13)	B	D(20)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
108	V26		AD(12)	B	D(27)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
109	V25		AD(11)	B	D(19)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
110	V24		AD(10)	B	D(28)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
111	V23		AD(09)	B	D(18)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
112	W26		AD(08)	B	D(17)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
113	W23		AD(07)	B			Config		B		Input (A)	Low			Input	Low			
114	Y25		AD(06)	B	D(29)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
115	Y24		AD(05)	B	D(30)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
116	AA26		AD(04)	B	D(31)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
117	AA25		AD(03)	B	D(61)	I	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
118	Y23		AD(02)	B	D(62)	I	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
119	AB26		AD(01)	B	D(63)	I	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
120	AC26		AD(00)	B	$\overline{\text{IRL}}$(O/D)	O	Config		B		Input (A)	Low	High-Z (B)		Input	Low	High-Z		
121	K23		C/$\overline{\text{BE}}$(3)	B	D(12)	B	Config		B		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
122	M26		C/$\overline{\text{BE}}$(2)	B	D(08)	B	Config		B		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
123	T24		C/$\overline{\text{BE}}$(1)	B	D(25)	B	Config		B		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
124	Y26		C/$\overline{\text{BE}}$(0)	B	D(16)	B	Config		B		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
125	T25		PAR	B	D(22)	B	Config		B		Input (A)	Low	Input (B)	Low	Input	Low	Input	Low	
126	N25		$\overline{\text{FRAME}}$	B	$\overline{\text{BS}}$	I	Config		B		Input (A)	High	Input (B)	High	Input	High	Input	High	
127	P26		$\overline{\text{TRDY}}$	B	RD/$\overline{\text{WR}}$	I	Config		B		Input (A)	High	Input (B)	High	Input	High	Input	High	
128	P25		$\overline{\text{IRDY}}$	B	$\overline{\text{FRAME}}$	I	Config		B		Input (A)	High	Input (B)	High	Input	High	Input	High	
129	R25		$\overline{\text{STOP}}$	B	D(23)	B	Config		B		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
130	R26		$\overline{\text{DEVSEL}}$	B	$\overline{\text{RDY}}$ (TriState)	O	Config		B		Input (A)	High	High-Z (B)		Input	High	High-Z		
131	E25		IDSEL	I	$\overline{\text{SH4\_CSB}}$	I	Config		I		Input (A)	Low	Input (B)	High	Input	Low	Input	High	
132	T26		$\overline{\text{PERR}}$	B	D(24)	B	Config		B		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
133	G23		$\overline{\text{SERR}}$(O/D)	O	$\overline{\text{DREQ}}$	O	Config		B		High-Z (A)		Output (B)	1	Input	High	Output	1	
134	F24		$\overline{\text{REQ}}$	O	DACK	I	Config		B		Output (A)	1	Input (B)	Low	Input	High	Input	Low	
135	F25		$\overline{\text{GNT}}$	I	DRAK	I	Config		I		Input (A)	High	Input (B)	Low	Input	High	Input	Low	
136	N24		PCI_CLK	I	CKIO	I	Config		I		Input (A)	Clock	Input (B)	Clock	Input	Clock	Input	Clock	
137	E26		$\overline{\text{RST}}$	I	$\overline{\text{RST}}$	I	Config		I		Input (A)	High	Input (B)	High	Input	High	Input	High	
138	F23		$\overline{\text{INTA}}$(O/D)	O	$\overline{\text{SH4\_CSA}}$	I	Config		B		Input (A)	High	Input (B)	High	Input	High	Input	High	

No.	Package No.		Function A	Signal Type: A		Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
139	AF11	SSI(0)	SSI0_FSY	B	UART(1)	UART1_TXD	O	M(6)*1		B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		*1: M(0) is prior to M(6) for UART1
140	AD12		SSI 0_SCK	B						B		Input (A)	Low	Input (A)	Low	Input	Low			M(6) can be 1 only when M(0) is 1
141	AC12		SSI 0_SDATA	B		UART1_RXD	I	M(6)*1		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	High	
142	AD11	SSI(1)	SSI 1_FSY	B					GPIO (19)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
143	AE11		SSI 1_SCK	B					GPIO (20)	B		Input(G)	High/Low	Input (G)	High/Low	Input	Low			
144	AC11		SSI 1_SDATA	B						B		Input (A)	Low	Input (A)	Low	Input	Low			
145	AE10	SSI(2)	SSI 2_FSY	B					GPIO (21)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
146	AF10		SSI 2_SCK	B					GPIO (22)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
147	AD10		SSI 2_SDATA	B						B		Input (A)	Low	Input (A)	Low	Input	Low			
148	AF12	SSI(3)	SSI 3_FSY	B	UART(2)	UART2_RXD	I	M(7)*2		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	High	*2: M(0) is prior to M(7) for UART2
149	AF13		SSI 3_SCK	B						B		Input (A)	Low	Input (A)	Low	Input	Low			M(7) can be 1 only when M(0) is 1
150	AE12		SSI 3_SDATA	B		UART2_TXD	O	M(7)*2		B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
151	AF8	I2C(0)	I2C0_SCL	B					GPIO (23)	B		Input (G)	High/Low	Input (G)	High/Low	Input	High			
152	AE8		I2C0_SDA	B					GPIO (24)	B		Input (G)	High/Low	Input (G)	High/Low	Input	High			
153	AF7	I2C(1)	I2C1_SCL	B						B		Input (A)	High	Input (A)	High	Input	High			
154	AD8		I2C1_SDA	B						B		Input (A)	High	Input (A)	High	Input	High			
155	AF6	CAN(0)	CAN0_RX	I						I		Input (A)	Low	Input (A)	Low	Input	Low			
156	AE6		CAN0_TX	O						O		Output (A)	1	High-Z (A)		High-Z				
157	AD7		$\overline{\text{CAN0\_NERR}}$	I						I		Input (A)	High	Input (A)	High	Input	High			
158	AE7						B		GPIO (51)			Input (G)	High/Low	Input(G)	High/Low	Input	Low			
159	AD6	CAN(1)	CAN1_RX	I	SPI(2)-part of	SPI2_CLK	B	M(5)	GPIO (48)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low	Input	Clock	
160	AE5		CAN1_TX	O		SPI2_MISO	I	M(5)	GPIO (49)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low	Input	Low	

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
161	AF5	CAN(1)	$\overline{\text{CAN1\_NERR}}$	I				GPIO (50)	B		Input (G)	High/Low	Input (G)	High/Low	Input	High			
162	C11	SPDIF transmitter	SPDIF_OUT	O				GPIO (28)	B		Input (G)	High/Low	Input (G)	High/Low	High-Z				
163	D11	SPDIF receiver	SPDIF_IN	I				GPIO (29)	B		Input (G)	High/Low	Input (G)	High/Low	Input	Low			
164	AF1	UART(0)	UART0_RXD	I	IRDA	IRDA_RXD	I	(internal)	GPIO (30)	B		Input (G)	High/Low	Input (G)	High/Low	Input	High	Input	High	
165	AD3		UART0_TXD	O		IRDA_TXD	O	(internal)	GPIO (31)	B		Input (G)	High/Low	Input (G)	High/Low	High-Z		High-Z		
166	AF2	UART(1)	UART1_RXD	I	AC	$\overline{\text{AC\_RES}}$	O	M(0)		B		Input (A)	High	Input (A)	High	Input	High	Output	1	
167	AE3		UART1_TXD	O		AC_SYNC	O	M(0)		O		High-Z (A)		High-Z (A)		High-Z		Output	0	
168	AD5	UART(2)	UART2_RXD	I		AC_SDATA_IN	I	M(0)		I		Input (A)	High	Input (A)	High	Input	High	Input	Low	
169	AE4		UART2_TXD	O		AC_SDATA_OUT	O	M(0)		O		High-Z		High-Z		High-Z		Output	0	
170	AF3	JTAG	$\overline{\text{TRSTN}}$	I						I		Input (A)	High (pull-up)	Input (A)	High(pull-up)	Input	High(pull-up)			
171	AF4	UART(3)	UART3_RXD	I	SPI(2)-part of	SPI2_SIMO (Tri)	O	M(5)		B		Input (A)	High	Input (A)	High	Input	High	High-Z		
172	AC6		UART3_TXD	O		$\overline{\text{SPI2\_CS}}$	B	M(5)		B		High-Z		Input (A)		High-Z		Input	High	
173	A1	GPIO	GPIO(2)	B	MIXED	PWM(3)	O	M(1)	GPIO (2)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low or High	Output	1	
174	B1		GPIO(1)	B		PWM(2)	O	M(1)	GPIO (1)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low or High	Output	1	
175	D8		GPIO(0)	B	SPI(1)-part of	SPI1_SIMO (Tri)	O	M(4)*3	GPIO (0)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low or High	High-Z		*3: M(1) is prior to M(4) for SPI1
176	D7		INT(7)	I					GPIO (4)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	High-Z		M(4) can be 1 only when M(1) is 1
177	A6		INT(6)	I					GPIO (3)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	High-Z		
178	C8		INT(5)	I	SPI(1)-part of	SPI1_MISO	I	M(4)*3	GPIO (7)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	Input	Low	
179	A8		INT(4)	I		SPI1_CLK	B	M(4)*3	GPIO (6)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	Input	Clock	
180	B8		INT(3)	I		$\overline{\text{SPI1\_CS}}$	B	M(4)*3	GPIO (5)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	Input	High	
181	A7		INT(2)	I	AC-part of	AC_BIT_CLK	I	M(0)	GPIO (27)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	Input	Clock	

No.	Package No.		Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
182	B7	GPIO	INT(1)	I				GPIO (26)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
183	C7		INT(0)	I				GPIO (25)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
184	A10		TIMER/CTR(3)	B				GPIO (11)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
185	B10		TIMER/CTR(2)	B				GPIO (10)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
186	C10		TIMER/CTR(1)	B				GPIO (9)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
187	D10		TIMER/CTR(0)	B				GPIO (8)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
188	C3		PWM(1)	O				GPIO (13)	B		Input (G)	Low or High	Input (G)	Low or High	Output	1			
189	D3		PWM(0)	O				GPIO (12)	B		Input (G)	Low or High	Input (G)	Low or High	Output	1			
190	B6		PWM(3)	O					O		Output (A)	1	Output (A)	1	Output	1	High-Z		
191	C6		PWM(2)	O					O		Output (A)	1	Output (A)	1	Output	1	High-Z		
192	A5	SPI(0)	SPI0_SIMO (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z		High-Z		
193	C5		SPI0_MISO	I					B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
194	D5		SPI0_CLK	B					B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
195	B4		$\overline{\text{SPI0\_CS}}$	B					B		Input (A)	High	Input (A)	High	Input	High	High-Z		
196	B3	SPI(1)	SPI1_SIMO (Tri)	O				GPIO (14)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	High-Z		
197	C4		SPI1_MISO	I				GPIO (15)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	High-Z		
198	A3		SPI1_CLK	B				GPIO (16)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	High-Z		
199	A4		$\overline{\text{SPI1\_CS}}$	B				GPIO (17)	B		Input (G)	Low or High	Input (G)	Low or High	Input	High	High-Z		
200	A2		GPIO(18)	B				GPIO (18)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low	Input	High	
201	D21	ATAPI	AT_DSD(15)	B				GPIO (47)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
202	B20		AT_DSD(14)	B				GPIO (46)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			

No.	Package No.		Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
203	A19	ATAPI	AT_DSD(13)	B				GPIO (45)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
204	C19		AT_DSD(12)	B				GPIO (44)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
205	A17		AT_DSD(11)	B				GPIO (43)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
206	C17		AT_DSD(10)	B				GPIO (42)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
207	B16		AT_DSD(9)	B				GPIO (41)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
208	D16		AT_DSD(8)	B				GPIO (40)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
209	C16		AT_DSD(7)	B				GPIO (39)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
210	D17		AT_DSD(6)	B				GPIO (38)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
211	B17		AT_DSD(5)	B				GPIO (37)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
212	D19		AT_DSD(4)	B				GPIO (36)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
213	B19		AT_DSD(3)	B				GPIO (35)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
214	C20		AT_DSD(2)	B				GPIO (34)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
215	A20		AT_DSD(1)	B				GPIO (33)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
216	C21		AT_DSD(0)	B				GPIO (32)	B		Input (G)	Low or High	Input (G)	Low or High	Input	Low			
217	B24		AT_DSA(2) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
218	A24		AT_DSA(1) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
219	C24		AT_DSA(0) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
220	B23		$\overline{\text{AT\_DMACK0}}$ (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
221	B21		AT_DMARQ0	IS					IS		Input (A)	Low	Input (A)	Low	Input	Low			
222	A25		$\overline{\text{AT\_DCS}}$(1) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
223	A26		$\overline{\text{AT\_DCS}}$(0) (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
224	A21		$\overline{\text{AT\_DIOW}}$ (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
225	C22		$\overline{\text{AT\_DIOR}}$ (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
226	B22	ATAPI AT_DCHRDY0	IS					IS		Input (A)	High	Input (A)	High	Input	High			
227	A23	AT_DIRQ1	IS					IS		Input (A)	Low	Input (A)	Low	Input	Low			
228	A16	$\overline{\text{AT\_RESET}}$ (Tri)	O					O		High-Z (A)		High-Z (A)		High-Z				
229	AC14	OS8104 interface MPAD(1)	O	Expansion bus EX_ADDR(1)	O	M(2)		O		High-Z (A)		High-Z (A)		High-z		High-Z		
230	AF14	MPAD(0)	O	EX_ADDR(0)	O	M(2)		O		High-Z (A)		High-Z (A)		High-z		High-Z		
231	AC15	MDATA(7)	B	EX_DATA(7)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
232	AD15	MDATA(6)	B	EX_DATA(6)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
233	AE15	MDATA(5)	B	EX_DATA(5)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
234	AF15	MDATA(4)	B	EX_DATA(4)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
235	AC16	MDATA(3)	B	EX_DATA(3)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
236	AD16	MDATA(2)	B	EX_DATA(2)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
237	AE16	MDATA(1)	B	EX_DATA(1)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
238	AF16	MDATA(0)	B	EX_DATA(0)	B	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
239	AD17	$\overline{\text{MRD}}$	O	$\overline{\text{EX\_RD}}$	O	M(2)		O		Output (A)	1	Output (A)	1	Output	1	Output	1	
240	AC17	$\overline{\text{MWR}}$	O	$\overline{\text{EX\_WR}}$	O	M(2)		O		Output (A)	1	Output (A)	1	Output	1	Output	1	
241	AF17	$\overline{\text{MAINT}}$	I	$\overline{\text{EX\_CS0}}$	O	M(2)		B		Input (A)	High	Input (A)	High	Input	High	High-Z		
242	AE17	$\overline{\text{MINT}}$	I	$\overline{\text{EX\_CS1}}$	O	M(2)		B		Input (A)	High	Input (A)	High	Input	High	High-Z		
243	AF19	MERROR	I	EX_ADDR(2)	O	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
244	AE19	$\overline{\text{MRESET}}$	O	EX_ADDR(3)	O	M(2)		O		Output (A)	0	Output (A)	0	Output	0	High-Z		
245	AD19	MFRAME_SYNC	I	EX_ADDR(4)	O	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
246	AC19	MSRC_FLOW	I	EX_ADDR(5)	O	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
247	AC20	MCP_FLOW	I	EX_ADDR(6)	O	M(2)		B		Input (A)	Low	Input (A)	Low	Input	Low	High-Z		
248	A15	USB host USB1HP	A					A	1	Input (A)	Low	Input (A)	Low	Input	Low			
249	A14	USB1HM	A					A	1	Input (A)	High	Input (A)	High	Input	High			
250	C15	USB1PENC	O					O		Output (A)	0	Output (A)	0	Output	0			
251	D15	USB1OVC	I					I		Input (A)	Low	Input (A)	Low	Input	Low			
252	D14	VCCUSB1	D3PG					D3PG	1									
253	B15	VSSUSB1	D3PG					D3PG	1									
254	B14	USB host USB2HP	A	USB function USB2HP	A	(internal)		A	1	Input (A)	High	Input (A)	High	Input	High	Input	High	
255	B13	USB2HM	A	USB2HM	A	(internal)		A	1	Input (A)	Low	Input (A)	Low	Input	Low	Input	Low	
256	A11	USB2PENC	O					O		Output (A)	0	Output (A)	0	Output	0			

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
257	B11	USB host	USB2OVC	I				I		Input (A)	Low	Input (A)	Low	Input	Low			
258	C13		VCCUSB2	D3PG				D3PG	1									
259	C14		VSSUSB2	D3PG				D3PG	1									
260	C26	PLL system	VCCPLLA1	APG				APG	1									
261	D25		VSSPLLA1	APG				APG	1									
262	AF24	PLL display output	VCCPLLA2	APG				APG	1									
263	AE23		VSSPLLA2	APG				APG	1									
264	C12	USB crystal	XTAL_USB	A				A	1									
265	A12		EXTAL_USB	A(I)				A	1									
266	AE14	Audio crystal	XTAL_AUD	A				A	1									
267	AE13		EXTAL_AUD	A				A	1									
268	AC13	AUDIO_CLK	AUDIO_CLK	B				B		Input (A)	Clock or low	Input(A)	Clock or low	Input	Clock or low			
269	D23	Others	Config	I				I		Input (A)	High (PCI)	Input(A)	Low (MPX)	Input	Low (MPX)/ High (PCI)			
270	E24		$\overline{\text{RESO}}$	O				O		Output (A)	0	Output (A)	0	Output	0			
271	D4		$\overline{\text{PLL\_ENABLEN}}$	I				I		Input (A)	High/Low	Input (A)	High/Low	Input	High/Low			
272	AA23		scan_mode	I				I		Input (A)	Low	Input (low)	Low	Input	Low			For testing only use
273	B25	JTAG	TDI	I				I		Input (A)	High (pull-up)	Input (A)	High (pull-up)	Input	High (pull-up)			
274	C25		TDO (Tri)	O				O		High-Z (A)		High-Z (A)		High-Z				
275	B26		TMS	I				I		Input (A)	High (pull-up)	Input (A)	High (pull-up)	Input	High (pull-up)			
276	D26		TCK	I				I		Input (A)	Low	Input (A)	Low	Input	Low			
277	B2		ITEST1*															
278	B12		ITEST2*															
279	D13		ITEST3*															
280	D9		VDDI1	D1PG														
281	D18		VDDI2	D1PG														
282	P23		VDDI3	D1PG														
283	AC18		VDDI4	D1PG														
284	AC9		VDDI5	D1PG														

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
285	N4	VDDI6	D1PG															
286	C9	VSSI1	D1PG															
287	C18	VSSI2	D1PG															
288	P24	VSSI3	D1PG															
289	AD18	VSSI4	D1PG															
290	AD9	VSSI5	D1PG															
291	N3	VSSI6	D1PG															
292	B5	VCCQ1	D3PG															
293	B9	VCCQ2	D3PG															
294	A13	VCCQ3	D3PG															
295	B18	VCCQ4	D3PG															
296	A22	VCCQ5	D3PG															
297	C23	VCCQ6	D3PG															
298	D24	VCCQ7	D3PG															
299	G25	VCCQ8	D3PG															
300	J25	VCCQ9	D3PG															
301	L24	VCCQ10	D3PG															
302	N26	VCCQ11	D3PG															
303	R24	VCCQ12	D3PG															
304	U24	VCCQ13	D3PG															
305	W25	VCCQ14	D3PG															
306	AB25	VCCQ15	D3PG															
307	AC24	VCCQ16	D3PG															
308	AD23	VCCQ17	D3PG															
309	AE18	VCCQ18	D3PG															
310	AD14	VCCQ19	D3PG															
311	AE9	VCCQ20	D3PG															
312	AD4	VCCQ21	D3PG															
313	AD1	VCCQ22	D3PG															
314	AA1	VCCQ23	D3PG															
315	V2	VCCQ24	D3PG															
316	R3	VCCQ25	D3PG															

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
317	M1	VCCQ26	D3PG															
318	K3	VCCQ27	D3PG															
319	G2	VCCQ28	D3PG															
320	E3	VCCQ29	D3PG															
321	D6	VSSQ1	D3PG															
322	A9	VSSQ2	D3PG															
323	D12	VSSQ3	D3PG															
324	A18	VSSQ4	D3PG															
325	D20	VSSQ5	D3PG															
326	D22	VSSQ6	D3PG															
327	E23	VSSQ7	D3PG															
328	G24	VSSQ8	D3PG															
329	J24	VSSQ9	D3PG															
330	L23	VSSQ10	D3PG															
331	N23	VSSQ11	D3PG															
332	R23	VSSQ12	D3PG															
333	U23	VSSQ13	D3PG															
334	W24	VSSQ14	D3PG															
335	AA24	VSSQ15	D3PG															
336	AB23	VSSQ16	D3PG															
337	AC22	VSSQ17	D3PG															
338	AF18	VSSQ18	D3PG															
339	AD13	VSSQ19	D3PG															
340	AF9	VSSQ20	D3PG															
341	AC5	VSSQ21	D3PG															
342	AB4	VSSQ22	D3PG															
343	Y4	VSSQ23	D3PG															
344	V3	VSSQ24	D3PG															
345	R4	VSSQ25	D3PG															
346	N1	VSSQ26	D3PG															
347	K4	VSSQ27	D3PG															
348	H4	VSSQ28	D3PG															

No.	Package No.	Function A	Signal Type: A	Function B	Signal Type: B	Mode Bit	GPIO (52–55 don't exist)	Signal Type	Non Digital	Initial Condition after Poweron Reset (PCI)	Initial I/O after Poweron Reset (PCI)	Initial Condition after Poweron Reset (MPX)	Initial I/O after Poweron Reset (MPX)	Function A Pin state at Initial Condition	Function A I/O at Initial Condition	Function B Pin state at Initial Condition	Function B I/O at Initial Condition	Note about PAD
349	F4	VSSQ29	D3PG															
350	AC10	VBBENV	I							Input (A)	Low	Input (A)	Low	Input	Low			
351	AC8	VBGP	O (monitor)							Output (A)	No connect	Output (A)	No connect	Output	No connect			
352	AC7	VBGN	O (monitor)							Output (A)	No connect	Output (A)	No connect	Output	No connect			

Note: * It must be open.

Input (A)	Input and func A is available
Input (B)	Input and func B is available
Output (A)	Output and func A is available
Input (G)	Input and GPIO is available

Pin Type	Pin Type Description
I	Input pin
IS	Schmitt trigger input pin
O	Output pin
B	Bi-directional pin
ST	Sustained tri-state (see PCI specification)
T	Tri-state output
A	Analog pin
APG	Analog power/ground pin
VCCPLLA1, VCCPLLA2	Analog 3.3V
D3PG	Digital 3.3V power/ground pin
D1PG	Digital 1.5(76C)V power/ground pin

Note: Bidirectional pins can be used to support OpenDrain configuration.

1.5 Pin Description

The following table describes the signals of each block that may be configured to be connected to an I/O pin. The configuration is shown in the pin-mode section.

Table 1.2 Block External Pin Out Description

Item		Description
SDRAM I/f	SD_DATA(31:0)	Bidirectional data bus
(32 bits)	SD_ADDR(12:0)	Row/Column address
	$\overline{RAS}$	Row address strobe
	$\overline{CAS}$	Column address strobe
	$\overline{WE}$	Write enable
	$\overline{CS}$	Chip select
	$\overline{BA0}$	Bank address 0
	$\overline{BA1}$	Bank address 1
	SD_CLK	Synchronous clock
	SD_CKE	Synchronous clock enable
	DQM(3:0)	Data qualifier mask. Masks individual bytes on writes to the SDRAM.
Video Input	VI_Data(7:0)	CCIR656 encoded CCIR601 video data
	VI_Clk	Clock for video input data
Display Out Digital		Direct support for digital TFT displays. PWM signals for contrast etc and GPIO signals for power sequencing are detailed below. Also used during reset for configuration
	DO_Data(17:0)	Digital display data. 6 bit color for each of R,G and B
	$\overline{DO\_VSYNC}$	Vertical frame sync
	$\overline{DO\_HSYNC}$	Horizontal frame sync/Composite sync
	DO_DEN	Defines active display I.e. combines vertical and horizontal blank
	DOT_CLK	Display clock
PCI		These signals are defined fully in the PCI specification (version 2.1)
	AD(31:0)	Multiplexed address/data bus
	C/$\overline{BE}$(3:0)	Command/Byte enable
	PAR	Parity for AD bus
	$\overline{FRAME}$	Frame start

Item		Description
PCI	$\overline{\text{TRDY}}$	Target ready
	$\overline{\text{IRDY}}$	Initiator ready
	$\overline{\text{STOP}}$	Stop transaction
	$\overline{\text{DEVSEL}}$	Device select
	IDSEL	Configuration device select
	$\overline{\text{PERR}}$	Parity error
	$\overline{\text{SERR}}$	System error
	$\overline{\text{REQ}}$	Bus request
	$\overline{\text{GNT}}$	Bus grant
	PCI_CLK	Synchronous PCI clock
	$\overline{\text{RST}}$	Chip reset
	$\overline{\text{INTA}}$	Interrupt output A
SH-4 I/f		Supports multiplexed bus of SH-4
	D(31:0)	Multiplexed address/data bus
	$\overline{\text{DREQ}}$	DMA request
	$\overline{\text{FRAME}}$	Frame start cycle. This is used to identify burst access
	$\overline{\text{RDY}}$	Slave ready signal. Used to insert hardware controlled wait states Tristate output.
	$\overline{\text{BS}}$	Bus Start
	RD/$\overline{\text{WR}}$	Read/Write signal
	SH4_$\overline{\text{CSA}}$	Chip select A. This defines an area of 64 Mbytes.
	SH4_$\overline{\text{CSB}}$	Chip select B. This defines an area of 64 Mbytes.
	DACK	DMA transfer acknowledge
	DRAK	DMA request acknowledge
	D(63)	Bit 63 of SH-4 data bus. Needs to connect D(31) of SH7751
	D(62)	Bit 62 of SH-4 data bus. Needs to connect D(30) of SH7751
	D(61)	Bit 61 of SH-4 data bus. Needs to connect D(29) of SH7751
	CKIO	Bus clock from SH-4
	$\overline{\text{RST}}$	Chip reset
	$\overline{\text{IRL}}$	Interrupt to the SH-4. Tristate output

Item		Description
SSI(0)		Supports Philips format and its derivatives. These ports can operate as either master or slave. High speed support is also provided
	SSI0_FSY	Frame sync, defines left/right channels
	SSI0_SCK	Shift clock
	SSI0_SDATA	Bidirectional serial data
SSI(1)	SSI1_FSY	Frame sync
	SSI1_SCK	Shift clock
	SSI1_SDATA	Bidirectional serial data
SSI(2)	SSI2_FSY	Frame sync
	SSI2_SCK	Shift clock
	SSI2_SDATA	Bidirectional serial data
SSI(3)	SSI3_FSY	Frame sync
	SSI3_SCK	Shift clock
	SSI3_SDATA	Bidirectional serial data
I2C(0)	I2C0_SCL	Data transfer clock
	I2C0_SDA	Transmit/Receive data
I2C(1)	I2C1_SCL	Data transfer clock
	I2C1_SDA	Transmit/Receive data
CAN(0)		Supports CAN 2.0b for speeds up to 1 Mbit/s
	CAN0_RX	Receive data
	CAN0_TX	Transmit data
	$\overline{\text{CAN0\_NERR}}$	Transceiver error
CAN(1)		For slow speed CAN additional GPIO pins can be used for control
	CAN1_RX	Receive data
	CAN1_TX	Transmit data
	$\overline{\text{CAN1\_NERR}}$	Transceiver error
SPDIF transmitter		Signal defined within IEC60958 specification
	SPDIF_OUT	Control/data output signal
SPDIF receiver		Signal defined within IEC60958 specification
	SPDIF_IN	Control/data input signal

Item		Description
UART(0)		These UART's support asynchronous transmit and receive. Hardware flow control signals can be implemented through GPIO if required.
	UART0_RXD	Asynchronous receive data
	UART0_TXD	Asynchronous transmit data
IrDA		Signals as defined in IrDA Serial Infrared Physical layer specification, Version 1.3. Maximum data rate is 115.2 kbits/s
	IRDA_RXD	Asynchronous receive data
	IRDA_TXD	Asynchronous transmit data
UART(1)	UART1_RXD	Asynchronous receive data
	UART1_TXD	Asynchronous transmit data
UART(2)	UART2_RXD	Asynchronous receive data
	UART2_TXD	Asynchronous transmit data
UART(3)	UART3_RXD	Asynchronous receive data
	UART3_TXD	Asynchronous transmit data
Audio Codec	$\overline{\text{AC\_RES}}$	Audio Codec reset. Used for recovering from power down modes
	AC_SYNC	Frame sync
	AC_SDATA_IN	Serial data in
	AC_SDATA_OUT	Serial data out
	AC_BIT_CLK	Synchronous serial clock
GPIO		General purpose IO indicates that the pin can be used as either an output under program control or an input where the state of the pin can be read.
	GPIO(2:0)	e.g. display power control, ENVEE,ENCTL,ENVDD
	INT(7:6)	External interrupt inputs
	INT(5:3)	External interrupt inputs
	INT(2)	External interrupt inputs
	INT(1:0)	External interrupt inputs
	TIMER/CTR(3:0)	Configurable timers and counter support
	PWM(3:0)	Programmable pulse width modulation outputs

Item		Description
SPI(0)	SPI0_SIMO	Serial transmit data
	SPI0_MISO	Serial receive data
	SPI0_CLK	Shift clock
	$\overline{\text{SPI0\_CS}}$	Chip select for device 0
SPI(1)	SPI1_SIMO	Serial transmit data
	SPI1_MISO	Serial receive data
	SPI1_CLK	Shift clock
	$\overline{\text{SPI1\_CS}}$	Chip select for device 1
SPI(2)	SPI2_SIMO	Serial transmit data
	SPI2_MISO	Serial receive data
	SPI2_CLK	Shift clock
	$\overline{\text{SPI2\_CS}}$	Chip select for device 2
ATAPI		Supports two devices on one channel
	AT_DSD(15:0)	Bi-directional data bus
	AT_DSA(2:0)	Address bus
	$\overline{\text{AT\_DMACK0}}$	DMA acknowledge
	AT_DMARQ0	DMA request
		Schmidt trigger input pin
	$\overline{\text{AT\_DCS}}$(1:0)	Chip select
	$\overline{\text{AT\_DIOW}}$	Disk write
	$\overline{\text{AT\_DIOR}}$	Disk read
	AT_DCHRDY0	Ready signal
		Schmidt trigger input pin
	AT_DIRQ1	Interrupt request
		Schmidt trigger input pin
	$\overline{\text{AT\_RESET}}$	ATAPI device reset

Item		Description
OS8104 I/f		Direct connection to Most transceiver(OS8104)
	MPAD(1:0)	Parallel address select
	MDATA(7:0)	Data bus
	$\overline{\text{MRD}}$	Read control
	$\overline{\text{MWR}}$	Write control
	$\overline{\text{MAINT}}$	Asynchronous message interrupt
	$\overline{\text{MINT}}$	Control message and power-on interrupt
	MERROR	Error indicator
	$\overline{\text{MRESET}}$	Resets the Most transceiver
	MFRAME_SYNC	Frame sync I/O
	MSRC_FLOW	Parallel flow control
	MCP_FLOW	Control port flow control
Expansion bus		This is a general purpose expansion bus with SRAM type operation.
	EX_ADDR(6:0)	Address bits
	EX_DATA(7:0)	Data bus
	$\overline{\text{EX\_RD}}$	Read select
	$\overline{\text{EX\_WR}}$	Write select
	$\overline{\text{EX\_CS0}}$	Chip select 0
	$\overline{\text{EX\_CS1}}$	Chip select 1
USB	USB1HP	USB port 1 D+ (Host only)
	USB1HM	USB port 1 D- (Host only)
	USB1PENC	USB port 1 Power enable control
	USB1OVC	USB port 1 Over-Current detect
	VCCUSB1	USB port 1 Transceiver power
	VSSUSB1	USB port 1 Transceiver Ground
	USB2HP	USB port 2 D+ (Host or Function)
	USB2HM	USB port 2 D- (Host or Function)
	USB2PENC	USB port 2 Power enable control (Host) (High active) /USB port 2 D+ Pullup Enable (Function) (Low active)
	USB2OVC	USB port 2 Over-Current detect (Host) (Low active) /port 2 cable connection monitor pin.(Function) (High active)
	VCCUSB2	USB port 2 Transceiver power
	VSSUSB2	USB port 2 Transceiver Ground

Item		Description
PLL system		Main system clock PLL
	VCCPLLA1	Analog power for PLL
	VSSPLLA1	Analog ground for PLL
PLL Display output		Display output clock
	VCCPLLA2	Analog power for PLL
	VSSPLLA2	Analog ground for PLL
USB Crystal		USB clock generation (48 MHz)
	XTAL_USB	Output for USB crystal resonator
	EXTAL_USB	Input for External USB input clock/crystal resonator
Audio Crystal		Supports the connection of an external crystal for generating audio clock (512*fs)
	XTAL_AUD	Output for Audio clock crystal resonator
	EXTAL_AUD	Input for Audio clock crystal resonator
	AUDIO_CLK	Audio output clock, same frequency as crystal input, Becomes audio external clock input by setting a register.
Others	Config	Selects PCI or SH-4 multiplex bus(PCI = 1, MPX = 0)
	$\overline{\text{RESO}}$	Reset output from device.
	$\overline{\text{PLL\_ENABLEN}}$	Low: PLL is active (normal state), High: PLL is disable (PLL CLOCK is stopped.)
	scan_mode	Scan test mode signal Fixed low
	VBBENV	Back Bias enable: This pin is used for testing. Fixed low.
	VBGP	Back Bias monitor 1 (output) This pin is used for testing. This pin must be left open.
	VBGN	Back Bias monitor 2 (output) This pin is used for testing. This pin must be left open.
JTAG		Full JTAG support
	$\overline{\text{TRSTN}}$	Dedicated JTAG reset signal
	TDI	Test data input
	TDO	Test dataoutput
	TMS	Test mode select
	TCK	Test clock

1.6 Operating Voltage

This device use 1.5 V for internal digital logic and 3.3 V for I/O and analogue modules.

This device will support automotive specifications including a temperature range of –40 to 85°C. (Note: This device cannot have its inputs or bi-directional signals connected to 5 V devices).

1.7 Package

The device is packaged in a TBGA352.

1.8 Detailed Architecture

1.8.1 Main Clocking

For a PCI based system the memory interface and graphics engine will work at 1, 2 or 3 times PCI bus speed with a maximum frequency of 100 MHz. When SH-4 MPX is used as the system bus then the clock will be derived from the MPX clock.

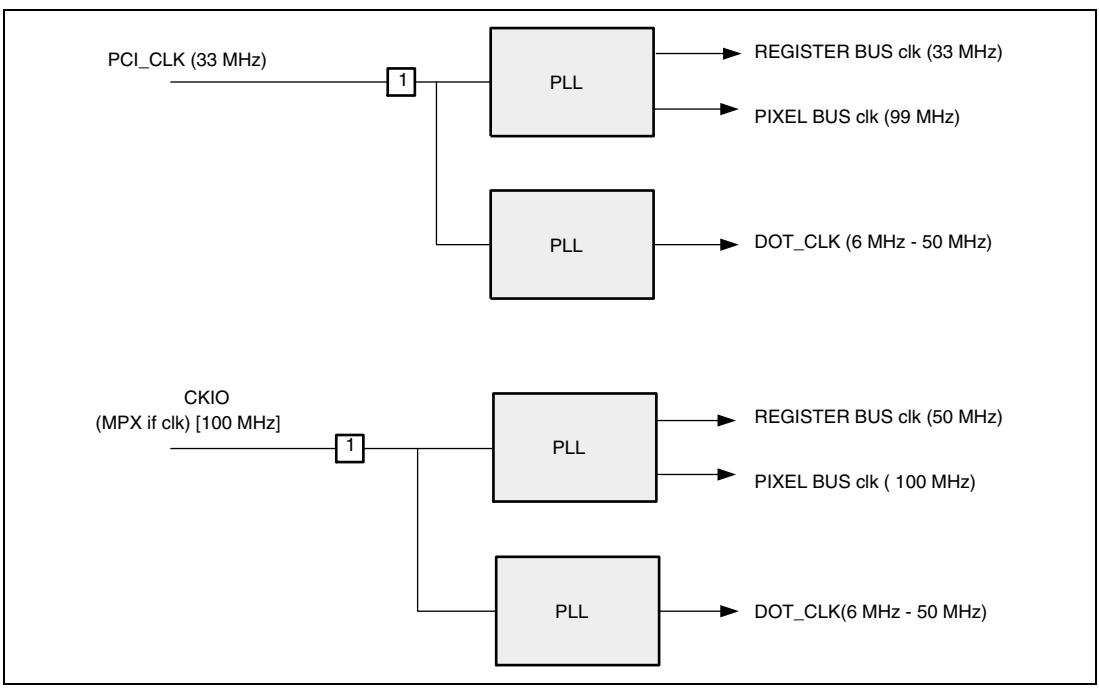


Figure 1.2 Main Clock

Several modules have the condition to meet the specifications.

For this reason, the selectable clock frequencies are shown in Table1.3 and Table 1.4.

Table 1.3 HD64404 Clock Table for MPX

Module	Condition
CAN	The quotient of dividing register bus clock by an integer must be an integer from 8 to 25
	with oscillator tolerance of less than 1.58% for up to 125 Kbaud
MOST	Register bus clock must be from 29 MHz to 50 MHz.
UART	Show the case of Baud rate tolerance that is less than +2.3 to -2.5%
IrDA	Baud rate tolerance must be less than +/- 0.87%
DisplayOut	CKIO affects dotclock using internal dot clock mode. See displayout module manual
	Dot clock tolerance depends on Display Unit
Pixel bus	CKIO is proportional to Pixel bus clock (1:1 for MPX, 1:3 for PCI), 100MHz max
	CKIO is directly related to Pixel bus module performance
Register Bus	CKIO is proportional to Register bus clock (2:1 for MPX, 1:1 for PCI)
	CKIO is directly related to Register bus module performance

(1) Case 1: CAN, MOST, IrDA, and UART are used

Register bus clock (MHz)	Pixel bus clock (MHz)	CAN	MOST	IrDA	UART		
		OK/NG	OK/NG	OK/NG	Max Baudrate	Error (%)	Max Baudrate
30	60	OK	OK	NG	—	1.73	115200
31	62	NG	OK	NG(*)	(57600)	-1.07	57600
32	64	OK	OK	NG	—	2.12	57600
33	66	OK	OK	OK	115200	-0.54	115200
34	68	OK	OK	NG	—	-1.18	38400
35	70	NG	OK	OK	57600	-0.06	57600
36	72	OK	OK	NG	—	-2.34	115200
37	74	NG	OK	OK	115200	0.37	115200
38	76	OK	OK	NG	—	-1.83	57600
39	78	OK	OK	OK	57600	0.76	57600
40	80	OK	OK	NG	—	-1.36	115200
41	82	NG	OK	NG	—	1.11	115200
42	84	OK	OK	NG(**)	(57600)	-0.93	57600
43	86	NG	OK	NG	—	1.43	57600
44	88	OK	OK	OK	115200	-0.54	115200
45	90	OK	OK	NG	—	1.73	115200
46	92	OK	OK	OK	57600	-0.17	57600
47	94	NG	OK	NG	—	-1.93	115200
48	96	OK	OK	OK	115200	0.16	115200
49	98	NG	OK	NG	—	2.25	115200
50	100	OK	OK	OK	57600	0.47	57600

Legend:

NG (*) : error is 1.07%

NG (**) : error is 0.92%

(2) Case 2: MOST, IrDA, and UART are used, CAN is not used

Register bus clock (MHz)	Pixel bus clock (MHz)	CAN	MOST	IrDA	UART		
		OK/NG	OK/NG	OK/NG	Max Baudrate	Error (%)	Max Baudrate
22.1184	44.2368	NG	NG	0	115200	0	115200
25.8048	51.6096	NG	NG	0	115200	0	115200
29.4912	58.9824	NG	OK	0	115200	0	115200
33.1776	66.3552	NG	OK	0	115200	0	115200
36.8640	73.7280	NG	OK	0	115200	0	115200
40.5504	81.1008	NG	OK	0	115200	0	115200
44.2368	88.4736	NG	OK	0	115200	0	115200
47.9232	95.8464	NG	OK	0	115200	0	115200

Table 1.4 HD64404 Clock Table for PCI

Module	Condition
CAN	The quotient of dividing register bus clock by an integer must be an integer from 8 to 25
	with oscillator tolerance of less than 1.58% for up to 125 Kbaud
MOST	register bus clock must be from 29 MHz to 50 MHz.
UART	Show the case of Baud rate tolerance that is less than +2.3 to -2.5%
IrDA	Baud rate tolerance must be less than +/- 0.87%
DisplayOut	PCI clock affects dotclock using internal dot clock mode. See displayout module manual
	Dot clock tolerance depends on Display Unit
Pixel bus	PCI clock is proportional to Pixel bus clock (1: 3 for PCI), 100 MHz max
	PCI clock is directly related to Pixel bus module performance
Register bus	PCI clock is proportional to Register bus clock (1: 1 for PCI)
	PCI clock is directly related to Register bus module performance

(1) Case 1 : CAN , MOST, IrDA, and UART are used

Register bus clock (MHz)	Pixel bus clock (MHz)	CAN	MOST	IrDA	UART		
		OK/NG	OK/NG	OK/NG	Max Baudrate	Error (%)	Max Baudrate
30	90	OK	OK	NG	—	1.73	115200
31	93	NG	OK	NG(*)	(57600)	-1.07	57600
32	96	OK	OK	NG	—	2.12	57600
33	99	OK	OK	OK	115200	-0.54	115200

Legend:
NG (*) : error is 1.07%

(2) Case 2: MOST, IrDA, and UART are used, CAN is not used

Register bus clock (MHz)	Pixel bus clock (MHz)	CAN	MOST	IrDA	UART		
		OK/NG	OK/NG	OK/NG	Max Baudrate	Error (%)	Max Baudrate
22.1184	66.3552	NG	NG	0	115200	0	115200
25.8048	77.4144	NG	NG	0	115200	0	115200
29.4912	88.4736	NG	OK	0	115200	0	115200
33.1776	99.5328	NG	OK	0	115200	0	115200

Display Window Size:

Available display window size depends on HD64404 operational frequency and the condition of 3 planes.

The operational frequency is the same as Pixel Bus clock frequency. 3 planes are Foreground plane, Background plane, and Video plane.

Case 1 to Case 4 are shown as follows.

Note: These tables are not taken into consideration how much bus space can be allocated to CPU interface must be kept. Also the following conditions.

a. SDRAM CAS latency = 2
b. Memory Base 1/2/3 registers in VideoIN module must be set as 32 byte boundary.
c. Yscale in VideoIN module must be equal to or less than 1.
d. DAMn start address in DMAC module must be set as 32 byte boundary.
e. When using 2DGE of GE module, the number of the page misses of SDRAM from 2DGE access must be limited to 52 times during 25 µs. Q2SD/RU in GE does not have any constraints.

(1) Case1: Fore Ground (16bit/pixel): ON
BackGround (16bit/pixel): ON
Video In: ON

	Screen refresh	Screen Width	Screen height	Dotclock (MHz)	Pixel bus clock (MHz)					
					100	99	90	88	78	66
QWVGA	60	400	240	7.6	PASS	PASS	PASS	PASS	PASS	PASS
QWVGA	60	480	234	9.2	PASS	PASS	PASS	PASS	PASS	PASS
HVGA	60	640	240	12.2	PASS	PASS	PASS	PASS	PASS	PASS
VGA	60	640	480	25.2	PASS	PASS	No Good	No Good	No Good	No Good
WVGA	60	800	480	31.5	PASS	PASS	No Good	No Good	No Good	No Good
WVGA	60	854	480	33.6	PASS	PASS	No Good	No Good	No Good	No Good
VGA	75	640	480	31.5	PASS	PASS	No Good	No Good	No Good	No Good
WVGA	75	800	480	39.4	No Good	No Good	No Good	No Good	No Good	No Good
WVGA	75	854	480	42.1	No Good	No Good	No Good	No Good	No Good	No Good

This table is also applied to the following conditions.

case 1-2: Fore ground (8 bit/pixel) : ON or OFF
Back ground (16 bit/pixel) : ON
Video IN : ON

case 1-3 : Fore ground (16 bit/pixel) : ON
Back ground (8 bit/pixel) : ON or OFF
Video IN : ON

(2) Case2: Fore Ground (16bit/pixel): ON
BackGround (16bit/pixel): ON
Video In : OFF

	Screen refresh	Screen Width	Screen height	Dotclock (MHz)	Pixel bus clock (MHz)					
					100	99	90	88	78	66
QWVGA	60	400	240	7.6	PASS	PASS	PASS	PASS	PASS	PASS
QWVGA	60	480	234	9.2	PASS	PASS	PASS	PASS	PASS	PASS
HVGA	60	640	240	12.2	PASS	PASS	PASS	PASS	PASS	PASS
VGA	60	640	480	25.2	PASS	PASS	PASS	PASS	No Good	No Good
WVGA	60	800	480	31.5	PASS	PASS	No Good	No Good	No Good	No Good
WVGA	60	854	480	33.6	PASS	PASS	No Good	No Good	No Good	No Good
VGA	75	640	480	31.5	PASS	PASS	No Good	No Good	No Good	No Good
WVGA	75	800	480	39.4	No Good	No Good	No Good	No Good	No Good	No Good
WVGA	75	854	480	42.1	No Good	No Good	No Good	No Good	No Good	No Good

This table is also applied to the following conditions.

case 2-2: Fore ground (8 bit/pixel) : ON or OFF
Back ground (16 bit/pixel) : ON
Video IN : OFF

case 2-3 : Fore ground (16 bit/pixel) : ON
Back ground (8 bit/pixel) : ON or OFF
Video IN : OFF

(3) Case3: Fore Ground (8bit/pixel): ON
BackGround (8bit/pixel): ON
Video In : ON

	Screen refresh	Screen Width	Screen height	Dotclock (MHz)	Pixel bus clock (MHz)					
					100	99	90	88	78	66
QWVGA	60	400	240	7.6	PASS	PASS	PASS	PASS	PASS	PASS
QWVGA	60	480	234	9.2	PASS	PASS	PASS	PASS	PASS	PASS
HVGA	60	640	240	12.2	PASS	PASS	PASS	PASS	PASS	PASS
VGA	60	640	480	25.2	PASS	PASS	PASS	PASS	PASS	PASS
WVGA	60	800	480	31.5	PASS	PASS	PASS	PASS	PASS	PASS
WVGA	60	854	480	33.6	PASS	PASS	PASS	PASS	PASS	PASS
VGA	75	640	480	31.5	PASS	PASS	PASS	PASS	PASS	PASS
WVGA	75	800	480	39.4	PASS	PASS	PASS	PASS	PASS	No Good
WVGA	75	854	480	42.1	PASS	PASS	PASS	PASS	No Good	No Good

(4) Case4: Fore Ground (8bit/pixel): ON
BackGround (8bit/pixel): ON
Video In : OFF

	Screen refresh	Screen Width	Screen height	Dotclock (MHz)	Pixel bus clock (MHz)					
					100	99	90	88	78	66
QWVGA	60	400	240	7.6	PASS	PASS	PASS	PASS	PASS	PASS
QWVGA	60	480	234	9.2	PASS	PASS	PASS	PASS	PASS	PASS
HVGA	60	640	240	12.2	PASS	PASS	PASS	PASS	PASS	PASS
VGA	60	640	480	25.2	PASS	PASS	PASS	PASS	PASS	PASS
WVGA	60	800	480	31.5	PASS	PASS	PASS	PASS	PASS	PASS
WVGA	60	854	480	33.6	PASS	PASS	PASS	PASS	PASS	PASS
VGA	75	640	480	31.5	PASS	PASS	PASS	PASS	PASS	PASS
WVGA	75	800	480	39.4	PASS	PASS	PASS	PASS	PASS	No Good
WVGA	75	854	480	42.1	PASS	PASS	PASS	PASS	No Good	No Good

1.8.2 Pixel Bus

The pixel bus is dedicated to accessing the SDRAM. To maximize the bandwidth of the SDRAM, all accesses to this bus are based on burst transfers (i.e. multiple words). The size of these bursts is variable depending on the functionality of the block.

The blocks connected to the pixel bus are responsible for generating their DMA addresses for data transfers to/from the SDRAM. This is required because only these modules can determine the addresses for the transfer e.g. graphics engine defines the addresses for the draw operation. These devices will maintain their own internal FIFO's.

This bus is based on a separate command and data transfer phases. This allows multiple commands to be sent to the memory controller in advance so that overlapping DRAM accesses can be achieved to maximize SDRAM efficiency.

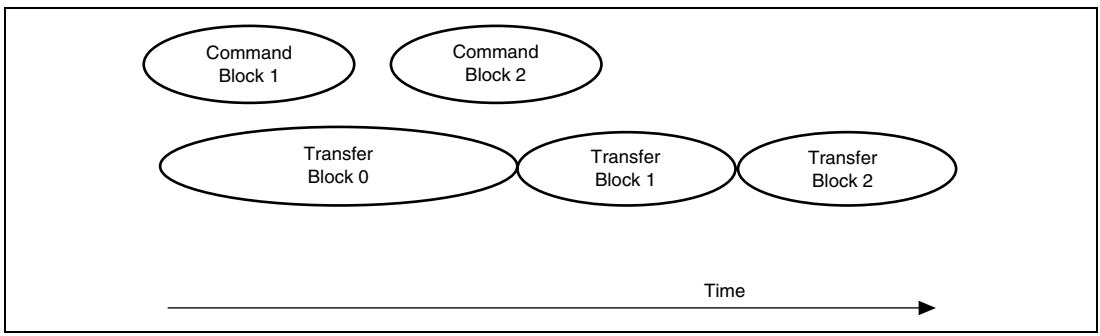


Figure 1.3 Transfer on the Pixel bus

The bus contains a separate write and read bus to allow overlapping of commands within the memory controller. This allows the use of multiplexors rather than tri-state buses to improve manufacturing testability.

All transfers on the pixel bus are between one block and the SDRAM.

The memory arbiter controls the sharing of this bus. This priority-decoding combines fixed and round robin arbitration for both real-time and non-real time blocks, e.g. the display output will be real time but the graphics renderer will be non real time. The sizes of the transfers are arranged to compromise between maximum SDRAM throughput and latency.

1.8.3 Register Bus

The register bus is a 32 bit bus. This operates at 33 MHz in a PCI system. In Super H MPX mode the bus speed is 1/2 the pixel bus speed.

This bus requires 2 clocks to do a write or read where the first cycle is the address stage and the second is the data transfer stage.

This bus supports a single master which is the DMA controller (DMAC), because the modules that connected in the register bus are only peripherals, and each peripherals is accessed by only DMAC. Access to this bus from the Super H processor is enabled through the DMAC. As transfers on the register bus are single words the maximum latency for the processor to access the bus is 2 clock cycles. In the event that is accessed from PCI/MPX bus, additional latency occurs.

The DMAC has a dedicated SRAM connected to it. This is used as a common FIFO for the peripherals connected to the register bus. Each peripheral is allocated to an area within this FIFO buffer.

The peripherals only have local double buffering which means that one register can be read/written while the other is shifting in/out for example.

The DMAC is then responsible for single word accesses to the peripherals to transfer the data from/to the peripheral to/from Channel FIFO buffer of the DMAC.

Transfer to/from this DMAC to the Super H or to the pixel bus can then be performed in bursts.

This DMAC arbitrates between requesting peripherals for access to the register bus and will also arbitrate which of the peripherals will request access onto the pixel bus or PCI/MPX bus.

The figure below shows the operation of the DMAC in terms of data-flow.

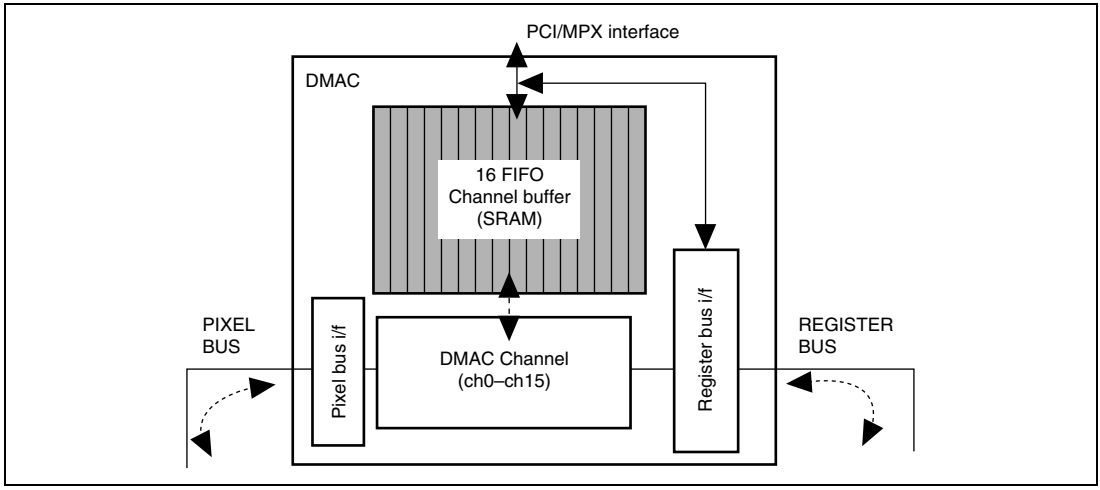


Figure 1.4 DMAC Block Diagram

1.8.4 System Interface

The system interface can be configured either to use the PCI or SH-4 bus. Both these buses are 32 bit multiplexed buses. The major differences between the two buses are

- MPX clock speed is 100 MHz, PCI clock speed is 33 MHz
- HD64404 device can be master (initiator) in PCI configuration.
- MPX bursts are fixed sizes

This block controls the interface between the SH-4 and this device. It provides a bridge between the PCI/MPX interface and the internal register bus and also to the pixel bus. Data to be presented onto the pixel bus will be buffered into blocks where possible before getting access to the bus.

The PCI/MPX interface is connected both to the pixel bus and the DMAC. The pixel bus is used primarily for UMA (Unified Memory Architecture) systems in the MPX mode, direct drawing from the processor and the transfer of bitmaps. The latter is used for control and low speed data transfer to the low speed peripherals.

The Super H processor can access FIFO buffer of the DMAC for direct data-flow communication with the peripherals. This is primarily aimed at systems using the MPX bus to allow the DMA controller within the Super H to transfer data between the companion chip and the local memory connected directly to the Super H.

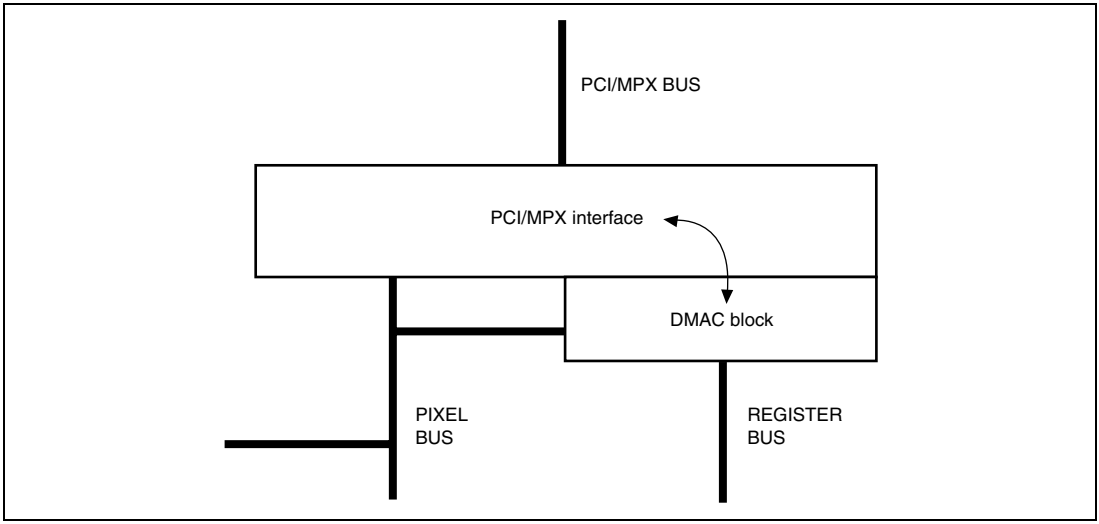


Figure 1.5 System Interface

1.8.5 PCI

The interface is a 32 bit PCI operating at 33 MHz.

The PCI interface has initiator functionality to allow direct DMA from the peripherals on the register bus to the system memory and to other components e.g. MPEG decoder connected to PCI/MPX BUS.

1.8.6 MPX

This interface is 32 bits operating at up to 100 MHz.

The interface is always the slave as a separate multi-function bus controller is not included.

1.8.7 Graphics Memory (SDRAM) Controller

The memory interface allows the direct connection of 100 MHz or faster SDRAM's. The controller accepts a second command while processing the first to allow maximum performance by activating multiple banks.

This memory interface is 32 bits and operates synchronously with the pixel bus.

The memory controller is programmable to support different sizes of SDRAM's and different operating speeds. To support this the number of bits used for the RAS and CAS address are programmable along with the CAS latency and other parameters

As the memory controller is the only connection to the SDRAM's the controller can take advantage of leaving the page open to reduce overhead.

1.8.8 Interrupt Controller

The interrupt priority block supports 30 interrupts. Each interrupt can be programmed to have an interrupt priority between 0 and 31. Only one interrupt should have each priority. Based on this the interrupt with the highest priority will be passed to the processor.

Each block will generate only one interrupt line. Each block will be responsible for having a register with a mask for each source of interrupt and a corresponding status register showing the internal interrupts that are active.

Thus, the mechanism for determining the interrupt is to read the interrupt status register of the interrupt priority block. This will identify the block responsible for generating the interrupt. The status register within the block should then be read. This identifies the actual source of the interrupt.

1.8.9 Power Saving

Each peripheral block can enter a low power mode by stopping the clock to that module. This is under software control and is handled by the Power Control & Configuration module.

In addition a deeper level of sleep mode of the SDRAM can be entered under software control.

The interrupt controller working in low power mode provides wake-up.

1.9 Endian Support

1.9.1 Definitions

Byte	8 bits of data
Word	16 bits of data
Long-word	32 bits of data
Little-endian	A byte or word of data occupying the LSB's of a long-word would be at address 0
Big-endian	A byte or word of data occupying the MSB's of a long-word would be at address 0
Graphics memory(GM)	SDRAM connected to the memory interface of HD64404
System memory	RAM connected directly to the SH-4
Unified Memory Architecture	System where Graphics memory would be used for the program memory as well as graphics memory

1.9.2 Description

The HD64404 device supports both little and big endian systems. The SH-4 can also operate in either little or big endian. In order for proper operation, both devices must obviously use the same endian format.

As there are two main buses within the HD64404 device and there are a number of different formats of data the following descriptions need to be understood to minimize the need for any software byte re-ordering.

There are three types of blocks within HD64404,

- Blocks which connect only to the register bus.
- Blocks which connect to the pixel bus and have register set-up through the register bus
- Blocks which can transfer data on both the pixel and register bus.

1.9.3 Register Bus

The register bus is a big-endian bus.

Therefore DMAC handles the endian conversion by checking the access size of the data if the data on the MPX/PCI bus between Super H and HD64404 is configured as little. If the access size is byte or word, in HD64404 only HCAN is the case, DMAC will convert the endian.

For PCI mode in HD64404, DMAC will do this automatically because PCI bus is always little.

For MPX mode, Super H first needs to set a register, DMA_External_Select[6] in DMAC module to let DMAC know which endian is used on the MPX/PCI bus.

In case Longword data contains 4 bytes or 2 words data, Byte or Word swapping is supported in the following manner.

1. The majority of blocks can only be accessed as a long word with only one data item occupying that address. I.e. multiple data items are not packed into registers. This effectively means that endian has no meaning on the bus when no data packing is used.
2. A second format is blocks that have multiple data items packed into a larger element i.e. two bytes packed into a word or four bytes or two words packed into a long word. These peripheral modules can be programmed to support either big or little endian packing.

Even though the peripheral modules don't support packing, if the data is being transferred using a DMA channel of DMAC module, then the DMAC can extract data that is smaller than a long-word and pack that into a long-word in either big or little endian format. Additionally, packed data written to the FIFO's of the DMAC can be extracted and written to the peripheral block in its natural data size.

In effect, the DMAC can perform endian conversion into and out of its FIFO's as part of its packing unpacking function for DMA transfers. The related register is DMAnControl Register, bits[15:14, 3:0].

Please see the DMAC section in more detail.

1.9.4 Pixel Bus

The pixel bus can operate in either little or big endian format. This allows data to be stored in the graphics memory in either format. The endian is controlled by setting a bit in the pixel bus connected modules. Internal peripheral modules connected to the pixel bus will then pack their data according to this setting.

The endian format conversion between PCI/MPX bus and pixel bus is as follows:

1. PCI mode

   PCI interface has an endian setting register for PCI DMA master and PCI target respectively.

   a. HD64404 is PCI master

      PCIDCR0 or PCIDCR1 with respect to the DMA channel 0 or 1 and PCIMD5R need to be set.

   b. HD64404 is PCI target

      b-1 PCIMD5R is set.

      Automatically PCI interface assumes LW(LongWord, 32bit) data is four Byte data and execute to byteswap.

      b-2 all byte/word swap is done by software

      Please see the PCI module spec in detail.

   Default value of all those registers is little endian. So if graphics memory is configured as little, no need to set those registers.

   We recommend the configuration that graphics memory is little in PCI mode.

2. MPX mode

   a. MPX PIO access to Graphics Memory

      MPXCTL[20] in MPX i/f needs to be set as 0 when graphics memory is configured as big.(Default).

      MPXCTL[20] in MPX i/f needs to be set as 1 when graphics memory is configured as little

   b. MPX DMA access Graphics Memory

      SYSR Bit 6 and Bit 4 need to be set when Byte/Word swap is necessary between pixel bus and MPX bus.

Endian conversion between PCI/MPX bus and pixel bus is bidirectional.

1.9.5 System Types

There are four types of system configuration

- SH-4 in little endian, MPX connection to HD64404
- SH-4 in little endian, PCI connection to HD64404
- SH-4 in big endian, MPX connection to HD64404
- SH-4 in big endian, PCI connection to HD64404 (not recommended)

This last case has a potential issue with endian support. Although the SH7751 will be set to big endian, the PCI from the SH7751 will still operate in little endian (this is part of PCI definition). The SH7751 has some support for the conversion between the two formats, but when transferring a long-word, the structure of the data within that long-word is unknown, so byte-swapping/word swapping could be incorrect. The suggested solution is to disable the byte/word swapping and where necessary perform the required swapping in software.

1.9.6 Register Bus Summary

Block	Data Size	Data Register Access Size	Data Packing	Endian Setting Register in the Module
DMAC (PIO access to register bus)	8/16/32	32	No	MPX: DMA_External_Select[6] PCI: unconditionally
DMAC (DMA)	8/16/32	32	Yes (Big or Little: To/from external memory)	DMA_n_Control Bit 15: ENDD Bit 14: ENDS Bit 3, 2: CWD Bit 1, 0: CWS
Interrupt_priority	Not applicable	32	Not applicable	—
Memory_interface	Not applicable	32	Not applicable	—
Power control and configuration	Not applicable	32	Not applicable	—
Video_in	Not applicable	32	Not applicable	—
Display_out	Not applicable	32	Not applicable	—
Audio Codec	20/32	32	No	—
SSI	8/16/18/20/22/24/32	32	No	—
SPDIF	24	32	No	—
I2C	8	32	No	—
SPI	8	32	No	—
Expansion bus	8	32	Yes (Big or Little: four bytes)	ex_Mode_Config Bit 4
ATAPI	16	32	Yes (Big or Little: two words, only using DMA mode and PIO FIFO mode)	ATAPI control 2register Bit 1 Default: Big
HCAN-2	16,8	8/16	Yes (Big or Little: two bytes) for Mailbox data	MCR Bit 4 (CAN Endian Mode)
Most Interface	8	32	Yes (Big or Little: Four bytes)	MIM_Module_Config Bit 24 Default: Big
UART	8	32	No	—

Block	Data Size	Data Register Access Size	Data Packing	Endian Setting Register in the Module
IrdA	8	32	No	—
USB Host	Not applicable (Using pixel bus)	32	Not applicable	—
USB function	8	32	No	—
Interrupt input	Not applicable	32	Not applicable	—
Timer/Counter	Not applicable	32	Not applicable	—
PWM	Not applicable	32	Not applicable	—
GPIO	Not applicable	32	Not applicable	—
CSC	16	32	No	—

The blocks that are defined as “not applicable” are blocks that only use the register bus for configuration and where there is no concept of data flow traffic.

1.9.7 Pixel Bus Summary

All transfers on the pixel bus are long-words though there are four byte enables to validate the data. All blocks pack byte and word data into long word quantities.

Block	Control of Endian	Register to be set	Default
MPX i/f (PIO access)	Register internal to MPX i/f	MPXCTL Bit 20	Big
MPX i/f (DMA access)	Register internal to MPX i/f	SYSR Bit 6, Bit 4 (Byte/Word swap in Longword)	Same as MPX bus
PCI i/f (DMA master)	Register internal to PCI i/f	PCIDCR0/1 Bit 10, 9 (Byte/Word swap in Longword) PCIMD5R Bit 0	Little
PCI i/f (Target)	Register internal to PCI i/f	PCIMD5R Bit 0 (For Byte swap in Longword)	Little
Renderer (GE)	Register internal to Memory interface	MCR Bit 4 in Memory interface module	Little
Video In	Register internal to Video In	MC Bit 6	Little
Display Out	Register internal to Display Out	DO_ECR Bit 18	Big
USB	Register internal to USB host	Configuration Control Bit31	Little
ATAPI	Register internal to ATAPI	ATAPI Control 2 register Bit 1 Wordswap 1: Little, 0: Big	Big

1.10 HD64404 Memory Map

1.10.1 MPX Mode

Graphic Memory can be configured as either 64MB space which uses only SH-4 $\overline{CSB}$ (64MB mode) or 128MB space that uses both SH-4 $\overline{CSA}$ and SH-4 $\overline{CSB}$ (128MB mode).

Table 1.5 MPX Mode HD64404 Address Map

Module	Module Address Area (Size in Bytes)	Chip Select		Address (26bit Byte Address)		Clocks Used by Module
		64M Mode	128M Mode	64M Mode	128M Mode	
Graphics Memory	64M Mode: 64MB–64KB 128M Mode: 128MB–64KB	SH $\overline{CSB}$	SH $\overline{CSA}$ & SH $\overline{CSB}$	H'000 0000 to H'3FE FFFF	SH $\overline{CSA}$: H'000 0000 to H'3FF FFFF SH $\overline{CSB}$: H'000 0000 to H'3FE FFF	SD_CLK
Graphics Engine	H'4000	SH $\overline{CSB}$		H'3FF 0000 to H'3FF 3FFF		pix_clk, rbclk
Display output	H'1000	SH $\overline{CSB}$		H'3FF 4000 to H'3FF 4FFF		pix_clk, rbclk
reserved	—	—		H'3FF 5000 to H'3FF 57FF		
USB Host	H'400	SH $\overline{CSB}$		H'3FF 5800 to H'3FF 5BFF		pix_clk, rbclk
USB Function	H'400	SH $\overline{CSB}$		H'3FF 5C00 to H'3FF 5FFF		rbclk
Audio Codec	H'80	SH $\overline{CSB}$		H'3FF 6000 to H'3FF 607F		rbclk
reserved	—	—		H'3FF6080 to H'3FF 60FF		
Timer/Counter	H'40	SH $\overline{CSB}$		H'3FF 6100 to H'3FF613F		rbclk
reserved	—	—		H'3FF6140 to H'3FF 617F		
Interrupt input	H'10	SH $\overline{CSB}$		H'3FF 6180 to H'3FF 618F		rbclk
Expansion bus (registers)	H'40	SH $\overline{CSB}$		H'3FF 6200 to H'3FF 623F		rbclk
Hitachi S/PDIF Interface	H'40	SH $\overline{CSB}$		H'3FF 6240 to H'3FF 627F		rbclk
Memory interface	H'40	SH $\overline{CSB}$		H'3FF 6280 to H'3FF 62BF		pix_clk, rbclk
reserved	—	—		H'3FF62C0 to H'3FF62FF		
Expansion bus (expansion ports)	H'100	SH $\overline{CSB}$		H'3FF 6300 to H'3FF 63FF		rbclk
Video Input	H'100	SH $\overline{CSB}$		H'3FF 6400 to H'3FF 64FF		pix_clk, rbclk
ATAPI	H'100	SH $\overline{CSB}$		H'3FF 6500 to H'3FF 65FF		pix_clk, rbclk
reserved	—	—		H'3FF 6600 to H'3FF 661F		
UART0	H'20	SH $\overline{CSB}$		H'3FF 6620 to H'3FF 663F		rbclk

Module	Module Address Area (Size in Bytes)	Chip Select		Address (26bit Byte Address)		Clocks Used by Module
		64M Mode	128M Mode	64M Mode	128M Mode	
UART1	H'20	SH $\overline{CSB}$		H'3FF 6640 to H'3FF 665F		rbclk
UART2	H'20	SH $\overline{CSB}$		H'3FF 6660 to H'3FF 667F		rbclk
UART3	H'20	SH $\overline{CSB}$		H'3FF 6680 to H'3FF 669F		rbclk
Power Control & Configuration	H'20	SH $\overline{CSB}$		H'3FF 66A0 to H'3FF 66BF		rbclk
PWM	H'20	SH $\overline{CSB}$		H'3FF 66C0 to H'3FF 66DF		rbclk
HSPI0	H'20	SH $\overline{CSB}$		H'3FF 66E0 to H'3FF 66FF		rbclk
HSPI1	H'20	SH $\overline{CSB}$		H'3FF 6700 to H'3FF 671F		rbclk
HSPI2	H'20	SH $\overline{CSB}$		H'3FF 6720 to H'3FF 673F		rbclk
Interrupt Priority	H'20	SH $\overline{CSB}$		H'3FF 6740 to H'3FF 675F		rbclk
GPIO0	H'10	SH $\overline{CSB}$		H'3FF 6760 to H'3FF 676F		rbclk
reserved	—	—		H'3FF 6770 to H'3FF 677F		
Hitachi I2C0	H'40	SH $\overline{CSB}$		H'3FF 6780 to H'3FF 67BF		rbclk
Hitachi I2C1	H'40	SH $\overline{CSB}$		H'3FF 67C0 to H'3FF 67FF		rbclk
MOST Interface	H'80	SH $\overline{CSB}$		H'3FF 6800 to H'3FF 687F		rbclk
SSI0	H'10	SH $\overline{CSB}$		H'3FF 6880 to H'3FF 688F		rbclk
reserved	—	—		H'3FF 6890 to H'3FF 9F		
SSI1	H'10	SH $\overline{CSB}$		H'3FF 68A0 to H'3FF 68AF		rbclk
reserved	-	-		H'3FF 68B0 to H'3FF 68BF		
SSI2	H'10	SH $\overline{CSB}$		H'3FF 68C0 to H'3FF 68CF		rbclk
reserved	—	—		H'3FF 68D0 to H'3FF 68DF		
SSI3	H'10	SH $\overline{CSB}$		H'3FF 68E0 to H'3FF 68EF		rbclk
reserved	—	—		H'3FF 68F0 to H'3FF 68FF		
GPIO1	H'10	SH $\overline{CSB}$		H'3FF 6900 to H'3FF 690F		rbclk
reserved	—	—		H'3FF 6910 to H'3FF 691F		
Color Space Converter	H'20	SH $\overline{CSB}$		H'3FF 6920 to H'3FF 693F		rbclk
reserved	—	—		H'3FF 6940 to H'3FF 697F		
reserved	—	—		H'3FF 6980 to H'3FF 6BFF		
DMAC (registers)	H'200	SH $\overline{CSB}$		H'3FF 6C00 to H'3FF 6DFF		pix_clk, rbclk
reserved	—	—		H'3FF 6E00 to H'3FF 6FF		
DMAC (DMA_0_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7000 to H'3FF 70FF		pix_clk, rbclk
DMAC (DMA_1_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7100 to H'3FF 71FF		pix_clk, rbclk

Module	Module Address Area (Size in Bytes)	Chip Select		Address (26bit Byte Address)		Clocks Used by Module
		64M Mode	128M Mode	64M Mode	128M Mode	
DMAC (DMA_2_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7200 to H'3FF 72FF		pix_clk, rbclk
DMAC (DMA_3_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7300 to H'3FF 73FF		pix_clk, rbclk
DMAC (DMA_4_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7400 to H'3FF 74FF		pix_clk, rbclk
DMAC (DMA_5_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7500 to H'3FF 75FF		pix_clk, rbclk
DMAC (DMA_6_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7600 to H'3FF 76FF		pix_clk, rbclk
DMAC (DMA_7_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7700 to H'3FF 77FF		pix_clk, rbclk
DMAC (DMA_8_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7800 to H'3FF 78FF		pix_clk, rbclk
DMAC (DMA_9_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7900 to H'3FF 79FF		pix_clk, rbclk
DMAC (DMA_10_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7A00 to H'3FF 7AFF		pix_clk, rbclk
DMAC (DMA_11_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7B00 to H'3FF 7BFF		pix_clk, rbclk
DMAC (DMA_12_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7C00 to H'3FF 7CFF		pix_clk, rbclk
DMAC (DMA_13_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7D00 to H'3FF 7DFF		pix_clk, rbclk
DMAC (DMA_14_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7E00 to H'3FF EFF		pix_clk, rbclk
DMAC (DMA_15_FIFO)	H'100*	SH $\overline{CSB}$		H'3FF 7F00 to H'3FF 7FFF		pix_clk, rbclk
HCAN 0	H'800	SH $\overline{CSB}$		H'3FF 8000 to H'3FF 87FF		rbclk
HCAN 1	H'800	SH $\overline{CSB}$		H'3FF 8800 to H'3FF 8FFF		rbclk
MPX i/f	H'10	SH $\overline{CSB}$		H'3FF 9000 to H'3FF 900F		pix_clk
reserved	—	—		H'3FF 9010 to H'3FF FFFF		

Note: * Each FIFO channel has only one accessible read/write port so that the different address accesses during each FIFO address space (= 256 bytes) lead to the same result. DMAC does not decode least 8-bit address for FIFO channel access.

1.10.2 PCI Mode

Graphic Memory can be configured as either 64 MB space or 128 MB space.

Table 1.6 PCI mode HD64404 Address Map (Configuration)

Module	Module Address Area (Size in Bytes)	Configuration Space Address (8-Bit Byte Address)	Clocks Used by Module
PCI IF	H'FC	H'00 to H'FB	rbclk

Table 1.7 PCI mode HD64404 Address Map (I/O space)

Module	Module Address Area (Size in Bytes)	The I/O Space Base Address (8-Bit Byte Address)	Clocks Used by Module
PCI IF	H'FC	H'00 to H'FB	rbclk

Table 1.8 PCI Mode HD64404 Address Map (Memory space)

Module	Module Address Area (Size in Bytes)	The Memory Space Base Address (27-bit Byte Address)	Clocks used by Module
Graphics Memory	0 to 64 MB	H'000 0000 to H'3FF FFFF	SD_CLK
	64 to 128 MB	H'000 0000 to H'7FE FFFF	
Graphics Engine	H'4000	H'7FF 0000 to H'7FF 3FFF	pix_clk, rbclk
Display output	H'1000	H'7FF 4000 to H'7FF 4FFF	pix_clk, rbclk
reserved	—	H'7FF 5000 to H'7FF 57FF	
USB Host	H'400	H'7FF 5800 to H'7FF 5BFF	pix_clk, rbclk
USB Function	H'400	H'7FF 5C00 to H'7FF 5FFF	rbclk
Audio Codec	H'80	H'7FF 6000 to H'7FF 607F	rbclk
reserved	—	H'7FF 6080 to H'7FF 60FF	
Timer/Counter	H'40	H'7FF 6100 to H'7FF 613F	rbclk
reserved	—	H'7FF 6140 to H'7FF 617F	
Interrupt input	H'10	H'7FF 6180 to H'7FF 618F	rbclk
reserved	—	H'7FF 6190 to H'7FF 61FF	
Expansion bus (registers)	H'40	H'7FF 6200 to H'7FF 623F	rbclk
Hitachi S/PDIF Interface	H'40	H'7FF 6240 to H'7FF 627F	rbclk
Memory interface	H'40	H'7FF 6280 to H'7FF 62BF	pix_clk, rbclk
reserved	—	H'7FF 62C0 to H'7FF 62FF	
Expansion bus (expansion ports)	H'100	H'7FF 6300 to H'7FF 63FF	rbclk
Video Input	H'100	H'7FF 6400 to H'7FF 64FF	pix_clk, rbclk
ATAPI	H'100	H'7FF 6500 to H'7FF 65FF	pix_clk, rbclk
reserved	—	H'7FF 6600 to H'7FF 661F	
UART0	H'20	H'7FF 6620 to H'7FF 663F	rbclk
UART1	H'20	H'7FF 6640 to H'7FF 665F	rbclk
UART2	H'20	H'7FF 6660 to H'7FF 667F	rbclk
UART3	H'20	H'7FF 6680 to H'7FF 669F	rbclk
Power Control & Configuration	H'20	H'7FF 66A0 to H'7FF 66BF	rbclk
PWM	H'20	H'7FF 66C0 to H'7FF 66DF	rbclk
HSPI0	H'20	H'7FF 66E0 to H'7FF 66FF	rbclk
HSPI1	H'20	H'7FF 6700 to H'7FF 671F	rbclk
HSPI2	H'20	H'7FF 6720 to H'7FF 673F	rbclk

Module	Module Address Area (Size in Bytes)	The Memory Space Base Address (27-bit Byte Address)	Clocks used by Module
Interrupt Priority	H'20	H'7FF 6740 to H'7FF 675F	rbclk
GPIO0	H'10	H'7FF 6760 to H'7FF 676F	rbclk
reserved	—	H' 7FF 6770 to H'7FF 677F	
Hitachi I2C0	H'40	H'7FF 6780 to H'7FF 67BF	rbclk
Hitachi I2C1	H'40	H'7FF 67C0 to H'7FF 67FF	rbclk
MOST Interface	H'80	H'7FF 6800 to H'7FF 687F	rbclk
SSI0	H'10	H'7FF 6880 to H'7FF 688F	rbclk
reserved	—	H'7FF 6890 to H'7FF 689F	
SSI1	H'10	H'7FF 68A0 to H'7FF 68AF	rbclk
reserved	—	H'7FF 68B0 to H'7FF 68BF	
SSI2	H'10	H'7FF 68C0 to H'7FF 68CF	rbclk
reserved	—	H'7FF 68D0 to H'7FF 68DF	
SSI3	H'10	H'7FF 68E0 to H'7FF 68EF	rbclk
reserved	—	H'7FF 68F0 to H'7FF68FF	
GPIO1	H'10	H'7FF 6900 to H'7FF 690F	rbclk
reserved	—	H'7FF 6910 to H'7FF 691F	
Color Space Converter	H'20	H'7FF 6920 to H'7FF 693F	rbclk
reserved	—	H'7FF 6940 to H'7FF 697F	
reserved	—	H'7FF 6980 to H'7FF 6BFF	
DMAC (registers)	H'200	H'7FF 6C00 to H'7FF 6DFF	pix_clk, rbclk
reserved	—	H'7FF 6E00 to H'7FF 6FFF	
DMAC (DMA_0_FIFO)	H'100*	H'7FF 7000 to H'7FF 70FF	pix_clk, rbclk
DMAC (DMA_1_FIFO)	H'100*	H'7FF 7100 to H'7FF 71FF	pix_clk, rbclk
DMAC (DMA_2_FIFO)	H'100*	H'7FF 7200 to H'7FF 72FF	pix_clk, rbclk
DMAC (DMA_3_FIFO)	H'100*	H'7FF 7300 to H'7FF 73FF	pix_clk, rbclk
DMAC (DMA_4_FIFO)	H'100*	H'7FF 7400 to H'7FF 74FF	pix_clk, rbclk
DMAC (DMA_5_FIFO)	H'100*	H'7FF 7500 to H'7FF 75FF	pix_clk, rbclk
DMAC (DMA_6_FIFO)	H'100*	H'7FF 7600 to H'7FF 76FF	pix_clk, rbclk
DMAC (DMA_7_FIFO)	H'100*	H'7FF 7700 to H'7FF 77FF	pix_clk, rbclk
DMAC (DMA_8_FIFO)	H'100*	H'7FF 7800 to H'7FF 78FF	pix_clk, rbclk
DMAC (DMA_9_FIFO)	H'100*	H'7FF 7900 to H'7FF 79FF	pix_clk, rbclk
DMAC (DMA_10_FIFO)	H'100*	H'7FF 7A00 to H'7FF 7AFF	pix_clk, rbclk
DMAC (DMA_11_FIFO)	H'100*	H'7FF 7B00 to H'7FF 7BFF	pix_clk, rbclk

Module	Module Address Area (Size in Bytes)	The Memory Space Base Address (27-bit Byte Address)	Clocks used by Module
DMAC (DMA_12_FIFO)	H'100*	H'7FF 7C00 to H'7FF 7CFF	pix_clk, rbclk
DMAC (DMA_13_FIFO)	H'100*	H'7FF 7D00 to H'7FF 7DFF	pix_clk, rbclk
DMAC (DMA_14_FIFO)	H'100*	H'7FF 7E00 to H'7FF 7EFF	pix_clk, rbclk
DMAC (DMA_15_FIFO)	H'100*	H'7FF 7F00 to H'7FF 7FFF	pix_clk, rbclk
HCAN 0	H'800	H'7FF 8000 to H'7FF 87FF	rbclk
HCAN 1	H'800	H'7FF 8800 to H'7FF 8FFF	rbclk
reserved	—	H'7FF 9000 to H'7FF FFFF	

Note: * Each FIFO channel has only one accessible read/write port so that the different address accesses during each FIFO address space (= 256 bytes) lead to the same result. DMAC does not decode least 8 bit address for FIFO channel access.

1.11 System Configuration Example

1.11.1 MPX System Example 1

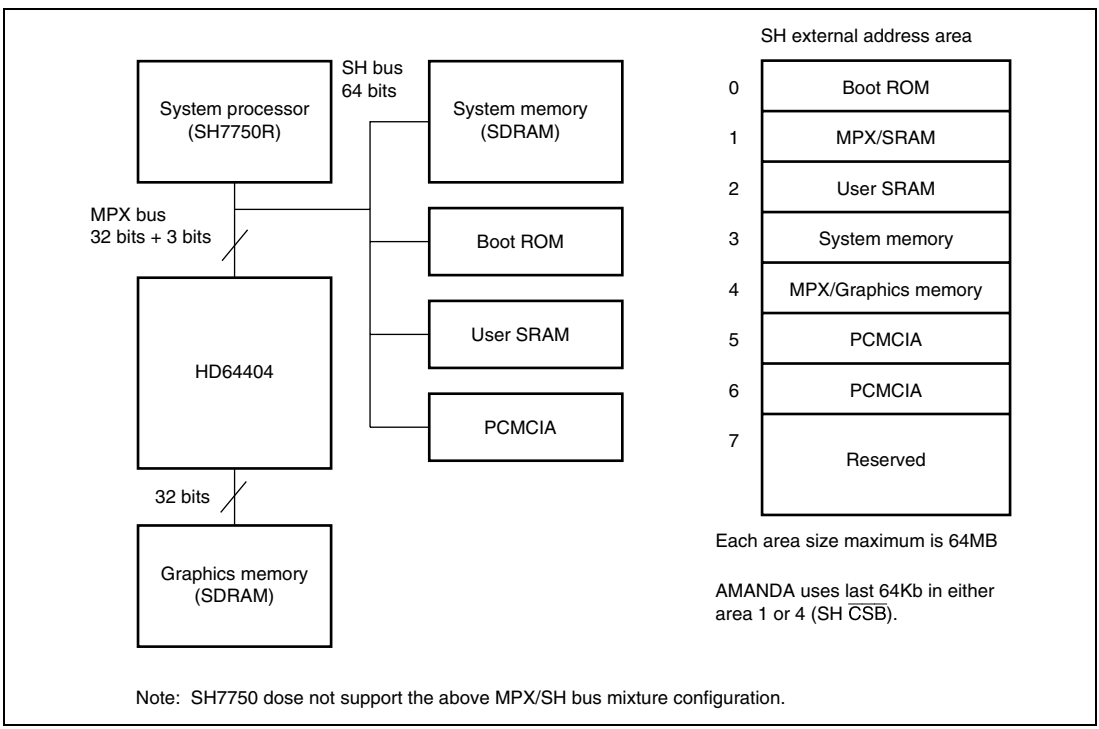


Figure 1.6 MPX System Example 1

1.11.2 MPX System Example 2—UMA (Unified Memory Architecture) Configuration

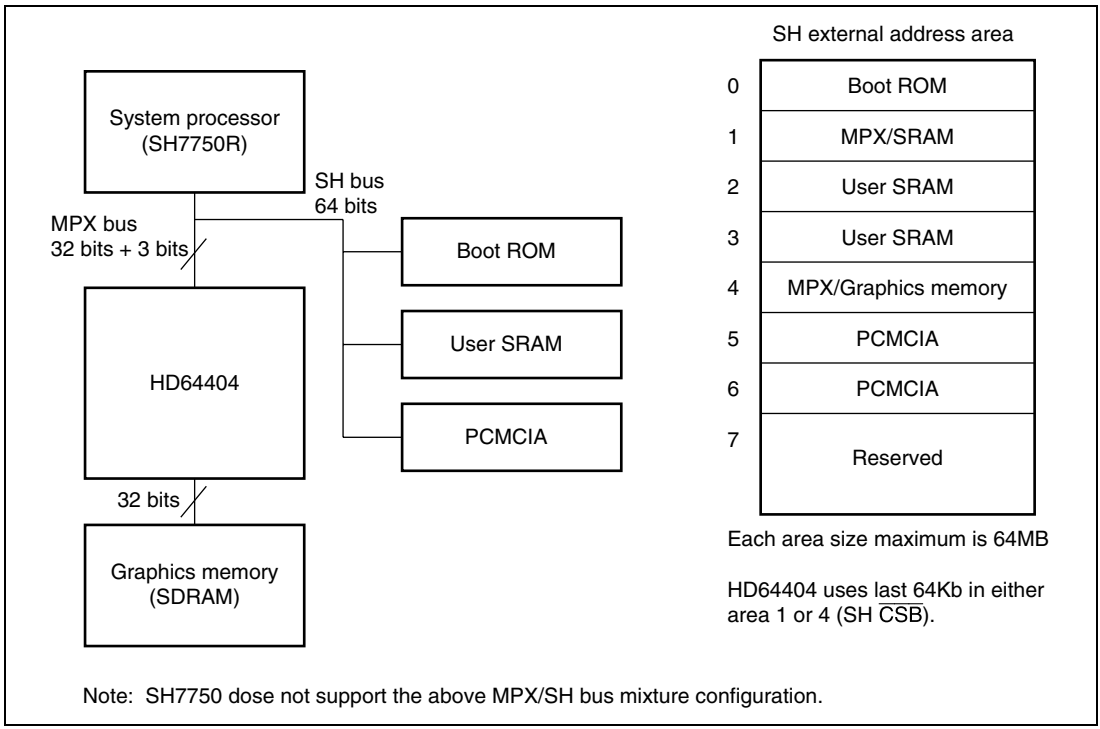


Figure 1.7 MPX System Example 2

1.11.3 PCI System Example

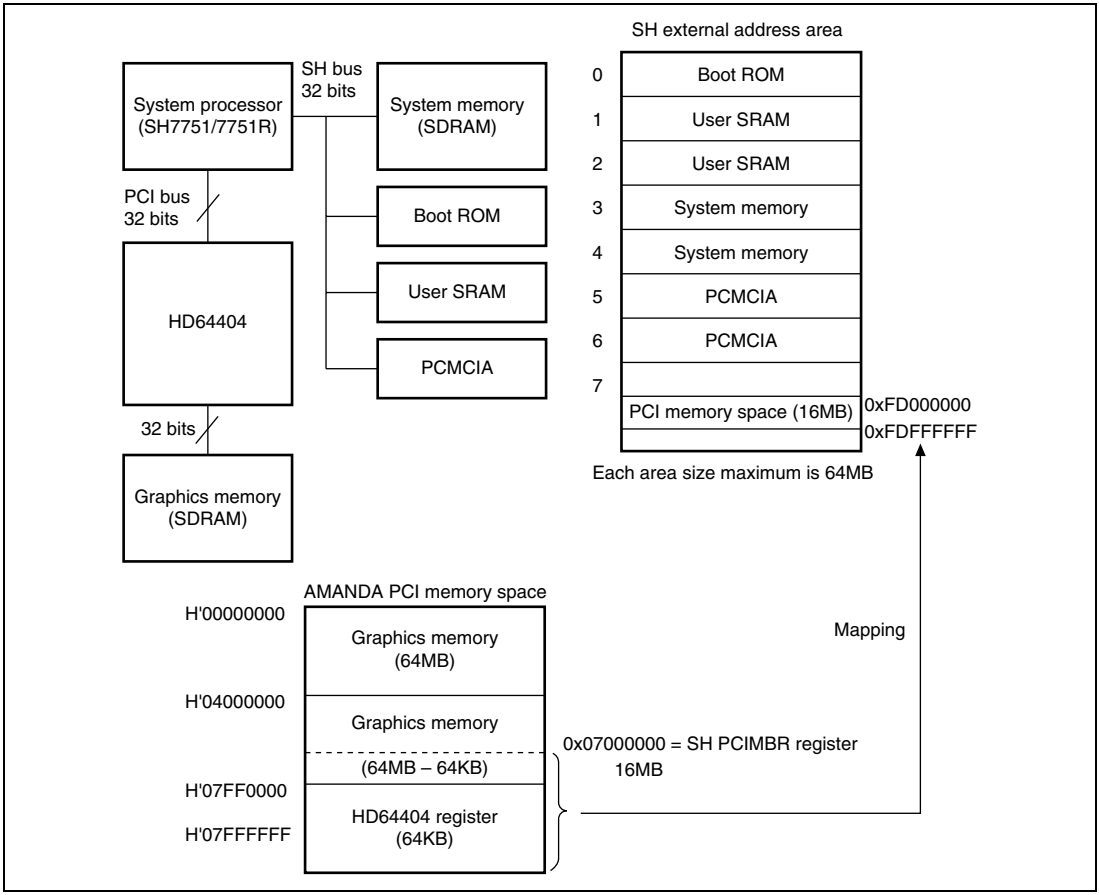


Figure 1.8 PCI System Example

HD64404 PCI I/F Register Setting

PCICONF5	= H'0000 0000	PCILAR0	= H'0000 0000
PCICONF6	= H'0400 0000	PCILAR1	= H'0400 0000
PCILSR0	= H'03F0 0000	PCIPAR	= H'07FF 0000
PCILSR1	= H'03F0 0000	PCIPSR	= H'0000 0000

PCI Configuration Limitation

1. No UMA (Unified Memory Architecture) support
2. Only the last 16MB–64KB Graphics Memory is PIO accessible from System Processor
3. Only software running in Super H privileged mode can access PCI memory space, i.e. if an OS
   runs I/O driver in user mode, this OS can not support PCI configuration.

Section 2 DMAC

2.1 General Description

The DMA Controller (DMAC) organises the data transfer between DMA capable Peripheral Modules on the Register Bus and either External Memory or other Peripheral Modules. External Memory can be either System Memory connected to PCI/MPX bus or Graphics Memory connected to Memory Interface Module via Pixel Bus. The CPU interface type, either PCI Bus or MPX Bus, is selected by a config pin external to the DMAC module.

A dual port RAM FIFO Buffer is internal to DMAC module and is used as temporary storage for receiving and transmitting data. DMAC has 16 DMA channels. The RAM FIFO Buffer can be configured as 16 FIFO Channel Buffers associated with 16 DMA channels. Each FIFO Channel Buffer size is configurable. The RAM FIFO Buffer allocates one port on the Pixel Bus so that Graphics Memory data can be always accessed without port reservation. The other port of RAM FIFO Buffer will be shared dynamically between PCI/MPX bus and Register Bus.

DMAC acts as MPX bus slave and Register Bus master. In case of PCI bus, each DMA channel can be independently configured as either bus master or bus slave to the PCI Bus.

Peripheral Modules on the Register Bus have pre-allocated DMA Request Numbers. There are 31 DMA Request Numbers. A DMA Request Number is used as an index to a register where register address within the specific Peripheral Module is specified for either source or destination of DMA.

DMAC supports four types of DMA modes:

Master DMA mode: DMAC acts as the bus master to either PCI bus or Pixel bus. DMAC will transfer data between a Peripheral Module and an External Memory location. Since DMAC cannot be MPX bus master, data transfer to or from System Memory is supported only in PCI Bus. Double buffer scheme is supported for DMA data transfer to or from External Memory location in this mode.

Slave DMA mode: DMAC acts as bus slave to either PCI or MPX bus. DMAC will transfer data between a Peripheral Module and a FIFO Channel Buffer. System Processor will transfer data between the FIFO Channel Buffer and a System Memory location using PIO access. Each FIFO Channel Buffer can be monitored for data availability (i.e. not empty) for single read, some amount of data availability for burst read, space availability (i.e. not full) for single write and some amount of space availability for burst write operations. Buffer status can be read in status registers and data/space availability can be reported as interrupts.

Inter-module DMA mode: DMAC will transfer data between Peripheral Modules.

For master DMA, slave DMA and Inter-module DMA modes, DMAC will conduct the DMA operation.

External DMA mode: System Processor DMAC will conduct the DMA operation using a DMA channel of this DMAC module and transfers data between a Peripheral Module and a System Memory location. In this mode, CPU interface must be MPX Bus. Only one DMA channel can be configured as External DMA mode.

DMA mode and applicable buses are summarized in Table 2.9.

DMA data transfer will occur between Primary DMA Address and Secondary DMA Address. Primary DMA address is always a register address of a Peripheral Module and Secondary DMA Address is either an External Memory address or another register address of a Peripheral Module. Data transfer direction as to whether data is transferred from Primary DMA Address to Secondary DMA Address or vice versa will be specified in Direction flag (DR) of channel control registers as well as other data transfer options. Data transfer always occurs from source address to destination address, so in other words, DR flag decides whether primary DMA address is source address or destination address and vice versa for secondary DMA address.

If Peripheral Module data width is specified as either 8 bits or 16 bits, data will be packed to 32-bit when data is transferred from Peripheral Module to External Memory and unpacked from 32-bits when data is transferred from External Memory to Peripheral Module as well as Endian conversion is taken place during packing/unpacking. Data transfer length is counted as packed data and should be multiple of 4 bytes. If Peripheral Module data width is specified as 32-bits, no Endian conversion takes place between PCI/MPX, Pixel and Register Buses whatever the System Processor's Endian mode is set to.

Three priority encoders are supported for Pixel Bus, PCI/MPX Bus and Register Bus respectively.

Completion of DMA operation, called Terminal Count event for a FIFO Channel Buffer to/from a External Memory location data transfer and Peripheral Terminal Count event for a Peripheral Module to/from a FIFO Channel Buffer data transfer, can be monitored either by interrupt or status flag or both.

A DMA channel supports data transfer to or from a Peripheral Module location. A DMA channel in master mode supports data transfer to or from an External Memory location.

In this specification, the character "n" refers to a specific DMA channel in the range 0 to 15. The character "q" refers to a DMA Request Number in the range 0 to 30.

2.2 Features

- 16 DMA Channels.
- 32-bit system address space.
- 65,532 ,i.e. 64 K to 4 bytes maximum length DMA transfers in fixed length mode. No length limit in continuous data transfer mode.
- RAM FIFO Buffer is directly accessible from System Processor or PCI master.

- RAM FIFO Buffer can be configured as 16 FIFO Channel Buffers of 256 bytes long each, i.e. 4 KB total.
- Endian conversion and data packing/unpacking are supported between PCI/MPX/Pixel Bus and Register Bus for 8-bit and 16-bit data on Register Bus.
- Four DMA mode support: Master DMA mode, Slave DMA Mode, Inter-module DMA mode and External DMA.
- Provide direct PIO access from the System Processor to Peripheral Module Registers.
- Round Robin Priority encoder for PCI/MPX Bus, Pixel Bus and Register Bus
- Double buffer scheme support for DMA transfers to or from External Memory
- Five types of DMA status/interrupt flags are reported: FIFO Status, FIFO Burst Status, Peripheral Terminal Count status, Terminal Count status, Peripheral Request status
- Interrupts on each channel can be enabled and masked individually.
- Sophisticated buffer control for enhanced data transfer error handling: Peripheral Request Status Monitoring, DMA FIFO Flush operations are supported.

2.3 Limitations

1. All DMA addresses should be at longword (i.e.4 bytes) boundary and data transfer length should be multiple of longword.
2. Possible FIFO Channel Buffer size is 16, 32 and 64 longwords. FIFO Channel Buffer Start address should be at 64-longword boundaries.
3. DMA Request Number 31 is reserved and should not be specified.
4. Request Numbers assigned for RAM FIFO Buffer should be unique. In other words, no Request Number should be used multiple times among all the FIFO Buffer Channels.
5. Not all bus configurations can be applicable to four DMA modes. For data transfer between System Memory and Peripheral Module PCI and MPX Bus has restriction below,

   MPX Bus: Master DMA mode is not supported. Either Slave DMA or External DMA mode should be used.

   PCI Bus: External DMA mode is not supported. Either Slave DMA or Master DMA should be used.

   Please refer to Table 2.9, 2.10.
6. Not all DMA control flags and fields described in DMA n Control Register can be applicable to four DMA modes. Please refer to Table 2.12 and 2.13 in “2.11 Appendix 3 DMA Mode Parameters”.
7. In continuous data transfer mode, DMAC does not count data transferred. In Master DMA, Slave DMA mode, destination Peripheral Module should support data transfer completion interrupt. In Inter-module DMA mode, destination Peripheral Module should support DMA stop function. Whether data counting and data transfer completion interrupt are necessary or not depend on device use case. See Table 2.13 for detail.

8. In continuous data transfer and Master DMA mode, DMA n Length Register should be larger than FIFO Channel Buffer size.
9. Data pack/unpack function is not supported in Inter-module DMA Transfer mode. External Memory data width should be always 32-bits. Peripheral Module data width can be either 8 or 16-bits for this function.
10. Only one External DMA channel could be allocated in RAM FIFO Buffer.
11. In External DMA mode, data transfer length should always be multiple of Burst Size specified in DMA n Length Register. Burst size in DMA n Length Register should be equal to burst data transfer size specified in System Processor, esp. in case of SH7751, CHCR register, i.e., 1, 2 or 8 longwords.
12. For system processor PIO access to FIFO Channel Buffer in Slave DMA mode, FIFO Channel Buffer can be configured as either one of read or write access. No read write access should be done with a FIFO channel buffer. Also FIFO Burst Status or FIFO Status should be checked before every access to FIFO Channel Buffer. If read on empty FIFO buffer, System Processor will be held off in case of PCI Bus and get wrong data in case of MPX Bus. If write on full FIFO buffer, System Processor will be held off in case of PCI Bus and write nothing in case of MPX Bus.
13. FIFO Channel Buffer resizing should not be done dynamically. The DMAC does not recognise the contents of this register until the command DMA_FIFO_Flush is applied to the relevant channel n.
14. PT and PCOUNT do not work in continuous data transfer mode. Also PT and PCOUNT do not work in Master DMA mode where DR bit = 0. TC and MCOUNT only work in Master DMA mode where DR bit = 0 or in continuous data transfer mode. See Table 2.13 for details.
15. Burst Size is only valid in Slave DMA mode and External DMA mode.
16. FIFO Burst Interrupt and FIFO Status Interrupt as well as respective status flag only works in Slave DMA mode.
17. DMA Length does not work in continuous data transfer of Slave DMA and Inter-module DMA modes.

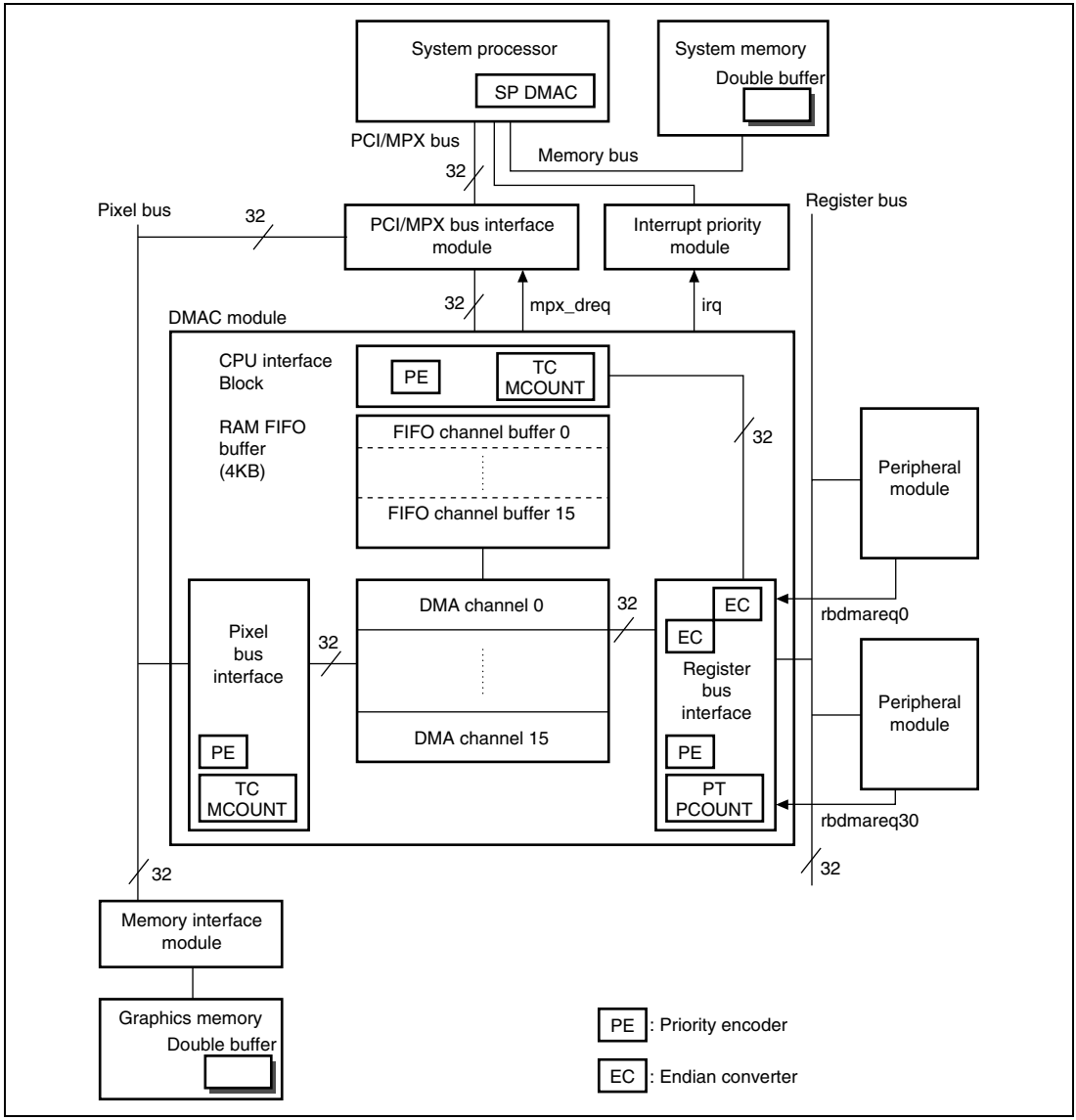


Figure 2.1 System Diagram

2.4 Digital Inputs/Outputs

Table 2.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From
Register Bus	—		Access to registers	
Pixel Bus	—		Access to Graphics Memory	
rbdmareq	31	In	DMA request from each Peripheral that supports DMA	Register Bus
mpx_dreq	1	Out	DMA request to System Processor in MPX mode	MPX bus interface module
irq	1	Out	Interrupt Line	Interrupt priority module

Note: The Register Bus and Pixel Bus provide their own clocks.

2.5 Address Map

Table 2.2 shows the base address for each Peripheral Modules, DMAC Registers and RAM FIFO Buffers in the 64 Kbyte System Processor IO address space.

The register addresses of each Peripheral Module are defined in their respective module specifications. The DMAC registers are shown in Tables 2.3 through 2.5.

Table 2.2 Module Select Addresses

Name	Base Address in Bytes	Size in Bytes
Graphics Engine	H'0000	H'4000
Display Out	H'4000	H'1000
USB Host	H'5800	H'400
USB Function	H'5c00	H'400
Audio Codec Interface	H'6000	H'80
Timer	H'6100	H'40
Interrupt Input	H'6180	H'10
Expansion Bus	H'6200	H'40
SPDIF	H'6240	H'40
Memory Interface	H'6280	H'40
Expansion Bus Peripheral	H'6300	H'100
Video Input	H'6400	H'100
ATAPI	H'6500	H'100
UART0	H'6620	H'20
UART1	H'6640	H'20
UART2	H'6660	H'20
UART3	H'6680	H'20
Power control & Configuration	H'66a0	H'20
Pulse Width Mdulation	H'66c0	H'20
SPI0	H'66e0	H'20
SPI1	H'6700	H'20
SPI2	H'6720	H'20
Interrupt Priority	H'6740	H'20
GPIO0	H'6760	H'10
I^2C0	H'6780	H'40
I^2C1	H'67c0	H'40
MOST Interface	H'6800	H'80
SSI0	H'6880	H'10
SSI1	H'68a0	H'10
SSI2	H'68c0	H'10
SSI3	H'68e0	H'10
GPIO1	H'6900	H'10

Name	Base Address in Bytes	Size in Bytes
Color Space Converter	H'6920	H'20
DMAC registers	H'6c00	H'200
DMA Channel registers	H'6e00	H'200
DMA_0_FIFO	H'7000	H'100
DMA_1_FIFO	H'7100	H'100
DMA_2_FIFO	H'7200	H'100
DMA_3_FIFO	H'7300	H'100
DMA_4_FIFO	H'7400	H'v100
DMA_5_FIFO	H'7500	H'100
DMA_6_FIFO	H'7600	H'100
DMA_7_FIFO	H'7700	H'100
DMA_8_FIFO	H'7800	H'100
DMA_9_FIFO	H'7900	H'100
DMA_10_FIFO	H'7a00	H'100
DMA_11_FIFO	H'7b00	H'100
DMA_12_FIFO	H'7c00	H'100
DMA_13_FIFO	H'7d00	H'100
DMA_14_FIFO	H'7e00	H'100
DMA_15_FIFO	H'7f00	H'100
HCAN 0	H'8000	H'800
HCAN 1	H'8800	H'800
Reserved for MPX I/F	H'9000	H'10

2.5.1 DMAC Registers

Table 2.3 DMAC Register Addresses

Module Addresses (Bytes)	Register Name	R/W	Access Size (Bits)
H'6C00	DMA External Select	R/W	32
H'6C04	DMA Status	R/W	32
H'6C08	DMA FIFO Status	R/W	32
H'6C0C	DMA FIFO Flush	W	32
H'6C10	PIO Monitor	R/W	32
H'6C14	PIO Monitor Status	R/W	32
H'6C18	DMA Peripheral Request Status	R/W	32
H'6C1C	DMA Interrupt Source	R (Bits 15 to 0) R/W (Bits 31 to 16)	32
H'6C20 to H'6CFC	Reserved	—	—
H'6D00	DMA 0 Request Address	R/W	32
H'6D04	DMA 1 Request Address	R/W	32
H'6D08	DMA 2 Request Address	R/W	32
H'6D0C	DMA 3 Request Address	R/W	32
H'6D10	DMA 4 Request Address	R/W	32
H'6D14	DMA 5 Request Address	R/W	32
H'6D18	DMA 6 Request Address	R/W	32
H'6D1C	DMA 7 Request Address	R/W	32
H'6D20	DMA 8 Request Address	R/W	32
H'6D24	DMA 9 Request Address	R/W	32
H'6D28	DMA 10 Request Address	R/W	32
H'6D2C	DMA 11 Request Address	R/W	32
H'6D30	DMA 12 Request Address	R/W	32
H'6D34	DMA 13 Request Address	R/W	32
H'6D38	DMA 14 Request Address	R/W	32
H'6D3C	DMA 15 Request Address	R/W	32
H'6D40	DMA 16 Request Address	R/W	32
H'6D44	DMA 17 Request Address	R/W	32
H'6D48	DMA 18 Request Address	R/W	32
H'6D4C	DMA 19 Request Address	R/W	32

Module Addresses (Bytes)	Register Name	R/W	Access Size (Bits)
H'6D50	DMA 20 Request Address	R/W	32
H'6D54	DMA 21 Request Address	R/W	32
H'6D58	DMA 22 Request Address	R/W	32
H'6D5C	DMA 23 Request Address	R/W	32
H'6D60	DMA 24 Request Address	R/W	32
H'6D64	DMA 25 Request Address	R/W	32
H'6D68	DMA 26 Request Address	R/W	32
H'6D6C	DMA 27 Request Address	R/W	32
H'6D70	DMA 28 Request Address	R/W	32
H'6D74	DMA 29 Request Address	R/W	32
H'6D78	DMA 30 Request Address	R/W	32
H'6D7C to H'6DFC	Reserved	—	—

2.5.2 DMA Channel Registers

Table 2.4 DMA Channel Register Addresses

Address (Bytes)	Register Name	R/W	Access Size (Bits)
H'6E00	DMA 0 Start Address	R/W	32
H'6E04	DMA 0 Length	R/W	32
H'6E08	DMA 0 Control	R/W	32
H'6E0C	DMA 0 RAM Buffer Size	R/W	32
H'6E10	DMA 1 Start Address	R/W	32
H'6E14	DMA 1 Length	R/W	32
H'6E18	DMA 1 Control	R/W	32
H'6E1C	DMA 1 RAM Buffer Size	R/W	32
H'6E20	DMA 2 Start Address	R/W	32
H'6E24	DMA 2 Length	R/W	32
H'6E28	DMA 2 Control	R/W	32
H'6E2C	DMA 2 RAM Buffer Size	R/W	32
H'6E30	DMA 3 Start Address	R/W	32
H'6E34	DMA 3 Length	R/W	32
H'6E38	DMA 3 Control	R/W	32
H'6E3C	DMA 3 RAM Buffer Size	R/W	32
H'6E40	DMA 4 Start Address	R/W	32
H'6E44	DMA 4 Length	R/W	32
H'6E48	DMA 4 Control	R/W	32
H'6E4C	DMA 4 RAM Buffer Size	R/W	32
H'6E50	DMA 5 Start Address	R/W	32
H'6E54	DMA 5 Length	R/W	32
H'6E58	DMA 5 Control	R/W	32
H'6E5C	DMA 5 RAM Buffer Size	R/W	32
H'6E60	DMA 6 Start Address	R/W	32
H'6E64	DMA 6 Length	R/W	32
H'6E68	DMA 6 Control	R/W	32
H'6E6C	DMA 6 RAM Buffer Size	R/W	32
H'6E70	DMA 7 Start Address	R/W	32
H'6E74	DMA 7 Length	R/W	32
H'6E78	DMA 7 Control	R/W	32
H'6E7C	DMA 7 RAM Buffer Size	R/W	32
H'6E80	DMA 8 Start Address	R/W	32

Address (Bytes)	Register Name	R/W	Access Size (Bits)
H'6E84	DMA 8 Length	R/W	32
H'6E88	DMA 8 Control	R/W	32
H'6E8C	DMA 8 RAM Buffer Size	R/W	32
H'6E90	DMA 9 Start Address	R/W	32
H'6E94	DMA 9 Length	R/W	32
H'6E98	DMA 9 Control	R/W	32
H'6E9C	DMA 9 RAM Buffer Size	R/W	32
H'6EA0	DMA 10 Start Address	R/W	32
H'6EA4	DMA 10 Length	R/W	32
H'6EA8	DMA 10 Control	R/W	32
H'6EAC	DMA 10 RAM Buffer Size	R/W	32
H'6EB0	DMA 11 Start Address	R/W	32
H'6EB4	DMA 11 Length	R/W	32
H'6EB8	DMA 11 Control	R/W	32
H'6EBC	DMA 11 RAM Buffer Size	R/W	32
H'6EC0	DMA 12 Start Address	R/W	32
H'6EC4	DMA 12 Length	R/W	32
H'6EC8	DMA 12 Control	R/W	32
H'6ECC	DMA 12 RAM Buffer Size	R/W	32
H'6ED0	DMA 13 Start Address	R/W	32
H'6ED4	DMA 13 Length	R/W	32
H'6ED8	DMA 13 Control	R/W	32
H'6EDC	DMA 13 RAM Buffer Size	R/W	32
H'6EE0	DMA 14 Start Address	R/W	32
H'6EE4	DMA 14 Length	R/W	32
H'6EE8	DMA 14 Control	R/W	32
H'6EEC	DMA 14 RAM Buffer Size	R/W	32
H'6EF0	DMA 15 Start Address	R/W	32
H'6EF4	DMA 15 Length	R/W	32
H'6EF8	DMA 15 Control	R/W	32
H'6EFC	DMA 15 RAM Buffer Size	R/W	32
H'6F00	DMA 0 MCOUNT	R	32
H'6F04	DMA 0 PCOUNT	R	32
H'6F08 to H'6F0C	Reserved	—	—
H'6F10	DMA 1 MCOUNT	R	32
H'6F14	DMA 1 PCOUNT	R	32
H'6F18 to H'6F1C	Reserved	—	—

Address (Bytes)	Register Name	R/W	Access Size (Bits)
H'6F20	DMA 2 MCOUNT	R	32
H'6F24	DMA 2 PCOUNT	R	32
H'6F28 to H'6F2C	Reserved	—	—
H'6F30	DMA 3 MCOUNT	R	32
H'6F34	DMA 3 PCOUNT	R	32
H'6F38 to H'6F3C	Reserved	—	—
H'6F40	DMA 4 MCOUNT	R	32
H'6F44	DMA 4 PCOUNT	R	32
H'6F48 to H'6F4C	Reserved	—	—
H'6F50	DMA 5 MCOUNT	R	32
H'6F54	DMA 5 PCOUNT	R	32
H'6F58 to H'6F5C	Reserved	—	—
H'6F60	DMA 6 MCOUNT	R	32
H'6F64	DMA 6 PCOUNT	R	32
H'6F68 to H'6F6C	Reserved	—	—
H'6F70	DMA 7 MCOUNT	R	32
H'6F74	DMA 7 PCOUNT	R	32
H'6F78 to H'6F7C	Reserved	—	—
H'6F80	DMA 8 MCOUNT	R	32
H'6F84	DMA 8 PCOUNT	R	32
H'6F88 to H'6F8C	Reserved	—	—
H'6F90	DMA 9 MCOUNT	R	32
H'6F94	DMA 9 PCOUNT	R	32
H'6F98 to H'6F9C	Reserved	—	—
H'6FA0	DMA 10 MCOUNT	R	32
H'6FA4	DMA 10 PCOUNT	R	32
H'6FA8 to H'6FAC	Reserved	—	—
H'6FB0	DMA 11 MCOUNT	R	32
H'6FB4	DMA 11 PCOUNT	R	32
H'6FB8 to H'6FBC	Reserved	—	—
H'6FC0	DMA 12 MCOUNT	R	32
H'6FC4	DMA 12 PCOUNT	R	32
H'6FC8 to H'6FCC	Reserved	—	—
H'6FD0	DMA 13 MCOUNT	R	32
H'6FD4	DMA 13 PCOUNT	R	32
H'6FD8 to H'6FDC	Reserved	—	—
H'6FE0	DMA 14 MCOUNT	R	32

Address (Bytes)	Register Name	R/W	Access Size (Bits)
H'6FE4	DMA 14 PCOUNT	R	32
H'6FE8 to H'6FEC	Reserved	—	—
H'6FF0	DMA 15 MCOUNT	R	32
H'6FF4	DMA 15 PCOUNT	R	32
H'6FF8 to H'6FFC	Reserved	—	—

2.5.3 DMA FIFO Channels

Table 2.5 DMA FIFO Buffer Addresses

Address (Bytes)	Register Name	R/W	Access Size (Bits)
H'70XX	DMA 0 FIFO	Either Read or Write	32
H'71XX	DMA 1 FIFO	Either Read or Write	32
H'72XX	DMA 2 FIFO	Either Read or Write	32
H'73XX	DMA 3 FIFO	Either Read or Write	32
H'74XX	DMA 4 FIFO	Either Read or Write	32
H'75XX	DMA 5 FIFO	Either Read or Write	32
H'76XX	DMA 6 FIFO	Either Read or Write	32
H'77XX	DMA 7 FIFO	Either Read or Write	32
H'78XX	DMA 8 FIFO	Either Read or Write	32
H'79XX	DMA 9 FIFO	Either Read or Write	32
H'7AXX	DMA 10 FIFO	Either Read or Write	32
H'7BXX	DMA 11 FIFO	Either Read or Write	32
H'7CXX	DMA 12 FIFO	Either Read or Write	32
H'7DXX	DMA 13 FIFO	Either Read or Write	32
H'7EXX	DMA 14 FIFO	Either Read or Write	32
H'7FXX	DMA 15 FIFO	Either Read or Write	32

Legend:

XX: Don't care

2.5.4 DMA Request Numbers

The DMA Request Number allocated to each Peripheral Module is shown in Table 2.6. The DMA Request Numbers are pre-allocated and fixed. They should be programmed in DMA n Control Register and DMA n Start Address Registers correctly for a given DMA channel.

Multiplexed Peripheral Module such as SSI2 has the same DMA Request Number. Similarly a Peripheral Module can have only one DMA Request Number shared between operating modes, such as transmit mode and receive mode. The correct register address, depending on the operating mode selected should be programmed into the "DMA q Request Address" Register in the DMAC.

Peripherals sharing the same Request Number cannot use the DMA at the same time. Actually those Peripherals are exclusively configured in the system by either config pin or by configuration register of Peripheral Module.

Please refer to the respective Peripheral Module specification for information on shared DMA requests between modes.

Table 2.6 DMA Request Number Lists

DMA Peripheral Name	Register Name of Address Programmed into DMA_q_Request_address	DMA Request Shared between Peripherals	DMA Request Number
MOST Packet Tx	MIM_PacketTx	Yes	0
Expansion Bus 0	Expansion Port 0	Yes	0
MOST Packet Rx	MIM_PacketRx	Yes	1
Expansion Bus 1	Expansion Port 1	Yes	1
MOST Stream 1	MIM_Stream1	No	2
MOST Stream 2	MIM_Stream2	No	3
MOST Stream 3	MIM_Stream3	No	4
MOST Stream 4	MIM_Stream4	No	5
SSI0	Transmit Data Register 0/ Receive Data Register 0	No	6
SSI1	Transmit Data Register 1/ Receive Data Register 1	No	7
SSI2	Transmit Data Register 2/ Receive Data Register 2	Yes	8
SSI3	Transmit Data Register 3/ Receive Data Register 3	No	9
SPDIF Tx	Transmitter DMA Audio Data	No	10
SPDIF Rx	Receiver DMA Audio Data	No	11
SPI0 Tx	Transmit Buffer Register 0	Yes	12

DMA Peripheral Name	Register Name of Address Programmed into DMA_q_Request_address	DMA Request Shared between Peripherals	DMA Request Number
SPI0 Rx	Receive Buffer Register 0	No	13
SPI1 Tx	Transmit Buffer Register 1	No	14
SPI1 Rx	Receive Buffer Register 1	No	15
Color Space Converter 0	Indata	No	16
Color Space Converter 1	Outdata	No	17
Audio Codec Tx	TX DMA Register	No	18
Audio Codec Rx	RX DMA Register	No	19
UART0 Tx	Transmit Data Register 0	No	20
UART0 Rx	Receive Data Register 0	No	21
UART1 Tx	Transmit Data Register 1	No	22
UART1_Rx	Receive Data Register 1	No	23
UART2 Tx	Transmit Data Register 2	No	24
UART2 Rx	Receive Data Register 2	No	25
UART3 Tx	Transmit Data Register 3	Yes	26
SPI2 Tx	Transmit Buffer Register 2	Yes	26
UART3 Rx	Receive Data Register 3	Yes	27
SPI2 Rx	Receive Buffer Register 2	Yes	27
ATAPI	Data Register	No	28
USB Function1	EP1 data register	No	29
USB Function2	EP2 data register	No	30

2.6 Register Description

There is a set of registers for each of the 16 DMA channels, and one set of registers for each of the 31 DMA Peripheral Module requests, which is located in the address space of the PCI or MPX bus.

Legends for Register Description

Initial value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, read value undefined.

2.6.1 DMA Channel Registers (DMA Channel Number n = 0 to 15)

DMA n Start Address Register

In Master DMA mode

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Start Address															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Start Address														0	0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	Start Address	0	R/W	**DMA Start Address in External Memory**
1, 0	0	0	R/W	Always 0 should be specified

In Inter-module DMA mode

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												Request Number				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R/W	**Reserved**
4 to 0	Request Number	0	R/W	**Secondary DMA Request Number** This register specifies the Secondary DMA Address using DMA channel n. When data transfer mode is Master DMA mode, this register specifies the External Memory address in longwords. When data transfer mode is Inter-module DMA mode, this register specifies the secondary DMA Request Number of Peripheral Module. Actual Secondary DMA Address is specified in the associated DMA q Request Address Register. Primary DMA address is always a register address of a Peripheral Module and specified in DMA n Control Register. DR flag of the DMA n Control Register decides data transfer direction. Start Address should be at longword (4 byte) boundary. The DMA Request Number of 31 is reserved and should not be used. In Slave DMA and External DMA mode, this register is not used.

DMA n Length Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											Burst Size					
Initial:	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMA Length														0	0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	—	R	**Reserved**
21 to 16	Burst Size	0	R/W	**Burst Size** Number of longwords available in FIFO Channel Buffer n or number of longwords of space available in FIFO Channel Buffer n that causes the burst flag to be asserted. Burst size is only used to set the threshold at which the FIFO burst status flag in DMA FIFO Status Register is set. If this register is set at half the buffer size defined in DMA n RAM Buffer Size then the flag will operate as a half full or half empty flag or interrupt. The flag or interrupt operates as a part full flag if the channel is set to write to Peripheral Module and a part empty if the channel is set up as read from Peripheral Module. Maximum Burst Size is 63 longwords. Burst Size is only valid in Slave DMA mode and External DMA mode.
15 to 2	DMA Length	0	R/W	**DMA Length** DMA Length specifies length of DMA transfer in longwords.
1, 0	0	0	R/W	Always 0 should be specified

Maximum size is H'3FFF longwords, i.e.16,383 longwords or 65,532 bytes. Note that the last four bytes out of 64 Kbytes could not be transferred in one DMA transfer.

DMA Length does not work in continuous data transfer of Slave DMA and Inter-module DMA modes.

DMA n Control Register

This register specifies various DMA controlling functions below:

- Specifies the DMA mode along with DMA External Select Register
- Specifies the Primary DMA Address and data transfer direction
- Specifies the data pack/unpack as well as Endian conversion options
- Specifies the data transfer mode, i.e. continuous data transfer mode or fixed length data transfer mode
- Specifies which interrupts and buffer status should be reported
- And lastly trigger the DMA transfer operation by writing into DTRA and RTRA flags

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
											PTEN	CSEL				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENDD	ENDS	FBEN	FSEN	TCEN	DBEN	ML	RBEN	MM	DTRA	DR	RTRA	CWD		CWS	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	0	R	**Reserved**
20 to 16	CSEL	0	R/W	**Channel Select (CSEL)** Specify the primary DMA Request Number of DMA channel n. Primary DMA Address is specified in the associated DMA q Request Address Register. Data transfer direction is specified in DR flag of this register. Secondary DMA address is specified in DMA n Start Address Register. The DMA Request Number of 31 is reserved and should not be used. Unique Request Number should be specified in each DMA Channel.

Bit	Bit Name	Initial Value	R/W	Description
15	ENDD	0	R/W	**Endian**
14	ENDS	0	R/W	These flags specify External Memory byte data alignments of the source and destination data. If Little Endian is specified, the DMAC will re-align the data on the fly, except for the Inter-module DMA mode in which Endian conversion will not be conducted. **Bit 15: Destination Data alignment (ENDD)** This flag specifies the Endian of data in the destination address. If destination is not External memory, this flag is invalid and should be 0. 1: Big Endian 0: Little Endian **Bit 14: Source Data alignment (ENDS)** This flag specifies the Endian of data in the source address. If source is not External memory, this flag is invalid and should be 0. 1: Big Endian 0: Little Endian

Bit	Bit Name	Initial Value	R/W	Description
21	PTEN	0	R/W	**Interrupt Enable**
13	FBEN	0	R/W	These flags are used to specify whether Terminal Count interrupt, Peripheral Terminal Count, FIFO Status interrupt and FIFO Burst Status interrupt will be asserted or not when the specific condition will be met.
12	FSEN	0	R/W	
11	TCEN	0	R/W	

Bit 21: Peripheral Terminal Count Interrupt Enable (PTEN)

1: PT Interrupt will be asserted when DMA data transfer between a Peripheral Module and a FIFO Channel Buffer n is completed.

0: Peripheral Terminal Count interrupt is disabled.

Peripheral Terminal Count interrupt does not work in continuous data transfer mode. Peripheral Terminal Count also does not work in Master DMA mode where DR bit equals to 0. So in those modes, PTEN should be disabled.

Bit 13: FIFO Burst Interrupt Enable (FBEN)

1: FIFO Burst interrupt will be asserted when either burst read or burst write operation condition is met for the FIFO Channel Buffer n.

0: FIFO Burst interrupt is disabled.

FBEN flag only works in Slave DMA mode. So in other modes, FBEN should be disabled.

Bit 12: FIFO Status Interrupt Enable (FSEN)

1: FIFO Status interrupt will be asserted when either single read or single write operation condition is met for the FIFO Channel Buffer n.

0: FIFO Status interrupt is disabled.

FSEN flag only works in Slave DMA mode. So in other modes, FSEN should be disabled.

Bit 11: Terminal Count Interrupt Enable (TCEN)

1: Terminal Count Interrupt will be asserted when DMA data transfer between an External Memory location and a FIFO Channel Buffer n is completed

0: Terminal Count Interrupt is disabled.

Terminal Count interrupt only works in Master DMA mode where DR bit equals to 0 or in continuous data transfer mode. So in other modes, TCEN should be disabled.

Bit	Bit Name	Initial Value	R/W	Description
10	DBEN	0	R/W	**Double Buffer Enable (DBEN)** This flag sets the data transfer mode to be either continuous or fixed length and can apply to all DMA modes except External DMA mode. 1: Continuous data transfer mode 0: Fixed length data transfer mode For master DMA mode, DBEN requires double buffer in External Memory for continuous data transfer but for other DMA modes, DBEN means continuous data transfer and does not requires double buffer. In continuous data transfer mode, DMA Length specified in the DMA n Length does not specify the actual data transfer length. In continuous data transfer mode, DMAC does not count data transferred. So continuous data transfer mode can be used only when the conditions below are met, In Master DMA, Slave DMA mode, destination Peripheral Module should support both DMA counter and data transfer completion interrupt. Data counting should be done both in Peripheral Module and software. In Inter-module DMA mode, destination Peripheral Module should support DMA stop function. Whether data counting and data transfer completion interrupt are necessary or not depend on device use case. See Table 2.13 for detail. • Master DMA mode In continuous data transfer mode, data is transferred continuously between External Memory and a Peripheral Module. The DMAC will continuously cycle between two buffers in the External Memory which is arranged as a contiguous External Memory block, Terminal Count event occurs at the end of each buffer transfer and then switch automatically to the other buffer. The start address and buffer length are set in registers DMA n Start Address and DMA n Length respectively. The System Processor must write to or read from the data buffer that is not being accessed. Both buffers will have the same length, buffer 1 is at address (DMA n Start Address) and buffer 2 is at (DMA n Start Address + DMA n Length). DMA n Length should be larger than FIFO Channel Buffer size. In fixed length data transfer mode, the Terminal Count event occurs when the DMA address counter reaches the end of the buffer and the transfer will stop.

Bit	Bit Name	Initial Value	R/W	Description
10	DBEN	0	R/W	• Slave DMA mode In continuous data transfer mode, data transfer between FIFO Channel Buffer and Peripheral Module is continuous. Clearing DBEN flags will terminate the data transfer after DMA n Length of data have been transferred. In fixed length data transfer mode, DMA n Length of data will be transferred. • Inter-module DMA mode In continuous data transfer mode, data transfer will be continuously conducted between two Peripheral Modules. Clearing DBEN flags will terminate the data transfer after DMA n Length of data have been transferred. In fixed length data transfer mode, DMA n Length of data will be transferred. This data transfer only succeeds when the source Peripheral Module can wait until the FIFO Channel Buffer space is available because DMAC will simply does not respond to DMA request from the source Peripheral Module when FIFO Channel Buffer is full. Also the destination Peripheral Module should be able to wait until FIFO Channel Buffer is not empty. Any kind of flow control scheme between source and destination Peripheral Modules should be conducted outside DMAC module if necessary. • External DMA mode Data transfer mode is not supported. DBEN flag should be 0.
9	ML	0	R/W	**External Memory Location (ML)** In Master DMA mode, this flag specifies where the external memory connected. This flag is ignored in other DMA modes. 1: System Memory connected to PCI/MPX Bus 0: Graphics Memory connected to Memory Interface Module via Pixel Bus

Bit	Bit Name	Initial Value	R/W	Description
8	RBEN	0	R/W	**Register Bus Enable (RBEN)** If this flag is set to 1, then the data transfer will be in Inter-module DMA mode for DMA Channel Buffer n. 1: Specifies the Inter-module DMA mode, i.e. inter-Peripheral Modules 0: Specifies that the DMA will be between Peripheral Module and External Memory Note that RBEN flag should be 0 in other than Inter-module DMA mode. Endian conversion is not supported in Inter-module DMA mode. If RBEN flag is set to 1, RTRA flag should be set to 1 also.
7	MM	0	R/W	**Master Mode (MM)** Each channel in the DMAC can be configured either as master or Slave DMA mode in terms of PCI/MPX bus operation. In Master DMA mode, DMAC module controls the flow of data between the RAM FIFO Buffer and External Memory. In Slave DMA mode, the FIFO Channel Buffers are directly accessible by either the System Processor or by a device on the PCI Bus. In Slave DMA mode, the external device is responsible for controlling the data transfer and the channel start address register value is ignored. 1: DMA Channel is in Master DMA mode. 0: DMA Channel is not in master DMA mode Note that MM flag should be 0 in either Inter-module DMA mode or External DMA mode.

Bit	Bit Name	Initial Value	R/W	Description
6	DTRA	0	R/W	**Start Master DMA Transfer (DTRA)** In Master DMA mode, when this flag is set to 1, DMAC initiates the DMA data transfer from the FIFO Channel Buffer n to External Memory. • Master DMA can operate in both data transfer modes specified by DBEN flag. • Fixed length data transfer (DBEN = 0) Writing a '1' to DTRA flag starts the master DMA transfer. This flag will automatically be cleared to 0 at the end of the transfer when the specified DMA transfer is completed. Clearing this flag will stop the data transfer. If DMA length of 0 is specified, this flag is cleared to 0 without DMA transfer. 1: Start master DMA Transfer between FIFO Channel Buffer and External Memory 0: Stop data transfer and reset address counters (not necessary in normal DMA completion) • Continuous data transfer (DBEN = 1) Writing a '1' to DTRA flag starts the master DMA transfer, the transfer will be continuous until DTRA flag is cleared to 0 by the System Processor. Data transfer will then stop at the end of the buffer that it is currently transferring and address counters set to their initial state. In order to stop continuous data transfer, then the DMA stop operation should be conducted. Please refer to "2.12 HD64404 DMA Driver Design Note". 1: Start master DMA Transfer between FIFO Channel Buffer and External Memory 0: Stop data transfer at end of current buffer and reset address counters. In either slave DMA or Inter-module DMA or External DMA mode, DTRA flag should be 0.
5	DR	0	R/W	**Direction (DR)** Direction flag specifies the data transfer direction between the Primary DMA Address specified in CSEL field of DMA n Control Register and the Secondary DMA Address specified in DMA n Start Address Register (See table 2.7).

Bit	Bit Name	Initial Value	R/W	Description
4	RTRA	0	R/W	**Start Register Bus Transfer (RTRA)** Writing a '1' to this flag initiates the DMA data transfer in Register Bus. In fixed length data transfer mode, RTRA flag will be cleared to 0 when data transfer completed. In continuous data transfer mode, RTRA flag will not be cleared to 0 by DMAC. If RTRA is cleared to 0 during a DMA then the transfer will stop. If DMA is stopped midway through a transfer then there could be data left in the FIFO Channel. Refer to error handling section in "2.12 HD64404 DMA Driver Design Note". The RTRA flag controls the transfer of data across the Register Bus. In case of data being transferred from a Peripheral Module, the transfer will start as soon as this flag is set to 1 and will continue until the FIFO Channel Buffer is full. In the case of data transfer to a Peripheral Module then this will only happen if this flag is set to 1 and there is data in the FIFO Channel Buffer. 1: Start Register Bus DMA transfer 0: Stop data transfer and reset address counters (not necessary in normal DMA completion)
3, 2	CWD	0	R/W	**Bits 3, 2: Channel Width Destination (CWD)**
1, 0	CWS	0	R/W	**Bits 1, 0: Channel Width Source (CWS)** CWS specifies the data width in the source DMA address and CWD specifies the data width in the destination DMA address except for Inter-module DMA mode. 00: 32 bits 01: 16 bits 11: 8 bits CWD is only valid if destination address is Primary DMA Address and CWS is only valid if source address is Primary DMA Address. If CWD or CWS is invalid, then value should be 00. If CWD is valid, ENDS is valid and CWS/ENDD are invalid. If CWS is valid, ENDD is valid and CWD/ENDS are invalid. In Inter-module DMA mode, all CWD/CWS/DNDD/ENDS are invalid.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	CWD	0	R/W	These fields in conjunction with the ENDS/ENDD flags control the transfer of data to and from External Memory and Peripheral Module, which are not 32-bit wide when data packing and unpacking is required. To accomplish data packing and unpacking the Endian of the data and Peripheral Module size is needed so that data alignment and order is correct across the data transfer. All DMA transfers are longword (32-bit wide), however, when transferring data to or from a Peripheral Module that is not 32-bit wide, the data can be packed in External Memory, these flags indicate the data width of a given Peripheral Module. Data packing or unpacking can be performed by the DMAC, if the data size on the Peripheral Module is set to 16 bits or 8 bits. The External Memory will always be 32 bits. In the case of an 8-bit Peripheral Module data transferred to 32-bit External Memory, CWS = 11 and CWD = 00. Depending on the destination Endian, in this case External Memory, each byte will be written to its correct position in the FIFO channel buffer n, the data is then transferred to External Memory in the correct format. In the case of an 8-bit Peripheral Module data received from 32-bit External Memory, CWS = 00 and CWD = 11. Each longword will be written to the FIFO channel buffer and depending on the source Endian, in this case External Memory, the data will then be transferred a byte at time from the correct position in the FIFO channel buffer n to the Peripheral Module. When unpacked data is being written to or read from Peripheral Modules, the data is aligned to the least significant word or byte. The ENDS/ENDD Flags of this register set the Endian of the data in the source and destination DMA addresses. Peripherals that do not have 32-bit registers will perform byte swapping if necessary. The CWS and CWD flags control data packing unpacking, so if this function is not required even though the Peripheral Module data width is not 32 bits then these flags should be set to 32 bits. Endian conversion is not supported for data transfer in Inter-module DMA mode. In this case Peripheral Modules that are connected together must be matched in both Endian and data size. The size flags must be set to 32 bits to indicate that the transfers are treated as 32-bit wide, however in this case not all the bits will be valid.
1, 0	CWS	0	R/W	

Table 2.7 Data Transfer Direction

DR flag	Source DMA address	Destination DMA address
1	Secondary DMA Address	Primary DMA Address
0	Primary DMA Address	Secondary DMA Address

Note: Channel Clear Operation

If the channel is reprogrammed for another DMA Request Number then this register should be written to with the previous DMA Request Number programmed into the CSEL field and all other flags and fields should be cleared to 0. The channel can then be programmed with the new DMA Request Number. This ensures that previous settings for the channel are cleared to 0

DMA n RAM Buffer Size Register

This register controls the size and location of the FIFO Channel Buffers in the RAM FIFO Buffer allocated to DMA channel n.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					Start								Length			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11 to 4	Start	0	R/W	**FIFO Channel Buffer Start Address** Specifies the start address offset of FIFO Channel Buffer n within the RAM FIFO Buffer in 4 longwords. Multiply the value in Start field by 4 to give the start address of the buffer in longwords.
3 to 0	Length	0	R/W	**FIFO Channel Buffer Length** Specifies the length of FIFO Channel Buffer n within the RAM FIFO Buffer in 4 longwords. Multiply the value in Length field by 4 and add 4 to give length of buffer in longwords. Buffer length can be either 16, 32 or 64 longwords. There is a limitation to a FIFO Channel Buffer start address and length. Start Address and Length combination is summarized in table below. (See table 2.8.)

Table 2.8 DMA Channel Buffer Addresses and lengths

DMA Channel Number	Start address value	Possible Lengths Value/meaning		
0	H'00	H'3/64 bytes	H'7/128 bytes	H'f/256 bytes
1	H'10	H'3/64 bytes	H'7/128 bytes	H'f/256 bytes
:	:	:	:	:
14	H'E0	H'3/64 bytes	H'7/128 bytes	H'f/256 bytes
15	H'F0	H'3/64 bytes	H'7/128 bytes	H'f/256 bytes

Warning: FIFO Channel Buffer resizing should not be done dynamically during DMA transfer. The DMAC does not recognise the contents of this register until the command DMA_FIFO_Flush is applied to the relevant channel n.

DMA n FIFO Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFO Data															
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFO Data															
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FIFO Data	—	R/W	**DMA_Channel_n** FIFO Channel Buffer n data can be accessed directly from this register. The Direction (DR) flag of the respective DMA n Control Register controls the function of this register. FIFO status flag and FIFO burst flag can be read from the DMA FIFO Status Register. • Direction (DR) flag: 1 FIFO Channel Buffer n is in input mode; a write to this register will place a new word into the buffer. • Direction (DR) flag: 0 FIFO Channel Buffer n is in output mode; a read from this register will remove the current data word to be replaced by the next word in the buffer.

DMA n PCOUNT Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCOUNT															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 0	PCOUNT	0	R/W	**DMA Peripheral Count (PCOUNT)** The number of DMA transfers in bytes that have completed between a Peripheral Module and the FIFO Channel Buffer n can be read in PCOUNT field. This register will be cleared to 0 after the DMA transfer has completed, i.e. if PCOUNT is not equal to 0, DMA transfer between the Peripheral Module and FIFO Channel Buffer n is not completed. PCOUNT does not work in continuous data transfer mode. PCOUNT also does not work in Master DMA mode where DR bit equals to 0.

DMA n MCOUNT Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																MCOUNT
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MCOUNT															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16 to 0	MCOUNT	0	R/W	**DMA External Memory Count (MCOUNT)** The number of DMA transfers in bytes that have completed between the FIFO Channel Buffer n and External Memory can be read in MCOUNT field. This register will be cleared to 0 after the DMA transfer has completed, i.e. if MCOUNT is not equal to 0, DMA transfer between External Memory and FIFO Channel Buffer n is not completed. In continuous data transfer mode (DBEN = 1), bit 16 indicates which of the two buffers are currently transferring data. Bit 16: 0 Data transfer is using Buffer 1 Bit 16: 1 Data transfer is using Buffer 2. For a description of continuous data transfer mode please refer to DMA n Control Register (DBEN flag) description in this section. MCOUNT only works in Master DMA mode where DR bit equals to 0 or in continuous data transfer mode.

2.6.2 DMA Peripheral Request Registers (DMA Request Number q = 0 to 30)

DMA q Request Address Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			Address													
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved**
13 to 0	Address	0	R/W	**DMA_q_Request_Address** Specifies the Peripheral Module register address in the DMAC address space, using DMA Request Number q in longwords. To calculate this address, the byte address offset of the data register in the Peripheral Module requesting service should be added to the byte base address of the Peripheral Module and then shift right by two bits. See Table 2.6 for Request Address of each Request Number.

2.6.3 DMA Configuration and Status Registers

DMA External Select Register

This register specifies the External DMA related setting except DDEN flag. External DMA can only apply to MPX Bus. b Enable (DDEN) can apply to both PCI/MPX Buses.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										MEND	DDEN	EDMA	EDMS			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6	MEND	0	R/W	**MPX Bus Endian (MEND) MPX Bus only** If MPX Bus is used, this flag sets the Endian of System Processor PIO accesses into DMAC address space. 1: Big Endian 0: Little Endian
5	DDEN	0	R/W	**Dummy DMA Enable (DDEN) PCI/MPX Buses** If this flag is set to 1, then the next Register Bus access to a Peripheral Module on the Register Bus will look to the Peripheral Module the same as a DMA cycle. There are some Peripheral Module whose DMA request will make spurious DMA request and has no mechanism programmed into the Peripheral Module to clear its own DMA request. This condition can be checked in RS flag in DMA Peripheral Request Status. This flag must be programmed to 0 after the access to the Peripheral Module register. 1: PIO access programmed for dummy DMA cycle 0: PIO access does not use dummy DMA cycle

Bit	Bit Name	Initial Value	R/W	Description
4	EDMA	0	R/W	**Start External DMA(EDMA) MPX Bus only** If this flag is set to 1, then External DMA will start to transfer data to and from the FIFO Channel Buffers specified in EDMS field. If this flag is cleared to 0, then the DMA transfer will stop. If DMA is stopped midway through a transfer then there could be data left in the FIFO Channel Buffer. Refer to error handling section in "2.12 HD64404 DMA Driver Design Note". Special care should be taken for external DMAC error recovery procedure. Refer to the System Processor manual how to recover from the error. The system supports only one external DMA channel when MPX Bus is used. External DMA channel can be allocated to any of one the 16 DMA channels. If the EDMA flag is set to 1 then channel addressed by EDMS field will be configured as for a normal DMA transfer. The FIFO Channel Buffer status flags are routed to the external DMA controller as a DMA request. In this way, System Processor conducts flow control to FIFO Channel. 1: Start External DMA data transfer 0: Stop External DMA data transfer (not necessary in normal DMA completion)
3 to 0	EDMS	0	R/W	**Eternal DMA mode Channel Selected (EDMS) MPX Bus only** Specifies the DMA channel that has been selected for System Processor DMA. If there is no External DMA mode used, EDMS channel number can be arbitrarily chosen but EDMA flag should always be 0.

DMA Status Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PT15	PT14	PT13	PT12	PT11	PT10	PT9	PT8	PT7	PT6	PT5	PT4	PT3	PT2	PT1	PT0
Initia:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TC15	TC14	TC13	TC12	TC11	TC10	TC9	TC8	TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
Initia:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit	Bit Name	Initial Value	R/W	Description
31	PT15	0	R/WC0	**Peripheral Terminal Count Status (PT)**
30	PT14	0	R/WC0	Specifies Peripheral Terminal Count status of the respective DMA channel. PTn flag corresponds to DMA Channel n status. 1: Peripheral Terminal Count condition is met. 0: Peripheral Terminal Count condition is not met
29	PT13	0	R/WC0	
28	PT12	0	R/WC0	
27	PT11	0	R/WC0	
26	PT10	0	R/WC0	This respective flag for a given channel will be set to 1 when DMA transfers in DMA n Length Register have occurred between the Peripheral Module and FIFO Channel Buffer on the respective channel. Clearing the relevant flag to 0 clears the Peripheral Terminal Count event.
25	PT9	0	R/WC0	
24	PT8	0	R/WC0	
23	PT7	0	R/WC0	
22	PT6	0	R/WC0	
21	PT5	0	R/WC0	If Peripheral Terminal Count interrupt is enabled in DMA n Control Register, Peripheral Terminal Count interrupt is raised when Peripheral Terminal Count condition met. Clearing the relevant flag to 0 will clear the interrupt.
20	PT4	0	R/WC0	
19	PT3	0	R/WC0	
18	PT2	0	R/WC0	
17	PT1	0	R/WC0	PT does not work in continuous data transfer mode. PT also does not work in Master DMA mode where DR bit equals to 0.
16	PT0	0	R/WC0	

Bit	Bit Name	Initial Value	R/W	Description
15	TC15	0	R/WC0	**Terminal Count Status (TC)**
14	TC14	0	R/WC0	Specifies Terminal Count status of the respective DMA channel. TCn flag corresponds to DMA Channel n status.
13	TC13	0	R/WC0	
12	TC12	0	R/WC0	1: Terminal Count condition is met.
11	TC11	0	R/WC0	0: Terminal Count condition is not met
10	TC10	0	R/WC0	This respective flag for a given channel will be set to 1 when DMA transfers in DMA n Length Register have occurred between the FIFO Channel Buffers and External Memory on the respective channel.
9	TC9	0	R/WC0	
8	TC8	0	R/WC0	
7	TC7	0	R/WC0	
6	TC6	0	R/WC0	Clearing the relevant flag to 0 in the status register clears the Terminal Count status.
5	TC5	0	R/WC0	
4	TC4	0	R/WC0	If Terminal Count interrupt is enabled in DMA n Control Register, Terminal Count interrupt is raised when Terminal Count condition met. Clearing the relevant flag to 0 will clear the interrupt.
3	TC3	0	R/WC0	
2	TC2	0	R/WC0	
1	TC1	0	R/WC0	
0	TC0	0	R/WC0	TC only works in Master DMA mode where DR bit equals to 0 or in continuous data transfer mode.

DMA FIFO Status Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
Initial:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FS15	FS14	FS13	FS12	FS11	FS10	FS9	FS8	FS7	FS6	FS5	FS4	FS3	FS2	FS1	FS0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit	Bit Name	Initial Value	R/W	Description
31	FB15	1	R/WC0	**FIFO Burst Status (FB)**
30	FB14	1	R/WC0	FIFO Channel Buffer status for burst read/write operation. FBn flag corresponds to DMA Channel n status.
29	FB13	1	R/WC0	
28	FB12	1	R/WC0	1: FIFO Channel Buffer is ready to operate with burst read/write operation. Precisely,
27	FB11	1	R/WC0	
26	FB10	1	R/WC0	A length of data specified in DMA n Length Register or more are available in FIFO Channel Buffer n for read operation, i.e. data transfer from Peripheral Module to System Memory
25	FB9	1	R/WC0	
24	FB8	1	R/WC0	
23	FB7	1	R/WC0	
22	FB6	1	R/WC0	Or
21	FB5	1	R/WC0	A length of space specified in DMA n Length Register or more are available in FIFO Channel Buffer n for write operation, i.e. data transfer from System Memory to Peripheral Module
20	FB4	1	R/WC0	
19	FB3	1	R/WC0	
18	FB2	1	R/WC0	
17	FB1	1	R/WC0	0: FIFO Channel Buffer is not ready to operate with burst read/write operation.
16	FB0	1	R/WC0	
				If FIFO Burst Status interrupt is enabled in DMA n Control Register, FIFO Burst Status interrupt is raised when FIFO Channel Buffer becomes ready. Clearing the relevant flag to 0 will clear the interrupt. If the cause of the interrupt is not satisfied then the interrupt will re-occur after it has been cleared.
				FB only works in Slave DMA mode.
				For practical and performance reason, either only FIFO Burst Status or FIFO Status condition should be used for a DMA Channel. Also using both FIFO Status and FIFO Burst Status for different DMA channels is not recommended.
				If FIFO Burst Status is necessary for a DMA Channel, using FIFO Burst Status for all the DMA Channels is recommended to reduce the complexity of device driver design.

Bit	Bit Name	Initial Value	R/W	Description
15	FS15	0	R/WC0	**FIFO Status (FS)**
14	FS14	0	R/WC0	Specifies FIFO Channel Buffer status for single read/write operation. FSn flag corresponds to DMA Channel n status.
13	FS13	0	R/WC0	
12	FS12	0	R/WC0	1: FIFO Channel Buffer is ready to operate with single read/write operation. Precisely,
11	FS11	0	R/WC0	
10	FS10	0	R/WC0	One longword or more data are available in FIFO Channel Buffer n for read operation, i.e. data transfer from Peripheral Module to System memory
9	FS9	0	R/WC0	
8	FS8	0	R/WC0	Or
7	FS7	0	R/WC0	
6	FS6	0	R/WC0	One longword or more spaces are available in FIFO Channel Buffer n for write operation, i.e. data transfer from System Memory to Peripheral Module.
5	FS5	0	R/WC0	
4	FS4	0	R/WC0	
3	FS3	0	R/WC0	0: FIFO Channel Buffer is not ready to operate with single read/write operation.
2	FS2	0	R/WC0	
1	FS1	0	R/WC0	If FIFO Status interrupt is enabled in DMA n Control Register, FIFO Status interrupt is raised when FIFO Channel Buffer becomes ready. Clearing the relevant flag to 0 will clear the interrupt. If the cause of the interrupt is not satisfied then the interrupt will re-occur after it has been cleared.
0	FS0	0	R/WC0	
				FS only works in Slave DMA mode

DMA Interrupt Source Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8	CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CI15	CI14	CI13	CI12	CI11	CI10	CI9	CI8	CI7	CI6	CI5	CI4	CI3	CI2	CI1	CI0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	CM15	0	R/W	**Channel Interrupt Mask (CM)**
30	CM14	0	R/W	Specifies whether DMA Channel Interrupt should assert irq pin and report to System Processor or not. CMn flag corresponds to DMA Channel n interrupt mask.
29	CM13	0	R/W	
28	CM12	0	R/W	
27	CM11	0	R/W	1: DMA Channel n interrupt will not assert irq pin.
26	CM10	0	R/W	0: DMA Channel n interrupt will assert irq pin.
25	CM9	0	R/W	Channel Interrupt Mask does not affect the Channel Interrupt Status. Clearing CM flag to 0 does not clear the interrupt itself.
24	CM8	0	R/W	
23	CM7	0	R/W	
22	CM6	0	R/W	
21	CM5	0	R/W	
20	CM4	0	R/W	
19	CM3	0	R/W	
18	CM2	0	R/W	
17	CM1	0	R/W	
16	CM0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
15	CI15	0	R	**Channel Interrupt (CI)**
14	CI14	0	R	Indicates interrupt status of DMA channels. CIn flag corresponds to DMA Channel n interrupt status. 1: Interrupt on channel 0: No interrupt on channel The OR of all the interrupt sources for a respective channel that are currently requesting an interrupt can be identified by reading this register. The possible interrupt sources are Terminal Count (TC), Peripheral Terminal count (PT), FIFO Status (FS) and FIFO Burst Status (FB) and are described in the DMA Status and DMA FIFO Status Register descriptions in this section. Specifying respective flags in DMA n Control Register enable respective interrupts. Channel Interrupt Mask does not affect the Channel Interrupt Status.
13	CI13	0	R	
12	CI12	0	R	
11	CI11	0	R	
10	CI10	0	R	
9	CI9	0	R	
8	CI8	0	R	
7	CI7	0	R	
6	CI6	0	R	
5	CI5	0	R	
4	CI4	0	R	
3	CI3	0	R	
2	CI2	0	R	
1	CI1	0	R	
0	CI0	0	R	

DMA FIFO Flush Register

Writing a 1 to the respective flag in this register will empty the respective FIFO Channel Buffer n of its contents without completing the DMA data transfer. Writing H'0ffff will empty all 16 FIFO Channel Buffers.

After every completion of DMA, FIFO Flush should be issued to clear all outstanding DMAC states. See the procedure below.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initia:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FF15	FF14	FF13	FF12	FF11	FF10	FF9	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/	-/
	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1	WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	—	**Reserved**
15	FF15	0	—/WC1	**FIFO Flush Operation**
14	FF14	0	—/WC1	FFn flag corresponds to DMA channel n flush operation. Writing 1 will empty the relevant FIFO and configure the relevant channel's FIFO Channel Buffer n size after the register DMA n RAM Buffer Size has been programmed. 1: Specify DMA channel flush operation 0: No operation FIFO Flush Operation does not complete the DMA data transfer from FIFO Channel Buffer to External Memory or Peripheral Module. Before issuing FIFO Flush, any outstanding status flags should be cleared by writing all 0 except CSEL field to DMA n Control. So correct FIFO Flush Operation sequence is below, 1. Write to DMA n Control All flags and fields equals to 0 other than CSEL(should equal to Primary DMA address) 2. Write to DMA FIFO Flush FFn bit is set to 1 DMA FIFO Flush will end within the register write cycle. This register is write only.
13	FF13	0	—/WC1	
12	FF12	0	—/WC1	
11	FF11	0	—/WC1	
10	FF10	0	—/WC1	
9	FF9	0	—/WC1	
8	FF8	0	—/WC1	
7	FF7	0	—/WC1	
6	FF6	0	—/WC1	
5	FF5	0	—/WC1	
4	FF4	0	—/WC1	
3	FF3	0	—/WC1	
2	FF2	0	—/WC1	
1	FF1	0	—/WC1	
0	FF0	0	—/WC1	

DMA Peripheral Request Status Register

The Peripheral Module DMA requests are latched within the DMAC as 1 until they are serviced by the DMAC. When a Peripheral Module DMA request has been serviced by the DMAC its value is reset to 0. This function is transparent to the user, however, there are some Peripheral Modules who will make spurious DMA Requests and checking these flags is required after DMA data transfer is completed normally. By writing a 1 to the relevant flag will clear the internal latched condition of the Peripheral Module request.

This command should be issued before starting DMA as DMA Pre-processing. Refer to DMA Pre-processing procedure in "2.12 HD64404 DMA Driver Design Note" for details.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		RS30	RS29	RS28	RS27	RS26	RS25	RS24	RS23	RS22	RS21	RS20	RS19	RS18	RS17	RS16
Initia:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RS15	RS14	RS13	RS12	RS11	RS10	RS9	RS8	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
Initia:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1	R/ WC1

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	**Reserved** Should not be set
30 to 0	RS30 to RS0	All 0	R/WC1	**DMA Request Peripheral Status (RS)** 1: When read, peripheral module is requesting a DMA cycle. 0: No operation. Writing a 1 to the respective flag will clear the internal DMA Peripheral Module request.

PIO Monitor Register

This register controls the PIO monitor function, which if enabled will restrict the Register Bus accesses of the System Processor.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																TCNT
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNT											PUP				EPM
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	Reserved
16 to 5	TCNT	0	R/W	**PIO Monitor Threshold Count (TCNT)** The value programmed into this field sets the threshold count at which the System Processor will be held off. The optimum value for this count will depend on the bus activity on the Register Bus and the acceptable maximum latency that can be tolerated by any Peripheral Module.
4 to 1	PUP	0	R/W	**PIO Monitor Up Count (PUP)** The value programmed into this field set the number that the PIO monitor counter will increment by on every System Bus clock when the System Processor is accessing the Register Bus, the PIO monitor counter will decrement on every System Bus clock when the System Processor is not accessing the Register Bus.
0	EMP	0	R/W	**Enable PIO Monitor (EPM)** 1: Enable PIO Monitor function 0: Disable PIO Monitor function Refer to PIO Bus Activity Monitor of functional description section for functional details. If EPM is set to 1 with TCNT = 0 and PUP = 0, PIO access from System Processor will be held off until EPM is cleared to 0.

PIO Monitor Status Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MCNT											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11 to 0	MCNT	0	R	**PIO Monitor Count (MCNT)** PIO monitor counter can be monitored via this register to be used to monitor PIO accesses from the System Processor of the Register Bus. This counter will operate at all times but will only cause the System Processor to be held off, if the function is enabled by the EPM flag in the PIO Monitor Register. Refer to PIO Bus Activity Monitor of functional description section for functional details.

2.7 Functional Description

The DMAC controls the transfer of data using DMA between Peripheral Modules on the Register Bus and External Memory or another Peripheral Module. All transfers are via RAM FIFO Buffer in the DMAC, which are organised as FIFO Channel Buffers. Data can be written to or read from the FIFO Channel Buffer using burst access and then transferred to or from Peripheral Modules on the Register Bus using single accesses. In this way the system busses are used efficiently and not slowed up by multiple accesses to slow Peripheral Modules. The FIFO Channel Buffers give the Peripheral Modules burst capability and by locating the buffers in one place they can be more easily configured by software. The FIFO Channel Buffers can be accessed directly from the System Processor or by PCI bus masters or the DMAC can address the External Memory directly by being a PCI master or by requesting access to the Pixel Bus.

When being accessed directly from external devices the FIFO Status flags have to be read to establish if there is data available to be read or space available to be written to. The FIFO Channel Buffers have status flags to show data transfer is ready. The burst flag is set to 1 when a pre-programmed size of data is available to be either read or written. If interrupts are enabled then these flags are available as interrupts.

The monitoring of status flags as described above is necessary because the DMA transfers can be to relatively slow Peripheral Modules. If an access was attempted to the FIFO Channel Buffer and space was not available then the access would be held off until Peripheral Module transfers the data, which could be at an audio frequency data rate. Monitoring of the FIFO status is only necessary when the buffers are directly accessed; when the DMAC is configured to get the data, the FIFO status flags are monitored directly by the DMAC state machines.

2.7.1 DMA Data Transfer

Data can be transferred across buses: —

Register Bus ⇔ Register Bus
PCI/MPX Bus ⇔ Register Bus
Register Bus ⇔ Pixel Bus

Graphics Memory can be accessed via the Pixel Bus. In addition to transferring data between the areas shown above, it is also possible to transfer data between Peripheral Modules. In this mode the DMAC will provide temporary storage for the transfer.

DMA capable Peripheral Modules will provide a RBDMAREQ signal to the DMAC. DMAC will respond by writing or reading data to the Peripheral Module, a common RBDMACYCLE signal is provided to each Peripheral Module and can be used to identify a DMA cycle if required.

The DMAC is set up for a burst data transfer; the DMAC will monitor the RBDMAREQ signals and transfer small bursts of data to or from the FIFO Channel Buffer until the transfer is complete.

A FIFO Channel Buffer is allocated to each of the Peripheral Modules that are configured for DMA.

Refer to Appendices for data transfer set up details.

2.7.2 Programming DMAC

There is some important considerations as to the order of programming the registers. DMA n Control Register must always be programmed last. DMA n RAM Buffer Size Register must be programmed for each channel before DMA n Control Register is programmed for any of the channels. DMA FIFO Flush Register must be done before the values programmed into the DMA n RAM Buffer Size Register are accepted by the DMAC. The sequence of programming the registers is shown below.

- Initialise and configure DMAC

1. Do DMAC Initialisation Procedure explained in "2.12 HD64404 DMA Driver Design Note"

- Start DMA transfer

1. DMA n Start Address Register Master DMA mode and Inter-module DMA mode only
2. DMA n Length Register if length or burst size are valid for the DMA mode
3. Do DMA Pre-processing explained in "2.12 HD64404 DMA Driver Design Note"
4. DMA n Control Register Start DMAC
   See Table 2.12, 2.13 and "2.12 HD64404 DMA Driver Design Note" for details
5. Start DMA in Peripheral Module see Peripheral Module Manual
6. DMA External Select Register In order to set EDMA to 1. External DMA mode only

- When DMA completed (getting interrupt or DMA abort operation),

1. DMA External Select Register In order to clear EDMA to 0. External DMA mode only
2. Stop DMA in Peripheral Module and clear PM interrupt if applicable. See Peripheral Module Manual
3. Do DMA Post-processing explained in "2.12 HD64404 DMA Driver Design Note"
4. Clear DMAC interrupt if applicable

DMAC is very sensitive to register access sequence. Above sequence should always be followed. See "2.12 HD64404 DMA driver design note" for each scenario details.

2.7.3 DMA Channels

The DMAC has 16 channels, which can each be configured for data transfer from External Memory to a Peripheral Module or from a Peripheral Module to External Memory and Peripheral Module to Peripheral Module. Data transmitted by a Peripheral Module, for example, a serial audio link would be considered to be DMA write. Each channel will be programmed with the start

address in External Memory and length of transfer. The address of the Peripheral Module on the Register Bus will be programmed in the DMAC. DMA terminal count can be monitored by interrupt or by status flag.

Inter-module DMA mode transfers are done using RBDMAREQ as flow control, in this mode the FIFO Channel Buffer will be used as a temporary buffer.

2.7.4 Priority Encoders

Access to each of the three interfaces, which are PCI/MPX, Pixel Bus and Register Bus, are organised by three priority encoders, to ensure that the active channels have access in turn to each of the interfaces. The pixel bus priority encoder in the DMAC will decide which of the channels has the next access to the Pixel Bus, a request will then be made to the pixel bus DMA controller for access to the Graphics Memory.

The PCI/MPX priority encoder will select which channel has next access to the PCI/MPX interface. The Register Bus priority controller is internal to the DMAC and will select the next Peripheral Module DMA request to be serviced and will control access to the Register Bus from either the DMAC or the PCI/MPX bus.

2.7.5 RAM FIFO Buffer

The RAM FIFO Buffer is used as temporary storage for DMA transfers. The buffer is a dual port RAM (DPRAM) that is divided up into 16 separate FIFO Channel Buffers one for each DMA channel. One port of the buffer is accessed from the Pixel Bus interface and the other port is shared between the Register Bus and PCI/MPX bus. The accesses from the PCI/MPX to the FIFO Channel Buffers are interleaved between Register Bus DMA accesses so as not to impact on the Register Bus bandwidth for DMA accesses. The buffer sizes for each channel can be configured by software by programming registers in the DMAC. The buffer size is a maximum of 256 bytes (64 longwords) and a minimum of 64 bytes (16 longwords).

2.7.6 Direct Access to FIFO Channel Buffers

In slave mode the DMAC can be directly accessed from the external MPX/PCI Bus. This access can be a burst, if the respective external bus supports this mode. When the system is configured for the MPX bus, the DMAC cannot be an external bus master, so the only method to transfer data via the DMAC is by direct access to the FIFO Channel Buffers. Software flow control will be required in this mode where status flags are monitored either by interrupt or polling. The behaviour of the direct access to FIFO Channel Buffer is slightly different between MPX PIO mode and PCI target mode.

In PCI target mode, if the condition that the FIFO Channel Buffer is full during in target write or empty during target read will force PCI bus into the condition to stop (disconnect) until FIFO

Channel Buffer is ready to be accessed again. In MPX PIO mode FIFO Channel Buffer full or empty condition does not block MPX PIO access and so in this particular case the pointers in FIFO Channel Buffer will go back to their initial values and the data will be corrupted.

Therefore FIFO Burst Status or FIFO Status should be checked before every PIO access to FIFO Channel Buffer. If read on empty FIFO buffer, System Processor will be held off in case of PCI Bus and get wrong data in case of MPX Bus. If write on full FIFO buffer, System Processor will be held off in case of PCI Bus and write nothing in case of MPX Bus.

The FIFO Channel Buffers are in the PIO address space as defined in Table 2.4. It is only possible to write to a FIFO Channel Buffer if the channel is configured for a write to Peripheral Module and it is only possible to read from a FIFO Channel Buffer if the channel is configured for a read from Peripheral Module. It is not possible read and write to or from the same channel.

In MPX mode, burst bus transfer size is either 2 or 8 longwords. So the minimum FIFO Channel Buffer is set as 16 longwords in order to hold enough space for burst bus transfer.

2.7.7 Pixel Bus Interface

The DMAC is allocated one channel on the Pixel Bus and only access the Graphics Memory on this bus

2.7.8 Register Bus

The DMAC is the master of the Register Bus and allow the System Processor access to the Register Bus via the DMAC.

2.7.9 External DMA

One DMA channel can be allocated to External DMA via the MPX Bus. The signals associated with this interface are mpx_dreq, mpx_dack and mpx_drak signals. The register "External DMA Select" controls the selection of the specific DMA channel.

The system supports only one external DMA channel, which can be allocated to any of one the 16 DMA Channels. If the EDMA flag is set to 1 then the channel addressed by this register will be configured as for a normal DMA transfer but the FIFO Channel Buffer status flags are routed to the external DMA controller as a DMA request. In this way flow control to FIFO Channel Buffers can be controlled by System Processor.

The burst flag controls the mpx_dreq signal, which is software programmable in the DMA n Length Register. A request will be made when there is a burst size of data in the FIFO Channel Buffer if the channel is configured for a read from Peripheral Module, or a burst size of space in the FIFO Channel Buffer if the channel is configured for a write to Peripheral Module. The length of transfer should therefore be an integral number of bursts for correct operation. Also burst size in

DMA n Length Register should be equal to burst data transfer size specified in System Processor, esp. in case of SH7751, CHCR register, i.e., 1, 2 or 8 longwords. If this is not the case then the last burst of data to be transferred could be less than the bust size and data corruption could occur during the last burst. If it is not possible to make the transfer in an integral number of bursts, then the transfer should be split up so that this condition is satisfied.

When using the SH7750/1 DMAC, make the following DMAC settings in SH7750/1. For DMA transfer in dual address mode

1. DACK output in write cycle
2. Active-high DACK output
3. DMA destination start address, which is the initial address for Graphic memory, is 32-Byte address boundary.
4. Source address incremented
5. External request, dual address mode
6. DREQ falling-edge detection

2.7.10 Endian Conversion for PCI and MPX PIO Accesses

The Register Bus is always big Endian, the PCI Bus is little Endian and the MPX Bus can be either little Endian or big Endian. It is necessary to do Endian conversion during PIO accesses in both MPX and PCI modes, this is only necessary during byte and word accesses.

For PCI Bus, the DMAC does Endian conversion from little Endian to big Endian during PIO writes and big Endian to little Endian conversion during PIO reads. For MPX Bus, the DMAC is programmed which Endian mode the System Processor is operating in by programming a flag in the DMA External Select Register. The appropriate Endian conversion is then performed during PIO MPX word and byte accesses.

In DMA data transfers Endian conversion is performed on the data based on the source and destination Endian, which is programmed into the DMA n Control Register.

2.7.11 PIO Bus Activity Monitor

The PIO Bus Activity Monitor, if enabled, restricts the PIO accesses from the system processor to the register bus to allow sufficient bandwidth on the register bus for real time DMA transfers.

The activity of the register bus is monitored using an up-down counter to monitor the number of clock cycles that the system processor is occupying the register bus for PIO accesses. A count value is programmed into threshold detector, which compares the value in the counter with the programmed threshold value. The count will not be allowed to exceed this threshold by holding off the PIO access until the count has counted down to below the threshold. The counter counts up

when the register bus is executing PIO accesses and counts down when it is idle or executing DMA transfers.

The rate at which the counter counts up is programmable, but the rate that it counts down is always one count per register bus clock cycle. The up count is programmable in steps of 1 to 15 and will increment by this value for every clock cycle that the PIO is active. If the PIO Bus Activity Monitor is not enabled the counter value can exceed the threshold set in TCNT and increase up to its maximum value of 4095, the counter will not wrap around to zero. If the average PIO activity is less than is programmed into TCNT and PUP then the count will never exceed the threshold and will count down to zero.

2.8 Programming the PIO Monitor

2.8.1 Monitoring Mode

This is useful for measuring the PIO bus activity

1. Program TCNT
2. Program PUP

Periodically (using a timer) monitor MCNT, PUP should be adjusted until a stable count is reached. The bus activity can then be calculated from the value programmed into PUP.

2.8.2 PIO Monitor Active Mode

PIO monitor is used when the system needs to secure the worst latency time of each DMA channel response in order for some peripherals to transfer data on a real time basis.

Worst tolerable latency: The following modules will require the real time data transfer. They are all audio related modules that require certain amount of data every audio frame.

1. MOST packet, MOST stream in MIM

   MIM can use six individual DMA channels for packet and stream transfer. MIM packet transfer can be allocated to maximum 36 byte per up to 1/48 kHz. MIM stream transfer can be allocated to maximum 32 byte per up to 1/48 kHz.
2. SSI

   SSI supports multiple channel data transfer which is up to 256 bit, ie 32 byte per up to 1/48 kHz.
3. SPDIF

   SPDIF supports stereo samples per up to 1/96 kHz.
4. Audio Codec i/f

   Audio Codec i/f supports stereo samples per 1/48 kHz.

MIM and SSI multi-channel mode will require more data transfer per an audio frame than SPDIF and Audio Codec i/f. Therefore the worst tolerable latency for HD64404 real time modules depends on the data size of MOST packet/stream or SSI per an audio frame. In other words, SSI single channel mode, SPDIF and Audio Codec i/f do not require PIO monitor function to guarantee the worst tolerable latency.

Worst tolerable latency = A * B/C

A: a period of time of an audio frame.
B: available data size on the register bus every DMA transaction
C: data size to be transferred per an audio frame

For example, using 36Byte MIM packet,

the worst tolerable latency is 1/48kHz * 4B/36B = 2.315 us

Worst latency time for each DMA: The worst latency time for DMA is a function of the following factors.

1. the number of DMA channels that possibly run at the same time

   DMAC uses the round-robin scheme for DMA channel arbitration so the more DMA channels are used, the longer the worst latency time becomes.
2. PUP in PIO_monitor register
3. TCNT in PIO_monitor register
4. Register bus operating frequency

The system using HD64404 register bus DMA must make sure that the worst latency time for DMA is less than the worst tolerable latency time for peripherals.

How to program PUP and TCNT: PUP and TCNT are chosen by the three parameters.

1. MIM or SSI data transfer size per an audio frame and per a DMA channel
2. The Number of DMA channels to be used simultaneously (maximum 16 channels)
3. Register bus clock frequency

   According to those conditions, the following tables show PUP and TCNT combinations.

PCI mode

Data transfer size per 1 audio frame and 1 DMA channel	# of DMA ch	Register bus clock freq 33MHz	
		PUP	TCNT
36 Byte	16ch	3	16
	14ch	3	32
	12ch	2	16
	10ch	2	32
	8ch	2	32
	4ch	Off(*)	Off
32 Byte	16ch	3	32
	14ch	2	16
	12ch	2	32
	10ch	2	32
	8ch	2	32
	4ch	Off	64
16 Byte	16 ch	2	128
	14 ch	Off	Off
8 Byte	16 ch	Off	Off

Note: (*) Off means EMP bit in PIO_monitor is 0.

MPX mode

Data transfer size per 1 audio frame and 1 DMA channel	# of DMA ch	Register bus clock freq							
		50MHz		44MHz		39MHz		33MHz	
		PUP	TCNT	PUP	TCNT	PUP	TCNT	PUP	TCNT
36 Byte	16 ch	2	96	2	32	3	32	4	64
	14 ch	Off	Off	2	96	2	32	3	32
	12 ch	Off	Off	Off	Off	2	96	2	32
	10 ch	Off	Off	Off	Off	Off	Off	2	64
	8 ch	Off	Off	Off	Off	Off	Off	Off	Off
32 Byte	16 ch	Off	Off	2	96	2	32	3	32
	14 ch	Off	Off	Off	Off	2	64	2	32
	12 ch	Off	Off	Off	Off	Off	Off	2	64
	10 ch	Off	Off	Off	Off	Off	Off	Off	Off
24 Byte	16 ch	Off	Off	Off	Off	Off	Off	2	96
	14 ch	Off	Off	Off	Off	Off	Off	Off	Off
16 Byte	16 ch	Off	Off	Off	Off	Off	Off	Off	Off

2.9 Appendix 1 HD64404 Data Path

Table 2.9 shows data path in HD64404, i.e., Data transfer path and applicable buses are summarized.

Table 2.9 HD64404 Data Path and DMA Modes

Data Path Number	DMA Modes	Data path Supported	DMA Master	Applicable Buses			
				RB	PB	MPX	PCI
1	Master DMA	SM⇔PM	DMAC	√			√
2		GM⇔PM		√	√	Not used	
3	Slave DMA	SM⇔PM	DMAC*1	√		√	√
4	Inter-module DMA	PM⇔PM	DMAC	√		Not used	
5	External DMA	SM⇔PM	SP DMAC	√		√	
6	SP DMA*3	SM⇒GM	SP DMAC		√	√	
7	PCI Master DMA*4	SM⇔GM	PCI Module DMAC		√		√
8	SP PIO1*2	SP⇔PM	None	√		√	√
9	SP PIO2*5	SP⇔GM	None		√	√	
10	SP PIO3*6	SP⇔GM	None		√		√

SM:	System Memory	RB:	Register Bus
GM:	Graphics Memory	PB:	Pixel Bus
PM:	Peripheral Module	MPX:	MPX Bus
SP:	System Processor	PCI:	PCI Bus
DMAC:	This DMAC Module	√:	Applicable and used

Not used: Applicable but not used

*1 In slave DMA mode, DMAC supports DMA only between PM and RAM FIFO Buffer. SP will do PIO access between RAM FIFO Buffer and External Memory location.

*2 SP PIO1 access is supported in DMAC Module. It is not DMA mode, but included here for showing complete data access paths.

*3 System Processor DMA is supported in MPX Bus I/F Module. External DMA and System Processor DMA shares the same System Processor DMAC Channel, so either one of DMA transfer should be active at any moment. Serialization of request should be controlled in DMA software library.

*4 PCI Master DMA is supported in PCI Bus I/F Module. Two DMA channels are supported. Those DMA channel can be used in parallel to DMAC Module DMA channels.

Also please refer to "2.12 HD64404 DMA Driver Design Note", section 2.12.6, "Consideration on External DMA mode and DMA modes supported by CPU I/F modules" for more notes on System Processor DMA and PCI Master DMA.

*5 SP PIO2 access is supported in MPX Bus I/F Module. It is not DMA mode, but included here for showing complete data access paths.

*6 SP PIO3 access is supported in PCI Bus I/F Module. It is not DMA mode, but included here for showing complete data access paths.

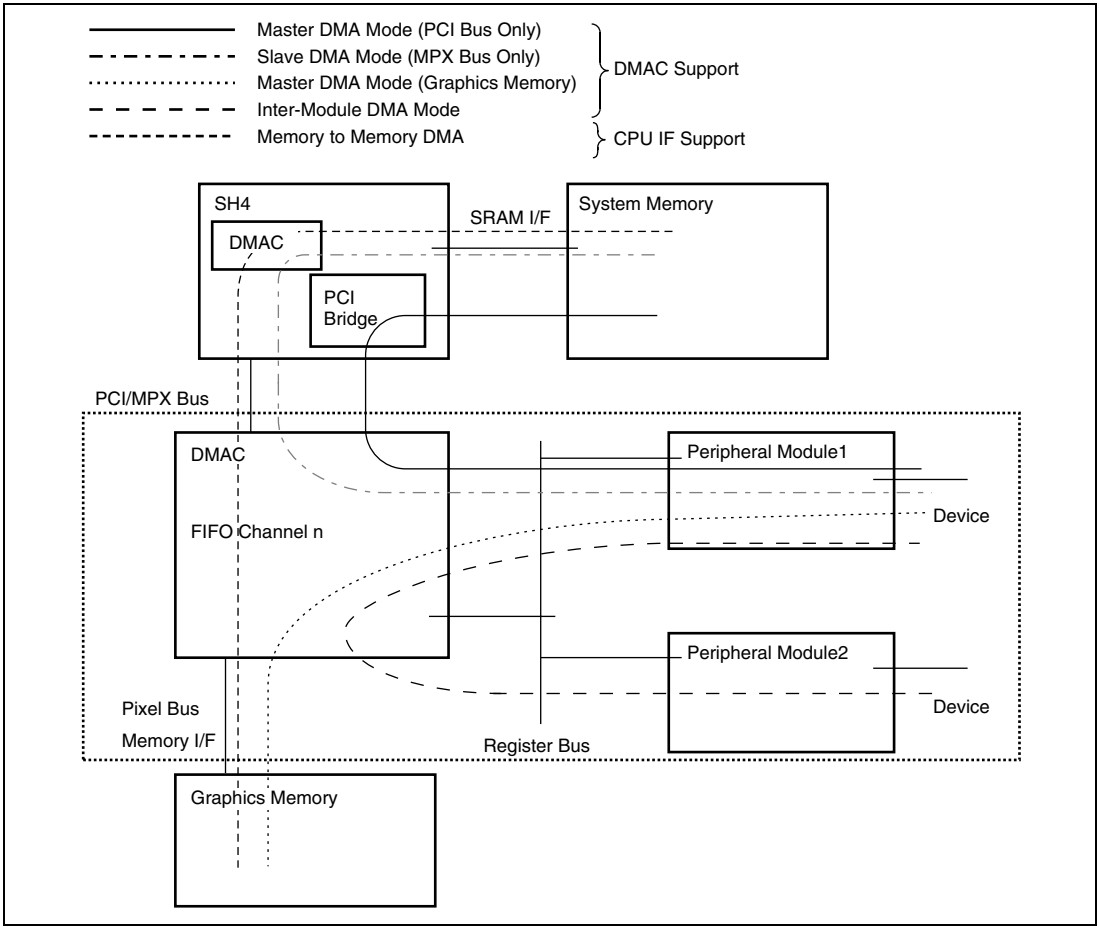


Figure 2.2 HD64404 DMA Data paths

2.10 Appendix 2 DMA Modes in DMAC Module

Table 2.10 summarizes how DMA mode is specified by combination of DMA n Control and DMA External Select Registers.

Table 2.10 How DMA Modes are Specified

	DMA_External_ Select	DMA_n_Control			Applicable External Memory		Applicable Bus	
DMA Modes	EDMA	RBEN	MM	ML	System Memory	Graphics Memory	MPX Bus	PCI Bus
Master DMA	0	0	1	1	√			√
	0	0	1	0		√	Not used	
Slave DMA	0	0	0	—	√		√	√
Inter-module DMA	0	1	0	—			Not used	
External DMA*	1	0	0	—	√		√	

—: Don't care

√ : Applicable and used

Not used: Applicable but not used

Note: * Only one DMA Channel can be specified as External DMA channel.

2.11 Appendix 3 DMA Mode Parameters

In Table 2.12 and 2.13, all the parameters for specifying each DMA mode are summarized. In Table 2.11 below, legends for looking into Table 2.12 and 2.13 are summarized.

Table 2.11 Legends for DMA Mode Parameter Tables

Abbreviation	Stands for	Defined in
Big	Big Endian	
Burst Size	Burst size	DMA n Length Register
CI	Channel Interrupt flag	DMA Interrut Source Register
CM	Channel Interrupt Mask flag	DMA Interrut Source Register
Cont.	Continuous Data Transfer mode	
CSEL	Channel Select	DMA n Control Register
CWD	Channel Width Destination	DMA n Control Register
CWS	Channel Width Source	DMA n Control Register
DBEN	Double Buffer Enable flag	DMA n Control Register
DDEN	Dummy DMA Enable flag	DMA External Select Register
Dest.	Destination of Data Transfer	
DR	Direction flag	DMA n Control Register
DTRA	Start Master DMA Transfer flag	DMA n Control Register
EDMA	Start External DMA flag	DMA External Select Register
ENDD	Endian flag for Destination Data	DMA n Control Register
ENDS	Endian flag for Source Data	DMA n Control Register
FB	FIFO Burst flag	DMA FIFO Status Register
FBEN	FIFO Burst Interrupt Enable flag	DMA n Control Register
FF	FIFO Flush Operation flag	DMA FIFO Flush Register
FIFO	FIFO Channel Buffer	
FIFO Flush	FIFO Flush Operation	DMA FIFO Flush Register
Fixed.	Fixed Length Data Transfer mode	
FS	FIFO Status flag	DMA FIFO Status Register
FSEN	FIFO Status Interrupt Enable flag	DMA n Control Register
GM	Graphics Memory	
Length	DMA Length field	DMA n Length Register
Little	Little Endian	
MCOUNT	MCOUNT field	DMA n MCOUNT Register
MEND	MPX Bus Endian flag	DMA External Select Register
ML	External Memory Location flag	DMA n Control Register
MM	Master Mode flag	DMA n Control Register

Abbreviation	Stands for	Defined in
MPX	MPX Bus	
PCI	PCI Bus	
PCOUNT	PCOUNT field	DMA n PCOUNT Register
PM	Peripheral Module	
PM Intrpt.	Peripheral Module Interrupt	
PT	Peripheral Terminal Count Status flag	DMA Status Register
PTEN	Peripheral Terminal Count Interrupt Enable flag	DMA n Control Register
RBEN	Register Bus Enable flag	DMA n Control Register
Req_Addr	Start Address	DMA n Start Address Register
Req_number	DMA Request Number	DMA n Start Address and DMA q Request Address Register
Req_Status	DMA Request Peripheral Status flag	DMA Peripheral Request Status Register
RTRA	Start Register Bus Transfer flag	DMA n Control Register
SM	System Memory	
SP DMA Intrpt.	System Processor DMA Interrupt	
Src.	Source of Data Transfer	
TC	Terminal Count Status flag	DMA Status Register
TCEN	Terminal Count Interrupt Enable flag	DMA n Control Register
Timed Out	Timed out of Software Timer	
1	Write 1 or read 1	
0	Write 0 or read 0	
0/1	Either 0 or 1 read/write depends on how device driver is designed	
0/1 1/0	Either one of those flags should be 1	
Bold	Indicates DMAC module register field/flag	
√	Valid and read/write value to be specified in register description	
—	Invalid access	
—(X)	Invalid and should be X	
← →	DMA direction by DMAC module	
← →	PIO direction by System Processor	
⇦ ⇨	DMA direction by System Processor DMAC	

Scenario Number	DMA Mode				CPU I/F	Primary DMA Address	Data Transfer Entity and Direction: DMA		FIFO Channel Buffer	DMA/ PIO	Secondary DMA Address			Direc-tion	Data Transfer Option: Conti-nuous/ fixed length transfer	Pack/Unpack Endian Conversion		MPX Endian	FB Control	Transfer Length
	Definition	MM	RBEN	EDMA	PCI/ MPX	CSEL& Req_Addr	PM	← →	FIFO	← →	SM/ GM/ PM	ML	Start_ Address	DR	DBEN	CWS/ ENDD*	CWD/ ENDS*	MEND	Buret Size	Length
1	Master DMA	1	0	0	PCI	√	PM	→	FIFO	→	SM	1	External Memory Address	0	0(Fixed)	8/16/32/ Lttle	—(00)/ —(0)	—	—	√
2															1(Cont.)					
3					PCI/ MPX		PM		FIFO		GM	0			0(Fixed)	8/16/32/ Big/Lttle				
4															1(Cont.)					
5					PCI		PM	←	FIFO	←	SM	1		1	0(Fixed)	—(00)/ —(0)	8/16/32/ Big/Lttle			
6															1(Cont.)					
7					PCI/ MPX		PM		FIFO		GM	0			0(Fixed)		8/16/32/ Big/Lttle			
8															1(Cont.)					
9	Slave DMA	0	0	0	PCI/ MPX	√	PM	→	FIFO	→	SM	—(0)	—(0)	0	0(Fixed)	8/16/32/ Big/Lttle	—(00)/ —(0)	—	√	√
10															1(Cont.)					—
11							PM	←	FIFO	←	SM			1	0(Fixed)	—(00)/ —(0)	8/16/32/ Big/Lttle			√
12															1(Cont.)					—
13	Inter-module DMA	0	1	0	PCI/ MPX	√	PM	→	FIFO	→	PM	—(0)	Req_ number	0	0(Fixed)	—(00)/ —(0)	—(00)/ —(0)	—	—	√
14															1(Cont.)					—
15							PM	←	FIFO	←	PM			1	0(Fixed)					√
16															1(Cont.)					—
17	External DMA	0	0	1	MPX	√	PM	→	FIFO	⇨	SM	—(0)	—(0)	0	—(0)	8/16/32/ Big/Lttle	—(00)/ —(0)	—	√	√
18							PM	←	FIFO	⇦	SM			1		—(00) —(0)	8/16/32/ Big/Lttle			

Note: * ENDS/ENDD is always Little Endian if CPU I/F is PCI Bus.

Scenario Number	DMA Mode	DMA Start Trigger		Flow Control				DMA Stop Trigger Exclusively used between DMA abort and Intertupt Enable							Int Handler Aid	DMA Completion Status Check					Error Handling & Recovery Procedure	
		CPU write		Interrupt				CPU must abort DMA		Enable transfer completion Interrupt				CPU Timer	Intrpt Src & Mask	DMAC Status					FIFO Flush	Dummy DMA Cycle
	Definition	RTRA	DTRA	TCEN	PTEN	FSEN	FBEN	RTRA	DTRA	TCEN	PTEN	PM Intrpt	SP DMA Intrpt	Timed Out to detect underrun	CI/CM	PT PCOUNT	TC MCOUNT	FS	FB	Req_ Status	FF	DDEN
1	Master DMA	1	1	1	—	—(0)	—(0)	—	—	1	—(0)	0	—(0)	1	√	—	√	—	—	Must read	√	√
2								0	0			1										
3								—	—			0										
4								0	0			1										
5				—				—	—	—(0)	0/1	1/0				√	—					
6				1				0	0	1	—(0)	1				—	√					
7				—				—	—	—(0)	0/1	1/0				√	—					
8				1				0	0	1	—(0)	1				—	√					
9	Slave DMA	1	0	—	—	0	1	—	—	—(0)	0	0	—(0)	1	√	√	—		√	Must read	√	√
10								0	0		—(0)	1				—		—				
11								—	—		0/1	1/0				√						
12								0	0		—(0)	1				—		—				
13	Inter-module DMA	1	0	—	—	—(0)	—(0)	—	—	—(0)	0/1	1/0	—(0)	1	√	√	—	—	—	Must read	√	√
14								0	0		—(0)	0/1				—						
15								—	—		0/1	1/0				√						
16								0	0		—(0)	0/1				—						
17	External DMA	1	0	—	—	—(0)	—(0)	—	—	—(0)	0	0	1	1	√	√	—	—	—	Must read	√	√
18											0/1	1/0	0									

2.12 HD64404 DMA Driver Design Note

2.12.1 General Description

This Design Note is written for the device driver designer who will write HD64404 Peripheral Module device driver using DMA capability of DMAC Module, PCI I/F Module DMAC and also System Processor DMAC.

Reader should be familiar with the specification of those modules.

For the DMA block diagram, refer to figure 2.3.

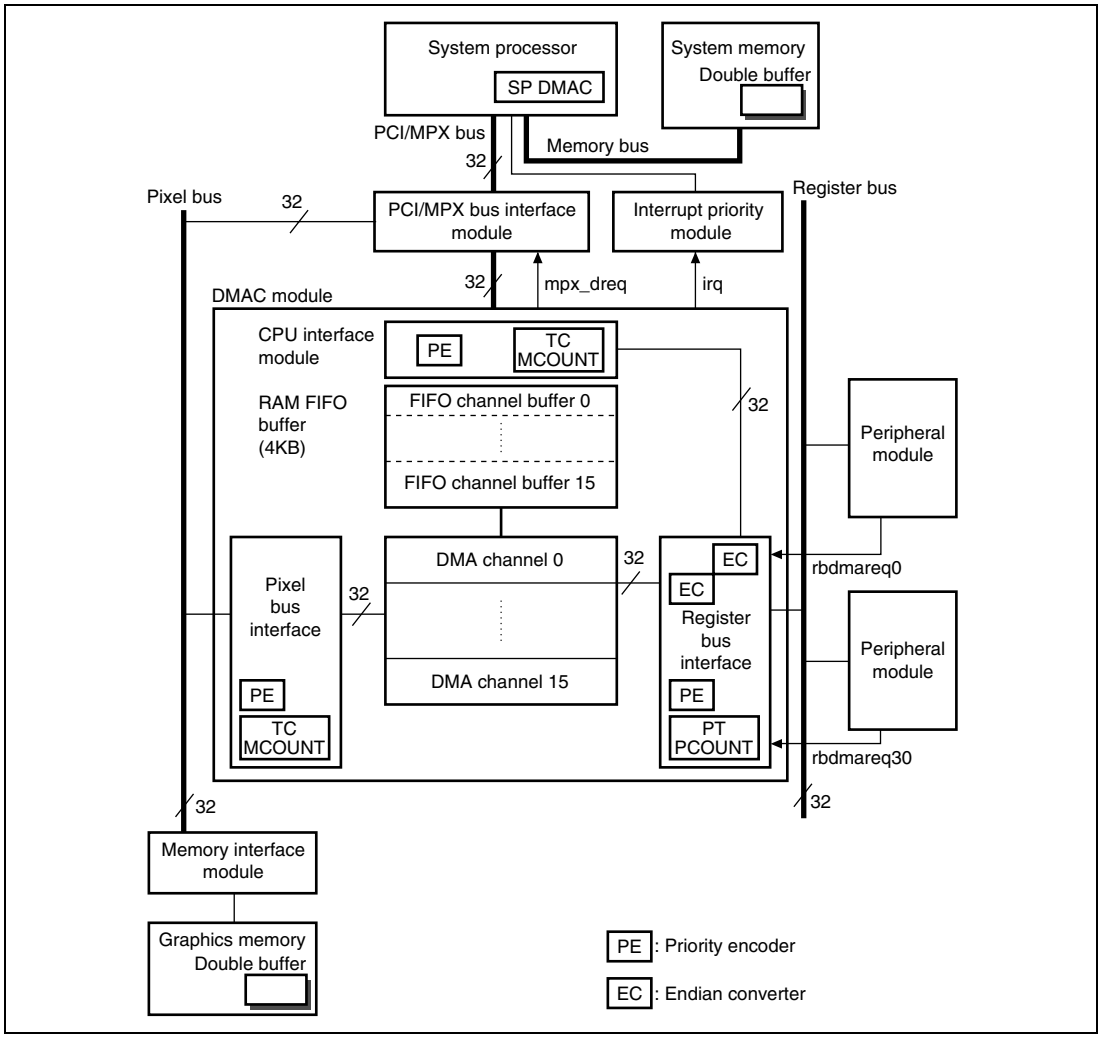


Figure 2.3 System Diagram

2.12.2 References

- Section 4 PCI I/F
- SH7751 Series Hardware Manual

In this note, some useful design considerations for HD64404 device driver design related to DMAC are explained. The following topics are treated in this order.

1. Bus Configuration and Endian Support
2. DMA Channel Allocation
3. DMA Channel Parameter Design
4. Consideration on External DMA mode and DMA modes supported by CPU I/F modules
5. Access control of DMAC registers
6. Data Transfer procedure for each DMA scenario
7. DMAC Initialization Procedure
8. DMA Pre-, Post- and Abort Processing
9. DMA Interrupt Handling

2.12.3 Bus Configuration and Endian Support

HD64404 supports PCI/MPX buses as CPU I/F and has both Register Bus and Pixel Bus inside. Some rules of thumb are summarized how to support Endian.

1. PCI is always Little Endian
2. Register Bus is always Big Endian
3. MPX Bus and Pixel Bus have the same Endian as System Processor has.
4. OS has its preferable Endian. For example, WindowsCE only supports Little Endian and VxWORKS supports both Big and Little Endian.
5. Little Endian for MPX/Pixel Bus is preferable in HD64404.

2.12.4 DMA Channel Allocation

HD64404 supports 16 DMA channels. Data transfer direction in a channel is one way, so that two channels are necessary for a Read/Write Peripheral device, i.e., Request Number in DMAC terminology. HD64404 can utilize DMAC in system processor as Peripheral device stream, i.e. External DMA mode if CPU I/F is MPX Bus.

So if the number of DMA channels is not enough to support all the device data streams, then DMA channel should be dynamically allocated among peripheral device streams.

First important design consideration is to decide how many DMA Channels should be statically allocated to some device streams and the rest of DMA Channels should be shared among device streams. For statically allocated Channels, specific Request Numbers are assigned.

DMA Channel Configuration like Table 2.14 should be drawn first. Principles used in this example are followings,

- Some channels are allocated statically to most heavy traffic modules, Modules A through F in this example.
- If continuous data transfer is required, channels are allocated statically, based on the assumption that data transfer length is larger that 64KB or might vary at each transfer.
- One channel is allocated statically to External DMA. But the channel is shared between modules.
- Inter-Module DMA channels are allocated statically to specific modules.
- Rest of the channels are allocated dynamically to modules. In this example, there are 7 dynamic channels allocated.

Hint: Channel Management Table (CMT) should be prepared in software DMA channel library. CMT should manage whether channel is free or not and provide a wait queue to manage DMA requests.

Content of DMA n Control Register should be saved in CMT. When DMA n Control Register is read, the value DTRA and RTRA might change from the value written, as described in the register description of DMAC Block specification. So if checking the DMA n Control register written contents necessary, use the saved value instead of read value.

Checking DTRA and RTRA flag to know the data transfer completion timing is not recommended in this note because this does not reflect the precise data completion timing between Peripheral module and External Memory in many scenarios. DTRA only works in Master DMA mode and RTRA only guarantees the data was delivered to the Peripheral modules, i.e. not guaranteed that Peripheral Module correctly delivered data to the outside device.

Table 2.14 DMA Channel Configuration Table Examples

Channel Number n	Direction	Peripheral	Static/ Dynamic	FIFO Size Bytes	DMA Length Bytes	Fixed/ Continuous	DMA Mode
0	PM → SM	Module A	Static	256	4096	Fixed	Master
1	SM ← PM	Module A	Static	256	4096	Fixed	Master
2	PM → SM	Module B	Static	256	32768	Continuous	Master
3	SM ← PM	Module B	Static	256	32768	Continuous	Master
4	PM → SM	Module C	Static	256	512	Fixed	Master
5	GM ← PM	Module D	Static	256	1024	Fixed	Master
6	PM → PM	Module E	Static	256	32768	Continuous	Inter-module
7	PM → PM	Module F	Static	256	Dynamic	Continuous	Inter-module
8	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
9	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
10	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
11	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
12	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
13	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
14	Dynamic	Shared	Dynamic	256	Dynamic	Fixed	Master/Slave
15	Dynamic	Shared	Static	256	Dynamic	Dynamic	External

2.12.5 DMA Channel Parameter Design

Once DMA Channel allocation is done, next step is to design DMA Channel option detail. For each Peripheral Modules, device characteristics and use case, required performance measure should be defined or given. Also what levels of software controllability over the device should be defined or given in the device driver API parameters. Especially, how to implement arbitral length of data transfer and also how to pack/unpack device data should be considered. Table 2.15 summarizes several options to do arbitral length DMA transfer. In Table 2.15, assumption was made that the page size is 4 Kbytes and also DMA buffers in System Memory is page based.

Table 2.15 Extending Data Length Options

DMA Data length	Implementation Options
4 bytes to 60 Kbytes	Use fixed length data transfer
60 Kbytes or more	Use continuous data transfer
	Use fixed length data transfer multiple times internally
	Use External DMA (MPX Bus only)
	Use System Processor DMA (MPX/PCI Bus) see next section

2.12.6 Consideration on External DMA mode and DMA modes supported by CPU I/F Modules

DMAC support External DMA mode and DMA supported by CPU I/F modules. In Table 2.16, those DMA modes are summarized.

See Table 2.9 in this chapter for legends.

Table 2.16 DMA modes using DMAC outside DMAC Module

DMA Mode	DMAC	Number of DMA Channels	Direction	CPU I/F	Supported by	Limitation
External DMA	SP DMAC	1	PM → SM SM → PM	MPX	DMAC Module	Either one of DMA channel can be active at all time
System Processor DMA	SP DMAC	1	SM → GM	MPX	MPX Bus Module	
PCI Master DMA	DMAC in PCI Module	2	SM → GM GM → SM	PCI	PCI Bus Module	Two DMA channels can be used in parallel to DMAC FIFO Channels

System Processor DMA supported by MPX Bus I/F Module and PCI Master DMA supported by PCI Bus I/F Module both provide the DMA between System Memory and Graphics Memory. In System Processor DMA, there are some restrictions explained below:

- Number of DMA channel is one and the channel is shared with External DMA mode supported by DMAC Module. So only one of System Processor DMA or External DMA can be active at all time. Easiest way to control serialization is using External DMA channel of the Channel Management Table (CMT) explained in "DMA Channel Allocation" section as a semaphore data structure.
- DMA direction is from System Memory to Graphics memory only. No directional restriction on PIO access from System Processor onto Graphics Memory though.

2.12.7 Access Control of DMAC Registers

Due to the nature of DMAC –Peripheral Module mutual dependency, designing DMAC driver as a fully independent driver is not so good idea. In order to reduce the unnecessary critical section overhead in device driver, DMAC register owner can be designed as shown below.

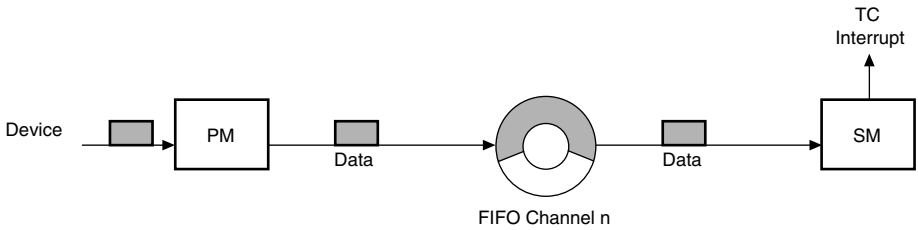
Table 2.17 DMAC Register Owner and Access Control

Registers	Owner	Mutual Exclusion Control
DMA n RAM Buffer Size	DMAC Driver	Start-up timing only
DMA External Select.MEND	DMAC Driver	Start-up timing only
DMA External Select.EDMS	DMAC Driver	Start-up timing only
DMA q Request Address	DMAC Driver	Start-up timing only
PIO Monitor	DMAC driver	No sharing
PIO Monitor Status	DMAC driver	No sharing
DMA n Length	Peripheral Module Driver	No Sharing
DMA n Start Address	Peripheral Module Driver	No Sharing
DMA n Control	Peripheral Module Driver	No Sharing
DMA External Select.EDMA	Peripheral Module Driver	No sharing
DMA n FIFO	Peripheral Module Driver	No Sharing
DMA FIFO Flush	Peripheral Module Driver	Write 1 to PM Channel Write 0 to other flags
DMA Peripehral Request Status	Peripheral Module Driver	Write 1 to PM Request Number Write 0 to other flags
DMA Interrupt Source	Interrupt Handler	Disable all HD64404 Interrupts
DMA FIFO Status	Interrupt Handler	Disable all HD64404 Interrupts
DMA External Select.DDEN	No use case right now	Disable all SP Interrupts to make sure there is no PIO access

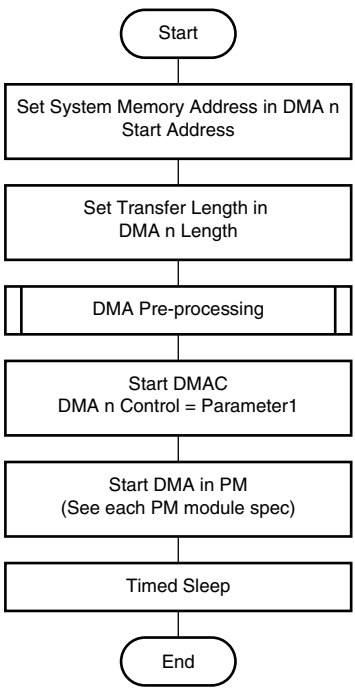
2.12.8 Data Transfer Procedure for Each DMA Scenario

For each data transfer scenario described in Table 2.12 and 2.13 of how data transfer should be controlled is explained in this section.

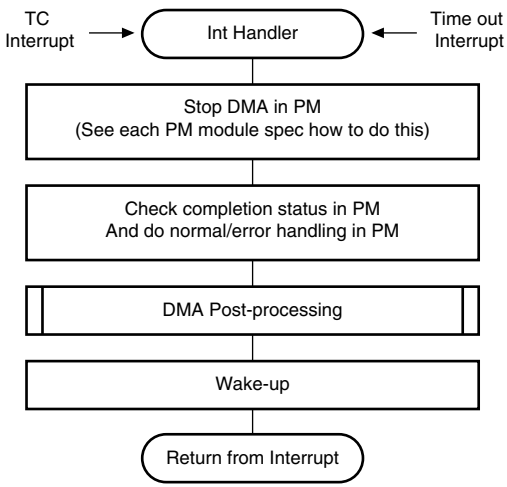
Scenario 1 PM → FIFO → SM Fixed length data transfer mode



Scenario 1 Data Transfer Chart



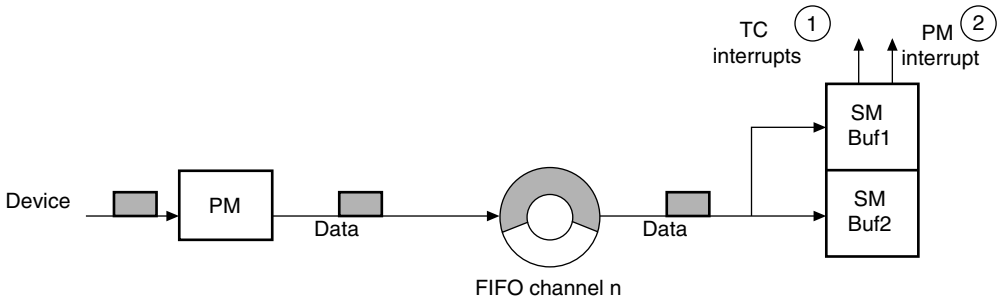
Flowchart 1



Flowchart 2

DMA n Control Register Parameter 1
1: MM,ML,RTRA,DTRA,TCEN,
0: RBEN,DR,DBEN,ENDD,CWD,ENDS,FSEN,FBEN,PTEN
Valid: CSEL,CWS

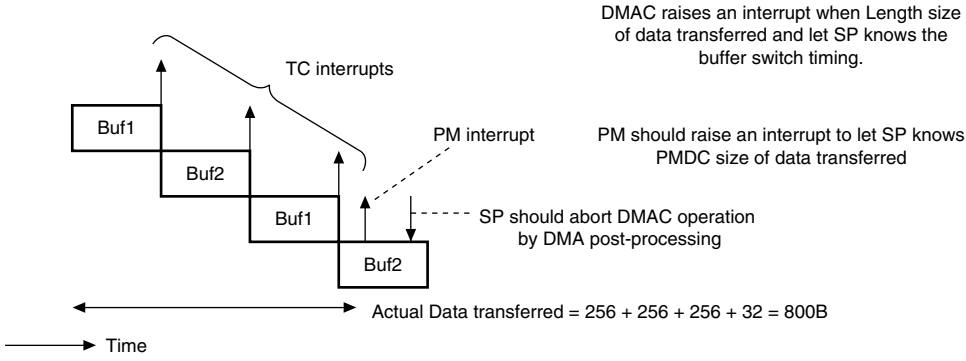
Scenario 2 PM → FIFO → SM Continuous data transfer mode



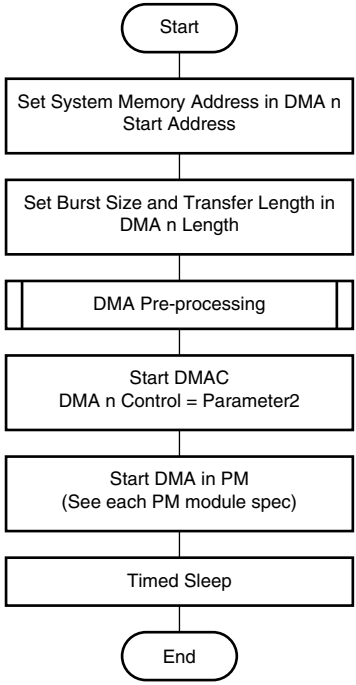
Scenario 2 Data Transfer Chart

This mode only works when PM has DMA counter (PMDC) and completion interrupt (PM Interrupt) because DMAC can not stop DMA at specific DMA count but just abort DMA. See below.

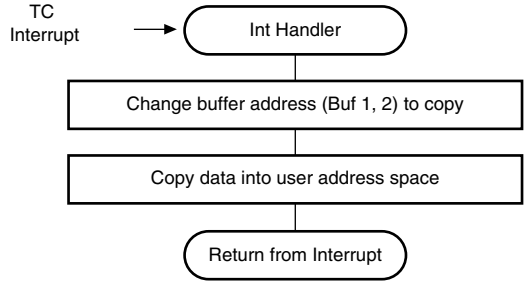
Example scenario: PMDC = 800B, DMA_n_Length.Length = 256B Buf1,Buf2:256B



Scenario 2 Data Timing Chart

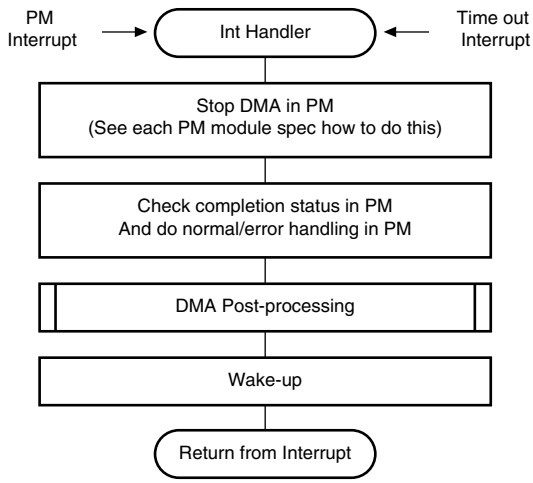


Flowchart 3



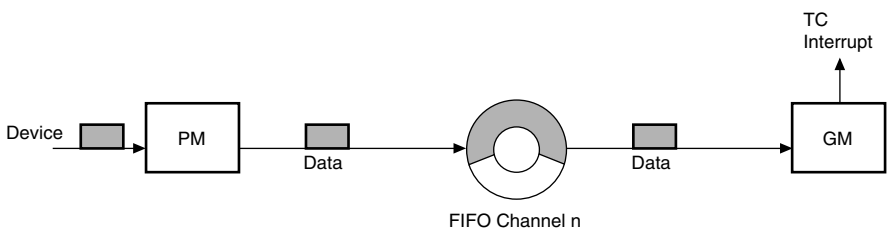
Flowchart 4

DMA n Control Register Parameter2
1: MM,ML,RTRA,DTRA,TCEN,DBEN
0: RBEN,DR,ENDD,CWD,ENDS,FSEN,FBEN,PTEN
Valid: CSEL,CWS



Flowchart 5

Scenario 3 PM → FIFO → GM Fixed length data transfer mode

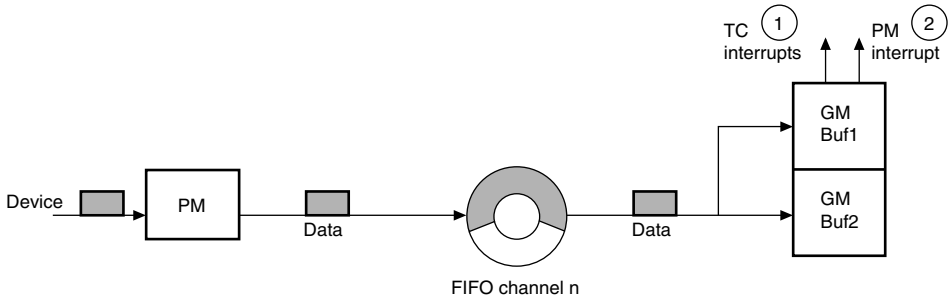


Scenario 3 Data Transfer Chart

Flowcharts 1 and 2 Replace Parameter1 of Parameter3

DMA n Control Register Parameter3
1: MM,RTRA,DTRA,TCEN
0: ML,RBEN,DR,DBEN,CWD,ENDS,FSEN,FBEN,PTEN
Valid: CSEL,CWS,ENDD

Scenario 4 PM → FIFO → GM Continuous data transfer mode

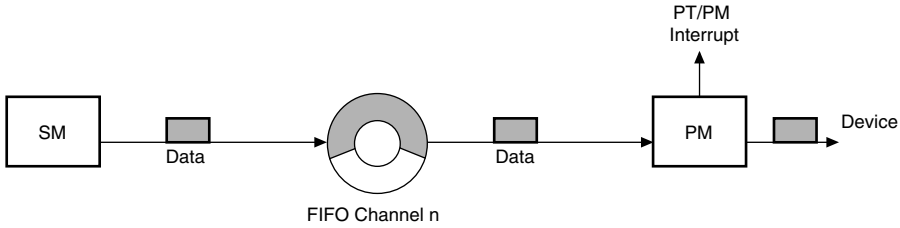


Scenario 4 Data Transfer Chart

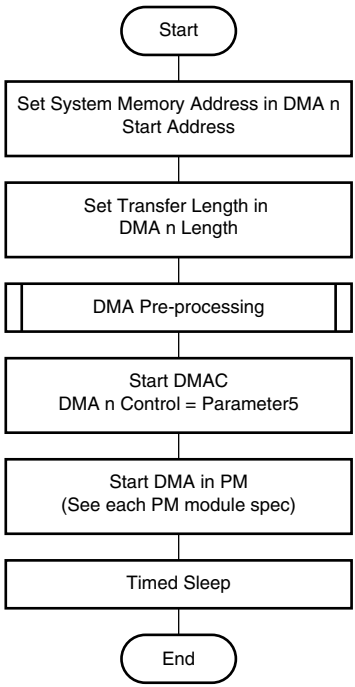
Flowcharts 3, 4 and 5 Replace Parameter2 of Parameter4

DMA n Control Register Parameter4
1: MM,RTRA,DTRA,TCEN,DBEN
0: ML,RBEN,DR,CWD,ENDS,FSEN,FBEN,PTEN
Valid: CSEL,CWS,ENDD

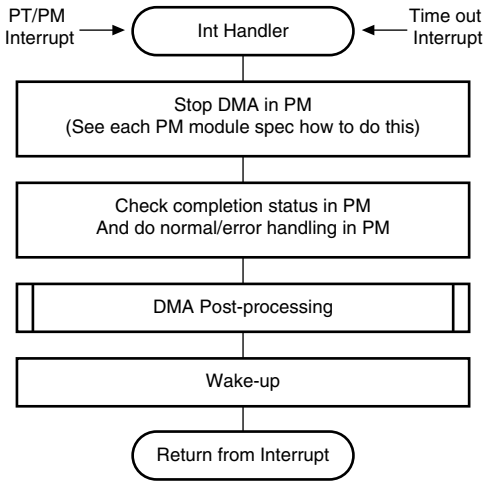
Scenario 5 SM → FIFO → PM Fixed length data transfer mode



Scenario 5 Data Transfer Chart



Flowchart 6

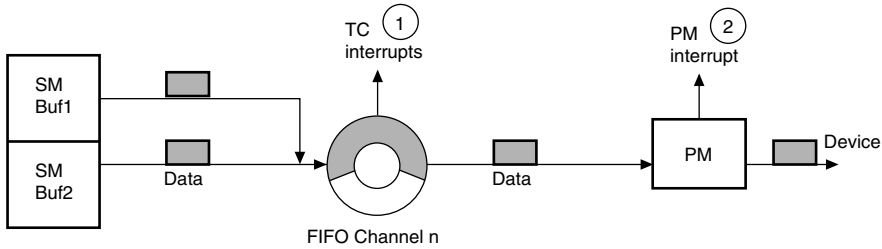


Flowchart 7

In this mode, either PT Interrupt or PM interrupt should be used.

DMA n Control Register Parameter5
1: MM,ML,DR,RTRA,DTRA
0: RBEN,DBEN,ENDD,CWS,ENDS,FSEN,FBEN,TCEN
Valid: CSEL,CWD,PTEN

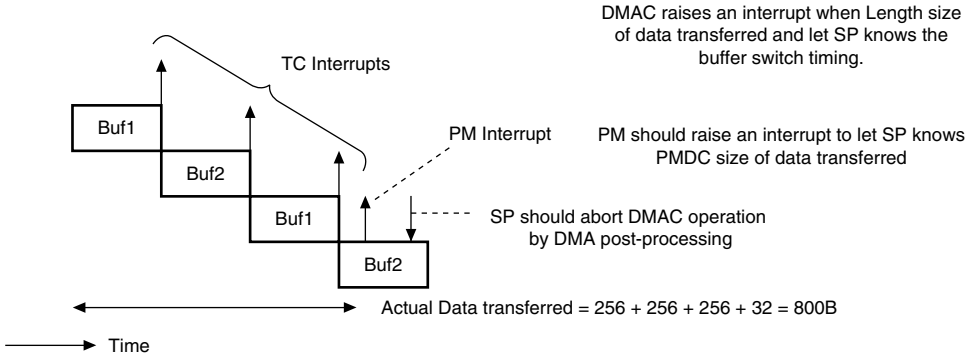
Scenario 6 SM → FIFO → PM Continuous data transfer mode



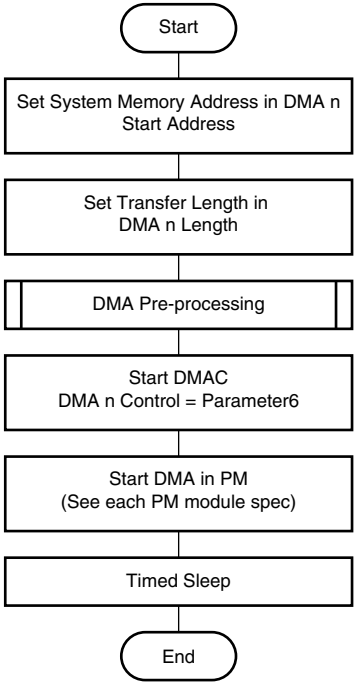
Scenario 6 Data Transfer Chart

This mode only works when PM has DMA counter (PMDC) and completion interrupt (PM Interrupt) because DMAC cannot stop DMA at specific DMA count but just abort DMA. See below.

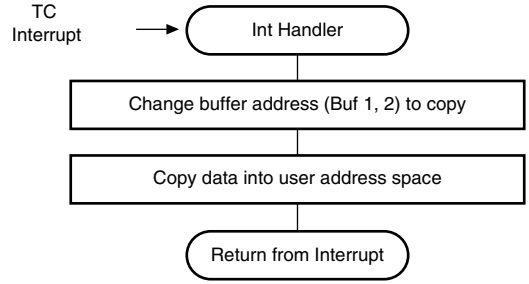
Example scenario: PMDC = 800B, DMA_n_Length.Length = 256B Buf1,Buf2:256B



Scenario 6 Data Timing Chart



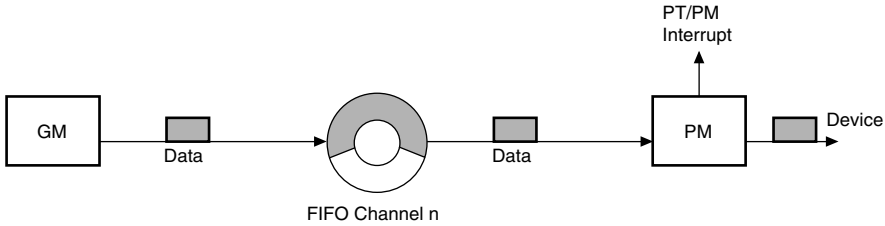
Flowchart 8



Flowchart 9

DMA n Control Register Parameter6
1: MM,ML,DR,RTRA,DTRA,DBEN,TCEN
0: RBEN,ENDD,CWS,ENDS,FSEN,FBEN,PTEN
Valid: CSEL,CWD

Scenario 7 GM → FIFO → PM Fixed length data transfer mode



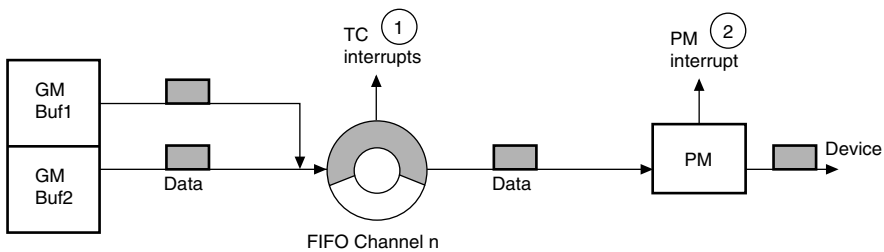
Scenario 7 Data Transfer Chart

Flowcharts 6 and 7 Replace Parameter5 of Parameter7

In this mode, either PT Interrupt or PM interrupt should be used.

DMA n Control Register Parameter7
1: MM,DR,RTRA,DTRA
0: ML,RBEN,DBEN,ENDD,CWS,FSEN,FBEN,TCEN
Valid: CSEL,CWD,ENDS,PTEN

Scenario 8 GM → FIFO → PM Continuous data transfer mode



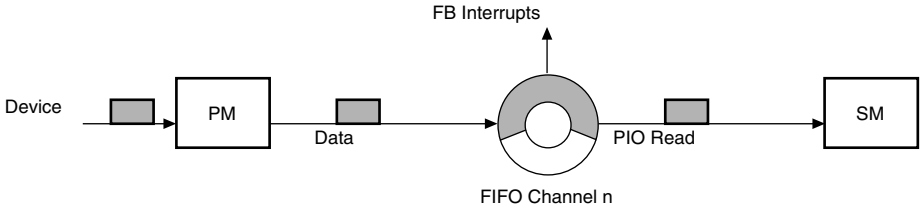
Scenario 8 Data Transfer Chart

See Scenario 6 for restriction.

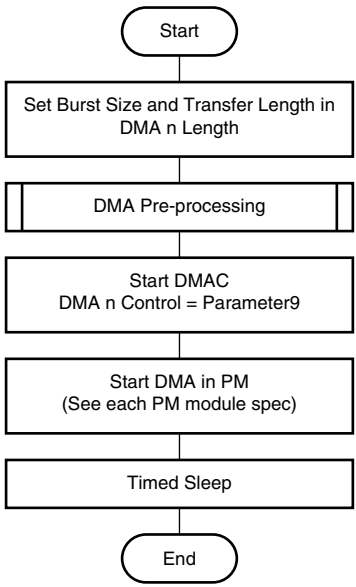
Flowcharts 8, 9 and 5 Replace Parameter6 of Parameter8

DMA n Control Register Parameter8
1: MM,DR,RTRA,DTRA,DBEN,TCEN
0: ML,RBEN,ENDD,CWS,FSEN,FBEN,PTEN
Valid: CSEL,CWD,ENDS

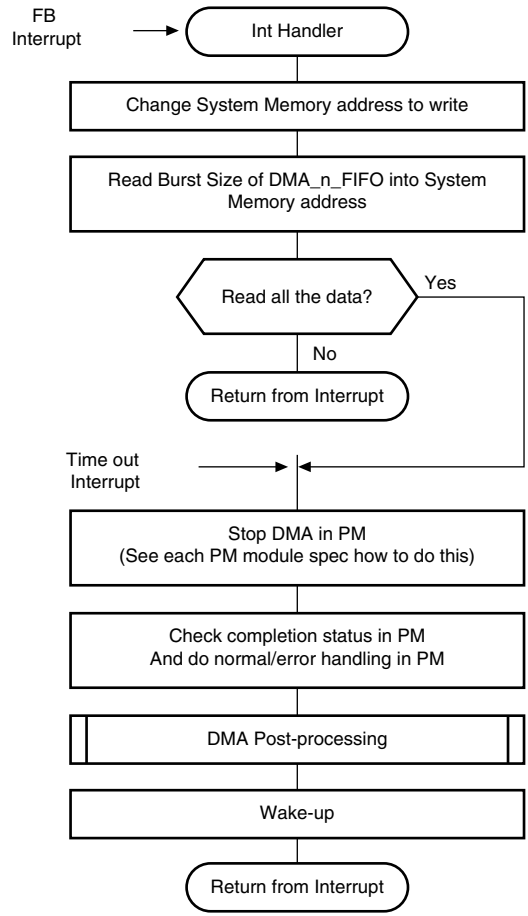
Scenario 9 PM → FIFO → SM Fixed length data transfer mode



Scenario 9 Data Transfer Chart



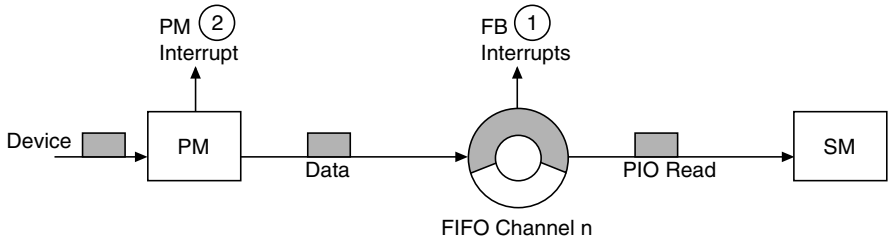
Flowchart 10



Flowchart 11

DMA n Control Register Parameter9
1: RTRA,FBEN
0: MM, ML,RBEN,DR,DBEN,CWD,ENDS, DTRA,FSEN, PTEN, TCEN,
Valid: CSEL,CWS,ENDD

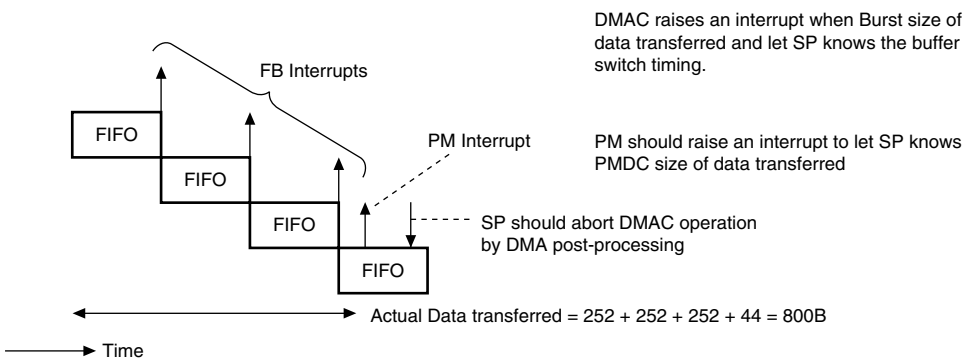
Scenario 10 PM → FIFO → SM Continuous data transfer mode



Scenario 10 Data Transfer Chart

This mode only works when PM has DMA counter (PMDC) and completion interrupt (PM Interrupt) because DMAC cannot stop DMA at specific DMA count but just abort DMA. See below.

Example scenario: PMDC = 800B, DMA_n_Length.Burst Size = 252B (Notice MAX = 252B)

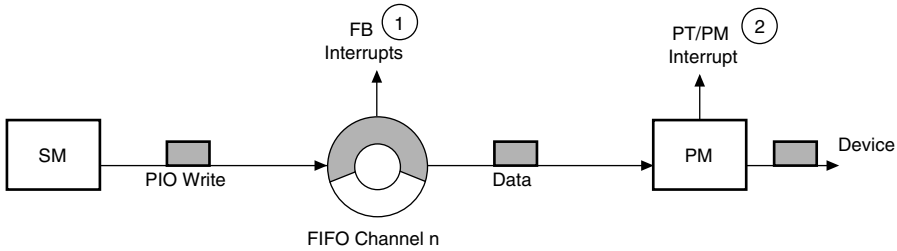


Scenario 10 Data Timing Chart

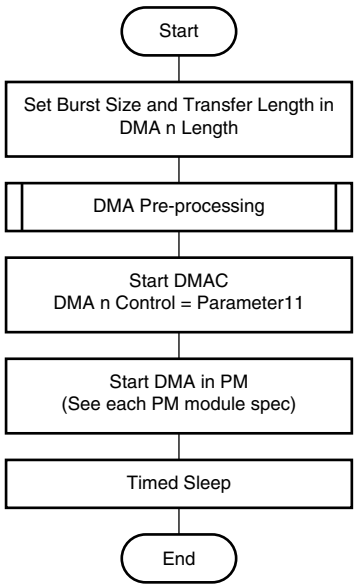
Flowcharts 10 and 11 Replace Parameter9 of Parameter10 and
Time Out Interrupt of "Time Out Interrupt or PM Interrupt"

DMA n Control Register Parameter10
1: RTRA,FBEN,DBEN
0: MM, ML,RBEN,DR,CWD,ENDS, DTRA,FSEN, PTEN, TCEN,
Valid: CSEL,CWS,ENDD

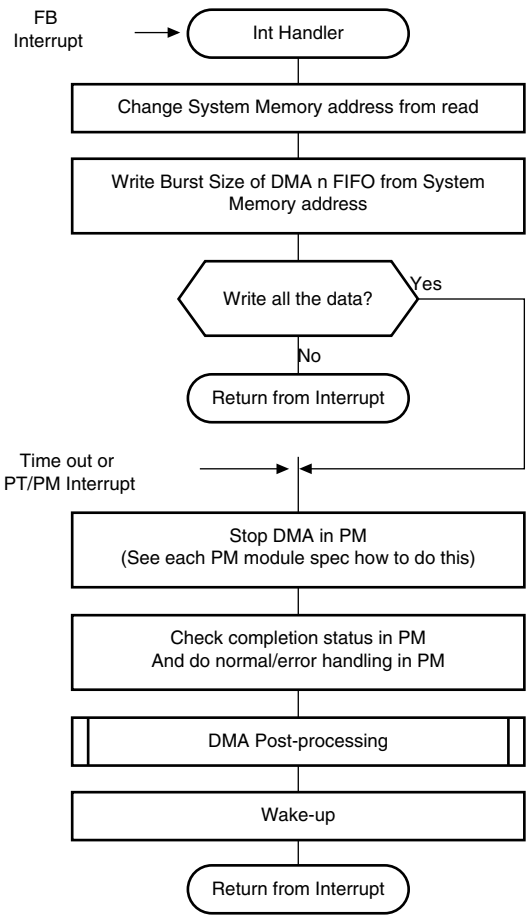
Scenario 11 SM → FIFO → PM Fixed length data transfer mode



Scenario 11 Data Transfer Chart



Flowchart 12

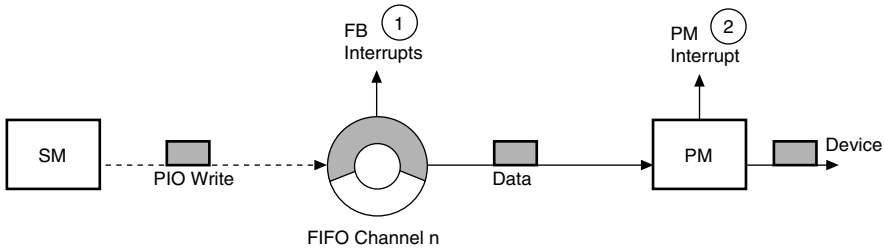


Flowchart 13

In this mode, either PT Interrupt or PM interrupt should be used.

DMA n Control Register Parameter11
1: RTRA,FBEN,DR
0: MM, ML,RBEN,DBEN,CWS,ENDD, DTRA,FSEN, TCEN,
Valid: CSEL,CWD,ENDS,PTEN

Scenario 12 SM → FIFO → PM Continuous data transfer mode



Scenario 12 Data Transfer Chart

See Scenario 10 restriction.

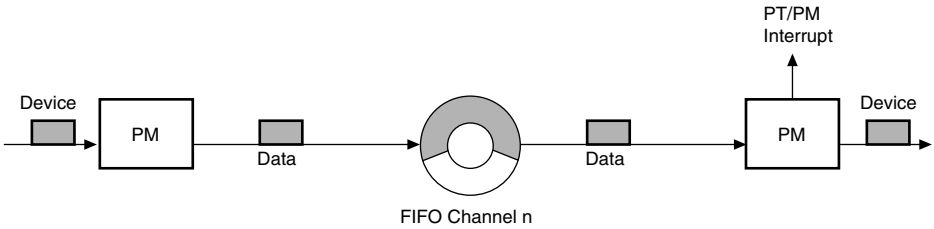
Difference between Scenario 11 and 12 is the maximum size of data transfer.

Scenario 11: DMA n Length Register, Scenario 12: decided by PM

Flowcharts 12 and 13 Replace Parameter11 of Parameter12

DMA n Control Register Parameter12
1: RTRA,FBEN,DR,DBEN
0: MM, ML,RBEN,CWS,ENDD, DTRA,FSEN, TCEN,PTEN
Valid: CSEL,CWD,ENDS

Scenario 13 PM → FIFO → PM Fixed length data transfer mode



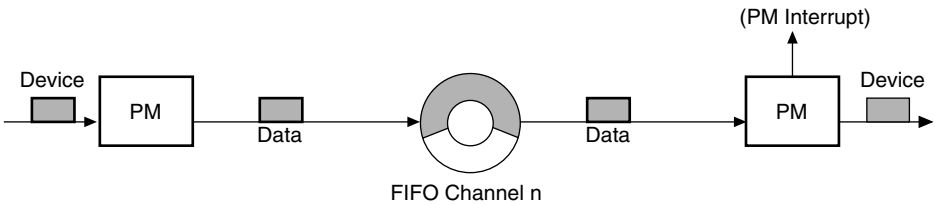
Scenario 13 Data Transfer Chart

In this mode, either PT Interrupt or PM interrupt should be used.

Flowcharts 6 and 7 Replace Parameter5 of Parameter13

DMA n Control Register Parameter13
1: RTRA,RBEN
0: MM, ML,DR,DBEN,CWS,ENDD,CWD,ENDS,DTRA,FSEN,FBEN,TCEN
Valid: CSEL,PTEN

Scenario 14 PM → FIFO → PM Continuous data transfer mode



Scenario 14 Data Transfer Chart

In this mode, PM interrupt could be used.

SP should know somehow when to stop this DMA.

Flowcharts 6 and 7 Replace Parameter5 of Parameter14

When PM interrupt is not used. SP should directly execute interrupt handler procedure to stop DMA.

DMA n Control Register Parameter14
1: RTRA,RBEN,DBEN
0: MM, ML,DR,CWS,ENDD,CWD,ENDS,DTRA,FSEN,FBEN,TCEN,PTEN
Valid: CSEL

Scenario 15 PM → FIFO → PM Fixed length data transfer mode

Scenario 13 Data Transfer Chart

In this mode, either PT Interrupt or PM interrupt should be used.

Flowcharts 6 and 7 Replace Parameter5 of Parameter15

DMA n Control Register Parameter15
1: RTRA,RBEN,DR
0: MM, ML,DBEN,CWS,ENDD,CWD,ENDS,DTRA,FSEN,FBEN,TCEN
Valid: CSEL,PTEN

Scenario 16 PM → FIFO → PM Continuous data transfer mode

Scenario 14 Data Transfer Chart

In this mode, PM interrupt could be used.

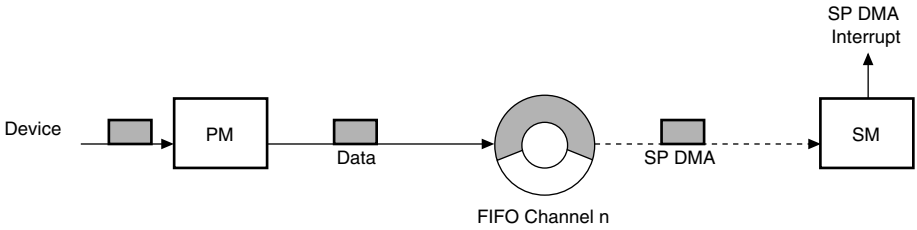
SP should know somehow when to stop this DMA.

Flowcharts 6 and 7 Replace Parameter5 of Parameter16

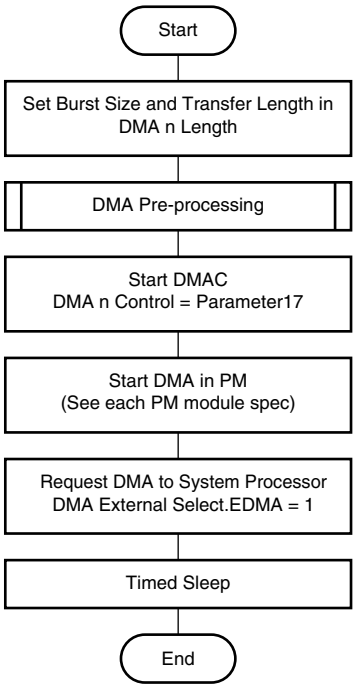
When PM interrupt is not used. SP should directly execute interrupt handler procedure to stop DMA.

DMA n Control Register Parameter16
1: RTRA,RBEN,DR,DBEN
0: MM, ML,CWS,ENDD,CWD,ENDS,DTRA,FSEN,FBEN,TCEN,PTEN
Valid: CSEL

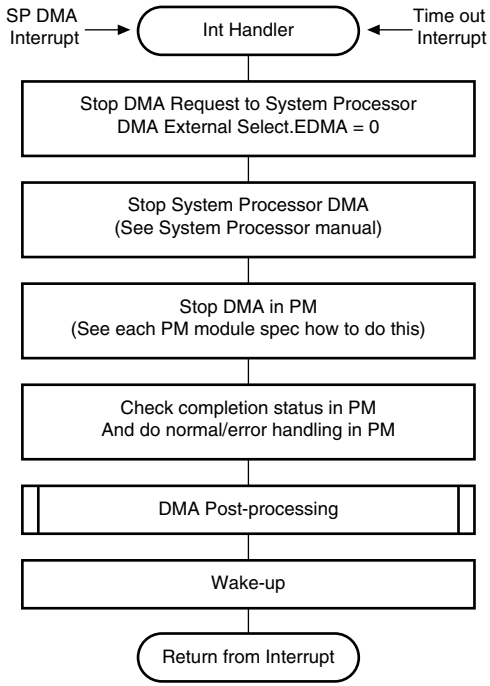
Scenario 17 PM ➔ FIFO ⇨ SM External DMA mode



Scenario 17 Data Transfer Chart

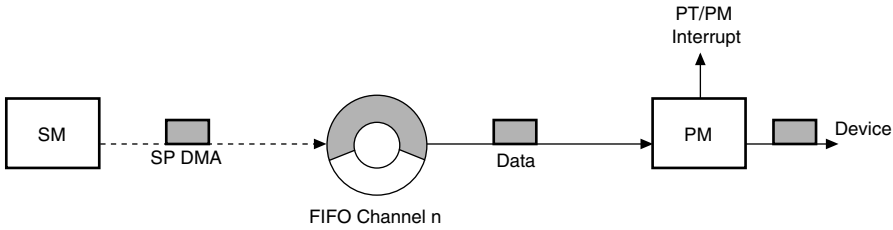


Flowchart 14

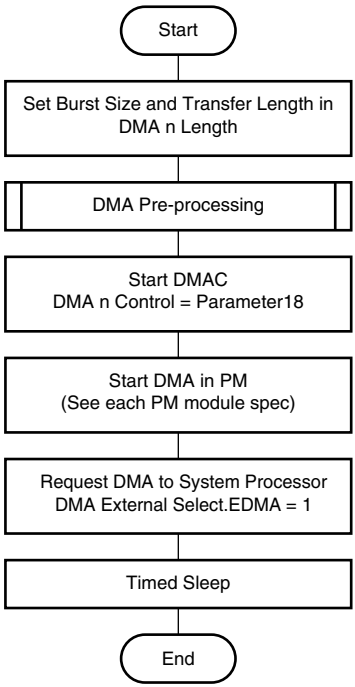


Flowchart 15

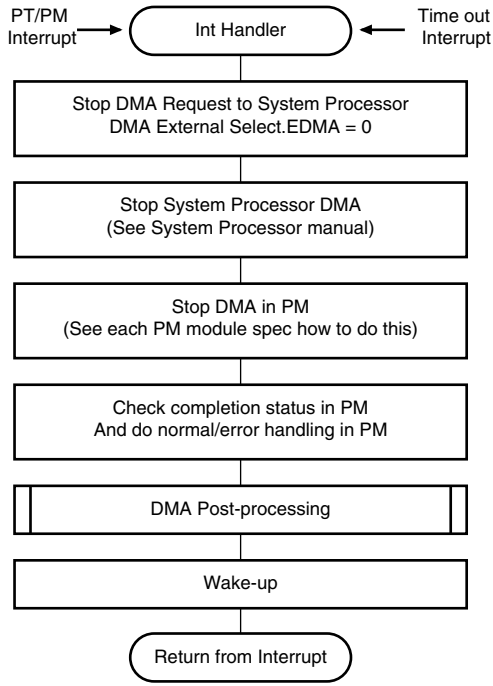
DMA n Control Register Parameter17
1: RTRA
0: MM,ML,DTRA,RBEN,DR,DBEN,CWD,ENDS,FSEN,FBEN,PTEN,TCEN
Valid: CSEL,CWS,ENDD



Scenario 18 Data Transfer Chart



Flowchart 16



Flowchart 17

In this mode, either PT Interrupt or PM interrupt should be used.

DMA n Control Register Parameter18
1: RTRA,DR
0: MM,ML,DTRA,RBEN,DBEN,CWS,ENDD,FSEN,FBEN,TCEN
Valid: CSEL,CWD,ENDS,PTEN

DMA Control Parameter Summary

DMA n Control Register Parameter for each scenario is summarized below.

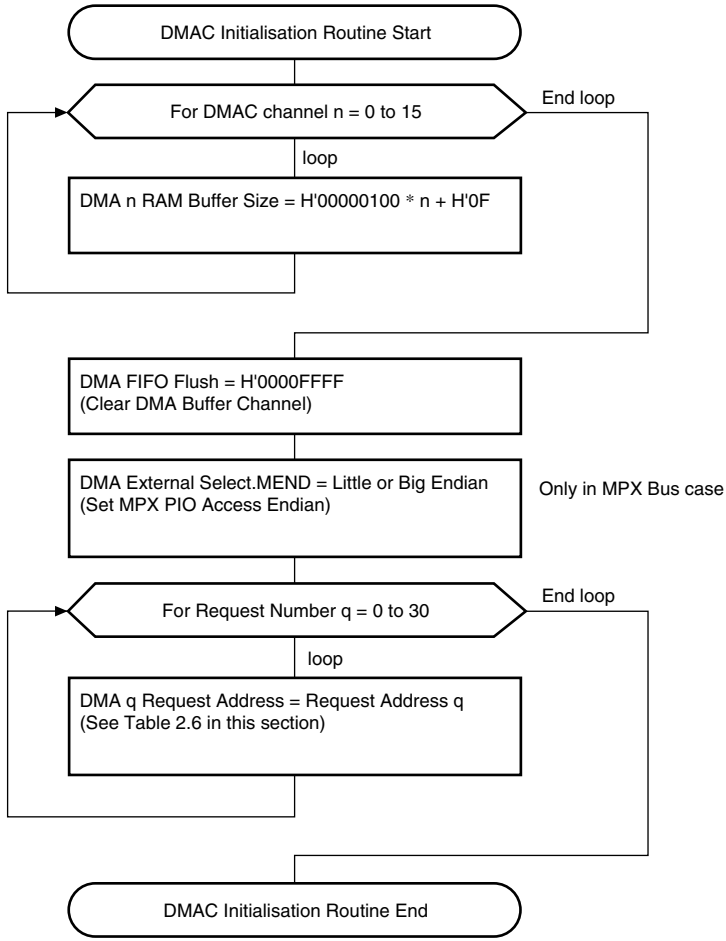
Control Parameter = Fixed Parameter + Variable Parameters

Table 2.18 DMA n Control Parameters Summary

		Variable Parameters Fields/Bit Mask			
		PTEN	CSEL	ENDD/CWS	ENDS/CWD
Parameter No (Scenario No)	Fixed Parameter	H'00200000	H'001F0000	H'00008003	H'0000400C
1	H'00000AD0	—	CSEL	CWS	—
2	H'00000ED0	—	CSEL	CWS	—
3	H'000008D0	—	CSEL	ENDD/CWS	—
4	H'00000CD0	—	CSEL	ENDD/CWS	
5	H'000002F0	PTEN	CSEL	—	CWD
6	H'00000EF0	—	CSEL	—	CWD
7	H'000000F0	PTEN	CSEL	—	ENDS/CWD
8	H'00000CF0	—	CSEL	—	ENDS/CWD
9	H'00002010	—	CSEL	ENDD/CWS	—
10	H'00002410	—	CSEL	ENDD/CWS	—
11	H'00002030	PTEN	CSEL	—	ENDS/CWD
12	H'00002430	—	CSEL	—	ENDS/CWD
13	H'00000110	PTEN	CSEL	—	—
14	H'00000510	—	CSEL	—	—
15	H'00000130	PTEN	CSEL	—	—
16	H'00000530	—	CSEL	—	—
17	H'00000010	—	CSEL	ENDD/CWS	—
18	H'00000030	PTEN	CSEL	—	ENDS/CWD
19 (Stop DMAC)	H'00000000	—	CSEL	—	—

2.12.9 DMAC Initialisation Procedure

DMAC should be initialised during the system boot strap sequence as below.

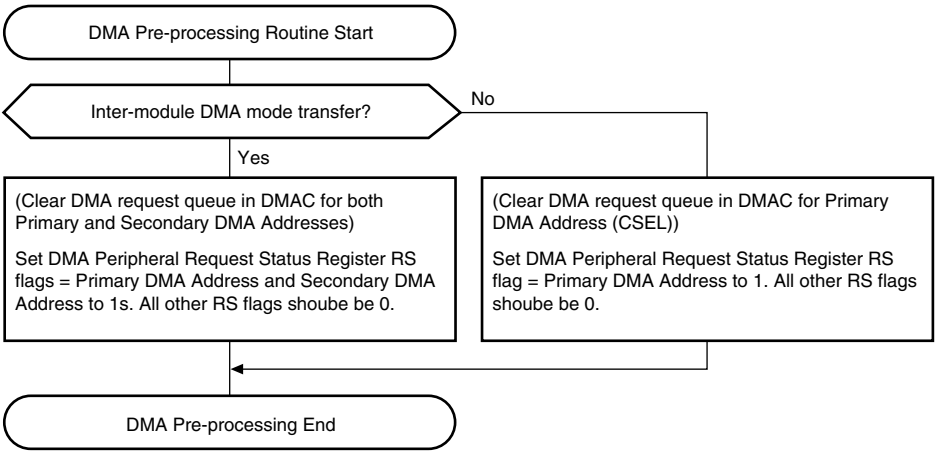


Flowchart 18

2.12.10 DMA Pre-, Post- and Abort processing

DMA Pre-processing

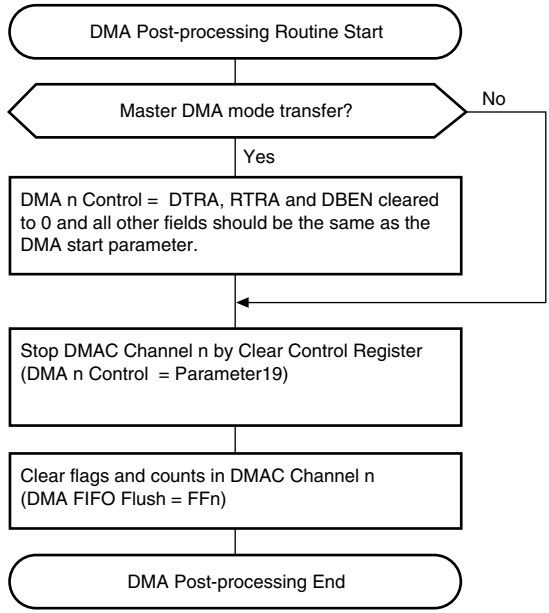
There are some Peripheral Modules, which will raise spurious DMA request after successful completion of DMA data transfer. In order to clear this spurious DMA request, DMA Pre-processing is necessary. In order to avoid a racing condition between PM and DMAC, DMA Pre-processing should be conducted before starting DMA. DMA Pre-processing routine flowchart is shown next



Flowchart 19

DMA Post-processing

Procedure below is necessary to stop DMAC after completion interrupts for all DMA modes.



Flowchart 20

DMA n Control Parameter19 (Stop DMAC Channel)
1: —
0: MM, ML, RTRA, DTRA, DR, TCEN, RBEN, DBEN, ENDD, CWD, CWS, ENDS, FSEN, FBEN, PTEN
Valid: CSEL

CSEL should be specified, otherwise if CSEL = 0, Request Number 0 Channel will be stopped.

DMA Abort Processing

Aborting DMA during data transfer should be avoided as much as possible since data will corrupt. But there are some specific situations when DMA abort processing is required.

1. In Inter-module DMA mode, if data transfer completion interrupt in Peripheral Module is not used, DMA abort processing should be conducted.
2. When CPU detects time out for DMA transfer, due to underrun error or others, DMA abort processing should be conducted. This procedure is already described in the flowchart 2 through 19.

DMA Abort Processing procedure is exactly same as the data transfer completion interrupt handling procedures already described for each scenario. There is no way to know how many bytes are correctly transferred before DMA Abort Processing.

DMA Interrupt Handling

See Interrupt Priority Module Pseudo Code for 1st Level HD64404 Interrupt Handling Procedure. Also see the DMAC interrupt scenario handler described in " Data Transfer procedure for each DMA scenario" section in this note.

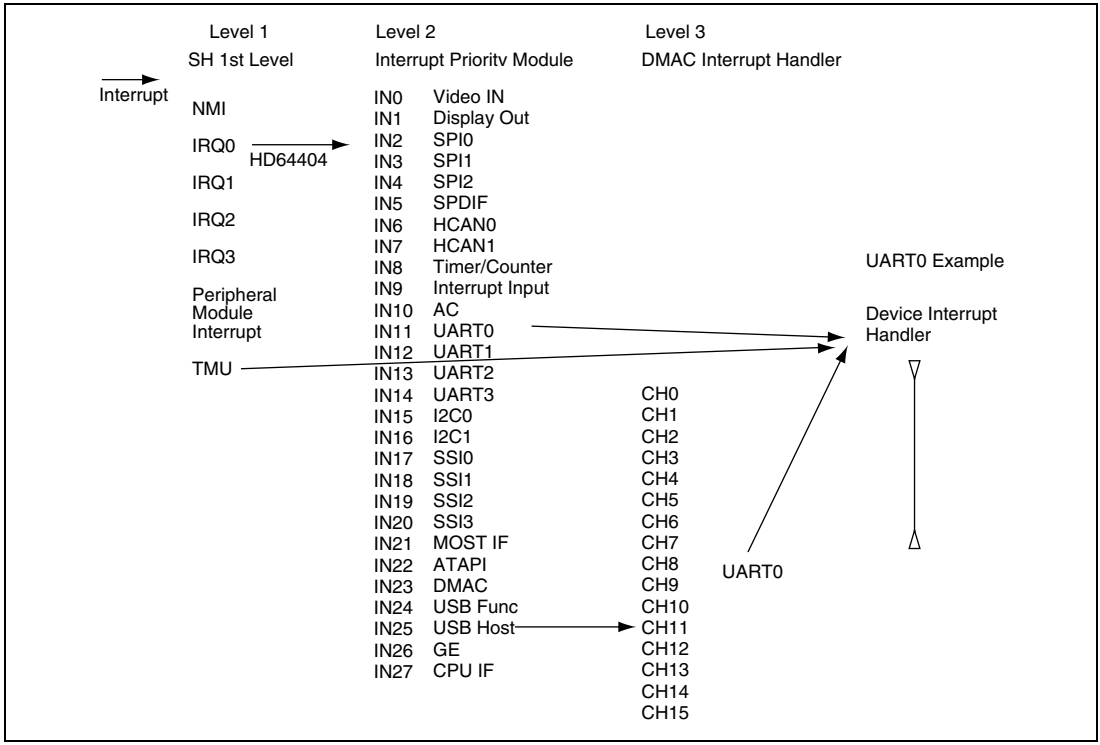


Figure 2.4 Interrupt Handling Hierarchy in HD64404: UART0 Example

```
Int32  intSrc, intChannel;
// DMAC Interrupt Handler
// Interrupt Number 25
{
intSrc = read DMA_Interrupt_Source
intChannel = find highest priority channel in intSrc  // Algorithm is system
dependent
check whether this interrupt on intChannel is expected interrupt;
Get the Scenario by consulting with CMT(intChannel);
switch ( Interrupt type) {
   case TC:
            call respective interrupt scenario handler (intChannel);
           // clear TC interrupt. where n = intChannel;
            mask all DMAC interrupt;
            write 0 to TCn bit in DMA_Status;
            restore DMAC interrupt mask;
            break;
   case PT:
            call respective interrupt scenario handler (intChannel);
           // clear PT interrupt. where n = intChannel;
            mask all DMAC interrupt;
            write 0 to PTn bit in DMA_Status;
            restore DMAC interrupt mask;
            break;
   case FB:
            call respective interrupt scenario handler (intChannel);
           // clear FB interrupt. where n = intChannel;
            mask all DMAC interrupt;
            write 0 to FBn bit in DMA_FIFO_Status;
            restore DMAC interrupt mask;
            break;
   case FS:   // currently no use case for FS, add procedure if necessary
            break;
}

// Peripheral module interrupt handler
// using this DMAC
// Interrupt Number X
```

```
// In each interrupt scenario handler, peripheral module interrupt should be
cleared
{
    get the interrupt cause and check it whether this is expected interrupt or
not;
     // channel number should be stored in driver table;
     intChannel = get the DMAC channel number;
     call respective interrupt scenario handler (intChannel);
}

// DMAC software watch-dog timer interrupt handler
// Interrupt Number 8
{
     if this interrupt is not for DMAC software watch-dog then {
         other processing
    }
    else { // DMAC software time out
         get the interrupt cause and check it whether this is expected interrupt
or not;
         clear timer interrupt;   // Timer/Counter module specification
         // channel number should be stored in timer table;
         intChannel = get the DMAC channel number;
         call respective interrupt scenario handler (intChannel);
    }
}
```

Table 2.19 DMA Mode Table

		Data Transfer Entity and Direction																					Data Transfer Option									DMA Start Trigger	Flow Control							DMA Stop Trigger Exclusively used between DMA abort and Interrupt Enable								Int Handler Aid		DMA Completion Status Check						Error Handling & Recovery Procedure	
		DMA Mode			CPU I/F	Primary DMA Address			DMA FIFO Chanel Buffer	DMA/ PIO		Secondary DMA Address			Direc tion	Conti nuous/ fixed length transfer	Pack/Unpack Endian Conversion			MPX Endian FB Control	Transfer Length	CPU write		Interupt				CPU must abort DMA	Enable Transfer Completion Interrupt					CPU Timer	Intrpt Src & Mask	DMAC Status							FIFO Flush	Dummy DMA Cycle
ScenarioNumber	Definition	MM	RBEN	EDMA	PCI/MPX	CSEL & Re q_Addr	PM		FIFO		SM/GM/PM	ML	Start_Address	DR	DBEN	CWS/ENDD*	CWD/ENDS*	MEND	Burst Size	Length	RTRA	DTRA	TCEN	PTEN	FSEN	FBEN	RTRA	DTRA	TCEN	PTEN	PM Intrpt	SP DMA Intrpt	Timed Out to detect underrun	CI/CM	PT PCOUNT	TC MCOUNT	FS	FB	Req_Status	FF	DDEN
1	Master DMA	1	0	0	PCI	√	PM	→	FIFO	→	SM	1	External Memory Address	0	0 (Fixed)	8/16/32/ Little	—(00)/ —(0)	—	—	√	1	1	1	—	—(0)	—(0)	—	—	1	—(0)	0	—(0)	1	√	—	√	—	—	Must	√	√
2															1 (Cont.)											0	0			1										
3					PCI/ MPX		PM		FIFO		GM	0			0 (Fixed)	8/16/32/ Big/Little										—	—			0										
4															1 (Cont.)											0	0			1										
5					PCI		PM	←	FIFO	←	SM	1		1	0 (Fixed)	—(00)/ —(0)	8/16/32/ Little						—	—			—	—	—(0)	0/1	1/0			√	—							
6															1 (Cont.)								1			0	0	1	—(0)	1			—	√							
7					PCI/ MPX		PM		FIFO		GM	0			0 (Fixed)		8/16/32/ Big/Little						—			—	—	—(0)	0/1	1/0			√	—							
8															1 (Cont.)								1			0	0	1	—(0)	1			—	√							
9	Slave DMA	0	0	0	PCI/ MPX	√	PM	→	FIFO	→	SM	—(0)	—(0)	0	0 (Fixed)	8/16/32/ Big/Little	—(00)/ —(0)	—	√	√	1	0	—	—	0	1	—	—	—(0)	0	0	—(0)	1	√	√	—	√	√	Must read	√	√
10															1 (Cont.)					—						0	0		—(0)	1			—							
11							PM	←	FIFO	←	SM			1	0 (Fixed)	—(00)/ —(0)	8/16/32/ Big/Little		√							—	—		0/1	1/0			√							
12															1 (Cont.)					—						0	0		—(0)	1			—							
13	Inter module DMA	0	1	0	PCI/ MPX	√	PM	→	FIFO	→	PM	—(0)	Req_ number	0	0 (Fixed)	—(00)/ —(0)	—(00)/ —(0)	—	—	√	1	0	—	—	—(0)	—(0)	—	—	—(0)	0/1	1/0	—(0)	1	√	√	—	——		Must read	√	√
14															1 (Cont.)					—						0	0		—(0)	1			—							
15							PM	←	FIFO	←	PM			1	0 (Fixed)				√							—	—		0/1	1/0			√							
16															1 (Cont.)					—						0	0		—(0)	1			—							
17	External DMA	0	0	1	MPX	√	PM	→	FIFO	⇨	SM	—(0)	—(0)	0	—(0)	8/16/32/ Big/Little	—(00)/ —(0)	—	√	√	1	—(0)	—	—	—(0)	—(0)	—	—	—(0)	0	0	1	1	√	√	—	——		Must read	√	√
18							PM	←	FIFO	⇦	SM			1		—(00)/ —(0)	8/16/32/ Big/Little												0/1	1/0	0										

Note : * ENDS/ENDD is always Little Endian if CPU I/F is PCI Bus.

2.12.11 Software Test Case: DMAC Flowchart for MIM

Scope

This section will describe the configuration for the DMAC for operation of the MOST Interface Module (MIM) in two example test cases. For information on the configuration of the Peripheral Modules in the test cases below, MIM and I²S, refer to the respective Peripheral Module specifications.

Case 1:
Transmission of 4 audio signals (each 4 bytes/ MOST frame) to a MOST node.
Channel 0: SSI0 to MIM1 Scenario13 Length:2048 bytes

Case 2:
Transmission of 32 video signals (each 32 bytes/ MOST frame) to a MOST node.
Channel 2: SM to MIM3 Scenario5 Length:2048 bytes

Programming

Assumption was made that channel configuration is already done. See DMAC initialisation procedure section for detail. Also for DMA interrupt handling, see "DMA interrupt handling" section in this note. On interrupt, general flow is below,

1. 1st Level: HD64404 interrupt handler // See interrupt priority module specification
2. 2nd Level: DMA interrupt handler // See DMA interrupt handling section
3. 3rd Level: DMA Scenario handler // See below for MIM and also see each scenario
   // flowchart in this note

Pseudo code descriptions

```
// Case1 pseudo code
// Transmission of 4 audio signals (each 4 byte/ MOST frame) to a MOST node.
// Channel 0:  SSI0 to MIM1  Scenario13    Length:2048bytes
//  start channel 0 routine
#define MIM1_RQ 2
#define SSI0_RQ 6
#define CSELSHIFT 16
#define Parameter13 H'00000110
#define PTEN  H'00200000
channel_0_start( ){
    DMA_0_Start_Address = MIM1_RQ;   // set  MIM1  request number
    DMA_0_Start_Length = H'800;
    Call DMA_Pre_processing( channel = 0, DMA_type = Inter-module );
    DMA_0_Control = Parameter13 | PTEN | (SSI0_RQ << CSELSHIFT);
   Start DMA in SSI0 and MIM1;  // See SSI and MIM module specifications
   Wait_for_interrupt ( sleepChannel, timeout);
   Check for completion status;
   Wakeup user task if necessary;
}
// channel 0 scenario interrupt handler
channel_0_interrupt ( ch:channel)  {
    stop DMA in SSI0 and MIM1;  // See SSI and MIM module specification
    check completion status in PM;
    call DMA_post_Processing(channel = ch, DMA_type=Inter-module);
    wakeup_driver_thread ( sleepChannel);
}


// Case2 pseudo code
// Transmission of 32 video signals (each 32 byte/ MOST frame) to a MOST node.
// Channel 2:  SM to MIM3  Scenario5    Length:2048bytes //  start channel 0
routine
#define MIM3_RQ 4
#define CSELSHIFT 16
#define Parameter5 H'000002F0
#define PTEN  H'00200000
#define CWD_CH2  0
```

```
channel_2_start( mem_addr){
    DMA_2_Start_Address = mem_addr;   // set  SM address
    DMA_2_Start_Length = H'800;
    Call DMA_Pre_processing( channel = 2, DMA_type = Master-DMA );
    DMA_2_Control = Parameter5 | PTEN | CWD_CH2 | (MIM3_RQ << CSELSHIFT);
   Start DMA in MIM3;  // See MIM module specifications
   Wait_for_interrupt ( sleepChannel, timeout);
   Check for completion status;
   Wakeup user task if necessary;
}
// channel 2 scenario interrupt handler
channel_2_interrupt ( ch:channel)  {
    stop DMA in MIM3;  // See MIM module specification
    check completion status in PM;
    call DMA_post_Processing(channel = ch,DMA_type=Master-DMA);
    wakeup_driver_thread(sleepChannel);
}
```

Section 3 MPX I/F

3.1 General Description

MPX i/f provides Address and Data multiplex CPU interface. MPX i/f supports up to 100-MHz operation though it also depends on the busload. It enables the burst access that is controlled by D[63:61]. The system configuration chooses either MPX i/f or PCI i/f.

MPXi/f and PCI i/f also support linear addressing to tile addressing conversion.

Note that SH7751 D[31:29] pins have to be connected to both D[31:29] and D[63:61] in HD64404. Please refer to ″3.6 Functional Description″.

3.2 Features

- Address/Data multiplex CPU interface
- Slave mode only
- 1-/2-/4-/8-/32-Byte access
- Pin multiplex with PCI interface
- Super H DMAC transfer mode support (write only, write to Graphic Memory)
- Linear addressing to tile addressing conversion support
  (supported only for the transfer between GM and MPX i/f module)

Note: When HD64404 power on sequence is executed, then whole address area of Graphics Memory is Tiled space. But System processor DMA is selected Linear space transfer in the power on sequence. Please refer to LTAD, LTAM and DTMR2 register.

- Support 2 modes of chip select: 128-MB address space mode and 64-MB address space mode

3.3 External interface (MPX Bus)

Table 3.1 Pin Configuration

Signal	Function	Direction
D[63:61]	Burst Access Size	IN
D[31:0]	Bi-directional Multiplexed address/data bus	IN/OUT
$\overline{\text{FRAME}}$	Frame start cycle	IN
$\overline{\text{RDY}}$	Slave ready signal	OUT
$\overline{\text{BS}}$	Bus Start	IN
RD/$\overline{\text{WR}}$	Read/Write signal	IN
SH4_$\overline{\text{CSA}}$	Chip select A This defines an area of 64 Mbytes	IN
SH4_$\overline{\text{CSB}}$	Chip select B This defines an area of 64 Mbytes	IN
CKIO	MPX bus clock from SH-4	IN
$\overline{\text{IRL}}$	Interrupt to the SH-4	OUT
$\overline{\text{DREQ}}$	SH7751 DMA request	OUT
DRAK	SH7751 DMA request acknowledge	IN
DACK	SH7751 DMA transfer acknowledge	IN
$\overline{\text{RST}}$	System reset	IN

3.4 Block Diagram

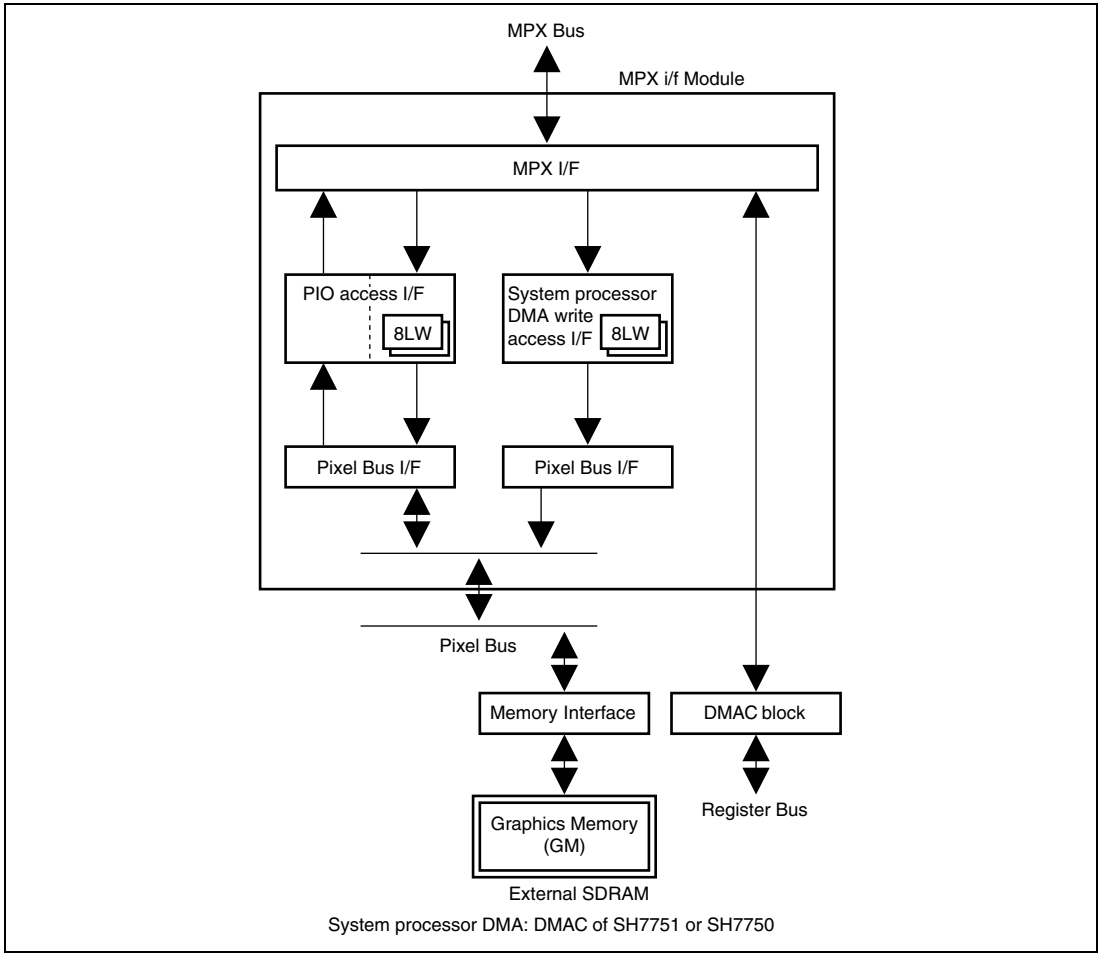


Figure 3.1 Block Diagram

3.5 Register Description

There is a set of registers which are located in the address space of MPX.

3.5.1 MPX interface registers

Table 3.2 MPX i/f Register Map

Address (Bytes)	Register Name	Abbreviation	Access Size
H'9000	CPU System Control Register	SYSR	32
H'9004	CPU Status Register	SR	32
H'9008	CPU Status Register clear register	SRCR	32
H'900C	CPU Interrupt Enable Register	IER	32
H'9010	Data Transfer Mode Register	DTMR	32
H'9014	Data Transfer Mode Register2	DTMR2	32
H'9018	DMA Transfer Word Count Registers	DMAWR	32
H'901C	Linear to Tile Convert Address Register	LTAD	32
H'9020	Linear to Tile Convert Address MASK	LTAM	32
H'9024	CPU System Control Register 2	SYSR2	32
H'9028	MPX ConTroL Register	MPXCTL	32
H'903C	Auxiliary System Control Registers	SYSR_AUX	32

Legends for register description:

Initial Value : Register value after reset
— : Undefined value
R/W : Read and write register
R/WC : When reading, always 0 is read. 0 write is ignored. 1 write enable to clear the related status register.
R : Read only register , for write always 0 write

CPU System Control Register (SYSR)

Register Address: H'00

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										DMAEC		DTRS0	DMA			
Initial:	—	—	—	—	—	—	—	—	—	0	—	0	0	0	—	—
R/W:	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	R	**Reserved**
6	DMAEC	0	R/W	**DMA Endian Convert (DMAEC)** 0: Endian disable 1: Endian enable
5	—	—	R	**Reserved**
4	DTRS0	0	R/W	**DMA Endian transfer unit (DTRS0)** 0: Data is Byte boundary 1: Data is Word boundary
3, 2	DMA	0	R/W	**DMA mode (DMA)** On these bits, 0 write is not available. Write 1 when necessary. (0, 0): DMA disable mode (Initial value) (0, 1): DMA write mode to Graphis Memory This bit is cleared automatically when the number of data transfer left is equal to zero. The number of data transfer is initially set by the DMAWR register. (1, 0) (1, 1): Reserved
1, 0	—	—	R	**Reserved**

CPU Status Register (SR)

Register Address: H'04

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DMF
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	**Reserved**
0	DMF	0	R	**DMA Flag (DMF): Read only register** 0: Indicating DMA transfer mode has not been initiated at all since DMF flag clearing by the DMCL bit in SRCR, or the next DMA transfer mode has been initiated and the remaining transfer count has not reached 0. 1: Indicating DMA transfer mode has been initiated and the transfer word count has reached 0.

—: Indicates undefined

CPU Status Register Clear Register (SRCR)

Register Address: H'08

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DMCL
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	**Reserved**
0	DMCL	0	R/WC	**DMA flag clear (DMCL): clear DMF** 0: It is ignored. 1: The DMF bit of the related register (SR) is cleared in 0.

Note: —: Indicates undefined
R/WC: When reading, always 0 is read. 0 write is ignored. 1 write is enabled to clear the related status register.

CPU Interrupt Enable Register (IER)

Register Address: H'0C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DME
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	**Reserved**
0	DME	0	R/W	**DMA flag enable (DME)** 0: Interrupts initiated by the DMF flag in SR are disabled. 1: Interrupts initiated by the DMF flag in SR are enabled.

—: Indicates undefined

The Interrupt Enable Register (IER) is a 32-bit readable/writable register that enables or disables interrupts by the corresponding flags in the Status Register (SR). When a bit in SR is set to 1 and the bit at the corresponding bit position in the IER register is also 1, $\overline{IRL}$ is driven low and an interrupt request is sent to the CPU.

The interrupt generation condition is as follows.

Interrupt generation condition = $\overline{IRL} = \bar{a}$

a = DMF · DME

Data Transfer Mode Register (DTMR)

Register Address: H'10

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											CPUMWX1	CPUMWX0				CPUGBM
Initial:	—	—	—	—	—	—	—	—	—	—	0	0	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R/WC

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	—	R	**Reserved**
5	CPUMWX1	0	R/W	**Memory width for CPU soft rendering write (CPUMWX)**
4	CPUMWX0	0	R/W	The Memory width for image data in case of CPU soft rendering write. (bit 5, bit 4) (0, 0): 512 pixels (0, 1): 1024 pixels (1, 0): 2048 pixels (1, 1): 4096 pixels
3 to 1	—	—	R	**Reserved**
0	CPUGBM	0	R/W	**CPU rendering graphic bit mode (CPUGBM)** 1: 8 bits/pixel 0: 16 bits/pixel

Correspondence between Memory Physical Addresses (Bytes) and Rendering Coordinates and Multi-valued Source Coordinates

8 bits/pixel (CPUGBM = 1), 512 pixels (CPUMWX = 0) Y(vertical) address = A[25:9], X(horizontal) address = A[8:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:13]													A[8:5]				A[12:9]				A[4:0]				

8bits/pixel (CPUGBM=1) , 1024 pixels (CPUMWX = 1) Y(vertical)address = A[25:10], X(horizontal) address = A[9:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:14]													A[9:5]				A[13:10]				A[4:0]				

8bits/pixel (CPUGBM=1) , 2048 pixels (CPUMWX = 2) Y(vertical)address = A[25:11], X(horizontal) address = A[10:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:15]											A[10:5]						A[14:11]				A[4:0]				

8bits/pixel (CPUGBM=1) , 4096 pixels (CPUMWX = 3) Y(vertical) address = A[25:12], X(horizontal) address = A[11:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:16]										A[11:5]							A[15:12]				A[4:0]				

16bits/pixel (CPUGBM=0) , 512 pixels (CPUMWX = 0) Y(vertical) address = A[25:10], X(horizontal) address = A[9:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:14]													A[9:5]				A[13:10]				A[4:1]				0

16bits/pixel (CPUGBM=0) , 1024 pixels (CPUMWX = 1) Y(vertical) address = A[25:11], X(horizontal) address = A[10:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:15]											A[10:5]						A[14:11]				A[4:1]				0

16bits/pixel (CPUGBM=0) , 2048 pixels (CPUMWX = 2) Y(vertical) address = A[25:12], X(horizontal) address = A[11:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:16]										A[11:5]							A[15:12]				A[4:1]				0

16bits/pixel (CPUGBM=0) , 4096 pixels (CPUMWX = 3) Y(vertical) address = A[25:13], X(horizontal) address = A[12:0]

A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[25:17]									A[12:5]								A[16:13]				A[4:1]				0

Upper line: Memory physical addresses (bytes)
Lower line: Logical coordinates (X, Y)

Data Transfer Mode Register2 (DTMR2)

Register Address: H'14

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											DMAMWX1	DMAMWX0			DMAGBM	TileEn
Initial:	—	—	—	—	—	—	—	—	—	—	0	0	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	—	R	**Reserved**
5	DMAMWX1	0	R/W	**Memory width for DMA data transfer write (DMAMWX)**
4	DMAMWX0	0	R/W	The Memory width for image data in case of DMA write. (bit 5, bit 4) (0, 0): 512 pixels (0, 1): 1024 pixels (1, 0): 2048 pixels (1, 1): 4096 pixels
3, 2	—	—	R	**Reserved**
1	DMAGBM	0	R/W	**DMA data transfer graphic bit mode (DMAGBM)** 1: 8 bits/pixel 0: 16 bits/pixel
0	TileEn	0	R/W	**DMA data transfer to Tile Space or Linear Space (TileEn)** 1: Tiled Space 0: Linear Space

Correspondence between Memory Physical Addresses (bytes) and Rendering Coordinates and Multi-valued Source Coordinates: Same as DTMR. See DTMR register description.

DMA Transfer Word Count Register (DMAWR)

The DMA Transfer Word Count Register (DMAWR) is a 24 bit readable/writable register that specifies the number of transfer counts in DMA transfer.

Value set in this register should be the same as the value in DMATCR register of SH7751 (or SH7750) DMA function.

If the value of this register is modified during a series of DMA operations from the time bits DMA1 and DMA0 in the system control register (SYSR) are set to 01 by the CPU until they are cleared automatically by the HD64404, operation will be unstable.

If 0s are written to all the bits or the register is set to its initial value, the maximum value (DMA count 16777216 times) is set.

This register is not decremented when DMA transfer is performed.

Register Address: H'18

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									DMAWH							
Initial:	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMAWL															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	**Reserved**
23 to 16	DMAWH	0	R/W	
15 to 0	DMAWL	0	R/W	

Linear to Tile Convert Address Register (LTAD)

Register Address: H'1C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LTAD										
Initial:	—	—	—	—	—	0	0	0	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
26	LTAD26	0	R/W	The start address of Linear to Tile conversion available for CPU soft rendering write or read. LTAD26-0 (always LTAD18 – 0 are all 0) indicates the local start address of Linear to Tile conversion available. LTAD is defined by HD64404 linear address mapping used in pixel bus. HD64404 can be configured as either 64-MB address mapped device or 128-MB address mapped device by UMM64Mbit in MPXCTL register. LTAD[26:0] can cover 128-MB addressing space.
25	LTAD25	0	R/W	
24	LTAD24	0	R/W	
23	LTAD23	0	R/W	
22	LTAD22	0	R/W	
21	LTAD21	0	R/W	
20	LTAD20	0	R/W	
19	LTAD19	0	R/W	
18 to 0	—	—	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

Linear to Tile Convert Address MASK (LTAM)

Register Address: H'20

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LTAM										
Initial:	—	—	—	—	—	0	0	0	0	0	0	0	0	—	—	—
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
26	LTAM26	0	R/W	**Mask register of LTAD (LTAM)**
25	LTAM25	0	R/W	LTAMn indicates the mask bit of LTADn.
24	LTAM24	0	R/W	1: Related LTAD bit is valid. 0: Related LTAD bit is invalid. The available values of LTAM[26:19] are as follows: H'00, H'80, H'C0, H'E0, H'F0, H'F8, H'FC, H'FE, H'FF The set of other values of LTAM[26:19] is prohibited. For example (1) LTAD[26:19] = b'11111111 LTAM[26:19] = b'11111000 Linear to Tile convert region is where a[26] – a[22] in the local address is b'11111. The allocated space is 4MB Example (2) LTAD[26:19] = b'01010101 LTAM[26:19] = b'11111100 Linear to Tile convert region is where a[26] – a[21] in the local address is b'010101. The allocated space is 2MB. This function is available for CPU soft rendering write. For example (3) LTAD[26:19] = b'0000 0000(default) LTAM[26:19] = b'0000 0000(default) Linear to Tile convert region is where a[26] – a[19] in the local address is don't care. This means the allocated space is 128-MB(the whole address area is now Tiled space.) For example (4) LTAD[26:19] = b'0000 0000(default) LTAM[26:19] = b'1111 1111 Linear to Tile convert region is where a[26] – a[19] in the local address is b'0000 0000. The allocated space is 512KB. This function is available for CPU soft rendering write.
23	LTAM23	0	R/W	
22	LTAM22	0	R/W	
21	LTAM21	0	R/W	
20	LTAM20	0	R/W	
19	LTAM19	0	R/W	
18 to 0	—	—	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

Register Address: H'24

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DMUCL
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	**Reserved**
0	DMUCL	0	R/WC	**DMA Force Clear (DMUCL)** 0: DMA write operation is not cleared 1: DMA write operation clear SYSR[3:2] will be cleared as (0, 0).

—: Indicates undefined

R/WC: When reading, always 0 is read. 0 write is ignored. 1 write is enabled to clear the related status register.

Note: Write is possible for this register only at the time of DMA write mode. (B'01(DMA) is written to the address H'00(SYSR).) It is reset again after the DMA compulsory clear completion.

MPX ConTroL Register (MPXCTL)

Register Address: H'28

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								RDYN_END_HIGH				MPXED				UMM64M
Initial:	—	—	—	—	—	—	—	1	—	—	—	0	—	—	—	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															RBCLKEN	PIXCLKEN
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/WC

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	—	R	**Reserved**
24	RDYN_END_HIGH	1	R/W	**$\overline{\text{RDY}}$ drive high (RDYN_END_HIGH)** 0: $\overline{\text{RDY}}$ is driven low at the end of bus access cycle 1: $\overline{\text{RDY}}$ is driven high at the end of bus access cycle. $\overline{\text{RDY}}$ is a tristate signal and goes to Hiz when SH4_$\overline{\text{CSA}}$/SH4_$\overline{\text{CSB}}$ is deasserted.
23 to 21	—	—	R	**Reserved**
20	MPXED	0	R/W	**MPX i/f Endian Mode: Set Endian Mode on MPX bus** 0: Big Endian 1: Little Endian This bit indicates which endian the SH processor is configured as.
19 to 17	—	—	R	**Reserved**

Bit	Bit Name	Initial Value	R/W	Description
16	UMM64M	0	R/W	**UMM 64-Mbyte mode (UMM64M)** 0: The memory area mode of the 128-Mbyte This is intended for the system using both SH4_$\overline{CSA}$ and SH4_$\overline{CSB}$ Each chip select indicates 64-Mbyte area. 1: The memory area mode of the 64 Mbyte This is intended for the system using only SH4_$\overline{CSB}$, that is, 64-Mbyte space is allocated to HD64404 including Graphic memory. HD64404 has a linear addressing memory and peripheral I/O space as follows.*
15 to 2	—	—	R	**Reserved**
1	RBCLKEN	0	R/W	**PLL output clock control signals (RBCLKEN, PIXCLKEN)** After PLL outputs clocks and rbclk and pix-clk are stable, these flags will be set as 1. Please refer to Power Control & Configuration block specification too.
0	PIXCLKEN	0	R/W	

Signal name	Initial	rbclk and pix_clk are stable
RBCLKEN	0	1
PIXCLKEN	0	1

—: Indicates undefined

Note:* 1) UMM64M = 0 (default)

HD64404 address mapped space	Chip select	Super H address D[25:0]	Pixel bus address [26:0] used in DMAC module
Graphic Memory	SH4_$\overline{CSA}$	H'0000 0000 to H'03FF FFFF	H'0000 0000 to H'03FF FFFF
	SH4_$\overline{CSB}$	H'0000 0000 to H'03FE FFFF	H'0400 0000 to H'07FE FFFF
Peripheral	SH4_$\overline{CSB}$	H'03FF 0000 to H'03FF FFFF	Not available for pixel bus This area is used on register bus address space.

2) UMM64M = 1

HD64404 address mapped space	Chip select	Super H address D[25:0]	Pixel bus address [26:0] used in DMAC module
Graph Memory	SH4_$\overline{CSB}$	H'0000 0000 to H'03FE FFFF	H'0000 0000 to H'03FE FFFF
Peripheral	SH4_$\overline{CSB}$	H'03FF 0000 to H'03FF FFFF	Not available for pixel bus This area is used on register bus address space.

Auxiliary System Control Registers (SYSR_AUX)

Auxiliary System Control registers (SYSR_AUX) are 32 bit readable/writable registers that specify special extension modes for HD64404.

Register Address: H'3C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																PIO CLR
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															CPU EC	DTR S2
Initial:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	—	R	**Reserved**
16	PIOCLR	0	R/W	**PIO memory access logic initialization (PIOCLR)** 0: No initialized 1: Initialized
15 to 2	—	—	R	**Reserved**
1	CPUEC	0	R/W	**CPU i/f soft read/write Endian Conversion (CPUEC)** 0: No endian conversion 1: Endian Conversion enable
0	DTRS2	0	R/W	**CPU soft read/write Endian Conversion** This bit is effective when Bit1 = 1. 0: Byte 1: Word

Note: It is desirable that bit1 and bit 0 should not be used in a standard system because they require the processor software control for the different kinds of data transfer. The recommendation is the processor and Graphic Memory are using the same endian in either Big or Little.

3.6 Functional Description

3.6.1 General Functionality

MPX interface supports Address/Data multiplex CPU interface. CPU configuration chooses either PCI interface or MPX interface. It supports the burst access controlled by D[63:61].

The address is output to D25 to D0, and the access size to D63 to D61.

For details of access sizes and data alignment, see section 13.3.1 in SH7751 manual, Endian/Access Size and Data Alignment.

Table 3.3 MPX Access Size

D63	D62	D61	Access Size
0	0	0	BYTE
		1	WORD
	1	0	LONG WORD
		1	QUAD WORD
1	X	X	32BYTE WORD

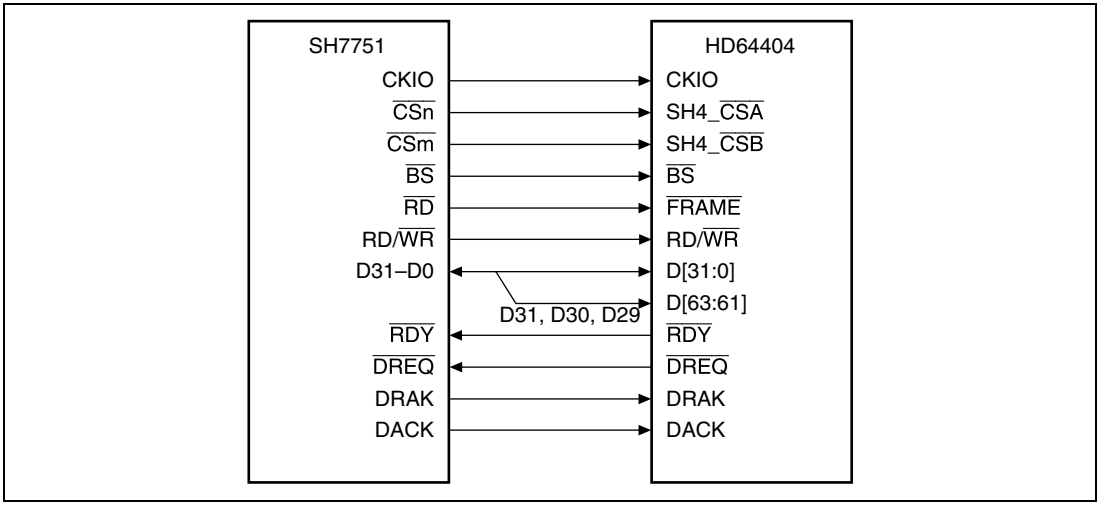


Figure 3.2 Example of SH7751 connection

SH4_$\overline{\text{CSA}}$, SH4_$\overline{\text{CSB}}$

SH4_$\overline{\text{CSA}}$: Chip selects A. This defines an area of 64MBytes.
SH4_$\overline{\text{CSB}}$: Chip selects B. This defines an area of 64MBytes.

Peripheral I/O in HD64404 is the bottom 64Kbyte space (H'03FF0000 to H'03FFFFFF) on SH4 CSB.

HD64404 can be defined as either 64-MB address mapped device or 128-MB address mapped device by the register MPXCTL UMM64M bit. See the MPXCTL register description.

DMA Writes

The CPU can perform write DMA access, using cycle stealing, to the Graphic Memory. To perform DMA access, DMA Transfer Count Register (DMAWR), and System Control Register DMA mode must be made. After the DMA mode settings are made, the HD64404 drives the $\overline{\text{DREQ}}$ signal low as soon as its preparations are completed. When the DMA controller receives this signal, it drives the DACK signal high and begins DMA access. The destination address (Graphic Memory address) is set as the DMA transfer start address register (DAR) in SH7751 and this value has to be set as 32 Byte boundary address. The number of words set in the DMA Transfer Word Count Register (DMAWR) are transferred. DMAWR has to be set to be equal to DMATCR in SH7751 DMAC.

DMA transfer is performed using dual address transfer timing. In this case, access to the HD64404 should be performed by driving DACK high. The DMA mode is set to 01 for UGM access. Other address-mapped registers cannot be accessed. The destination address (UGM address) is set to the DMA Transfer Start Address Register (DMSARH, DMSARL), and the number of words set to the

DMA Transfer Word Count Register (DMAWR) are transferred. Addresses input from off-chip are not used.

When making another DMA mode setting after DMA transfer ends, first check that the DMF bit is set to 1 in the status register.

When using the DMAC, make the following DMAC settings in SH7751. For DMA transfer in dual address mode

1. DACK output in write cycle
2. Active-high DACK output
3. DMA destination start address, which is the initial address for Graphic memory, is 32 Byte address boundary.
4. Source address incremented
5. External request, dual address mode
6. $\overline{\text{DREQ}}$ falling-edge detection

Do not write to SYSR and DMAWR register until DMA is completed after initiating.

Notes:
1. Take the following procedure for setting DMA transfer.
   a) Set the DMA transfer of SH4 (dual address and cycle steal mode). Refer to an SH4 hardware manual for more details.
   b) Set 1 to the DMA transfer end interrupt bit (DME) of the IER register.
   c) Set a DMA transfer word count to the DMAWR register. The set value must be the same as that of DMATCR n set in a).
   d) Set H'01 to the DMA transfer start bit (DMA) of the SYSR register to start DMA transfer.
   e) When the DMA transfer count reaches the set value in c), a DMA transfer end interrupt (the DMF bit of the SR register or $\overline{\text{IRL}}$ for external interface) is issued to SH-4. Then, set 1 to the DMA transfer end interrupt clear bit (DMCL) of the SRCR register to clear the DMF bit.

   Repeat steps a), c), d), and e) on and after the second DMA transfer.
2. If 1 set to the DMUCL bit of the SYSR2 register to forcibly stop the DMA transfer, the setting procedure of this module for DMA transfer must be taken again.

Data coherency between CPU writes data and DMAC DMA data

When CPU writes data to SDRAM thorough MPX I/F and then DMAC's DMA is initiated and DMA data is read through pixel bus, it can possibly happen that DMAC would read SDRAM data before CPU finishes to write data to SDRAM because a write buffer in MPX IF makes some delay depending on pixel bus round robin arbitration mechanism.
In order to avoid this case, the following procedure has to be taken to initiate DMAC′s DMA read from SDRAM.

1. After CPU writes the last data or before DMAC's DMA is initiated, dummy read operation is executed. This dummy read guarantees the write buffer in MPX IF is flushed and last data is correctly stored in SDRAM.
2. DMAC′s pixel bus DMA is initiated by writing DMA_n_Control register.

Section 4 PCI I/F

4.1 General Description

The PCI Controller (PCIC) controls the PCI bus and transfers data between Graphic memory or registers (peripheral) inside HD64404 and a PCI device connected to the PCI bus. The ability for PCI devices to be connected directly not only facilitates the design of systems using PCI buses but also enables systems to be more compact and capable of high-speed data transfer.

4.2 Features

The PCIC has the following features:

- Supports PCI version 2.2;
- Compatible with PCI bus operating speeds of 33 MHz;
- Compatible with 32-bit PCI bus
- Up to four PCI master devices running at 33 MHz
- Can operate as master or target
- When operating as master, DMA transfer are available;
- Two DMA transfer channels
- Four 32-bit × 16 longword internal FIFO (one for target reading, one for target writing, and two for DMA transfer)
- Support non-host mode only
- Linear addressing to tile addressing conversion support (in the data transmission to/from Graphic Memory)

Note: When HD64404 power on sequence is executed, then whole address area of Graphics Memory is Tiled space. Please refer to PCILTAD, PCILTAM and PCITILEMODE register.

4.3 Block Diagram

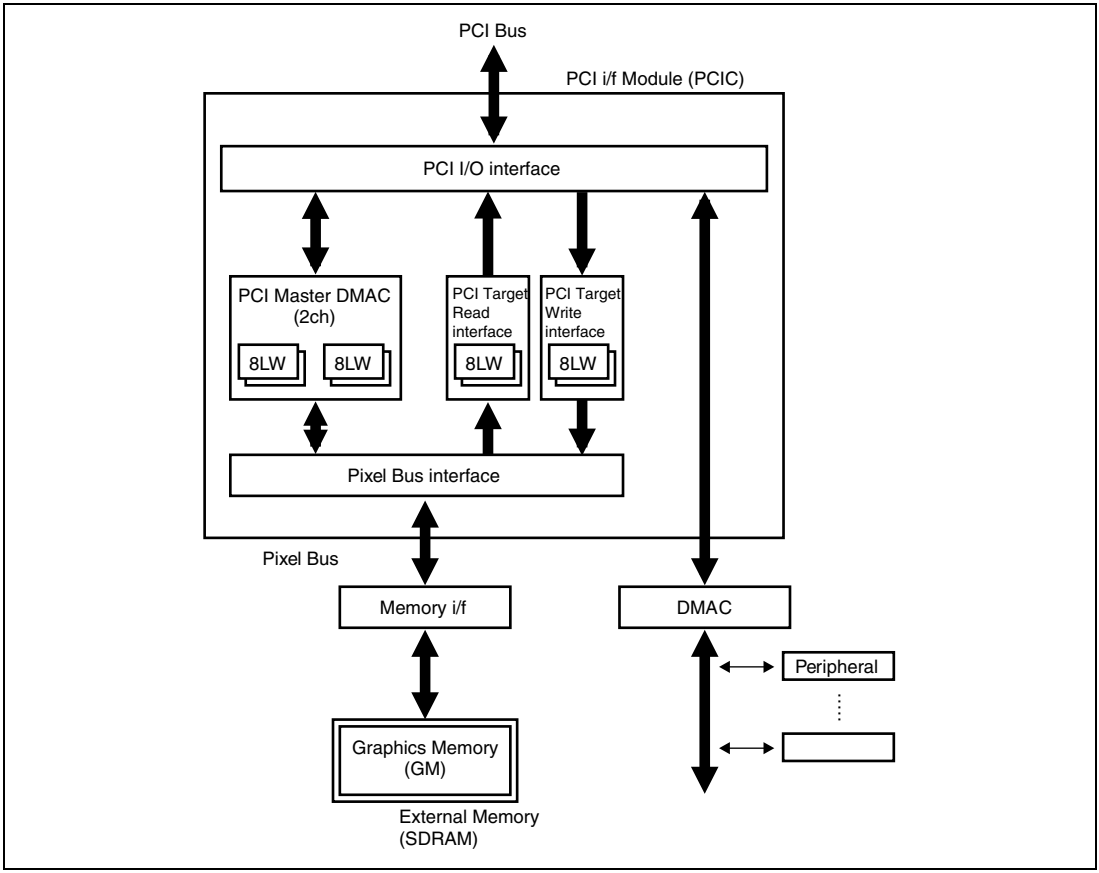


Figure 4.1 Block Diagram

4.4 External interface

Table 4.1 shows the configuration of I/O pins of the PCIC.

Table 4.1 Pin Configuration

No.	Pin Name	PCI Standard Signal Name	Function	IO Type	Pull-up Resistor	I/O status in Operating Mode		
						Master	Target	Remarks
1	PCI_CLK	CLK	PCI input clock (33 MHz)	in		I	I	
2	$\overline{\text{RST}}$	—	Reset input	in		I	I	
3	AD[31:0]	AD[31:0]	Address/data	t/s		I/O	I/O	
4	C/$\overline{\text{BE}}$[3:0]	C/BE[3:0]	Command/byte enable	t/s		O	I	
5	PAR	PAR	Parity	t/s		I/O	I/O	
6	$\overline{\text{FRAME}}$	FRAME	Bus cycle	s/t/s	◊	O	I	
7	$\overline{\text{IRDY}}$	IRDY	Initiator ready	s/t/s	◊	O	I	
8	$\overline{\text{TRDY}}$	TRDY	Target ready	s/t/s	◊	I	O	
9	$\overline{\text{STOP}}$	STOP	Transaction stop	s/t/s	◊	I	O	
10	$\overline{\text{DEVSEL}}$	DEVSEL	Device select	s/t/s	◊	I	O	
11	$\overline{\text{GNT}}$	$\overline{\text{GNT}}$	Bus grant	t/s		I	—	
12	$\overline{\text{REQ}}$	$\overline{\text{REQ}}$	Bus request	t/s		O	—	
13	$\overline{\text{PERR}}$	PERR	Parity error	s/t/s	◊	I/O	O	
14	$\overline{\text{SERR}}$	SERR	System error	o/d	♦	O	O	
15	$\overline{\text{INTA}}$	INTA	Interrupt (sync/async)	o/d	♦	O	O	(*)
16	IDSEL	IDSEL	Config device select	in		I	I	

Legend: ◊: Pull-up resistor required for s/t/s signals.

♦: Pull-up resistor required for o/d signals.

t/s: Tristate, s/t/s: sustained tristate, o/d: open drain

(*) asynchronous output for standby mode, synchronous output for normal mode

4.5 Register Configuration

The PCIC has the PCI Configuration Registers and PCI Control Registers shown in table 4.2, 4.3 and 4.4. Not only do these registers control the PCI bus but also enable high-speed data transfers between the PCI device and memory on HD64404.

Table 4.2 List of PCI Configuration Registers

Name	Abbreviation	PCI R/W	Initial Value	PCI Configuration Address (Byte Address)	Access Size
PCI configuration register 0	PCICONF0	R	H'350B1054	H'00	32
PCI configuration register 1	PCICONF1	R/W	H'02900080	H'04	32
PCI configuration register 2	PCICONF2	R/W[31:8] R (Rest)	H'xxxxxx**	H'08	32
PCI configuration register 3	PCICONF3	R/W[15:8] R (Rest)	H'00000000	H'0C	32
PCI configuration register 4	PCICONF4	R/W	H'00000001	H'10	32
PCI configuration register 5	PCICONF5	R/W	H'00000000	H'14	32
PCI configuration register 6	PCICONF6	R/W	H'00000000	H'18	32
PCI configuration register 7	PCICONF7	R	H'00000000	H'1C	32
PCI configuration register 8	PCICONF8	R	H'00000000	H'20	32
PCI configuration register 9	PCICONF9	R	H'00000000	H'24	32
PCI configuration register 10	PCICONF10	R	H'00000000	H'28	32
PCI configuration register 11	PCICONF11	R/W	H'xxxxxxxx	H'2C	32
PCI configuration register 12	PCICONF12	R	H'00000000	H'30	32
PCI configuration register 13	PCICONF13	R	H'00000040	H'34	32
PCI configuration register 14	PCICONF14	R	H'00000000	H'38	32
PCI configuration register 15	PCICONF15	R/W[7:0] R (Rest)	H'00000100	H'3C	32
PCI configuration register 16	PCICONF16	R/W[18:16] R (Rest)	H'00010001	H'40	32
PCI configuration register 17	PCICONF17	R/W[1:0] R (Rest)	H'00000000	H'44	32
Reserved	—	R	H'XXXXXXXX	H'48 to H'FC	32

* The register values will be changed according to the logic version.

X: the value is undefined

Table 4.3 PCI Configuration Register Configuration

PCI CFG Register Address	PCI Configuration Register				PCI R/W
	31 to 24	23 to 16	15 to 8	7 to 0	
H'00	Device ID	Device ID	Vendor ID	Vendor ID	R
H'04	Status	Status	Command	Command	R/W
H'08	Class code	Class code	Class code	Revision ID	R/W[31:8] R (Rest)
H'0C	BIST	Header type	PCI latency timer	Cache line size	R/W[15:8] R (Rest)
H'10	Base address (I/O)	Base address (I/O)	Base address (I/O)	Base address (I/O)	R/W
H'14	Base address (Memory 0)	Base address (Memory 0)	Base address (Memory 0)	Base address (Memory 0)	R/W
H'18	Base address (Memory 1)	Base address (Memory 1)	Base address (Memory 1)	Base address (Memory 1)	R/W
H'1C	Reserved	Reserved	Reserved	Reserved	R
H'20	Reserved	Reserved	Reserved	Reserved	R
H'24	Reserved	Reserved	Reserved	Reserved	R
H'28	Reserved	Reserved	Reserved	Reserved	R
H'2C	Subsystem ID	Subsystem ID	Subsystem vendor ID	Subsystem vendor ID	R/W
H'30	Reserved	Reserved	Reserved	Reserved	R
H'34	Reserved	Reserved	Reserved	Cap_ptr	R
H'38	Reserved	Reserved	Reserved	Reserved	R
H'3C	Max_Lat	Min_Gnt	Interrupt pin	Interrupt line	R/W[7:0] R (Rest)
H'40	Power management related	Power management related	Power management related	Power management related	R/W[18:16] R (Rest)
H'44	Power management related	Power management related	Power management related	Power management related	R/W[1:0] R (Rest)
H'48 to H'0FC	Reserved	Reserved	Reserved	Reserved	R

Table 4.4 List of PCIC Local Registers

Name	Abbreviation	PCI R/W	Initial Value	PCI I/O Byte Address	Access Size
PCI Control Register	PCICR	R/W	H'00000021	H'00	32
Local Space Register 0 for PCI	PCILSR0	R/W	H'00000000	H'04	32
Local Space Register 1 for PCI	PCILSR1	R/W	H'00000000	H'08	32
Local Address* Register 0 for PCI	PCILAR0	R/W	H'00000000	H'0C	32
Local Address Register 1 for PCI	PCILAR1	R/W	H'00000000	H'10	32
PCI Interrupt Register	PCIINT	R/W	H'00000000	H'14	32
PCI Interrupt Mask Register	PCIINTM	R/W	H'00000000	H'18	32
Error Address Data Register for PCI	PCIALR	R	H'xxxxxxxx	H'1C	32
Error Command Data Register for PCI	PCICLR	R	H'xx00000x	H'20	32
Reserved	—	—	H'xxxxxxxx	H'24 to H'2C	32
DMA Transfer Arbitration Register for PCI	PCIDMABT	R/W	H'00000000	H'30	32
Reserved	—	—	H'xxxxxxxx	H'34 to H'3C	32
DMA Transfer PCI Address Register 0 for PCI	PCIDPA0	R/W	H'xxxxxxxx	H'40	32
DMA Transfer HD64404 Starting Address Register 0 for PCI	PCIDLA0	R/W	H'00000000	H'44	32
DMA Transfer Count Register 0 for PCI	PCIDTC0	R/W	H'00000000	H'48	32
DMA Control Register 0 for PCI	PCIDCR0	R/W	H'00000000	H'4C	32
DMAPCI Address Register 1 for PCI	PCIDPA1	R/W	H'xxxxxxxx	H'50	32
DMA Transfer HD64404 Starting Address Register 1 for PCI	PCIDLA1	R/W	H'00000000	H'54	32
DMA Transfer Count Register 1 for PCI	PCIDTC1	R/W	H'00000000	H'58	32
DMA Control Register 1 for PCI	PCIDCR1	R/W	H'00000000	H'5C	32
Reserved	—	—	H'xxxxxxxx	H'60 to H'6C	32
PCI TRDY Enable Control Register	PCITRDYENB	R/W	H'00000000	H'70	32
Reserved	—	—	H'xxxxxxxx	H'74 to H'7C	32
PCI Tile Mode Register	PCITILEMODE	R/W	H'00000000	H'80	32
PCI Data Transfer Mode Register	PCIDTMR	R/W	H'00000000	H'84	32
PCI Linear toTile Address	PCILTAD	R/W	H'00000000	H'88	32
PCI Linear toTile Convert Address Mask	PCILTAM	R/W	H'00000000	H'8C	32

Name	Abbreviation	PCI R/W	Initial Value	PCI I/O Byte Address	Access Size
PCI Peripheral Base Address Register	PCIPAR	R/W	H'00000000	H'90	32
PCI Peripheral Space Register	PCIPSR	R/W	H'00000000	H'94	32
reserved	—	—	H'xxxxxxxx	H'98 to H'9C	32
PCI Pixel Bus Endian Register	PCIMD5R	R/W	H'00000000	H'A0	32
reserved	—	—	H'xxxxxxxx	H'A4 to H'DC	32
PLL Control Register in PCI mode	PCIPLLCTL	R/W	H'0000000C	H'E0	32
reserved	—	—	H'xxxxxxxx	H'E4 to H'EC	32
PCI TRDY enable wait cycle counter	PCITRDYCNT	R/W	H'0000000F	H'F0	32
reserved	—	—	H'xxxxxxxx	H'F4 to H'FC	32

Note: * Local address is the address that indicates Pixel bus and Register bus addressing space.

4.6 PCIC Register Descriptions

Legends for register description:

Initial Value	: Register value after reset
—	: Undifined value
R/W	: Read and Write , write value can be read.
R	: Read only , for write always 0 write
R/WC0	: Read and Write , 0 write clear , 1 write is ignored
R/WC1	: Read and Write , 1 write clear , 0 write is ignored
W	: Write only , Read prohibited . If reserved, write always 0.
—/W	: Write only, Read value undefined.

4.6.1 PCI Configuration Register 0 (PCICONF0)

PCI Configuration Register 0 (PCICONF0) is a 32-bit read-only register that includes the Device ID and Vender ID stipulated in the PCI local bus specifications. The Hitachi ID (H'1054) is read from bits 15 to 0.

All bits of the PCICONF0 are fixed in hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEV ID15	DEV ID14	DEV ID13	DEV ID12	DEV ID11	DEV ID10	DEV ID9	DEV ID8	DEV ID7	DEV ID6	DEV ID5	DEV ID4	DEV ID3	DEV ID2	DEV ID1	DEV ID0
Initial value	0	0	1	1	0	1	0	1	0	0	0	0	1	0	1	1
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VND ID15	VND ID14	VND ID13	VND ID12	VND ID11	VND ID10	VND ID9	VND ID8	VND ID7	VND ID6	VND ID5	VND ID4	VND ID3	VND ID2	VND ID1	VND ID0
Initial value	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	DEVID15	0	R	**DEVID15 to DEVID0**
30	DEVID14	0	R	These bits specify the device ID allocated by the PCI device vendor. (HD64404: fixed in hardware)
29	DEVID13	1	R	
28	DEVID12	1	R	
27	DEVID11	0	R	
26	DEVID10	1	R	
25	DEVID9	0	R	
24	DEVID8	1	R	
23	DEVID7	0	R	
22	DEVID6	0	R	
21	DEVID5	0	R	
20	DEVID4	0	R	
19	DEVID3	1	R	
18	DEVID2	0	R	
17	DEVID1	1	R	
16	DEVID0	1	R	
15	VNDID15	0	R	**DNVID15 to DNVID0**
14	VNDID14	0	R	These bits specify the PCI device maker (vendor ID). (Hitachi: fixed in hardware)
13	VNDID13	0	R	
12	VNDID12	1	R	
11	VNDID11	0	R	
10	VNDID10	0	R	
9	VNDID9	0	R	
8	VNDID8	0	R	
7	VNDID7	0	R	
6	VNDID6	1	R	
5	VNDID5	0	R	
4	VNDID4	1	R	
3	VNDID3	0	R	
2	VNDID2	1	R	
1	VNDID1	0	R	
0	VNDID0	0	R	

4.6.2 PCI Configuration Register 1 (PCICONF1)

Bits 31 to 27 and 24 are write-clear bits that are cleared when 1 is written to them.

PCI Configuration Register 1 (PCICONF1) is a 32-bit read/partial-write register that includes the status and Command stipulated in the PCI local bus specifications. The status is read from bits 31 to 16 (status register) in the event of an error on the PCI bus. Bits 15 to 0 (command register) contain the settings required for initiating transfers on the PCI bus.

The PCICONF1 register is initialized to H'02900080 at a power-on reset.

Always write to this register before initiating transfers on the PCI bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DPE	SSE	RMA	RTA	STA	DEV1	DEV0	DPD	FBBC	UDF	66M	PM				
Initial value:	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0
PCI-R/W:	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R	R	R/WC1	R	R/W	R/W	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PBBE	SER	WCC	PER	VPS	MWIE	SPC	BUM	MES	IOS
Initial value	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31	DPE	0	R/WC1	**Parity Error Detection Status (DPE)** 0: No parity error detected by device 1: Parity error detected by device Set this bit regardless of the parity error response bit (bit 6) on the device
30	SSE	0	R/WC1	**System Error Output Status (SSE)** 0: Device not asserting $\overline{\text{SERR}}$ 1: Device asserting $\overline{\text{SERR}}$
29	RMA	0	R/WC1	**Master abort receive status (RMA)** 0: No transaction termination using bus master abort 1: Detection by bus master of transaction termination by bus master abort. However, in the case of a master abort in a special cycle, notify the master devices that are not set.
28	RTA	0	R/WC1	**Target Abort Receive Status (RTA)** 0: No transaction termination using target abort 1: Detection by bus master of transaction termination by target abort
27	STA	0	R/WC1	**Target Abort Execution Status (STA)** Notification by master device. 0: No transaction termination using target abort by target device 1: Transaction termination by target abort by target device.
26	DEV1	0	R	**$\overline{\text{DEVSEL}}$ Timing Status (DEV1 and DEV0)**
25	DEV0	1	R	00: High-speed (not supported) 01: Medium speed 10: Low speed (not supported) 11: Reserved
24	DPD	0	R/WC1	**Data Parity Status (DPD)** 0: Data parity not detected 1: Data parity occurred. Data parity occurs when the following three conditions are satisfied: Bus master asserts $\overline{\text{PERR}}$, or detects $\overline{\text{PERR}}$; This device is the bus master when an error occurs; The parity error response bit (bit 6) is set.

Bit	Bit Name	Initial Value	PCI R/W	Description
23	FBBC	1	R	**High-Speed Back-To-Back Status (FBBC)** 0: The target does not have a high-speed back-to-back transaction function for use with other targets (not supported) 1: The target has a high-speed back-to-back transaction function for use with other targets
22	UDF	0	R/W	**User Defined Function System (UDF)** 0: This device does not support user functions 1: This device supports user functions (not supported)
21	66M	0	R/W	**66 MHz Operating Status (66M)** 0: This device supports 33-MHz operation 1: This device supports 66-MHz operation (not supported)
20	PM	1	R	**PCI Power Management (PM)** Extended function 0: Power management not supported 1: Power management supported
19 to 10	—	0	R	**Reserved**
9	PBBE	0	R	**High-Speed Back-To-Back Control (PBBE)** 0: Allows high-speed back-to-back control only with same target device 1: Allows high-speed back-to-back control with another device (not supported)
8	SER	0	R/W	**$\overline{\text{SERR}}$ Output Control (SER)** 0: $\overline{\text{SERR}}$ output disabled 1: $\overline{\text{SERR}}$ output enabled
7	WCC	1	R/W	**Wait Cycle Control (WCC)** 0: Disable address/data stepping control 1: Enable address/data stepping control
6	PER	0	R/W	**Parity Error Response (PER)** 0: Ignore detected parity errors 1: Respond to detected parity error
5	VPS	0	R	**VGA Pallet Snoop Control (VPS)** 0: VGA-compatible device 1: The device does not respond to pallet register writes (not supported)

Bit	Bit Name	Initial Value	PCI R/W	Description
4	MWIE	0	R	**Memory Write and Invalidate Control (MWIE)** 0: The device uses memory write 1: The device can execute memory write and invalidate commands (not supported)
3	SPC	0	R	**Special Cycle Control (SPC)** 0: Ignore special cycle 1: Monitor special cycle (not supported)
2	BUM	0	R/W	**PCI Bus Master Control (BUM)** 0: Disable bus master operation 1: Enable bus master operation
1	MES	0	R/W	**Memory Space Control (MES)** In target mode, this bit controls the access to memory space. When this bit is set to 0, all memory transfers to PCIC are ended by Master Abort. 0: Disable access to memory space 1: Enable access to memory space
0	IOS	0	R/W	**I/O Space Control (IOS)** In target mode, this bit controls the access to I/O space. When this bit is set to 0, all I/O transfers to PCIC are ended by Master Abort. 0: Disable access to I/O space 1: Enable access to I/O space

4.6.3 PCI Configuration Register 2 (PCICONF2)

The PCI Configuration Register 2 (PCICONF2) is a 32-bit read/partial-write register that includes the Class Code and Revision ID PCI Configuration Registers stipulated in the PCI local bus specifications. Bits 31 to 8 (class code) set the device functions. The chip logic version can be read from bits 7 to 0 (revision ID).

The PCICONF2 register class codes are not initialized at a reset.

Always initialize this register before PCI transaction is started.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLASS23	CLASS22	CLASS21	CLASS20	CLASS19	CLASS18	CLASS17	CLASS16	CLASS15	CLASS14	CLASS13	CLASS12	CLASS11	CLASS10	CLASS9	CLASS8
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CLASS7	CLASS6	CLASS5	CLASS4	CLASS3	CLASS2	CLASS1	CLASS0	REVID7	REVID6	REVID5	REVID4	REVID3	REVID2	REVID1	REVID0
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	CLASS23	—	R/W	**Base Class Code (CLASS23 to CLASS16)**
30	CLASS22	—	R/W	See table 4.5.
29	CLASS21	—	R/W	
28	CLASS20	—	R/W	
27	CLASS19	—	R/W	
26	CLASS18	—	R/W	
25	CLASS17	—	R/W	
24	CLASS16	—	R/W	
23	CLASS15	—	R/W	**Sub Class Codes (CLASS15 to CLASS8)**
22	CLASS14	—	R/W	For details, please see Appendix D of the PCI Local Bus Specifications, Revision 2.1.
21	CLASS13	—	R/W	
20	CLASS12	—	R/W	
19	CLASS11	—	R/W	
18	CLASS10	—	R/W	
17	CLASS9	—	R/W	
16	CLASS8	—	R/W	
15	CLASS7	—	R/W	**Register Level Programming Interface (CLASS7 to CLASS0)**
14	CLASS6	—	R/W	For details, please see Appendix D of the PCI Local Bus Specifications, Revision 2.1.
13	CLASS5	—	R/W	
12	CLASS4	—	R/W	
11	CLASS3	—	R/W	
10	CLASS2	—	R/W	
9	CLASS1	—	R/W	
8	CLASS0	—	R/W	
7	REVID7	—	R	**Revision ID (REVID7 to 0)**
6	REVID6	—	R	Shows the PCIC revision. The initial value differs according to the logic version of the chip. It is fixed in hardware.
5	REVID5	—	R	
4	REVID4	—	R	
3	REVID3	—	R	
2	REVID2	—	R	
1	REVID1	—	R	
0	REVID0	—	R	

Table 4.5 List of CLASS31 to 24 Base Class Codes

CLASS31 to 24 Base Class	Meaning
H'00	Device designed prior to class code being defined
H'01	High-capacity storage controller
H'02	Network controller
H'03	Display controller
H'04	Multimedia device
H'05	Memory controller
H'06	Bridge device
H'07	Simple communication device
H'08	Basic peripheral device
H'09	Input device
H'0A	Docking station
H'0B	Processor
H'0C	Serial bus controller
H'0D to H'FE	Reserved
H'FF	Device not categorized in defined class

4.6.4 PCI Configuration Register 3 (PCICONF3)

The PCI Configuration Register 3 (PCICONF3) is a 32-bit read/partial-write register that includes the BIST function, header type, latency timer, and Cache Line Size PCI Configuration Registers stipulated in the PCI local bus specification. The BIST function is read from bits 31 to 24, the header type from bits 23 to 16, the cache line size from bits 7 to 0. The guaranteed time for the PCIC to occupy the PCI bus when the PCIC is master is set in bits 15-8 (latency timer).

Bits 15 to 8 can be written to. Other bits are fixed in hardware.

The PCICONF3 register is initialized to H'00000000 at a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BIST7	BIST6	BIST5	BIST4	BIST3	BIST2	BIST1	BIST0	HEAD7	HEAD6	HEAD5	HEAD4	HEAD3	HEAD2	HEAD1	HEAD0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LAT7	LAT6	LAT5	LAT4	LAT3	LAT2	LAT1	LAT0	CACHE7	CACHE6	CACHE5	CACHE4	CACHE3	CACHE2	CACHE1	CACHE0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	BIST7	0	R	**BIST7** BIST function (not support) 0: Function not supported 1: Function supported (not supported)
30	BIST6	0	R	**BIST6** BIST starting (not supported) 0: Execution completed 1: Executing (not supported)
29	BIST5	0	R	**BIST5 and BIST4**
28	BIST4	0	R	Reserved bits.

Bit	Bit Name	Initial Value	PCI R/W	Description
27	BIST3	0	R	**BIST3 to BIST0**
26	BIST2	0	R	BIST status on completion of operation (not supported)
25	BIST1	0	R	
24	BIST0	0	R	H'0: Passed test H'1 to H'F: Test failed (not supported)
23	HEAD7	0	R	**Multifunction status (HEAD7)** 0: Only single-function devices supported. 1: Device has between 2 and 8 functions (not supported)
22	HEAD6	0	R	**Configuration Layout Type (HEAD6 to HEAD0)**
21	HEAD5	0	R	H'00: Configuration register address H'10 to H'3F layout supported H'01: Inter-PCI bridge configuration register address H'10 to H'3F layout supported (not supported) H'02 to H'3F: Reserved
20	HEAD4	0	R	
19	HEAD3	0	R	
18	HEAD2	0	R	
17	HEAD1	0	R	
16	HEAD0	0	R	
15	LAT7	0	R/W	**Latency Timer Register (LAT7 to LAT0)**
14	LAT6	0	R/W	Specifies the latency time of the PCI bus.
13	LAT5	0	R/W	
12	LAT4	0	R/W	
11	LAT3	0	R/W	
10	LAT2	0	R/W	
9	LAT1	0	R/W	
8	LAT0	0	R/W	
7	CACHE7	0	R	**Cache Line Size (CACHE7 to CACHE0)**
6	CACHE6	0	R	Not supported. Memory target is set cache-disabled, and SDONE and $\overline{\text{SBO}}$ are ignored.
5	CACHE5	0	R	
4	CACHE4	0	R	
3	CACHE3	0	R	
2	CACHE2	0	R	
1	CACHE1	0	R	
0	CACHE0	0	R	

4.6.5 PCI Configuration Register 4 (PCICONF4)

The PCI Configuration Register 4 (PCICONF4) is a 32-bit read/partial-write register that accommodates the I/O Space Base Address PCI Configuration Register stipulated in the PCI local bus specifications. The register holds the high 24 bits (bits 31 to 8) of the address used when a device on the PCI bus accesses a local register in the PCIC using I/O transfer commands. Allocate 256 Bytes of space as PCI bus I/O space.

Bits 31 to 8 can be written to. Bits 7 to 2, and 0, are defined in hardware.

The PCICONF4 Register is initialized to H'00000001 at a power-on reset.

Always write to this register prior to executing I/O transfers (accessing the local registers in the PCIC) to or from the PCIC from the PCI bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE31	BASE30	BASE29	BASE28	BASE27	BASE26	BASE25	BASE24	BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE15	BASE14	BASE13	BASE12	BASE11	BASE10	BASE9	BASE8	BASE7	BASE6	BASE5	BASE4	BASE3	BASE2		ASI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	BASE31	0	R/W	**BASE31 to BASE8**
30	BASE30	0	R/W	Base address of the local register (I/O space) in the PCIC
29	BASE29	0	R/W	
28	BASE28	0	R/W	
27	BASE27	0	R/W	
26	BASE26	0	R/W	
25	BASE25	0	R/W	
24	BASE24	0	R/W	
23	BASE23	0	R/W	
22	BASE22	0	R/W	
21	BASE21	0	R/W	
20	BASE20	0	R/W	
19	BASE19	0	R/W	
18	BASE18	0	R/W	
17	BASE17	0	R/W	
16	BASE16	0	R/W	
15	BASE15	0	R/W	
14	BASE14	0	R/W	
13	BASE13	0	R/W	
12	BASE12	0	R/W	
11	BASE11	0	R/W	
10	BASE10	0	R/W	
9	BASE9	0	R/W	
8	BASE8	0	R/W	
7	BASE7	0	R	**BASE7 to BASE2**
6	BASE6	0	R	All 0: Fixed in hardware
5	BASE5	0	R	
4	BASE4	0	R	
3	BASE3	0	R	
2	BASE2	0	R	
1	—	0	R	**Reserved**
0	ASI	1	R	**Address Space Indicator (ASI)** 0: Memory space (Not supported) 1: I/O space

4.6.6 PCI Configuration Register 5 (PCICONF5)

The PCI Configuration Register 5 (PCICONF5) is a 32-bit read/partial-write register that accommodates the Memory Space Base Address PCI Configuration Register stipulated in the PCI local bus specifications. This register holds the high bits (12 max. in bits 31 to 20) of the address used when a device on the PCI bus accesses local memory on the HD64404 using memory transfer commands. Allocate at least the capacity set in the local space register 0 (PCILSR0) as PCI bus memory space.

Bits 19 to 0 are fixed in hardware. Of writable bits 31 to 20, those that hold valid values differ according to the value set in PCILSR0.

Table 4.6 Memory Space Base Address Register (BASE0)

PCILSR0 [26:20] Register Value	Required Address Space	BASE0[31:20] Valid Writable Bits
b'000_0000	1 MB	Bits 31 to 20
b'000_0001	2 MB	Bits 31 to 21
b'000_0011	4 MB	Bits 31 to 22
:	:	:
b'011_1111	64 MB	Bits 31 to 26
b'111_1111	128 MB	Bits 31 to 27

The PCICONF5 Register is initialized to H'00000000 at a power-on reset.

Always write to this register before transferring data to and from the PCIC memory from the PCI bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE031	BASE030	BASE029	BASE028	BASE027	BASE026	BASE025	BASE024	BASE023	BASE022	BASE021	BASE020	BASE019	BASE018	BASE017	BASE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE015	BASE014	BASE013	BASE012	BASE011	BASE010	BASE09	BASE08	BASE07	BASE06	BASE05	BASE04	LA0 PREF	LA0 TYPE1	LA0 TYPE0	LA0 ASI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	BASE031	0	R/W	**BASE0 31 to BASE20**
30	BASE030	0	R/W	Base address of local address space 0.
29	BASE029	0	R/W	
28	BASE028	0	R/W	
27	BASE027	0	R/W	
26	BASE026	0	R/W	
25	BASE025	0	R/W	
24	BASE024	0	R/W	
23	BASE023	0	R/W	
22	BASE022	0	R/W	
21	BASE021	0	R/W	
20	BASE020	0	R/W	
19	BASE019	0	R	**BASE0 19 to BASE4**
18	BASE018	0	R	All 0: Fixed in hardware
17	BASE017	0	R	
16	BASE016	0	R	
15	BASE015	0	R	
14	BASE014	0	R	
13	BASE013	0	R	
12	BASE012	0	R	
11	BASE011	0	R	
10	BASE010	0	R	
9	BASE09	0	R	
8	BASE08	0	R	
7	BASE07	0	R	
6	BASE06	0	R	
5	BASE05	0	R	
4	BASE04	0	R	
3	LA0PREF	0	R	**LA0PREF** Shows availability of perfecting of the local address space 0. 0: Prefetch disabled 1: Prefetch enabled (not supported)

Bit	Bit Name	Initial Value	PCI R/W	Description
2	LA0TYPE1	0	R	**LA0TYPE1 and LA0TYPE0**
1	LA0TYPE0	0	R	Shows the memory type of the local address space 0. 00: Base address can be set to 32-bit width, 32-bit space 01: Base address can be set to 32-bit width, less than 1-MB space (not supported) 10: Base address is 64-bit width (not supported) 11: Reserved
0	LA0ASI	0	R	**LA0ASI** Local address space 0 address space indicator. 0: Memory space 1: I/O space (Not supported)

4.6.7 PCI Configuration Register 6 (PCICONF6)

The PCI Configuration Register 6 (PCICONF6) is a 32-bit read/partial-write register that accommodates the Memory Space Base Address PCI Configuration Register stipulated in the PCI local bus specifications. This register contains the most significant bits (maximum 12 in bits 31 to 20) of the address used when a device on the PCI bus accesses local memory on the HD64404 using memory transfer commands. Minimally, allocate the capacity set in the Local Space Register 1 (PCILSR1) to PCI bus memory space.

Bits 19 to 0 are fixed in hardware. The number of valid bits of those that can be written to (bit 31 to 20) differs according to the value set in PCILSR1.

Table 4.7 Memory Space Base Address Register (BASE1)

PCILSR1 [26:20] Register Value	Required Address Space	Valid BASE1 [31:20] Write Bits
b'000_0000	1 MB	Bits 31 to 20
b'000_0001	2 MB	Bits 31 to 21
b'000_0011	4 MB	Bits 31 to 22
:	:	:
b'011_1111	64 MB	Bits 31 to 26
b'111_1111	128 MB	Bits 31 to 27

The PCICONF6 Register is initialized to H'00000000 at a power-on reset.

Always write to this register prior to transferring data to or from the PCIC memory from the PCI bus.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE 131	BASE 130	BASE 129	BASE 128	BASE 127	BASE 126	BASE 125	BASE 124	BASE 123	BASE 122	BASE 121	BASE 120	BASE 119	BASE 118	BASE 117	BASE 116
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BASE 115	BASE 114	BASE 113	BASE 112	BASE 111	BASE 110	BASE 19	BASE 18	BASE 17	BASE 16	BASE 15	BASE 14	LA1 PREF	LA1 TYPE1	LA1 TYPE0	LA1 ASI
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	BASE1 31	0	R/W	**BASE1 31 to 20**
30	BASE1 30	0	R/W	Base address of local address space 1.
29	BASE1 29	0	R/W	
28	BASE1 28	0	R/W	
27	BASE1 27	0	R/W	
26	BASE1 26	0	R/W	
25	BASE1 25	0	R/W	
24	BASE1 24	0	R/W	
23	BASE1 23	0	R/W	
22	BASE1 22	0	R/W	
21	BASE1 21	0	R/W	
20	BASE1 20	0	R/W	

Bit	Bit Name	Initial Value	PCI R/W	Description
19	BASE1 19	0	R	**BASE1 19 to 4**
18	BASE1 18	0	R	All 0: Fixed in hardware
17	BASE1 17	0	R	
16	BASE1 16	0	R	
15	BASE1 15	0	R	
14	BASE1 14	0	R	
13	BASE1 13	0	R	
12	BASE1 12	0	R	
11	BASE1 11	0	R	
10	BASE1 10	0	R	
9	BASE1 9	0	R	
8	BASE1 8	0	R	
7	BASE1 7	0	R	
6	BASE1 6	0	R	
5	BASE1 5	0	R	
4	BASE1 4	0	R	
3	LA1PREF	0	R	**LA1PREF** Shows the availability of local address space 1 prefetches. 0: Prefetch disabled 1: Prefetch enabled (not supported)
2	LA1TYPE1	0	R	**LA1TYPE1 and LA1TYPE0**
1	LA1TYPE0	0	R	This shows the local address space 1 memory type. 00: The base address can be set to 32-bit width, 32-bit space 01: The base address can be set to 32-bit width, but less than 1MB (not supported) 10: The base address has 64-bit width (not supported) 11: Reserved
0	LA1ASI	0	R	**LA1ASI** Local address space 1 address space indicator. 0: Memory space 1: I/O space (Not supported)

4.6.8 PCI Configuration Register 7 (PCICONF7) to PCI Configuration Register 10 (PCICONF10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	—	0	R	**Reserved**

4.6.9 PCI Configuration Register 11 (PCICONF11)

The PCI Configuration Register 11 (PCICONF11) is a 32-bit read/write register that accommodates the Subsystem ID and Subsystem Vendor ID PCI Configuration Registers stipulated in the PCI local bus specifications. The register contains the ID of the add-in board that HD64404 is installed on its subsystem (bits 31 to 16) as well as the subsystem vendor ID (bits 15 to 0).

The PCICONF11 register is not initialized at a reset.

Always initialize this register before PCI transaction is started.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SSID 15	SSID 14	SSID 13	SSID 12	SSID 11	SSID 10	SSID 9	SSID 8	SSID 7	SSID 6	SSID 5	SSID 4	SSID 3	SSID 2	SSID 1	SSID 0
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SVID 15	SVID 14	SVID 13	SVID 12	SVID 11	SVID 10	SVID 9	SVID 8	SVID 7	SVID 6	SVID 5	SVID 4	SVID 3	SVID 2	SVID 1	SVID 0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31	SSID15	—	R/W	**SSID15 to SSID0**
30	SSID14	—	R/W	Specifies the subsystem ID.
29	SSID13	—	R/W	
28	SSID12	—	R/W	
27	SSID11	—	R/W	
26	SSID10	—	R/W	
25	SSID9	—	R/W	
24	SSID8	—	R/W	
23	SSID7	—	R/W	
22	SSID6	—	R/W	
21	SSID5	—	R/W	
20	SSID4	—	R/W	
19	SSID3	—	R/W	
18	SSID2	—	R/W	
17	SSID1	—	R/W	
16	SSID0	—	R/W	
15	SVID15	—	R/W	**SVID15 to SVID0**
14	SVID14	—	R/W	Specifies the PCI subsystem vendor ID.
13	SVID13	—	R/W	
12	SVID12	—	R/W	
11	SVID11	—	R/W	
10	SVID10	—	R/W	
9	SVID9	—	R/W	
8	SVID8	—	R/W	
7	SVID7	—	R/W	
6	SVID6	—	R/W	
5	SVID5	—	R/W	
4	SVID4	—	R/W	
3	SVID3	—	R/W	
2	SVID2	—	R/W	
1	SVID1	—	R/W	
0	SVID0	—	R/W	

4.6.10 PCI Configuration Register 12 (PCICONF12)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	—	0	R	**Reserved**

4.6.11 PCI Configuration Register 13 (PCICONF13)

The PCI Configuration Register 13 (PCICONF13) is a 32-bit read-only register that accommodates the Extended Function Pointer PCI Configuration Register stipulated in the PCI power management specifications. The address offset of the extended function is read from bits 7 to 0.

All bits are fixed in hardware.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

PCI-R/W:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									CAP PTR7	CAP PTR6	CAP PTR5	CAP PTR4	CAP PTR3	CAP PTR2	CAP PTR1	CAP PTR0
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 8	—	0	R	**Reserved**
7	CAPPTR7	0	R	**CAPPTR**
6	CAPPTR6	1	R	These bits specify the address offset of the extended functions (power management). The initial value is H'40 (fixed).
5	CAPPTR5	0	R	
4	CAPPTR4	0	R	
3	CAPPTR3	0	R	
2	CAPPTR2	0	R	
1	CAPPTR1	0	R	
0	CAPPTR0	0	R	

4.6.12 PCI Configuration Register 14 (PCICONF14)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	—	0	R	**Reserved**

4.6.13 PCI Configuration Register 15 (PCICONF15)

The PCI Configuration Register 15 (PCICONF15) is a 32-bit read/partial-write register that accommodates the Maximum Latency, Minimum Grant, Interrupt Pin, and Interrupt Line PCI Configuration Registers stipulated in the PCI local bus specifications. The interrupt pins used by the HD64404 are read from bits 15 to 8. Bits 7 to 0 indicate to which of the interrupt request signal lines of an interrupt controller the interrupt line is connected.

Bits 31 to 8 are fixed in hardware. Bits 7 to 0 can be written to from PCI bus.

The PCICONF15 register is initialized to H'00000100 at a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MLAT7	MLAT6	MLAT5	MLAT4	MLAT3	MLAT2	MLAT1	MLAT0	MGNT7	MGNT6	MGNT5	MGNT4	MGNT3	MGNT2	MGNT1	MGNT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IPIN7	IPIN6	IPIN5	IPIN4	IPIN3	IPIN2	IPIN1	IPIN0	ILIN7	ILIN6	ILIN5	ILIN4	ILIN3	ILIN2	ILIN1	ILIN0
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31	MLAT7	0	R	**Maximum Latency Specification (MLAT7 to MLAT0)**
30	MLAT6	0	R	
29	MLAT5	0	R	Specify whether the PCI device accesses the bus each time. (Not supported)
28	MLAT4	0	R	
27	MLAT3	0	R	
26	MLAT2	0	R	
25	MLAT1	0	R	
24	MLAT0	0	R	
23	MGNT7	0	R	**Minimum Grant Specification (MGNT7 to MGNT0)**
22	MGNT6	0	R	
21	MGNT5	0	R	Specify the burst interval required by the PCI device. (Not supported)
20	MGNT4	0	R	
19	MGNT3	0	R	
18	MGNT2	0	R	
17	MGNT1	0	R	
16	MGNT0	0	R	
15	IPIN7	0	R	**Interrupt Pin Specification (IPIN7 to IPIN0)**
14	IPIN6	0	R	H'01: $\overline{\text{INTA}}$ (interrupt pin fixed)
13	IPIN5	0	R	
12	IPIN4	0	R	
11	IPIN3	0	R	
10	IPIN2	0	R	
9	IPIN1	0	R	
8	IPIN0	1	R	
7	ILIN7	0	R/W	**Interrupt Line Specification (ILIN7to ILIN0)**
6	ILIN6	0	R/W	
5	ILIN5	0	R/W	
4	ILIN4	0	R/W	
3	ILIN3	0	R/W	
2	ILIN2	0	R/W	
1	ILIN1	0	R/W	
0	ILIN0	0	R/W	

4.6.14 PCI Configuration Register 16 (PCICONF16)

The PCI Configuration Register 16 (PCICONF16) is a 32-bit read/partial-write register that accommodates the Power Management Function (PMC), Next-Item Pointer, and Extended Function ID Power Management Registers stipulated in the PCI power management specifications. The power management related functions are read from bits 31 to 16 (PMC), the address offset of the next function in the extended function list is read from bits 15 to 8 (next item pointer), and the power management ID (H'01) is read from bits 7 to 0 (extended function ID).

The PCICONF16 Register is initialized to H'00010001 at a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PME SPT4	PME SPT3	PME SPT2	PME SPT1	PME SPT0	D2SPT	D1SPT				DS1		PME CLK	VER 2	VER 1	VER 0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NIP7	NIP6	NIP5	NIP4	NIP3	NIP2	NIP1	NIP0	CAPID7	CAPID6	CAPID5	CAPID4	CAPID3	CAPID2	CAPID1	CAPID0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	PMESPT4	0	R	**PME Support (PMESPT)**
30	PMESPT3	0	R	Not supported. Defines the function state supporting $\overline{\text{PME}}$ output.
29	PMESPT2	0	R	
28	PMESPT1	0	R	
27	PMESPT0	0	R	
26	D2SPT	0	R	**D2 Support (D2SPT)** Not supported. Specifies whether D2 state is supported.
25	D1SPT	0	R	**D1 Support (D1SPT)** Not supported. Specifies whether D1 state is supported.
24 to 22	—	0	R	**Reserved**

Bit	Bit Name	Initial Value	PCI R/W	Description
21	DS1	0	R	**DSI** Specifies whether bit-device-specific initialization is required.
20	—	0	R	**Reserved**
19	PMECLK	0	R	**PMECLK** Specifies whether a clock is required for PME support.
18	VER2	0	R/W	**VER** Specify the version of power management specifications.
17	VER1	0	R/W	
16	VER0	1	R/W	
15	NIP7	0	R	**NIP** Specify the offset to the next extended function register (Next Item Pointer)
14	NIP6	0	R	
13	NIP5	0	R	
12	NIP4	0	R	
11	NIP3	0	R	
10	NIP2	0	R	
9	NIP1	0	R	
8	NIP0	0	R	
7	CAPID7	0	R	**CAPID** Extended function (Capability Identifier) ID.
6	CAPID6	0	R	
5	CAPID5	0	R	
4	CAPID4	0	R	
3	CAPID3	0	R	
2	CAPID2	0	R	
1	CAPID1	0	R	
0	CAPID0	1	R	

4.6.15 PCI Configuration Register 17 (PCICONF17)

The PCI Configuration Register 17 (PCICONF17) is a 32-bit read/partial-write register that accommodates the Power Management Control/status (PMCSR), Bridge-Compatible PMCSR Extended (PMCSR_BSE), and Data Power Management Registers stipulated in the PCI power management specifications. Bits 31 to 24 (data) and bits 23 to 16 (PMCSR_BSE) are not supported. The power management status is read from bits 15 to 0 (PMCSR).

Bits 1 and 0 can be written to from the PCI bus. Other bits are fixed in hardware.

PCICONF17 is initialized to H'00000000 at a power-on reset.

When B'11 is written to bits 1 and 0 and a transition is made to power state D3 (power down mode), PCIC operation as a master target is disabled, regardless of the setting of bits 2 to 0 of the PCICONF1 (bus master control, memory and I/O space access control) (these bits are masked). When B'00 is written to bits 1 and 0 and a transition is made to power state D0 (normal operating mode), the mask is canceled.

As stipulated in the PCI power management specifications, the register must be initialized after recovery from the power down mode to normal operating mode. However, the values of bits 31 to 27 and 24, as well as the value in the PCIINT Register, are retained even when the transition is made to power down mode. Therefore, on recovering the normal operating mode, first write 1s to these bits to clear them before initializing the register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMEST	DTATSCL1	DTATSCL0	DATASEL3	DATASEL2	DATASEL1	DATASEL0	PMEEN							PWRST1	PWRST0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31	DATA7	0	R	**DATA**
30	DATA6	0	R	Not supported. Data field for power management.
29	DATA5	0	R	
28	DATA4	0	R	
27	DATA3	0	R	
26	DATA2	0	R	
25	DATA1	0	R	
24	DATA0	0	R	
23 to 16	—	0	R	**Reserved**
15	PMEST	0	R	**PME Status (PMEST)** Not supported. Shows the status of the $\overline{\text{PME}}$ bit. This bit is set when the signal is output.
14	DTATSCL1	0	R	**Data Scale (DTATSCL)**
13	DTATSCL0	0	R	Not supported. Scaling value for the value in the data field.
12	DATASEL3	0	R	**Data Select (DATASEL)**
11	DATASEL2	0	R	Not supported. Select the value to be output to the data field.
10	DATASEL1	0	R	
9	DATASEL0	0	R	
8	PMEEN	0	R	**$\overline{\text{PME}}$ Enable (PMEEN)** Not supported.
7 to 2	—	0	R	**Reserved**
1	PWRST1	0	R/W	**PWRST1 and PWRST0**
0	PWRST0	0	R/W	Specifies the power state. No state transition is effected when a non-supported state is specified. (Normal termination, no error output.) 00: D0 state (normal state) 01: D1 state (not supported) 10: D2 state (not supported) 11: D3 state (power down mode)

4.6.16 Reserved Area

Reserved area.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	—	0	R	**Reserved** Note: Reserved area is H'48 to H'FC in PCI Configuration Registers address.

4.6.17 PCI Control Register (PCICR)

The PCI Control Register (PCICR) is a 32-bit register that monitors the status of the mode pin at initialization and controls the basic operation of the PCIC.

The PCICR Register is initialized at a power-on reset to H'00000021.

This register can be written to only when bits 31 to 24 are H'A5.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							TRDSGL						SERR			
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
PCI-R/W:	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 10	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
9	TRDSGL	0	R/W	**Target Read Single Buffer (TRDSGL)** 0: Use 2 target read buffers 1: Use 1 target read buffer only
8 to 6	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
5	—	1	R	**Reserved** This bit always return 1 when read. Always write 1 to this bit when writing.
4	—	0	R	**Reserved** This bit always return 0 when read. Always write 0 to this bit when writing.
3	SERR	0	R/W	**SERR** Not supported. Software control of $\overline{\text{SERR}}$ output. This bit is valid only when the SER bit of the PCICONFI register is 1. This bit always returns 0 when read. 0: $\overline{\text{SERR}}$ is Hiz (using Pull Up resistor) 1: Assert $\overline{\text{SERR}}$(not supported)
2, 1	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
0	—	1	R	**Reserved** This bit always return 1 when read. Always write 1 to this bit when writing.

4.6.18 PCI Local Space Register [26:20] (PCILSR [26:20])

The PCI Local Space Register [26:20] (PCILSR [26:20]) specifies the capacities of the two Local Address Spaces (Address Space 0 and Address Space 1) Registers supported when a device on the PCI bus performs a memory read/memory write of the PCIC using target transfers. This is a 32-bit register that can be read and written from the PCI bus.

The PCILSR [26:20] Register is initialized to H'00000000 at a power-on reset.

Always write to this register before performing target transfers to specify the capacity of the address space being used. Specify the value " (capacity –1) bytes" in bits 26 to 20. For example, to secure a 32-MB space, set the value H'01F00000.

The available values for PLSR26 - PLSR20 are as follows;

H'00, H'01, H'03, H'07, H'0F, H'1F, H'3F, H'7F. Setting other values are prohibited.

If you specify all zeros(PLSR26 - PLSR20 = H'00) , a 1-MB space is reserved. You can specify an address space up to 128MB.

A PCI address can be converted to a local address by using the PCI address for the portion equivalent within the capacity specified in this register and using the value in the PCI Local Address Register for the address above.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						PLSR 26	PLSR 25	PLSR 24	PLSR 23	PLSR 22	PLSR 21	PLSR 20				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 27	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
26	PLSR26	0	R/W	**PLSR26 to 20** Specifies the capacity of address space 0/1 in Mbytes. Specifying (capacity –1) bytes. A 1MB space is secured if all zeros are specified.
25	PLSR25	0	R/W	
24	PLSR24	0	R/W	
23	PLSR23	0	R/W	
22	PLSR22	0	R/W	
21	PLSR21	0	R/W	
20	PLSR20	0	R/W	
19 to 0	—	0	R	0 fixed

4.6.19 PCI Local Address Register [27:20] (PCILAR [27:20])

The PCI Local Address Register [27:20] (PCILAR [27:20]) specifies the starting address (physical address) of the two local address spaces (address space 0 and address space 1) supported when performing memory read/memory write operations due to target transfers to the PCIC. It is a 32-bit register that can be read and written from the PCI bus.

The PCILAR [27:20] Register is initialized to H'00000000 at a power-on reset.

The valid bits of the local address specified by this register vary according to the capacity of the address space specified in the PCILSR [26:20] Register. For example, when the capacity of the local address space is set to 32MB (PCILSR: H'01F00000), bits 27 to 25 of the local address are valid. Only the value set in these bits is used as the physical address of the local address space.

Always write to this register prior to target transfers. Specify the starting address (physical address) of the memory installed on the HD64404 according to the address space being used.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					LAR 27	LAR 26	LAR 25	LAR 24	LAR 23	LAR 22	LAR 21	LAR 20				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 28	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits.
27	LAR27	0	R/W	**LAR27 to LAR20** Specify bits 27 to 20 of the starting address of the local address space.
26	LAR26	0	R/W	
25	LAR25	0	R/W	
24	LAR24	0	R/W	
23	LAR23	0	R/W	
22	LAR22	0	R/W	
21	LAR21	0	R/W	
20	LAR20	0	R/W	
19 to 0	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits.

4.6.20 PCI Interrupt Register (PCIINT)

The PCI Interrupt Register (PCIINT) is a 32-bit register that saves the error source when an error occurs on the PCI bus as a result of the PCIC attempting to invoke a transfer on the PCI bus, or when the PCIC is the PCI master or PCI target. This register can be read from PCI bus. Also, 1 can be written from PCI bus to perform a write-clear in which the detection bit is cleared to its initial value (0).

The PCIINT Register is initialized to H'00000000 at a power-on reset.

When an error occurs, the bit corresponding to the error content is set to 1. Each interrupt detection bit can be cleared to its initial status (0) by writing 1 to it. (Write clear)

Note that the error detection bits can be set even when the interrupt is masked.

The error source holding circuit can only store one error source. For this reason, any second or subsequent error factors are not stored if errors occur consecutively.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		T_TGT_ABORT					TGT_RETRY	MST_DIS	ADRPERR		T_DPERR_WT	T_PERR_DET	M_TGT_ABORT	M_MST_ABORT	M_DPERR_WT	M_DPERR_RD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R / WC1	R	R	R	R	R / WC1	R / WC1	R / WC1	R	R / WC1	R / WC1	R / WC1	R/ WC1	R / WC1	R / WC1

Note: WC1: Cleared by writing 1.

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 15	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits.
14	T_TGT_ABORT	0	R/WC1	**Target Target Abort Interrupt (T_TGT_ABORT) [Target]** When the PCIC is target, an illegal byte enable was detected in I/O transfer.
13 to 10	—	0	R	**Reserved:** These bits always return 0 when read. Always write 0 to these bits.
9	TGT_RETRY	0	R/WC1	**Target Memory Read Retry Timeout Interrupt (TGT_RETRY) [Target]** When the PCIC is target, the master did not attempt a retry within the prescribed number of clocks (2^{15}) (unit: PCI_CLK) (detected only in the case of memory read operations).
8	MST_DIS	0	R/WC1	**Master Function Disable Error Interrupt (MST_DIS) [Master]** Although operation as bus master was limited (when the PCI bus master control bit of the Configuration Register is 0), a master operation (DMA transfer) was performed.

Bit	Bit Name	Initial Value	PCI R/W	Description
7	ADRPERR	0	R/WC1	**Address Parity Error Detection Interrupt (ADRPERR) [Target]** Address parity error detected.
6	—	0	R	**Reserved**
5	T_DPERR_WT	0	R/WC1	**Target Write Data Parity Error Interrupt (T_DPERR_WT) [Target]** When the PCIC is target, a data parity error was detected while receiving a target write transfer (only detected when PCICONFI bit 6 (PER) is 1).
4	T_PERR_DET	0	R/WC1	**Target Read $\overline{\text{PERR}}$ Detection interrupt (T_PERR_DET) (Target)** When the PCIC is target, $\overline{\text{PERR}}$ was detected when receiving a target read transfer.
3	M_TGT_ABORT	0	R/WC1	**Master Target Abort Interrupt (M_TGT_ABORT) [Master]** When the PCIC is master, a target abort ($\overline{\text{DEVSEL}}$ suddenly negated) was detected.
2	M_MST_ABORT	0	R/WC1	**Master Master Abort Interrupt (M_MST_ABORT) [Master]** When the PCIC is master, a master abort ($\overline{\text{DEVSEL}}$ not detected) is detected.
1	M_DPERR_WT	0	R/WC1	**Master Write $\overline{\text{PERR}}$ Detection Interrupt (MDPERR_WT) (Master)** When the PCIC is master, $\overline{\text{PERR}}$ received from the target while writing data to the target.
0	M_DPERR_RD	0	R/WC1	**Master Read Data Parity Error Interrupt (M_DPERR_RD) (Master)** When the PCIC is master, a parity error was detected during a data read from the target.

4.6.21 PCI Interrupt Mask Register (PCIINTM)

The PCI Interrupt Mask Register (PCIINTM) sets the respective interrupt masks for the interrupts generated when errors occur in PCI transfers. It is a 32-bit read/write register that can be accessed from both the PCI bus. When set to 0, the respective interrupt is disabled, and enabled when set to 1.

The PCIINTM Register is initialized to H'00000000 at a power-on reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		T_TGT_ABORT					TGT_RETRY	MST_DIS	ADRPERR		T_DPERR_WT	T_PERR_DET	M_TGT_ABORT	M_MST_ABORT	M_DPERR_WT	M_DPERR_RD
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 16	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits.
15	—	0	R	Always write 0
14	T_TGT_ABORT	0	R/W	**Target Target Abort Interrupt Mask (T_TGT_ABORT) [Target]**
13 to 10	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits.
9	TGT_RETRY	0	R/W	**Target Retry Timeout Interrupt Mask (TGT_RETRY) [Target]**
8	MST_DIS	0	R/W	**Master Function Disable Error Interrupt Mask (MST_DIS) [Master]**
7	ADRPERR	0	R/W	**Address Parity Error Detection Interrupt Mask (ADRPERR) [Target]**

Bit	Bit Name	Initial Value	PCI R/W	Description
6	—	0	R/W	Always write 0
5	T_DPERR_WT	0	R/W	**Target Write Data Parity Error Interrupt Mask (T_DPERR_WT) [Target]**
4	T_PERR_DET	0	R/W	**Target Read $\overline{\text{PERR}}$ Detection Interrupt Mask (T_PERR_DET) [Target]**
3	M_TGT_ABORT	0	R/W	**Master Target Abort Interrupt Mask (M_TGT_ABORT) [Master]**
2	M_MST_ABORT	0	R/W	**Master Master Abort Interrupt Mask (M_MST_ABORT) [Master]**
1	M_DPERR_WT	0	R/W	**Master Write Data Parity Error Interrupt Mask (M_DPERR_WT) [Master]**
0	M_DPERR_RD	0	R/W	**Master Read Data Parity Error Interrupt Mask (M_DPERR_RD) [Master]**

4.6.22 PCI Address Data Register at Error (PCIALR)

The PCI Address Data Register at error (PCIALR) stores the PCI address data (ALOG [31:0]) of errors that occur on the PCI bus. It is a 32-bit register that can be read from PCI bus.

The PCIALR Register is not initialized at a power-on reset. This register holds its valid value only when any of bits in PCIINT is set to 1.

The error source holding circuit can only store one error source. For this reason, any second or subsequent error factors are not stored if errors occur consecutively.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ALOG31	ALOG30	ALOG29	ALOG28	ALOG27	ALOG26	ALOG25	ALOG24	ALOG23	ALOG22	ALOG21	ALOG20	ALOG19	ALOG18	ALOG17	ALOG16
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALOG15	ALOG14	ALOG13	ALOG12	ALOG11	ALOG10	ALOG9	ALOG8	ALOG7	ALOG6	ALOG5	ALOG4	ALOG3	ALOG2	ALOG1	ALOG0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	ALOG31 to ALOG0	—	R	PIC address data (value of A/D line) at time of error.

4.6.23 PCI Command Data Register at Error (PCICLR)

The PCI Command Data Register at error (PCICLR) stores the type of transfer (MSTDMA0, MSTDMA1, MSTDMA2 or TGT) when an error occurs on the PCI bus, and the PCI command (CMDLOG [3:0]). It is a 32-bit register that can be read from PCI bus.

The PCICLR Register is not initialized at a power-on reset. Its initial value is undefined. The relevant bit is set to 1 on detection of an error.

The error source holding circuit can only store one error source. For this reason, any second or subsequent error factors are not stored if errors occur consecutively.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		MSTDMA0	MSTDMA1	MSTDMA2		TGT										
Initial value	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CMDLOG3	CMDLOG2	CMDLOG1	CMDLOG0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	-	-	-	-
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31	—	—	R	**Reserved**
30	MSTDMA0	—	R	**MSTDMA0** Error occurred in DMA channel 0 transfer. (Initial value is undefined.)
29	MSTDMA1	—	R	**MSTDMA1** Error occurred in DMA channel 1 transfer. (Initial value is undefined.)
28	MSTDMA2	—	R	**MSTDMA2** Error occurred in DMA channel 2(RBDMAC) transfer. (Initial value is undefined.)
27	—	—	R	**Reserved**
26	TGT	—	R	**TGT** Error occurred in target read or target write transfer. (Initial value is undefined.)
25 to 4	—	0	R	**Reserved** These bits always return 0 when read.
3	CMDLOG3	—	R	**CMDLOG3 to CMDLOG0** PCI transfer command data at error. (Initial value is undefined.)
2	CMDLOG2	—	R	
1	CMDLOG1	—	R	
0	CMDLOG0	—	R	

4.6.24 PCI DMA Transfer Arbitration Master (PCIDMABT)

The PCI DMA Transfer Arbitration Master (PCIDMABT) is a register that controls the arbitration mode in the case of DMA transfers. Two types of DMA arbitration mode can be selected: priority-fixed and round-robin. This 32-bit read/write register can be accessed from PCI bus.

The PCIDMABT Register is initialized to H'00000000 at a power-on reset.

Always write to this register to specify the DMA arbitration mode prior to starting DMA transfers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DMABT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 1	—	0	R	**Reserved** These bit always returns 0 when read. Always write 0 to these bits when writing.
0	DMABT	0	R/W	**DMABT** Controls the DMA arbitration mode. 0: Priority-fixed(channel 0 > channel 1 > DMAC module) 1: Round-robin

4.6.25 PCI DMA Transfer PCI Address Register 0/1 (PCIDPA0/1)

The DMA Transfer PCI Address Register0/1 (PCIDPA0/1) specifies the starting address at the PCI when performing DMA transfers. This 32-bit read/write register can be accessed from PCI bus.

The PCIDPA Register is not initialized at a power-on reset. The initial value is undefined. When read during a DMA transfer, the next transfer address is returned.

The two least significant bits of the register are ignored, and 32-bit width data transfers are performed.

Always write to this register prior to starting DMA transfers. Always re-set this register before starting a new DMA transfer after a DMA transfer has completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PDPA31	PDPA30	PDPA29	PDPA28	PDPA27	PDPA26	PDPA25	PDPA24	PDPA23	PDPA22	PDPA21	PDPA20	PDPA19	PDPA18	PDPA17	PDPA16
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDPA15	PDPA14	PDPA13	PDPA12	PDPA11	PDPA10	PDPA9	PDPA8	PDPA7	PDPA6	PDPA5	PDPA4	PDPA3	PDPA2	PDPA1	PDPA0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	PDPA31 to PDPA0	—	R/W	Set the PCI starting address for DMA transfer.

4.6.26 PCI DMA Transfer HD64404 Start Address Register 0/1 (PCIDLA0/1)

The DMA Transfer HD64404 Start Address Register 0/1 (PCIDLA0/1) specifies the pixel bus starting address at the HD64404 when performing DMA transfers.

This 32-bit read/write register can be accessed from PCI bus.

The PCIDLA Register is not initialized at a power-on reset. The initial value is undefined. When read during a DMA transfer, the next transfer address is returned.

The two least significant bits of the register are ignored, and 32-bit width data transfers performed. Also, note that the HD64404 starting address set in this register is the physical address.

Always write to this register prior to starting DMA transfers. Always re-set this register before starting a new DMA transfer after a DMA transfer has completed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					PDLA27	PDLA26	PDLA25	PDLA24	PDLA23	PDLA22	PDLA21	PDLA20	PDLA19	PDLA18	PDLA17	PDLA16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDLA15	PDLA14	PDLA13	PDLA12	PDLA11	PDLA10	PDLA9	PDLA8	PDLA7	PDLA6	PDLA5	PDLA4	PDLA3	PDLA2	PDLA1	PDLA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 28	—	0	R	**Reserved**
27 to 0	PDLA27 to PDLA0	0	R/W	**PDLA27 to PDLA0** Set the pixel bus (HD64404) starting address for DMA transfer.

4.6.27 PCI DMA Transfer Counter Register 0/1 (PCIDTC0/1)

The DMA Transfer Counter Register0/1 (PCIDTC0/1) specifies the number of bytes for DMA transfers. This 32-bit read/write register can be accessed from PCI bus. When read during a DMA transfer, it returns the remaining number of bytes in the DMA transfer.

Bits 25 to 0 are used to specify the number of transfer bytes. When set to H'00000000, the maximum 64MB transfer is performed.

Always write to this register prior to starting a DMA transfer. Please re-set this register when starting a new DMA transfer after a DMA transfer completes.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							PTC 25	PTC 24	PTC 23	PTC 22	PTC 21	PTC 20	PTC 19	PTC 18	PTC 17	PTC 16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PTC 15	PTC 14	PTC 13	PTC 12	PTC 11	PTC 10	PTC9	PTC8	PTC7	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 26	—	0	R	**Reserved**
25 to 0	PTC25 to PTC0	0	R/W	**PTC25 to PTC0** Specify the number of bytes in DMA transfer. The maximum number of bytes is 64MB (when set to H'00000000).

4.6.28 PCI DMA Control Register0/1 (PCIDCR0/1)

The DMA Transfer Control Register0/1 (PCIDCR0/1) specifies the operating mode of the respective channels and the method of transfer, etc. This 32-bit read/write register can be accessed from PCI bus.

The PCIDCR Register is initialized to H'00000000 at a power-on reset.

Writing 1 to bit 0 (DMASTRT) starts DMA transfer. Always re-set the value in this register before starting a new DMA transfer after completion of a DMA transfer.

When setting the DMASTOP bit, do not write 1'b1 to the DMASTART bit. Also, write the same setting at the start of transfer to the DMAIM, DMAIS, LAHOLD, PAHOLD, IOSEL and DIR bits.

Example: Starting transfer with PCIDCR = H'00000085
Forced DMA termination PCIDCR = H'00000086

The value in memory is changed if DMA termination is enforced with a value other than the starting value.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						ALN MD10	ALM MD9	DMA ST	DMAI M	DMAI S	LAHO LD	PAHO LD	IOSEL 0	DIR0	DMA STOP	DMA STRT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R/W	R/W	R	R/W	R/ WC1	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 11	—	0	R	**Reserved**
10	ALNMD10	0	R/W	**Alignment Mode (ALNMD)**
9	ALMMD9	0	R/W	Sets data alignment when Pixel bus is big endian. 00: Byte boundary mode 01: W/LW boundary mode (LW data is sent as byte × 4) 10: W/LW boundary mode (LW data is sent as word × 2) 11: W/LW boundary mode (LW data is sent as longword) W: Word, LW:Long Word
8	DMAST	0	R	**DMA Transfer End Status (DMAST)** 0: Normal termination 1: Abnormal termination
7	DMAIM	0	R/W	**DMA Transfer Termination Interrupt Mask (DMAIM)** 0: Interrupt disabled 1: Interrupt enabled
6	DMAIS	0	R/WC1	**DMA Transfer Termination Interrupt Status (DMAIS)** The interrupt status is set even when the interrupt mask is set. • When writing 0: Ignored 1: Status clear • When reading 0: Interrupt not detected 1: Interrupt detected
5	LAHOLD	0	R/W	**LAHOLD** Pixel Bus address control during DMA transfer 0: Incremented 1: High address fixed (Address A[4:0] is incremented)
4	PAHOLD	0	R/W	**PAHOLD** PCI address control during DMA transfer 0: Incremented 1: Fixed

Bit	Bit Name	Initial Value	PCI R/W	Description
3	IOSEL0	0	R/W	**IOSEL** Type of PCI address space during transfer 0: Memory space 1: I/O space
2	DIR0	0	R/W	**DIR** Transfer direction during DMA transfer 0: Transfer from PCI bus to HD64404 (Pixel Bus) 1: Transfer ROM HD64404 (Pixel Bus) to PCI bus
1	DMASTOP	0	R/W	**DMASTOP** Forced termination of DMA transfer • When writing 0: Ignored 1: Forced termination of DMA transfer • When reading Zero read
0	DMASTRT	0	R/W	**DMASTRT** Controls start of DMA transfer to channel 0 • When writing 0: Ignored 1: Start • When reading 0: End of transfer 1: Busy (in transfer)

4.6.29 Reserved

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 0	—	—	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

4.6.30 PCI TRDY Enable Control (PCITRDYENB)

The PCI TRDY Enable Control Register (PCITRDYENB) controls PCI target access.for each PCI memory region 1 and 0.

Setting 1 in LOCAL1/LOCAL0 is to make HD64404 move the WAIT stage at maximum 15 clocks when HD64404 is accessed toward Local Address region 1/Local Address region 0 as a target mode until it can output the data.

Setting 0 in LOCAL1/LOCAL0 is to make HD64404 move the RETRY stage when HD64404 is initially accessed toward a specific address of Local Address region 1/Local Address region 0 as a target mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															LOCAL1	LOCAL0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 2	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
1	LOCAL1	0	R/W	**LOCAL1** Controls the mode of PCI target access to HD64404 for Local Address region 1. 0: RETRY mode for initial access toward a specific address 1: WAIT mode, Maximum wait cycle is 15.
0	LOCAL0	0	R/W	**LOCAL0** Controls the mode of PCI target access to HD64404 for Local Address region 0. 0: RETRY mode for initial access toward a specific address 1: WAIT mode, Maximum wait cycle is 15.

4.6.31 PCI Tile Mode Register (PCITILEMODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															Linear	Tile
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 2	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
1	Linear	0	R/W	**Unconditional Linear Mode (Linear)** 1: All region of Graphic Memory is mapped as a linear addressing space regardless of PCIDTMR, PCILTAD and PCILTAM 0: PCIDTMR, PCILTAD and PCILTAM are available to use
0	Tile	0	R/W	**Unconditional Tile mode (Tile)** 1: All region of Graphic Memory is mapped as a tile addressing space regardless of PCIDTMR, PCILTAD and PCILTAM 0: PCIDTMR, PCILTAD and PCILTAM are available to use

Note: Setting Bit 1, Bit 0 = (1, 1) is prohibited.

4.6.32 PCI Data Transfer Mode Register (PCIDTMR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									PCIMWX1	PCIMWX0						PCIGBM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 8	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
7	PCIMWX1	0	R/W	**Memory width for PCI i/f soft rendering write (PCIMWX)**
6	PCIMWX0	0	R/W	The Memory width for image data in case of PCI i/f soft rendering write (bit 7, bit 6) (0,0): 512 pixels (0,1): 1024 pixels (1,0): 2048 pixels (1,1): 4096 pixels
5 to 1	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
0	PCIGBM	0	R/W	**PCI i/f rendering graphic bit mode (PCIGBM)** 1: 8 bit/pixel 0: 16 bit/pixel

Correspondence between Memory Physical Addresses (bytes) and Rendering Coordinates and Multi-valued Source Coordinates

8 bits/pixel (PCIGBM=1), 512 pixels (PCIMWX = 0) Y(vertical)address = A[26:9], X(horizontal) address = A[8:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:13]														A[8:5]				A[12:9]				A[4:0]				

8 bits/pixel (PCIGBM=1), 1024 pixels (PCIMWX = 1) Y(vertical)address = A[26:10], X(horizontal) address = A[9:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:14]													A[9:5]					A[13:10]				A[4:0]				

8 bits/pixel (PCIGBM=1), 2048 pixels (PCIMWX = 2) Y(vertical)address = A[26:11], X(horizontal) address = A[10:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:15]												A[10:5]						A[14:11]				A[4:0]				

8 bits/pixel (PCIGBM=1), 4096 pixels (PCIMWX = 3) Y(vertical)address = A[26:12], X(horizontal) address = A[11:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:16]											A[11:5]							A[15:12]				A[4:0]				

16 bits/pixel (PCIGBM=0), 512 pixels (PCIMWX = 0) Y(vertical)address = A[26:10], X(horizontal) address = A[9:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:14]													A[9:5]					A[13:10]				A[4:1]				0

16 bits/pixel (PCIGBM=0), 1024 pixels (PCIMWX = 1) Y(vertical)address = A[26:11], X(horizontal) address = A[10:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:15]												A[10:5]						A[14:11]				A[4:1]				0

16 bits/pixel (PCIGBM=0), 2048 pixels (PCIMWX = 2) Y(vertical)address = A[26:12], X(horizontal) address = A[11:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:16]											A[11:5]							A[15:12]				A[4:1]				0

16 bits/pixel (PCIGBM=0), 4096 pixels (PCIMWX = 3) Y(vertical)address = A[26:13], X(horizontal) address = A[12:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:17]										A[12:5]								A[16:13]				A[4:1]				0

Upper line: Memory physical addresses (bytes)
Lower line: Logical coordinates (X,Y)

4.6.33 PCI Linear to Tile Convert Address Register (PCILTAD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LTA D26	LTA D25	LTA D24	LTA D23	LTA D22	LTA D21	LTA D20	LTA D19			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 27	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
26	LTAD26	0	R/W	**The start address of Linear to Tile conversion available:** LTAD26 to 0 (always LTAD18 to 0 is all 0) indicates the local start address of Linear to Tile conversion available. Set the value within either Local region 1 or Local region 0 set by PCILSR0/1 and PCILAR0/1.
25	LTAD25	0	R/W	
24	LTAD24	0	R/W	
23	LTAD23	0	R/W	
22	LTAD22	0	R/W	
21	LTAD21	0	R/W	
20	LTAD20	0	R/W	
19	LTAD19	0	R/W	
18 to 0	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

4.6.34 PCI Linear to Tile Convert Address MASK (PCILTAM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LTAM26	LTAM25	LTAM24	LTAM23	LTAM22	LTAM21	LTAM20	LTAM19			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 27	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
26	LTAM26	0	R/W	**Mask register of PCILTAD (PCILTAM)**
25	LTAM25	0	R/W	PCILTAMn indicates the mask bit of PCILTADn.
24	LTAM24	0	R/W	1: Related PCILTAD bit is valid.
23	LTAM23	0	R/W	0: Related PCILTAD bit is invalid.
22	LTAM22	0	R/W	The available values of LTAM[26:19] are as follows;
21	LTAM21	0	R/W	H'00, H'80, H'C0, H'E0, H'F0, H'F8, H'FC, F'FE, H'FF
20	LTAM20	0	R/W	For example (1)
19	LTAM19	0	R/W	PCILTAD[26:19] = b'11111111 PCILTAM[26:19] = b'11110000 Linear to Tile convert region is where a[26] – a[23] in the local address is b'1111. The allocated space is 8MB Example (2) PCILTAD[26:19] = b'10101010 PCILTAM[26:19] = b'11111000 Linear to Tile convert region is where a[26] – a[22] in the local address is b'10101. The allocated space is 4MB Example (3) PCILTAD[26:19] = b'00000000(default) PCILTAM[26:19] = b'00000000(default) Linear to Tile convert region is where a[26] – a[19] in the local address is don't care. This means the allocated space is 128MB(the whole address area is now Tiled space.) Example (4) PCILTAD[26:19] = b'00000000 PCILTAM[26:19] = b'11111111 Linear to Tile convert region is where a[26] – a[19] in the local address is b'00000000. The allocated space is 512KB

Bit	Bit Name	Initial Value	PCI R/W	Description
18 to 0	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

4.6.35 PCI Peripheral Base Address Register (PCIPAR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					PAR 27	PAR 26	PAR 25	PAR 24	PAR 23	PAR 22	PAR 21	PAR 20	PAR 19	PAR 18	PAR 17	PAR 16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 28	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
27	PAR27	0	R/W	**PCI Peripheral Base Address Register**
26	PAR26	0	R/W	PCIPAR[27:0] (PCIPAR[15:0] is always 0) allocates the local start address of HD64404's Peripheral Modules. Set the values within either the local region 1 or the local region 0. PCIC compares PCIPAR[27:16] with Local Address[27:16] generated in PCIC by PCICONF5/6, PCILAR0/1, PCILSR0/1 and PCI address (See the figure 4.2). If these two register addresses are matched, then PCIC accesses Peripheral addressing space through register bus. Otherwise PCIC accesses Graphic Memory. Please note that DMAC base address (see DMAC specification) is a relative address to PCIPAR.
25	PAR25	0	R/W	
24	PAR24	0	R/W	
23	PAR23	0	R/W	
22	PAR22	0	R/W	
21	PAR21	0	R/W	
20	PAR20	0	R/W	
19	PAR19	0	R/W	
18	PAR18	0	R/W	
17	PAR17	0	R/W	
16	PAR16	0	R/W	

Bit	Bit Name	Initial Value	PCI R/W	Description
				Example: • Local Address Space 0 For Graphic memory, Space 64MB, PCI StartAddress = 32'H0400-0000, Local start address = 32'H0000-0000 • Local Address Space 1 For HD64404 Peripheral Space 64MB (actual use is 64KB), PCI Start Address = 32'H0800-0000 Local start address = 32'H0400-0000 Setting registers: PCICONF5 = 32'H0400-0000, PCICONF6 = 32'h0800-0000 PCILSR0/1 = 32'H03f0-0000, PCILAR0 = 32'h0000-0000, PCILAR1 = 32'H0400-0000 PCIPAR = 32'H0400-0000
15 to 0	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

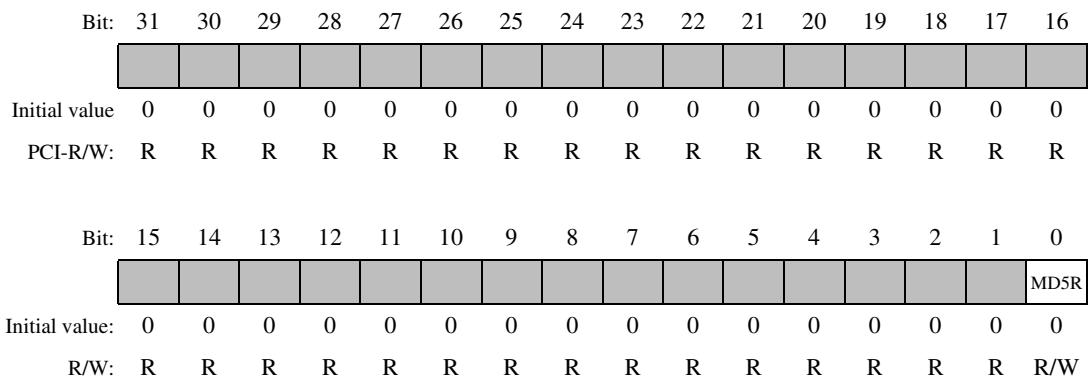
4.6.36 PCI Peripheral Address Space Register (PCIPSR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						PSR 26	PSR 25	PSR 24	PSR 23	PSR 22	PSR 21	PSR 20	PSR 19	PSR 18	PSR 17	PSR 16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 27	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
26	PSR26	0	R/W	**PCI Peripheral Address Space Register** PCIPSR[26:0] (PCIPSR[15:0] is always 0) allocates the local address space of HD64404's Peripheral Modules. Set the values to meet all HD64404 peripheral's addressing space. Default values PCIPSR[26:0] = 0 allocates 64KB space for all HD64404 peripherals except Graphic memory space. Unless the HD64404 peripheral addressing space is not changed from 64KB, it is unnecessary to set this register. Please see DMAC specification.
25	PSR25	0	R/W	
24	PSR24	0	R/W	
23	PSR23	0	R/W	
22	PSR22	0	R/W	
21	PSR21	0	R/W	
20	PSR20	0	R/W	
19	PSR19	0	R/W	
18	PSR18	0	R/W	
17	PSR17	0	R/W	
16	PSR16	0	R/W	
15 to 0	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.

4.6.37 PCI PixelBus Endian Register (PCIMD5R)



Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 1	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
0	MD5R	0	R/W	**MD5R: Pixel Bus Endian Mode** 1: Pixel bus endian conversion enable This affects endian conversion between PCI bus and Pixel bus as follows: HD64404 PCI master transfer: ALNMD bit in PCIDCR0/1 register is now available to use. HD64404 PCI target transfer: This bit sets all data transfer to byte data boundary mode 0: Pixel bus endian conversion disable (default) This bit enables endian conversion between Pixel Bus and PCI Bus. This bit is used when Graphic memory is configured as Big endian while PCI interface is Little. See section 4.8 Endians. Note: It is not recommended that by setting this register, Graphic memory is configured as Big endian because HD64404 does not know the data boundary of LW data from external device so that HD64404 cannot convert endian correctly for the target transfer mode. It is recommended that Graphic memory is used as Little Endian when HD64404 uses PCI interface.

4.6.38 PCI PLL Control Register (PCIPLLCTL)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															RBCLKEN	PIXCLKEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	PCIR/W	Description
31 to 4	—	0	R	**Reserved** These bits always return 0 when read. Always write 0 to these bits when writing.
3, 2	—	1	R	**Reserved** These bits always return 1when read. Always write 1 to these bits when writing.
1	RBCLKEN	0	R/W	**RBCLKEN: Register Bus Clock Enable** 1: Register bus clock is enable 0: Register bus clock is disable This bit controls the input clock of power management block. After writing 1 to this register, HD64404 can supply register bus clock to each module controlled by power management block. Please also refer to Power Control & Configuration block specification.
0	PIXCLKEN	0	R/W	**PIXCLKEN: Pixel Bus Clock Enable** 1: Pixel bus clock is enable 0: Pixel bus clock is disable This bit controls the input clock of power management block. After writing 1 to this register, HD64404 can supply pixel bus clock to each module controlled by power management block. Please also refer to Power Control & Configuration block specification.

4.6.39 PCI TRDY Enable wait cycle counter (PCITRDYCNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PCI-R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TRDY CNT4	TRDY CNT3	TRDY CNT2	TRDY CNT1	TRDY CNT0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	0	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	PCI R/W	Description
31 to 5	—	—	R	**Reserved** Return undefined value when read. Always write 0 to these bits when writing.
4	TRDYCNT4	0	R/W	This register controls the number of wait cycles of $\overline{\text{TRDY}}$. This register is available only when PCITRDYENB[1] or PCITRDYENB[0] is set to 1. TRDYCNT4 to 0 =0 to 5 Those values are interpreted as wait 5 cycles in PCI i/f. TRDYCNT4 to 0 =6 to 31 Those values are interpreted as the actual number of wait cycles. Once PCITRDYENB[1] or PCITRDYENB[0] is set to 1, this register makes HD64404 move the WAIT stage at maximum n PCI CLK cycles that is set in TRDYCNT4 to 0 when HD64404 is accessed toward Local Address region 1/Local Address region 0 as a target mode until it can output the data.
3	TRDYCNT3	1	R/W	
2	TRDYCNT2	1	R/W	
1	TRDYCNT1	1	R/W	
0	TRDYCNT0	1	R/W	

4.7 Functional Description

4.7.1 Operating Modes

PCIC operating mode is the non-host mode and the external input via the PCI_CLK pin is the operating clock for the PCI bus.

4.7.2 PCI Commands

Table 4.8 lists the PCI commands.

Table 4.8 PCI Commands

	I/O State in Operating Modes		
Command	**Master**	**Target**	**Remarks**
Memory read	O	O	
Memory read line	×	Δ	Operates as memory read
Memory read multiple	×	Δ	Operates as memory read
Memory write	O	O	
Memory write invalidate	×	Δ	Operates as memory write
I/O read	O	O	
I/O write	O	O	
Configuration read	—	O	
Configuration write	—	O	
Interrupt acknowledge cycle	×	×	
Special cycle	—	×	
Dual address cycle	×	×	

Legend: O: Support
Δ: Conditional support (see remarks)
×: Not supported

When PCIC Operates as Master: The PCIC supports the memory read command, memory write command, I/O read command, and I/O writes command.

When PCIC Operates as Target: The PCIC receives the memory read command, memory write command, I/O read command, and I/O writes command. The memory read line command and memory read multiple command function as memory reads, while the memory write invalidate command functions as a memory write. The PCIC accepts the configuration command.

4.7.3 PCIC Initialization

The PCIC's internal configuration registers and local registers must be initialized before any PCI transaction. The PCIC can be accessed from the PCI bus.

In particular, the 9 following registers must be initialized: PCI Configuration Registers 1, 2, 11 (PCICONF1, 2, 11), PCI Local Space Register0/1 (PCILSR0/1), PCI Local Address Register 0/1 (PCILAR0/1), PCI Peripheral Base Address Register (PCIPAR), PCI Peripheral Address Space Register (PCIPSR).

4.7.4 Local Register Access

Only longword (32-bit) access of the PCIC's internal local registers and configuration registers from the CPU is supported.

If an attempt is made to access these registers using other than the prescribed access size, zero is returned when reading and writing is ignored. The same is true if you attempt to access the reserved areas in the register area in the PCIC.

When accessing from a PCI device, the PCI bus cycle is caused to wait until the read or write operation has actually completed.

4.7.5 Target Transfers

The following commands are available for transferring data in target transfers.

- Memory read and memory write
- I/O read and I/O write (access to PCIC local registers)
- Fast back-to-back, and back-to-back are supported by PCI target. PCI host does not support them
- Address stepping is supported.

In the case of memory read and memory write commands, both single transfers and burst transfers are supported on the PCI bus. Byte, word, and longword access sizes are supported.

When using I/O read and I/O write commands in relation to the PCIC local registers, only single transfers are supported. Also, only the longword access size is supported.

Only single transfers are supported in the case of configuration read and configuration write operations. Only the longword access size is supported.

If a memory read line command or memory read multiple command is received, they operate as memory reads. Similarly, when a memory write invalidate command is received, it functions as a memory write.

No response is made on reception of special cycle commands.

Memory Read/Memory Write Commands: Data must be set in the following registers prior to performing target transfers using memory read or memory write commands: PCI Configuration Register 5 (PCICNF5), PCI Configuration Register 6 (PCICNF6), PCI Local Space Register 0 (PCILSR0), PCI Local Space Register 1 (PCILSR1), PCI Local Address Register 0 (PCILAR0), and PCI Local Address Register 1 (PCILAR1), PCI Peripheral Base Address Register (PCIPAR), PCI Peripheral Address Space Register (PCIPSR).

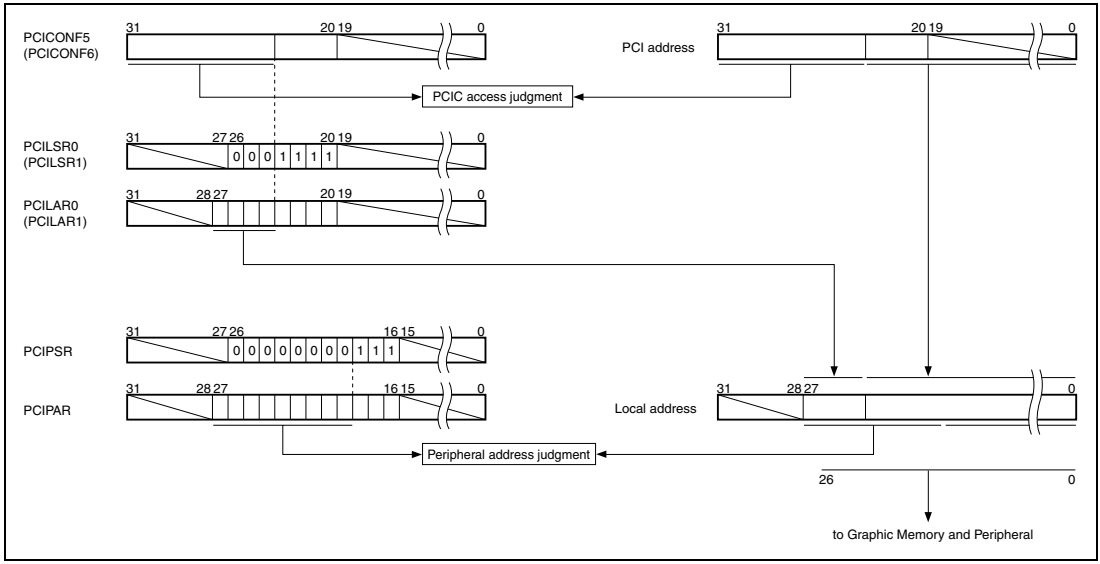


Figure 4.2 Local Address Space

The PCIC supports two local address spaces (address space 0 and address space 1).

The PCI Configuration Register 5, PCI Local Space Register 0, and PCI Local Address Register 0 control the address space 0.

PCI Configuration Register 5 specifies the starting address (logical address) of the PCI bus allocated to address space 0. From the combination of this address and the HD64404 memory capacity, the PCIC determines if the PCI address specified in the PCI command is an address in address space 0 in the PCIC.

The PCI Local Space Register 0 of the PCIC specifies the capacity of the address space 0 and the valid portion of the PCI address. The capacity of address space 0 can be specified between 1MB and 128MB by bits 26 to 20 of the PCI Local Space Register 0. The capacity is specified by setting the capacity (in bytes) of address space 0 -1 in the PCI Local Space Register 0. For example, if the capacity of address space 0 is 16MB (bits 23 to 0 of the PCI address are valid), set bits 23 to 20. For another example, setting bits 26 to 20 to all 0s indicates the capacity to 1MB.

The PCI Local Address Register 0 specifies the starting address (physical address) of memory installed on the HD64404. The address on the HD64404 is determined from this PCI Local Address Register 0 and the valid portion of the PCI address.

In the case of target transfers, the PCI Configuration Register 5 (starting address) and PCI Local Space Register 0 (memory capacity) determine if the PCI address specified in the PCI command is within the PCIC address space 0. If it is an address in address space 0, the PCI address is converted to a physical address on the HD64404 and the data read/write performed in relation to that converted physical address.

If the address specified in the PCI command is not within the PCIC, no response is made to the PCI command.

Address space 1 is, like address space 0, controlled by the PCI Configuration Register 6, PCI Local Space Register 1, and PCI Local Address Register 1.

In this way, it is possible to set two address spaces. In systems with two or less HD64404 areas that can be accessed from the PCI bus, separate address spaces can be allocated to each of them.

To make it possible to access three or more areas from the PCI bus, set the address spaces so that multiple areas are covered. In this case, we can assume that the address space includes areas for which no memory is installed. Note that, in this case, it is not possible to disable target transfers to areas for which no memory is installed.

PCIC Local Register Access: Accessing the local registers is made possible by setting the PCI address in PCI Configuration Register 4 (PCICNF4). Only longword access is supported in the case of local registers.

Data coherency between CPU write data and DMAC DMA data: When CPU writes data to SDRAM through PCI I/F and then DMAC's DMA is initiated and DMA data is read through pixel bus, it can possibly happen that DMAC would read SDRAM data before CPU finishes to write data to SDRAM because a write buffer in PCI IF makes some delay depending on pixel bus round robin arbitration mechanism.

In order to avoid this case, the following procedure has to be taken to initiate DMAC's DMA read from SDRAM.

- After CPU writes the last data or before DMAC's DMA is initiated, dummy read operation is executed. This dummy read guarantees the write buffer in PCI IF is flushed and last data is correctly stored in SDRAM.
- DMAC's pixel bus DMA is initiated by writing DMA_n_Control register.

4.7.6 DMA Transfers between External PCI Device and Graphic Memory through Pixel Bus

DMA transfers allow the high-speed transfer of data between Graphic Memory connected to the HD64404 and PCI bus when the PCIC has bus privileges as master. The following commands are supported in the case of DMA transfers:

- Memory read, memory write, I/O read, and I/O write
- Locked transfers are not supported.

There are two DMA channels for the data transfer between Graphic Memory and PCI bus. A maximum of 64MB can be set for each transfer, the number of transfer bytes and the starting address for the transfer being set at a longword boundary.

Note that locked transfers are not supported in the case of DMA transfers.

Starting DMA Transfer: The following registers exist to control DMA transfers: DMA Transfer Arbitration Register (PCIDMABT) and, for two channels, the DMA Transfer PCI Address Register0/1 (PCIDPA0/1), DMA Transfer Pixel Bus Starting Address Register0/1 (PCIDLA0/1), DMA Transfer Count Register0/1 (PCIDTC0/1), and DMA Control Register0/1 (PCIDCR0/1).

Set the arbitration mode in PCIDMABT prior to starting the DMA transfer. Also select the DMA channel to be used, set the PCI bus starting address and pixel bus starting address in the appropriate PCIDPA and PCIDLA for the selected channel, respectively, set the number of bytes in the transfer in PCIDTC, set the DMA transfer mode in the PCIDCR, and specify a transfer start request.

Because the least significant two bits of these registers are ignored, the transfer is performed in longword units. Also, note that the pixel bus starting address set in PCIDLA is the physical address.

PCIDPA, PCIDLA, and PCIDTC are updated during data transfer. If another DMA transfer is to be performed on completion of one DMA transfer, new values must be set in these registers.

When performing DMA transfers, the address of the pixel bus and the size of data to be transferred can be set to a 32-byte boundary to ensure that data transfers on the pixel bus are as efficient as possible.

PCIDCR can be used to control the abortion of DMA transfers, the direction of DMA transfers, to select PCI commands (memory/I/O) whether to update the PCI address, whether to update the pixel bus address, whether to use transfer termination interrupts, and, when the pixel bus is big endian, the method of alignment.

DMA Transfer End: The following describes the status on termination of a DMA transfer.

- Normal termination

  DMA transfer ends after the set number of bytes has been transferred. In the case of normal termination, the DMA end status bit (DMAST) of the PCIDCR and the DMA transfer start control bit (DMASTART) are cleared, and the DMA transfer termination interrupt status bit (DMAIS) is set.

  If the DMA transfer interrupt mask bit (DMAIM) is set to 1, the DMA transfer termination interrupt is issued.

  Note that the DMAIS bit is set even if the DMAIM bit is set to 0. The DMAIS bit is maintained until it is cleared. Therefore, the DMAIS bit must be cleared before starting the next DMA transfer.

- Abnormal termination

  The DMA transfer may terminate abnormally if an error occurs during data transfer or the DMA transfer is forcibly terminated.

  — Error in data transfer

    When an error occurs during DMA transfer, the DMA transfer is forcibly terminated on the channel in which the error occurred. There is no effect on data transfers on other channels.

  — Forced termination of DMA transfer

    When the PCIDCR and DMASTOP bits for a channel are set, data transfer on that channel is forcibly terminated. However, when the DMASTOP bit is set, do not write 1 to the DMASTRT bit.

  In the case of an abnormal termination, the DMA termination status bit (DMAST) in the PCIDCR is set when the cause of that abnormal termination (error detection or forced termination of DMA transfer) occurs. After the data transfer terminates, the DMA transfer start control bit (DMASTART) is cleared and the DMA transfer termination interrupt status bit (DMAIS) is set.

  If the DMA transfer interrupt mask bit (DMAIM) is set to 1, the DMA transfer termination interrupt is issued.

  In the event of an abnormal termination, the transferred data is not guaranteed.

4.7.7 Arbitration in PCIC

Because target read transfers, target write transfers, and DMA transfers are performed, arbitration is required for these transfers in the PCIC. When multiple data transfer request occur simultaneously in the PCIC, these data transfers are performed in the predetermined order of priority. There are then two choices as to order of priority: fixed or round-robin. The mode is selected using the DMABT bit of the PCI's DMA transfer arbitration register (PCIDMABT).

For arbitration to be performed in such a way as to maintain high-speed data transfer, there are four FIFOs (32-byte × 2 buffer structure) for target reads, target writes, and the two DMA transfer channels. The FIFOs have a 2-buffer structure, enabling one buffer to be accessed from the PCI bus while the other is being accessed from the pixel bus. Depending on the direction of the transfer, the input port of the FIFO for DMA transfers can be connected either to the pixel bus or PCI bus, and the output port either to the PCI bus or the pixel bus.

The arbitration circuit monitors the data transfer requests (data write requests to the FIFO when the FIFO is empty and read requests from the FIFO when it is full) for the five data transfer control circuits (target reads, target writes, 2 DMA transfer channels and RBDMAC DMA transfer channel) to control the data transfers. A maximum of 32 bytes of data is transferred for each data transfer request.

Fixed Priority Mode (DMABT = 0): In fixed priority mode, the order of priority of data transfer requests is fixed and cannot be changed. The order is as follows:

> Target read transfer > Target write transfer > Channel 0 DMA transfer > Channel 1 DMA transfer > RBDMAC DMA transfer

Target read take the highest priority and RBDMAC DMA transfers take the lowest priority. When data transfer requests occur simultaneously, the data transfer with the highest priority takes precedence.

Let's look at data transfers from the pixel bus to the PCI bus in fixed priority mode. The arbitration circuit monitors the transfer requests from the respective data transfer control circuits and writes data read from the pixel bus to the data transfer FIFO that not only is empty but also has the highest priority.

On the other hand, it checks if transfer data exists in the respective FIFOs and reads that data from the data transfer FIFO in which there is data and which has the highest priority, and outputs that data to the PCI bus.

For example, if channel 1 FIFO is empty, the arbitration circuit writes the data from the pixel bus into the channel 1 FIFO. Next, if data of 32 bytes or more is in the channel 1 FIFO, it outputs that data to the PCI bus.

If data has been written to both buffers of the channel 1 FIFO, the channel 1 FIFO is busy while data is output from one of those buffers to the PCI bus. While it is busy, data is written from the pixel bus to the channel 2 FIFO, which has the next highest order of priority. When all data has been output from the channel 1 FIFO to the PCI bus, data is output from the channel 2 FIFO, which still contains data, to the PCI bus.

Thus, in fixed priority mode, execution alternates between the two data transfers with the highest priority.

That is, if DMA transfers are performed simultaneously on 3 channels, the data transfers start with alternation between channels 1 and 2 and then move to alternating between 2 and 3 when all the data in channel 1 has been transferred.

This pattern is the same when data is transferred from the PCI bus to the pixel bus.

Pseudo round-robin mode (DMABT = 1): In pseudo round-robin mode, each time data (byte, word, longword, or 32-byte) is transferred, the order of priority is changed so that the priority level of the completed data transfer is lowest. The order of priority of target reads, target writes, and DMA transfers changes.

When there are no data transfer requests, the initial order of priority in round-robin mode is identical to the fixed-priority mode.

4.7.8 PCI Bus Arbitration

The PCI bus arbitration function in the PCIC is disabled and PCI bus arbitration is performed according to the specifications of the externally connected PCI bus arbiter.

In this case, the PCIC must request PCI bus privileges from the PCI bus arbiter (system host device). The $\overline{\text{REQ}}$ pins are used for the bus request signals, and the $\overline{\text{GNT}}$ pins are used for the bus grant signals. When the bus grant signals are asserted when the bus request signals are not asserted, the PCIC performs bus parking.

Also, when the PCIC is used as a target device that does not request bus privileges, the $\overline{\text{REQ}}$ pins must be fixed at the high level.

4.7.9 PCI Bus Basic Interface

The PCI interface of this LSI supports the PCI version 2.1 stipulations and can be connected directly to a device with a PCI bus interface.

The master performing parking is determined according to the GNT output by the external arbiter. When the master performing parking is not the same master as that starting the subsequent transfer, a high impedance state of at least one clock is generated prior to the address phase.

Master Read/Write Cycle Timing: Figure 4.4 is an example of a burst read cycle. And Figure 4.3 is an example of a burst write cycle. Note that the response speed of DEVSEL and TRDY differs according to the connected target device.

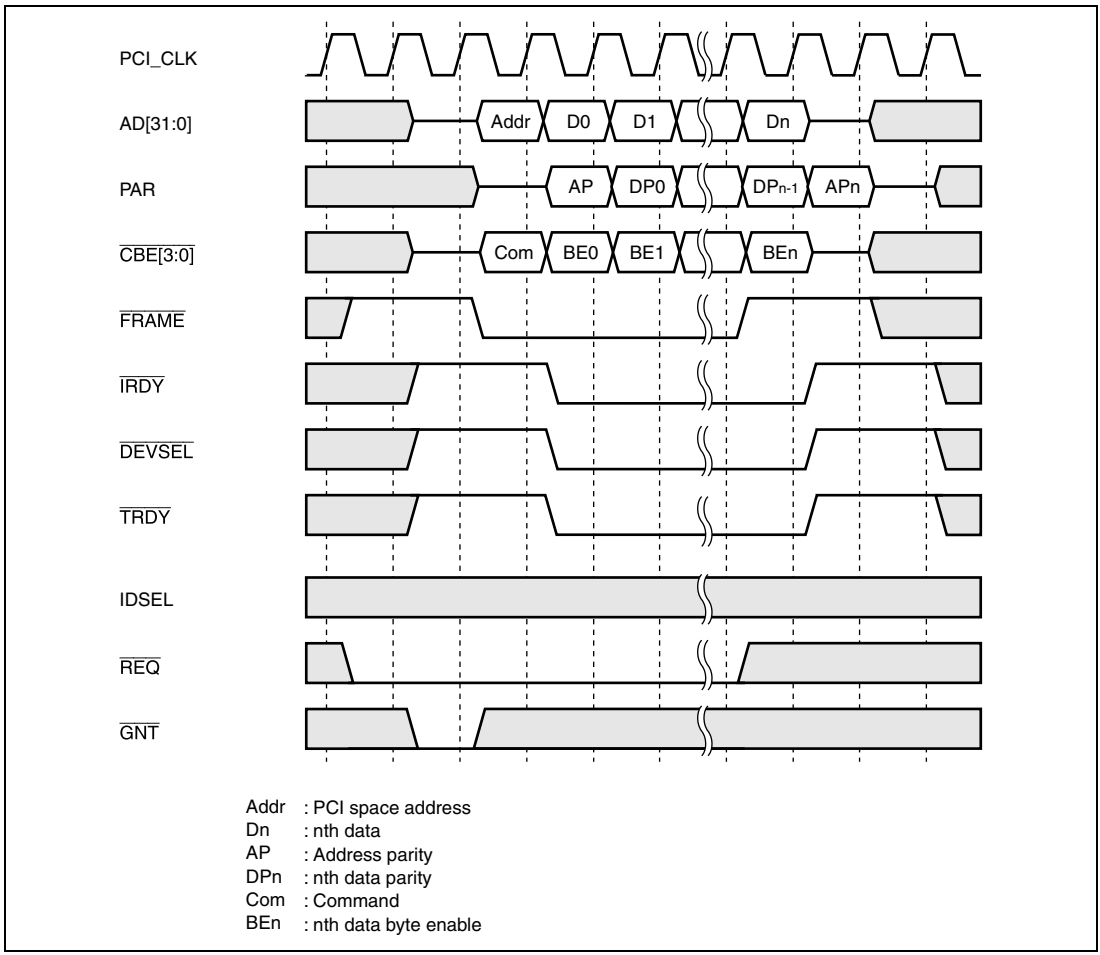


Figure 4.3 Master Memory Write Cycle in Non-Host Mode (Burst)

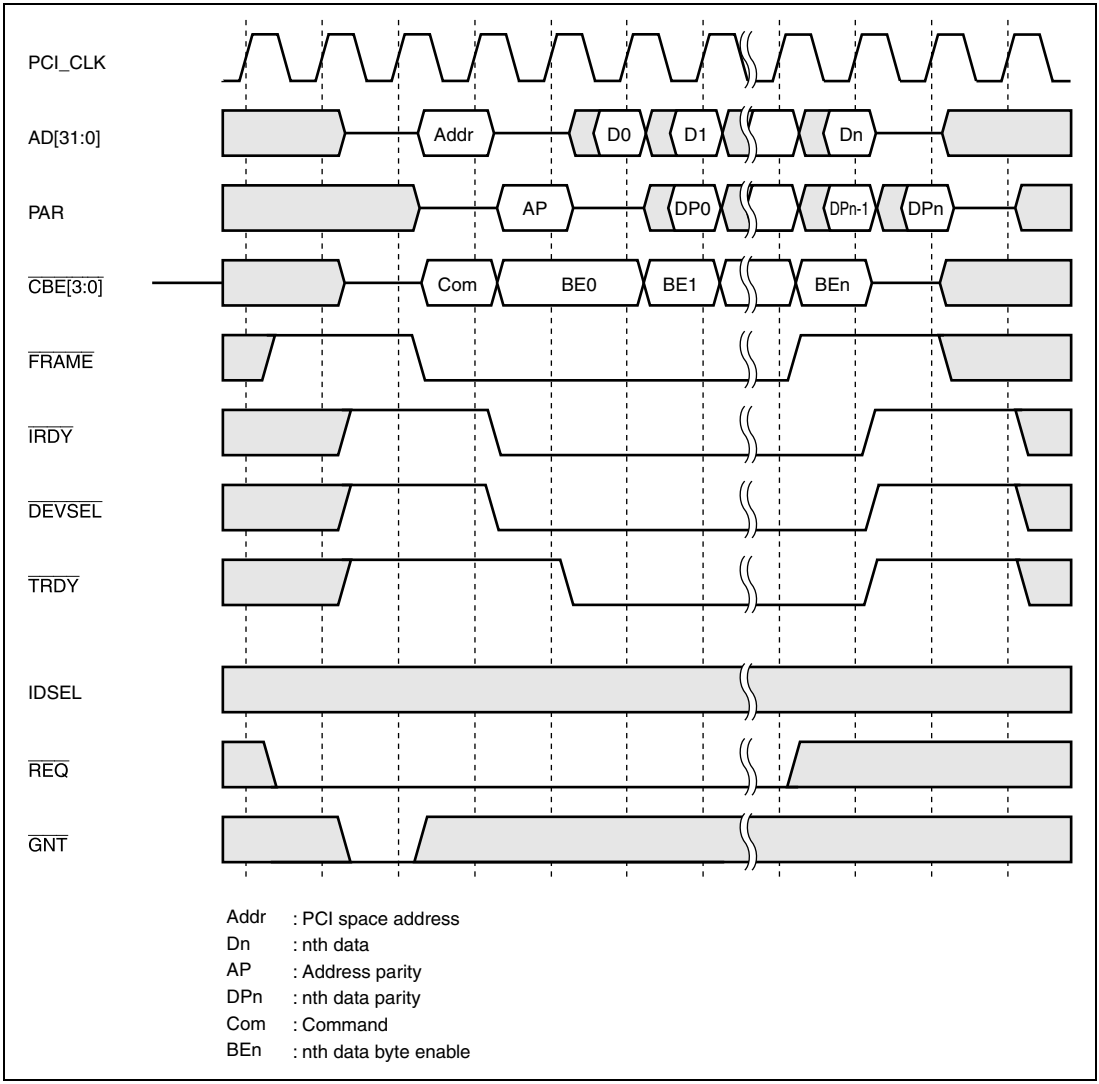


Figure 4.4 Master Memory Read Cycle in Non-Host Mode (Burst)

Target Read/Write Cycle Timing

Retry Mode: When LOCAL1 = 0 in PCITRDYENB Register, The PCIC responds to target memory read accesses from an external master by retires until data are prepared in the PCIC's internal FIFO. That is, it always responds to the first target read with a retry.

When LOCAL1 = 1 in PCITRDYENB register, The PCIC responds to target memory read accesses from an external master by waiting until data are prepared in the PCIC's internal FIFO. The maximum wait cycle is 15 PCI clocks.

Wait Mode: The PCIC responds to target memory read accesses from an external master by wait until the data are prepared in the FIFO.

Also, if a target memory write access is made, the PCIC responds to all subsequent target memory accesses with a retry until the write data is completely written to local memory. Thus, the content of the data is guaranteed when data written to the target is immediately subject to a target read operation.

Only single transfers are supported in the case of target accesses of the configuration space and I/O space. If there is a burst access request, the external master is disconnected on completion of the first transfer.

Note that the DEVSEL response speed is fixed at 2 clocks (Median) in the case of target access of the PCIC.

Figure 4.4 shows an example target single read cycle in non-host mode. Figure 4.5 shows an example target single write cycle in non-host mode.

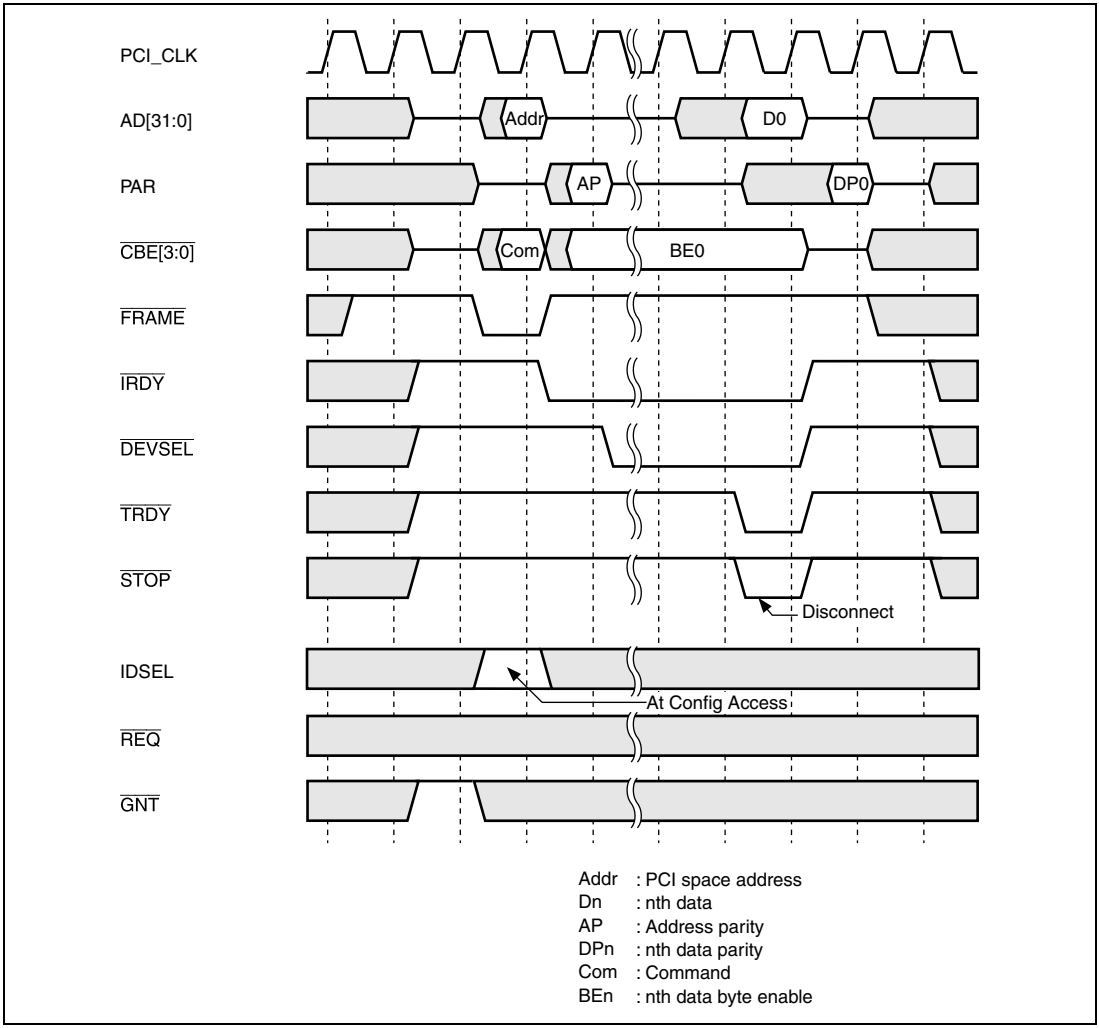


Figure 4.5 Target Read Cycle in Non-Host Mode (Single)

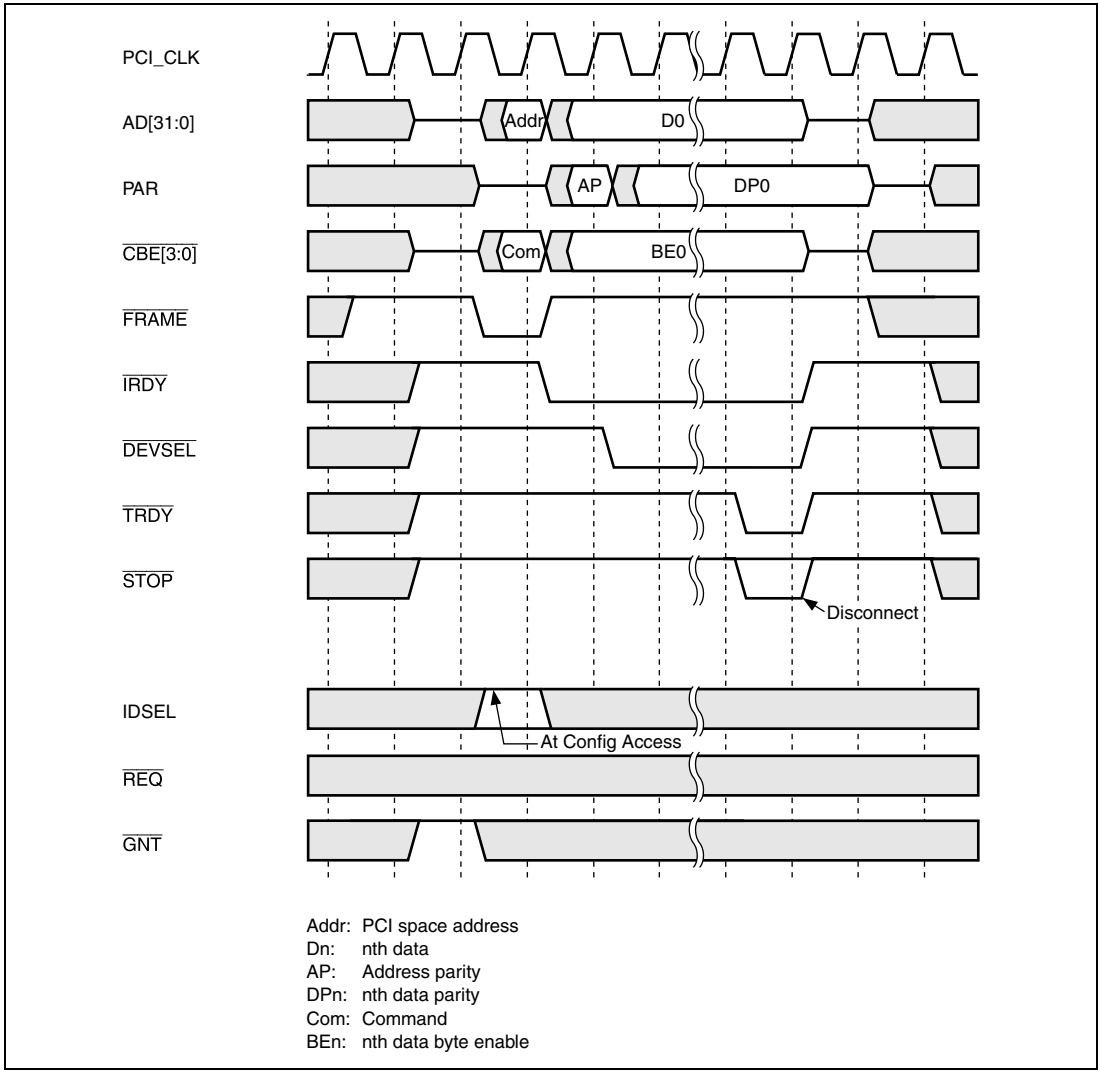


Figure 4.6 Target Write Cycle in Non-Host Mode (Single)

Address/Data Stepping Timing: By writing 1 to the WCC bit (bit 7 of the PCICONF1), a wait (stepping) of one clock can be inserted when the PCIC is driving the AD bus. As a result, the PCIC drives the AD bus over 2 clocks. This function can be used when there is a heavy load on the PCI bus and the AD bus does not achieve the stipulated logic level in one clock.

4.8 Endians

Note: it is not recommended that Graphic memory is configured as Big endian because HD64404 does not know the data boundary of LW data from external device so that PCIC can not convert endian correctly for the target transfer mode.

Though PCIC does have endian conversion described below, it is recommended that Graphic memory is used as Little Endian when HD64404 uses PCI interface.

4.8.1 Endian Control on the pixel bus

When the Pixel bus is used for big endian, big/little endian conversion is therefore required. The PCIC supports four endian conversion modes. These modes are selected by the setting of bits 10 and 9 (ALNMD) of the PCI DMA Control Register (PCIDCR0/1).

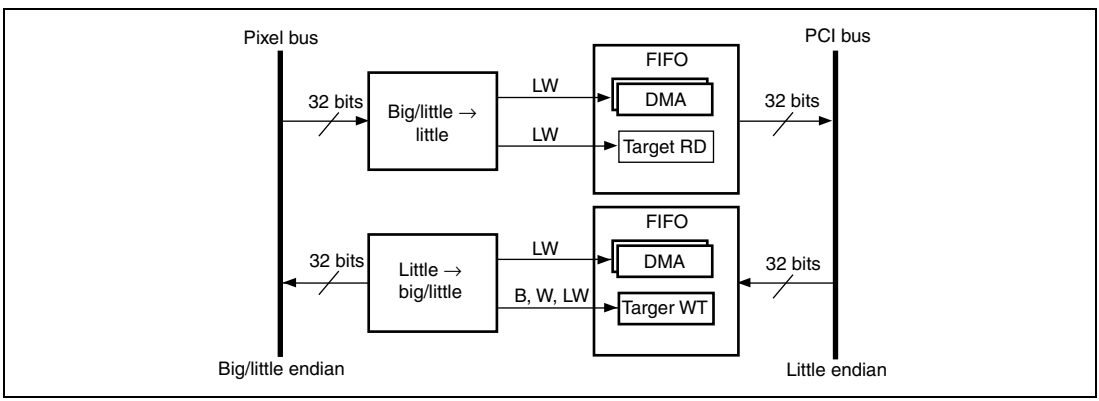


Figure 4.7 Endian Control on the Pixel Bus

4.8.2 Endian Control in DMA Transfers

Though DMA transfer only supports longword access, The following four endian conversion formats can be selected according to 4-bytes longword data, 2-word longword data or 1 longword data.

To change conversion modes PCI DMA Control Register(PCIDCR0/1) bit 10,9(ALNMD) is set.

1. Byte data boundary mode: Big/little endian conversion is performed on the assumption that all data is on a byte boundary. (ALNMD = b'00)
2. Word/longword boundary mode 1: Longword data is transferred as byte data × 4. (ALNMD = b'01)
3. Word/longword boundary mode 2: Longword data is transferred as word data × 2. (ALNMD = b'10)

4. Word/longword boundary mode 3: Longword data is transferred as longword data × 1. (ALNMD = b'11)

Only longword access size is supported in the case of DMA transfers.

Figure 4.8 shows the data alignment in the respective boundary modes in DMA transfers.

DMA transfer when pixel bus set for big endian

Transfer direction	Pixel bus		PCI bus			
	Size		W/LW boundary mode	Byte data boundary mode		$\overline{CBE}$
Pixel bus ↔ PCI bus	LW (W × 4)	B0 B1 B2 B3	B3 B2 B1 B0	B3 B2 B1 B0		0000
	LW (W × 2)	B0 B1 B2 B3	B2 B3 B0 B1	B3 B2 B1 B0		0000
	LW	B0 B1 B2 B3	B0 B1 B2 B3	B3 B2 B1 B0		0000

LW: Long word
BX4: Byte data X4
WX2: Word data X2

DMA transfer when pixel bus set for little endian

Transfer direction	Pixel bus		PCI bus	
	Size			$\overline{CBE}$
Pixel bus ↔ PCI bus	LW	B3 B2 B1 B0	B3 B2 B1 B0	0000

Figure 4.8 Data Alignment in Respective Boundary Modes

4.8.3 Endian Control in Target Transfers

As in DMA transfers, big/little endian conversion is required when the pixel bus is set for big endians in target transfers. Word/longword boundary modes are not supported in the case of target transfers. Set all transfers to byte data boundary mode (ALNMD = b'00).

The access sizes supported in the case of target transfers are as follows: For target reads (the pixel bus to PCI bus), longword only. For target writes (PCI bus to the pixel bus), longword/word/byte. In the case of target writes, data is transferred to the pixel bus inside the PCIC as one or two transfers, depending on the byte enable signal.

Table 4.9 shows the access size and endian conversion modes for transfers between the pixel bus and PCI bus.

Table 4.9 Access Size and Endian Conversion Modes between Pixel bus and PCI bus

				Transfer Mode	
Big/Little	**Access Destination**		**Access Size**	**W/LW Boundary Mode**	**Byte Data Boundary Mode**
Big	Target read		LW	No	Yes
	Target write		B, W, LW	No	Yes
	DMA	Pixel bus → PCI	LW	Yes	Yes
		Pixel bus ← PCI	LW	Yes	Yes
Little	Alignment not required				

4.9 Resetting

Reset Input in Non-Host Mode: The PCIC has no dedicated reset input pin. System reset pin, $\overline{\text{RST}}$ is used.

4.10 Interrupts in PCIC

There are 3 interrupts, as shown in the following table, that can be generated by the PCIC for the CPU.

Error Interrupt: Shows error detection by the PCIC. The error interrupt is asserted when either of the following errors is detected:

- Interrupts detected by PCI Interrupt Register (PCIINT)

The interrupts that can be detected by this register can also be masked. The PCI Interrupt Mask Register (PCIINTM) masks the PCIINT interrupts. See the descriptions of the registers for details.

The following are also set in relation to error interrupts: of the PCI Configuration Register 1 (PCICONF1), the parity error output status (DPE) the system error output status (SSE), the master abort reception status (RMA), the target abort reception status (RTA), the target abort execution status (STA) and the data parity status (DPD).

DMA Channel 0 Transfer Termination Interrupt: The DMA termination interrupt status (DMAIS) bit of the DMA Control Register 0 (PCIDCR0) is set. The interrupt mask is set by the DMA termination interrupt mask (DMAIM) bit of the same register.

DMA Channel 1 Transfer Termination Interrupt: The DMA termination interrupt status (DMAIS) bit of the DMA control Register 1 (PCIDCR1) is set. The interrupt mask is set by the DMA termination interrupt mask (DMAIM) bit of the same register.

Note: see RBDMAC specification for RBDMAC DMA transfer termination interrupt.

$\overline{\text{INTA}}$

The $\overline{\text{INTA}}$ output is used for interrupts to the host device. $\overline{\text{INTA}}$ is open collector output.

$\overline{\text{INTA}}$ is an output from the interrupt signal of Interrupt Priority module.

For normal mode, it is a synchronous output. For standby mode it is an asynchronous output.

4.11 Error Detection

The PCIC can store error information generated on the PCI bus. The address information (ALOG [31:0]) at the time of the error is stored in the PCI Error Address Data Register (PCIALR). The PCI Error Command Information Register (PCICLR) stores the type of transfer (MSTDMA0, MSTDMA1, MSTDMA2, TGT) at the time of the error, and the PCI command (CMDLOG [3:0]).

The error information storage circuit can only store information for one error. Therefore, when errors occur consecutively, no information is stored for the second or subsequent errors.

Error information is cleared by resets.

Notes:

Version Management: The PCIC version management is performed by writing to the PCI configuration register revision ID (8 bits).

Electrical Characteristics: See the section on electrical characteristics for details and before port design.

4.12 References

SH7751 Hardware manual

Section 5 Interrupt Priority Module

5.1 Introduction

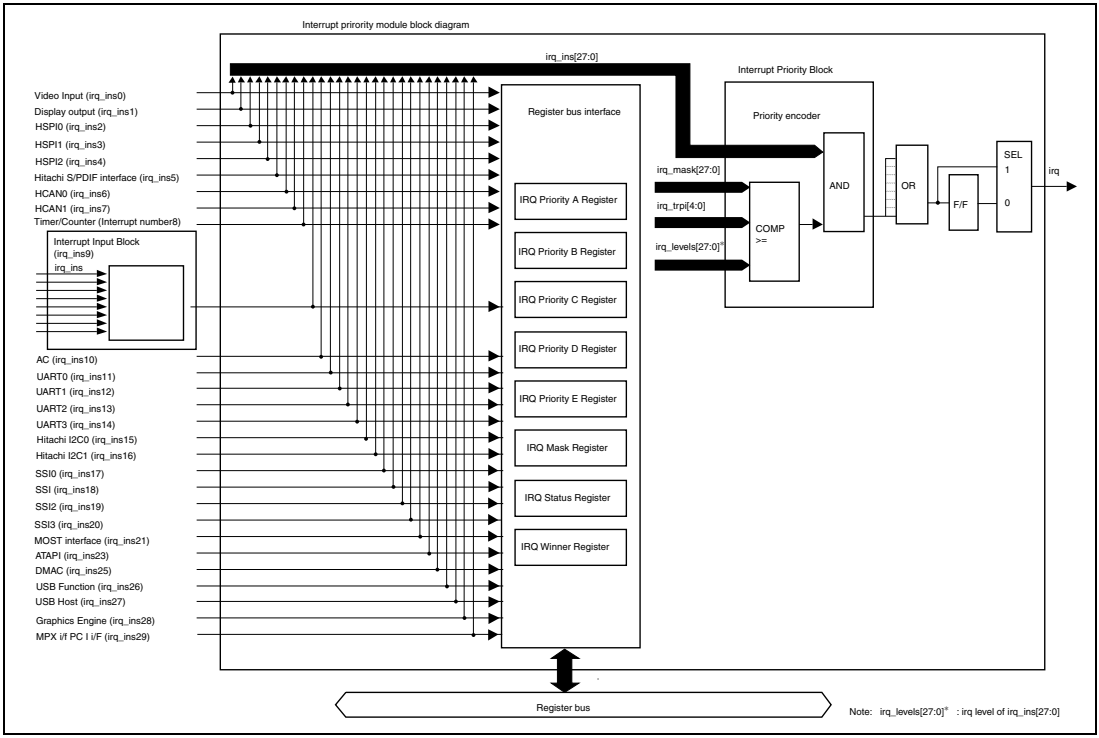
This module is the central interrupt controller and receives interrupts from each of the other blocks within the system in order to prioritise them to the processor. The interrupt priority controller supports up to 28 interrupts. Each interrupt has a programmable priority of value 0 to 31. Bigger the priority value, higher the priority. Each interrupt can be masked. The unmasked interrupt with the highest associated priority is passed on as output to the processor through the system interface and the status recorded. This priority decoding occurs on each clock cycle while the module is not in standby. In standby mode, still priority decoding occurs but is done in non-latched way so that the interrupt pin can be asserted while the input clock stops.

5.2 Features

- All interrupts have fully programmable priority.
- Priority decoding is performed each cycle in normal mode.
- Interrupts can be masked in two ways in order to prevent interrupt pin from being asserted,
  - **Individual mask**: Individual interrupt can be masked
  - **Threshold mask**: Interrupts whose priority are lower than a certain threshold priority can be masked.

  If either individual mask or threshold mask is set for an interrupt, the interrupt will not be reflected to the winning interrupt indicators and not to assert the interrupt pin.
- Interrupt priority module can be set standby mode, in which mode interrupt pin can be asserted even while input clock stops.

5.3 Block Diagram



5.4 Interfaces

5.4.1 Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 5.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From	Synchronization to Clocks
irq	1	Out	Interrupt active	System i/f	rbclk
irq_ins	28	In	Input interrupts	Units	rbclk
Register bus	—	—	System bus	Register busmaster	rbclk

5.4.2 Software Interfaces

The registers accessible by the software are listed in the following table:

Table 5.2 Interrupt Priority Block Register Map

Address (Bytes)	Register name	Mnemonic or Symbol	R/W	Access Size
H'6744	IRQ Priority A	IRQA	R/W	32
H'6748	IRQ Priority B	IRQB	R/W	32
H'674C	IRQ Priority C	IRQC	R/W	32
H'6750	IRQ Priority D	IRQD	R/W	32
H'6754	IRQ Priority E	IRQE	R/W	32
H'6758	IRQ Mask	IRQM	R/W	32
H'675C	IRQ Status	IRQS	R	32
H'6740	IRQ Winner	IRQW	R/W	32

5.5 Register Descriptions

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored.
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

5.5.1 IRQ PriorityA Register (IRQA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			IRQ5					IRQ4					IRQ3			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ3	IRQ2					IRQ1					IRQ0				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	**Reserved**
29 to 25	IRQ5	0	R/W	**Priority for interrupt 5** Priority allocated to the interrupt number 5.
24 to 20	IRQ4	0	R/W	**Priority for interrupt 4** Priority allocated to the interrupt number 4.
19 to 15	IRQ3	0	R/W	**Priority for interrupt 3** Priority allocated to the interrupt number 3.
14 to 10	IRQ2	0	R/W	**Priority for interrupt 2** Priority allocated to the interrupt number 2.
9 to 5	IRQ1	0	R/W	**Priority for interrupt 1** Priority allocated to the interrupt number 1.
4 to 0	IRQ0	0	R/W	**Priority for interrupt 0** Priority allocated to the interrupt number 0.

5.5.2 IRQ PriorityB Register (IRQB)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			IRQ11					IRQ10					IRQ9			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ9	IRQ8					IRQ7					IRQ6				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	**Reserved**
29 to 25	IRQ11	0	R/W	**Priority for interrupt 11** Priority allocated to the interrupt number 11.
24 to 20	IRQ10	0	R/W	**Priority for interrupt 10** Priority allocated to the interrupt number 10.
19 to 15	IRQ9	0	R/W	**Priority for interrupt 9** Priority allocated to the interrupt number 9.
14 to 10	IRQ8	0	R/W	**Priority for interrupt 8** Priority allocated to the interrupt number 8.
9 to 5	IRQ7	0	R/W	**Priority for interrupt 7** Priority allocated to the interrupt number 7.
4 to 0	IRQ6	0	R/W	**Priority for interrupt 6** Priority allocated to the interrupt number 6.

5.5.3 IRQ PriorityC Register (IRQC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			IRQ17					IRQ16					IRQ15			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ15	IRQ14					IRQ13					IRQ12				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	**Reserved**
29 to 25	IRQ17	0	R/W	**Priority for interrupt 17** Priority allocated to the interrupt number 17.
24 to 20	IRQ16	0	R/W	**Priority for interrupt 16** Priority allocated to the interrupt number 16.
19 to 15	IRQ15	0	R/W	**Priority for interrupt 15** Priority allocated to the interrupt number 15.
14 to 10	IRQ14	0	R/W	**Priority for interrupt 14** Priority allocated to the interrupt number 14.
9 to 5	IRQ13	0	R/W	**Priority for interrupt 13** Priority allocated to the interrupt number 13.
4 to 0	IRQ12	0	R/W	**Priority for interrupt 12** Priority allocated to the interrupt number 12.

5.5.4 IRQ PriorityD Register (IRQD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			IRQ23					IRQ22					IRQ21			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ21	IRQ20					IRQ19					IRQ18				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	**Reserved**
29 to 25	IRQ23	0	R/W	**Priority for interrupt 23** Priority allocated to the interrupt number 23.
24 to 20	IRQ22	0	R/W	**Priority for interrupt 22** Priority allocated to the interrupt number 22.
19 to 15	IRQ21	0	R/W	**Priority for interrupt 21** Priority allocated to the interrupt number 21.
14 to 10	IRQ20	0	R/W	**Priority for interrupt 20** Priority allocated to the interrupt number 20.
9 to 5	IRQ19	0	R/W	**Priority for interrupt 19** Priority allocated to the interrupt number 19.
4 to 0	IRQ18	0	R/W	**Priority for interrupt 18** Priority allocated to the interrupt number 18.

5.5.5 IRQ PriorityE Register (IRQE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													IRQ27			
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ27	IRQ26					IRQ25					IRQ24				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	—	R	**Reserved**
19 to 15	IRQ27	0	R/W	**Priority for interrupt 27** Priority allocated to the interrupt number 27.
14 to 10	IRQ26	0	R/W	**Priority for interrupt 26** Priority allocated to the interrupt number 26.
9 to 5	IRQ25	0	R/W	**Priority for interrupt 25** Priority allocated to the interrupt number 25.
4 to 0	IRQ24	0	R/W	**Priority for interrupt 24** Priority allocated to the interrupt number 24.

5.5.6 IRQ Mask Register (IRQM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					IRQ_MASK[27:16]											
Initial:	-	-	-	-	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ_MASK[15:0]															
Initial:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 28	—	—	R	**Reserved**
27 to 0	IRQ_MASK [27:0]	1	R/W	**Interrupt Individual Mask(IIM)** If set, bit n of corresponding mask bit specifies the interrupt of interrupt number n is individually masked and reflected to none of IRQ_WINN, IRQ_WINP and the **irq** pin. Writing IRQ Mask Register does not affect the IRQ Status Register at all. In order to clear/enable the interrupt, Interrupt Control Register in each peripheral module should be manipulated. 0: Interrupt is individually unmasked 1: Interrupt is individually masked Reading IRQ_MASK bit returns which interrupt is currently individually masked. In reset status, all interrupts are individually masked. So individually unmasking appropriate interrupts is software program responsibility.

5.5.7 IRQ STATUS Register (IRQS)

Reset Value: Bit 27-0 reflect interrupt status latched from individual peripherals.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					IRQ_STATUS [27:16]											
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ_ STATUS [15:0]															
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	—	R	**Reserved**
27 to 0	IRQ_STATUS [27:0]	—	R	**Interrupt Status(IS)** Status bit n corresponds to the interrupt of interrupt number n. In Normal Mode, if set, it indicates the corresponding interrupt is active and pending. In Standby Mode, IRQ Status Register holds the last value when it latched in Normal Mode. Manipulating IRQ Mask Register and/or IRQ TPRI field of IRQ Winner Register do not affect IRQ Status Register.

5.5.8 IRQ Winner Register (IRQW)

WINN and WINP indicate the winning interrupt which has the highest priority among all the active and individually unmasked interrupts whose priority equals to or above TRPI. If there is none of them, WINN equals to 31 and WINP is unknown. If there is a winning interrupt, then **irq** pin is asserted.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	STB											IRQ_TPRI				
Initial:	0	-	-	-	-	-	-	-	-	-	-	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				IRQ_WINP								IRQ_WINN				
Initial:	-	-	-	-	-	-	-	-	-	-	-	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	STB	0	R/W	**Standby(STB)** 0: Normal Mode 1: Standby Mode In Standby Mode the interrupt priority module does not latch the priorities in the WINN and WINP fields but it holds the last values when it latched in Normal Mode. It will however still output a signal on the irq pin as specified in general description.
30 to 21	—	—	R	**Reserved**
20 to 16	IRQ_TPRI	0	R/W	**Threshold Mask Priority(TPRI)** Specifies interrupt priority such that all the active interrupts, whose priorities are lower than this threshold mask priority, are threshold masked. If TPRI equals to 0, then all the interrupts are threshold unmasked.
15 to 13	—	—	R	**Reserved**
12 to 8	IRQ_WINP	—	R	**Winning Interrupt Priority(WINP)** Indicates interrupt priority corresponding to WINN. Writing WINP has no effect.
7 to 5	—	—	R	**Reserved**
4 to 0	IRQ_WINN	1	R	**Winning Interrupt Number(WINN)** Indicates interrupt number of active and unmasked interrupt which has the highest priority. Writing WINN has no effect.

5.6 Functional Description

5.6.1 General Functionality

The interrupt priority controller receives thirty interrupts. Each interrupt is identified by interrupt number of 0 through 29. Interrupt number of 31 specifies that there is no active interrupt. Valid interrupt priority value are from 0 through 31. The bigger the priority value, the higher the priority. Interrupts can be masked in two ways, such as individual interrupt can be masked by setting IRQ Mask Register, called individual mask or the interrupts of lower priority than the threshold mask priority as specified in IRQ_TPRI can be masked , called threshold mask. An interrupt is masked if it is either individually masked or threshold masked or both. An interrupt is unmasked if it is neither individually masked nor threshold masked.

In each cycle, if any of the thirty interrupts are set, then the interrupt with the highest active and unmasked priority, which is called the winning interrupt, has its interrupt number and interrupt priority set the IRQ Winner Register. If there is no winning interrupt, then IRQ_WINN equals to 31 and IRQ_WINP is unknown. If there is a wining interrupt, then the irq output line is set. The priority mechanism will continue even after the irq line is set, but the read of the IRQ_WINN will always return wining interrupt number at the point of the read. Individual peripherals will maintain there interrupt once set, until it is cleared by the software.

Standby Mode objective is to support external interrupts which will wake up the host processor while system is in Standby Mode, i.e. power is on but clock is off. Assumption here for software is that after setting Standby Mode, system clock will be set to stop. Setting Standby Mode triggers the interrupt priority module to stop latching the interrupt status at each clock cycle. Still IRQ_WINN,IRQ_WINP and IRQ_TPRI values are held to the last value of Normal Mode. When the external interrupts are asserted again and it is unmasked, then irq pin output is enabled and an interrupt to host processor is asserted. Software interrupt handler should check whether this is standby mode interrupt or not by reading IRQ_STB bit and set Normal Mode by setting IRQ_STB bit to let the interrupt priority module knows that system woke up.

Interrupt Number to Peripheral Modules and Interrupt Status Register

It is not the responsibility of the interrupt priority controller to maintain the interrupts or clearing of interrupts. This is the responsibility of the source units.

The mapping of interrupt bits number to peripheral modules is defined below

Note: That for proper operation, all interrupts must have the unique interrupt priorities.

The Interrupt Status Register of each module listed in Table 5.3 must be read again to guarantee interrupt clear end after clearing the device interrupt. Otherwise, a supurious interrupt is caused, which affects system performance.

Table 5.3 shows Interrupt Status Register names and clear conditions corresponding to each module.

Table 5.3 Interrupt Bits Number to Peripheral Modules

Module	Interrupt Number	Name of Interrupt Status Register	How to clear the Interrupt Status Register	Readable
Video Input	0	Interrupt Status (INTS) bit31, 4, 3, 2, 1, 0: R/WC1	Writing 1 to the bit.	Readable
Display output	1	Display Out Status Register (DO_SR) bit15, 11, 8, 6, 5: R/WC0	writing 1 to the Display Out Status Register Clear Register (DO_SRCR) or Writing 0 to the D0_SR.	Readable
HSPI0	2	Status Register 0 (SR 0) bit 10 to 5, 2 to 0:R bit 4, 3 :R/WC0	bit 10 to 5, 2 to 0: cleared automatically bit 4, 3: Writing 0 to the bit.	Readable
HSPI1	3	Status Register 1 (SR 1) bit 10 to 5, 2 to 0:R bit 4, 3 :R/WC0	bit 10 to 5, 2 to 0: cleared automatically bit 4, 3: Writing 0 to the bit.	Readable
HSPI2	4	Status Register 2 (SR 2) bit 10 to 5, 2 to 0:R bit 4, 3 :R/WC0	bit 10 to 5, 2 to 0: cleared automatically bit 4, 3: Writing 0 to the bit.	Readable
Hitachi S/PDIF interface	5	Status Register (STAT) bit 13 to 6:R/WC0 bit 5 to 0 :R	3 to 6: Writing 0 to the bit. bit 5 :Reading from receiver user information register. bit 4 :Writing to transmitter user register. bit 3 :Reading from receiver channel status registers. bit 2 :Reading from receiver audio channel registers. bit 1 : Writing to transmitter channel status registers. bit 0 : Writing to transmitter audio channel registers.	Readable
HCAN0	6	Interrupt Request Register 0 (IRR 0) (16-bit R/WC1 register)	Writing 1 to the IRR 0.	Readable
HCAN1	7	Interrupt Request Register 1 (IRR 1) (16-bit R/WC1 register)	Writing 1 to the IRR 1.	Readable

Module	Interrupt Number	Name of Interrupt Status Register	How to clear the Interrupt Status Register	Readable
Timer/Counter	8	IRQ status Register bit11 to 0:R/WC0	Writing 0 to the IRQ status Register.	Readable
Interrupt input	9	IRQ status Register R/WC0	Writing 0 to the IRQ status Register	Readable
Audio Codec	10	- TX Status Register (TSR) bit31 to 28, 9, 8: R/WC0 - RX Status Register (RSR) bit22 to 19, 13, 12: R/WC0	-Writing 0 to the TSR. - Writing 0 to the RSR.	Readable
UART0	11	Serial Status Register 0 (SSR0) bit 7 to 3:R/WC0 bit 2: R	These flags can be cleared to 0 only if they have first been read while set to 1.	Readable
UART1	12	Serial Status Register 1 (SSR1) bit 7 to 3: R/WC0 bit 2: R	These flags can be cleared to 0 only if they have first been read while set to 1.	Readable
UART2	13	Serial Status Register 2 (SSR2) bit 7 to 3: R/WC0 bit 2: R	These flags can be cleared to 0 only if they have first been read while set to 1.	Readable
UART3	14	Serial Status Register 3 (SSR3) bit 7 to 3: R/WC0 bit 2: R	These flags can be cleared to 0 only if they have first been read while set to 1.	Readable
Hitachi I2C0	15	- Master Status Register 0 (R/WC0) - Slave Status Register 0 bit 4 to 0: R/WC0	-Writing 0 to the Master Status Register. - Writing 0 to the Slave Status Register.	Readable
Hitachi I2C1	16	- Master Status Register1 (R/WC0) Slave Status Register1 bit 4 to 0: R/WC0	-Writing 0 to the Master Status Register. -Writing 0 to the Slave Status Register.	Readable
SSI0	17	Status Register 0 Bit 27 (UIRQ): R/WC0 Bit 26 (OIRQ): R/WC0 Bit 25 IIRQ: R Bit 24 (DIRQ): R	Bit 27: Writing 0 to the bit. Bit 26: Writing 0 to the bit. Bit 25: cannot be cleared by writing to the bit. Bit 24: cannot be cleared by writing to the bit.	Readable

Module	Interrupt Number	Name of Interrupt Status Register	How to clear the Interrupt Status Register	Readable
SSI1	18	Status Register 1 Bit 27 (UIRQ): R/WC0 Bit 26 (OIRQ): R/WC0 Bit 25 IIRQ: R Bit 24 (DIRQ): R	Bit 27: Writing 0 to the bit. Bit 26: Writing 0 to the bit. Bit 25: cannot be cleared by writing to the bit. Bit 24: cannot be cleared by writing to the bit.	Readable
SSI2	19	Status Register 2 Bit 27 (UIRQ): R/WC0 Bit 26 (OIRQ): R/WC0 Bit 25 IIRQ: R Bit 24 (DIRQ): R	Bit 27: Writing 0 to the bit. Bit 26: Writing 0 to the bit. Bit 25: cannot be cleared by writing to the bit. Bit 24: cannot be cleared by writing to the bit.	Readable
SSI3	20	Status Register 3 Bit 27 (UIRQ): R/WC0 Bit 26 (OIRQ): R/WC0 Bit 25 IIRQ: R Bit 24 (DIRQ): R	Bit 27: Writing 0 to the bit. Bit 26: Writing 0 to the bit. Bit 25: cannot be cleared by writing to the bit. Bit 24: cannot be cleared by writing to the bit.	Readable
MOST interface	21	MIM Interrupt Status Register R/WC0	-Writing 0 to the MIM Interrupt Status Register.	Readable
ATAPI	22	ATAPI status Register bit 8, 7, 5, 3, 2, 1: R/WC0 bit 4, 0: R	bit 8, 7, 5, 3, 2, 1: Writing 0 to the bits. bit 4:Since this register doesn't hold its status in HD64404 chip, if AT_DIRQ 1 becomes 0, this register will also become 0. Read the Status Register in the ATAPI device to clear this bit 4 Additionally, the Alternate Status Register in ATAPI device must be read to guarantee end. Otherwise, a spurious interrupt is caused, which affects system performance. bit 0: This bit is automatically cleared when DMA is completed. So this bit should not be used as an interrupt source.	Readable.

Module	Interrupt Number	Name of Interrupt Status Register	How to clear the Interrupt Status Register	Readable
DMAC	23	-DMA Status Register bit31 to 0:R/WC0 -DMA FIFO Status Register bit31 to 0:R/WC0	Writing 0 to the Register	Readable
USB Function	24	-Interrupt Flag Register 0 (USBIFR0) bit7, 5, 3 to 0: R/WC0 bit6, 4: R -Interrupt Flag Register 1 (USBIFR1) bit3 : R bit2 to 0: R/WC0	-Interrupt Flag Register 0 bit7, 5, 3 to 0:Writing 0 to the bit. bit6, 4 : cannot be cleared by writing to the bit. -Interrupt Flag Register 1 bit3 : This bit has the same value of USB2OVC pin bit2 to 0 : Writing 0 to the bit.	Readable
USB Host	25	HcInterruptStatus bit30, 6 to 0: R/WC1	Writing 1 to the Register	Readable
Graphics Engine	26	Status Register (SR) bit 2 to 0: Read only	Writing 1 to Status Register Clear Register (SRCR) bit 2 to 0: Writing 1 to the bit.	Readable
MPX i/f, PCI i/f	27	MPX IF -CPU Status Register (SR) bit0:Read only PCI IF -PCI Interrupt Register (PCIINT) bit 14,9 to 7,5 to 0 : R/WC1 -PCI DMA Control Register0/1 (PCIDCR0/1) bit 6 : R/WC1	MPX IF -Writing 1 to CPU Status Register clear register (SRCR) PCI IF -Writing 1 to PCI Interrupt Register -Writing 1 to bit 6 in PCI DMA Control Register	Readable

5.6.2 Reset Strategy

All registers will be equipped with a synchronised asynchronous reset.

5.6.3 Power Saving and Clocking Strategy

The interrupt priority controller will operate synchronous to the register bus clock.

Recommended handling procedure for spurious interrupts is as follows,

5.6.4 Spurious Interrupt Handling

For use for having robust interrupt handling design in order to handle intermittent spurious interrupt correctly, here is a sample procedure of interrupt handler including Standby Mode and spurious interrupt handling as follows,

5.6.5 Sample Interrupt Handler Pseudo Procedure

```
// A sample interrupt handling procedure with standby mode
// and spurious interrupt handling

//HD64404 Interrupt handler
#define  STB  0x80000000
#define  WINN 0x0000001f
INT32 winner
int   winnerCode
{
     winner = Read IRQ_WINNER          // read winner register
     if winner & STB {                 // standby mode check
         IRQ_WINNER = (winner & ~STB)  // switch stand-by to normal
         winner = Read IRQ_WINNER      // read winner register again to
                                       // get the real interrupt number
     }
     winnerCode = WINN & winner        // get interrupt number
vector jump to HD64404 interrupt routine using winnerCode as the index
}
```

```
//Interrupt Input Module interrupt routine  winnerCode=9
#define STBY 0x01000000
#define ST   0x000000ff
INT32  control
int    status
{
    control = IRQ_CONTROL              // read control register
    if control & STBY {                // standby mode check
       IRQ_CONTROL = (control & ~STBY) // switch stand-by to normal
    }
    status = IRQ_STATUS & ST           // get the interrupt status
    // interrupt handling continues here
}

//Interrupt Routine for Error handling  winnerCode=31
{
     mask all the interrupt         // IRQ_MASK = 0xFFFFFFFF
                                    // This is necessary to de-assert
                                    // the irq-pin
     increment the spurious int count
     if spurious int count reaches LIMIT {
     // there is serious and permanent malfunctioning
     else
     // assuming the phenomenon is intermittent
     }
     // If spurious interrupt is intermittent, restoring int mask
     // does not cause another spurious interrupt again
     restore the interrupt mask
}

// Interrupt routine other than winnerCode=9 or 31
{
// normal interrupt handling procedure
}
```

Section 6 Memory Interface

6.1 General Description

The memory controller supports overlapping SDRAM command access and multi-bank activation for reduced pre-charge and activation delays. It supports up to two SDRAM devices and from 8MB to 128MB memory capadity.

6.2 Features

- Programmable memory size configurations.
- Programmable SDRAM timing.
- Supports overlapping SDRAM command access and multi-bank activation.
- Power save mode.
- Supports up to two SDRAM devices
- Supports from 8MB to 128MB memory capacity.

6.2.1 Digital Inputs/Outputs

Table 6.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function
SD_CLK	1	OUT	Drives the external SDRAM clock via a bi-directional pin.
SD_DATA(31:0)	32	INOUT	Data bus from the SDRAM
$\overline{\text{BA(1:0)}}$	2	OUT	Bank address bits to the SDRAM
SD_ADDR(12:0)	13	OUT	Address bits to the SDRAM
$\overline{\text{CS}}$	1	OUT	SDRAM chip select
$\overline{\text{RAS}}$	1	OUT	SDRAM RAS signal
$\overline{\text{CAS}}$	1	OUT	SDRAM CAS signal
$\overline{\text{WE}}$	1	OUT	SDRAM write enable signal
DQM(3:0)	4	OUT	SDRAM byte write/OE signal
SD_CKE	1	OUT	SDRAM clock enable

6.2.2 Software Interfaces

The registers accessible by the software are listed in the following table:

Table 6.2 Register List

Address (Bytes)	Register name	Mnemonic or Symbol	R/W	Access Size
H'6280	Memory Control Register	MCR	R/W	32

6.2.3 Functional Description

This block allows the direct connection to SDRAM's and can accept commands that are either sequential or random addressed. When it is a sequential command only the start address is required and the memory controller will perform a full-page burst with a burst stop. When it is a random access command the address is created by the source coincident with the data.

The memory controller will accommodate different size SDRAM's through the programming of the MCR register. As part of this function it will automatically control the refreshes, page misses and initialization.

6.3 Register Descriptions

Legends for register description:

Initial value : Register value after reset
— : Undefined value
R/W : Read and Write, write value can be read.
R : Read only, for write always 0 write
R/WC0 : Read and Write, 0 write clear, 1 write is ignored
R/WC1 : Read and Write, 1 write clear, 0 write is ignored
W : Write only, Read prohibited. If reserved, write always 0.
—/W : Write only, Read value undefined.

6.3.1 Memory Control Register (MCR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ME	SR				RP										TA
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TA		TC			CL				PC		EN	RW		CW	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	ME	0	R/W	**Memory Interface Enable (ME)** 0: Memory interface is disabled. 1: Memory interface is enabled.
30	SR	0	R/W	**Self Refresh Enable (SR)** 0: Normal operation 1: Puts the memory into self refresh mode.
29 to 27	—	0	R	**Reserved Bits** Always write 0, undefined value for read.
26 to 17	RP	0	R/W	**Refresh Period (RP)** Number of memory clock cycles × 16 per refresh.
16 to 14	TA	0	R/W	**Tras Setting (TA)** Timing value Tras in clock cycles.
13 to 11	TC	0	R/W	**Trcd Setting (TC)** Timing value Trcd in clock cycles.
10, 9	CL	0	R/W	**CAS Latency (CL)** 00: Reserved 01: Reserved 10: Two Cycles 11: Three Cycles
8	—	0	R	**Reserved Bit**
7	—	0	R	**Reserved bit** Always write 0, undefined value for read.

Bit	Bit Name	Initial Value	R/W	Description
6	PC	0	R/W	**Precharge Control Bit (PC)** 0: Bit 10 of output address. 1: Bit 11 of output address.
5	—	0	R	**Reserved Bit** Always write 0, undefined value for read
4	EN	0	R/W	**Endian Mode** 0: Little Endian 1: Big Endian The memory interface does not perform any endian conversion this bit, is output to modules attached to the pixel bus, to indicate the endian they should use in accessing SDRAM.
3, 2	RW	0	R/W	**Row Address Width (RW)** 00: 11-bit wide 01: 12-bit wide 10: 13-bit wide 11: Reserved.
1, 0	CW	0	R/W	**Column Address Width (CW)** 00: 8-bit wide 01: 9-bit wide 10: 10-bit wide 11: Reserved

6.4 Power saving

There are two methods of power saving with this interface.

Memory disabled: By clearing the ME bit the memory interface and the SDRAM's will be disabled. No refresh will occur and the contents of the memory will be lost.

Self refresh: The SDRAM supports a mode where they will refresh themselves without any intervention from the memory controller. This saves power in the state machines of the memory controller but also in the pins connected to the SDRAM.

Self refresh can be entered by setting the SR bit. This mode will then be entered as soon as the memory interface becomes idle.

6.4.1 Power-On sequence

The memory controller utilizes the following power up sequence:

After reset the memory controller module outputs NOP commands to the SDRAM and the DQM ports are held high. A delay as specified in the datasheet of the connected SDRAM should be adhered to before the memory controller is enabled.

Once the memory controller has been enabled it will enter its power up sequence firstly a PRECHARGE all banks command is issued followed by 8 AUTO REFRESH CYCLES. Then the mode register of the SDRAM is loaded using the information set in the memory controller MCR register, after the LOAD MODE REGISTER command is issued the memory controller enters an idle state waiting for a read/write command to be issued from the DMAC controller.

6.4.2 Memory interface Power Down Sequence with Self Refresh

1. Wait until all transactions on the pixel bus from pixel bus modules have completed.*
2. Execute Self refresh (ME = SR = 1).

Refer to table 6.3 for the ME and SR bit settings.

Note: * Before executing the Self Refresh mode, it is important that all the modules on the pixel bus are disabled as a general requirement of the memory interface power down sequence.

6.4.3 Module Standby Mode

The Memory Interface module (MEM) allows clock gating to reduce power consumption. The module standby mode can be executed by controlling bit 6 in the Clock Control 2(CC2) Register in the Power Control module.

To wake up the module, MEM bit 6 in the Clock Control 2(CC2) Register must be enabled. After enabling this bit all access to the memory interface module can be possible.

To power down the module using the power control module, the following procedure is required:

1. Wait until all transactions on the pixel bus from pixel bus modules have completed.*
2. Disable memory interface (ME=0).
3. Disable MEM bit 6 in the Clock Control 2(CC2) Register.

Refer to table 6.3 for the ME bit settings.

Note: * Before disabling MEM bit 6 in the Clock Control 2(CC2) Register (Power Control Module), it is important that all the modules on the pixel bus are disabled as a general requirement of the memory interface power down sequence.

The table below shows the memory controller state transitions that depend on the ME and SR bit settings.

Table 6.3 Memory Interface State Transition about ME and SR bit

Before		After		
ME	**SR**	**ME**	**SR**	
0	0	0	0	No change
0	0	0	1	No change
0	0	1	0	Power-On Sequence
0	0	1	1	Power-On Sequence and Self refresh after that
0	1	0	0	no change
0	1	0	1	no change
0	1	1	0	Power-On Sequence
0	1	1	1	Power-On Sequence and Self refresh after that
1	0	0	0	Disable Memory interface
1	0	0	1	Disable Memory interface
1	0	1	0	no change
1	0	1	1	Execute Self refresh
1	1	0	0	Change SD_CKE pin from Low to High
1	1	0	1	Change SD_CKE pin from Low to High
1	1	1	0	Recover from Self refresh to Normal mode
1	1	1	1	no change

6.5 SDRAM Mode Register setting

Memory controller set the SDRAM Mode Register below.

- OPCODE: 0
- A7: 0
- CAS Latency: 2 or 3
- Burst Type: 0 Sequential
- Burst length: 111 Full Page

Table 6.4 SDRAM Mode Register Setting

HD64404 Pin	Memory Interface Setting	Meanings
$\overline{BA1}$	0	OPCODE
$\overline{BA0}$	0	
SD_ADDR 12	0	
SD_ADDR 11	0	
SD_ADDR 10	0	
SD_ADDR 9	0	
SD_ADDR 8	0	
SD_ADDR 7	0	
SD_ADDR 6	0	CAS Latency 2 or 3
SD_ADDR 5	1	
SD_ADDR 4	0 or 1	
SD_ADDR 3	0	Burst Type: Sequential
SD_ADDR 2	1	Burst Length: Full Page
SD_ADDR 1	1	
SD_ADDR 0	1	

6.6 SDRAM configuration for UM (unified memory)

Conditions are described below that can be used as UM. For electrical characteristics, refer to section 31 Electrical specification.

- Memory capacity is 128 Mbytes or less.
- 16-bit memory × 2 or 32-bit memory × 1
- Up to two memories can be connected.
- Column address width: 8, 9, or 10
- 4-bank configuration
- Burst write and burst read modes are supported.
- Burst length: Full pages supported (Burst type: Sequential)
- SDRAM commands (listed below) are supported that are used in HD64404
- Memory is required to operate at between 83 MHz and 100 MHz.

SDRAM Commands that Are Used in HD64404

- Ignore command (DESL)
- Auto Refresh (REF)
- Self Refresh (SELF)
- Precharge All Bank (PALL)
- Precharge Select Bank (PRE)
- Row Address Strobe and Bank Active (ACTV)
- Column Address and Read Command (READ)
- Column Address and Write Command (WRITE)
- Mode Register Set (MRS)
- Burst Stop in Full Page (BST)

6.7 Example of Synchronous DRAM Connection

Figure 6.1 shows the example for connection of 256-Mbit × 16-bit Synchronous DRAM.

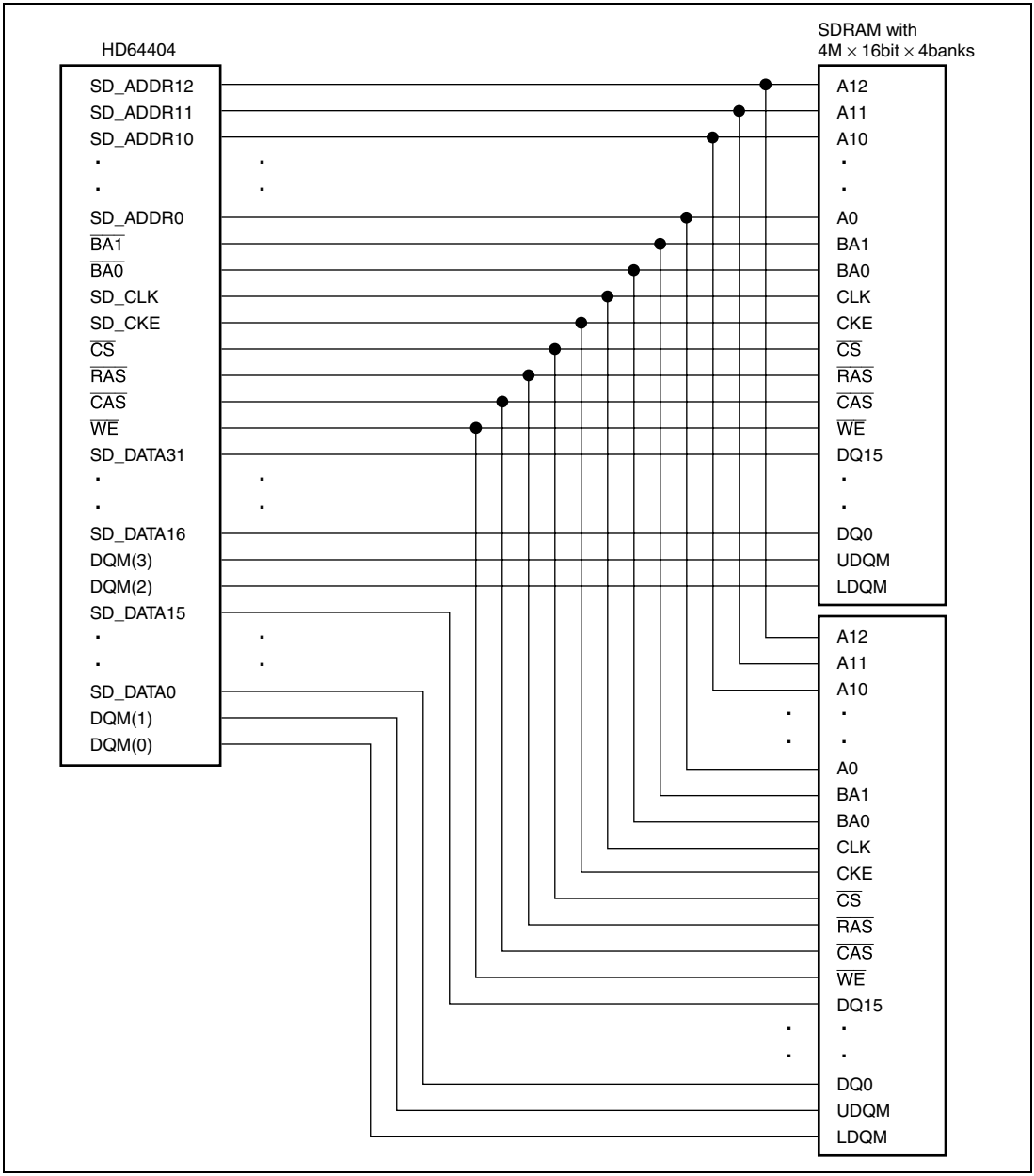


Figure 6.1 Example of Connection of 256-Mbit 16-bit Synchronous DRAM

6.8 Example of Setting Refresh Period (RP)

Set the number of clock cycles × 16 per refresh to bits 26 to 17 (Refresh Period (RP)) in the MCR register. For example, if refresh specification of memory to be used is 4096 refresh cycles/64ms, the refresh period is 15.625µs per one refresh. If the refresh request is generated during bus cycle execution, refresh execution is not done until the bus cycle ends. In this case, the value must be set to the RP bits in consideration of this waiting period. This period is approximately 55 pixel bus clock cycles in the worst case. Assuming that 1 pixel bus clock is 100MHz, 1 refresh period is calculated by the following formula: 64ms/4096 −55 × 10ns = 15.625µs −0.55µs = 15.075µs. Consequently, the MCR refresh period is 15.075µs/(10ns × 16) = 94, and the value set to RP is D'94 (H'5E).

Section 7 Memory Arbiter

7.1 General Description

This block is responsible for arbitrating between the DMA requests from the blocks connected to the pixel bus. The arbitration will be decided each cycle but the arbitration will only be accepted if the memory controller is ready to accept another transaction command.
In addition it tracks the commands that are being processed by the memory controller so it can request data or send data to the correct block.
The memory controller has a 2-stage command queue to allow the starting of a second SDRAM command before the first one is complete. This maximises the potential bandwidth of the pixel bus.

7.2 Features

- Arbitrates between all blocks on the pixel bus
- Multiplexes the data from the appropriate block to the memory controller.
- Maintains track of the current memory controller transaction.

7.3 Register description

This block has no programmable registers.

7.4 Functional description

7.4.1 Arbitration

There are both real-time and non-real time blocks connected to the pixel bus. E.g. Display output is real time, graphics renderer is non real-time. As the real time blocks will also only require a fixed bandwidth they can be allocated a fixed priority. Devices that are non-real time may also try to swamp the bus, so their allocation to the bus must be limited.

The blocks to be considered are:

- Display output plane 2 (DO2)
- Display output plane 1 (DO1)
- Display output cursor (DOC)
- Video Input (VI)
- Graphics Engine (RU)
- SH processor (SH)

- DMAC (RB)
- USB Host (US)
- Atapi (AT)

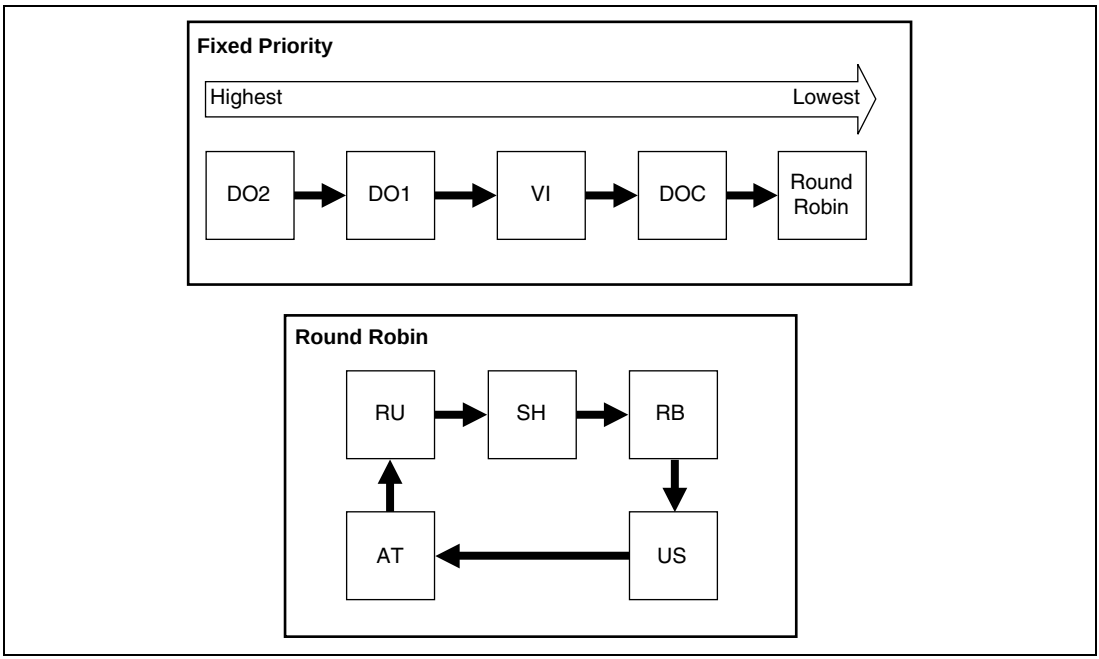


Figure 7.1 Arbitration Diagram

The arbitration adopted uses a mixture of fixed and round-robin arbitration as is shown in the diagram above.

This arbitration works as follows, DO2, DO1, VI and DOC have fixed priority with DO2 having the highest priority. The priority of each of these is higher than any of the other units. If none of these high priority units is requesting a transfer then one of the units within the round-robin scheme can be granted. The unit chosen within the round-robin scheme rotates from the previous unit within the round-robin that was granted.

7.4.2 Data transfers on the pixel bus

As there may be multiple commands within the memory controller, the memory arbiter must track when commands start and end so that it can decide which unit should supply the data to the memory controller and which unit should receive the data from the memory controller. The memory controller will still control the timing of the data transfers within a command.

Section 8 Power Control & Configuration

8.1 General Description

This module serves five purposes:

1. Generate the internal reset and external reset signals.
2. Pin mode control.
3. Clock gating.
4. USB and Audio clock control.
5. Software reset.

The two reset signals resn (internal) and reso (external), are the internal active low reset and external active low reset signals respectively.

8.2 Features

- Interfaces with Register Bus
- Software control of External reset (reso).
- Provides access to mode (pin mode) select registers
- Provides clock stopping for each module
- Crystal pad enables
- Software reset function equal to hardware reset

8.3 Digital Inputs/Outputs

Table 8.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/ Out	Description
Register Bus	—	—	Register bus interface signals
pixclk	1	In	Pixel bus clock
mode	10	Out	Mode select bits
scan_mode	1	In	Muxscan signal
resn_pix	1	Out	Internal reset signal
resn_rb	1	Out	Internal reset signal
reso_sh	1	Out	Internal reset signal for CPUIF
reso	1	Out	External reset signal
reso_en	1	Out	External reset enable signal
rbclk_mod	28	Out	Register bus clock to each peripheral
rbclk_mod2	7	Out	Register bus clock to modules with Pixel bus clock
pix_clk_mod	7	Out	Pixel bus clock to each peripheral
rbclk_rbdmac	1	Out	Register bus clock for RBDMAC
pixclk_rbdmac	1	Out	Pixel bus clock for RBDMAC
pixclk_renderer_n	1	Out	Reversal clock of Pixel bus clock for Renderer
shclk_rbdmac	1	Out	shclk for RBDMAC
usb_xtal_cont	3	Out	USB crystal control
aud_xtal_cont	2	Out	Audio clock crystal control
pixclk_enable_cpu	1	In	Pixel bus clock enable
rbclk_enable_cpu	1	In	Register bus clock enable
config	1	In	choice of CPU interface

8.4 Software Interfaces

The registers accessible by the software are listed in the following table:

Table 8.2 Register List

Address (Bytes)	Register Name	Abbreviation	Access Size
H'66A0	Mode	M	32
H'66A4	Reso	RESO	32
H'66A8	Clock Control1	CC1	32
H'66AC	Clock Control2	CC2	32
H'66BA0	Xtal Control	XTC	32
H'66BA4	Software reset	SRST	32
H'66BA8	Compare Match	CMR	32

8.5 Functional Description

The module interfaces to the register bus and has five separate functions:

The first function of this module is to hold the external reset signal (reso) until the HD64404 device has initialised completely and to generate the internal reset signal resn. As soon as RSTn (the processor reset signal received via the CPU interface) switches active low, the reso signal also switches active low. A switch of external reset signal (reso) can control software.

Note: reso refers to the name of the signal that is output from the output pin (pad) of HD64404.

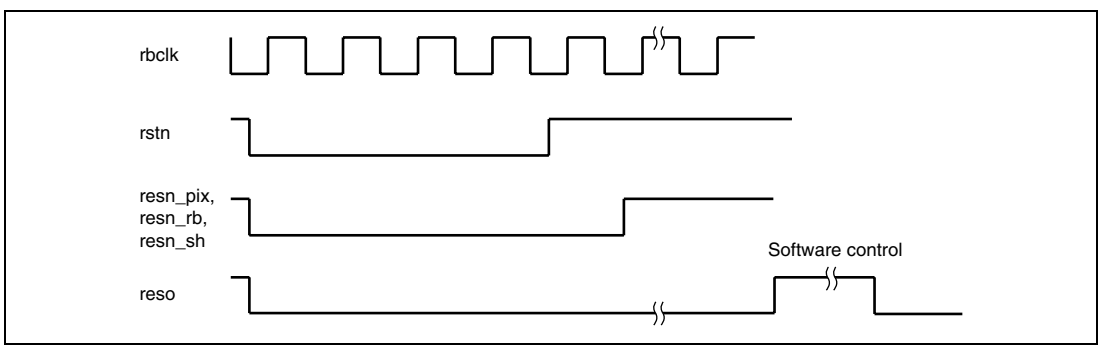


Figure 8.1 External Reset Signal Timing

The second function is the mode bits. The mode bits which are responsible for setting up the functionality of the HD64404 device. I.e. the selecting of the mode of the shared pins.

The third function is clock gating. Every module except for CPU interface and DMAC can have its clock stopped via software control. There are two registers to achieve this. Clock Control 1 Register is for register bus only modules. Clock Control2 is for pixel bus and register bus modules. Note that disabling the clocks on pixel_bus modules will stop both the pixel bus clock and the register bus clock in that module. The DMAC Clock Control Register is held within the CPU I/F(PCI I/F and MPX I/F) though the actual clock gating is performed centrally within this module.

The fourth function is X'TAL clock control bits. These bits can control an X'TAL pad for USB and Audio individually. Input a clock to the EXTAL_USB pin in USB clock input mode, or to the AUDIO_CLK pin in Audio clock input mode.

The fifth function is software-reset mode. Since software-reset mode has the same function as that of a hardware reset, register values in each module are initialized in this mode. Since the specified module(s) is (are) automatically reset when this bit is set, they cannot be accessed until recovery from reset. Recovery time is approximately 10 msec.

PLL output control (RBCLKEN bit, PIXCLKEN bit) of MPX I/F module in MPX mode (of PCI I/F module in PCI mode) must be enabled before accessing the registers of this module.

Recovery time from standby can be specified by setting the corresponding bit of the cmr register. Setting 0 will specify 10 msec as recovery time; setting 1 will 10 usec.

8.6 Register Descriptions

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, read value undefined.

8.6.1 Mode Register (M)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MODE							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
7 to 0	MODE	0	R/W	**Mode Select Bits (MODE)** Each bit sets up different functionality in the HD64404 device. For more details, refer to the section of mode bits on pin_mode_replacement. Each mode is remarked as M (No.). Mode (3) is reserved. Note: GPIO_inactive resister information in GPIO ,GPIO1 are given priority than Mode Select Bits. Switch mode using this register only when the corresponding module is not in operation. Otherwise, an unpredicted error may be caused.

8.6.2 Reso Register (RESO)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																RESO
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
0	RESO	0	R/W	**External reset (RESO)** This bit controls the reset for external devices. 0: RESO signal is Low. (default) 1: RESO signal is High.

For all bits: 0: Pin function A (default)
1: Pin function B

8.6.3 Clock Control 1 Register (CC1)

This bit controls a register bus clock.

For all bits: 0: Clock stopped. (all bits default)
1: Clock active.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					CSC		GIO1	EXP		MOST	PWM	TIME	CAN1	CAN0	SPD	INTP
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INTI	GIO0	AC	UAR3	UAR2	UAR1	UAR0	SPI2	SPI1	SPI0	SSI3	SSI2	SSI1	SSI0	I2C1	I2C0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
27	CSC	0	R/W	**Color Space Converter (CSC)**
26	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
25	GIO1	0	R/W	**General Purpose IO 1 (GIO1)**
24	EXP	0	R/W	**Expansion Bus (EXP)**
23	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
22	MOST	0	R/W	**Most Interface (MOST)**
21	PWM	0	R/W	**Pulse Width Modulation (PWM)**
20	TIME	0	R/W	**Timer (TIME)**
19	CAN1	0	R/W	**Hitachi Can 1 (CAN1)**
18	CAN0	0	R/W	**Hitachi Can 0 (CAN0)**
17	SPD	0	R/W	**SPDIF Interface (SPD)**
16	INTP	0	R/W	**Interrupt Priority Controller (INTP)**
15	INTI	0	R/W	**Interrupt Input (INTI)**
14	GIO0	0	R/W	**General Purpose IO 0 (GIO0)**
13	AC	0	R/W	**Audio Codec (AC)**
12	UAR3	0	R/W	**Uart 3 (UAR3)**
11	UAR2	0	R/W	**Uart 2 (UAR2)**
10	UAR1	0	R/W	**Uart 1 (UAR1)**
9	UAR0	0	R/W	**Uart 0 (UAR0)**
8	SPI2	0	R/W	**Serial Peripheral Interface 2 (SPI2)**
7	SPI1	0	R/W	**Serial Peripheral Interface 1 (SPI1)**
6	SPI0	0	R/W	**Serial Peripheral Interface 0 (SPI0)**
5	SSI3	0	R/W	**Serial Sound Interface 3 (SSI3)**
4	SSI2	0	R/W	**Serial Sound Interface 2 (SSI2)**
3	SSI1	0	R/W	**Serial Sound Interface 1 (SSI1)**
2	SSI0	0	R/W	**Serial Sound Interface 0 (SSI0)**
1	I2C1	0	R/W	**Inter IC Communication 1 (I2C1)**
0	I2C0	0	R/W	**Inter IC Communication 0 (I2C0)**

Change each bits value only when the corresponding module is not in operation. If the bit value is changed while accessing the internal bus or the external interface, on unpredicted error may be caused.

8.6.4 Clock Control 2 Register (CC2)

These bits control clock supply for the pixel bus and the register bus.

For all bits: 0: Clock stopped. (ATA, USB, VI and MEM default)
1: Clock active. (REND and DO default)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										MEM	VI	DO	ATA	USB		REND
Initial:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
6	MEM	0	R/W	**Memory Interface (MEM)**
5	VI	0	R/W	**Video Input (VI)**
4	DO	1	R/W	**Display Output (DO)**
3	ATA	0	R/W	**ATAPI Interface (ATA)**
2	USB	0	R/W	**USB Host and Function (USB)**
1	—	0	R	**Reserved**
0	REND	1	R/W	**Renderer (REND)**

Change each bits value only when the corresponding module is not in operation. If the bit value is changed while accessing the internal bus or the external interface, on unpredicted error may be caused.

8.6.5 Xtal Control Register (XTC)

These bits control the X'TAL pad (USB: 48MHz, AUDIO: 22.5792MHz or 24.576MHz)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												AUDX	AUD	USBX	USB0	USB1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R	**Reserved** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
4	AUDX	0	R/W	**AUDIO X'TAL (AUDX)** 0: Clock input 1: X'TAL oscillation For audio clock input, input a clock to the AUDIO_CLK pin instead of EXTAL_AUD.
3	AUD	1	R/W	**AUDIO (AUD)** 0: Audio Clock enabled. 1: Audio Clock disabled. Note: Audio Clock is the clock supplied to SSI0,1,2,3 and SPDIF modules.
2	USBX	1	R/W	**USB X'TAL (USBX)** 0: Clock input 1: X'TAL oscillation For clock input, input a clock to the EXTAL_USB pin.
1	USB0	1	R/W	**USB Host (USB0)** 0: Clock enabled. 1: Clock disabled. Note: Clock is the meaning of the clock supplied to USB HOST module.
0	USB1	1	R/W	**USB Function (USB1)** 0: Clock enabled. 1: Clock disabled. Note: Clock is the meaning of the clock supplied to USB Function module.

Change each bits value only when the corresponding module is not in operation. If the bit value is changed while accessing the internal bus or the external interface, on unpredicted error may be caused.

8.6.6 Software Reset Register (SRST)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SWRT
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	0	R	**Reserved bit** Reserved bits return 0 when read. When writing to these bits, 0 should be written to.
0	SWRT	1	R/W	**Software reset (SWRT)** This bit controls the reset for HD64404. 0: Internal reset signal is Low. 1: Internal reset signal is High. (default) Since this is a low-active bit, write 0 to this bit for software reset. This bit will be automatically set to 1 after recovery from reset. Recovery time is approximately 10 msec.

8.6.7 Compare Match Register (CMR)

Reset value : H'00000000

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—		R/W	**Reserved bit**
0	CMR	0	R/W	**Compare Match (CMR)** This bit selects recovery time from reset. 0: 10 msec 1: 10 usec Use this bit only to shorten test time (e.g., for standby test).

8.7 Power Saving and Clocking Strategy

Internal logic is clocked by the input register bus clock rbclk.

PLL output control (RBCLKEN bit ,PIXCLKEN bit) of MPX I/F module in MPX mode (of PCI I/F module in PCI mode) must be enabled before accessing the registers of this module. Since these bits control the clock that is input to this module, the registers of this module cannot be controlled unless these bits are enabled.

In case there are some registers that are not used or this module is used in low-power consumption mode, stop clock supply by setting 0 to the corresponding bit in Clock Control 1 Register (bits 27 to 0), Clock Control 2 Register (bits 6 to 0), or Xtal Control Register (bits 3, 1, 0).

To write 1 to AUDIO (AUD) bit in Xtal Control Register, SSI01, 2, 3 and SPDIF must all be stopped.

To write 1 to USB Host (USB0) bit in Xtal Control Register, USB Host (USB) in Clock Control 2 Register must be stopped. To write 1 to USB Function (USB1) bit in Xtal Control Register, USB function (USB) in Clock Control 2 Register must be stopped. Other bits in Clock Control 1 Register and Clock Control 2 Register can be independently set.

X'TAL input and clock input can be selected for USB and AUDIO. When X'TAL is selected, XTAL_AUD and EXTAL_AUD is used for AUDIO; XTAL_USB and EXTAL_USB is used for USB.

When the clock input mode is selected, EXTAL_USB pin is used for USB; AUDIO_CLK pin is used for AUDIO.

The clock input mode for USB is set by USBX bit in Xtal Control Register. The clock input mode for AUDIO is set by AUDX = 0 in Xtal Control Register.

Note: EXTAL_AUD is not available for the external clock input. It is for only X'TAL usage.

Recovery time from standby can be specified by setting the CMR bit. Setting 0 will specify 10 msec, and setting will 10 usec. In the operation of a mass-production chip, however, recovery time is only 10 msec.

8.7.1 Procedure for Power On Sequence

1. Start supplying a clock to the PCI_CLK pin (to the CKIO pin in MPX mode).
2. Input low to the PLL_ENABLEN and $\overline{\text{RST}}$ pins.
3. Wait for 11msec.
4. Turn $\overline{\text{RST}}$ into High.
5. Enable PLLCTL in CPU I/F to supply register bus clock and pixel bus clock.
6. Supply clock in each module by setting an arbitrary bit of the Clock Control 1 Register, Clock Control 2 Register, and Xtal Control Registers.

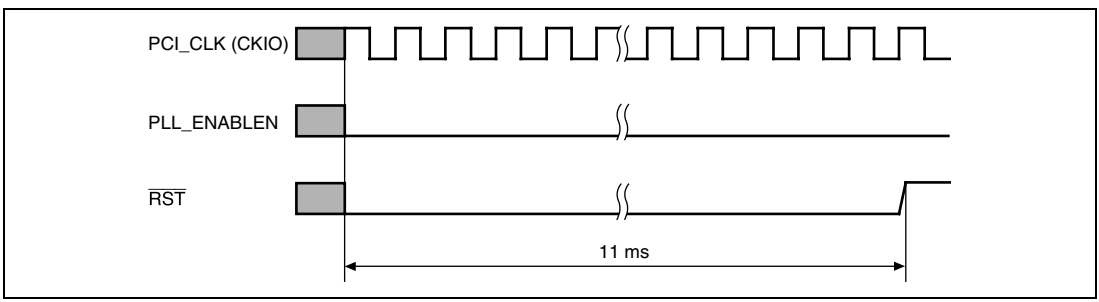


Figure 8.2 Power-On Sequence Timing

8.7.2 Procedure for Power Down and Wake Up

The following power down modes are provided :

1. Module Standby mode
2. Standby mode
3. Deep Standby mode

Table 8.3 Power down modes

Module\mode	Module Standby	Standby	Deep Standby
Each Module except DMAC, Power Control, CPU I/F, and system PLL	Clock is off for each module which Clock control 1, Clock Control 2 or Xtal Control Register is set as 0.	Clock is off for all modules	Clock is off for all modules
DMAC and Power Control	Clock is on	Clock is off	Clock is off
CPU I/F and system PLL	Clock is on	Clock is on	Clock is off
Suggested System Processor power saving mode	normal	standby	standby

Notes: 1. Do not access a module while its clock is stopped.

2. Setting MEM bit to 0, disabling Memory interface, in Clock Control 2 Register is available only after all other pixel bus modules are disabled, i.e. Clock Control 2 Register [5:0] = "000000". On the other hand, setting MEM to 1, enabling Memory interface, in Clock Control 2 Register has to be done before any other module is enabled.
3. While HD64404 is in either the Standby mode and Deep Standby mode, all output and bi-directional pin states are held . HD64404 does not support Deep Standby mode where all the signal is in High-Z. So if Graphics Memory power needs to be off, HD64404 power needs to be off either.
4. Difference between Standby and Deep Standby is that wake up time is faster in Standby mode because there is no need to wait for PLL stabilization period.

(1) Power Down Procedure

Procedure for Power down into Module Standby Mode:

If the Graphics Memory (MEM) is set to module standby mode, Graphics memory must not be accessed.

1. Write "0" (= Clock stopped) to the desired module bit in the Clock Control 1 Register, Clock Control 2 Register, and Xtal Control Registers.

Note: When setting Graphics memory I/F to standby mode, it is important that all the modules on the pixel bus are disabled as a general requirement of the memory interface power down sequence before executing the Self Refresh mode.

Procedure for Power down into Standby Mode: In this mode, software should run in ROM since Graphics memory could set in self refresh mode.

1. Wait until all transactions on the pixel bus from pixel bus modules have completed.
2. Set the IRQ_LEVEL_EDGE bits of IRQ Control Register in Interrupt input Module to be active level.
3. Write "1" to STBY bit of IRQ Control Register in Interrupt input Module.
4. Write "1" to STB bit of IRQ Winner Register in Interrupt Priority Module.
   Then Interrupt Priority Module does not latch the priorities at each clock cycle. However it will be able to output a signal on the irq pin when the external interrupts are asserted.
5. Execute Self refresh (ME = SR = 1) mode. Refer to table 6.3 for the ME and SR bit settings in Memory Interface Module document .
6. Write "0" (= Clock stopped) to all of module bits in the Clock Control 1 Register, Clock Control 2 Register, and Xtal Control Registers.
7. Disable PLL output control (RBCLKEN bit ,PIXCLKEN bit) of MPX I/F module in MPX mode (of PCI I/F module in PCI mode) .
8. Set System Processor in Standby mode.

Note: Before executing the Self Refresh mode, it is important that all the modules on the pixel bus are disabled as a general requirement of the memory interface power down sequence.

Procedure for Power down into Deep Standby mode: In this mode, software should run in ROM since Graphics memory could set in self refresh mode.

1. Wait until all transactions on the pixel bus from pixel bus modules have completed.
2. Set the IRQ_LEVEL_EDGE bits of IRQ Control Register in Interrupt input Module to be active level.
3. Write "1" to STBY bit of IRQ Control Register in Interrupt input Module.
4. Write "1" to STB bit of IRQ Winner Register in Interrupt Priority Module.

   Then Interrupt Priority Module does not latch the priorities at each clock cycle. However it will be able to output a signal on the irq pin when the external interrupts are asserted.
5. Execute Self refresh (ME = SR = 1) mode. Refer to table 6.3 for the ME and SR bit settings in Memory Interface Module document .
6. Write "0" (= Clock stopped) to all of module bits in the Clock Control 1 Register, Clock control 2 Register, and Xtal Control Registers.
7. Disable PLL output control (RBCLKEN bit ,PIXCLKEN bit) of MPX I/F module in MPX mode (of PCI I/F module in PCI mode) .
8. Input high to the PLL_ENABLEN pin to stop PLL.
9. Stop clock supply to the PCI_CLK pin (to the CKIO pin in MPX mode).
10. Set System Processor in Standby Mode.

(2) Wake Up Procedure

Reset is necessary for return from each standby mode.

Procedure for Wake up from Module Standby mode:

1. Write "1" (= Clock active) the desired module bits in the Clock Control 1 Register, Clock Control 2 Register, and Xtal Control Registers.

Procedure for Wake up from Standby mode: In this mode, software should run in ROM since Graphics memory could set in self refresh mode.

1. Interrupt Input trigger the System Processor to wake up system
2. Enable PLL output control (RBCLKEN bit, PIXCLKEN bit) of MPX I/F module in MPX mode (of PCI I/F module in PCI mode).
3. Write "1" (= Clock active) to the desired mode bits in the Clock Control 1 Register, Clock Control 2 Register, and Xtal Control Registers.
4. Recover from Self refresh to Normal mode (ME = 1, SR = 0) . Refer to table 6.3 for the ME and SR bit settings in Memory Interface Module document.
5. Write "0" to STB bit of IRQ Winner Register in Interrupt Priority Module.
6. Write "0" to STBY bit of IRQ Control Register in Interrupt input Module.
7. Jump to OS wake-up procedure

Note: When the external inputs are asserted and they are unmasked in Interrupt input module, irq pin output is enabled and interrupt to host processor is asserted.

Procedure for Wake up from Deep Standby Mode: In this mode, software should run in ROM since Graphics memory could set in self refresh mode.

1. Interrupt Input trigger the System Processor to wake up system
2. Restart supplying a clock to the PCI_CLK pin (to the CKIO pin in MPX mode).
3. Input low to the PLL_ENABLEN pin to start PLL.
4. Wait for 11msec.
5. Enable PLL output control (RBCLKEN bit, PIXCLKEN bit) of MPX I/F module in MPX mode (of PCI I/F module in PCI mode).
6. Write "1" (= Clock active) to the desired mode bits in the Clock Control 1 Register, Clock Control 2 Register, and Xtal Control Registers.
7. Recover from Self refresh to Normal mode (ME = 1, SR = 0) . Refer to table 6.3 for the ME and SR bit settings in Memory Interface Module document .
8. Write "0" to STB bit of IRQ Winner Register in Interrupt Priority Module.
9. Write "0" to STBY bit of IRQ Control Register in Interrupt input Module.
10. Jump to OS wake-up procedure

Note: In case of Module Standby mode and Standby mode, refer to respective module specifications.

8.8 Clock Pulse Generator

Figure 8.3 shows the block diagram of HD64404 clock pulse generator. PLL Circuit 1: PLL circuit 1 (PLL1) has a function for multiplying the clock frequency from the CKIO (PC1_CLK) pin. PLL also has a function for 11 msec delay signal that supplies the stabilized clock to MPX i/f and PCI i/f module.

PLL circuit 2: PLL circuit 2 (PLL2) generates the dot clock for display out.

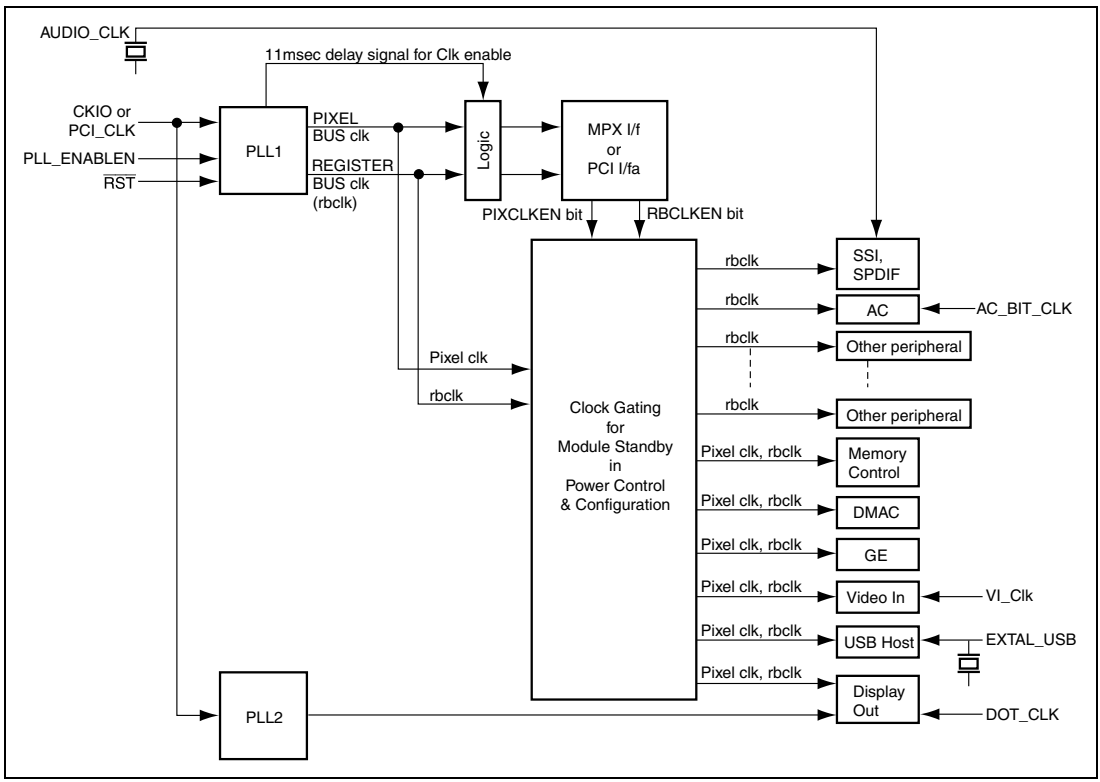


Figure 8.3 Block Diagram of Clock Pulse Generator

Section 9 Video Input Module

9.1 Overview

This video interface module accepts video input data from a video decoder in ITU-R BT.656 format which is a nine bit interface operating at 27 MHz. This 4:2:2 YCbCr input format can be converted to 5:6:5 RGB format using an in-built color space converter (CSC) matrix. This module supports a maximum of 720 × 288 pixel field size. This module is also capable of up and down scaling in the y direction using bilinear interpolation, and down scaling in the x direction using a multiphase filter. The scaled image is stored in the linear mapped graphic memory.

9.1.1 Features

- ITU-R BT.656 interface.
- Size clipping before/after scaling.
- Horizontal down scaling using a 9 tap multiphase filter.
- Vertical up/down scaling using bilinear interpolation.
- Color space conversion and dithering from 4:2:2 YCbCr to RGB 5:6:5.

9.1.2 Block Diagram

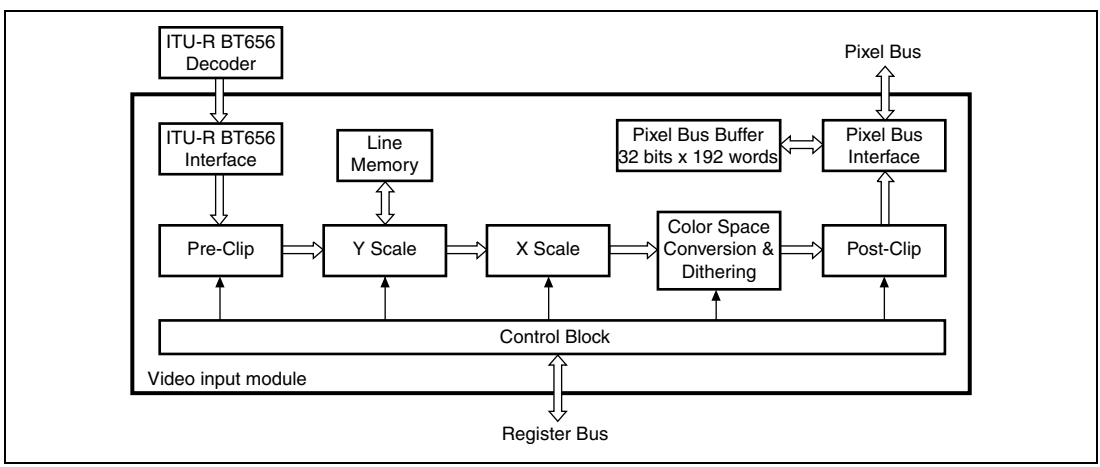


Figure 9.1 System Diagram

9.2 Pin Descriptions

Table 9.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From
VI_CLK	1	In	Video Clock	Video Decoder
VI_Data (7:0)	8	In	Video Data	Video Decoder

Note: The register bus and pixel bus provide their own clocks.

9.3 Register Description

9.3.1 Register Summary

The registers accessible by the software are listed in the following table:

Table 9.2 Video interface Register Map

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6400	Main Control	MC	32
H'6404	Module Status	MS	32
H'6408	Frame Capture	FC	32
H'640C	Start Line Pre-Clip	SLPrC	32
H'6410	End Line Pre-Clip	ELPrC	32
H'6414	Start Pixel Pre-Clip	SPPrC	32
H'6418	End Pixel Pre-Clip	EPPrC	32
H'641C	Start Line Post-Clip	SLPoC	32
H'6420	End Line Post-Clip	ELPoC	32
H'6424	Start Pixel Post-Clip	SPPoC	32
H'6428	End Pixel Post-Clip	EPPoC	32
H'642C	Image Stride	IS	32
H'6430	Memory Base 1	MB1	32
H'6434	Memory Base 2	MB2	32
H'6438	Memory Base 3	MB3	32
H'643C	Line Count	LC	32
H'6440	Interrupt Enable	IE	32
H'6444	Interrupt Status	INTS	32

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6448	Scan line Interrupt	SI	32
H'6450	Y Scale	YS	32
H'6454	X Scale	XS	32
H'6480	Coeff Set 1a	C1A	32
H'6484	Coeff Set 1b	C1B	32
H'6488	Coeff Set 1c	C1C	32
H'6490	Coeff Set 2a	C2A	32
H'6494	Coeff Set 2b	C2B	32
H'6498	Coeff Set 2c	C2C	32
H'64A0	Coeff Set 3a	C3A	32
H'64A4	Coeff Set 3b	C3B	32
H'64A8	Coeff Set 3c	C3C	32
H'64B0	Coeff Set 4a	C4A	32
H'64B4	Coeff Set 4b	C4C	32
H'64B8	Coeff Set 4c	C4C	32
H'64C0	Coeff Set 5a	C5A	32
H'64C4	Coeff Set 5b	C5B	32
H'64C8	Coeff Set 5c	C5C	32
H'64D0	Coeff Set 6a	C6A	32
H'64D4	Coeff Set 6b	C6B	32
H'64D8	Coeff Set 6c	C6C	32
H'64E0	Coeff Set 7a	C7A	32
H'64E4	Coeff Set 7b	C7B	32
H'64E8	Coeff Set 7c	C7C	32
H'64F0	Coeff Set 8a	C8A	32
H'64F4	Coeff Set 8b	C8B	32
H'64F8	Coeff Set 8c	C8C	32

Legends for register description:

Initial Value : Register value after reset
— : Read → undefined value, Write → always "0" write
* : Value is retained
R/W : Read and Write register
R : Read only register, for write always 0 write

9.3.2 Register Descriptions

Main Control Register (MC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						VUP				EN	EC	IM[1:0]				ME
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	0	R	**Reserved**
10	VUP	0	R/W	**Vsync Update (VUP)** This bit determines the timing for updating register setting. When this bit is set to 1, register values are updated at the timing of the change of field bit (F-bit) of ITU-R BT.656. When this bit is set to 0, register values are updated immediately. This vsync update mode is effective for SLPrC, ELPrC, SPPrC, EPPrC, SLPoC, ELPoC, SPPoC, EPPoC, IS, MB1, MB2, MB3, SI, BS, YS and XS Registers. 0: Immediate update 1: Vsync update
9 to 7	—	0	R	**Reserved**
6	EN	0	R/W	**Endian Type (EN)** This bit selects the endian type on pixel bus. RGB565 pixel data can be placed on pixel bus with either little endian or big endian packing. 0: Little endian 1: Big endian

Bit	Bit Name	Initial Value	R/W	Description
5	EC	0	R/W	**Error Correction (EC)** When this bit is set to 1, error correction is performed on the ITU-R BT.656 input. When this bit is set to 0 no error correction is performed on the ITU-R BT.656 input. If the device providing the ITU-R BT.656 data stream does not support protection bit encoding into the ITU-R BT.656 timing reference commands then error correction must be turned off for correct operation. 0: ITU-R BT.656 error correction is disabled. 1: ITU-R BT.656 error correction is enabled.
4	IM1	0	R/W	**Interlace Mode (IM)**
3	IM0	0	R/W	There are four frame modes supported by this module, the following is a description of each mode. Odd/Even Field Frame Capture mode cannot be used with Single Capture mode described in Frame Capture (FC) Register. The other modes can be used in either Single Capture or Continuous Capture mode. 00: Odd Field (Field1) Capture Only the odd field is processed by the module. 01: Odd/Even Field Frame Capture The odd and even field from an input frame are processed as single frames, producing two output frames for one input frame. 10: Even Field (Field2) Capture Only the even field is processed by the module. 11: Full Interlace Both the odd and even field are processed by the module. When the module is capturing in this mode, the odd field is captured first leaving gaps in memory to allow the even field to be interleaved when it is captured after the odd field has finished. The following example shown in Figure 9.2 is based on a stride of H'200 and the Memory Base 1 (MB1) Register set to H'0000. Note: Full interlace mode cannot be used in conjunction with vertical scaling.
2, 1	—	0	R	**Reserved**

Bit	Bit Name	Initial Value	R/W	Description
0	ME	0	R/W	**Module Enable (ME)** When this bit is set to 1 the module is enabled, when this bit is set to 0 the module is disabled. If the module is disabled during a frame grab operation the module will complete that frame grab before disabling. When the module is disabled no monitoring or processing is carried out on the input video data, and all internal state machines hold in an idle state. 0: Video input module is disabled. 1: Video input module is enabled.

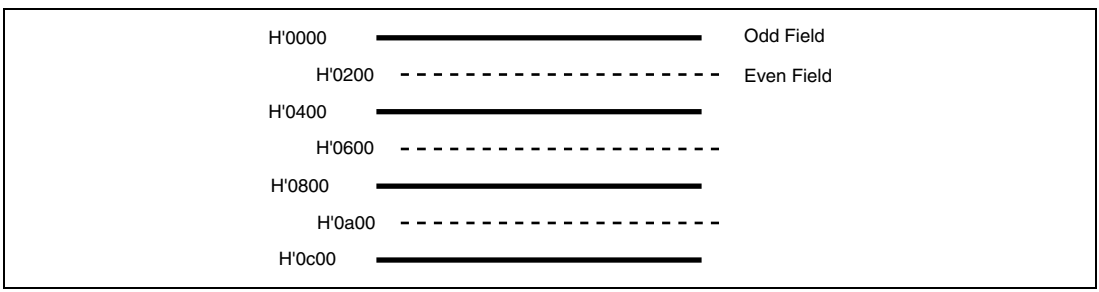


Figure 9.2 Interlaced Frame Memory Interleaving

Module Status Register (MS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FBS[1:0]		FS	AV	CA
Initial:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R	**Reserved**
4	FBS[1]	1	R	**Frame Buffer Status (FBS)**
3	FBS[0]	1	R	This bit field shows the frame buffer status. Video input module has three frame buffers and controls the read and write pointer for correct operation. The register bits Frame Buffer Status (FBS) shows the value of the read pointer. 00: The frame buffer with a base address as specified in the Memory Base 1 (MB1) Register is the latest valid frame buffer. 01: The frame buffer with a base address as specified in the Memory Base 2 (MB2) Register is the latest valid frame buffer. 10: The frame buffer with a base address as specified in the Memory Base 3 (MB3) Register is the latest valid frame buffer. 11: No frame buffer is valid
2	FS	0	R	**Bit 2 – Field State (FS)** This bit indicate the field type of current captured field. 0: Current field is Field 1 (Odd). 1: Current field is Field 2 (Even).
1	AV	0	R	**Active Video (AV)** This bit indicates whether the current pointer for capturing is in the range of active video region or not. Where input data is not captured, this bit goes to 0. Active video region is determined by the Pre-clipping Registers. 0: Current field is not in active video region. 1: Current field is in active video region.
0	CA	0	R	**Capture Active (CA)** This bit indicates whether currently video capture is activated or not. This status bit is set to 1 even if current field is not captured. For example, this is the case when the current video input field is Field2 and Interlace Mode (IM) equals to H'0 (Field1 capture). This bit is updated at the timing of the change of F-bit specified in ITU-R BT.656. 0: Video capture is not activated. 1: Video capture is activated.

Frame Capture Register (FC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															CC	SC
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	0	R	**Reserved**
1	CC	0	R/W	**Continuous Capture (CC)** Writing 1 to this bit causes the video in module to continuously capture frames, writing the first frame to the memory address stored in Memory Base 1 (MB1) Register, then writing the next frame to the address stored in Memory Base 2 (MB2) Register etc. Following the cycle Memory Base 1-2-3-1-2-3 etc. Writing 0 to this bit during a continuous capture operation, causes the video in to stop capturing at the end of the current frame or immediately if no frame is being currently captured.
0	SC	0	R/W	**Single Capture (SC)** Writing 1 to this bit causes a single frame to be captured, if the current frame line count is before the value in the Start Line Pre-Clip (SLPrC) Register the current frame is grabbed else the next frame is grabbed. Frames grabbed in single capture mode are placed at the start address defined in the Memory Base 1 (MB1) Register. After setting this SC bit, Frame Buffer Status (FBS) bit in Module Status (MS) Register immediately changes to the value H'3. Writing 1 to this bit when a single frame capture or continuous capture is in operation has no effect. This bit always returns zero. The capturing status can be checked by reading Capture Active (CA) bit in Module Status (MS) Register.

Start Line Pre-Clip Register (SLPrC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SLPrC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	SLPrC[9:0]	0	R/W	Register value is a number between 0 and 287, this sets the start line which is valid for capture. The value 0 represents the first line where V-bit specified in ITU-R BT.656 changes from V = 1 to V = 0. This value is used prior to vertical or horizontal scaling.

End Line Pre-Clip Register (ELPrC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							ELPrC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	ELPrC[9:0]	0	R/W	Register value is a number between 0 and 287, this sets the last line which is valid for capture. The value 0 represents the first line where V-bit specified in ITU-R BT.656 changes from V = 1 to V = 0. This value is used prior to vertical or horizontal scaling.

Start Pixel Pre-Clip Register (SPPrC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SPPrC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	SPPrC[9:0]	0	R/W	Register value is a number between 0 and 719, which indicates the first pixel on the current line which is valid for capture. This value is used prior to scaling. This module only allows pre-clipping in multiples of two due to the sub-sampled nature of the ITU-R BT.656 input (4:2:2 YCbCr). The LSB of this register is ignored. The register value (2n+1) is identical with the value (2n). The number of pixels within the horizontal pre-clip windows (End Pixel Pre-Clip(EPPrC) – Start Pixel Pre-Clip(SPPrC)) must be greater than or equal to 5.

End Pixel Pre-Clip Register (EPPrC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							EPPrC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	EPPrC[9:0]	0	R/W	Register value is a number between 0 and 719, which indicates the last pixel in the current line which is valid for capture. This value is used prior to scaling. This module only allows pre-clipping in multiples of two plus one due to the sub-sample nature of the ITU-R BT.656 input (4:2:2 YCbCr). The LSB of this register is ignored. The register value (2n) is identical with the value (2n+1). The number of pixels within the horizontal pre-clip windows {End Pixel Pre-Clip(EPPrC) – Start Pixel Pre-Clip(SPPrC)} must be greater than or equal to 5.

Start Line Post-Clip Register (SLPoC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SLPoC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	SLPoC[9:0]	0	R/W	Register value is a number between 0 and 863, this sets the first line of the scaled image to be written to memory. The value 0 represents the first line of the scaled image.

End Line Post-Clip Register (ELPoC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							ELPoC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	ELPoC[9:0]	0	R/W	Register value is a number between 0 and 863, this sets the last line of the scaled image to be written to memory. The value 0 represents the first line of the scaled image.

Start Pixel Post-Clip Register (SPPoC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SPPoC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	SPPoC[9:0]	0	R/W	Register value is a number between 0 and 719, this sets the first pixel of each line of the scaled image to be written to memory. Post-clipping if performed after the 4:2:2 YCbCr is converted to 4:4:4 YCbCr therefore there is no need to restrict post-clipping values to multiples of two.

End Pixel Post-Clip Register (EPPoC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							EPPoC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	EPPoC[9:0]	0	R/W	Register value is a number between 0 and 719, this sets the last pixel of each line of the scaled image to be written to memory.

Image Stride Register (IS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							IS[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	IS[9:0]	0	R/W	This register shows the stride of the image which is the number of long words written to memory. The register value must be larger than or equal to {End Pixel Post-Clip(EPPoC) – Start Pixel Post-Clip(SPPoC)+2}/2 for correct operation.

Memory Base 1 Register (MB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						MB1[26:16]										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MB1[15:2]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	0	R	**Reserved**
26 to 2	MB1[26:2]	0	R/W	This register is the default memory base register, when this module is operating in single frame capture mode, Memory Base 2 and Memory Base 3 Registers are ignored. The value in this register specifies the start memory address the captured frame is written to. The value is four byte aligned.
1, 0	—	0	R	**Reserved**

Memory Base 2 Register (MB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						MB2[26:16]										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MB2[15:2]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	0	R	**Reserved**
26 to 2	MB2[26:2]	0	R/W	This register contains a frame base memory address, when the video in module is operating in continuous frame capture mode, this register is used in the capture sequence as follows, Memory Base 1-2-3-1-2-3 etc. The value in this register specifies the start memory address the captured frame is written to. The value is four byte aligned
1, 0	—	0	R	**Reserved**

Memory Base 3 Register (MB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						MB3[26:16]										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MB3[15:2]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	0	R	**Reserved**
26 to 2	MB3[26:2]	0	R/W	This register contains a frame base memory address, when the video in module is operating in continuous frame capture mode, this register is used in the capture sequence as follows. Memory Base 1-2-3-1-2-3 etc. The value in this register specifies the start memory address the captured frame is written to. The value is four byte aligned.
1, 0	—	0	R	**Reserved**

Line Count Register (LC)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							LC[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	LC[9:0]	0	R	Register value is a number between 0 and 287, this value indicates the current line position in the current capture field. The value after pre-clipping operation is shown in this register. The value 0 represents that the current capture line points to Start Line Pre-Clip(SLPrC). The current field status can be determined by reading the FS bit of the Module Status(MS) Register.

Interrupt Enable Register (IE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIE2															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIE	CEE	SIE	EFE	FOE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	FIE2	0	R/W	**Field Interrupt Enable 2 (FIE2)** This bit activates the change of field interrupt. The interrupt signal is asserted at the point of the change of F-bit specified in ITU-R BT.656. The interrupt signal due to this enable bit is asserted even if capture operation is not activated. 0: Change of field interrupt is disabled. 1: Change of field interrupt is enabled.
30 to 28	—	0	R/W	**Reserved** Always write 0.
27 to 5	—	0	R	**Reserved**
4	FIE	0	R/W	**Field Interrupt Enable (FIE)** This bit activates the change of field interrupt. The interrupt signal is asserted at the point of the change of F-bit specified in ITU-R BT.656. When CA bit in Module Status (MS) Register equals to 1, this interrupt enable is effective. 0: Change of field interrupt is disabled. 1: Change of field interrupt is enabled.
3	CEE	0	R/W	**Timing Reference Code Error Enable (CEE)** This bit activates the interrupt due to the timing reference code (SAV/EAV) error which is described in ITU-R BT.656 specification. The interrupt is asserted when EC bit in Main Control(MC) Register is enabled and more than two bits error occurs. If one bit error occurs and the error correction is enabled, the interrupt signal due to this interrupt enable is not asserted. When CA bit in Module Status (MS) Register equals to 1, this interrupt enable is effective. 0: ITU-R BT.656 timing reference code error interrupt is disabled. 1: ITU-R BT.656 timing reference code error interrupt is enabled.

Bit	Bit Name	Initial Value	R/W	Description
2	SIE	0	R/W	**Scanline Interrupt Enable (SIE)** This bit activates the scanline interrupt. The line where interrupt signal is asserted is determined by Scanline Interrupt (SI) Register. When CA bit in Module Status (MS) Register equals to 1, this interrupt enable is effective. 0: Scanline interrupt is disabled. 1: Scanline interrupt is enabled.
1	EFE	0	R/W	**End of Frame Interrupt Enable (EFE)** This bit activates the end of frame interrupt. The interrupt signal is asserted at the end of Field 2 (even field). When CA bit in Module Status (MS) Register equals to 1, this interrupt enable is effective. 0: End of frame interrupt is disabled. 1: End of frame interrupt is enabled.
0	FOE	0	R/W	**Pixel Bus Buffer Overflow Interrupt Enable (FOE)** This bit activates the Pixel Bus Buffer overflow interrupt. The Pixel Bus Buffer is used for transferring pixel data from video input module to pixel bus. If the Pixel Bus Buffer overflows, pixel data will be lost. When CA bit in Module Status (MS) Register equals to 1, this interrupt enable is effective. 0: Pixel Bus Buffer overflow interrupt is disabled. 1: Pixel Bus Buffer overflow interrupt is enabled.

Interrupt Status Register (INTS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIS2															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												FIS	CES	SIS	EFS	FOS
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	FIS2	0	R/WC1	**Field Interrupt Status 2 (FIS2)** When this bit is 1 this indicates that the field has been changed, writing 1 to this bit clears it.
30 to 28	—	0	R	**Reserved** Read: Undefined value
27 to 5	—	0	R	**Reserved**
4	FIS	0	R/WC1	**Field Interrupt Status (FIS)** When this bit is 1 this indicates that the field has been changed, writing 1 to this bit clears it.
3	CES	0	R/WC1	**Timing Reference Code Error Status (CES)** When this bit is 1 this indicates that the current timing reference code, even after possible one bit error correction is not a valid next state, writing 1 to this bit clears it.
2	SIS	0	R/WC1	**Scanline Interrupt Status (SIS)** When this bit is 1 this indicates that the line set in the Scanline Interrupt (SI) Register has been reached, writing 1 to this bit clears it.
1	EFS	0	R/WC1	**End of Frame Interrupt Status (EFS)** When this bit is 1 this indicates that the end of frame has been reached, writing 1 to this bit clears it.
0	FOS	0	R/WC1	**Pixel Bus Buffer Overflow Interrupt Status (FOS)** When this bit is 1 this indicates that there has been a Pixel Bus Buffer overflow, writing 1 to this bit clears it.

Scanline Interrupt Register (SI)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SI[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	SI[9:0]	0	R/W	**Scanline Interrupt (SI)** Register value is a number between 1 and 287, which indicates at which line in each field an interrupt is initiated if the SIE bit is set in the Interrupt Enable (IE) Register. The value of this Register is compared with the value of Line Count (LC) Register, and when it equals to the value of Line Counter (LC) Register, the interrupt signal is asserted.

Burst Size Register (BS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												BS[4:0]				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R	**Reserved**
4 to 0	BS4	0	R/W	Between 4 and 31, these bits indicates the number of longwords that are written to the Pixel Bus Buffer before a write to memory is performed. The default burst size is 8.
	BS3	0	R/W	
	BS2	0	R/W	
	BS1	0	R/W	
	BS0	0	R/W	

Y Scale Register (YS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MantissaY[3:0]				FractionY[15:0]											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 12	MantissaY[3:0]	0	R/W	**Mantissa for Y scaling**
11 to 0	FractionY[11:0]	0	R/W	**Fraction for Y scaling**

This register holds the value used for up/down scaling in the y direction. Every mantissa.fraction line will be generated by bilinear interpolation. If the mantissa is ≥ 1 and the fraction > 0 the module will downscale in the y direction, if the mantissa is 0 and the fraction > 0 then the module will upscale in the y direction. The value in this register indicates the (number of lines captured per field)/(number of lines written to memory per field). Y scaling is disabled if both the mantissa and fraction are 0.

This module supports vertical scaling ratio ranging from 1/15 up to 3. Therefore the register value must be less than or equal to H'EFFF and must be greater than or equal to 0x0556.

To support real-time operation, the operation frequency, which is same as that of pixel bus clock, must be over 50 MHz when MantissaY.FractionY ≥ H'0800, and 66 MHz when MantissaY.FractionY ≥ H'0556.

X Scale Register (XS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MantissaX[3:0]				FractionX[11:0]											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 12	MantissaX[3:0]	0	R/W	**Mantissa for X scaling**
11 to 0	FractionX[11:0]	0	R/W	**Fraction for X scaling**

This register holds the value used for pixel selection in x direction downscaling. Every mantissa.fraction pixel will be generated from the multiphase filter. The value in this register determines (number of input pixels captured per line)/(number of pixels output to memory per line). X direction scaling is disabled if the mantissa is 0.

Coefficient Set Registers

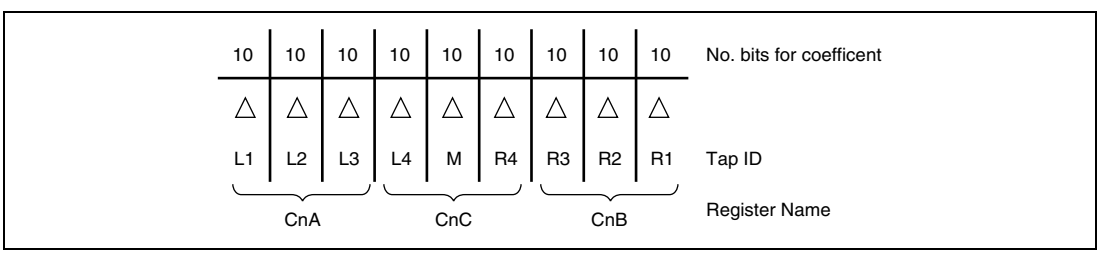


Figure 9.3 Coefficient Bit Sizes

The above diagram shows the number of bits used for each coefficient for the nine taps. Each of the eight coefficient sets comprise three 32-bit registers the packing for these three registers is as shown below. The most significant bit of each coefficient is sign bit.

Coefficient Set CnA (n = 1 to 8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			L1[9:0]										L2[9:6]			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L2[5:0]						L3[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	0	R	**Reserved**
29 to 20	L1[9:0]	0	R/W	
19 to 10	L2[9:0]	0	R/W	
9 to 0	L3[9:0]	0	R/W	

Coefficient Set CnB (n = 1 to 8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			R1[9:0]										R2[9:6]			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R2[5:0]						R3[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	0	R	**Reserved**
29 to 20	R1[9:0]	0	R/W	
19 to 10	R2[9:0]	0	R/W	
9 to 0	R3[9:0]	0	R/W	

Coefficient Set CnC (n = 1 to 8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			R4[9:0]										L4[9:6]			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	L4[5:0]						M[9:0]									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	0	R	**Reserved**
29 to 20	R4[9:0]	0	R/W	
19 to 10	L4[9:0]	0	R/W	
9 to 0	M[9:0]	0	R/W	

Where n is the coefficient set number.

9.4 Functional Description

The complete functionality is described by the following sub-functions:

- ITU-R BT.656 Interface
- Vertical Scaling.
- Horizontal Scaling.
- Size Clipping before/after scaling
- Color Space Conversion
- Dithering.
- Capture Mode
- Module Standby Mode

9.4.1 ITU-R BT.656 Interface

This module is capable of capturing video stream which is complied with ITU-R BT.656 specification. This module is also capable of correcting errors in timing reference codes (SAV/EAV). The timing reference code (SAV/EAV) for ITU-R BT.656 contains four protection bits, these bits are used to correct any 1-bit errors on the interface. If the CEE bit is set in the Interrupt Enable (IE) Register and the module cannot correct an error, then a interrupt will be

generated and the CES bit in the Interrupt Status (IS) Register will be also set. If the module can correct an error, no interrupt signal is generated.

When V-bit in SAV equals to 0, this module advances the line pointer which is used to store pixel data to graphic memory. And when F-bit in SAV/EAV is changed, the module controls the write pointer of frame buffers. Therefore SAV/EAV must have valid F-bit which complies with ITU-R BT.656 for correct operation.

If F-bit in SAV/EAV does not change correctly in a input video data stream, this module captures the video data until captured line reaches the line set by End Line Post-Clip (ELPoC) Register. After this line, no video data is captured unless the next F-bit change occurs. The register bits frame buffer status point at the last valid frame until the next F-bit change. The number of SAV with V-bit = 0 should be less than or equal to 288 between the last and the next coming F-bit change.

9.4.2 Vertical Scaling

Vertical scaling is performed in this module using bilinear interpolation. Both up and down scaling can be performed by setting the appropriate value in the Y Scale (YS) Register. The mantissa.fraction combination selects at which line position a new line will be generated. Setting both the fraction and mantissa in the Y Scale (YS) Register to 0 bypasses y scaling block. The line pointer has 4-bit accuracy in its position. The two adjacent lines are divided into sixteen segments, and the most significant four bits in the fraction part is used to point at the generated position.

Figure 9.4 is an example of how the upscaling is accomplished in this module. A line pointer is incremented by the mantissa.fraction value, from which a line position can be determined for the bilinear interpolation process.

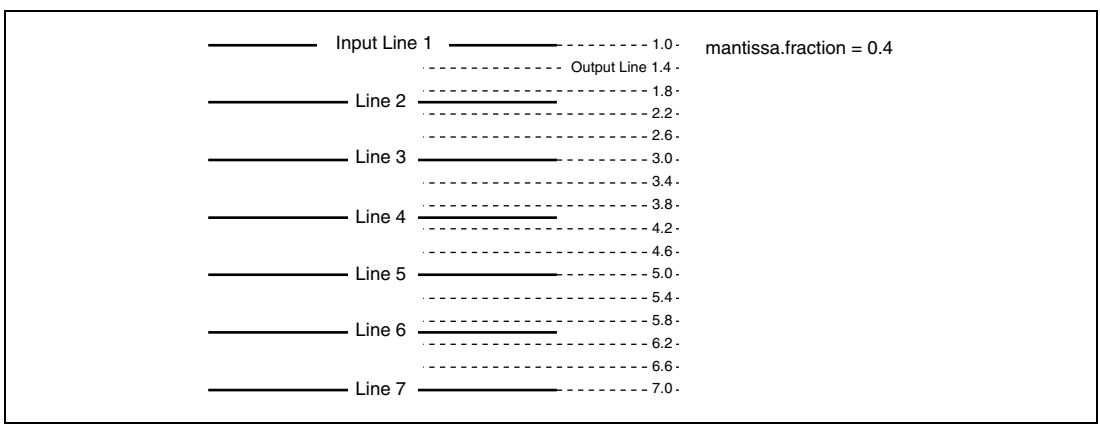


Figure 9.4 Vertical Up Scaling Example

Figure 9.4 shows down scaling as can be seen this is very similar to upscaling. This module is restricted to a maximum upscaling of 3.

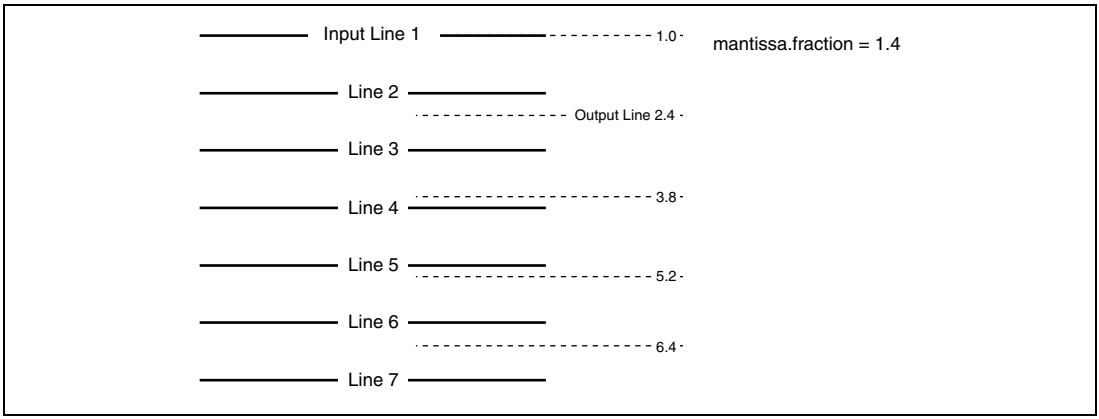


Figure 9.5 Vertical Down Scaling Example

The number of lines generated by this vertical scaling block is defined as follows.

$$Ny = \begin{cases} Int\left(\dfrac{4096 \times (ELPrC - SLPrC)}{4096 \times \text{Mantissa } Y + \text{Fraction } Y}\right) - 1, \\ \qquad \text{when } \{4096 \times (ELPrC - SLPrC)\}\ \%\ (4096 \times \text{Mantissa } Y + \text{Fraction } Y) = 0 \\ Int\left(\dfrac{4096 \times (ELPrC - SLPrC)}{4096 \times \text{Mantissa } Y + \text{Fraction } Y}\right), \text{otherwise} \end{cases}$$

where ELPrC and SLPrC are register value from End Line Pre-Clip (ELPrC) and Start Line Pre-Clip (SLPrC) Registers. And MantissaY and FractionY are register value from Y Scaling (YS) Register.

9.4.3 Horizontal Scaling

This module performs horizontal scaling using a nine tap multiphase filter. In the x direction only down scaling is supported. The mantissa.fraction combination set in the X Scale (XS) Register determines the pixel position at which a pixel is produced from the polyphase filter. The selected coefficient set is one of eight coefficient sets, which is determined by the output pixel position. Figure 9.6 shows an example with the mantissa.fraction set to 1.2. In this case, the coefficient set C2 is selected.

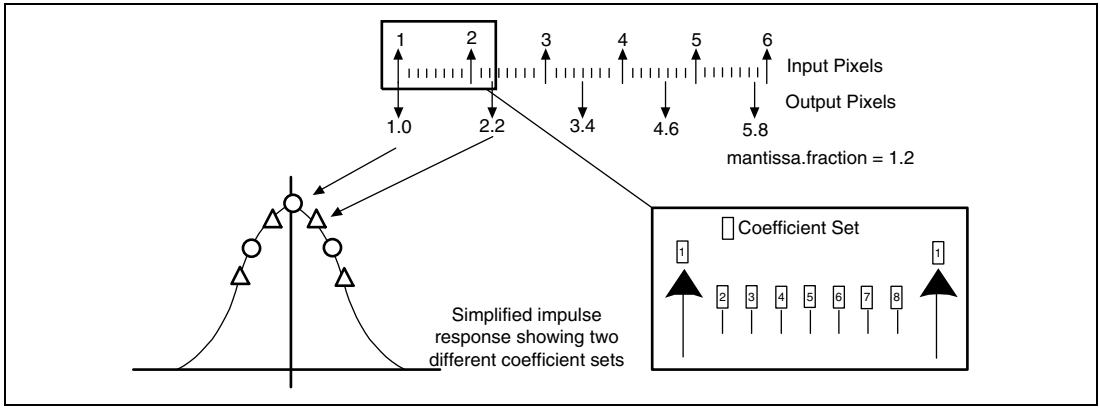


Figure 9.6 Pixel Position and Coefficient Set Selection

Figure 9.7 shows the examples of coefficient sets for different sub-sampled pixel position. Each coefficient set has nine coefficients. Total 72 coefficients are used for horizontal scaling. Each coefficient in a coefficient set has 10-bit width and MSB is sign bit. This horizontal scaling can be performed using the multi-phase filter as a single phase filter by loading the same coefficient set into all eight coefficient set registers.

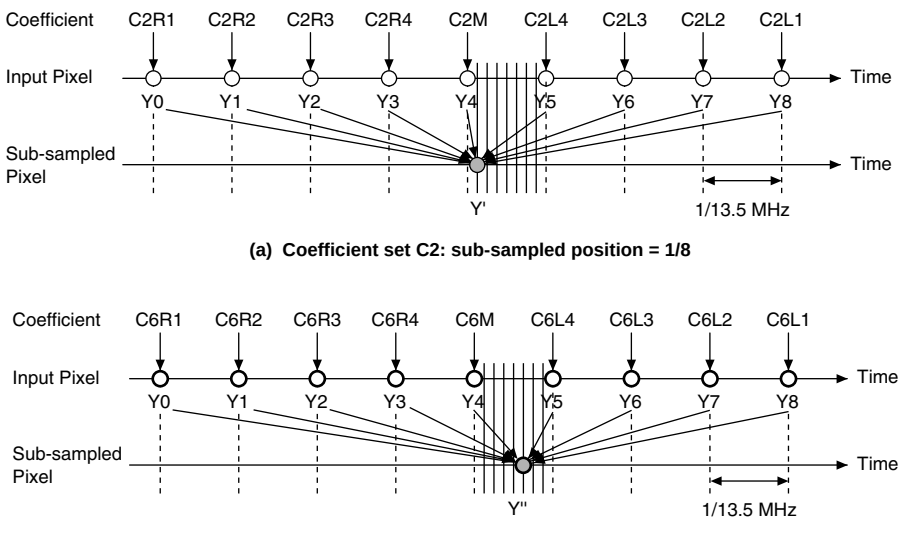


Figure 9.7 Examples of Coefficients in a Selected Coefficient Set

This scaling mechanism requires different coefficient values which depends on the scaling ratio. One of examples for selecting coefficient sets is shown as below. Coefficients, CnM, CnLi and CnRi (n = 1, 2, 3, ---, 8; i = 1, 2, 3 and 4), are determined by the following equations.

$$CnM = ß \cdot h(-(n-1))$$
$$CnRi = ß \cdot h(-(n-1) - 8(5-i))$$
$$CnLi = ß \cdot h(-(n-1) + 8(5-i))$$

$$h(t) = \begin{cases} \dfrac{\sin\left(\dfrac{\pi t}{T}\right)}{\dfrac{\pi t}{T}} \cdot \dfrac{\cos\left(\dfrac{\alpha\pi t}{T}\right)}{1 - \dfrac{4\alpha^2 t^2}{T^2}}, & \text{when } t \neq 0 \text{ and } \dfrac{4\alpha^2 t^2}{T^2} \neq 1 \\ \dfrac{\pi}{4} \cdot \dfrac{\sin\left(\dfrac{\pi}{2\alpha}\right)}{\dfrac{\pi}{2\alpha}}, & \text{when } t \neq 0 \text{ and } \dfrac{4\alpha^2 t^2}{T^2} = 1 \\ 1, & \text{when } t = 0 \end{cases}$$

$$T = 8 \times \text{Mantissa } X + \text{Fraction } X[11:9]$$

where MantissaX and FractionX are register value from X Scaling (XS) Register. The parameter is the normalization parameter. In the above equations, the function h(t) is known for showing raised cosine characteristics, and the value α ($0 < \alpha \leq 1$) is determined so that h(t) can be implemented with 9-tap filter.

For this rideo input module, $\alpha = 0.6$ should be used. Please refer to the sample program in clause 9.5 (section).

To obtain correct scaled images, each coefficient set must satisfy with the following equation for normalization. This can be executed by selecting proper β value in the above equation.

$$C_nM + \sum_{i=1}^{4} C_nR + \sum_{i=1}^{4} C_nL_i = 512$$

The example of values which satisfy the above equations is shown in the appendix of this block specification.

The number of pixels generated by this horizontal scaling block is defined as follows.

$$Ny = \begin{cases} Int\left(\dfrac{4096 \times (EPPrC - SPPrC + 1)}{4096 \times \text{Mantissa } X + \text{Fraction } X}\right), \\ \qquad \text{when } \{4096 \times (EPPrC - SPPrC + 1)\}\,\%\,(4096 \times \text{Mantissa } X + \text{Fraction } X) = 0 \\ Int\left(\dfrac{4096 \times (EPPrC - SPPrC + 1)}{4096 \times \text{Mantissa } X + \text{Fraction } X}\right) + 1, \text{ otherwise} \end{cases}$$

where EPPrC and SPPrC are register value from End Pixel Pre-Clip (EPPrC) and Start Pixel Pre-Clip (SPPrC) Registers. And MantissaX and FractionX are register value from X Scaling (XS) Register.

9.4.4 Size Clipping before/after Vertical or Horizontal Scaling

A rectangular region of the input video can be selected for grabbing prior to scaling using the Start Line Pre-Clip (SLPrC), End Line Pre-Clip (ELPrC), Start Pixel Pre-Clip (SPPrC) and End Pixel Pre-Clip (EPPrC) Registers. Once the image has been processed by the x and y direction scaling logic it can be clipped again using the Start Line Post-Clip (SLPoC), End Line Post-Clip (ELPoC), Start Pixel Post-Clip (SPPoC) and End Pixel Post-Clip (EPPoC) Registers. An example of this size clipping is shown in Figure 9.8.

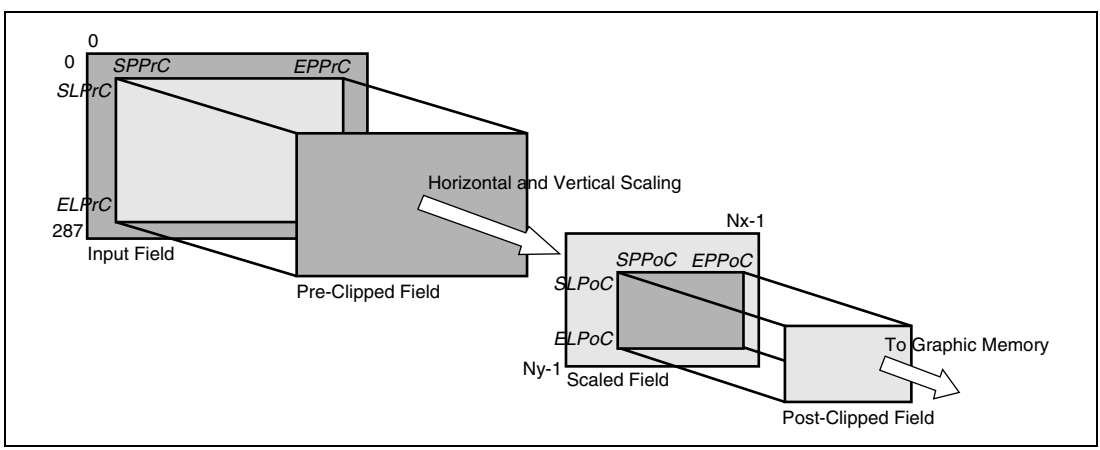


Figure 9.8 Selectable Grab Window

For all post-clipped lines, the length of each line written to memory is defined by the Image Stride (IS) Register, which can be greater than the post-clipped frame width but not less. The Image Stride (IS) Register is used such that at the end of each line of post-clipped frame written to memory the next line is written at the start address of the current line plus the Image Stride (IS) Register value.

9.4.5 Color Space Conversion

A 3x3 matrix is used to convert from YCbCr 4:2:2 into RGB 888 format. The matrix equation is

$$R' = 1.164(Y - 16) + 1.596(Cr - 128)$$
$$G' = 1.164(Y - 16) - 0.813(Cr - 128) - 0.392(Cb - 128)$$
$$B' = 1.164(Y - 16) + 2.017(Cb - 128)$$

Within the converter the range of *Y* is stretched from 16-235 to 0-255, also the range of *Cb* and *Cr* are stretched from 16-240 to 0-255. This stretching is built into the matrix equations shown above.

9.4.6 Dithering

Then dithering is performed to convert the RGB 888 to RGB 565 before writing to the Pixel Bus Buffer using an accumulated error mechanism as shown below:

$$R'_n[7:3] <= (R_n[7:0] + R_{n-1}[2:0]) >> 3$$
$$R'_n[7:2] <= (R_n[7:0] + R_{n-1}[2:0]) >> 2$$
$$R'_n[7:3] <= (R_n[7:0] + R_{n-1}[2:0]) >> 3$$

where (R'_n, G'_n, B'_n) = Output RGB565 pixel.
(R_n, G_n, B_n) = Input RGB888 pixel.
$(R_{n-1}, G_{n-1}, B_{n-1})$ = psuedorandom error, the least significant bits from the accumulated error.

9.4.7 Capture Mode

The module is capable of capturing frames in one of two modes, single frame capture and continuous frame capture. When operating in single frame capture mode setting the SC bit in the Frame Capture (FC) Register causes the next valid frame to be captured to the memory address set in the Memory Base 1 (MB1) Register.

When operating in continuous capture mode the module will capture and process all frames cycling through the addresses stored in the Memory Base 1-3 Registers to determine where in memory to place the captured frames. During continuous capture mode a indicator of the latest frame buffer to contain a full captured frame is output through the frame_buffer_status port of the module.

9.4.8 Module Standby Mode

This video input module allows clock gating to reduce power consumption. Both pixel bus clock and register bus clock can be gated. This module standby mode can be executed by controlling Clock Control 2 (CC2) Register in Power Control module.

To wake up the module, VI bit in Clock Control 2 (CC2) Register must be enabled. After enabling this bit, all access to video input module could be possible.

To power down the module, the following procedure is required.

1. Disable Module Enable (ME) bit in Main Control (MC) Register.
2. Wait until Capture Active (CA) bit in Module Status (MS) Register goes to 0.
3. Wait for the next field change. The register bit Field Interrupt Status 2 (FIS2) in Interrupt Status (IS) Register can be used for waiting the next field change.
4. Disable VI bit in Clock Control 2 (CC2) Register.

9.5 Example of Sample Program

The examples of sample program is described below for setting the video Input module.

- Initial setting of Video Input registers
- Function to Set the X direction filter coefficient

9.5.1 Example of Initial setting of Video Input registers

List 1 shows the example of initial setting routine for Video Input registers. In this example, NTSC (image size: 720 × 480) moving picture image is reduced to 360 × 240, and Field 1 and Field 2 of NTSC are taken in the different frame memories respectively, setVideoReg() is a function that sets the Video Input registers. The first armament of this function corresponds to the address of the Video Input register, and the second is the set value for that address.

List 1. Example of Video Input Initial Setting Routine

```
1.  setVideoReg (vi_mc, 0x0009);
2.  setVideoReg (vi_slprc, 0x0000);    // SLPrC = 0
3.  setVideoReg (vi_elprc, 0x00EF);    // ELPrC = 239
4.  setVideoReg (vi_spprc, 0x0000);    // SPPrC = 0
5.  setVideoReg (vi_epprc, 0x02CF);    // EPPoC = 719
6.  setVideoReg (vi_slpoc, 0x0000);    // SLPoC = 0
7.  setVideoReg (vi_elpoc, 0x00EF);    // ELPoC = 239
8.  setVideoReg (vi_sppoc, 0x0000);    // SPPoC = 0
9.  setVideoReg (vi_eppoc, 0x0167);    // EPPoC = 359
10. setVideoReg (vi_xs, 0x2000);
11. setVideoReg (vi_ys, 0x0000);
12. setVideoReg (vi_mb1, 0x00300000);
13. setVideoReg (vi_mb2, 0x00380000);
14. setVideoReg (vi_mb3, 0x00400000);
15. setVideoReg (vi_is, 0x0168);
16. setVideoReg (vi_ie, 0x0000);
17. setVideoReg (vi_ints, 0x0000);
18. setVideoReg (vi_si, 0x0001);
19. //======== video_in coeff registers ========
20. setVideoReg (vi_c1a, 0x000fa400);
21. setVideoReg (vi_c1b, 0x000fa400);
22. setVideoReg (vi_c1c, 0x09625902);
23. setVideoReg (vi_c2a, 0x00000000);
24. setVideoReg (vi_c2b, 0x00000000);
25. setVideoReg (vi_c2c, 0x00000000);
26. setVideoReg (vi_c3a, 0x00000000);
27. setVideoReg (vi_c3b, 0x00000000);
28. setVideoReg (vi_c3b, 0x00000000);
29. setVideoReg (vi_c4a, 0x00000000);
30. setVideoReg (vi_c4b, 0x00000000);
31. setVideoReg (vi_c4c, 0x00000000);
32. setVideoReg (vi_c5a, 0x00000000);
33. setVideoReg (vi_c5b, 0x00000000);
34. setVideoReg (vi_c5c, 0x00000000);
35. setVideoReg (vi_c6a, 0x00000000);
36. setVideoReg (vi_c6b, 0x00000000);
```

```
setVideoReg (vi_c6c, 0x00000000);
setVideoReg (vi_c7a, 0x00000000);
setVideoReg (vi_c7b, 0x00000000);
setVideoReg (vi_c7c, 0x00000000);
setVideoReg (vi_c8a, 0x00000000);
setVideoReg (vi_c8b, 0x00000000);
setVideoReg (vi_c8c, 0x00000000);
//======== Continuous capture starts ========
setVideoReg (vi_fc, 0x0002);
```

9.5.2 Function to Set the x Direction Filter Coefficient

The Video Input module employs the architecture that the number of filter taps is 9 for reduction to the X direction. In this architecture, filter coefficients to gain the satisfactory image quality differ according to the position of the pixel to be generated and the reduction rate of the image.

Generally, to gain high-quality image that is reduced, the frequency band is limited according to the reduction rate to prevent image ringing from being generated. Since the low pass filter generally requires the enormous number of taps to gain high-quality image, filter characteristics are required that can achieve this with the limited number of taps. The Rased Cosine function, which has such characteristics, is generally known. List 2 shows the program to get the filter coefficient appropriate to the Video Input module using this function.

The gen_tap function shown in List 2 uses creg[ ], xs and alpha100 as arguments. Filter coefficient creg[ ],xs can be set by substituting the values for arguments xs and alpha100. The table below gives the details on these arguments.

arguments	Type	Description
creg[ ][ ]	Unsigned long	The set values of can/CnB/CnC (N = 0 to 8) registers. The first affix of the array represents the position of the pixel to be generated. The affix value is 0 for can, 1 for CnB, and 2 for CnC. For example, creg[5][2] represents the C6C register.
xs	Unsigned long	The value of the X scale register (XS) of the Video Input module. Set the value reneging from H'1000 to H'FFFF with positive numbers.
alpha100	Int	The value 100 times the <value (roll off rate) of the Raised Cosine function. Set the value ranging between 0 ≤ alpha100 ≤ 100 with positive numbers.

The closer to 0 the<value (roll off rate) of the Raised Cosine function is, the more appropriate the characteristics of the Low Pass filter are. Since the Video Input module, however, uses the filter with the limited number of taps (9 taps), the appropriate characteristics cannot be achieved, which

causes a pseudo outline. The closer to 1 the α is, the frequency characteristics will be gently sloped closer to the cut off of the Low Pass filter, which causes dim image.

Consequently, set the <value to 0.6 (alpha100 = 60).

List 2. Function to Set the X direction Filter Coefficient

```
1.  #include <stdio.h>
2.  #include <math.h>
3.
4.  #define MAXBSIZ 255
5.  #define TAPSIDE 4
6.  #define TAPNUM (TAPSIDE*2+1)
7.  #define TAPDIV 8
8.
9.  #define CLIP 512
10. #define SCLE 512
11.
12. // control option
13. #define COEFFCLIP 1
14.
15. // #define DISPLAY
16. // doublecoeff[TAPNUM*TAPDIV];
17. // intc[TAPDIV][TAPNUM];
18.
19. char
20. gen_tap( unsigned long creg[][3], unsigned long xs, int alpha100 )
21. {
22. charoverflag;
23.
24. inti, iofst j, k;
25.
26. doublex, T, tT;
27. doublecoeff[TAPNUM*TAPDIV];
28. intc[TAPDIV][TAPNUM]
29. doublecoeff_diff;
30. doublesum;
31. intsumd;
32. doublealpha;
```

```
33.
34. // Raised Cosine Charactaristics Mode1
35. // 0.0 <= alpha <= 1.0
36. // 0   <= alpha100 <= 100
37. alpha = (double)(alpha100)/(double)100;
38.
39. //
40. // calc coeff (double order)
41. //
42. T = (double)((unsigned long)(xs>>9));
43. for ( j = 0; j < TAPDIV; j++) {
44. for ( I = -TAPSIDE; I <= TAPSIDE; I++) {
45. k = -j + I * TAPDIV;
46. iofst = k + (TAPSIDE+1)*TAPDIV-1;
47. if ( alpha100 == 0 ) {
48. if ( k == 0 ) coeff[iofst] = 1.0;
49. else {
50. x = (double)M_PI*(double)k/T;
51. coeff[iofst] = sin( x ) / x;
52. }
53. }
54. else {
55. if ( k == 0 ) coeff[iofst] = 1.0;
56. else {
57. x = (double)M_PI* (double)k/T;
58. sum = (double)4.0 * (double)(alpha100*alpha100) * (double)(k*k);
59. tT = (double)1.0 – sum/T/T/(double)10000;
60. /*                                                          */
61. /* if tT == 0 -> lim f(tT) = sin(PI/2alpha)/PI/2alpha * PI/4 */
62. /*              tT -> 0                                     */
63. if ( sum == T*T* (double)10000 ) {
64. coeff[iofst] = sin((double)M_PI/(2.0*alpha))/((double)M_PI/
    (2.0*alpha)) * (double)M_PI/4.0;
65. }
66. else {
67. coeff[iofst] = (sin(x)/x) * (cos(alpha*x)/tT);
68. }
69. }
```

```
70. }
71. }
72. }
73.
74. //
75. // trans : double -> integer
76. //
77. overflag = 0;
78. for ( j = 0; j < TAPDIV; j++) {
79. sum = 0.0;
80. for ( i = -TAPSIDE; i <= TAPSIDE; i++) {
81. k = -j + i * 8;
82. k += (TAPSIDE+1) *TAPDIV-1;
83. sum += coeff[k];
84. }
85.
86. coeff_diff = 0.0;
87. sumd = 0;
88. for ( i = -TAPSIDE; i < ((j<=(TAPDIV/2))?0:1); i++) {
89. k = -j + i * 8;
90. k += (TAPSIDE+1)*TAPDIV-1;
91. iofst = i+TAPSIDE;
92. c[j][iofst] = (int)( (double)SCLE*(coeff[k]/sum) + coeff_diff );
93.
94. // for jitter
95. coeff_diff = (double)SCLE*(coeff[k]/sum)) – (double)c[j][iofst];
96. sumd += c[j][iofst];
97. if( c[j][iofst] >= CLIP || c[j][iofst] < -CLIP ) {
98. overflag = 1;
99. }
100.}
101.coeff_diff = 0.0;
102.for ( i = TAPSIDE; i >= ((j<=(TAPDIV/2))?0:1); i-- ) {
103.k = -j + i * 8;
104.k += (TAPSIDE+1)*TAPDIV-1;
105.iofst = i+TAPSIDE;
106.c[j][iofst] = (int)( (double)SCLE*(coeff[k]/sum) + coeff_diff );
```

```
107.
108.// for jitter
109.if ( i != ((j<=(TAPDIV/2))?0:1) ) {
110.coeff_diff = ((double)SCLE*(coeff[k]/sum)) – (double)c[j][iofst];
111.sumd += c[j][iofst];
112.}
113.else {
114.c[j][iofst] = SCLE – sumd;
115.}
116.if( c[j][iofst] >= CLIP || c[j][iofst] < -CLIP ) {
117.overflag = 1;
118.}
119.}
120.
121.//
122.// when coeff[center] == CLIP && coeff[else] == 0 then coeff
   [center]--
123.//
124.if ( COEFFCLIP && overflag == 1) {
125.for ( i = -TAPSIDE; i <= TAPSIDE; i++) {
126.iofst = i+TAPSIDE;
127.if( c[j][iofst] >= CLIP ) {
128.c[j][iofst] = CLIP-1;
129.}
130.else if( c[j][iofst] < -CLIP ) {
131.c[j][iofst] = -CLIP;
132.}
133.}
134.overflag = 0;
135.}
136.}
137.
138.for ( j = 0; j < TAPDIV; j++) {
139.creg[j][0] = ((unsigned long)c[j][8]&0x3ff)<<20;
140.creg[j][0] |= ((unsigned long)c[j][7]&0x3ff)<<10;
141.creg[j][0] |= ((unsigned long)c[j][6]&0x3ff);
142.
143.creg[j][1] = ((unsigned long)c[j][0]&0x3ff)<<20;
```

```
144.creg[j][1] |= ((unsigned long)c[j][1]&0x3ff)<<10;
145.creg[j][1] |= ((unsigned long)c[j][2]&0x3ff);
146.
147.creg[j][2] = ((unsigned long)c[j][3]&0x3ff)<<20;
148.creg[j][2] |= ((unsigned long)c[j][5]&0x3ff)<<10;
149.creg[j][2] |= ((unsigned long)c[j][4]&0x3ff);
150.}
151.
152.return( overflag );
153.
154.}
```

Section 10 Display Out Module

10.1 Overview

The display output module is responsible for the display control. It support digital TFT type displays up to a maximum resolution of 1024 × 768*. This block supports the mixing of two frames and the inclusion of two hardware cursors. All sizes and timing controls are fully programmable.

The two frames are defined as the primary and secondary frames. The secondary frame can be alpha blended with the primary window or be opaque. It can also be configured to display video data as either full screen or within a sub-window (Picture in Picture).

Note: * The display size counter can handle as large as 1024 × 768, but this does not mean that HD64404 can display 1024 × 768 in all possible configurations and conditions. The dot clock frequency is up to 50MHz. Please see Section 1 Overview.

10.1.1 Features

- Two independent video/graphics planes.
- Picture in Picture for background plane.
- Alpha Blending for mixing the planes.
- Fully Programmable display size and sync signal generator.
- Two 64 × 64 pixel 8 bit hardware cursors.
- Binary Compatible with most Q2SD display functions.

10.1.2 Block Diagram

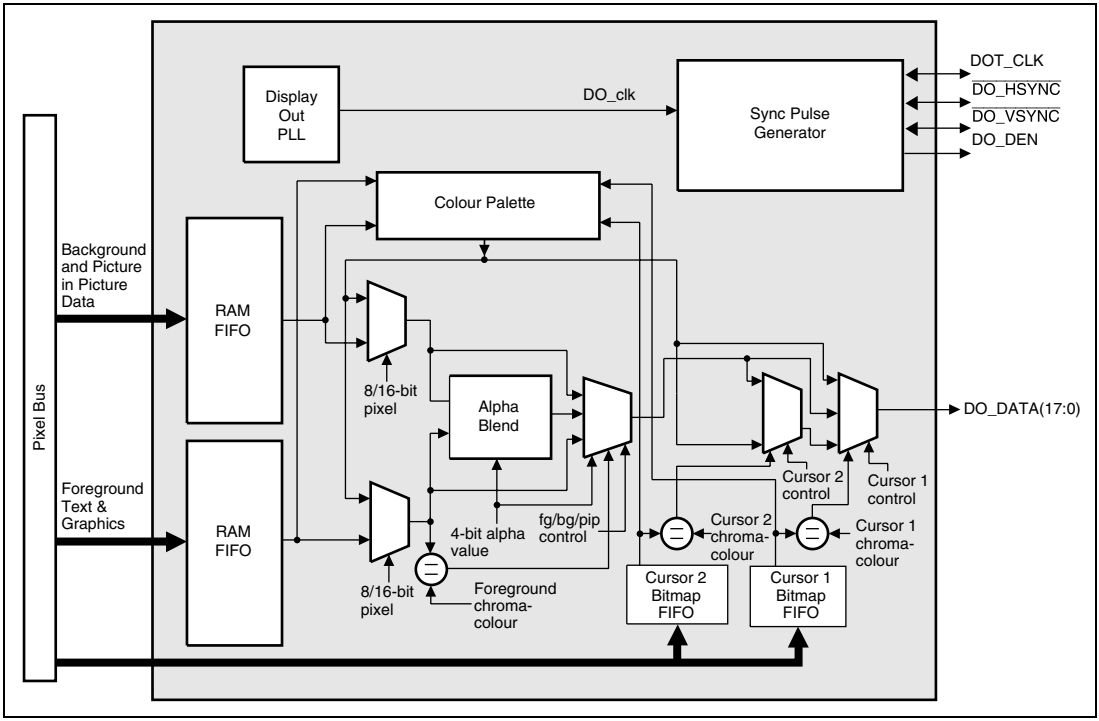


Figure 10.1 Block Diagram

10.2 Interfaces

10.2.1 Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 10.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From	Synchronization to Clocks
DOT_CLK	1	In/Out	Display Clock	TFT display	—
DO_DATA (17:0)	18	Out	18 bit RGB Display Data RGB data is mapped to this port as: Bits 17:12 = red(5:0) Bits 11:6 = green(5:0) Bits 5:0 = blue(5:0) (*)	TFT display	dot_clk
DO_HSYNC	1	In/Out	Horizontal sync or Combined Sync or external H-sync	TFT display	dot_clk
DO_VSYNC	1	In/Out	Vertical sync or external V-sync	TFT display	dot_clk
DO_DEN	1	Out	Display enable. Shows when display data active.	TFT display	dot_clk

Note: * details of RGB mapping between 18-bit data and 16-bit data can be seen in the Functional Description: General Functionality section.

10.2.2 Software Interfaces

The registers accessible by the software are listed in the following table:

Access is forbidden to address other than these listed below.

All registers listed below are long word addresses.

Table 10.2 Register List

Address (Bytes)	Register Name	Abbreviation	Access Size
H'4000	System Control Reg*	DO_SYSR	32
H'4004	Status Reg*	DO_SR	32
H'4008	Status Register Clear*	DO_SRCR	32
H'400C	Interrupt Enable*	DO_IER	32
H'4014	Display Mode Reg*	DO_DSMR	32
H'40AC	Display Mode Reg2*	DO_DSMR2	32
H'4018	Rendering Mode Reg*	DO_REMR	32
H'4154	FG Ext Rendering Mode	DO_DUFG	32
H'4158	BG Ext Rendering Mode	DO_DUBG	32
H'415C	PIP Ext Rendering Mode	DO_DUW	32
H'4020	Display Size Reg X*	DO_DSX	32
H'4024	Display Size Reg Y*	DO_DSY	32
H'4028	Display Start Address 0H Q2SD*	DO_DSAR0H	32
H'4190	Display Start Address 0H GE Tiled + Linear	DO_DSAR0H	32
H'41C0	Display Start Address 0L Q2SD	DO_DSAR0L	32
H'4194	Display Start Address 0L GE Tiled + Linear	DO_DSAR0L	32
H'402C	Display Start Address 1H Q2SD*	DO_DSAR1H	32
H'4198	Display Start Address 1H GE Tiled + Linear	DO_DSAR1H	32
H'41C4	Display Start Address 1L Q2SD	DO_DSAR1L	32

Address (Bytes)	Register Name	Abbreviation	Access Size
H'419C	Display Start Address 1L GE Tiled + Linear	DO_DSAR1L	32
H'404C	Horizontal Display Start Position*	DO_HDS	32
H'4050	Horizontal Display End Position*	DO_HDE	32
H'4054	Vertical Display Start Position*	DO_VDS	32
H'4058	Vertical Display End Position*	DO_VDE	32
H'405C	Horizontal Sync Pulse Width*	DO_HSWR	32
H'4060	Horizontal Scan Cycle*	DO_HCR	32
H'4064	Vertical Sync Position*	DO_VSPR	32
H'4068	Vertical Scan Cycle*	DO_VCR	32
H'406C	Display Off Output H*	DO_DOORH	32
H'4070	Display Off Output L*	DO_DOORL	32
H'40A4	Equalising Pulse Width*	DO_EQW	32
H'40A8	Separation Width*	DO_SPW	32
H'40B0	PIP Horizontal Display Start Position*	DO_HVP	32
H'40B4	PIP Vertical Display Start Position*	DO_VVP	32
H'40C4	PIP Start Address Register 0H*	DO_VSAR0H	32
H'40C8	PIP Start Address Register 0L*	DO_VSAR0L	32
H'40CC	PIP Start Address Register 1H*	DO_VSAR1H	32
H'40D0	PIP Start Address Register 1L*	DO_VSAR1L	32
H'40D4	PIP Start Address Register 2H*	DO_VSAR2H	32
H'40D8	PIP Start Address Register 2L*	DO_VSAR2L	32
H'40DC	PIP Window Size X*	DO_VSIZEX	32
H'40E0	PIP Window Size Y*	DO_VSIZEY	32
H'40E4	Video Incorporation Mode*	DO_VIMR	32
H'40E8	Cursor 1 Horizontal Display Start Position*	DO_HCS1	32
H'40EC	Cursor 1 Vertical Display Start Position*	DO_VCS1	32
H'40F0	Cursor 2 Horizontal Display Start Position*	DO_HCS2	32
H'40F4	Cursor 2 Vertical Display Start Position*	DO_VCS2	32
H'40F8	Cursor 1 Start Address*	DO_CSARL1	32
H'40FC	Cursor 2 Start Address*	DO_CSARL2	32
H'41B0	Background Area Start Address A	DO_LBGSA	32
H'41B4	Background Area Start Address B	DO_RBGSA	32

Address (Bytes)	Register Name	Abbreviation	Access Size
H'4098	Background A Start Position X*	DO_LBGSX	32
H'409C	Background A Start Position Y*	DO_LBGSY	32
H'4220	Background B Start Position X	DO_RBGSX	32
H'4224	Background B Start Position Y	DO_RBGSY	32
H'4234	Vertical Wraparound Size	DO_WRPY	32
H'4244	Display Blending 1	DO_DBR1	32
H'4248	Display Blending 2	DO_DBR2	32
H'4324	Foreground Transparent Color	DO_TRNFG	32
H'4328	Cursor 1 Transparent Color	DO_TRNC1	32
H'432C	Cursor 2 Transparent Color	DO_TRNC2	32
H'43C8	Display Out Extension Control Reg	DO_ECR	32
H'43CC	Line Interrupt Register	DO_LIR	32
H'43D0	Display Out PLL Reg	DO_PLL	32
H'4400 to H'4BF8	Color Palette H × 256*	CP000H to CP255H	32
H'4404 to H'4BFC	Color Palette L × 256*	CP000L to CP255L	32
H'43D4	Color Palette Read Register H	CPRRH	32
H'43D8	Color Palette Read Register L	CPRRL	32

Notes: All registers marked * are Q2SD binary compatible.

Any registers that are not defined as double buffered within their register description, should only written to during the V-blank period once the dot clock has been enabled.

All reserved or unused bits do not have a guaranteed value when read.

10.2.3 Register Description

Legends for register description:

Initial value:	Register value after reset
—:	Undefined value
R/W:	Read and Write, write value can be read.
R:	Read only, for write always 0 write
R/WC0:	Read and Write, 0 write clear, 1 write is ignored
R/WC1:	Read and Write, 1 write clear, 0 write is ignored
W:	Write only, Read prohibited. If reserved, write always 0.
—/W:	Write only, Read value undefined.

Display Out System Control Register * (DO_SYSR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		DRES	DEN				DC									Q2SD
Initial:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	0	R	**Reserved**
14	DRES	1	R/W	**Display Reset(DRES), Display Enable (DEN)**
13	DEN	0	R/W	00: Display operation is started. However, DRES cannot be set to 0 while RESET pin is LOW. When HD64404 is started from the initial status, DRES should be set to 0 after every control register is set. While DEN=0, the data output to the display is that contained in the DO_DOORH/L Registers. 01: Display operation is started. However, DRES and DEN cannot be set to 0 and 1 respectively while RESET pin is LOW. When HD64404 is started from the initial status, DRES and DEN should be set to 0 and 1 respectively after every control register is set. While DEN=1, and data stored in the UM is output to the Display data output pins from the next frame. 10: Display operation is not performed. (1) Display data has all '0' output. (2) Status register (DO_SR)/TV Sync Error (TVR) is cleared to 0. (3) Status register (DO_SR)/Vertical Blanking Flag (VBK) is cleared to 0. (4) Status register (DO_SR)/Line Interrupt Status (LIS) is cleared to 0. 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	—	0	R	**Reserved**
9	DC	0	R/W	**Display Area Change (DC)** Writing '1' to the DC bit will cause the frame buffer to be changed from FB0 to FB1 or vice versa at the end of the current frame. This bit is cleared automatically to '0' after a frame buffer switch. This bit is double buffered and will only cause the buffer to change after vblank start. After reset, the first buffer displayed is FB0. When compared with the Q2SD specification, only the "Manual Change Mode" is available.
8 to 1	—	0	R	**Reserved**
0	Q2SD	0	R/W	**Q2SD Compatibility Bit (Q2SD)** When this bit is set to '1', the display out module will operate in Q2SD compatibility mode, when set to '0' the extended functionality of the Graphics Engine (GE) mode will be available.

Display Out Status Register * (DO_SR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated. Status bits in this register can be cleared by writing '0' to the bit. Writing '1' will have no effect. This function is additional to the Q2SD spec and these bits can also be cleared using the DO_SRCR register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVR				VBK			DBF		LIS	VBA					
Initial:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W	R / WC0	R	R	R	R / WC0	R	R	R / WC0	R	R / WC0	R / WC0	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	TVR	0	R/WC0	**TV Sync Error (TVR)** This bit is set to '1' to indicate that a rise in ext_vsync has not been detected within the cycle time set in the Vertical Scan Cycle Register. This bit is only valid when bits TVM1&0 in the DSMR Register are set to TV sync mode.
14 to 12	—	0	R	**Reserved**
11	VBK	0	R/WC0	**Vertical Blanking Flag (VBK)** This bit is set to '1' at the start of the V-blank interval. Please see figure 10.3. This bit can also be cleared by writing to the VBCL bit in the SRCR. This bit will be duplicated in the Renderer and Display Out module. This bit is passed to the Renderer module as a side band signal.
10, 9	—	0	R	**Reserved**
8	DBF	0	R/WC0	**Display Buffer Frame (DBF)** When this bit is set to '0', register DSAR0 is used as the display start address, when set to '1' register DSAR1 is used as the display start address.
7	—	0	R	**Reserved**
6	LIS	0	R/WC0	**Line Interrupt Status (LIS)** This bit gives the status of the line interrupt. This bit can also be cleared by writing to the LICL bit in the SRCR. This bit is an extension to the Q2SD spec. Please see DO_LIR register.
5	VBA	1	R/WC0	**V-blank active (VBA)** This bit is set to 1 during the period blank is active, this bit is read only. Please see figure 10.3. This bit is an extension to the Q2SD spec.
4 to 0	—	0	R	**Reserved**

Display Out Status Register Clear Register* (DO_SRCR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated. This register is write only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVCL				VBCL					LICL						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R	R	R	R/WC1	R	R	R	R	R/WC1	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	TVCL	0	R/WC1	**TV Sync Error Flag Clear (TVCL)** When '1' is written to this bit, TVR bit is cleared in the SR Register.
14 to 12	—	0	R	**Reserved**
11	VBCL	0	R/WC1	**Vertical Blanking Flag Clear (VBCL)** When '1' is written to this bit, VBK bit is cleared in the SR Register. This bit will be duplicated in the Renderer and Display Out module.
10 to 7	—	0	R	**Reserved**
6	LICL	0	R/WC1	**Line Interrupt Status Clear (LICL)** When '1' is written to this bit, LIS bit is cleared in the SR Register. This bit is an extension to the Q2SD spec.
5 to 0	—	0	R	**Reserved**

Display Out Interrupt Enable Register* (DO_IER)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TVE				VBE					LIE						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R	R	R	R	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	TVE	0	R/W	**TV Sync Error Enable (TVE)** This bit is set to '1' to enable an interrupt on the DO_irq when the TVR bit is set in the SR Register. When set to '0' this interrupt is disabled.
14 to 12	—	0	R	**Reserved**
11	VBE	0	R/W	**Vertical Blanking Enable (VBE)** This bit is set to '1' to enable an interrupt on DO_irq when the VBK bit is set in the SR Register. When set to '0' this interrupt is disabled.
10 to 7	—	0	R	**Reserved**
6	LIE	0	R/W	**Line Interrupt Enable (LIE)** This bit is set to '1' to enable an interrupt on the DO_irq when the LIS bit is set in the SR Register. When set to '0' this interrupt is disabled. This bit is an extension to the Q2SD spec.
5 to 0	—	0	R	**Reserved**

Display Mode Register * (DO_DSMR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FLT	LWYS	RWYS	YSD	LWRP	LBG	RWRP	RBG	TVM1	TVM0						
Initial:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	FLT	0	R/W	**Filter Mode (FLT)** In Q2SD mode (SYSR bit 0 = '1'), when setting this bit to '0', the foreground (plane1) and background (plane2) are output to the display as the respective screens. When set to '1', the background and foreground are 50/50 alpha blended together (averaged) and displayed as the foreground. When SYSR bit 0 = '0', (GE mode) this bit has no effect.
14	LWYS	0	R/W	**Background A Screen Wraparound Y Size (LWYS)** When set to '0' the wrap around size for the Background A area is set to 512 pixels, when set to '1' the wrap around size is set to 1024 pixels. This bit is only valid in GE mode (SYSR bit 0 = '0') and background memory mode is set to tiled (DO_ECR bit 1 = '0').
13	RWYS	0	R/W	**Background B Screen Wraparound Y Size(RWYS)** When set to '0' the wrap around size for the Background B area is set to 512 pixels, when set to '1' the wrap around size is set to 1024 pixels. This bit is only valid in GE mode (SYSR bit 0 = '0') and background memory mode is set to tiled (DO_ECR bit 1 = '0').

Bit	Bit Name	Initial Value	R/W	Description
12	YSD	0	R/W	**Y-Size Disable (YSD)** When set to '1' the value of WRPY is independent of mode (Q2SD or Tiled) and the value for Y direction wrap size is taken from the WRPY register. When set to '0' (default) the value for the Y direction wrap size is dependent on the mode of operation. e.g. in Q2SD mode the WRPY value is fixed to 512.
11	LWRP	0	R/W	**Background A Screen Wraparound Configuration (LWRP)** When set to '1', the background A wraparound function is enabled. When set to '0', the background A wrap is disabled. When in Q2SD mode, this bit will control the background wrap function.
10	LBG	0	R/W	**Background A Screen Combination (LBG)** When set to '1', the background A is enabled. When set to '0', the background A plane is disabled and not output to the display. Should both A and B background combinations be enabled, only the background A will be displayed. When in Q2SD mode, this bit will control the background combination.
9	RWRP	0	R/W	**Background B Screen Wraparound Configuration (RWRP)** When set to '1', the B background wraparound function is enabled. When set to '0', the B background wrap is disabled.
8	RBG	0	R/W	**Background B Screen Combination (RBG)** When set to '1', the background B is enabled. When set to '0', the B background plane is disabled and not output to the display. Should both A and B background combinations be enabled, only the background A will be displayed.

Bit	Bit Name	Initial Value	R/W	Description
7	TVM1	1	R/W	**TV Sync Mode (TVM1, TVM0)**
6	TVM0	0	R/W	00: Master Mode, Sync Pulse Generator generates all synchronisation signals and DOT_CLK output. 01: Synchronisation Switching mode, This is used when switching between master and TV sync. In this mode all sync pins are outputs and the dot clk is stopped high. 10: TV Sync Mode, DOT_CLK, ex_vsync and ex_hsync are inputs. 11: External DOT_CLK mode, In this mode the DOT_CLK will be supplied externally but the SPG used to generate all other sync signals. This mode is an extension to the Q2SD spec.
5 to 0	—	0	R	**Reserved** For comparison with the Q2SD spec, Scan Mode bits 5:4, are reserved because only non-interlaced mode is supported.

Display Mode Register2* (DO_DSMR2)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					PRI2		HDIS		CSY1	CSY0	PRI		FBD	CE2	CE1	VWE
Initial:	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11	PRI2	0	R/W	**Window Priority, (PRI2,PRI)**
5	PRI	0	R/W	These bits specify the Window priority. Pixels are displayed from the plane with the highest priority unless the pixel in that plane is specified as transparent, when the pixel from the next highest priority frame will be displayed. 00: Screen Priority order is: cursor1, cursor2, foreground, PIP, background. 01: Setting prohibited. 10: Screen Priority order is: cursor1, foreground, PIP, cursor2, background. 11: Screen Priority order is: , foreground, PIP, cursor1, cursor2, background.
10	—	0	R	**Reserved**
9	HDIS	0	R/W	**Foreground Screen Start (HDIS)** When set to '0' the foreground screen1 starts at X = '0'.When set to '1' and a 1024 pixel memory width is used, the foreground screen1 starts at X = 512.
8	—	0	R	**Reserved**
7	CSY1	0	R/W	**CSYNC Mode (CSY1, CSY0)**
6	CSY0	1	R/W	These bits specify the CSYNC display output mode. 00: CSYNC is determined from Vsync XOR Hsync. CSYNC is output from the hsync/csync pin. 01: HSYNC is output from the hsync/csync pin. This is an extension to the Q2SD spec and the new default value. 10: Equalising pulses are output in 3 raster period from fall of vsync. Separation in next 3 raster period, equalising pulses in next 3 raster period, and hsync waveform in other periods. 11: Equalising pulses are output in 2.5 raster period starting 0.5 raster after fall of vsync. Separation in next 2.5 raster period, equalising pulses in next 2.5 raster period, and hsync waveform in other periods.
4	—	0	R	**Reserved**

Bit	Bit Name	Initial Value	R/W	Description
3	FBD	1	R/W	**Foreground Disable (FBD)** When set to '0' the foreground (plane1) is output to the display. When set to '1' the foreground is disabled (default value). This bit is double buffered and only updates at the start of v-blank.
2	CE2	0	R/W	**Cursor2 Enable (CE2)** When set to '1' cursor2 is output to the display. When set to '0' cursor2 is disabled. This bit is double buffered and only updates at the start of v-blank.
1	CE1	0	R/W	**Cursor1 Enable (CE1)** When set to '1' cursor1 is output to the display. When set to '0' cursor1 is disabled. This bit is double buffered and only updates at the start of v-blank.
0	VWE	0	R/W	**PIP Window Enable (VWE)** When set to '1' the PIP window is output to the display. When set to '0' the PIP window is disabled. This bit is double buffered and only updates at the start of v-blank.

Rendering Mode Register * (DO_REMR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		EREM								MWX					GBM1	GBM0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 15	—	0	R	**Reserved**
14	EREM	0	R/W	**Extended Rendering Mode Register Enable (EREM)** When this bit is set to '1', the values for MWX and GBM are taken from the Extended Rendering Mode Registers. When set to '0', the value for MWX is taken from bit 6, and the value for foreground and background GBM is taken from bits 1:0. The PIP GBM setting is fixed to 16 bits per pixel when EREM is set to '0'. This bit should always be set to '0' in Q2SD mode.
13 to 7	—	0	R	**Reserved**
6	MWX	0	R/W	**Memory Width (MWX)** These bits specify the X direction logical co-ordinate space (stride scaled by bits/pixel) of the SDRAM. When set to 0 the stride size in pixels is set to 512 pixels, when set to 1 the stride size in pixels is 1024 pixels. This bit will be duplicated in the Renderer and Display Out module. The bits in the display out module will be write only to avoid bus conflicts. When the EREM bit is set to '1' the values held in these bits will be ignored.
5 to 2	—	0	R	**Reserved**
1	GBM1	0	R/W	**Graphics Bit Mode (GBM1, GBM0)**
0	GBM0	0	R/W	These bits specify the configuration (bits per pixel) of the display data.

Bit 1 GBM1	Bit 0 GBM0	Foreground Bit Configuration	Background Bit Configuration
0	0	8 bits/pixel	8 bits/pixel
0	1	16 bits/pixel	16 bits/pixel
1	0	8 bits/pixel	16 bits/pixel
1	1	16 bits/pixel	8 bits/pixel

Note: When the EREM bit is set to '1' the values held in these bits will be ignored.

Foreground Extended Rendering Mode (DO_DUFG)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			MWXOFS						MWX		ALMWX			MWXD		GBM
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved**
13 to 8	MWXOFS	0	R/W	**Memory Width Offset (MWXOFS)** These bits specify the foreground stride offset measured in horizontal pixels. The total stride in pixels is set by MWXOFS plus MWX. These bits are only valid when foreground is set to GE Linear mode (DO_SYSR bit 0 = '0' and DO_ECR bit 0 = '1'). **Bit** — **Offset Value to be added to MWX** 8 — +32 pixels 9 — +64 pixels 10 — +128 pixels 11 — +256 pixels 12 — +512 pixels (this bit is not valid when MWX = 512 or the MWXD bit = '1'). 13 — +1024 pixels (this bit is not valid when MWX = 512 or the MWXD bit = '1').

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MWX	0	R/W	**Memory Width (MWX)** These bits specify the foreground stride measured in horizontal pixels

Bit 7 MWX1	Bit 6 MWX0	X direction logical co-ordinate space
0	0	512 pixels
0	1	1024 pixels
1	0	2048 pixels (not valid in Tiled mode, DO_ECR bit 0 = '0')
1	1	Reserved

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	ALMWX	0	R/W	**Additional Linear Memory Width Offset (ALMWX)** These bits specify the foreground stride additional offset measured in horizontal pixels. These bits are valid when the foreground is set to linear mode (DO_ECR bit 0 = '1'). When valid the total stride in pixels is set by ALMWX + MWXOFS + MWX and allow the memory width to be set to 4 pixel aligned. When the foreground is set to tiled mode these bits are ignored.

Bit	Offset Value to be added to MWX
3	+4 pixels
4	+8 pixels
5	+16 pixels

Bit	Bit Name	Initial Value	R/W	Description
2	MWXD	0	R/W	**MWX Disable (MWXD)** This bit disables the data in the MWX bits 7:6 to allow a memory width less than 512 pixels. When set to '1' the MWX bits 7:6 are ignored and just the data in the Memory Width Offset bits are used to calculate the memory width (the value of MWX is zero). This bit is only valid when the foreground is set to GE Linear mode.
1	—	0	R	**Reserved**
0	GBM	0	R/W	**Graphics Bit Mode (GBM)** This bit specifies the configuration (bits per pixel) of the foreground display data. When set to '0' the foreground data is 8 bits/pixel. When set to '1' the foreground data is 16 bits/pixel.

Background Extended Rendering Mode (DO_DUBG)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			MWXOFS						MWX		ALMWX			MWXD	GBMR	GBML
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved**
13 to 8	MWXOFS	0	R/W	**Memory Width Offset (MWXOFS)** These bits specify the background stride offset measured in horizontal pixels. The total stride in pixels is set by MWXOFS plus MWX. These bits are only valid when background is set to GE Linear mode (DO_SYSR bit 0 = '0' and DO_ECR bit 1 = '1').

Bit	Offset Value to be added to MWX
8	+32 pixels
9	+64 pixels
10	+128 pixels
11	+256 pixels
12	+512 pixels (this bit is not valid when MWX = 512 or the MWXD bit = '1').
13	+1024 pixels (this bit is not valid when MWX = 512 or the MWXD bit = '1').

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MWX	0	R/W	**Memory Width (MWX)** These bits specify the background stride measured in horizontal pixels

Bit 7 MWX1	Bit 6 MWX0	X direction logical co-ordinate space
0	0	512 pixels
0	1	1024 pixels
1	0	2048 pixels (not valid in Tiled mode, DO_ECR bit 1 = ‘0’)
1	1	3072 pixels (not valid in Tiled mode, DO_ECR bit 1 = ‘0’)

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	ALMWX	0	R/W	**Additional Linear Memory Width Offset (ALMWX)** These bits specify the background stride additional offset measured in horizontal pixels. These bits are valid when the background is set to linear mode (DO_ECR bit 1 = '1'). When valid the total stride in pixels is set by ALMWX + MWXOFS + MWX and allow the memory width to be set to 4 pixel aligned. When the background is set to tiled mode these bits are ignored.
2	MWXD	0	R/W	**MWX Disable (MWXD)** This bit disables the data in the MWX bits 7:6 to allow a memory width less than 512 pixels. When set to '1' the MWX bits 7:6 are ignored and just the data in the Memory Width Offset bits are used to calculate the memory width (the value of MWX is zero). This bit is only valid when the background is set to GE Linear mode.
1	GBMR	0	R/W	**Graphics Bit Mode B (GBMR)** This bit specifies the configuration (bits per pixel) of the B background display data. When set to '0' the B background data is 8 bits/pixel. When set to '1' the B background data is 16 bits/pixel.
0	GBML	0	R/W	**Graphics Bit Mode A (GBML)** This bit specifies the configuration (bits per pixel) of the background A display data. When set to '0' the background A data is 8 bits/pixel. When set to '1' the background A data is 16 bits/pixel.

Bit	Offset Value to be added to MWX
3	+4 pixels
4	+8 pixels
5	+16 pixels

PIP Extended Rendering Mode (DO_DUW)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			MWXOFS						MWX		ALMWX			MW XD		GBM
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved**
13 to 8	MWXOFS	0	R/W	**Memory Width Offset (MWXOFS)** These bits specify the PIP stride offset measured in horizontal pixels. The total stride in pixels is set by MWXOFS plus MWX. These bits are only valid when PIP is set to GE Linear mode (DO_SYSR bit 0 = '0' and DO_ECR bit 2 = '1').

Bit	Offset Value to be added to MWX
8	+32 pixels
9	+64 pixels
10	+128 pixels
11	+256 pixels
12	+512 pixels (this bit is not valid when MWX = 512 or the MWXD bit = ‘1’).
13	+1024 pixels (this bit is not valid when MWX = 512 or the MWXD bit = ‘1’).

Bit	Bit Name	Initial Value	R/W	Description
7, 6	MWX	0	R/W	**Memory Width (MWX)** These bits specify the PIP stride measured in horizontal pixels

Bit 7 MWX1	Bit 6 MWX0	X direction logical co-ordinate space
0	0	512 pixels
0	1	1024 pixels
1	0	2048 pixels (not valid in Tiled mode, DO_ECR bit 2 = ‘0’)
1	1	Reserved

Bit	Bit Name	Initial Value	R/W	Description
5 to 3	ALMWX	0	R/W	**Additional Linear Memory Width Offset (ALMWX)** These bits specify the PIP stride additional offset measured in horizontal pixels. These bits are valid when the PIP is set to linear mode (DO_ECR bit 2 = '1'). When valid the total stride in pixels is set by ALMWX + MWXOFS + MWX and allow the memory width to be set to 4 pixel aligned. When the PIP is set to tiled mode these bits are ignored.

Bit	Offset Value to be added to MWX
3	+4 pixels
4	+8 pixels
5	+16 pixels

Bit	Bit Name	Initial Value	R/W	Description
2	MWXD	0	R/W	**MWX Disable (MWXD)** This bit disables the data in the MWX bits 7:6 to allow a memory width less than 512 pixels. When set to '1' the MWX bits 7:6 are ignored and just the data in the Memory Width Offset bits are used to calculate the memory width (the value of MWX is zero). This bit is only valid when the PIP is set to GE Linear mode.
1	—	0	R	**Reserved**
0	GBM	0	R/W	**Graphics Bit Mode (GBM)** This bit specifies the configuration (bits per pixel) of the Picture In Picture (window) display data. When set to '0' the PIP data is 8 bits/pixel. When set to '1' the background data is 16 bits/pixel.

Display Size Register X* (DO_DSX)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DSX									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	DSX	0	R/W	**DSX** This register contains the number of horizontal pixels minus 1 to be displayed per scan line. (A single pixel line would have a value 10H'000 and a 1024 pixel line would have a value 10H'3FF).

Display Size Register Y* (DO_DSY)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DSY									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	DSY	0	R/W	**DSY** This register contains the number of scan lines minus 1 to be displayed. (A single line frame would have a value 10H'000 and a 1024 line frame would have a value 10H'3FF). Bit 9 is only valid when the Q2SD bit in the DO_SYSR = '0', and should be set to '0' when not valid.

Display Start Address Register 0H* (DO_DSAR0H)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

This register contains address data which when combined with DO_DSAR0L will make up a single 25 bit address for the top left pixel of first foreground (plane1) frame buffer. The mapping of bits to address register bits is dependent on the mode of operation. In linear mode only DO_DSAR0H is required to specify the 25-bit address. When the foreground is operated in tiled or Q2SD modes, the data written to registers DO_DSAR0H and DO_DSAR0L are combined to make up a single 25-bit address.

Q2SD mode: Address H'4028

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSALL0									DSAHL0						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 9	DSALL0	0	R/W	**DSALL0** When the Q2SD bit in the DO_SYSR bit 0 is '1', bits 15-9 will be valid and will map to A15-A9 in the address bus.
8, 7	—	0	R	**Reserved**
6 to 0	DSAHL0	0	R/W	**DSAHL0** When the Q2SD bit in the DO_SYSR bit is '1', bits 6-0 will be valid and will map to A22-A16 in the address bus.

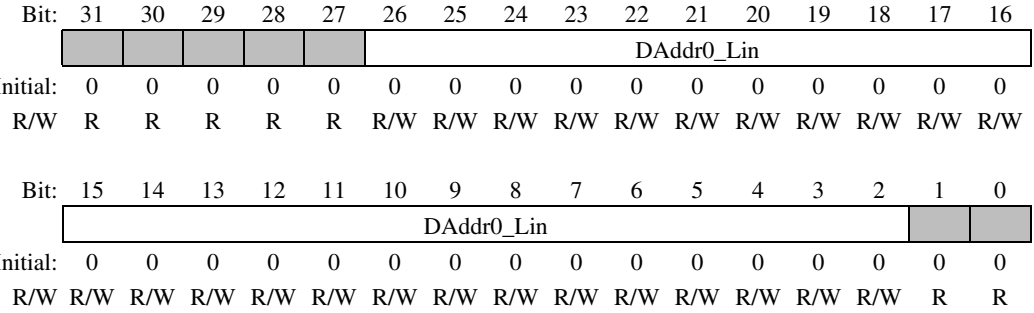
GE Tiled mode: Address H'4190

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																DSAHE0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DSAH0									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	DSAHE0	0	R/W	**DSAHE0** When the Q2SD bit in the DO_SYSR bit 0 is '0' and memory mode = tiled (bit 0 in DO_ ECR = '0'), bits 16 will be valid and will map to A26 in the address bus.
15 to 10	—	0	R	**Reserved**
9 to 0	DSAH0	0	R/W	**DSAH0** When the Q2SD bit in the DO_SYSR bit 0 is '0' and memory mode = tiled (bit 0 in DO_ECR = '0'), bits 9 to 0 will be valid and will map to A25 to A16 in the address bus.

GE Linear mode: Address H'4190

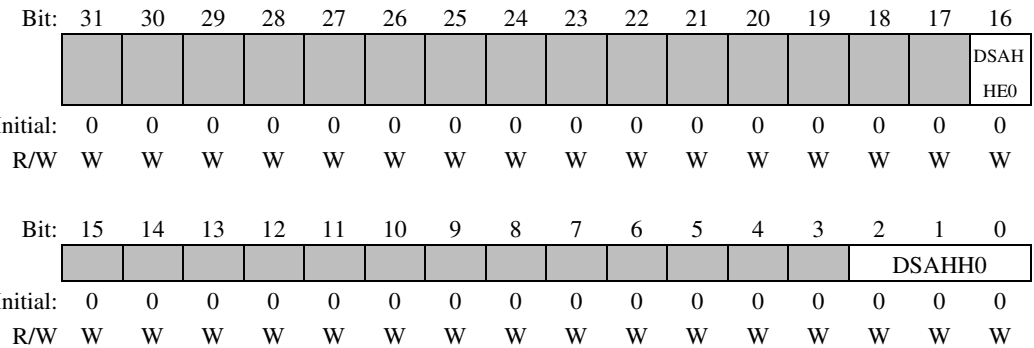


Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	DAddr0_Lin	0	R/W	**DAddr0_Lin** When the Q2SD bit in the DO_SYSR bit 0 is '0' and memory mode = linear (bit 0 in DO_ ECR = '1'), bits 26-2 will be valid and will map to A26-A2 in the address bus.
1, 0		0	R	**Reserved**

Display Start Address Register 0L (DO_DSAR0L)

This register contains address data which when combined with DO_DSAR0H will make up a single 25-bit address for the top left pixel of first foreground (plane1) frame buffer. The mapping of bits to address register bits is dependent on the mode of operation. When the foreground is operated in tiled or Q2SD modes, the data written to registers DO_DSAR0H and DO_DSAR0L are combined to make up a single 25-bit address. When in linear mode, the data stored in this register is ignored.

Q2SD mode: Address H'41C0



Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	W	**Reserved**
16	DSAHHE0	0	W	**DSAHHE0** When the Q2SD bit in the DO_SYSR, bit 0 is '1', bits 16 will be valid and will map to A26 in the address bus. This is an extension to the Q2SD spec
15 to 3	—	0	W	**Reserved**
2 to 0	DSAHH0	0	W	**DSAHH0** When the Q2SD bit in the DO_SYSR, bit 0 is '1', bits 2 to 0 will be valid and will map to A25 to A23 in the address bus.

GE Tiled mode: Address H'4194

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSAL0															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	W	**Reserved**
15 to 9	DSAL0	0	W	**DSAL0** When the Q2SD bit in the DO_SYSR, bit 0 is '0' and memory mode = tiled (DO_ECR bit 0 = '0'), bits 15 to 9 will be valid and will map to A15 to A9 in the address bus. When in linear memory mode (DO_SYSR bit 0 is '0' and DO_ECR bit 0 = '1') reading or writing to this register is prohibited.
8 to 0	—	0	W	**Reserved**

Display Start Address Register 1H * (DO_DSAR1H)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

This register contains address data which when combined with DO_DSAR1L will make up a single 25-bit address for the top left pixel of first foreground (plane1) frame buffer. The mapping of bits to address register bits is dependent on the mode of operation. In linear mode only DO_DSAR1H is required to specify the 25-bit address. When the foreground is operated in tiled or Q2SD modes, the data written to registers DO_DSAR1H and DO_DSAR1L are combined to make up a single 25-bit address.

Q2SD mode: Address H'402C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSALL1									DSAHL1						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	W	**Reserved**
15 to 9	DSALL1	0	W	**DSALL1** When the Q2SD bit in the DO_SYSR, bit 0 is '1', bits 15 to 9 will be valid and will map to A15 to A9 in the address bus.
8, 7	—	0	W	**Reserved**
6 to 0	DSAHL1	0	W	**DSAHL1** When the Q2SD bit in the DO_SYSR, bit 0 is '1', bits 6 to 0 will be valid and will map to A22 to A16 in the address bus.

GE Tiled mode: Address H'4198

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																DSAHE1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							DSAH1									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	DSAHE1	0	R/W	**DSAHE1** When the Q2SD bit in the DO_SYSR, bit 0 is '0' and memory mode = tiled (DO_ECR bit 0 = '0'), bits 16 will be valid and will map to A26 in the address bus.
15 to 10	—	0	R	**Reserved**
9 to 0	DSAH1	0	R/W	**DSAH1** When the Q2SD bit in the DO_SYSR, bit 0 is '0' and memory mode = tiled (DO_ECR bit 0 = '0'), bits 9 to 0 will be valid and will map to A25 to A16 in the address bus.

GE linear mode: Address H'4198

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						Daddr1_Lin										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Daddr1_Lin															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	Daddr1_Lin	0	R/W	When the Q2SD bit in the DO_SYSR, bit 0 is '1' and memory mode = linear (DO_ECR bit 0 = '1'), bits 26 to 2 will be valid and will map to A26 to A2 in the address bus.
1, 0		0	R	**Reserved**

Display Address Register 1L (DO_DSAR1L)

This register contains address data which when combined with DO_DSARH1 will make up a single 25 bit address for the top left pixel of first foreground (plane1) frame buffer. The mapping of bits to address register bits is dependent on the mode of operation. When the foreground is operated in tiled or Q2SD modes, the data written to registers DO_DSARH1 and DO_DSARL1 are combined to make up a single 25 bit address. When in linear mode, the data stored in this register is ignored.

Q2SD mode, Address H'41C4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																DSAHHE1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														DSAHH1		
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	W	**Reserved**
16	DSAHHE1	0	W	**DSAHHE1** When the Q2SD bit in the DO_SYSR, bit is '1', bits 16 will be valid and will map to A26 in the address bus. This is an extension to the Q2SD spec.
15 to 3	—	0	W	**Reserved**
2 to 0	DSAHH1	0	W	**DSAHH1** When the Q2SD bit in the DO_SYSR, bit 0 is '1', bits 2 to 0 will be valid and will map to A25 to A23 in the address bus.

GE Tiled mode, Address H'419C

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSAL1															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	W	**Reserved**
15 to 9	DSAL1	0	W	**DSAL1** When the Q2SD bit in the DO_SYSR, bit 0 is '0' and memory mode = tiled (DO_ECR bit 0 = '0'), bits 15 to 9 will be valid and will map to A15 to A9 in the address bus. When in linear memory mode (DO_SYSR bit 0 is '0' and DO_ECR bit 0 = '1') reading or writing to this register is prohibited.
8 to 0	—	0	W	**Reserved**

Horizontal Display Start Position * (DO_HDS)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HDS								
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	0	R	**Reserved**
8 to 0	HDS	0	R/W	**HDS** This register specifies the horizontal display start position in dot clock units. When the display synchronisation is in master mode or external master mode (DO_DSMR bit 7&6 = "00" or "11") HDS should be set to (HDS = HSW -1 + XS). When the display synchronisation is in TV Sync mode (DO_DSMR bit 7&6 = "10") HDS should be set to (HDS = HSW -4 + XS). See Functional Description: Sync-pulse generator for more details.

Horizontal Display End Position * (DO_HDE)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						HDE										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	0	R	**Reserved**
10 to 0	HDE	0	R/W	**HDE** This register specifies the horizontal display end position in dot clock units. When the display synchronisation is in master mode or external master mode (DO_DSMR bit 7&6 = "00" or "11") HDS should be set to (HDE = HSW -1 + XS + XW). When the display synchronisation is in TV Sync mode (DO_DSMR bit 7&6 = "10") HDE should be set to (HDE = HSW -4 + XS + XW). See Functional Description: sync-pulse generator for more details. Bit 10 is only valid when the DO_SYSR bit 0 is '0' (GE mode), and should be set to '0' when not valid.

Vertical Display Start Position * (DO_VDS)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								VDS								
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	0	R	**Reserved**
8 to 0	VDS	0	R/W	**VDS** This register specifies the vertical display start position in raster line units. (VDS = YS –2). See Functional Description: sync-pulse generator for more details.

Vertical Display End Position * (DO_VDE)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VDE									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VDE	0	R/W	**VDE** This register specifies the vertical display end position in raster line units. (VDE = YS + YW -2). See Functional Description: sync-pulse generator for more details. Bit 9 is only valid when the DO_SYSR bit 0 is set to '0' (GE mode), and should be set to '0' when not valid.

Horizontal Sync Pulse Width * (DO_HSWR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								HSWR								
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	0	R	**Reserved**
8 to 0	HSWR	0	R/W	**HSWR** This register contains the number of dot clock cycles (pixels) for the duration that H-sync is active. (HSWR = HSW –1). Bit 8 & 7 are only valid when the DO_SYSR bit 0 is set to '0' (GE mode), and should be set to '0' when not valid. This register is not used in TV Sync mode. See Functional Description: sync-pulse generator for more details.

Horizontal Scan Cycle Register * (DO_HCR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						HCR										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	0	R	**Reserved**
10 to 0	HCR	0	R/W	**HCR** This register specifies the duration of the horizontal cycle in dot clock cycles (HCR = HC - 1). See Functional Description: sync-pulse generator for more details.

Vertical Sync Position Register* (DO_VSP)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VSP									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VSP	0	R/W	**VSP** This register specifies the start position when the VSYNC signal becomes active, in raster line units. (VSP = VC – VSW –1). This register is not used in TV Sync mode. See Functional Description: sync-pulse generator for more details.

Vertical Scan Cycle Register * (DO_VCR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VCR									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VCR	0	R/W	**VCR** This register specifies the duration of the vertical cycle in raster scan units. (VCR = VC –1). See Functional Description: sync-pulse generator for more details.

Display Off Output Registers* (DO_DOORH, DO_DOORL)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

These registers specify the display data to be output while the display is off. A 6 bit value is given for each of RGB.

DO_DOORH

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RED							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 2	RED	0	R/W	These bits contain the 6 bit red data.
1, 0	—	0	R	**Reserved**

DO_DOORL

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GREEN								BLUE							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 10	GREEN	0	R/W	These bits contain the 6 bit green data.
9, 8	—	0	R	**Reserved**
7 to 2	BLUE	0	R/W	These bits contain the 6 bit blue data.
1, 0	—	0	R	**Reserved**

Equalising Pulse Width Register* (DO_EQWR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										EQWR						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6 to 0	EQWR	0	R/W	**EQWR** This register specifies the low level pulse width of csync signal equalising pulses in dot-clock units. Equalising pulses are generated at the start and middle of each raster. This register is only valid when CSY1 (bit 7 in the DSMR2 Register) is set to '1'.

Separation Width Register * (DO_SPWR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							SPWR									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	SPWR	0	R/W	**SPWR** This register specifies the low level pulse width of csync signal separation pulses in dot-clock units. Separation pulses are generated at the start and middle of each raster. Set the SPW value to less than half the horizontal scan interval. This register is only valid when CSY1 (bit 7 in the DSMR2 Register) is set to '1'.

PIP Horizontal Display Start Position* (DO_HVP)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							HVP									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	HVP	0	R/W	**HVP** This register specifies the horizontal position of the top left corner of the PIP window within the display window. It is specified in dot-clock units where H'000 would represent the far left pixel in the display. This register is double buffered and the value written will not take effect until after the next vblank start.

PIP Vertical Display Start Position Register* (DO_VVP)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VVP									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VVP	0	R/W	**VVP** This register specifies the vertical position of the top left corner of the PIP window within the display window. It is specified in raster line units where H'000 would represent the top pixel in the display. This register is double buffered and the value written will not take effect until after the next vblank start.

PIP Start Address Registers0, H/L* (DO_VSAR0H, DO_VSAR0L)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

These registers contain the memory address for the top left pixel of the fist PIP memory (VID0) area. When PIP start address mode DO_ECR bit 8 = '0', the most recent video image captured in PIP areas 1 to 3 will be displayed. When DO_ECR bit 8 = '1', this register is used to contain the PIP start address, independent of video capture.

DO_VSAR0H Q2SD or GE Tiled mode: The following bits are valid when in Q2SD mode (DO_SYSR bit 0 is set to '1') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0'), or when in GE mode, (DO_SYSR bit 0 is set to '0') and PIP start address mode (DO_ECR bit 8 = '0') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0').

Q2SD or GE Tiled Mode: Address H'40C4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																VSAHE0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VSAH0									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	VSAHE0	0	R/W	**PIP Start Address High Extension 0 (VSAHE0)** When valid VSAHE0 bits 16 map to bit 26 in the address bus.
15 to 10	—	0	R	**Reserved**
9 to 0	VSAH0	0	R/W	**PIP Start Address High 0 (VSAH0)** When valid VSAH0 bits 9:0 map to bits 25:16 in the address bus.

DO_VSAR0H Linear mode: These bits are valid when in the PIP Window is set to linear memory (DO_ECR bit 2 = '1').

GE Linear Mode:Address H'40C4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						VAddr0_Lin										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VAddr0_Lin															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	VAddr0_Lin	0	R/W	**Vaddr0_Lin** When valid DO_VSAH0 bits 26:2 map to bits 26:2 in the address bus. This function is an extension to the Q2SD spec.
1, 0		0	R	**Reserved**

DO_VSAR0L Q2SD or GE Tiled mode: The following bits are valid when in Q2SD mode (DO_SYSR bit 0 is set to '1') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0'), or when in GE mode, (DO_SYSR bit 0 is set to '0') and PIP start address mode (DO_ECR bit 8 = '0') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0').

Q2SD or GE Tiled Mode: Address H'40C8

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSAL0															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 10	VSAL0	0	R/W	**PIP Start Address Low 0 (VSAL0)** When valid VSAH0 bits 15:10 map to bits 15:10 in the address bus.
9 to 0	—	0	R	**Reserved**

PIP Start Address Registers1, H/L* (DO_VSAR1H, DO_VSAR1L)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

These registers contain the memory address for the top left pixel of the second PIP memory (VID1) area. When PIP start address mode = '0', the most recent video image captured in PIP areas 1 to 3 will be displayed.

DO_VSAR1H Q2SD or GE Tiled mode: The following bits are valid when in Q2SD mode (DO_SYSR bit 0 is set to '1') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0')., or when in GE mode, (DO_SYSR bit 0 is set to '0') and PIP start address mode (DO_ECR bit 8 = '0') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0').

Q2SD or GE Tiled Mode: Address H'40CC

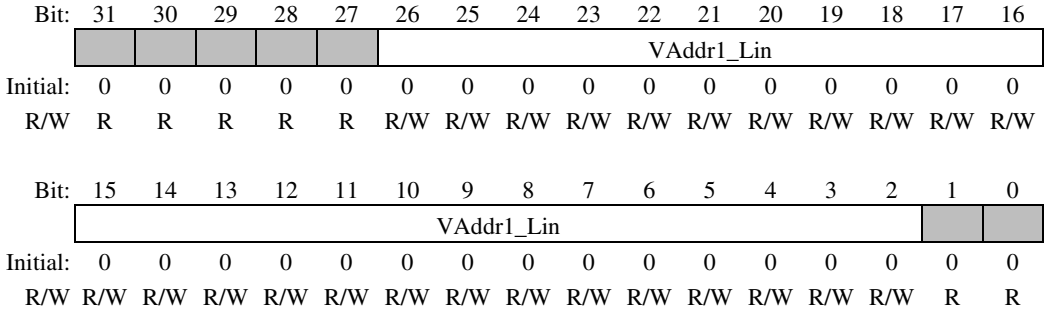
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																VSAHE1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VSAH1									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	VSAHE1	0	R/W	**PIP Start Address High Extension 1 (VSAHE1)** When valid VSAHE1 bits 16 map to bit 26 in the address bus.
15 to 10	—	0	R	**Reserved**
9 to 0	VSAH1	0	R/W	**PIP Start Address High 1 (VSAH1)** When valid VSAH1 bits 9:0 map to bits 25:16 in the address bus.

DO_VSAR1H Linear mode: These bits are valid when in the PIP Window is set to linear memory (DO_ECR bit 2 = '1').

GE Linear Mode: Address H'40CC



Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	VAddr1_Lin	0	R/W	**Vaddr1_Lin** When valid DO_VSAH1 bits 26:2 map to bits 26:2 in the address bus. This function is an extension to the Q2SD spec.
1, 0		0	R	**Reserved**

DO_VSAR1L Q2SD or GE Tiled mode: The following bits are valid when in Q2SD mode (DO_SYSR bit 0 = '1') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0'), or when in GE mode, (DO_SYSR bit 0 = '0') and PIP start address mode (DO_ECR bit 8 = '0') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0').

Q2SD or GE Tiled Mode: Address H'40D0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSAL1															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 10	VSAL1	0	R/W	**PIP Start Address Low 1 (VSAL1)** When valid VSAH1 bits 15:10 map to bits 15:10 in the address bus.
9 to 0	—	0	R	**Reserved**

PIP Start Address Registers2, H/L* (DO_VSAR2H, DO_VSAR2L)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

These registers contain the memory address for the top left pixel of the third PIP memory area (VID2). When PIP start address mode = '0', the most recent video image captured in PIP areas 1 to 3 will be displayed.

DO_VSAR2H Q2SD and GE Tiled mode: The following bits are valid when in Q2SD mode (DO_SYSR bit 0 = '1') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0'), or when in GE mode, (DO_SYSR bit 0 = '0') and PIP start address mode (DO_ECR bit 8 = '0') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0').

Q2SD or GE Tiled Mode: Address H'40D4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																VSAHE2
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VSAH2									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	VSAHE2	0	R/W	**PIP Start Address High Extension 2 (VSAHE2)** When valid VSAHE2 bit 16 map to bits 26 in the address bus.
15 to 10	—	0	R	**Reserved**
9 to 0	VSAH2	0	R/W	**PIP Start Address High 2 (VSAH2)** When valid VSAH2 bits 9:0 map to bits 25:16 in the address bus.

DO_VSAR2H Linear mode: These bits are valid when in the PIP Window is set to linear memory (DO_ECR bit 2 = '1').

GE Linear Mode: Address H'40D4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						VAddr2_Lin										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VAddr2_Lin															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	VAddr2_Lin	0	R/W	**Vaddr2_Lin** When valid DO_VSAH2 bits 26:2 map to bits 26:2 in the address bus. This function is an extension to the Q2SD spec.
1, 0		0	R	**Reserved**

DO_VSAR2L Q2SD or GE Tiled mode: The following bits are valid when in Q2SD mode (DO_SYSR bit 0 = '1') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0')., or when in GE mode, (DO_SYSR bit 0 = '0') and PIP start address mode (DO_ECR bit 8 = '0') and PIP Window is set to tiled memory (DO_ECR bit 2 = '0').

Q2SD or GE Tiled Mode: Address H'40D8

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSAL2															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 10	VSAL2	0	R/W	**PIP Start Address Low 2 (VSAL2)** When valid VSAH0 bits 15:10 map to bits 15:10 in the address bus.
9 to 0	—	0	R	**Reserved**

PIP Window Size Register X* (DO_VSIZEX)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VSIZEX									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VSIZEX	0	R/W	**VSIZEX** This register contains the number of horizontal pixels to be displayed in the PIP window. (A single pixel line would have a value 10H'001 and a 1023 pixel line would have a value 10H'3FF). Bit 0 is not valid when DO_SYSR bit 0 = '1' (Q2SD mode), and should be set to '0' when not valid.

PIP Window Size Register Y * (DO_VSIZEY)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VSIZEY									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VSIZEY	0	R/W	**VSIZEY** This register contains the number of vertical lines to be displayed in the PIP window. (A single line window would have a value 10'h001 and a 1023 line window would have a value 10'h3FF). Bit 0 is not valid when DO_SYSR bit 0 = '1' (Q2SD mode), and should be set to '0' when not valid.

Video Incorporation Mode Register* (DO_VIMR)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VID1	VID0														
Initial:	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	VID1	1	R	**Video Window Status (VID1, VID0)**
14	VID0	1	R	These bits are read only and indicate the most recent image incorporated from the video input. The data passed to this register will be passed from the Video In module as a side band signal. These bits are only valid when PIP start address mode (bit 8 in DO_ECR) is set to '0'. 00: Most recent image is in PIP area 0, When the PIP window is enabled (VWE = '1') PIP area 0 is displayed 01: Most recent image is in PIP area 1, When the PIP window is enabled (VWE = '1') PIP area 1 is displayed 10: Most recent image is in PIP area 2, When the PIP window is enabled (VWE = '1') PIP area 2 is displayed 11: Indicates initial state after reset, When the PIP window is enabled (VWE = '1') PIP area 0 is displayed
13 to 0	—	0	R	**Reserved** For comparison with the Q2SD specification, only non-interlaced video is supported and incorporated field select bits 3:2 is reserved as "00".

Cursor1 Horizontal Display Start Position* (DO_HCS1)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															BLKA	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BLKA						HCS1									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	0	R	**Reserved**
17 to 10	BLKA	0	R/W	**Cursor Blink Shape A Display Interval (BLKA)** These bits specify the duration in frames units that cursor shape A is displayed. Bits 16 and 17 are additional to the Q2SD spec. This field is used for both cursor1 and cursor2. The default setting is for each cursor to blink between its shape A and shape B. A setting of BLKA = 8H'00 would display shape A for 1 frame before switching to shape B.
9 to 0	HCS1	0	R/W	**Cursor1 Horizontal Display Start Position (HCS1)** These bits set the cursor1 horizontal display position in dot clock units. (Placing the left side of the cursor at the far left pixel of the display would have a value 10H'000). These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Cursor1 Vertical Display Start Position * (DO_VCS1)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															BLKB	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BLKB						VCS1									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	0	R	**Reserved**
17 to 10	BLKB	0	R/W	**Cursor Blink Shape B Display Interval (BLKB)** These bits specify the duration in frames units that cursor shape B is displayed. Bits 16 and 17 are additional to the Q2SD spec. This field is used for both cursor1 and cursor2. The default setting is for each cursor to blink between its shape A and shape B. A setting of BLKB = 8H'00 would display shape B for 1 frame before switching to shape A.
9 to 0	VCS1	0	R/W	**Cursor1 Vertical Display Start Position (VCS1)** These bits set the cursor1 vertical display position in dot clock units. (Placing the top side of the cursor at the top pixel of the display would have a value 10H'000). When in Q2SD mode (DO_SYSR bit 0 = '1'), only bits 8 to 0 are valid and only '0' should be written to bit 9. These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Cursor2 Horizontal Display Start Position* (DO_HCS2)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							HCS2									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	HCS2	0	R/W	**Cursor2 Horizontal Display Start Position (HCS2)** These bits set the cursor2 horizontal display position in dot clock units. (Placing the left side of the cursor at the far left pixel of the display would have a value 10H'000). These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Cursor2 Vertical Display Start Position * (DO_VCS2)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							VCS2									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	VCS2	0	R/W	**Cursor2 Vertical Display Start Position (VCS2)** These bits set the cursor2 vertical display position in dot clock units. (Placing the upper side of the cursor at the top pixel of the display would have a value 10'H000). When in Q2SD mode (DO_SYSR bit 0 = '1'), only bits 8 to 0 are valid and only '0' should be written to bit 9. These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Cursor1 Start Address Registers* (DO_CSAR1)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

This register contains the memory address for the Cursor1 memory area. The address defines the start of the cursor1 shape A area. The bit map for cursor1 shape B will immediately follow the shape A memory address. (For a 32 × 32 pixel cursor, shape B data will start at address CSAR1 + 1024. For a 64 × 64 pixel cursor, shape B data will start at address CSAR1 + 4096.)

The A10-A0 map of the address bus will be always 11'H0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																CSAHE1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSAL1						CSAH1									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	CSAHE1	0	R/W	**CSAHE1** Bit 16 (CSAHE1) map to A26 map of the address bus.
15 to 11	CSAL1	0	R/W	**CSAL1** Bits 15 to 11 (CSAL1) map to A15 to A11 of the address bus. Bits 12 & 11 are only valid when in 32 × 32 pixel mode (DO_ECR bit 9 = '0'). When not valid only '0' should be read or written to these bits.
10	—	0	R	**Reserved**
9 to 0	CSAH1	0	R/W	**CSAH1** Bits 9 to 0 (CSAH1) map to A25 to A16 of the address bus.

Cursor2 Start Address Registers* (DO_CSAR2)

The bits implemented in this register are binary compatible with the Q2SD unless otherwise stated.

This register contains the memory address for the Cursor2 memory area. The address defines the start of the cursor2 shape A area. The bit map for cursor2 shape B will immediately follow the shape A memory address. (For a 32 × 32 pixel cursor, shape B data will start at address CSAR2 + 1024. For a 64 × 64 pixel cursor, shape B data will start at address CSAR2 + 4096.)

The A10-A0 map of the address bus will be always 11'h0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																CSAHE2
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSAL2						CSAH2									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	CSAHE2	0	R/W	**CSAHE2** Bit 16 (CSAHE2) map to A26 map of the address bus.
15 to 11	CSAL2	0	R/W	**CSAL2** Bits 15-11 (CSAL2) map to A15-A11 of the address bus. Bits 12 & 11 are only valid when in 32 × 32 pixel mode (DO_ECR bit 10 = '0'). When not valid only '0' should be read or written to these bits
10	—	0	R	**Reserved**
9 to 0	CSAH2	0	R/W	**CSAH2** Bits 9 to 0 (CSAH2) map to A25 to A16 of the address bus.

Background Start Address Registers A (DO_LBGSAR)

This register contains the start address for the A background area used in the GE background wrap function. See Functional Description: Background Wrap for more details.

GE Tiled Mode: The following bits are valid when in GE mode (DO_SYSR bit 0 = '0') and the background memory configuration is tiled (DO_ECR bit 1 = '0'). The memory address written to this register must be tile aligned. One tile is defined by the expression 16 × MWX × bytes/pixel.

GE Tiled Mode: Address H'41B0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																LBGSAE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBGASA						LBGSA									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	LBGSAE	0	R/W	**A Back Ground Start Address Extension (LBGSAE)** When valid, bit 16 map to bits 26 of the address bus.
15 to 13	LBGASA	0	R/W	**A Back Ground Additional Start Address (LBGASA)** When valid, bits 15:13 map to bits 15:13 of the address bus.
12 to 10	—	0	R	**Reserved**
9 to 0	LBGSA	0	R/W	**A Back Ground Start Address (LBGSA)** When valid, bits 9:0 map to bits 25:16 of the address bus.

GE Linear Mode: The following bits are valid when in GE mode (DO_SYSR bit 0 = '0') and the background memory configuration is linear (DO_ECR bit 1 = '1')

GE Linear Mode: Address H'41B0

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						LBGSAddr_Lin										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBGSAddr_Lin															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	LBGSAddr_Lin	0	R/W	**A Back Ground Start Address Linear (LBGSAddr_Lin)** When valid, bits 26:2 map to bits 26:2 of the address bus.
1, 0		0	R	**Reserved**

Background Start Address Registers B (DO_RBGSAR)

This register contains the start address for the B background area used in the GE background wrap function. See Functional Description: Background Wrap for more details. This register should not be read or written to when in Q2SD mode.

GE Tiled Mode: The following bits are valid when in GE mode (DO_SYSR bit 0 = '0') and the background memory configuration is tiled (DO_ECR bit 1 = '0'). The memory address written to this register must be tile aligned. (One tile = 16 × MWX × bytes/pixel).

GE Tiled Mode: Address H'41B4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																RBGSAE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBGASA						RBGSA									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	RBGSAE	0	R/W	**B Back Ground Start Address Extension (RBGSAE)** When valid, bit 16 map to bits 26 of the address bus.
15 to 13	RBGASA	0	R/W	**B Back Ground Additional Start Address (RBGASA)** When valid, bits 15:13 map to bits 15:13 of the address bus.
12 to 10	—	0	R	**Reserved**
9 to 0	RBGSA	0	R/W	**B Back Ground Start Address (RBGSA)** When valid, bits 9:0 map to bits 25:16 of the address bus.

GE Linear Mode: The following bits are valid when in GE mode (DO_SYSR bit 0 = '0') and the background memory configuration is linear (DO_ECR bit 1 = '1')

GE Linear Mode: Address H'41B4

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						RBGSAddr_Lin										
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBGSAddr_Lin															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27		0	R	**Reserved**
26 to 2	RBGSAddr_Lin	0	R/W	**B Back Ground Start Address Linear (RBGSAddr_Lin)** When valid, bits 26:2 map to bits 26: of the address bus.
1, 0		0	R	**Reserved**

Background A Start Position X* (DO_LBGSX)

These bits specify the horizontal position of the top left corner of the A Background Display within the A Background Area. It is specified in pixel co-ordinates where H'000 would represent the far left pixel in the background area. In Q2SD or GE Tiled mode the maximum width of LBGSX = 1023, therefore bits 11:10 should always be '0'. See Functional Description: Background Wrap for more details. These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					LBGSX											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11 to 0	LBGSX	0	R/W	

Background A Start Position Y* (DO_LBGSY)

This register specifies the vertical position of the top left corner of the A Background Display within the A Background Area. It is specified in pixel co-ordinates where H'000 would represent the top most pixel in the background area. In Q2SD mode this register is used to define the start address for the background area in pixel co-ordinates, with address H'00000000 as the y co-ordinate 0. The start address for the background in Q2SD can be calculated by DO_LBGSY(13:9) * MWX * 512*bg bytes/pixel.

The linear start address for the background can be calculated from LBGSY multiplied by MWX. In GE Tiled mode the maximum setting for the Y background size is 1023 or 511 depending on the setting of DO_DSMR bit 14. The unused bit should always be written to '0'. See Functional Description: Background Wrap or the Q2SD spec for more details. These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			LBGSY													
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved**
13 to 0	LBGSY	0	R/W	

Background B Start Position X (DO_RBGSX)

These bits specify the horizontal position of the top left corner of the B Background Display within the B Background Area. It is specified in pixel co-ordinates where H'000 would represent the far left pixel in the background area. In GE Tiled mode the maximum width of RBGSX = 1023, therefore bits 11:10 should always be '0'. See Functional Description: Background Wrap for more details. These bits are double buffered and data written to these bits will only take effect after the next vblank start

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					RBGSX											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11 to 0	RBGSX	0	R/W	

Background B Start Position Y (DO_RBGSY)

This register specifies the vertical position of the top left corner of the B Background Display within the B Background Area. It is specified in pixel co-ordinates where H'000 would represent the top most pixel in the background area. In GE Tiled mode the maximum setting for the Y background size is 1023 or 511 depending on the setting of DO_DSMR bit 13. The unused bits should always be written to '0'. See Functional Description: Background Wrap for more details. These bits are double buffered and data written to these bits will only take effect after the next vblank start.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			RBGSY													
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved**
13 to 0	RBGSY	0	R/W	

Vertical Wraparound Size (DO_WRPY)

This register specifies the vertical size of the background wrap plane in pixels. This register is not used in Q2SD or GE Tiled mode. In Q2SD mode the Y direction wrap size is fixed to 512 pixels and in GE Tiled mode the wrap size is controlled by DO_DSMR bits 14:13. See Functional Description: Background Wrap for more details.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					WRPY											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11 to 0	WRPY	0	R/W	

Display Blending 1&2 (DO_DBR1, DO_DBR2)

These registers contain the data required to mix the PIP and background pixel data with the foreground. The alpha blending equation and further information on alpha blending can be found in Functional Description: Alpha Blending.

DO_DBR1

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPE	LBGE	RBGE						PIPAV							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	PIPE	0	R/W	**PIP Alpha Blend Enable (PIPE)** When this bit is set to '0' alpha blending is disabled and mixing of the foreground and PIP is performed using only transparency. When this bit is set to '1' alpha blending is enabled and the mixing of foreground and PIP uses the PIPAV alpha blend value.
14	LBGE	0	R/W	**Background A Alpha Blend Enable (LBGE)** When this bit is set to '0' alpha blending is disabled and mixing of the foreground and A background is performed using only transparency. When this bit is set to '1' alpha blending is enabled and the mixing of foreground and left background uses the LBGAV alpha blend value.
13	RBGE	0	R/W	**Background B Alpha Blend Enable (RBGE)** When this bit is set to '0' alpha blending is disabled and mixing of the foreground and B background is performed using only transparency. When this bit is set to '1' alpha blending is enabled and the mixing of foreground and B background uses the RBGAV alpha blend value.
12 to 8	—	0	R	**Reserved**
7 to 4	PIPAV	0	R/W	**PIP alpha value (PIPAV)** These bits contain the alpha value for blending the PIP pixel data with the foreground.
3 to 0	—	0	R	**Reserved**

DO_DBR2

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LBGAV								RBGAV							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 12	LBGAV	0	R/W	**Background A Alpha Value (LBGAV)** These bits contain the alpha value for blending the A Background pixel data with the foreground.
11 to 8	—	0	R	**Reserved**
7 to 4	RBGAV	0	R/W	**Background B Alpha Value (RBGAV)** These bits contain the alpha value for blending the B Background pixel data with the foreground.
3 to 0	—	0	R	**Reserved**

Foreground Transparent Color (DO_TRNFGR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																FGTE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRNFGR															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	FGTE	0	R/W	**Foreground Transparent Enable (FGTE)** When set to '1', this bit enables any foreground color matching the value in TRNFGR to be displayed as transparent. When this bit is set to '0' the transparent function is disabled.
15 to 0	TRNFGR	0	R/W	**Foreground Transparent Color (TRNFGR)** This register contains the chroma key (transparent) color for the foreground plane. When foreground is operated in 8 bit/pixel mode, bits 7:0 are replicated in bits 15:8.

Cursor1 Transparent Color (DO_TRNC1R)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TRNC1R							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 0	TRNC1R	0	R/W	**TRNC1R** This register contains the chroma key (transparent) color for Cursor1.

Cursor2 Transparent Color (DO_TRNC2R)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TRNC2R							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 0	TRNC2R	0	R/W	**TRNC2R** This register contains the chroma key (transparent) color for Cursor2.

Display Out Extension Control Register (DO_ECR)

This register must be set before any of the memory start address registers are written to. This register should only be written to prior to any display planes being enabled, or during the vblank period. IDOC, IDA, IVS, and IHS are updated during display reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					FGBS				BGBS					END		
Initial:	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		C2BF	C1BF	C2BE	C1BE	C2BS	C1BS	PSAM	IDOC	IDA	IVS	IHS	DAE	PPMM	BGMM	FGMM
Initial:	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	0	R	**Reserved**
27	FGBS	1	R/W	**Foreground DMA Burst Size (FGBS)**
26		0	R/W	Bits contain the foreground burst size of the DMA requests in linear mode. The burst size is fixed to 4'H8 in tiled mode. This level should be set with the DMA threshold size to allow maximum data through put, which will depend on the display out PLL clock or the external DOT_CLK speed. The burst size should be set to 4'H8 for most applications with FGBS set to 4'H8.
25		0	R/W	
24		0	R/W	
23	BGBS	1	R/W	**Background DMA Burst Size (BGBS)**
22		0	R/W	Bits contain the background burst size of the DMA requests in linear mode. The burst size is fixed to 4'H8 in tiled mode. This level should be set with the DMA threshold size to allow maximum data through put, which will depend on the display out PLL clock or the external DOT_CLK speed. The burst size should be set 8 for most applications with BGBS set to 4'H8.
21		0	R/W	
20		0	R/W	
19	—	0	R	**Reserved**
18	END	0	R/W	**Endian Select (END)** When this bit is set to 0, DMA data is unpacked big endian, when set to 1, DMA data is unpacked little endian.
17 to 15	—	0	R	**Reserved**

Bit	Bit Name	Initial Value	R/W	Description
14	C2BF	0	R/W	**Cursor2 Blink Function (C2BF)** When this bit is set to 0, Cursor2 will blink between shape A bitmap and shape B bitmap. Blink speed is controlled by BLKA and BLKB bits in DO_HCS1 and DO_VCS1 Registers respectively. When set to 1, Cursor 2 will blink between the shape A bitmap and transparent.
13	C1BF	0	R/W	**Cursor1 Blink Function (C1BF)** When set to 0, Cursor1 will blink between shape A bitmap and shape B bitmap. Blink speed is controlled by BLKA and BLKB bits in DO_HCS1 and DO_VCS1 Registers respectively. When this bit is set to 1, Cursor 1 will blink between the shape A bitmap and transparent.
12	C2BE	1	R/W	**Cursor2 Blink Enable (C2BE)** When this bit is set to 1, Cursor2 (C2) blink function is enabled and output to the display, when set to 0 the cursor is constant.
11	C1BE	1	R/W	**Cursor1 Blink Enable (C1BE)** When this bit is set to 1, Cursor1 (C1) blink function is enabled and output to the display, when set to 0 the cursor is constant.
10	C2BS	0	R/W	**Cursor2 Bit-map Size(C2BS)** When this bit is set to 0, the Cursor2 bitmap is size 32×32 pixels, when this bit is set to '1', the Cursor2 bitmap size is 64×64 pixels.
9	C1BS	0	R/W	**Cursor1 Bit-map Size (C1BS)** When this bit is set to 0, the Cursor1 bitmap is size 32×32 pixels, when this bit is set to '1', the Cursor1 bitmap size is 64×64 pixels.
8	PSAM	0	R/W	**PIP Start Address Mode (PSAM)** When this bit is set to '0', the PIP data will use the video area start address 0-2 and the data displayed will be set by bits 15 and 14 in the DO_VIMR Register. When this bit is set to '1', the PIP data will use the address stored in the Video Start Address Register 0, DO_VSAR0 H/L.
7	IDOC	0	R/W	**Invert DO_Clk (IDOC)** When this bit is set to 0, Data is output on the rising edge of DOT_CLK (dot clock), when set to 1 pixel data is output on the falling edge of the DOT_CLK.

Bit	Bit Name	Initial Value	R/W	Description
6	IDA	0	R/W	**Invert Display Active (IDA)** When this bit is set to 0, Display Active is active high, when set to 1 Display Active is active low.
5	IVS	0	R/W	**Invert V_Sync (IVS)** When this bit is set to 0, V-sync is active low, when set to 1 V-sync is active high.
4	IHS	0	R/W	**Invert H_Sync (IHS)** When this bit is set to 0, H-sync is active low, when set to 1 H-sync is active high.
3	DAE	1	R/W	**Display Active Enable (DAE)** When this bit is set to 1, the Display Active output is active during the duration that pixel data is output from the display out. (Display Active is equivalent to display interval in the Q2SD spec). When set to '0', Display Active Enable is held inactive,
2	PPMM	0	R/W	**Picture in Picture Memory Mode (PPMM)** When this bit is set to 0, the PIP data is accessed as tiled memory and is compatible with Q2SD functions. When this bit is set to '1', PIP data is accessed as linear memory. Linear addressing is an additional function to the Q2SD spec.
1	BGMM	0	R/W	**Background Memory Mode (BGMM)** When this bit is set to 0, the background data is accessed as tiled memory and is compatible with Q2SD functions. When this bit is set to '1', background data is accessed as linear memory. Linear addressing is an additional function to the Q2SD spec.
0	FGMM	0	R/W	**Foreground Memory Mode (FGMM)** When this bit is set to 0, the foreground data is accessed as tiled memory and is compatible with Q2SD functions. When this bit is set to '1', foreground data is accessed as linear memory. Linear addressing is an additional function to the Q2SD spec.

Line Interrupt Register (DO_LIR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							LIR									
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	0	R	**Reserved**
9 to 0	LIR	0	R/W	**LIR** This register contains the scan line co-ordinate at the start of which, when enabled, an interrupt will occur. (The first scan line of the frame would have a value 10'H000 and a 1024 line would have a value 10'H3FF).

Display Out PLL Register (DO_PLL)

This register contains the data which is used to control and set-up the Display Out PLL to produce the correct Display Out Clock and Data frequency.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
					RBF		ME	FDT						DIVC		DIVB
Initial:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DIVB			DIVA						DIVP		DIVN			PCKE	PLLE
Initial:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	0	R	**Reserved**
27	RBF	0	R/W	**Register Bus Frequency (RBF)**
26		0	R/W	These bits need to be set to match the system control of the register bus frequency. When set to "00" the register bus must operate at pix_clk frequency/2, and when set to "01", the register bus must operate at pix_clk frequency/3. When set to "10" the register bus must operate at anti-phase to the pix_clk frequency. This setting is for FPGA testing and is not valid for use in the ASIC. When operated in FPGA mode, a delay of at least 1 rbclk cycle must occur between consecutive color palette read and writes. When these bits are changed, software must ensure that the register bus module select signal become inactive for at least 2 rbclk cycles. This can be achieved by stopping register accesses or reading or writing data from another module. The correct value for this register must be set before reading or writing to the color palette.
25	ME	0	R/W	**Module Enable (ME)** When set to 0 (default value) all module pin outputs are tri-stated (except for DOT_CLK) which is set as an input). When set to 1 the module will operate normally and all external pins will be configured from normal operation registers.
24	FDT	0	R/W	**FIFO DMA Threshold (FDT)**
23		0	R/W	FDT must be set to 5'H17.
22		1	R/W	
21		1	R/W	
20		1	R/W	
19	—	0	R	**Reserved**
18	DIVC	0	R/W	**PLL divider C (DIVC)**
17		0	R/W	Bits contain data for the external PLL output-divider C. All divider values should be programmed as divider_ratio-1. See PLL section at the end of this document for more details on PLL settings.

Bit	Bit Name	Initial Value	R/W	Description
16	DIVB	0	R/W	**PLL divider B (DIVB)**
15		0	R/W	Bits contain data for the external PLL feedback-divider B. All divider values should be programmed as divider_ratio-1. See PLL section at the end of this document for more details on PLL settings.
14		0	R/W	
13		0	R/W	
12	DIVA	0	R/W	**PLL divider A (DIVA)**
11		0	R/W	Bits contain data for the external PLL pre-divider A. All divider values should be programmed as divider_ratio-1. See PLL section at the end of this document for more details on PLL settings.
10		0	R/W	
9		0	R/W	
8	—	0	R	**Reserved**
7		0	R	
6	DIVP	0	R/W	**PLL divider P (DIVP)**
5		1	R/W	Bits contain data for PLL divider P. See PLL section at the end of this document for more details on PLL settings. DIVP should be set to "01" or "10" .
4	DIVN	0	R/W	**PLL divider N (DIVN)**
3		0	R/W	Bits contain data for PLL divider N. See PLL section at the end of this document for more details on PLL settings.
2		0	R/W	
1	PCKE	0	R/W	**PLL Clock Enable (PCKE)** When set to 1 the input used to drive dot clock (either dot clock PLL or external dot clock will be output to the module, and the display out clock and sync signals will be generated. When set to 0, the input used to drive dot clock is held high, ther efore no clock drives the SPG (Sync Pulse Generator in Figure 10.1) and no display or sync signals are generated.
0	PLLE	0	R/W	**PLL Enable (PLLE)** When set to 1 the display out PLL will run freely, when set to 0, the PLL is put into standby mode.

Color Palette Registers (CP000RH/L to CP255RH/L)

These registers contain 18 bit color data for the 8bits/pixel mode. It is only possible to read or write to these registers while all 8-bit planes are disabled, or during the vblank active period.

For Q2SD compatibility, long word accesses must be used when writing data to the color pallet. In Q2SD mode data written to the CPXXXRH Register is only written to the color palette when the CPXXXRL Register is written to. Consecutive writes to the to the high register followed by the low register should always be performed.

The color palette can be written to like all other register bus registers, but as it is implemented in single port RAM, reading data from it must follow the following procedure. The desired read location must first be read from the color palette, but only zeros will be returned by the register bus. After two rbclk cycles the last read color palette data will be available to be read from the color palette read register.

Q2SD Mode:

CP000RH to CP255RH

Address: CP000RH = H'100, Address: CP001RH = H'102, Address: CP255RH = H'2FE.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RED							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 2	RED	0	R/W	**Red** These bits contain Red palette data.
1, 0	—	0	R	**Reserved**

CP000RL to CP255RL

Address: CP000RL = H'101, Address: CP001RL = H'103, Address: CP255RL = H'2FF.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GREEN								BLUE							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 10	GREEN	0	R/W	**Green** These bits contain Green palette data.
9, 8	—	0	R	**Reserved**
7 to 2	BLUE	0	R/W	**Blue** These bits contain Blue palette data.
1, 0	—	0	R	**Reserved**

GE Tiled and Linear Modes:

CP000R to CP255R

Address: CP000R = H'100, Address: CP001R = H'102, Address: CP255R = H'2FE.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									RED							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GREEN								BLUE							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	0	R	**Reserved**
23 to 18	RED	0	R/W	**Red** These bits contain Red palette data.
17, 16	—	0	R	**Reserved**
15 to 10	GREEN	0	R/W	**Green** These bits contain Green palette data.
9, 8	—	0	R	**Reserved**
7 to 2	BLUE	0	R/W	**Blue** These bits contain Blue palette data.
1, 0	—	0	R	**Reserved**

Color Palette Read Registers (CPRR H/L)

These register are read only and hold color palette data from the last color palette read access.

Q2SD Mode:

CPRR H

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RED							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 2	RED	0	R	**Red** These bits hold Red palette data from the last color palette read access.
1 to 0	—	0	R	**Reserved**

CPRR L

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GREEN								BLUE							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 10	GREEN	0	R	**Green** These bits hold Green palette data from the last color palette read access.
9, 8	—	0	R	**Reserved**
7 to 2	BLUE	0	R	**Blue** These bits hold Blue palette data from the last color palette read access.
1, 0	—	0	R	**Reserved**

GE Tiled and Linear Modes

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									RED							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GREEN								BLUE							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	0	R	**Reserved**
23 to 18	RED	0	R	**Red** These bits hold Red palette data from the last color palette read access.
17, 16	—	0	R	**Reserved**
15 to 10	GREEN	0	R	**Green** These bits hold Green palette data from the last color palette read access.
9, 8	—	0	R	**Reserved**
7 to 2	BLUE	0	R	**Blue** These bits hold Blue palette data from the last color palette read access.
1, 0	—	0	R	**Reserved**

10.3 Functional Description

The complete functionality is described by the following sub-functions:

- Memory Architecture
- Sync Pulse Generator.
- Foreground
- Background + wrap function
- Picture in Picture (Video Window)
- Cursors
- PLL
- Q2SD Compatibility.

10.3.1 General Functionality

The Display Output module reads two planes of video or graphics pixel data from the external RAM, which it can blend together. These planes can be combined with two hardware cursors and output to a TFT display as 18-bit digital RGB data.

The memory will be accessed as 32-bits per clock cycle and under the worst case, which will also be the most common case, there will be a set-up to the access of 7 clocks. If data is fetched in 64 word bursts, the efficiency will be 90%.

The peak bandwidth is 4×99 MHz = 396 Mbytes/sec.

Resolution		Refresh	Pixel Depth		Bandwidth	Bandwidth
Hpixels	Vpixels	(Hz)	(Bits)	Planes	(Mbytes/sec)	(%)
320	200	75	16	1	10	3
320	200	75	16	2	19	5
480	234	50	16	1	11	3
480	234	50	16	2	22	6
400	240	50	16	1	10	3
400	240	50	16	2	19	5
480	320	75	24	2	69	19
600	240	50	16	1	14	4
640	480	50	16	2	61	17
800	450	50	16	1	36	10
800	450	50	16	2	72	20
1024	768	50	16	2	157	44

All digital RGB data is output as 18-bit RGB data. When using the color palette the RGB data is defined in 18 bits, 6 bits for red, green and blue. When pixel data is input in 16-bit RGB format, it is converted to a 18-bit data output by mapping the MSB of red and blue data to their respective LSB data bits. See Diagram When connecting DO_Data to a 16-bit LCD display bits 12 and 0 are not connected.

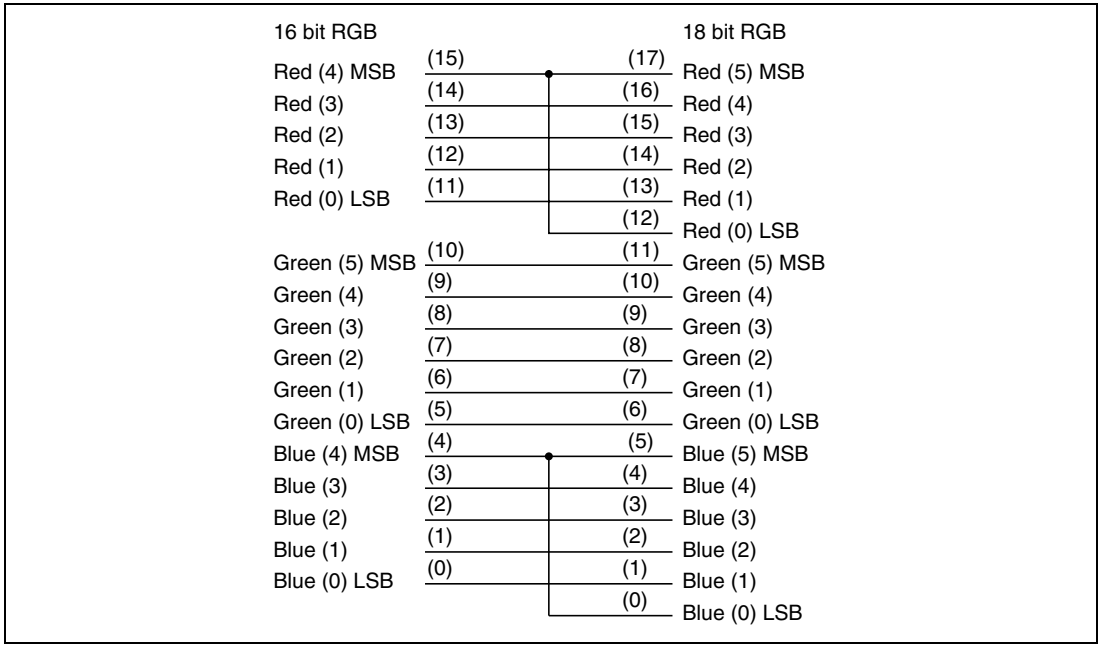


Figure 10.2 Connection of 16-Bit RGB to 18-Bit RGB

10.3.2 Sync Pulse Generator

The sync pulse generator is used to generate all the sync signals for the TFT display. It will also be able to receive external sync signals and output display data. All timings for sync signals are fully programmable and can support TFT displays up to 1024 × 768* pixels. The sync pulse generator will also control the timing of display out data.

Note: * The display size counter can handle as large as 1024 × 768, but this does not mean that HD64404 can display 1024 × 768 in all possible configurations and conditions. The dot clock frequency is up to 50 MHz.

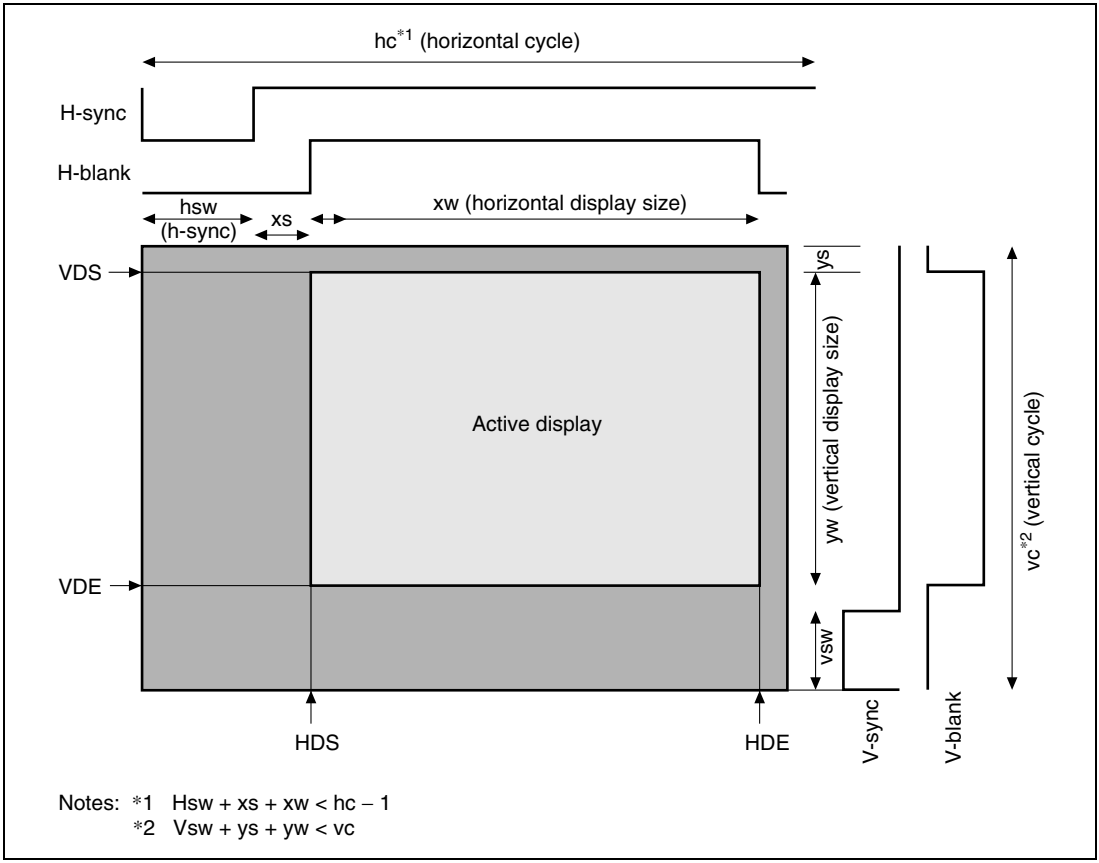


Figure 10.3 TFT Display

Equations for SPG Register Settings:

When Master mode,

HDS = HSW-1 + XS.
HDE = HSW-1 + XS + XW.
VDS = YS-2.
VDE = YS-2 + YW.
HSWR = HSW-1.
HCR = HC-1.
VSP = VC – VSW-1.
VCR = VC-1

When TV sync mode,

HDS = HSW -4 + XS
HDE = HSW -4 + XS + XW.

10.3.3 Memory Architecture

The memory architecture for each plane is individually controllable and can be set to tiled or linear mode, (configured using the ECR register). In Linear mode pixels are stored in memory in order starting from the top left pixel. When the end row one has been reached the next row down starts with it's left most pixel. The difference in address from the start of one consecutive row to the start of the next is known as the stride. (See diagram for more details).

Tiled memory architecture is used by the renderer module, and hence must be used when displaying any plane generated using the renderer. In tiled mode pixels are stored in memory in 32-byte × 16-row tiles. The number of pixels in each tile is therefore dependent on the bits per pixel format that the plane uses. The diagram shows addresses incrementing using 16 bits per pixel, hence each tile is made up from 16 × 16 pixels. Addresses increment linearly within each tile with the first pixel in row 1 having the consecutive address from the last (15^{th}) pixel in row0. The 1^{st} pixel in the tile1 will have the consecutive address from the last pixel in row15 of tile 0. The stride in tile mode is defined as the difference in address between the top left pixel in one tile to the start address on the next top left pixel in the tile below, divided by sixteen, the number of rows in each tile, (See diagram for more details).

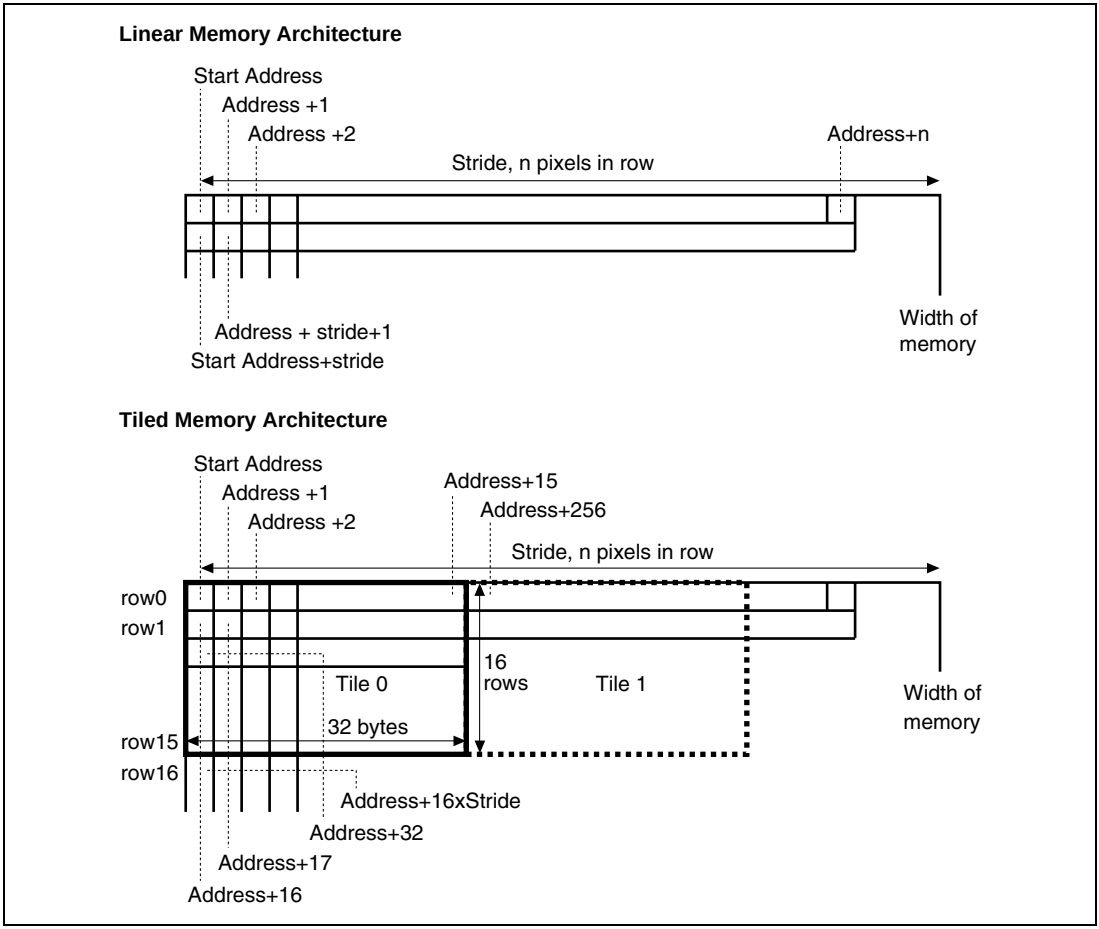


Figure 10.4 Memory Arohitecture

10.3.4 Foreground

Foreground display data is to be used to display overlay text and graphics over a background. Foreground data can be output either as a single plane to the display, or mixed using chroma key and/or alpha blending with the background.

Foreground data to be displayed will be defined using a start address, number of pixels in line, number of lines and the horizontal stride. The foreground plane must be the same size as the display size and no hardware image scrolling is possible with this plane.

10.3.5 Background and wrap function

Background display data is to be used as the background plane, to display map and video data behind a foreground. Background data can be output either as a single plane to the display, or alpha blended with the foreground. Part or all of the Background plane can be replaced with video data from a separate address using the picture in picture (video) function.

Background data to be displayed will be defined using a start address, number of pixels in line, number of lines and the horizontal stride. In linear mode, an image of maximum size of 4096 x 4096 pixels can be stored in one background frame buffer. An area of the large image, the size of the display screen can then be displayed. By giving a pixel co-ordinate, the screen will display the area of the image to the bottom right of the start co-ordinate.

The Wrap Function allows map data etc. to be scrolled on screen without the overhead of copying large amounts of background data. Using the wrap function, when the display window reaches the edge of background memory, the address wraps around and displays pixel data from the opposite side of the image. For example, if a display was scrolling to the right, as the display window approaches the right side of the background memory, data that will continue the map in the right direction will be written to the left side of the background memory. When the display window reaches the right edge, the map will be able to scroll by automatically wrapping and displaying the left side of the background memory.

A wrap function is available by setting the Wrap-around Combination bits in the DO_DSMR Register. When the Q2SD bit in the DO_SYSR Register is set to '1', the wrap operation will be binary compatible with the Q2SD. The background planes shall be either 1024 or 512 pixels horizontal and 512 pixels vertical and the start address is controlled using the LBGSY Register. (See Q2SD HD64413A user manual).

When the Q2SD bit in the DO_SYSR Register is set to '0' (GE mode), the extended wrap function becomes available which has two background plane buffers, A and B. When in GE Tiled mode, the background wrap-around plane size is limited in both X and Y directions to either 512 or 1024 pixels. The X direction size is controlled by MWX and Y direction size by DO_DSMR bits 14:13. In GE Tiled mode, LBGSAR and RBGSAR set the A and B background start address. This value must be tile aligned within memory.

If Linear Addressing is used for the background plane, the wrap-around plane size can be up to 4096×4096 pixels with a fully programmable vertical size (WRPY) and a horizontal size set by MWX + MWXOFS. The position of the displayed background area is set for both GE Tiled and linear modes using horizontal and vertical pixel co-ordinates (LBGSX and LBGSY).

While wrap around display is performed, X and Y of the rendering coordinates must not exceed the double of the memory width and vertical wrap around size respectively.

Diagram showing Wrap Function Co-ordinate and Address Data. (Only Background A Linear mode shown.)

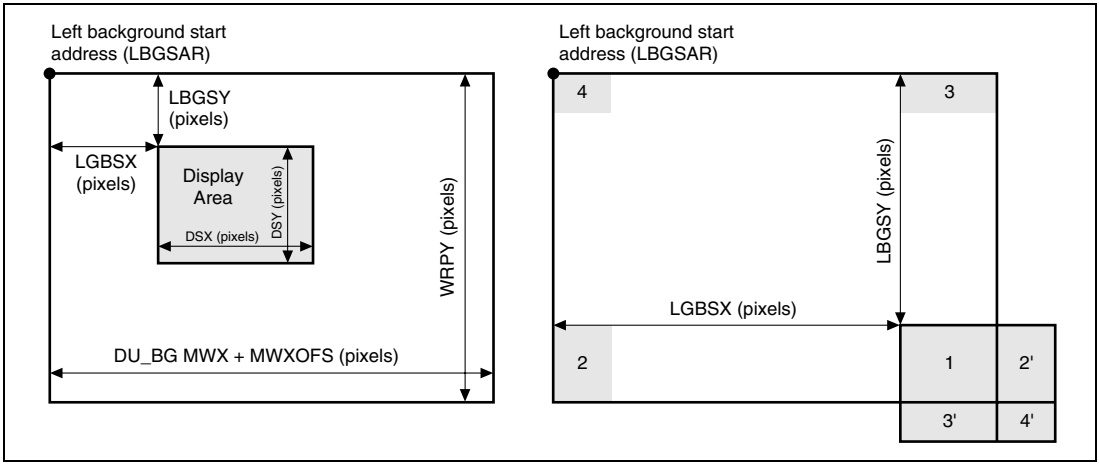


Figure 10.5 Wrap Function Co-ordinate and Address Data

10.3.6 Picture in Picture

Picture in picture (Video Window) describes a method of replacing all or part of Background with video data from a separate memory location. Picture in picture data will be defined using a start address, start pixel, start line number of pixels in line, number of lines and the horizontal stride. This will allow a movable window of video data to be located within the background image. Picture in picture can not be blended with the Background, but simply replaces Background data within the RAM FIFO's before it is blended with the Foreground. Picture in picture is binary compatible with the Q2SD video window display function when in Q2SD mode.

10.3.7 Alpha Blending

Mixing the two planes is controlled by a 4-bit value set in the DO_DBR. When 4'H0 is set, only the Background will be output to the display, likewise when 4'HF is set only Foreground will be output. All other values will produce a mixed plane output based on the equation:

$$P_{out} = \frac{(P_{DO1} \cdot \alpha + P_{DO2} \cdot (\alpha_{max} - \alpha))}{\alpha_{max}}$$

Where P = Red, green or blue 6-bit pixel data and α = 4-bit data register value and $\alpha_{max} = 16$.

Separate alpha values are defined for mixing Foreground with Background, and when mixing Foreground with PIP data. This can be used to implement a different transparency when using graphics over a background image to when using graphics over a video picture. By setting the PIP alpha value (PIPAB) to 4'H0, it is possible to prevent Foreground graphics from overwriting video data.

Alpha blending is not effective when both Foreground and Background planes are configured as 8 bit/pixel. In this case, the foreground plane is displayed.

10.3.8 Chroma-Key

The Chroma key function is used to enable Foreground text and graphics to be displayed over a background, without alpha blending the two planes together. The function operates by comparing foreground pixel data with the color stored in the Transparent-color register. When the foreground data matches the Chroma-color data, only background is displayed, otherwise the result from the alpha blender is displayed.

By setting the Chroma-Color Register to that of the foreground planes background color, and setting the background alpha value (LBGAV and RBGAV) to 4'HF, foreground text can be displayed over a the background.

The data in the Display off output register (DO_DOORH/L) is displayed when both PIP and background planes are off and both foreground plane and cursor contain transparent color.

10.3.9 Cursors

Two independent hardware cursors are to be implemented within the Display Out module. They will each consist of either a 64×64 pixel bitmaps or a 32×32 pixel bit map which can be defined as 8 bits/pixel using the color palette. A Chroma-key function is available for both cursors, which enables one of the color palette colors to be output transparent.

Both cursors are able to have two bitmaps which they can 'blink' for one to another, with the time each bit map is displayed being fully programmable. It is also possible for each cursor to blink between just one bit map and transparent so that the cursor will appear to flash on and off.

If both cursors should be placed with the same location within the display, Cursor1 will always be placed in front of Cursor2.

10.3.10 Q2SD Compatibility

The Display Out module will be binary compatible with the majority of Q2SD display out functions. When the Q2SD bit in the DO_SYSR Register is set to '0', only registers marked by * should be read or written to, and the Display Out module should operate with Q2SD software. The Q2SD spec referenced within this document is "HD64413A Q2SD User Manual, Rev 1.0, 9/21/99".

By setting bits in the DO_ECR Register, further extended display out functionality becomes available. There will be little of the Q2SD compatible logic redundant in this mode, as virtually all register and counter logic will be used by the extended mode. Many of the Q2SD compatible functions will use exactly the same hardware, but will use hardwired register settings to achieve

Q2SD software compatibility. When using Q2SD software with the Display Output module, the Q2SD address will have to be shifted up by 1 bit as the display out module uses long word read/write accesses only.

10.3.11 PLL

The Display Out module will contain a PLL to output the Display Out clock. This will be used to generate the sync signals and output the image data at the correct frequency. Three 4 bit external dividers (pre, feedback and post dividers) will be used with the PLL to produce the correct frequency.

The table below gives recommended settings for a range of Dot Clock frequencies. All values are gives in actual division ratios. When loading the PLL Register, 1 must be subtracted from the values given for N, A, B and C for the value to be loaded into the register. PSEL should be set to "1" or "2" for all cases below.

The dot clock frequency is set depending on the size of the display being used. For the relation between dot clock and the display size, please see Section 1 Overview.

Formula of obtaimed dot clock:

There are two constraints for the displayout module PLL.
RFCLK (reference clock for VCO) range has to be between 8MHz and 50MHz.
VCO (Voltage Control Oxcillator) in PLL has to be between 50MHz and 200MHz.

- For PCI, RFCLK = PCI clock/A
- For MPX, PFCLK = CKIO/A
- VCO = RFCLK $\times 2^{P} \times$ B $\times$ N
- Dot clock = VCO/ $(2^{P} \times$ C) = RFCLK $\times$ B $\times$ N/C

Dot Clock Generated from 33MHz clock input

Input Clock PCI Mode (MHz)	N	A	B	C	P	Obtained Dot Clock (MHz)
33	1	2	1	3	2	5.50
33	1	2	1	2	2	8.25
33	1	3	3	3	1	11.00
33	1	2	3	3	1	16.50
33	1	3	4	2	1	22.00
33	1	2	3	2	1	24.75
33	1	3	5	2	1	27.50
33	1	1	2	2	1	33.00
33	1	3	7	2	1	38.50
33	1	2	5	2	1	41.25

Dot Clock Generated from 66MHz clock input

Input Clock MPX Mode (MHz)	N	A	B	C	P	Obtained Dot Clock (MHz)
66	1	5	1	2	2	6.60
66	1	2	1	4	2	8.25
66	1	8	3	2	2	12.38
66	1	8	2	1	2	16.50
66	1	3	1	1	2	22.00
66	1	5	2	1	1	26.40
66	1	1	1	2	1	33.00
66	1	8	9	2	1	37.13
66	1	5	3	1	1	39.60

Dot Clock Generated from 78MHz clock input

Input Clock MPX Mode (MHz)	N	A	B	C	P	Obtained Dot Clock (MHz)
78	1	6	1	2	2	6.50
78	1	8	3	4	1	7.31
78	1	2	1	4	1	9.75
78	1	8	5	4	1	12.19
78	1	8	3	2	1	14.63
78	1	4	1	1	2	19.50
78	1	7	2	1	2	22.29
78	1	3	1	1	2	26.00
78	1	5	4	2	1	31.20
78	1	8	7	2	1	34.13
78	1	2	1	1	1	39.00

Dot Clock Generated from 83MHz clock input

Input Clock MPX Mode (MHz)	N	A	B	C	P	Obtained Dot Clock (MHz)
83	1	6	1	2	2	6.92
83	1	8	3	4	1	7.78
83	1	2	1	4	1	10.38
83	1	8	5	4	1	12.97
83	1	8	3	2	1	15.56
83	1	2	1	2	1	20.75
83	1	6	7	4	1	24.21
83	1	3	2	2	1	27.67
83	1	4	3	2	1	31.13
83	1	3	4	3	1	36.89
83	1	2	1	1	1	41.50

Dot Clock Generated from 88MHz clock input

Input Clock MPX Mode (MHz)	N	A	B	C	P	Obtained Dot Clock (MHz)
88	1	7	1	2	2	6.29
88	1	3	1	4	1	7.33
88	1	6	2	3	1	9.78
88	1	5	2	3	1	11.73
88	1	3	1	2	1	14.67
88	1	8	3	2	1	16.50
88	1	1	1	4	1	22.00
88	1	6	7	4	1	25.67
88	1	3	1	1	1	29.33
88	1	4	3	2	1	33.00
88	1	7	3	1	1	37.71

Dot Clock Generated from 100MHz clock input

Input Clock MPX Mode (MHz)	N	A	B	C	P	Obtained Dot Clock (MHz)
100	1	8	1	2	2	6.25
100	1	3	1	4	1	8.33
100	1	5	1	2	2	10.00
100	1	2	1	4	2	12.50
100	1	8	5	4	1	15.63
100	1	6	1	1	2	16.67
100	1	6	5	4	1	20.83
100	1	1	1	4	1	25.00
100	1	8	7	3	1	29.17
100	1	1	1	3	1	33.33
100	1	4	3	2	1	37.50
100	1	6	5	2	1	41.67

10.3.12 Reset Strategy

All registers will be equipped with an asynchronous reset.

10.3.13 Power Saving and Clocking Strategy.

The module can be powered down by first setting the module enable bit in the PLL register to '0'. After the following vblank interrupt all planes will be disabled, DMA accesses stopped and I/O ports will be set to inputs. The dot clock can then be stopped by setting the PLL clock enable bit to '0'. Only after module disable vblank and the dot clock is stopped can the rbclk and pix_clk be stopped using the Power Control & Configuration module.

Section 11 GE for HD64404

11.1 Overview

11.1.1 Overview

The GE is a 2-dimensional graphics accelerator module for HD64404.

11.1.2 Block Diagram

Command Fetch Unit: Performs fetching the display list from buffer for Q2SD/RU, interprets the display list, and passes drawing parameters to Q2SD/RU or 2DGE.

2D Graphics Engine (2DGE): Performs anti-alias font drawing and Bit Block Transfer (BitBLT) with 16 raster operations.

Buffer for 2DGE: Reads reference data for 2DGE drawing from frame memory via Pixel Bus, and writes output drawing data into frame memory via Pixel Bus.

Q2SD Rendering Unit (Q2SD/RU): Performs Q2SD's legacy drawing commands.

Buffer for Q2SD/RU: Reads display list or reference data for Q2SD/RU drawing from frame memory via Pixel Bus, and writes output drawing data into frame memory via Pixel Bus.

Note: Two graphics modules (Q2SD/RU, 2DGE) have the command fetch unit in common. Once the command fetch unit passes drawing parameters to one or the other graphics modules, it waits until the graphics module completes drawing. So two graphics modules do not operate in parallel.

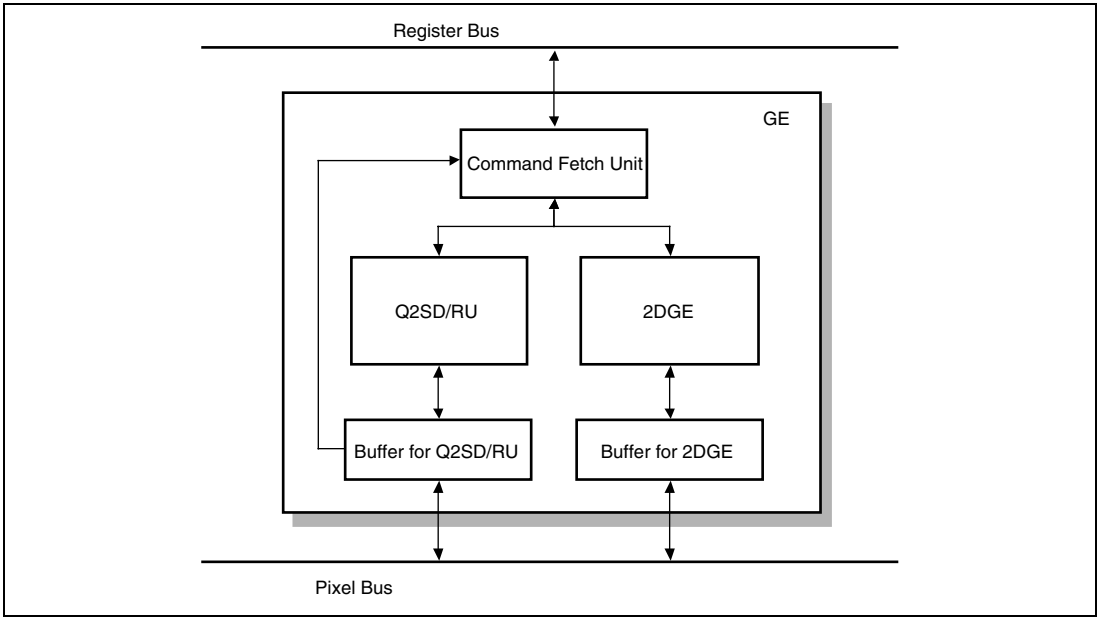


Figure 11.1 GE Block Diagram

11.1.3 Drawing Functions

(1) Q2SD Drawing Functions

The Q2SD/RU supports all Q2SD drawing functions.

Notes:

1. Double Buffer Switching

   The HD64404 does not provide the functions like "Auto Display Change Mode" and "Auto Rendering Mode" which are used in the Q2SD. As a double buffer switching does not happen automatically, it should be controlled by software. CPU can switch the display and drawing areas alternately by setting the display and drawing base addresses in the corresponding registers (Display Out: Display Start Address Register, GE: Rendering Start Address Regiser (RSAR)). CPU can know the appropriate switch timing by checking status flags (vertical Blanking Flag (VBK), Trap Flag (TRA)) or interrupt initiated by status flags. This switching method is similar to the Q2SD's "Manual Display Change Mode".

2. WPR Command

   WPR Command can not write the parameters into Display Start Address Register in Display Out module.

3. Rendering Mode Register (RMR)

   The HD64404 has the exclusive Memory Width X (MWX) bit and Graphic Bit Mode (GBM) bit for drawing in Rendering Mode Register (RMR), while these bits are used for display and drawing in common in the Q2SD.

   Rendering Mode Register (RMR) dose not have Rendering Start Address Enable (RSAE) bit. The value set in Rendering Start Address Register (RSAR) is always used as a base address of drawing area, and the value set in Display Start Address Register is not used.

(2) Anti-alias Font Drawing

2DGE expands a_value font (4-bit/pixel) to color by the following equation. This function is used to draw anti-alias fonts. 2DGE anti-alias font drawing supports 16-bit/pixel color format.

pixel ← foreground_value*a_value + destination*(1 - a_value)

foreground_value: foreground color (16-bit / pixel) , a_value (4-bit/pixel), destination: destination data (16-bit / pixel)

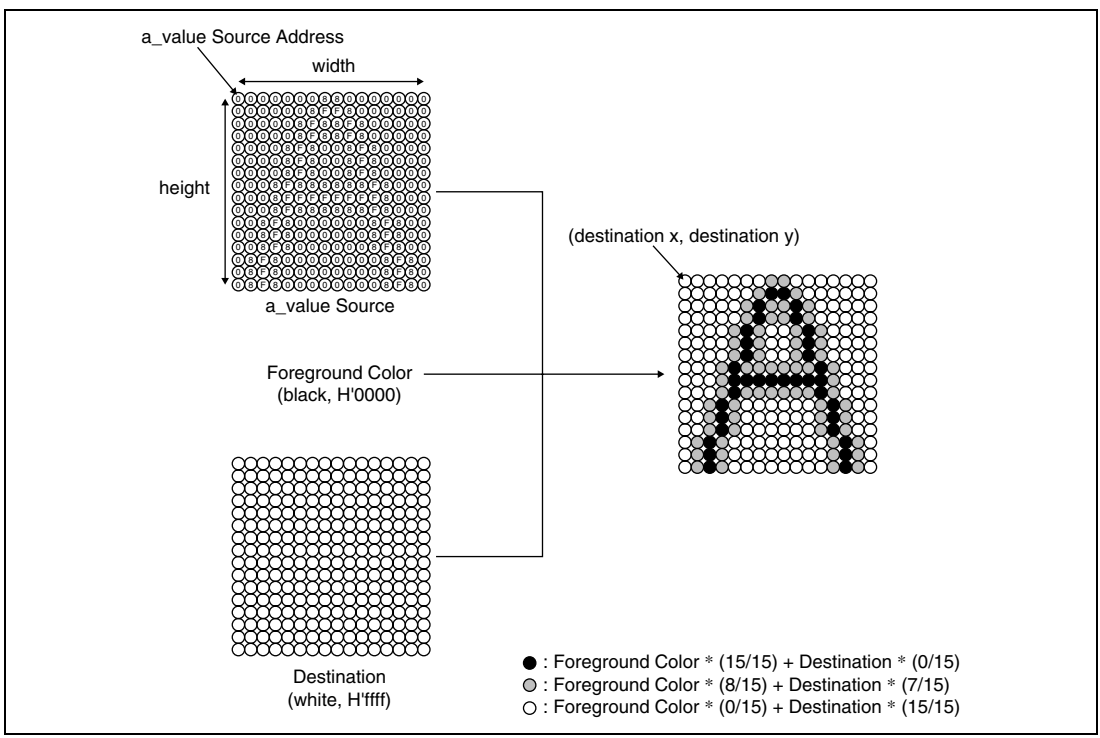
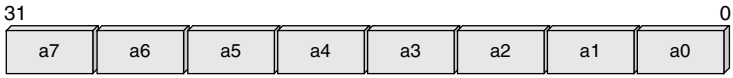
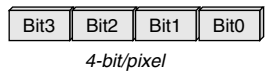


Figure 11.2 Anti-alias Font Drawing



	Bit 3	Bit 2	Bit 1	Bit 0
0/15	0	0	0	0
1/15	0	0	0	1
2/15	0	0	1	0
3/15	0	0	1	1
4/15	0	1	0	0
5/15	0	1	0	1
6/15	0	1	1	0
7/15	0	1	1	1
8/15	1	0	0	0
9/15	1	0	0	1
10/15	1	0	1	0
11/15	1	0	1	1
12/15	1	1	0	0
13/15	1	1	0	1
14/15	1	1	1	0
15/15	1	1	1	1



4-bit/pixel

Note: a_value is an approximate value.

Figure 11.3 a_value

(3) Bit Block Transfer (BitBLT) with 16 Raster Operations

Bit Block Transfer (BitBLT): A rectangular area of pixels is moved from one location to another. Stretch BitBLT is not allowed. 2DGE BitBLT supports both 8-bit/pixel and 16-bit/pixel color formats.

2-operand (source, destination) Raster Operation: 2DGE supports 16 raster operations.

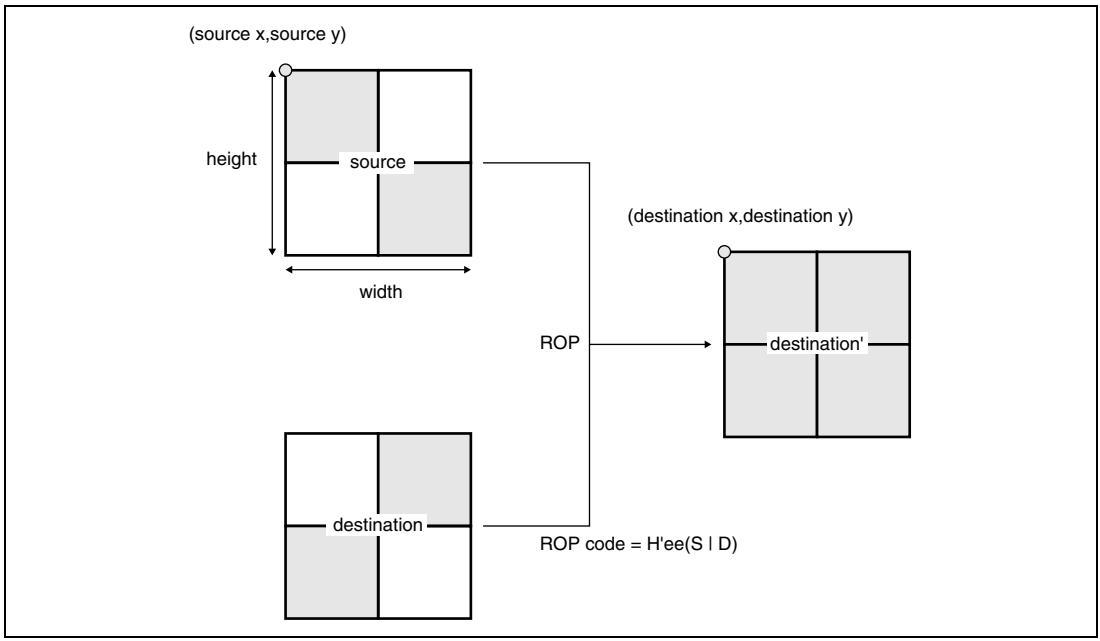


Figure 11.4 BitBLT with ROP

Source/Destination Color Transparency: Destination data are not written when these data are equal to Source Transparent Color Register value or Destination Transparent Color Register value.

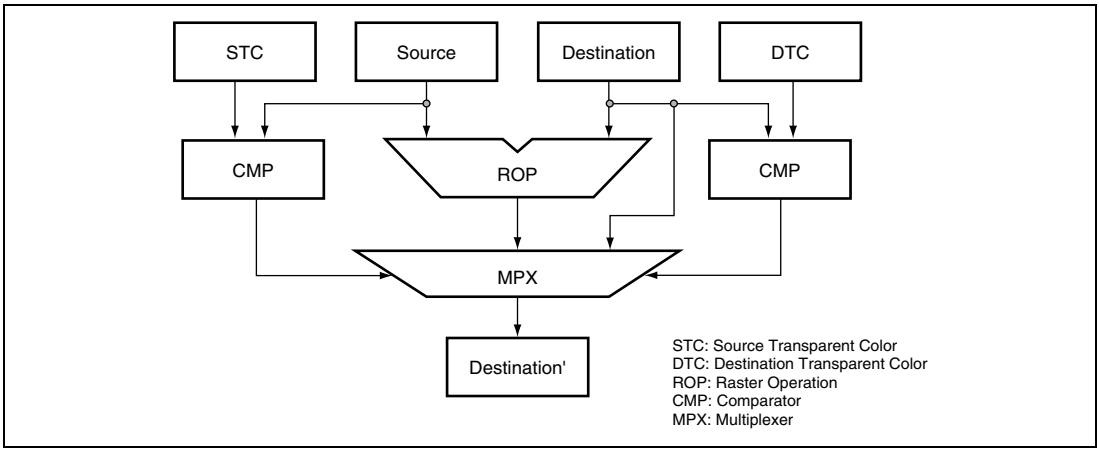


Figure 11.5 Color Transparency

11.1.4 Module Standby Mode

The Graphics Engine module allows a clock gating to reduce power consumption. Both Pixel Bus and Register Bus clocks can be gated. This module standby mode can be controlled by Clock Control 2 Register in Power Control module.

The following procedure is required to power down the Graphics Engine module.

- Confirm that the Graphics Engine module has already terminated the command operation. CPU can know the end of the command operation using the TRAP flag bit in the Status Register (SR).
- Set the Software Reset (SRES) bit in the Rendering Control Register (RCR) to 1.
- Disable REND bit in Clock Control 2 Register.

The register contents are retained.

The following procedure is required to wake up the Graphics Engine module.

- Enable REND bit in Clock Control 2 Register. After enabling REND bit, all accesses to Graphics Engine module are valid.

11.2 Basic Functions

11.2.1 Coordinate Systems

The Q2SD/RU and the 2DGE have coordinate systems independently.

The Q2SD/RU has four 2-dimensional coordinate systems (screen coordinates, rendering coordinates, multi-valued source coordinates, and work coordinates), and one 1-dimensional coordinate system (binary source coordinates).

The 2DGE (2-dimensional Graphics Engine) has three 2-dimensional coordinate systems (screen coordinates, rendering coordinates, and multi-valued source coordinates), and one 1-dimensional coordinate system (a_value source coordinates).

Screen coordinates are display control coordinates. Screen coordinate X corresponds to the horizontal dimension of the display screen, and Y to the vertical dimension, and the origin is at the upper left of the display screen. The screen coordinate positive directions are right for the X-axis and down for the Y-axis. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of one screen coordinate.

Rendering coordinates are drawing control coordinates. Rendering coordinates are shifted horizontally and vertically with respect to screen coordinates by the offset amounts specified in drawing commands. Drawing commands perform drawing operations using these coordinates. However, drawing commands that specify clipping use screen coordinates. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of one rendering coordinate.

Multi-valued source coordinates are drawing control coordinates. When a drawing command is executed, these are the source (rectangle) coordinates specified by the command. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of one multi-valued source coordinate.

Binary source coordinates are drawing control coordinates. When a drawing command is executed, these are the source data (1-dimensional) coordinates specified by the command. The data width of one binary source coordinate is 1 bit (1-bit/pixel). For one binary source, one physical address (top-left) and the horizontal width and vertical height of the binary source are specified.

Work coordinates are drawing control coordinates that correspond one-to-one with the screen coordinates. When a drawing command is executed, these are the work coordinates specified by the command. The data width of one work coordinate is 1 bit.

a_value source coordinates are drawing control coordinates. When a drawing command is executed, these are a_value source data (1-dimentional) coordinates specified by the command. The data width of one a_value source coordinate is 4-bit (4-bit/pixel). For one a_value source, one physical address (top-left) and the horizontal width and vertical height of the a_value source are specified.

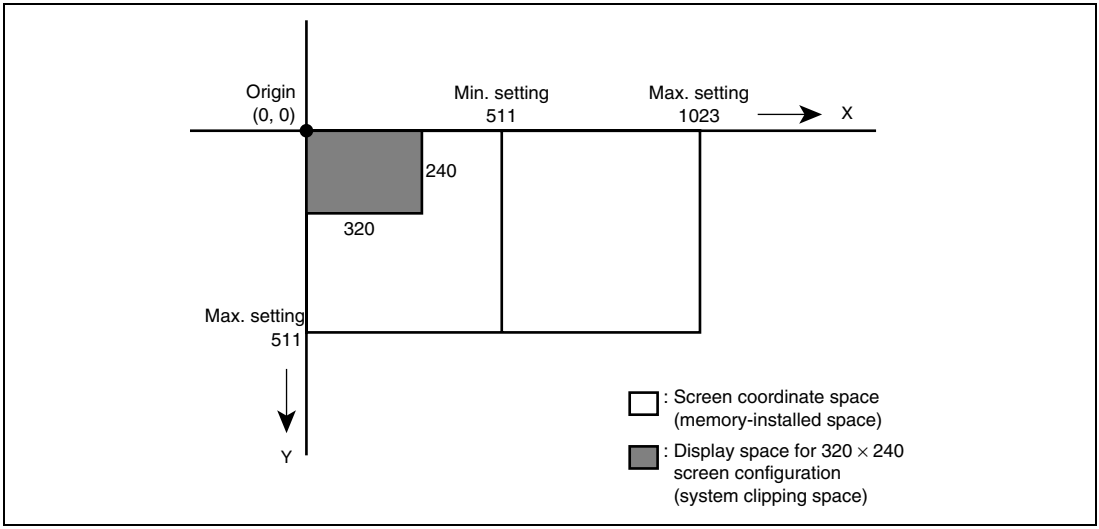


Figure 11.6 Screen Coordinates for Q2SD/RU

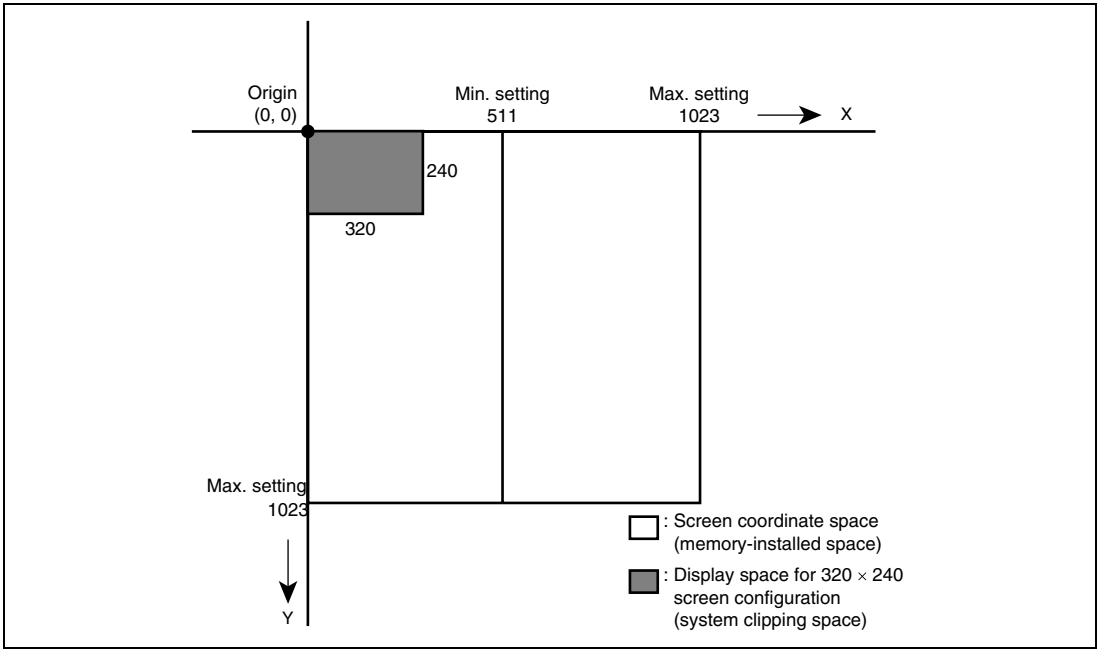


Figure 11.7 Screen Coordinates for 2DGE

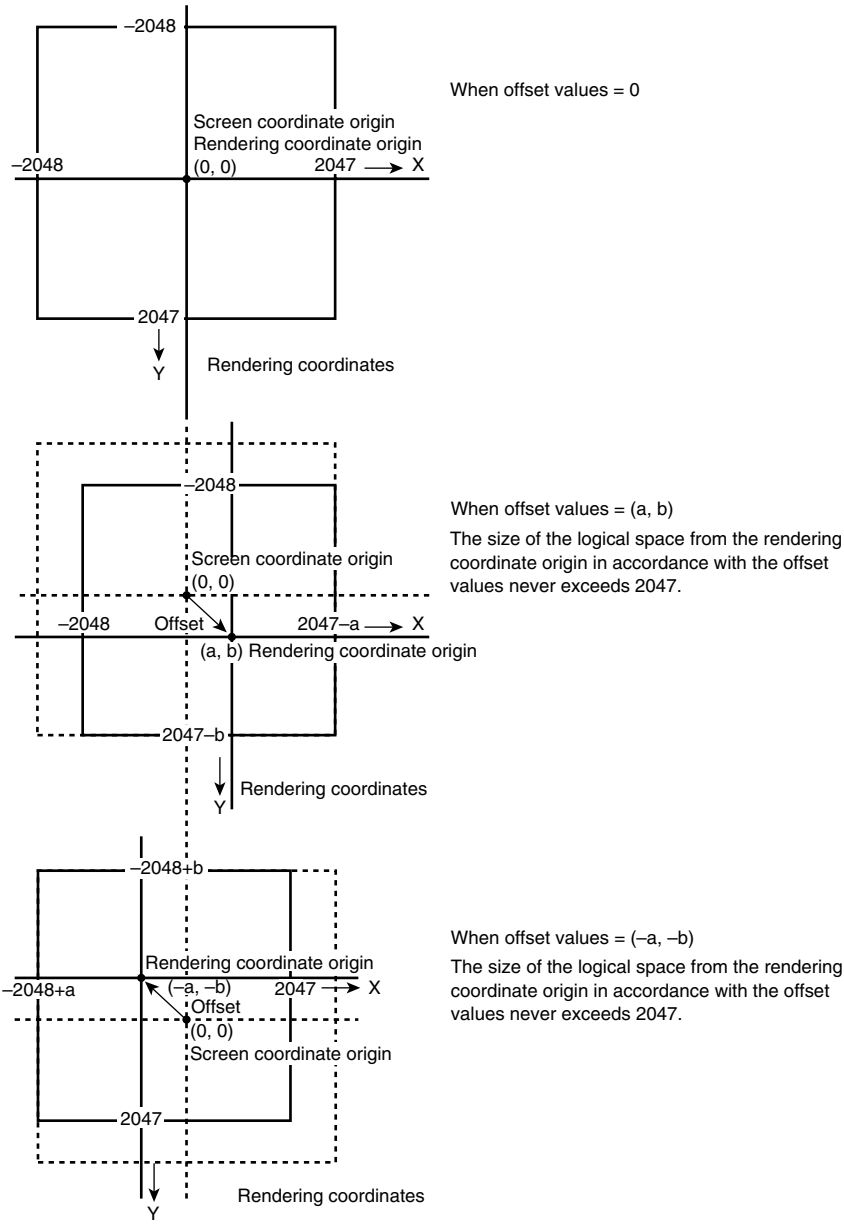


Figure 11.8 Rendering Coordinates for Q2SD/RU, 2DGE

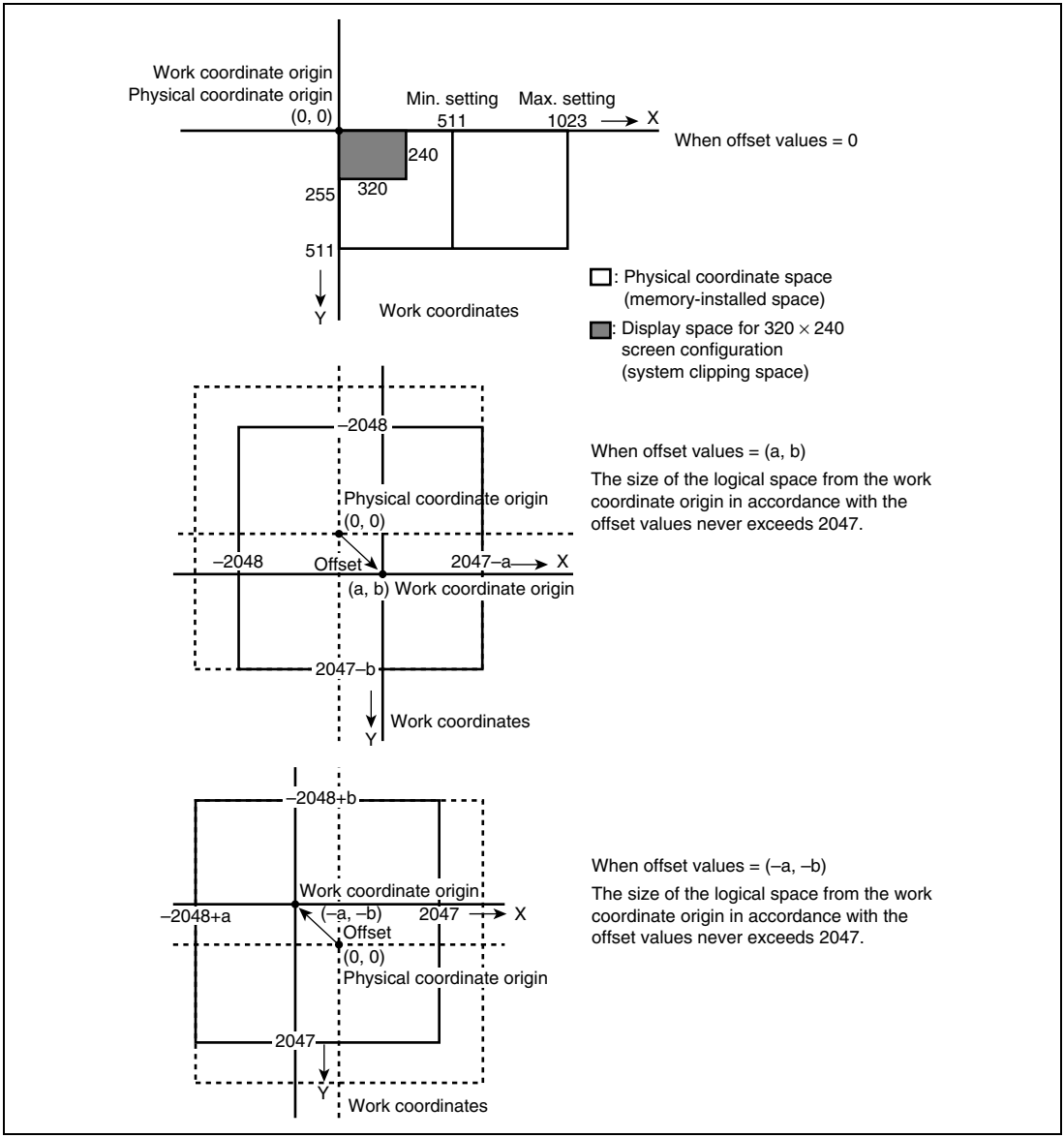


Figure 11.9 Work Coordinates for Q2SD/RU

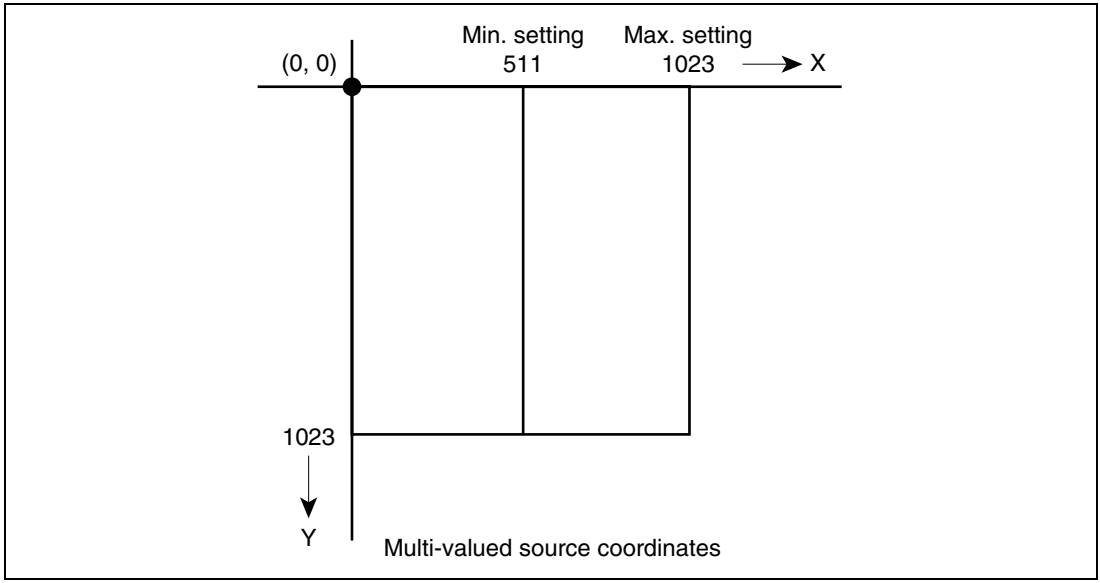


Figure 11.10 Multi-Valued Source Coordinates for Q2SD/RU, 2DGE

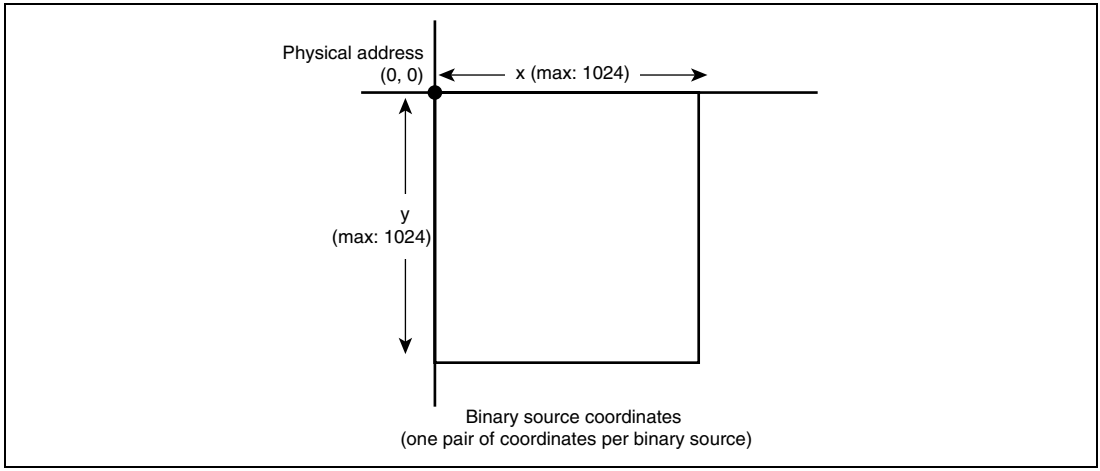


Figure 11.11 Binary Source Coordinates for Q2SD/RU

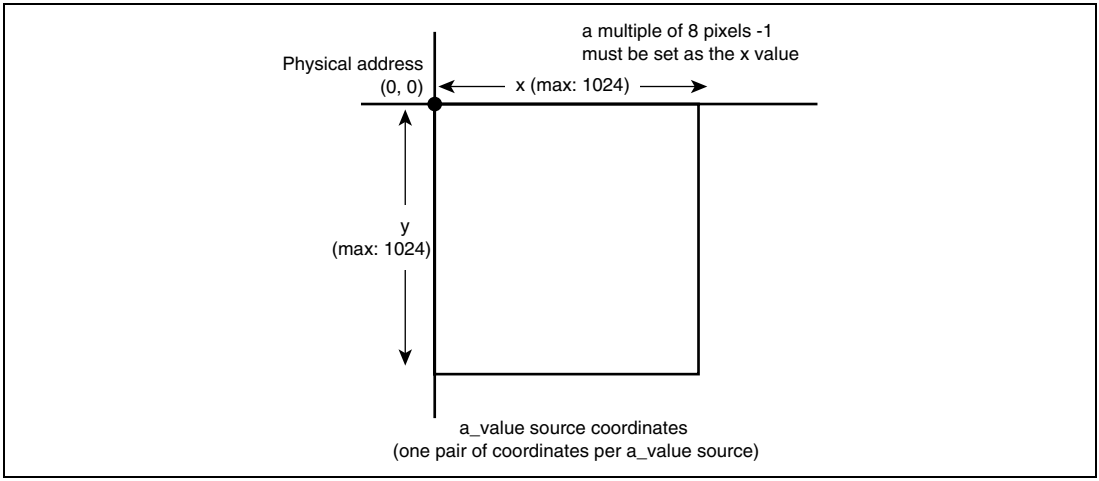


Figure 11.12 a_Value Source Coordinates for 2DGE

11.2.2 Clipping Area

The GE can perform a clipping. The Q2SD/RU block and the 2DGE block perform clipping area management independently. Therefore each block has the registers for clipping parameters.

There are two kinds of clipping: system clipping and user clipping. A system clipping area is always valid. A system clipping is always performed, and all destination pixels outside a system clipping area are not written. A user clipping area can be defined inside a system clipping area, and is valid when a user clipping enable bit is set. All destination pixels outside a user clipping area are not written when a user clipping is enabled.

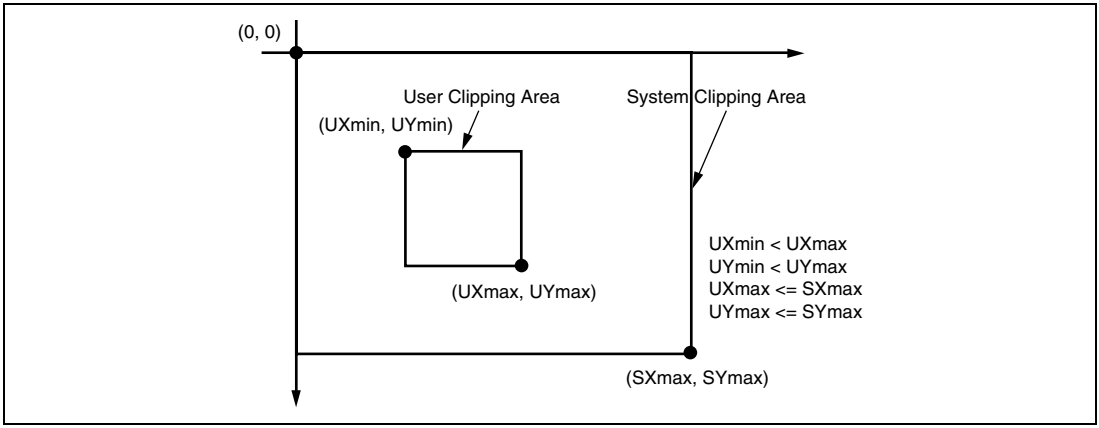


Figure 11.13 Clipping Area for Q2SD/RU, 2DGE

11.2.3 Pixel Data Format

The graphics memory includes the display list area, binary source area, work area, 8-bit/pixel source or 16-bit/pixel source area, 8-bit/pixel rendering or 16-bit/pixel rendering area, a_value area. The graphics memory is configured in 512-byte units, and a different memory configuration is used for each area. The memory configuration for each of the areas is shown in figure.11.14 to 11.17.

1-bit/pixel (work, binary source):

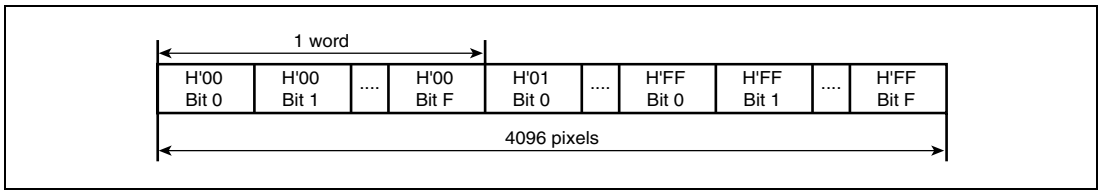


Figure 11.14 Configuration of One Memory Unit (512 Bytes) (1)

4-bit/pixel (a_value source):

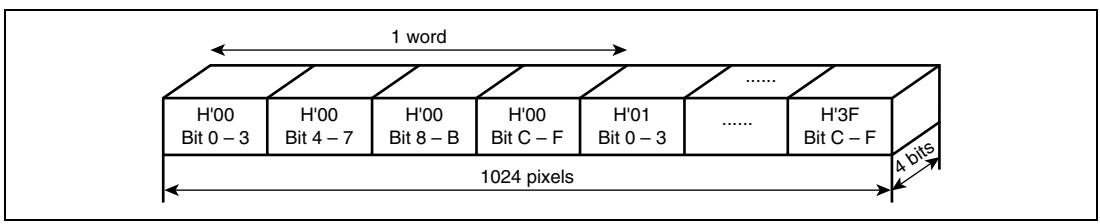


Figure 11.15 Configuration of One Memory Unit (512 Bytes) (2)

8 bits/pixel (multi-valued source, multi-valued destination):

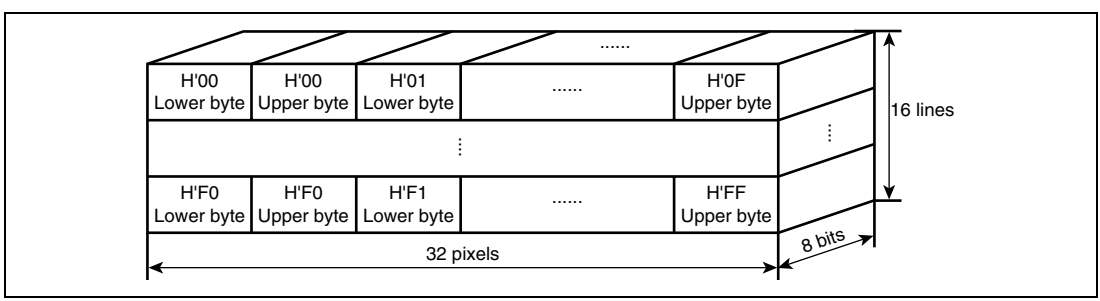


Figure 11.16 Configuration of One Memory Unit (512 Bytes) (3)

16 bits/pixel (multi-valued source, multi-valued destination):

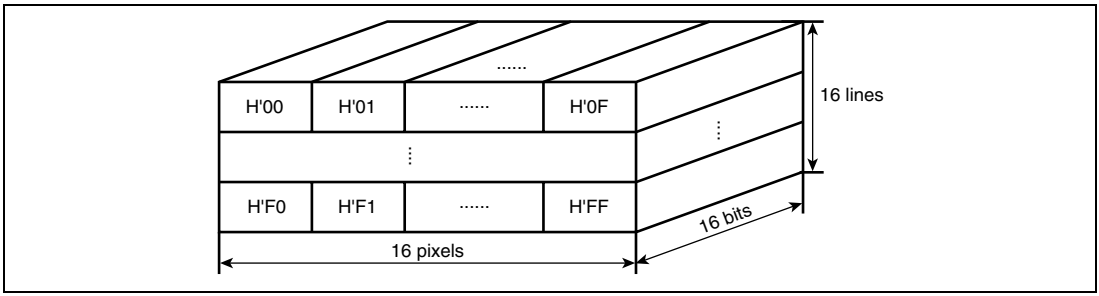


Figure 11.17 Configuration of One Memory Unit (512 Bytes) (4)

Display List: A display list, a group of drawing commands, is a 16-bit boundary data. The command fetch unit of GE fetches a display list and interprets it as a 16-bit boundary data.

11.2.4 Memory Map

(1) Tile addressing area

The graphics memory consists of addresses that are consecutive within one memory unit (linear addresses), as shown in figure 11.18.

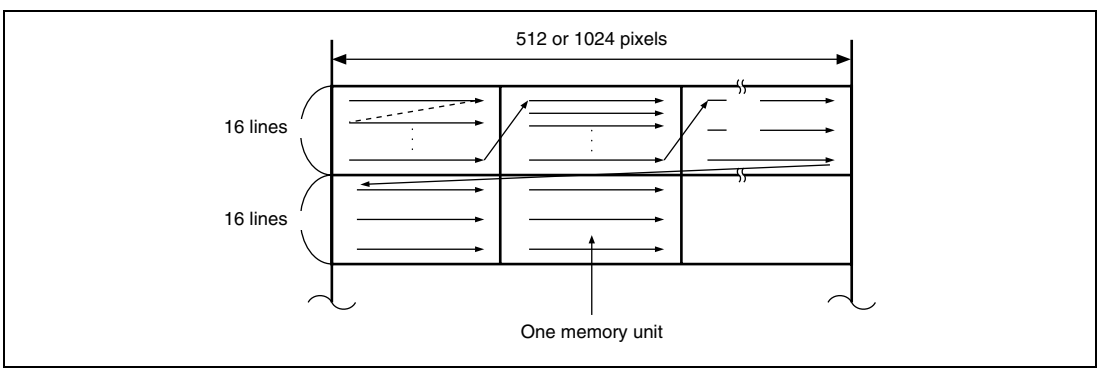


Figure 11.18 The graphics memory Address Transitions

Figure 11.19 shows a correspondence between memory physical addresses and logical coordinates. A combination of 8-bit/pixel and 16-bit/pixel areas can be used in the graphics memory, but area allocation must be carried out so that areas do not overlap.

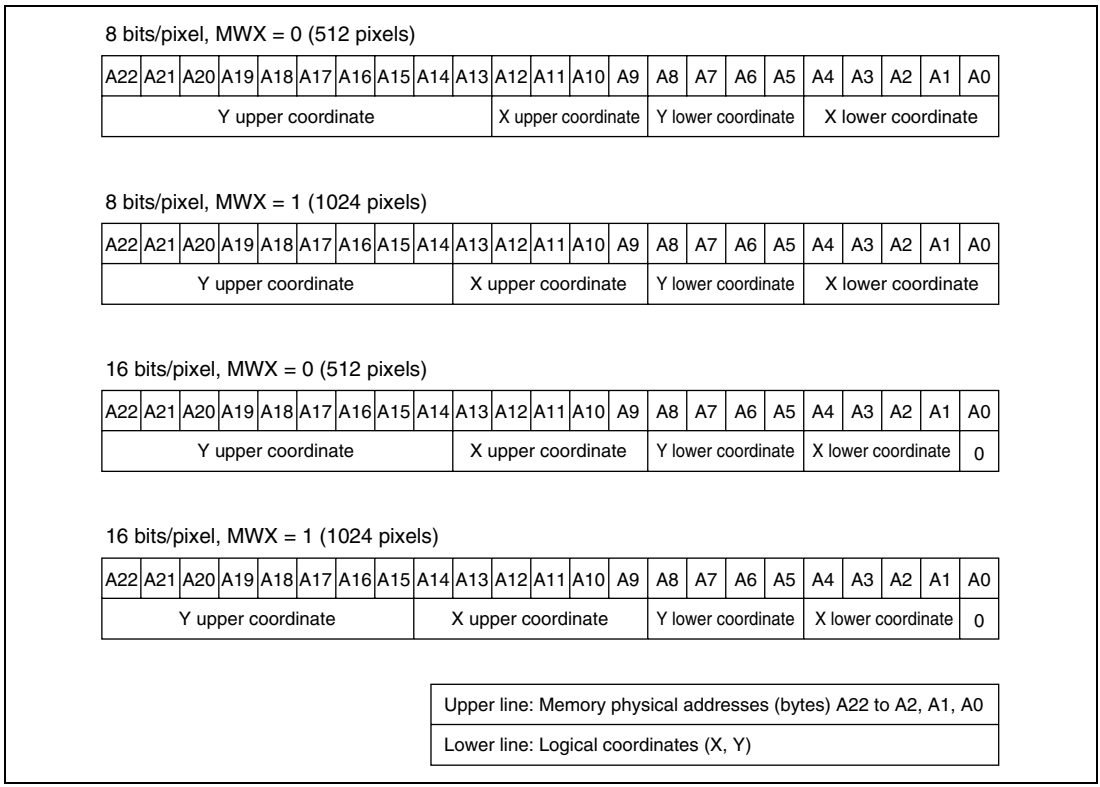
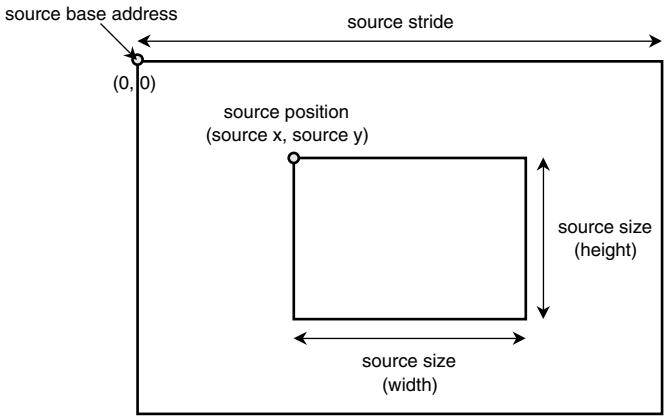


Figure 11.19 Correspondence between Memory Physical Addresses (Bytes) and Rendering Coordinates and Multi-Valued Source Coordinates (32-Bit Bus)

(2) Linear addressing area (2DGE Multi-valued Source only)



8-bit/pixel Source:

source start address = (source base address) + (source y) ' ((source stride) + 1) + (source x)

Note: Source start address and source base address are specified as a byte address.

16-bit/pixel Source:

source start address = (source base address) +{ (source y) ' ((source stride) + 1) + (source x)}' 2

Note: Source start address and source base address are specified as a byte address. But the lowest bit of the address is fixed to 0.

Figure 11.20 Linear addressing area (2DGE Multi-valued Source only)

11.3 Display List

11.3.1 Overview

The GE performs drawing on the basis of a group of drawing commands located in the graphics memory. This group of drawing commands is called a display list.

Table 11.1 lists the drawing commands. Table 11.2 lists the drawing command codes.

Table 11.1 Drawing Commands

Type	Command Name	Function
Four-vertex surface drawing	POLYGON4 Quadrilateral paint	Draws quadrilateral with four coordinates as vertices. Painting can be performed with source tiling and specified color.
	POLYGON4A	Four-vertex surface drawing with multi-valued source as transfer source
	POLYGON4B	Four-vertex surface drawing with binary source as transfer source
	POLYGON4C	Four-vertex surface drawing using specified color
Line drawing	LINE Polygonal line	Draws solid polygonal line from start coordinates through nodal coordinates.
	LINE	Polygonal line drawing (absolute coordinate specification)
	RLINE	Polygonal line drawing (relative coordinate specification)
	PLINE Polygonal line with line-type specification	Draws polygonal line with line type (pattern) from start coordinates through nodal coordinates.
	PLINE	Pattern-reference polygonal line drawing (absolute coordinate specification)
	RPLINE	Pattern-reference polygonal line drawing (relative coordinate specification)
Work surface drawing	FTRAP Trapezoid paint	Performs binary EOR painting of trapezoid with left side parallel to Y-axis.
	FTRAP	Binary EOR trapezoid fill (absolute coordinate specification)
	RFTRAP	Binary EOR trapezoid fill (relative coordinate specification)
	CLRW Rectangle zero-clear	Performs zero-painting of rectangle with diagonal designated by two coordinate points.

Type	Command Name	Function
Work line drawing	LINEW Polygonal line	Draws solid polygonal line from start coordinates through nodal coordinates.
	LINEW	Binary polygonal line drawing (absolute coordinate specification)
	RLINEW	Binary polygonal line drawing (relative coordinate specification)
Q2SD/RU Register setting	MOVE	Current pointer setting (absolute coordinate specification)
	RMOVE	Current pointer setting (relative coordinate specification)
	LCOFS	Local offset value setting (absolute coordinate specification)
	RLCOFS	Local offset value setting (relative coordinate specification)
	SCLIP	Sets rectangle with diagonal designated by origin and specified coordinate point as clipping area.
	UCLIP	Sets rectangle with diagonal designated by two coordinate points as clipping area.
	WPR	Sets a specific address-mapped register.
Sequence control	JUMP	Command sequence jump (branch)
	GOSUB	Subroutine call (branch)
	RET	Subroutine return
	NOP1	No operation: no processing executed.
	NOP3	No operation: no processing executed.
	VBKEM	Waits until vertical retrace line interval.
Drawing end	TRAP	Ends drawing processing and generates CPU interrupt.
2DGE Register setting	W2DP	Sets 2DGE registers

Table 11.2 Drawing Command Codes

CODE					COMMAND
0	0	0	0	0	POLYGON4A
0	0	0	0	1	POLYGON4B
0	0	0	1	0	POLYGON4C
0	0	0	1	1	test command
0	0	1	0	0	W2DP
0	0	1	0	1	(reserved)
0	0	1	1	0	(reserved)
0	0	1	1	1	test command
0	1	0	0	0	FTRAP
0	1	0	0	1	RFTRAP
0	1	0	1	0	LINEW
0	1	0	1	1	RLINEW
0	1	1	0	0	LINE
0	1	1	0	1	RLINE
0	1	1	1	0	PLINE
0	1	1	1	1	RPLINE
1	0	0	0	0	MOVE
1	0	0	0	1	RMOVE
1	0	0	1	0	LCOFS
1	0	0	1	1	RLCOFS
1	0	1	0	0	CLRW
1	0	1	0	1	UCLIP
1	0	1	1	0	WPR
1	0	1	1	1	SCLIP
1	1	0	0	0	JUMP
1	1	0	0	1	GOSUB
1	1	0	1	0	VBKEM
1	1	0	1	1	RET
1	1	1	0	0	(reserved)
1	1	1	0	1	NOP1
1	1	1	1	0	NOP3
1	1	1	1	1	TRAP

11.3.2 Command Fetching

The GE carries out drawing operations while fetching the display list. The display list consist of a number of linked GE drawing commands.

The GE performs sequential fetches in low-to-high address order, starting at the address set in the Display List Area Start Address Register (DLSAR). The fetch address can be changed midway, using a JUMP or GOSUB command. GE fetching can be terminated by placing a TRAP command at the end of the display list.

The GE has a dedicated command buffer, and an equivalent area of the graphics memory is accessed at one time. When processing of the commands in this buffer is completed, another command fetch is performed.

If the commands include a JUMP, GOSUB, or other command that changes the flow, the GE starts fetching again from the new address indicated by that command.

Figure 11.21 shows an example of the display list.

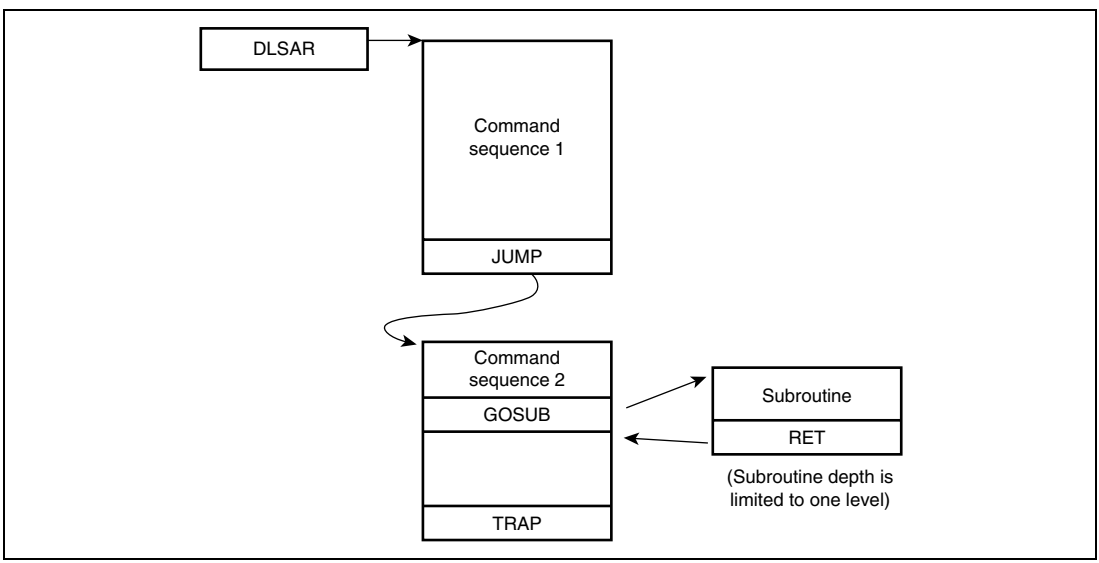


Figure 11.21 Example of Display List

The GE is provided with a drawing suspension facility to support execution control. This allows prioritized parallel execution of a number of drawing processes. An outline of the operation of this facility is given below.

Suspension Processing:

1. Set BRCL to 1 in the Status Register Clear Register (SRCR), clear the BRK bit to 0 in the Status Register (SR), and set the drawing suspension directive bit (RBRK) to 1 in the Rendering Control Register (RCR).
2. Next, monitor the BRK bit and TRA bit.
3. When BRK is observed to be set to 1, this means that the currently executing drawing command processing has ended and the drawing unit has halted (drawing has been suspended) at the point at which the next drawing command was fetched. Information required for software processing in anticipation of resumption processing should be read from the address-mapped registers and saved in memory. At this time, the RBRK bit is cleared to 0.

   When TRA is observed to be set to 1, this means that a TRAP command has been executed and GE drawing processing has ended. Therefore, ensure that no subsequent resumption processing is carried out. If drawing is to be performed after suspension processing, wait until the TRA flag is observed to be set to 1.

Resumption Processing:

1. The parameters saved immediately after suspension are restored. Some are written directly to the registers, and some are set by command. The former include the subroutine return address (which can also be set with the WPR command), and the latter, clip area, local offset, current pointer, and execution restart addresses. Of the latter, the execution restart address is restored by setting the command status register value at the time of the suspension as the jump destination of a JUMP command. For the other parameters in the latter group, settings should be made to provide for recovery by means of the appropriate command before execution of this JUMP command.
2. After performing a write for the purpose of subroutine return address restoration, and creating a command list to restore the other parameters, drawing can be resumed by setting the address of this command list in DLSAR and implementing a rendering start.

11.3.3 Q2SD/RU Basic Functions

(1) Rendering Coordinate Systems

Rendering Coordinates: This is the coordinate system used for drawing processing; it has a fixed size as shown in figure 11.22. The correspondence to the frame buffers is also fixed, but depends on the installed memory capacity and screen size. In an area other than one containing a frame buffer, although drawing operations are performed, nothing is written. The bit configuration of these coordinates is indicated by the graphic bit mode (GBM) in the rendering mode register

(RMR). When drawing is performed using the LCOFS command, coordinates after addition of offset values XO and YO set by the LCOFS command must be within the range shown in the following expressions.

When bold line attribute is specified:

$$-2045 \leq X + XO \leq 2044$$
$$-2045 \leq Y + YO \leq 2044$$

When bold line attribute is not specified:

$$-2048 \leq X + XO \leq 2047$$
$$-2048 \leq Y + YO \leq 2047$$

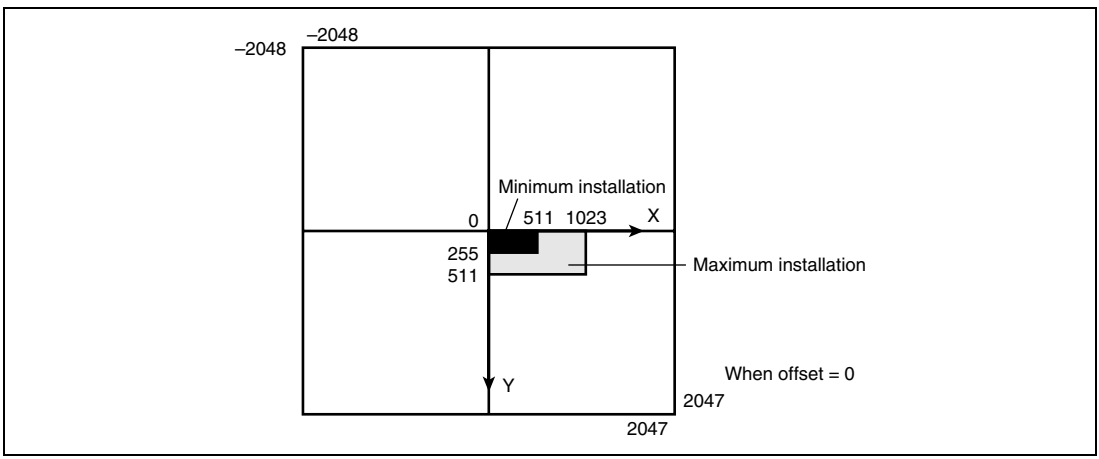


Figure 11.22 Rendering Coordinates

Multi-Valued Source Coordinates: The Q2SD/RU can use two kinds of multi-valued source coordinates according to the value of linear attribute LNi. When LNi = 0, the coordinate origin is specified by the multi-valued source area start address. Figure 11.23 shows the multi-valued source coordinates when LNi = 0. As shown in this figure, the maximum coordinate system size is represented by 1024×1024 positive coordinates, but the size depends on the installed memory capacity, screen size, and multi-valued source area start address. Depending on the multi-valued source start address, this coordinate system may entirely or partially overlap another coordinate system.

When LNi = 0, multi-valued source coordinates are configured based on the memory unit shown in section 3.3, Memory Map.

When LNi = 1, it is possible to use multi-valued source arranged in linear fashion in the UGM. The size of the multi-valued source in this case is determined by the TDX and TDY parameters of the POLYGON4A command. Figure 11.23 shows the multi-valued source coordinates when LNi

= 1. When this coordinate system is used, address conversion must be carried out as shown in section 11.2.4, Memory Map.

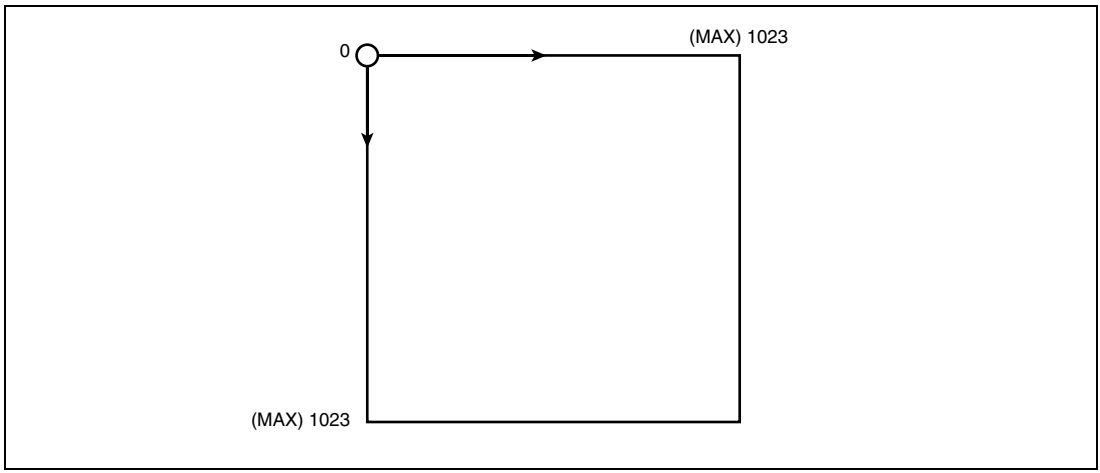


Figure 11.23 Multi-Valued Source Coordinates (LNi = 0)

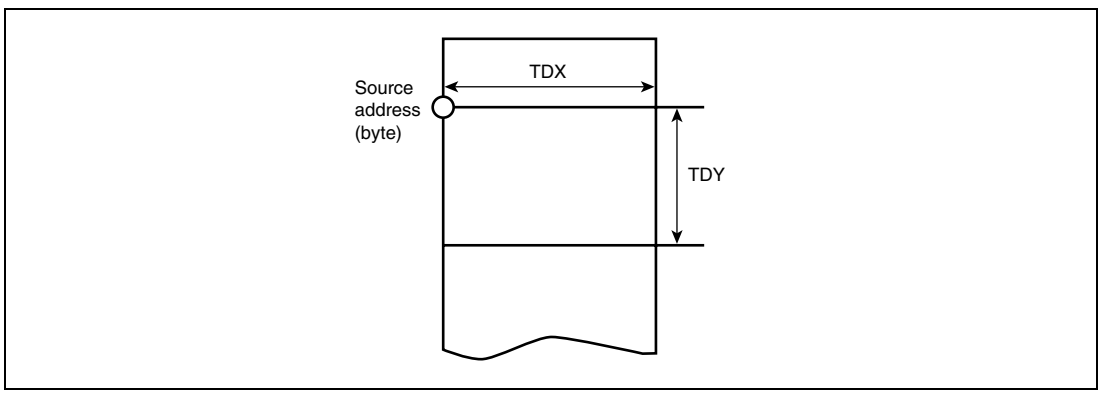


Figure 11.24 Multi-Valued Source Coordinates with LNi = 1 Specified (Linear Address)

Binary Source Coordinates: The binary (1-bit/pixel) source coordinate system is mapped directly onto 1-dimensional memory space. Any area and location can be used, and can be intermixed with the display list space. However, the start address of a source figure is always a byte address. The size of the figure is specified by POLYGON4B command parameters TDX and TDY.

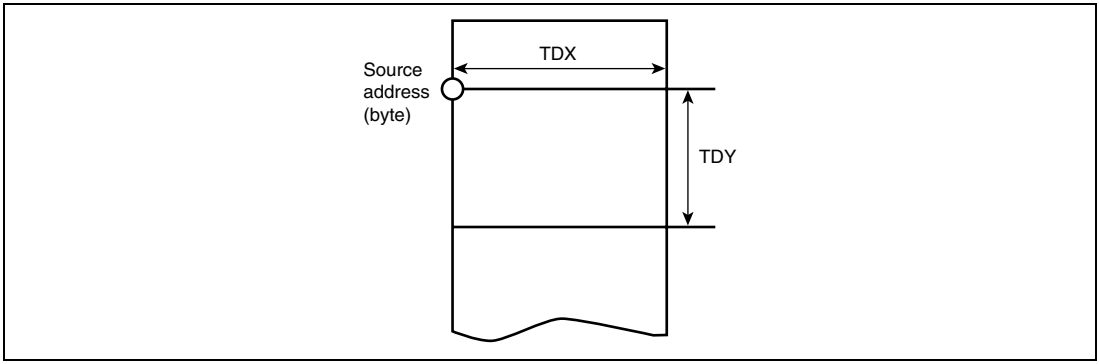


Figure 11.25 Binary Source Coordinates

Work Coordinate System: The work coordinate system corresponds on a one-to-one basis to the rendering coordinate system, as shown in figure 11.26. Therefore, clipping is also handled in the same way as for the rendering coordinate system.

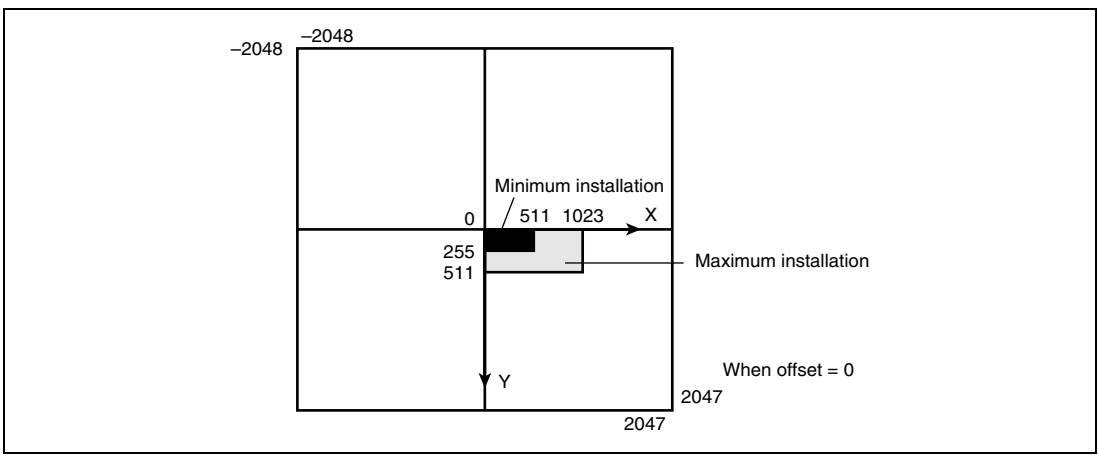


Figure 11.26 Work Coordinate System

Relationship between Work Coordinates and Addresses: Work coordinates are linear coordinates that start from the work area start address. Work coordinates comprise 2-dimensional coordinates reflected at each pixel (512 or 1024 pixels) specified by the MWX bit in the rendering mode register (RMR).

The memory capacity required for the work area is (the number of pixels specified by the MWX bit) × (SCLIP command YMAX + 1)/8 [bytes]. In general, one less than the number of display lines in the vertical direction should be set as YMAX in the SCLIP command.

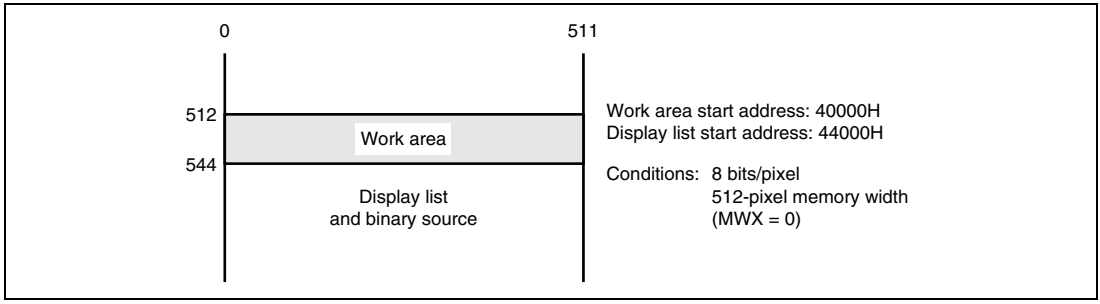


Figure 11.27 Example of Relationship between Work Coordinates and Physical Addresses

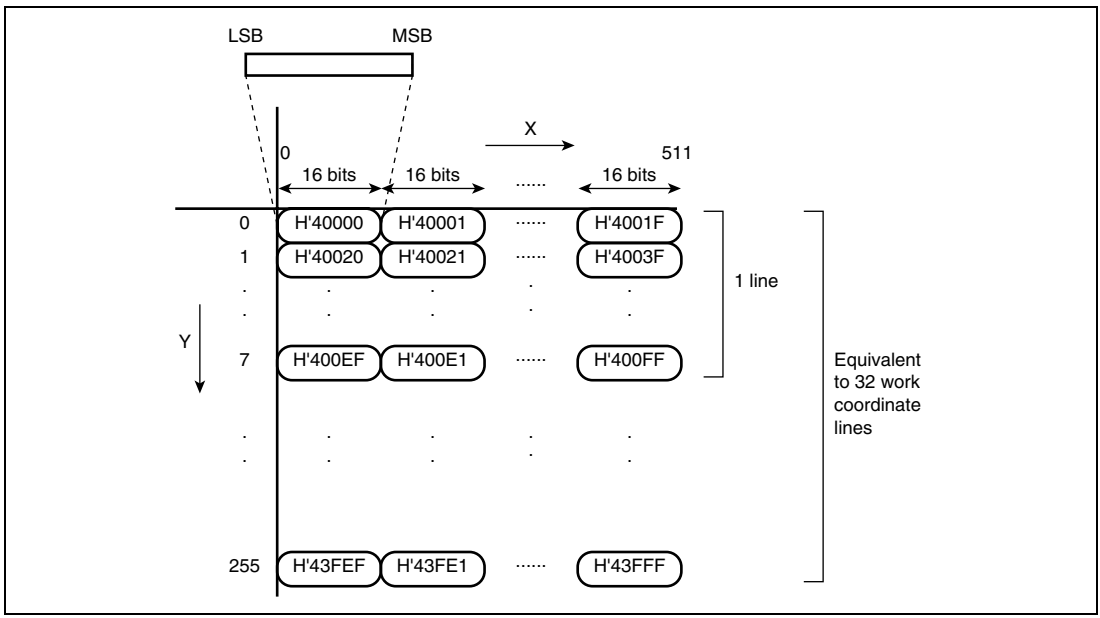


Figure 11.28 Relationship between Work Coordinates and Addresses

(2) Rendering Reference Data

Q2SD/RU drawing operations can be broadly divided into those that refer to the source data and those that do not. Drawing commands that refer to the source data are POLYGON4A, POLYGON4B, PLINE, and RPLINE. Drawing commands that do not refer to the source data are POLYGON4C, LINE, RLINE, FTRAP, RFTRAP, CLRW, LINEW, and RLINEW.

With drawing operations that refer to the source data, there are two reference data formats: multi-valued source data and binary source data.

Of the commands that do not reference source data, POLYGON4C, LINE, RLINE, LINEW, and RLINEW refer to the specified color data belonging to the command.

With POLYGON4 commands, it is possible to refer to a combination of multi-valued source data and binary source data, binary source data and binary work data, or specified color data and binary work data (see figure 11.29).

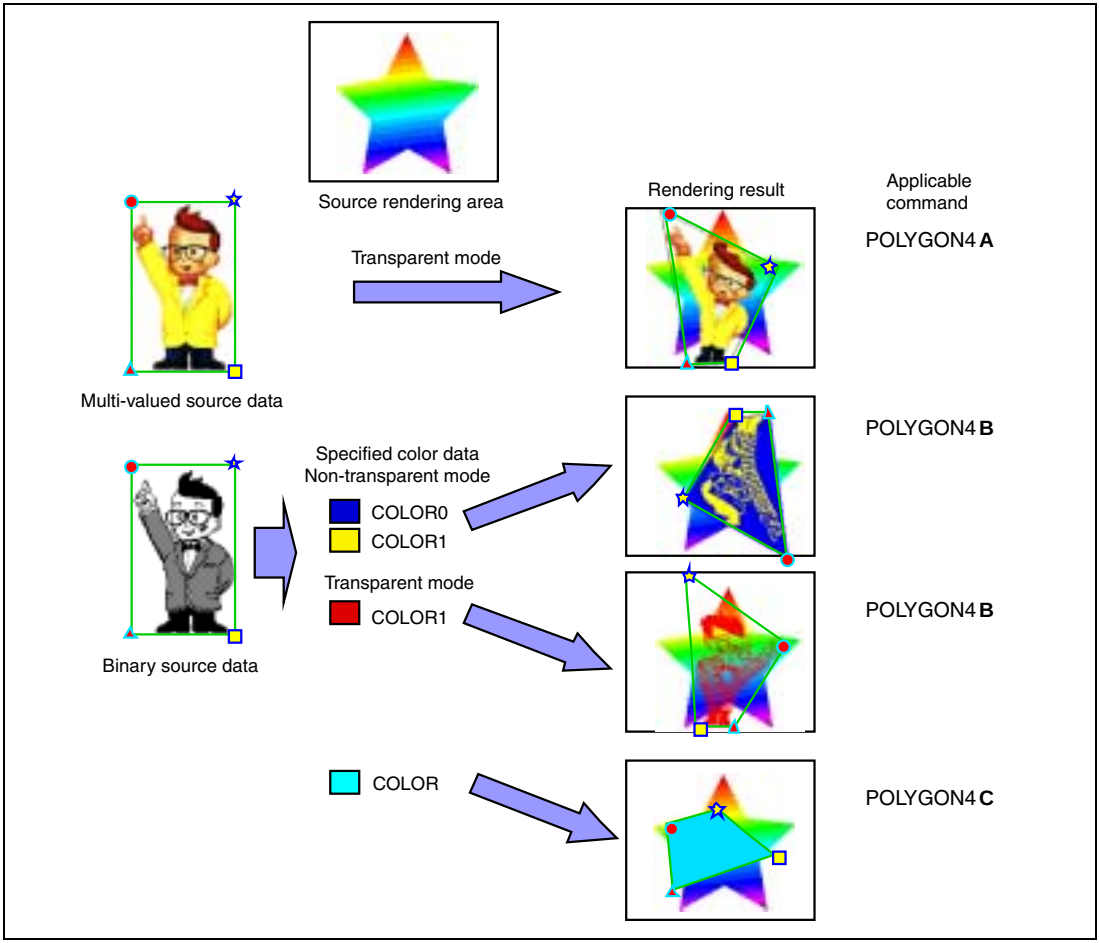


Figure 11.29 Example of POLYGON4 Transfer Data Combinations

Multi-Valued Source Data: Multi-valued source data is defined as multi-valued source coordinates (2-dimensional coordinates).

However, the horizontal width (TDX) is specified as a value of 8 pixels or more. The configuration of multi-valued source data is shown in figure 11.30.

A linear arrangement (LNi = 1) is also possible, in which case a multiple of 8 pixels should be set as the TDX value.

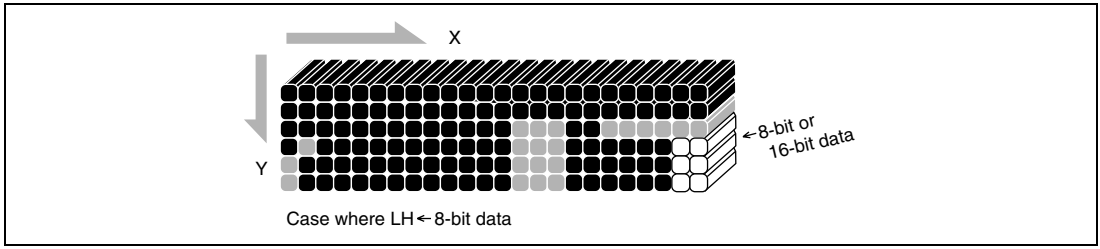


Figure 11.30 Multi-Valued Source Data Configuration

Binary Source Data: Binary source data is arranged in linear fashion in the binary source area in the UGM, and is managed as 2-dimensional coordinates (binary source coordinates) by TDX and TDY in the POLYGON4B command. The left-hand screen pixel must be located at the LSB of the binary source data.

However, the horizontal width (TDX) is specified as a multiple of 8 pixels. An example of binary source data is shown in figure 11.31.

A binary source is used for the definition of character data and line-type data. When drawing, 0s are converted to COLOR0 data, and 1s to COLOR1 data (in transparent mode, only 1s are converted to COLOR1 data for drawing).

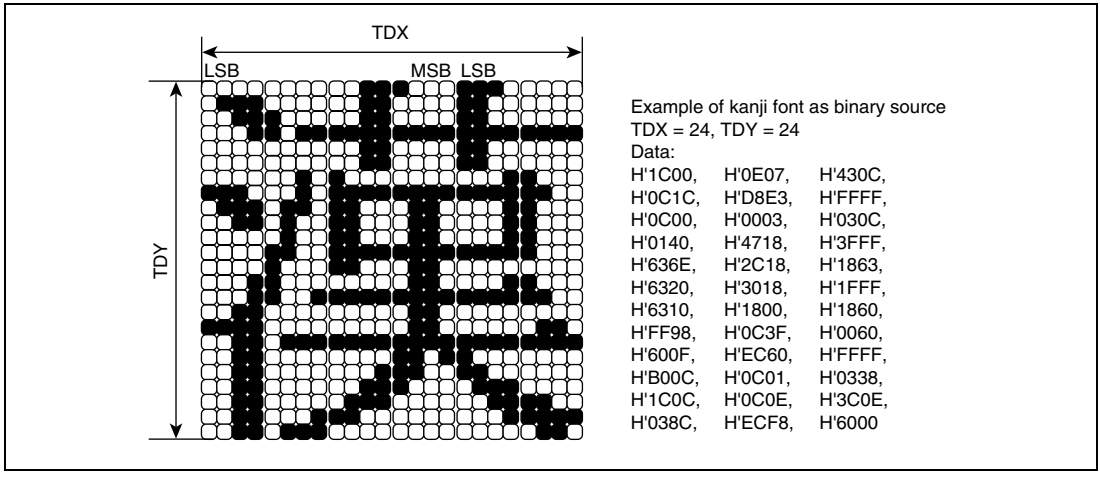


Figure 11.31 Example of Kanji Font as Binary Source (TDX = 24, TDY = 24)

Binary Work Data: Binary work data is defined as work coordinates (2-dimensional coordinates). Work data is used to implement polygon painting. Polygon outline data is created with the FTRAP command, etc., and the created figure data is used to delineate the rendering figure. If, for example, the POLYGON4C command is used jointly for work, the work area polygon can be drawn in the rendering area with the specified color value. The configuration of binary work data is shown in figure 11.32.

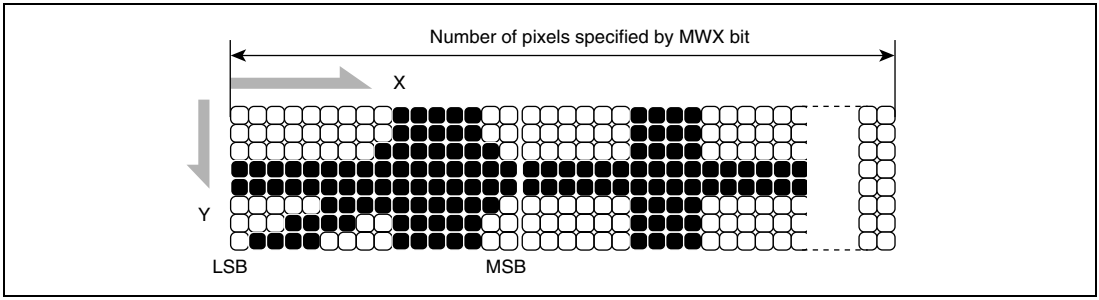


Figure 11.32 Binary Work Data Configuration

Specified Color Data: Specified color data is defined directly by drawing parameter color specifications (COLOR, COLOR0, COLOR1, LINE COLOR0, and LINE COLOR1). When the Q2SD/RU is used for 8-bit/pixel operation, the same color palette number is defined in the upper 8 bits and lower 8 bits in the drawing parameter color specification. When the Q2SD/RU is used for 16-bit/pixel operation, the R, G, and B values are defined directly by the drawing parameter color specification.

However, with LINEW and RLINEW, the value to be drawn at work coordinates is defined by the rendering attribute EOS bit.

Q2SD/RU Internal Buffers: The Q2SD/RU has three internal buffers—a command buffer, source buffer, and work buffer—as shown in figure 11.33.

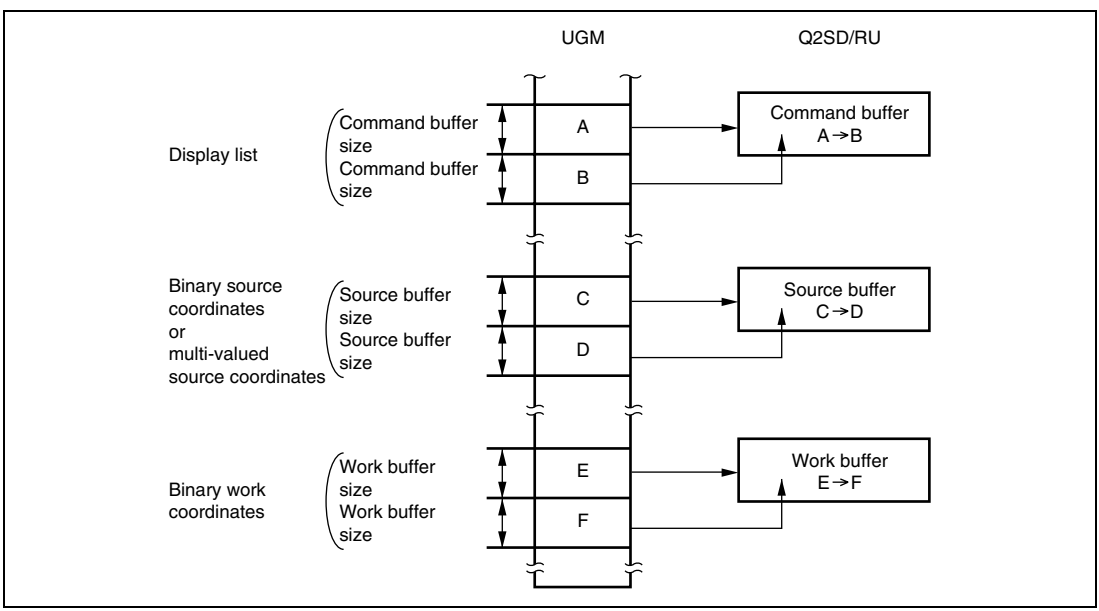


Figure 11.33 Updating of Q2SD/RU Internal Buffers

These buffers are used by the Q2SD/RU to temporarily store data held in the UGM. The Q2SD/RU uses the data stored in these buffers when executing drawing. The functions of these buffers are as follows:

- Command buffer (32 bytes × 2)

  Used by the Q2SD/RU to store a display list held in the UGM. The buffer size is 64 bytes.
- Source buffer (32 bytes × 2)

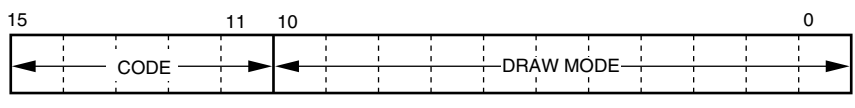
  Used by the Q2SD/RU to store a binary source or multi-valued source held in the UGM. The buffer size is 64 bytes.
- Work buffer (16 bytes)

  Used by the Q2SD/RU when performing drawing at binary work coordinates in the UGM. The buffer size is 16 bytes.

When buffer contents are not updated, (when the same address is referenced by data of or below the capacity of the buffer, or a reference ends at a location at or below the capacity of the buffer from the previous reference start location), the previous buffer contents will be used even though the data in the UGM is rewritten. To intentionally update buffer contents, the address of a location exceeding the buffer capacity should be referenced.

(3) Rendering Attributes

With the Q2SD/RU, 12 rendering attributes can be specified. The rendering attributes are embedded in the commands, and can be specified on an individual command basis. Figure 11.34 shows the bit arrangement for rendering attributes.



POLYGON4A		TRNS	STYL	CLIP	REL	NET	EOS	FST	LNi		WORK
POLYGON4B		TRNS	STYL	CLIP	REL	NET	EOS				WORK
POLYGON4C				CLIP		NET	EOS	FST			WORK
FTRAP, RFTRAP				CLIP			EOS	EDG			
LINEW, RLINEW				CLIP			EOS				
LINE, RLINE				CLIP		NET	EOS	FWUL	W2UL	FWDR	W2DR
PLINE, RPLINE		TRNS	1	CLIP		NET	EOS	EDG2		EDG1	1
JUMP, GOSUB					REL						
CLRW				CLIP							
(Commands other than the above)											

Note: Shaded areas should be cleared to 0.
FWUL, W2UL, FWDR, W2DR: Bold line drawing bits
REL: Relative address specification
EDG: Edge drawing bit

Figure 11.34 Rendering Attribute Bit Arrangement

Transparency Specification (TRNS): When color expansion of binary source data is performed, transparency or non-transparency can be selected on an individual drawing command basis with the TRNS bit. When transparency is selected, a 0 in the binary source data is transparent and a 1 has the value of the COLOR1 parameter. When non-transparency is selected, a binary data 0 has the value of the COLOR0 parameter, and a 1 has the value of the COLOR1 parameter. With multi-valued source data, all-0 data becomes a transparent color, and those pixels are not drawn. The transparency specification can be used with the POLYGON4A, POLYGON4B, PLINE, and RPLINE commands; in other commands, the TRNS bit should be cleared to 0.

Source Style Specification (STYL): When drawing a rectangle, the STYL bit can be used to select, on an individual drawing command basis, whether the source data is to be enlarged or reduced, or referenced repeatedly. If no style specification is made, the source data is enlarged or reduced in proportion to the size of the rendering area. When a style specification is made, the source data is referenced repeatedly in proportion to the size of the rendering area. This attribute is therefore used when drawing repeated patterns such as hatch patterns. The source style specification can be used with the POLYGON4A, POLYGON4B, PLINE, and RPLINE commands; in other commands, the STYL bit should be cleared to 0. When a source style specification is used, do not make a source half specification.

An example of a source style specification is shown in figure 11.35.

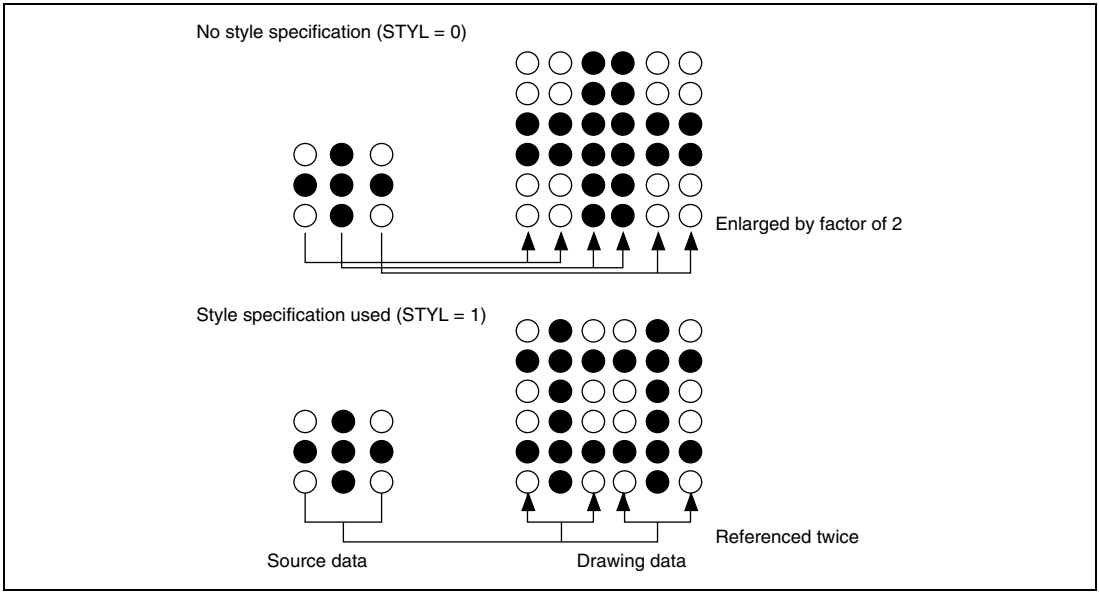


Figure 11.35 Example of Source Style Specification

Clipping Specification (CLIP): The Q2SD/RU can perform clipping area management. There are two kinds of clipping area: a system clipping area designated by the SCLIP command, and a user clipping area designated by the UCLIP command.

The system clipping area has a fixed drawing range. The system clipping area is always valid, regardless of attribute specifications.

A user clipping area can be designated as desired within the system clipping area. Whether or not clipping is performed in that area can be selected on an individual command basis with the rendering attribute CLIP bit. The boundary is drawn.

Clipping is set with screen coordinates.

An example of a clipping specification is shown in figure 11.36.

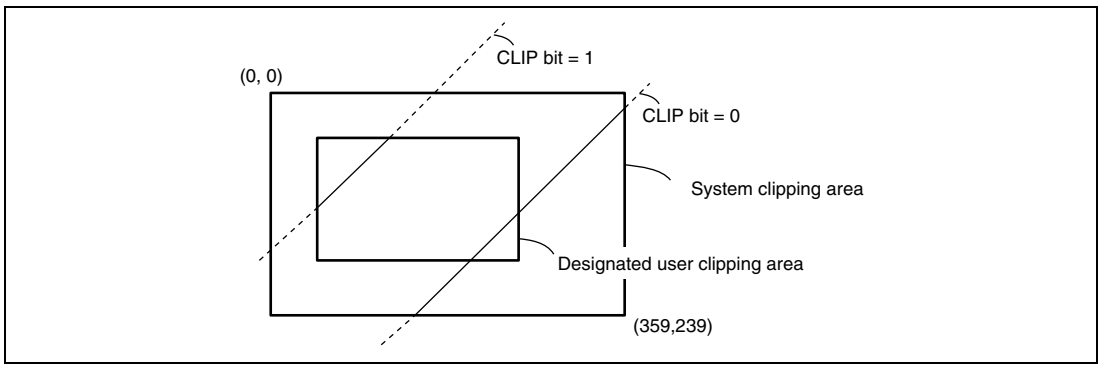


Figure 11.36 Example of Clipping Specification
[Specified system clipping area is (0, 0) to (359, 239)]

Net Drawing Specification (NET): The NET bit can be used to select, on an individual drawing command basis, whether or not net drawing is to be performed. Net drawing is a function for drawing only pixels at coordinates for which the condition "rendering coordinates X + Y = EOS (0: even number, 1: odd number) " is true. For example, if EOS = 0, drawing will only be performed at coordinates Y = 0, X = 0, 2, 4, 6, 8, ..., Y = 1, X = 1, 3, 5, 7, 9, ...

This function enables the drawn figure and ground to be mutually semi-superimposed.

The net drawing specification can be used with the POLYGON4 commands, and the LINE, RLINE, PLINE, and RPLINE commands; in other commands, the NET bit should be cleared to 0.

Even/Odd Select Specification (EOS): Even pixels are selected when EOS = 0, and odd pixels when EOS = 1.

The even/odd select specification is used together with the net drawing specification.

With the LINEW and RLINEW commands, drawing is performed at the work coordinates with 0 when EOS = 0, and with 1 when EOS = 1.

Work Specification (WORK): When drawing is performed at rendering coordinates with POLYGON4 commands, the WORK bit can be used to select, on an individual drawing command basis, whether or not binary work data is to be referenced.

When binary work data referencing is selected, drawing is performed if the work data for the pixel corresponding to the rendering coordinates is 1, but not if the work data is 0. The same shape as that drawn at work coordinates can thus be drawn at rendering coordinates. Drawing at work coordinates can be performed either by means of the FTRAP command or else by CPU. Ensure that UGM drawing access by command and UGM drawing access by the SuperH are not performed simultaneously. The work specification can be used with the POLYGON4A, POLYGON4B, and POLYGON4C commands; in other commands, the WORK bit should be cleared to 0.

With the PLINE and RPLINE commands, this attribute is specified but work references are not performed.

Bold Line Drawing Specification: Taking individual line segments of a polygonal line specified by parameters as reference lines, this specification makes the reference lines bold lines in the upper-left direction and lower-right direction, independently. Whether or not this attribute is enabled is specified by the FWUL bit and FWDR bit, while the width of a bold lines can be selected from line widths 1 to 5 by a combination of bits W2UL and W2DR. The FWUL bit enables bold-line implementation in the upper-left direction, while the FWDR bit enables bold-line implementation in the lower-right direction. The W2UL bit is valid when FWUL = 1, and the W2DR bit when FWDR = 1.

This function is valid for each segment of a polygonal line. Using the segment line main scanning axes, lines with the same slope in the up (left) and down (right) directions, and of the same length, are drawn repeatedly. Therefore, the shape of the segment linkage parts is not considered. This function can be used with the LINE and RLINE commands; in other commands, the FWUL, W2UL, FWDR, and W2DR bits should all be cleared to 0.

When performing bold line drawing, set the vertex coordinates so that the entire bold line area does not extend beyond the drawing area (both x and y in the range –2045 to 2044).

Table 11.3 Bold Line Drawing Settings

FWUL	W2UL	FWDR	W2DR	Line Width (Direction, Magnification)
0	0	0	0	1 (no magnification)
			1	1 (no magnification)
		1	0	2 (lower right 1)
			1	3 (lower right 2)
	1	0	0	1 (no magnification)
			1	1 (no magnification)
		1	0	2 (lower right 1)
			1	3 (lower right 2)
1	0	0	0	2 (upper left 1)
			1	2 (upper left 1)
		1	0	3 (upper left 1, lower right 1)
			1	4 (upper left 1, lower right 2)
	1	0	0	3 (upper left 2)
			1	3 (upper left 2)
		1	0	4 (upper left 2, lower right 1)
			1	5 (upper left 2, lower right 2)

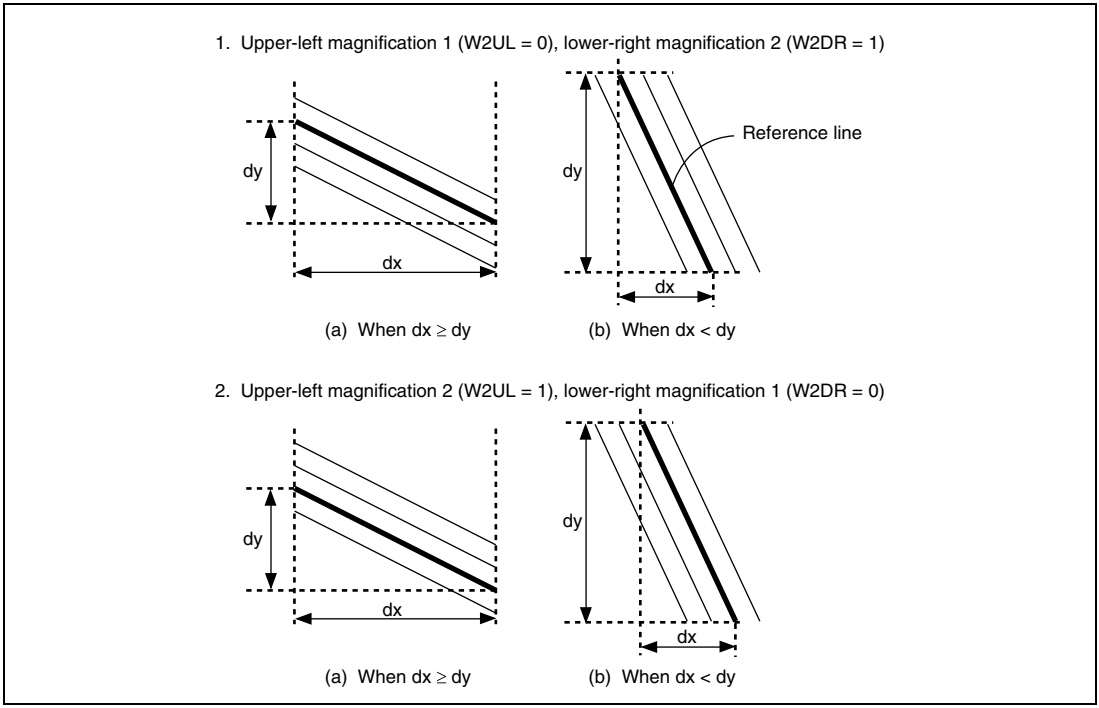


Figure 11.37 Examples of Bold Line Drawing (Line Width 4 Drawing) (FWUL = 1, FWDR = 1)

Source Address Linear Specification (LNi): Use of a 2-dimensional virtual address or a linear address as the source address can be selected, on an individual drawing command basis, by means of the LNi bit. To use a linear address, set this bit to 1.

This function can be used with the POLYGON4A command; in other commands, the LNi bit should be cleared to 0. For details of command operation, see the description of POLYGON4A in section 11.3.4.

4-Pixel-Unit Processing (FST): Whether or not 4-pixel unit processing is performed can be specified for individual drawing commands by means of the FST bit. To perform 4-pixel unit processing, set the FST bit to 1. In this case, no other rendering attributes except CLIP can be used. This function can be used with the POLYGON4A and POLYGON4C commands; in other commands, the FST bit should be cleared to 0.

When using this attribute, set the command parameters as indicated in the individual command descriptions.

Source Coordinate Relative Address Specification (REL): Setting the REL bit to 1 in POLYGON4A, POLYGON4B, JUMP, and GOSUB commands enables source referencing and branching to be performed at an address relative to (before or after) the command code. The source address must be a linear address. Also, for reasons relating to referencing of a multi-valued source arranged in linear fashion, the LNi bit must be set to 1 when using POLYGON4A; operation cannot be guaranteed if the LNi bit is 0.

The command code address is the relative address origin.

Edge Drawing (EDG): With the FTRAP and RFTRAP commands, setting the EDG bit to 1 enables edge lines to be drawn after completion of trapezoid painting. Whether edge line drawing is performed with 0 or with 1 is specified by the EOS bit.

Line Drawing Edge Specification (EDG1, EDG2): Whether or not edge drawing is performed for a polygonal line with line type can be specified for individual drawing commands by means of the EDG1 bit.

This function is valid for each segment of a polygonal line. Using the segment line main scanning axes, solid lines with the same slope and of the same length, are drawn either vertically or horizontally. Therefore, the shape of the polygonal line linkage parts is not considered. The solid edge lines have the value of COLOR1.

This function can be used with the PLINE and RPLINE commands; in other commands, the EDG1 bit should be cleared to 0. A source size of 8 or 16 can be used. Set 8 or 16 for source size parameter TDX.

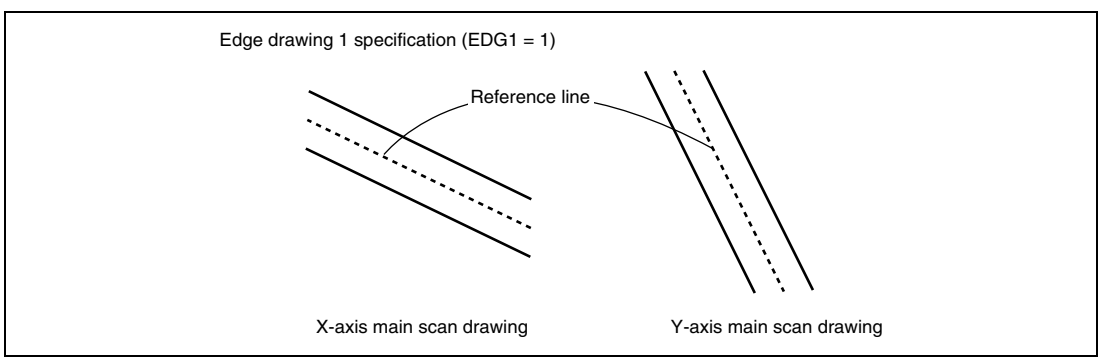


Figure 11.38 Edge Drawing 1

Whether or not edge drawing is performed for a polygonal line with line type can be specified for individual drawing commands by means of the EDG2 bit.

This function is valid for each segment of a polygonal line. Here, each segment of the polygonal line specified by the parameter is considered as a reference line. This function is implemented for each segment of a polygonal line, using the following procedure. First, the reference line is drawn

as a line with line type. Next, using the segment line main scanning axes, solid lines with the same slope and of the same length, are drawn either vertically or horizontally. Finally, the reference line is drawn as a solid line. Therefore, the shape of the polygonal line linkage parts is not considered. The solid line drawn last has the value of COLOR1.

This function can be used with the PLINE and RPLINE commands; in other commands, the EDG2 bit should be cleared to 0.

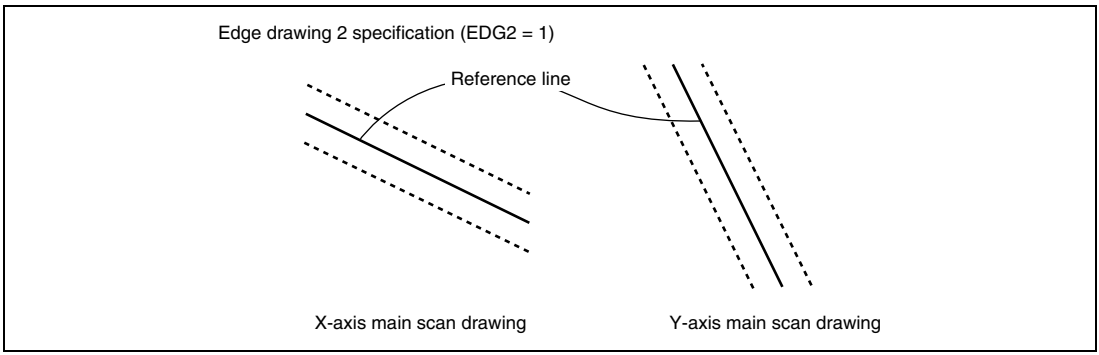


Figure 11.39 Edge Drawing 2

Do not set both EDG1 and EDG2 to 1 at the same time.

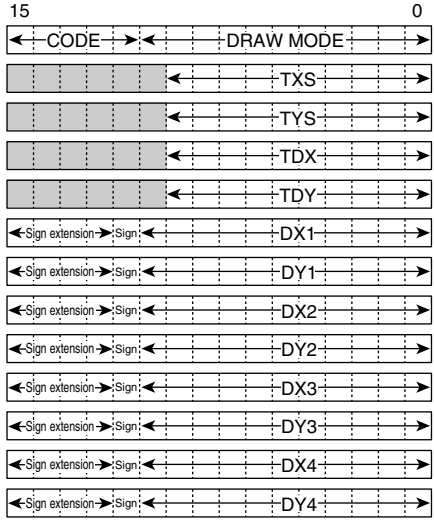
11.3.4 Q2SD/RU Drawing Commands

(1) POLYGON4A

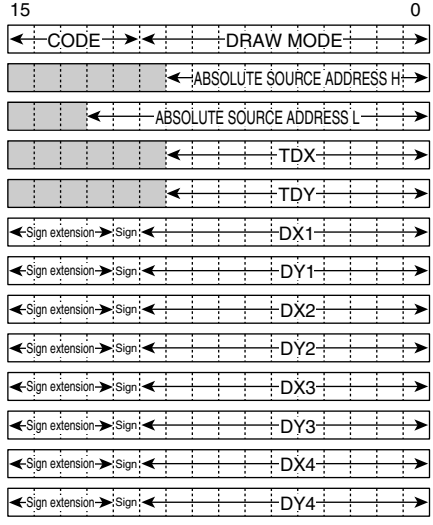
Function

Performs any four-vertex drawing at rendering coordinates while referencing a multi-valued (8- or 16-bit/pixel) source.

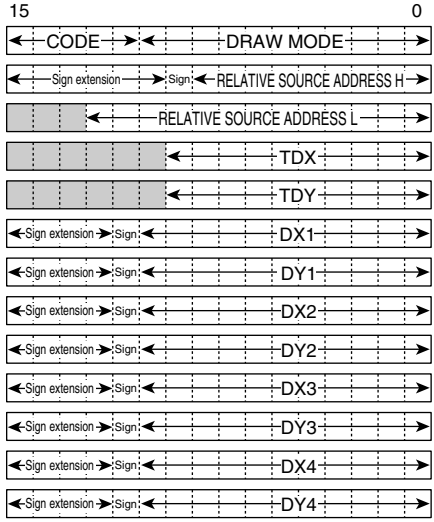
Command Format LNi = 0



LNi = 1, REL = 0



LNi = 1, REL = 1



1. Code
   B'00000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
O		A		O	

DRAW MODE

Reserved	TRNS	STYL	CLIP	REL	NET	EOS	FST	LNi	Reserved	WORK
Fixed at 0	*	*	O	Z	*	*	O	*	Fixed at 0	*

O: Valid
A: Valid when WORK = 1
*: Valid when FST = 0 (clear to 0 when FST = 1)
Z: Valid when LNi = 1 (clear to 0 when LNi = 0)

3. Command Parameters

   TXS, TYS: Source starting point

   ABSOLUTE/RELATIVE SOURCE ADDRESS H: Source starts upper address (byte address)

   ABSOLUTE/RELATIVE SOURCE ADDRESS L: Source starts lower address (byte address)

   TDX, TDY: Source size

   DXn, DYn (n = 1 to 4): Absolute values, rendering coordinates, negative numbers expressed as two's complement

Description

Transfers multi-valued (8- or 16-bit/pixel) source data to any quadrilateral rendering coordinates. The source is always scanned horizontally, but diagonal scanning may be used in the drawing, depending on the shape. In diagonally-scanned drawing, double-writing occurs to fill in gaps.

When LNi = 1, set a multiple of 8 pixels as the TDX value.

When LNi = 0, set 8 pixels or more as the TDX value.

If the TDX setting is less than 8 pixels, multi-valued source references will not be performed normally.

1. When repeated source referencing is selected as a rendering attribute (STYL = 1), the source is not enlarged or reduced, but is referenced repeatedly.
2. When work referencing is selected as a rendering attribute (WORK = 1), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.
3. When LNi = 0, make TXS and TYS settings in pixel units.
4. When LNi = 1, the linear address space in the UGM can be used for multi-valued source coordinates.

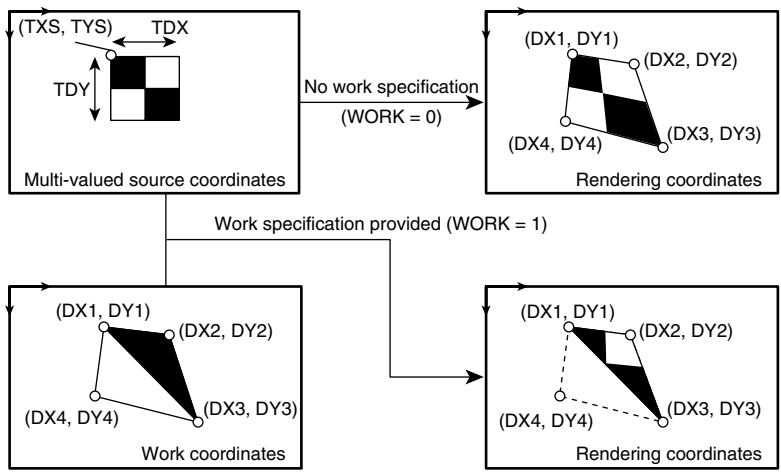
   When LNi = 1, set the upper bits of the source address in SOURCE ADDRESS H, and the lower bits in SOURCE ADDRESS L. When REL = 0, the source address can be specified as an absolute address. When REL = 1, the source address can be specified as a relative address with respect to the UGM address at which the POLYGON4A command code is located. Absolute addresses and relative addresses must be even numbers. If a relative address is negative, its two's complement should be used.

Note on FST Mode

When the register attribute FST bit is set to 1, processing is carried out in 4-pixel units. However, operation will be executed normally only if all the following conditions are satisfied; in other cases, operation cannot be guaranteed. Evaluation of these conditions is not performed internally.

- Make settings so that the source and destination are rectangles of the same size, with DX1 = DX4 = 4j – 4, DX2 = DX3 = 4k – 1, DY1 = DY2, DY3 = DY4, DX2 – DX1 = 32n – 1 (where j, k, and n are natural numbers).
- When FST = 1, no other rendering attributes except CLIP can be used.
- When this command is used with FST = 1, first use the MOVE, RMOVE, LCOFS, or RLCOFS command to change the clipping range and local offset values to the values given in the descriptions of the individual commands.
- Set a multiple of 4 for TXS and TYS.
- Operation is valid in 8-bit/pixel and 16-bit/pixel modes.
- The local offset values set by the LCOFS and RLCOFS commands must be non-negative.

Example

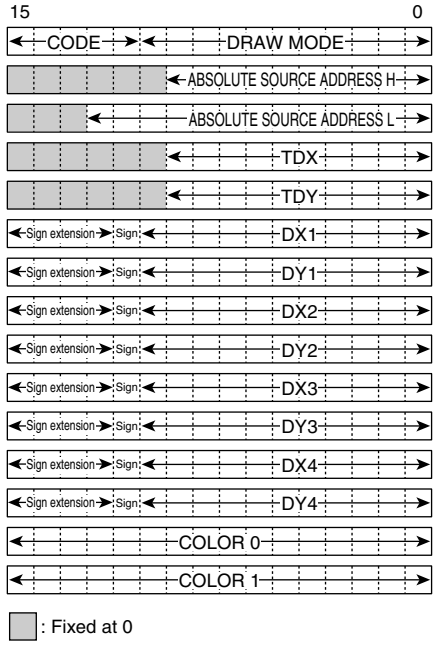


(2) POLYGON4B

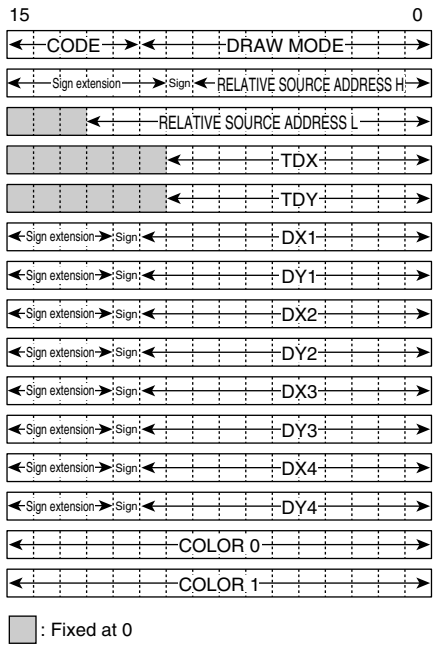
Function

Performs any four-vertex drawing at rendering coordinates while referencing a binary (1-bit/pixel) source.

Command Format REL = 0



REL = 1



1. Code
   B'00001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
	O	A		O	

DRAW MODE										
Reserved	TRNS	STYL	CLIP	REL	NET	EOS	Reserved	Reserved	Reserved	WORK
Fixed at 0	O	O	O	O	O	O	Fixed at 0	Fixed at 0	Fixed at 0	O

O: Valid
A: Valid when WORK = 1

3. Command Parameters

   ABSOLUTE/RELATIVE SOURCE ADDRESS H: 1-bit/pixel source start upper address (byte address)

   ABSOLUTE/RELATIVE SOURCE ADDRESS L: 1-bit/pixel source start lower address (byte address)

   TDX, TDY: Source size

   DXn, DYn (n = 1 to 4): Absolute values, rendering coordinates, negative numbers expressed as two's complement

   COLOR0, COLOR1: 8- or 16-bit/pixel color specifications

Description

Draws binary (1-bit/pixel) source data in any quadrilateral rendering area, using the colors specified by parameters COLOR0 and COLOR1.The source is always scanned horizontally, but diagonal scanning may be used in the drawing, depending on the shape. In diagonally-scanned drawing, double-writing occurs to fill in gaps.

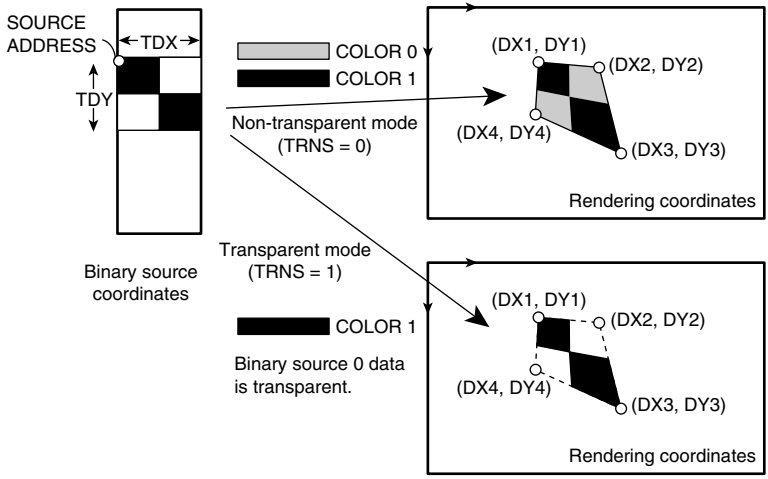
A multiple of 8 pixels must be set as the TDX value.

Binary source data is located in an area in the UGM. When REL = 0, the source address can be specified as an absolute address. When REL = 1, the source address can be specified as a relative address with respect to the UGM address at which the POLYGON4B command code is located.

Absolute addresses and relative addresses must be even numbers. If a relative address is negative, its two's complement should be used.

1. When repeated source referencing is selected as a rendering attribute (STYL = 1), the source is not enlarged or reduced, but is referenced repeatedly.
2. When work referencing is selected as a rendering attribute (WORK = 1), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.

Example

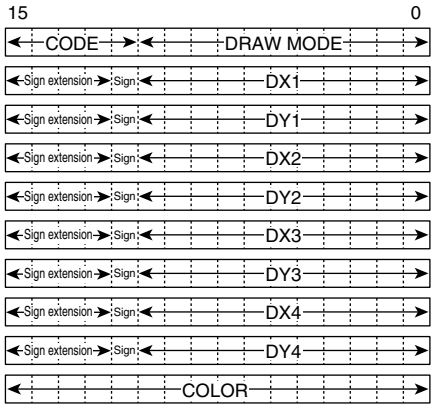


(3) POLYGON4C

Function

Performs any four-vertex drawing at rendering coordinates with a monochrome specification.

Command Format



1. Code
   B'00010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
		A	O	O	

DRAW MODE

Reserved			CLIP	Reserved	NET	EOS	FST	Reserved		WORK
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	*	*	O	Fixed at 0	Fixed at 0	*

O: Valid

A: Valid when WORK = 1

*: Valid when FST = 0(clear to 0 when FST = 1)

3. Command Parameters

   DXn, DYn (n = 1 to 4): Absolute values, rendering coordinates, negative numbers expressed as two's complement

   COLOR: 8- or 16-bit/pixel color specification

Description

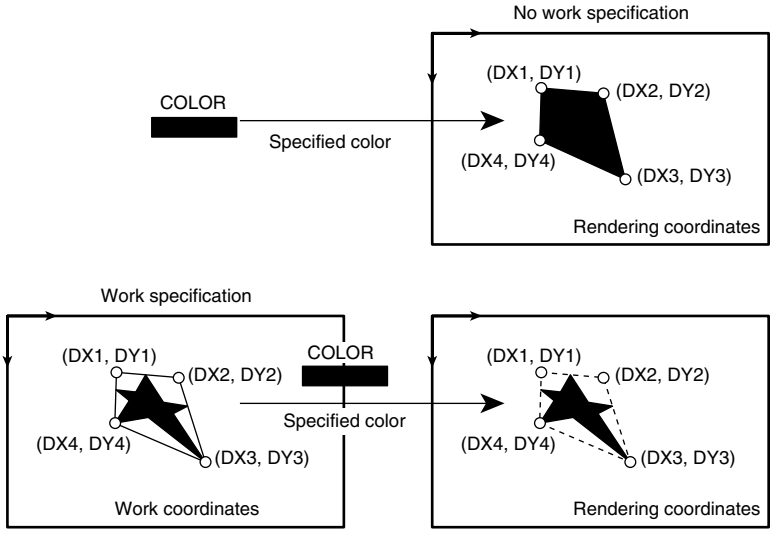
Draws any quadrilateral in the rendering area in the single color specified by the COLOR parameter.

When work referencing is selected as a rendering attribute (WORK = 1), only places where the work coordinate pixel is 1 are drawn at rendering coordinates while referencing work coordinates for the same coordinates as the rendering coordinates.

When the register attribute FST bit is set to 1, processing is carried out in 4-pixel units. However, operation will be executed normally only if all the following conditions are satisfied; in other cases, operation cannot be guaranteed. Evaluation of these conditions is not performed internally.

- Make settings so that the source and destination are rectangles of the same size, with DX1 = DX4 = 4j – 4, DX2 = DX3 = 4k – 1, DY1 = DY2, DY3 = DY4, DX2 – DX1 = 32n – 1 (where j, k, and n are natural numbers).
- When FST = 1, no other rendering attributes except CLIP can be used.
- When this command is used with FST = 1, first use the MOVE, RMOVE, LCOFS, or RLCOFS command to change the clipping range and local offset values to the values given in the descriptions of the individual commands.
- Operation is valid in 8-bit/pixel and 16-bit/pixel modes. In 8-bit/pixel mode, set the same 8-bit data for the upper and lower color attribute values.
- The local offset values set by the LCOFS and RLCOFS commands must be non-negative.

Example

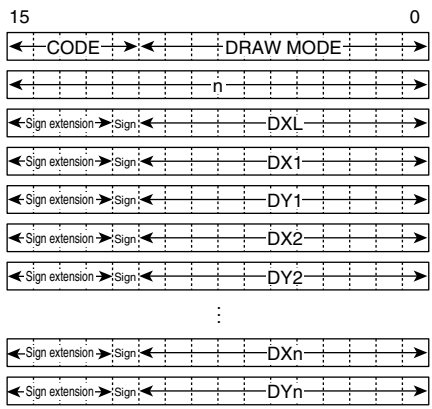


(4) FTRAP

Function

Draws a polygon at work coordinates.

Command Format



1. Code

   B'01000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
					O

DRAW MODE

Reserved			CLIP	Reserved		EOS	EDG	Reserved		
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	B	O	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid

B: Valid when EDG = 1 (clear to 0 when EDG = 0)

3. Command Parameters

   n (n = 2 to 65,535): Number of vertices

   DXL: Left-hand side coordinate

   DXn (n = 2 to 65,535): Absolute value, work coordinate, negative number expressed as two's complement

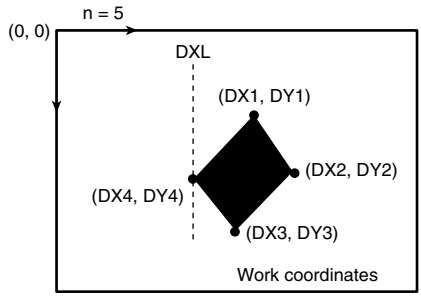
   DYn (n = 2 to 65,535): Absolute value, work coordinate, negative number expressed as two's complement

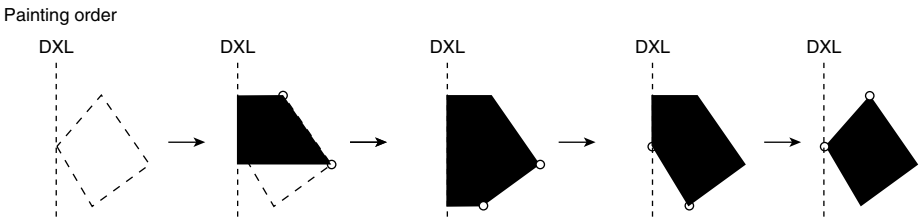
Description

Draws a polygon with n-1 vertices at work coordinates. Paints n-1 trapezoids at work coordinates using binary EOR, with X = DXL as the left-hand side, and line segments (DX1, DY1) – (DX2, DY2), (DX2, DY2) – (DX3, DY3), ..., (DXn-1, DYn-1) – (DXn, DYn) as the right-hand sides, and with top and bottom bases parallel to the X-axis. Bottom base drawing is not performed. Set the minimum value of DX1 to DXn as DXL.

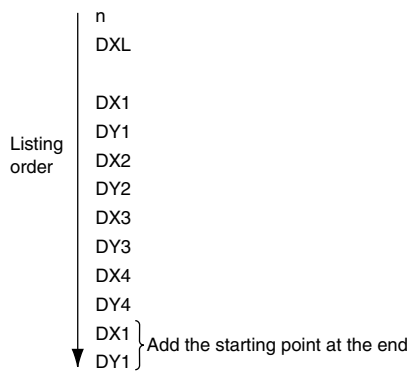
If the draw mode EDG bit is set to 1, an edge line is drawn after the paint operation. The line drawing data is selected with the EOS bit. When setting the EDG bit to 1, set (DXN, DYN) = (DX1, DY1) to give a closed figure.

Example





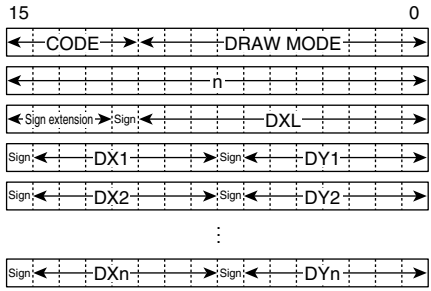
Order of Listing FTRAP Parameters



(5) RFTRAP

Draws a polygon at work coordinates.

Command Format



1. Code
   B'01001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
					O

DRAW MODE

Reserved			CLIP	Reserved		EOS	EDG	Reserved		
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	B	O	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid

B: Valid when EDG = 1 (clear to 0 when EDG = 0)

3. Command Parameters

   n (n = 1 to 65,535): Number of vertices

   DXL: Left-hand side coordinates, work coordinate, negative number expressed as two's complement

   DXn, DYn (n = 1 to 65,535): Relative values, work coordinates, negative numbers expressed as two's complement

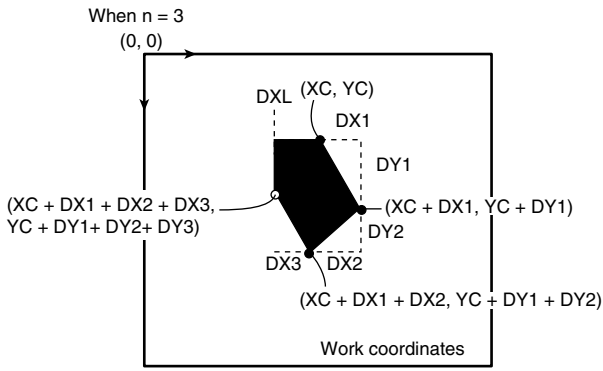
Description

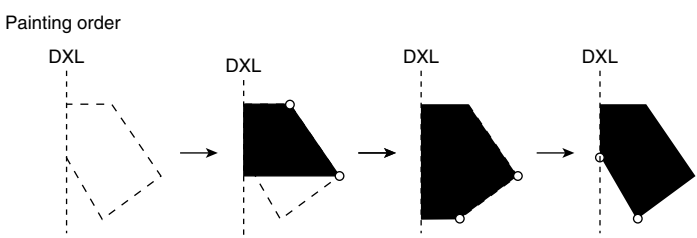
Paints n trapezoids at work coordinates using binary EOR, with X = DXL as the left-hand side, and line segments specified by the relative shift (DX, DY) from the current pointer values (XC, YC) ((XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn)) as the right-hand sides, and with top and bottom bases parallel to the X-axis. Bottom base drawing is not performed.

The final coordinate point is stored as the current pointer values (XC, YC).

If the draw mode EDG bit is set to 1, an edge line is drawn after the paint operation. The line drawing data is selected with the EOS bit. When setting the EDG bit to 1, set (DX1 + DX2 + ... + DXn = 0, DY1 + DY2 + ... + DYn = 0) to give a closed figure.

Example



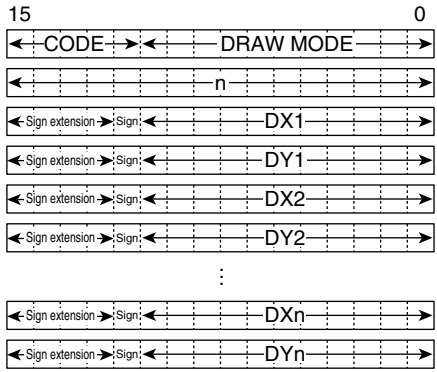


(6) LINEW

Function

Draws a 1-bit-wide solid line at work coordinates.

Command Format



1. Code
   B'01010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
			V		O

DRAW MODE

Reserved			CLIP	Reserved		EOS	Reserved			
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid
V: Valid (specified color is determined by EOS)

3. Command Parameters
   n (n = 2 to 65,535): Number of vertices
   DXn (n = 2 to 65,535): Absolute value, work coordinate, negative number expressed as two's complement

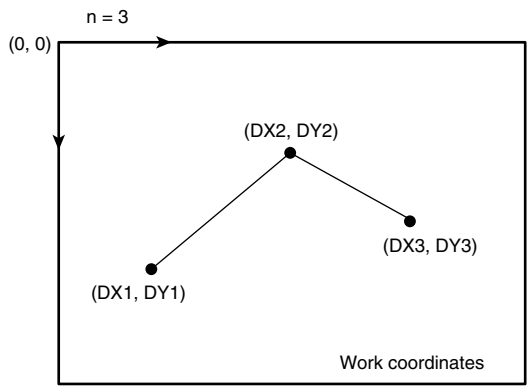
DYn (n = 2 to 65,535): Absolute value, work coordinate, negative number expressed as two's complement

Description

Performs binary drawing at work coordinates of a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn). 0 drawing or 1 drawing is selected with the drawing mode EOS bit. Drawing is performed at work coordinates with 0 when EOS = 0, and at work coordinates with 1 when EOS = 1. (Used for border drawing at work coordinates for a polygonal painted figure.)

Note: 8-point drawing is used.

Example

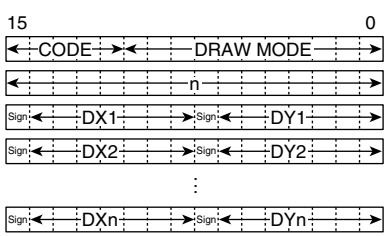


(7) RLINEW

Function

Draws a 1-bit-wide solid line at work coordinates.

Command Format



1. Code
   B'01011

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
			V		O

DRAW MODE

Reserved			CLIP	Reserved		EOS	Reserved			
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid
V: Valid (specified color is determined by EOS)

3. Command Parameters
   n (n = 1 to 65,535): Number of vertices
   DXn, DYn (n = 1 to 65,535): Relative values, work coordinates, negative numbers expressed as two's complement

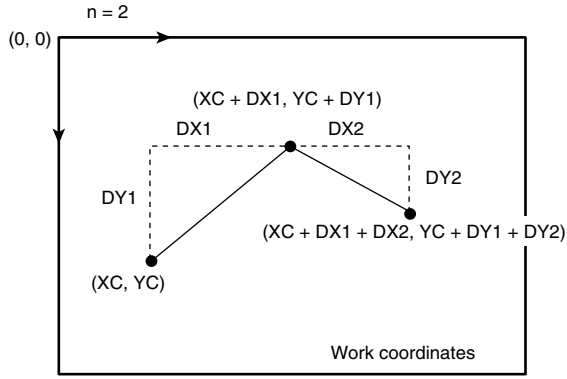
Description

Performs binary drawing at work coordinates of a polygonal line comprising line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer values (XC, YC). 0 drawing or 1 drawing is selected with the drawing mode EOS bit. Drawing is performed at work coordinates with 0 when EOS = 0, and at work coordinates with 1 when EOS = 1.

The final coordinate point is stored as the current pointer values (XC, YC). (Used for border drawing at work coordinates for a polygonal painted figure.)

Note: 8-point drawing is used.

Example

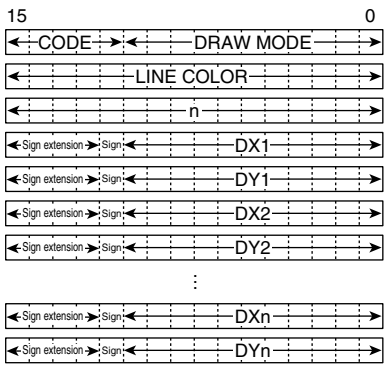


(8) LINE

Function

Draws a solid line 1 to 5 bits in width at rendering coordinates.

Command Format



1. Code

   B'01100

2. Rendering Attributes

Reference Data			Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work
			O	O	

DRAW MODE										
Reserved			CLIP	Reserved	NET	EOS	FWUL	W2UL	FWDR	W2DR
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	O	O	0000–1111			

O: Valid

3. Command Parameters

   LINE COLOR: 8- or 16-bit/pixel color specification

   n (n = 2 to 65,535): Number of vertices

   DXn (n = 2 to 65,535): Absolute values, rendering coordinates, negative numbers expressed as two's complement

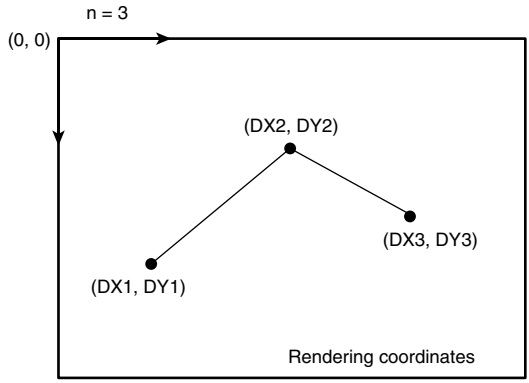
   DYn (n = 2 to 65,535): Absolute values, rendering coordinates, negative numbers expressed as two's complement

Description

Draws a polygonal line at rendering coordinates from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn), using the single color specified by parameter LINE COLOR.

Note: 8-point drawing is used.

Example

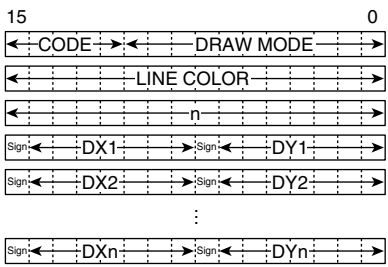


(9) RLINE

Function

Draws a solid line 1 to 5 bits in width at rendering coordinates.

Command Format



1. Code
   B'01101

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
			O	O	

Reserved			CLIP	Reserved	NET	EOS	FWUL	W2UL	FWDR	W2DR
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	O	O	0000–1111			

O: Valid

3. Command Parameters

   LINE COLOR: 8- or 16-bit/pixel color specification

   n (n = 1 to 65,535): Number of vertices

   DXn, DYn (n = 1 to 65,535): Relative values, rendering coordinates, negative numbers expressed as two's complement

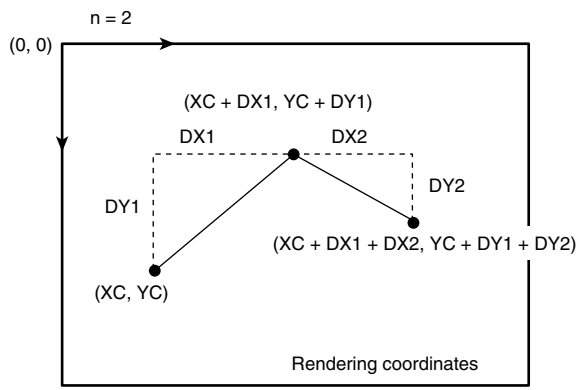
Description

Draws, at rendering coordinates, a polygonal line comprising line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn – 1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer values (XC, YC), using the single color specified by parameter LINE COLOR.

The final coordinate point is stored as the current pointer values (XC, YC).

Note: 8-point drawing is used.

Example

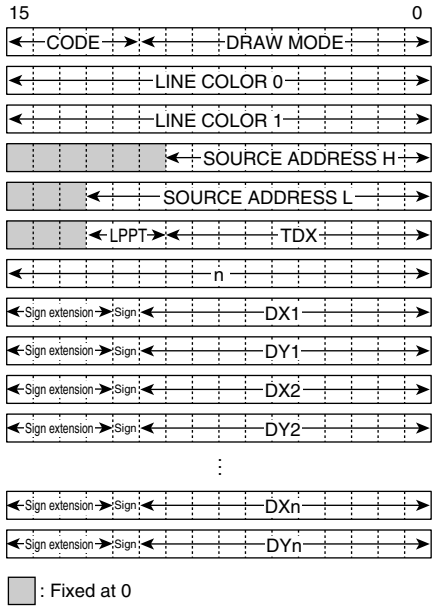


(10) PLINE

Function

Draws a polygonal line at rendering coordinates while referencing a binary source.

Command Format



1. Code
   B'01110

2. Rendering Attributes

Reference Data			Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
	O			O	

DRAW MODE

Reserved	TRNS	Reserved	CLIP	Reserved	NET	EOS	EDG2	Reserved	EDG1	Reserved
Fixed at 0	O	Fixed at 1	O	Fixed at 0	O	O	O	Fixed at 0	O	Fixed at 1

O: Valid

3. Command Parameters

   LINE COLOR0: 8- or 16-bit/pixel color specification

   LINE COLOR1: 8- or 16-bit/pixel color specification

   SOURCE ADDRESS H: 1-bit/pixel source start upper address (byte address)

   SOURCE ADDRESS L: 1-bit/pixel source start lower address (byte address)

   TDX: Source size

   LPPT: Line pattern pointer

   n (n = 2 to 65,535): Number of vertices

   DXn (n = 2 to 65,535): Absolute values, rendering coordinates, negative numbers expressed as two's complement

   DYn (n = 2 to 65,535): Absolute values, rendering coordinates, negative numbers expressed as two's complement

Description

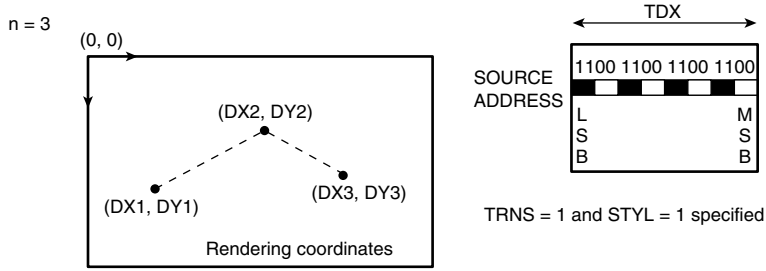
Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), ..., vertex n – 1 (DXn – 1, DYn – 1), to vertex n (DXn, DYn).

A multiple of 8 pixels must be set for the TDX value.

The reference start position of the binary source data can be adjusted by setting a value between 0 and 7 in the line pattern pointer. For example, if 0 is set, referencing starts at the beginning of the source data, while if 5 is set, referencing starts 5 pixels from the beginning of the source data. When STYL = 1, pattern repetition starts at the pixel after [source start position + TDX + LPPT – 1]. The source start address must be an even number.

Note: 4-point drawing is used.

Example

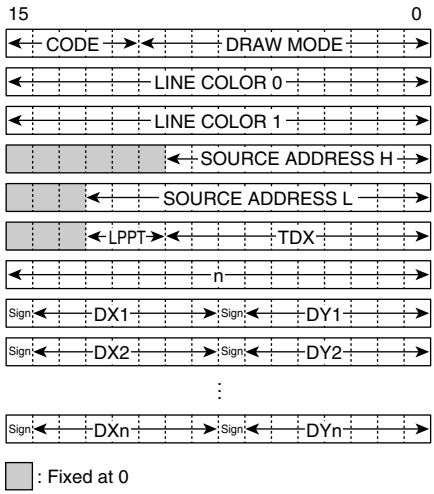


(11) RPLINE

Function

Draws a polygonal line at rendering coordinates while referencing a binary source.

Command Format



1. Code
   B'01111

2. Rendering Attributes

Reference Data			Drawing Destination		
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
	O			O	

DRAW MODE

Reserved	TRNS	Reserved	CLIP	Reserved	NET	EOS	EDG2	Reserved	EDG1	Reserved
Fixed at 0	O	Fixed at 1	O	Fixed at 0	O	O	O	Fixed at 0	O	Fixed at 1

O: Valid

3. Command Parameters

   LINE COLOR0: 8- or 16-bit/pixel color specification

   LINE COLOR1: 8- or 16-bit/pixel color specification

   SOURCE ADDRESS H: 1-bit/pixel source start upper address (byte address)

   SOURCE ADDRESS L: 1-bit/pixel source start lower address (byte address)

   LPPT: Line pattern pointer

   TDX: Source size

   n (n = 1 to 65,535): Number of vertices

   DXn, DYn (n = 1 to 65,535): Relative values, rendering coordinates, negative numbers expressed as two's complement

Description

Draws a polygonal line comprising line segments (XC, YC) – (XC + DX1, YC + DY1), (XC + DX1, YC + DY1) – (XC + DX1 + DX2, YC + DY1 + DY2), ..., (XC + ... + DXn – 1, YC + ... + DYn – 1) – (XC + ... + DXn –1 + DXn, YC + ... + DYn – 1 + DYn) to the coordinates specified by the relative shift (DX, DY) from the current pointer values (XC, YC).

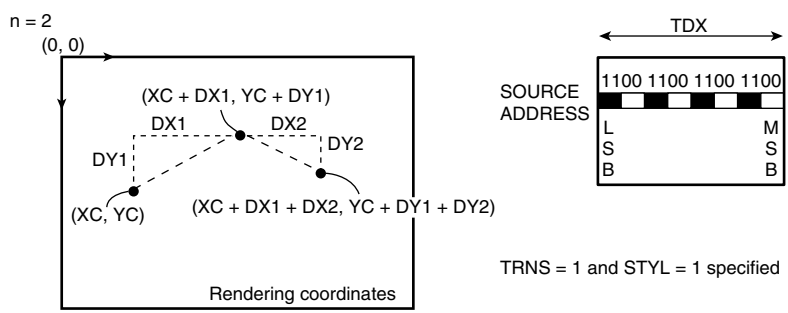
The final coordinate point is stored as the current pointer values (XC, YC).

A multiple of 8 pixels must be set for the TDX value.

The reference start position of the binary source data can be adjusted by setting a value between 0 and 7 in the line pattern pointer. For example, if 0 is set, referencing starts at the beginning of the source data, while if 5 is set, referencing starts 5 pixels from the beginning of the source data. When STYL = 1, pattern repetition starts at the pixel after [source start position + TDX + LPPT – 1]. The source start address must be an even number.

Note: 4-point drawing is used.

Example

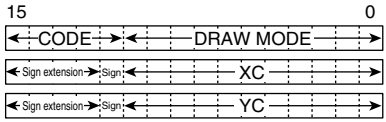


(12) MOVE

Function

Sets the current pointer.

Command Format



1. Code
   B'10000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

3. Command Parameters
   XC: Absolute value, rendering coordinate, work coordinate, negative number expressed as two's complement
   YC: Absolute value, rendering coordinate, work coordinate, negative number expressed as two's complement

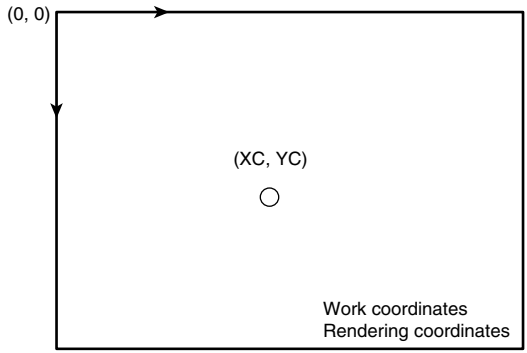
Description

Sets the values obtained by adding the local offset values to XC and YC in the current pointers. XC and YC are set as absolute coordinates. The current pointers are used by relative drawing commands only.

After issuing a MOVE command, use relative drawing commands in succession. If an absolute drawing command is used during this sequence, the current pointers will be used as registers for

internal computation, and the current pointer values will be lost. A MOVE command must be therefore be issued before using relative drawing commands again.

Example

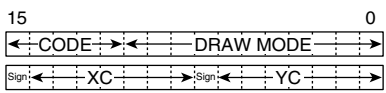


(13) RMOVE

Function

Sets the current pointer.

Command Format



1. Code
   B'10001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

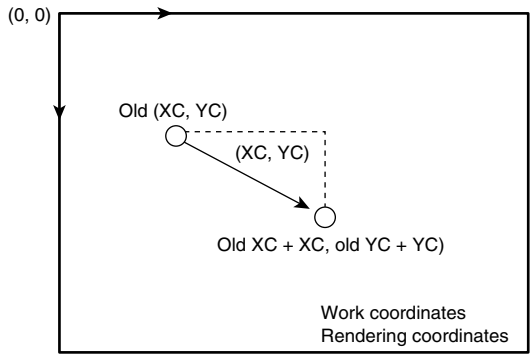
3. Command Parameters

   XC, YC: Relative values, rendering coordinates, work coordinates, and negative numbers expressed as two's complement

Description

Adds XC and YC to the current pointers.

Example

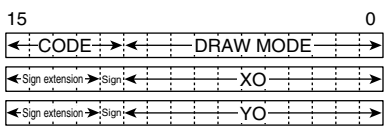


(14) LCOFS

Function

Sets the local offset.

Command Format



1. Code

   B'10010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**

Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

3. Command Parameters

   XO, YO: Local offset value absolute specifications, rendering coordinates, work coordinates, and negative numbers expressed as two's complement

Description

Sets the local offset with absolute coordinates. After these settings are made, these offset values are added in all subsequent coordinate specifications.

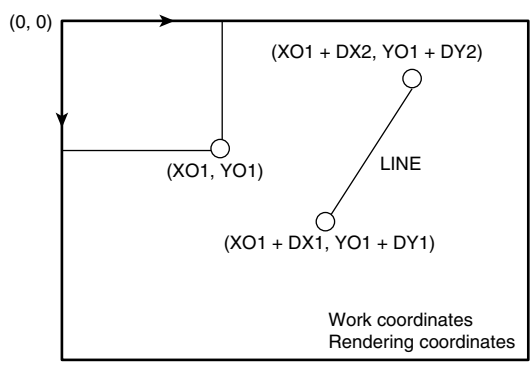
These settings must be made at the start of the display list (the initial values are undefined).

To reflect the local offset values in the current pointers, issue a MOVE command after the LCOFS command.

When using a command that employs the FST specification, a multiple of 4 must be set for the XO value.

Use non-negative values for both XO and YO.

Example

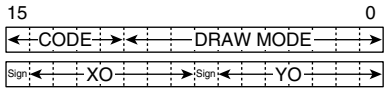


(15) RLCOFS

Function

Sets the local offset.

Command Format



1. Code
   B'10011

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

3. Command Parameters
   XO, YO: Local offset value relative specifications, rendering coordinates, work coordinates, and negative numbers expressed as two's complement

Description

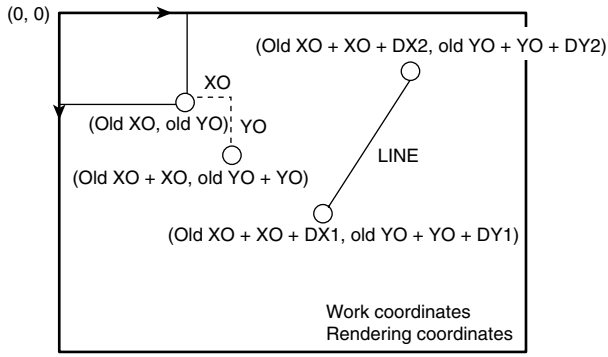
Adds XO and YO to the local offset. After these settings are made, these offset values are added in all subsequent coordinate specifications.

To reflect the local offset values in the current pointers, issue a MOVE command after setting the local offset with the LCOFS or RLCOFS command.

When using a command that employs the FST specification, the value obtained by adding XO to the local offset must be a multiple of 4.

The local offset values set by XO and YO must be non-negative.

Example

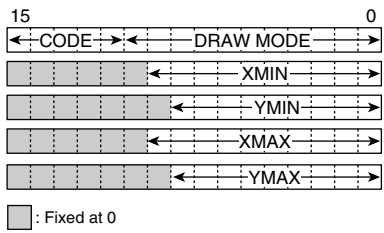


(16) UCLIP

Function

Sets the user clipping area.

Command Format



1. Code
   B'10101

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

3. Command Parameters

   XMIN, XMAX: Left and right X coordinate values, rendering coordinates, work coordinates

   YMIN, YMAX: Upper and lower Y coordinate values, rendering coordinates, work coordinates

Description

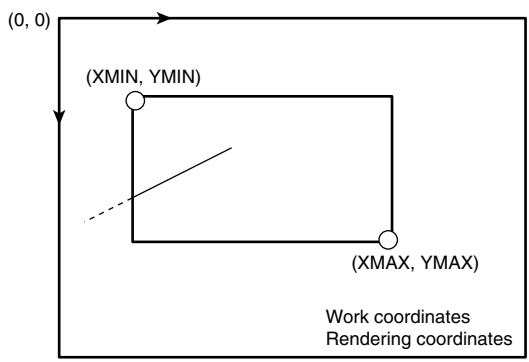
Designates the area specified by upper-left coordinates (XMIN, YMIN) and lower-right coordinates (XMAX, YMAX) in the rendering coordinate and work coordinate systems as a user clipping area. The local offset values specified by the LCOFS or RLCOFS command are not added to the coordinates set by this command.

When making these settings, ensure that XMIN < XMAX and YMIN < YMAX, and that the system clipping area is not exceeded.

This setting is valid when CLIP = 1.

When using a command that employs the FST specification, set a multiple of 4 as the XMIN value, and a multiple of 4 to 1 as the XMAX value.

Example

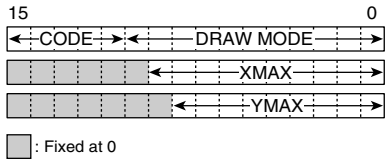


(17) SCLIP

Function

Sets the system clipping area.

Command Format



1. Code
   B'10111

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

3. Command Parameters
   XMAX: Left and right X coordinate values, rendering coordinates, work coordinates
   YMAX: Upper and lower Y coordinate values, rendering coordinates, work coordinates

Description

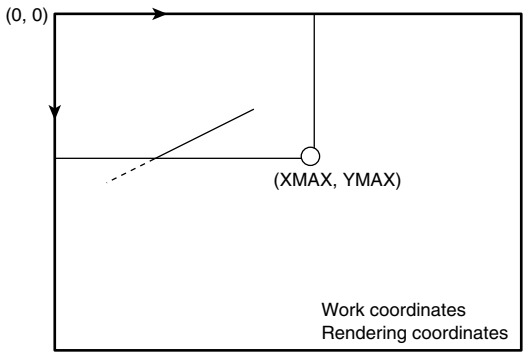
Designates the area specified by upper-left coordinates (0, 0) and lower-right coordinates (XMAX, YMAX) in the rendering coordinate and work coordinate systems as the system clipping area. The local offset values specified by the LCOFS or RLCOFS command are not added to the coordinates set by this command.

Set the maximum drawing range values for XMAX and YMAX. After powering on, the initial values of the clipping range are undefined. The clipping range must therefore be set with the SCLIP command at the start of the first display list executed.

For the set values given by this command, screen coordinates must be set as reference coordinates.

When using a command that employs the FST specification, set a multiple of 4 – 1 as the XMAX value.

Example

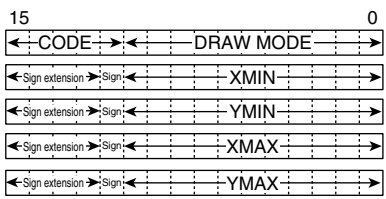


(18) CLRW

Function

Zeroizes the work coordinates.

Command Format



1. Code
   B'10100

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**
					O

Reserved			CLIP	Reserved						
Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid

3. Command Parameters

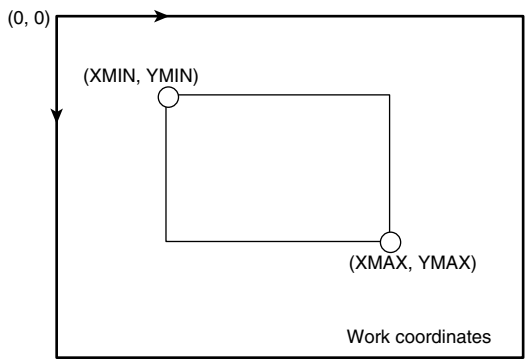
   XMIN, XMAX: Left and right X coordinate values, work coordinates, negative numbers expressed as two's complement

   YMIN, YMAX: Upper and lower Y coordinate values, work coordinates, negative numbers expressed as two's complement

Description

Zero-clears the area specified by upper-left coordinates (XMIN, YMIN) and lower-right coordinates (XMAX, YMAX) in the work coordinate system.

Example

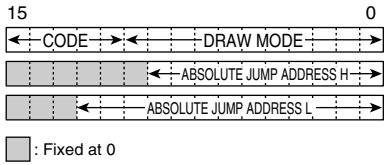


(19) JUMP

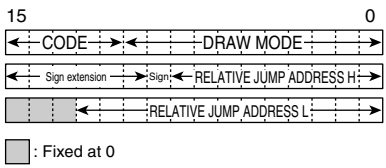
Function

Changes the display list fetch destination.

Command Format REL = 0



REL = 1



1. Code
   B'11000

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**

DRAW MODE

Reserved				REL	Reserved					
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid

3. Command Parameters

   ABSOLUTE/RELATIVE JUMP ADDRESS H: Absolute/relative jump destination upper address (byte address)

   ABSOLUTE/RELATIVE JUMP ADDRESS L: Absolute/relative jump destination lower address (byte address)

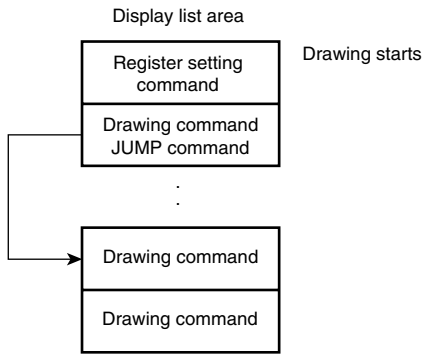
Description

Changes the display list fetch destination to the specified address.

When REL = 0, the jump destination address can be specified as an absolute address. When REL = 1, the source address can be specified as a relative address with respect to the UGM address at which the command code is located.

Absolute addresses and relative addresses must be even numbers. If a relative address is negative, its two's complement should be used.

Example

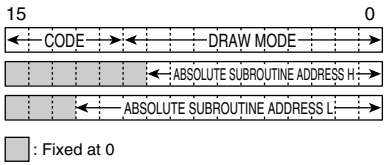


(20) GOSUB

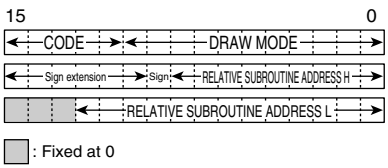
Function

Makes a subroutine call for the display list.

Command Format REL = 0



REL = 1



1. Code
   B'11001

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**

DRAW MODE

Reserved				REL	Reserved					
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	O	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

O: Valid

3. Command Parameters

   ABSOLUTE/RELATIVE SUBROUTINE ADDRESS H: Absolute/relative subroutine upper address (byte address)

   ABSOLUTE/RELATIVE SUBROUTINE ADDRESS L: Absolute/relative subroutine lower address (byte address)

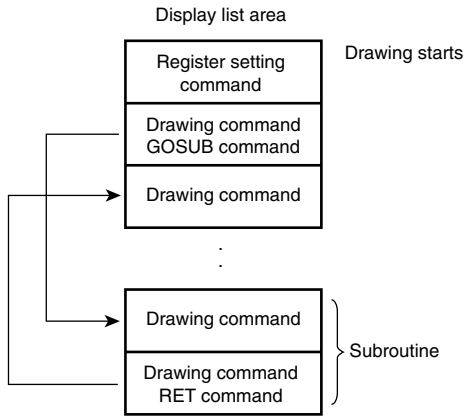
Description

Changes the display list fetch destination to the specified subroutine address. The fetch address is restored by a RET instruction. As only one level of nesting is permitted, it will not be possible to return if a subroutine call is issued within the subroutine.

When REL = 0, the subroutine destination address can be specified as an absolute address. When REL = 1, the address can be specified as a relative address with respect to the UGM address at which the command code is located.

Absolute addresses and relative addresses must be even numbers. If a relative address is negative, its two's complement should be used.

Example

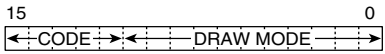


(21) RET

Function

Returns from a subroutine call made by the GOSUB command.

Command Format



1. Code
   B'11011

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

Description

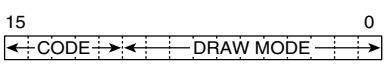
Restores the display list fetch destination to the address following the source of the subroutine call.

(22) TRAP

Function

Informs the Q2SD/RU of the end of the display list.

Command Format



1. Code
   B'11111

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**

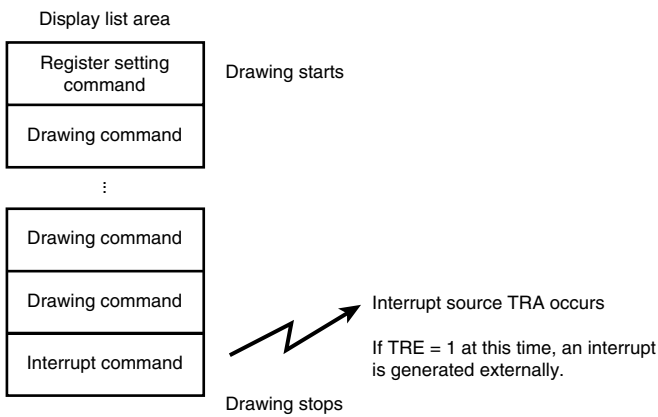
DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

Description

Halts the drawing operation and sets TRA to 1 in the status register (SR). If TRE is set to 1 in the interrupt enable register (IER), an interrupt is sent to CPU.

This command must be placed at the end of the display list.

Example

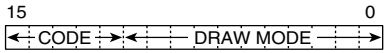


(23) NOP1

Function

Executes no operation.

Command Format



Command Parameters

1. DRAW MODE
   Fixed at all 0.

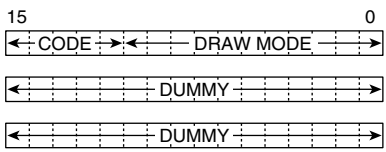
2. CODE
   B'11101

(24) NOP3

Function

Executes no operation.

Command Format



1. Code
   B'11110

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

Description

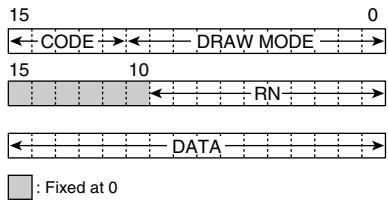
The NOP3 command does not perform any operation. This command, which consists of three words including the command code, simply fetches the next instruction without executing any processing.

(25) WPR

Function

Sets a value in a specific address-mapped register.

Command Format



1. Code
   B'10110

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	Binary Source	Binary Work	Specified Color	Rendering	Work

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

3. Command Parameters
   RN: Register number
   DATA: Data

Description

Writes data to the Q2SD/RU's address-mapped registers. The register number is set in RN, and the write data in DATA.

When a write is performed to an address-mapped register with this command, select the location to ensure that the currently executing drawing processing is not adversely affected.

Also ensure that there is no conflict with access by CPU.

This command is intended primarily for performing the operations shown in (a) to (d), and the registers that can be written to are limited to those listed below. If a write is performed to another register, subsequent operation cannot be guaranteed.

Register No.	Name
00E:	SSAR
00F:	WSAR
04C:	RSAR
006:	RMR
04A:	RTNH
04B:	RTNL

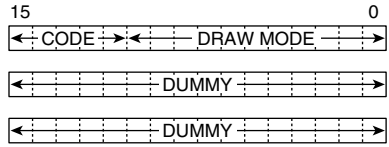
(a) Change of drawing start address (RN = 04C)
(b) Change of multi-valued source or work start address (RN = 00E, 00F)
(c) Change of graphic bit mode (RN = 006)
(d) Return address setting when performing resumption processing after drawing suspension (RN = 04A, 04B)

(26) VBKEM

Function

Performs synchronization with the frame change timing.

Command Format



1. Code
   B'11010

2. Rendering Attributes

Reference Data				Drawing Destination	
Multi-Valued Source	**Binary Source**	**Binary Work**	**Specified Color**	**Rendering**	**Work**

DRAW MODE										
Reserved										
Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0	Fixed at 0

Description

When this command is executed, the drawing operation is kept waiting until "du_vbkemclr" signal from display out module comes. This pulse signal is generated at the timing of the fall of the vertical sync signal.

11.3.5 2DGE Drawing Commands

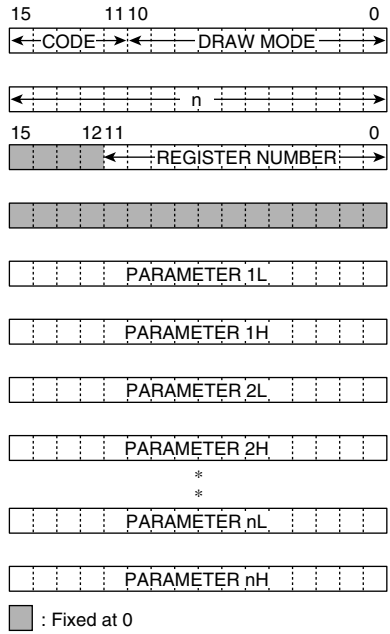
All parameters for anti-alias font drawing and Bit Block Transfer (BitBLT) with 16 raster operations must be set to 2DGE (2D graphics engine) registers. The GE provides the command which can set 2D graphics parameters to 2DGE registers.

W2DP:

Function

Writes 2D graphics parameters to 2DGE registers.

Command Format



Command Parameters

1. DRAW MODE
   Fixed at all 0.

2. CODE
   B'00100

3. Command Parameters
   This field defines the number of parameter registers to write. The maximum value of this field is 16. Setting 0 to this field is prohibited.

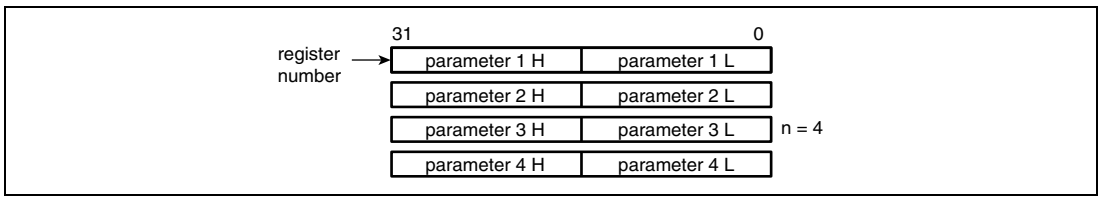


Figure 11.40 Set Parameters to Parameter Register 1 – Parameter Register 4

REGISTER NUMBER:

This field defines the first parameter register number listed below. The W2DP command performs a sequential parameter setting in low to high number order, starting at the first parameter register number. The setting of the unlisted register number is prohibited.

Anti-alias Font Command Registers	Register Number
reserved	100
reserved	104
Foreground Color	108
Destination Position	10C
a_value Source Address	110
a_value Source Size	114
reserved	118
Rendering Command	11C

BitBLT Command Registers	Register Number
reserved	140
reserved	144
reserved	148
Destination Position	14C
Source Position	150
Source Size	154
reserved	158
Rendering Command	15C

Command Common Registers	Register Number
User Clipping Area MIN	900
User Clipping Area MAX	904
System Clipping Area MAX	908
Destination Local Offset	90C
Source/Destination Stride	910
Source Transparent Color	A00
Destination Transparent Color	A04
Source Base Address	B00
Destination Base Address	B04

PARAMETER 1 – PARAMETER n:

2D graphics parameters for anti-alias font command registers, BitBLT command registers and command common registers. Reserved register must be set to all 0.

Note: Once parameters are set to 2DGE Registers by W2DP, 2DGE Registers keep the values until the next parameter setting by W2DP command (2DGE Registers are initialized after reset.). For example, if all parameters related to anti-alias font drawing need not to be changed, anti-alias font drawing can be executed by setting the value to Command Register For Anti-alias Font Command only. If destination position of anti-alias font drawing need to be changed, anti-alias font drawing can be executed by setting the values to Destination Position Register and Command Register For Anti-alias Font Command.

11.4 Register Description

11.4.1 Overview

The GE mainly consists of Q2SD/RU (Rendering Unit) and 2DGE (2-dimension Graphics Engine). Q2SD/RU and 2DGE have System Control Registers in common, and have Drawing Parameter Registers independently.

Note: 2DGE Registers

2DGE Registers are classified in 3 groups, "Anti-alias Font Command Registers", "BitBLT Command Registers" and "Command Common Registers". The parameters, which need to be changed every time anti-alias font drawing is executed, are gathered in "Anti-alias Font Command Registers". In the same way, the prameters, which need to be changed every time BitBLT is executed, are gathered in "Bit BLT Command Registers". the parameters, which need not to be changed frequently, are gathered in "Command Common Registers".

2DGE executes anti-alias font drawing and BitBLT with 16 raster operations. All parameters required by these two operations must be set to 2DGE Registers by W2DP command. 2DGE Registers are CPU read only registers. However, there are exceptions for Source/Destination Stride Register, Source Transparent Color Register, Destination Color Register, Source Base Address Register and Destination Base Address Register. These registers are CPU read/write registers.

Note: CPU Register Access Limitation

The register read/write operation by CPU is prohibited during the GE drawing operation. These are exceptions for Rendering Control Register and Status Register. CPU can access these registers during the GE drawing operation. However, setting RS bit in Rendering Control Register is prohibited during the GE drawing operation.

(GE drawing operation period: The starting of drawing operation is initiated by setting RS bit in Rendering Control Register. The end of drawing operation is indicated by TRA = 1 or BRK = 1 or CER = 1 in Status Register.)

Note: Register Access Size

Access size of GE Registers is longword (32-bit) only. Byte and word accesses are not allowed.

(1) System Control Registers

Addr. (byte)	CPU R/W	Reg. Name	Abbr.
H'2000	R/W	Rendering Control	RCR
H'2004	R	Status	SR
H'2008	W	Status Register Clear	SRCR
H'200C	R/W	Interrupt Enable	IER
H'0030	R/W	Display List Area Start Address H	DLSAR
H'0034	R/W	Display List Area Start Address L	
H'007C	R	Command Status H	CSTR
H'0080	R	Command Status L	
H'0128	R/W	Return Address H	RTNR
H'012C	R/W	Return Address L	

(2) Q2SD/RU Registers

Addr. (byte)	CPU R/W	Reg. Name	Abbr.
H'1018	R/W	Rendering Mode	RMR
H'1038	R/W	Source Area Start Address	SSAR
H'103C	R/W	Work Area Start Address	WSAR
H'1130	R/W	Rendering Area Start Address	RSAR
H'0100	R	Current Pointer X	CURR
H'0104	R	Current Pointer Y	
H'0108	R	Local Offset X	LCOR
H'010C	R	Local Offset Y	
H'0110	R	User Clipping Area XMIN	UCLR
H'0114	R	User Clipping Area YMIN	
H'0118	R	User Clipping Area XMAX	
H'011C	R	User Clipping Area YMAX	
H'0120	R	System Clipping Area XMAX	SCLR
H'0124	R	System Clipping Area YMAX	

(3) 2DGE Registers

Anti-alias Font Command Registers

Addr. (byte)	CPU R/W	Reg. Name	Abbr.
H'2100	—	reserved	—
H'2104	—	reserved	—
H'2108	R	Foreground Color	—
H'210C	R	Destination Position	—
H'2110	R	a_value Source Address	—
H'2114	R	a_value Source Size	—
H'2118	—	reserved	—
H'211C	R	Command	—

BitBLT Command Registers

Addr. (byte)	CPU R/W	Reg. Name	Abbr.
H'2140	—	reserved	—
H'2144	—	reserved	—
H'2148	—	reserved	—
H'214C	R	Destination Position	—
H'2150	R	Source Position	—
H'2154	R	Source Size	—
H'2158	—	reserved	—
H'215C	R	Command	—

Addr. (byte)	CPU R/W	Reg. Name	Abbr.
H'2900	R	User Clipping Area MIN	—
H'2904	R	User Clipping Area MAX	—
H'2908	R	System Clipping Area MAX	—
H'290C	R	Destination local Offset	—
H'2910	R/W	Source/Destination Stride	—
	—	reserved	—
H'2A00	R/W	Source Transparent Color	—
H'2A04	R/W	Destination Transparent Color	—
	—	reserved	—
H'2B00	R/W	Source Base Address	—
H'2B04	R/W	Destination Base Address	—

(4) System Control Registers #1

	Rendering Control	Status	Status Register Clear	Interrupt Enable
0	rendering start	Trap Flag	Trap Flag Clear	Trap Flag Enable
1	rendering break	Drawing Break Flag	Drawing Break Flag Clear	Drawing Break Flag Enable
2		Command Error Flag	Command Error Flag clear	Command Error Flag Enable
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31	software reset			

(4) System Control Registers #2

	Display List Area Start Address H	Display List Area Start Address L	Command Status H	Command Status L	Return Address H	Return Address L
0	address A16		address A16		address A16	
1	address A17		address A17	address A1	address A17	address A1
2	address A18		address A18	address A2	address A18	address A2
3	address A19		address A19	address A3	address A19	address A3
4	address A20		address A20	address A4	address A20	address A4
5	address A21	address A5	address A21	address A5	address A21	address A5
6	address A22 (8 MB)	address A6	address A22 (8 MB)	address A6	address A22 (8 MB)	address A6
7		address A7		address A7		address A7
8		address A8		address A8		address A8
9		address A9		address A9		address A9
10		address A10		address A10		address A10
11		address A11		address A11		address A11
12		address A12		address A12		address A12
13		address A13		address A13		address A13
14		address A14		address A14		address A14
15		address A15		address A15		address A15
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						

(5) Q2SD/RU Registers #1

	Rendering Mode	Source Area Start Address	Work Area Start Address	Rendering Area Start Address
0	Graphic Bit Mode	address A16	address A16	address A16
1		address A17	address A17	address A17
2		address A18	address A18	address A18
3		address A19	address A19	address A19
4	Memory Width X	address A20	address A20	address A20
5		address A21	address A21	address A21
6		address A22 (8MB)	address A22 (8MB)	address A22 (8MB)
7				
8				
9				
10				
11				
12				
13		address A13	address A13	
14		address A14	address A14	
15		address A15	address A15	
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				

(5) Q2SD/RU Registers #2

	Current Pointer X	Current Pointer Y	Local Offset X	Local Offset Y	User Clipping Area XMIN	User Clipping Area YMIN	User Clipping Area XMAX	User Clipping Area YMAX	System Clip Area XMAX	System Clip Area YMAX
0	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
1	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
2	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
3	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
4	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
5	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
6	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
7	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
8	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
9	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
10	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
11	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
12	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
13	XC	YC	XO	YO	UXMIN	UYMIN	UXMAX	UYMAX	SXMAX	SYMAX
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										

	Current Pointer X	Current Pointer Y	Local Offset X	Local Offset Y	User Clipping Area XMIN	User Clipping Area YMIN	User Clipping Area XMAX	User Clipping Area YMAX	System Clip Area XMAX	System Clip Area YMAX
28										
29										
30										
31										
	(2's comple ment)	(2's comple ment)	(2's comple ment)	(2's comple ment)						

(6) 2DGE Registers/Anti-alias Font Command Registers

	Foreground Color	Destination Position	a_value Source Address	a_value Source Size	Command
0	blue (16bpp)	destination x	address (=0)	width (=1)	
1	blue (16bpp)	destination x	address (=0)	width (=1)	
2	blue (16bpp)	destination x	address (=0)	width (=1)	
3	blue (16bpp)	destination x	address	width	
4	blue (16bpp)	destination x	address	width	
5	green (16bpp)	destination x	address	width	
6	green (16bpp)	destination x	address	width	
7	green (16bpp)	destination x	address	width	
8	green (16bpp)	destination x	address	width	
9	green (16bpp)	destination x	address	width (8-1024 pixels-1)	
10	green (16bpp)	destination x	address		destination transparency enable
11	red (16bpp)	destination x ((-2048)-2047)	address		destination transparency polarity
12	red (16bpp)		address		
13	red (16bpp)		address		
14	red (16bpp)		address		
15	red (16bpp)		address		
16		destination y	address	height	
17		destination y	address	height	
18		destination y	address	height	
19		destination y	address	height	
20		destination y	address	height	

	Foreground Color	Destination Position	a_value Source Address	a_value Source Size	Command
21		destination y	address	height	
22		destination y	address (8MB)	height	
23		destination y		height	
24		destination y		height	
25		destination y		height (1-1024 lines-1)	(=1)
26		destination y			
27		destination y ((-2048)-2047)			
28					
29					
30					clipping enable
31					
		(2's complement)			

	Destination Position	Source Position	Source Size	Command
0	destination x	source x	width	ROP code
1	destination x	source x	width	ROP code
2	destination x	source x	width	ROP code
3	destination x	source x	width	ROP code
4	destination x	source x	width	ROP code
5	destination x	source x	width	ROP code
6	destination x	source x	width	ROP code
7	destination x	source x	width	ROP code
8	destination x	source x	width	source transparency enable
9	destination x	source x (0-1023)	width (1-1024 pixels -1)	source transparency polarity
10	destination x			destination transparency enable
11	destination x ((-2048)-2047)			destination transparency polarity
12				
13				
14				BitBLT x direction
15				BitBLT y direction
16	destination y	source y	height	
17	destination y	source y	height	
18	destination y	source y	height	
19	destination y	source y	height	
20	destination y	source y	height	
21	destination y	source y	height	
22	destination y	source y	height	
23	destination y	source y	height	
24	destination y	source y	height	tile addressing source
25	destination y	source y (0-1023)	height (1-1024 lines-1)	(=1)
26	destination y			
27	destination y ((-2048)-2047)			
28				
29				
30				clipping enable
31				

(2's complement)

	User Clipping Area MIN	User Clipping Area MAX	System Clipping Area MAX	Destination Local Offset
0	umin x	umax x	smax x	offset x
1	umin x	umax x	smax x	offset x
2	umin x	umax x	smax x	offset x
3	umin x	umax x	smax x	offset x
4	umin x	umax x	smax x	offset x
5	umin x	umax x	smax x	offset x
6	umin x	umax x	smax x	offset x
7	umin x	umax x	smax x	offset x
8	umin x	umax x	smax x	offset x
9	umin x (0-1023)	umax x (0-1023)	smax x (0-1023)	offset x
10				offset x
11				offset x ((-2048)-2047)
12				
13				
14				
15				
16	umin y	umax y	smax y	offset y
17	umin y	umax y	smax y	offset y
18	umin y	umax y	smax y	offset y
19	umin y	umax y	smax y	offset y
20	umin y	umax y	smax y	offset y
21	umin y	umax y	smax y	offset y
22	umin y	umax y	smax y	offset y
23	umin y	umax y	smax y	offset y
24	umin y	umax y	smax y	offset y
25	umin y (0-1023)	umax y (0-1023)	smax y (0-1023)	offset y
26				offset y
27				offset y ((-2048)-2047)
28				
29				
30				
31				

(2's complement)

(9) 2DGE Registers/Command Common Registers #2

	Source/ Destination Stride	Source/ Destination Stride	Source Transparent Color	Destination Transparent Color	Source Base Address	Destination Base Address
0	destination stride (=1)	destination stride (=1)	blue (16bpp)/palette (8bpp)	blue (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
1	destination stride (=1)	destination stride (=1)	blue (16bpp)/palette (8bpp)	blue (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
2	destination stride (=1)	destination stride (=1)	blue (16bpp)/palette (8bpp)	blue (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
3	destination stride (=1)	destination stride (=1)	blue (16bpp)/palette (8bpp)	blue (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
4	destination stride (=1)	destination stride (=1)	blue (16bpp)/palette (8bpp)	blue (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
5	destination stride (=1)	destination stride (=1)	green (16bpp)/palette (8bpp)	green (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
6	destination stride (=1)	destination stride (=1)	green (16bpp)/palette (8bpp)	green (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
7	destination stride (=1)	destination stride (=1)	green (16bpp)/palette (8bpp)	green (16bpp)/palette (8bpp)	address (tile addressing source = 0)	
8	destination stride (=1)	destination stride (=1)	green (16bpp)	green (16bpp)	address (tile addressing source = 0)	
9	destination stride (=0/1,512/1024 pixels -1)	destination stride (=0/1,512/1024 pixels -1)	green (16bpp)	green (16bpp)	address (tile addressing source = 0)	
10			green (16bpp)	green (16bpp)	address (tile addressing source = 0)	
11			red (16bpp)	red (16bpp)	address (tile addressing source = 0)	
12			red (16bpp)	red (16bpp)	address (tile addressing source = 0)	
13			red (16bpp)	red (16bpp)	address	

	Source/ Destination Stride	Source/ Destination Stride	Source Transparent Color	Destination Transparent Color	Source Base Address	Destination Base Address
14			red (16bpp)	red (16bpp)	address	
15			red (16bpp)	red (16bpp)	address	
16	source stride	source stride(=1)			address	address
17	source stride	source stride(=1)			address	address
18	source stride	source stride(=1)			address	address
19	source stride	source stride(=1)			address	address
20	source stride	source stride(=1)			address	address
21	source stride	source stride(=1)			address	address
22	source stride	source stride(=1)			address (8MB)	address (8MB)
23	source stride	source stride(=1)				
24	source stride	source stride(=1)				
25	source stride(1-1024 pixels -1)	source stride (=0/1, 512/1024 pixels -1)				
26						
27						
28						
29	color format	color format				
30						
31						
	Note: (tile addressing source = 0)	Note: (tile addressing source = 1) (source stride = destination stride)			"Note: Source base address bit 0 must be set to zero if pixel data format is 16-bit/pixel."	

11.4.2 System Control Registers

Legends for register description:

Initial Value : Register value after hardware reset (rsth_ru) or software reset (SRES in SR)
— : Reserved bit (read: undefined value, write: 0)

* : Undefined values

(1) Rendering Control Register (RCR)

Register Address (Byte): H'2000

The Rendering Control Register (RCR) is a 32-bit readable/writable register that specifies GE system control.

Bit:	31	30	29	· ·	3	2	1	0
	SRES	—	—	· ·	—	—	RBRK	RS
Initial value:	1	—	—	· ·	—	—	0	0
R/W:	R/W	—	—	· ·	—	—	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	SRES	1	R/W	**Software Reset (SRES)** Resets GE drawing operation. 0: GE drawing operation is enabled. 1: GE drawing operation is reset. All GE registers are reset to initial values. The SRES bit must be set to 1 for at least 1 VSYNC cycle.
30 to 2	—	—	—	**Reserved** Only 0 should be written to these bits.

Bit	Bit Name	Initial Value	R/W	Description
1	RBRK	0	R/W	**Rendering Break (RBRK)** Controls rendering (drawing) break. 0: The TRA bit in the status register (SR) is set to 1 by TRAP command execution, and drawing is terminated. 1: When the RBRK bit is set to 1 while the GE is performing the drawing operation, the GE drawing operation is suspended after the GE completes the current drawing command, and fetches the next drawing command. At the same time, the RBRK bit is cleared to 0, the BRK bit in the Status Register (SR) is set to 1, and the start address of the next command is stored in the Command Status Register (CSTR). When the RBRK is set to 1 when the GE is "not" performing the drawing operation, nothing happens (Note: The RBRK bit is "not" cleared to 0). The RBRK bit must be set to 1 when the BRK bit in the Status Register (SR) is cleared to 0.
0	RS	0	R/W	**Rendering Start (RS)** Specifies the start of rendering. 0: Rendering is not started. 1: Rendering is started. This bit is cleared to 0 after rendering starts.

(2) Status Register (SR)

Register Address (Byte): H'2004

The Status Register (SR) is a 32-bit read-only register used to read the internal status of the GE from outside.

Bit:	31	30	29	. . . . .	3	2	1	0
	—	—	—	. . . . .	—	CER	BRK	TRA
Initial value:	—	—	—	. . . . .	—	0	0	0
R/W:	—	—	—	. . . . .	—	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	—	—	**Reserved** These bits are always read as 0.
2	CER	0	R	**Command Error Flag (CER)** Flag that indicates that an illegal command has been fetched. 0: An illegal command has not been fetched. 1: The GE drawing operation is halted because an illegal command having undefined command code is fetched. The CER flag can be cleared by setting the CECL bit in SRCR or the SRES bit in RCR. Note: The CER flag is not set if the command having the command code "00011" or "00111" is fetched. These command codes are used for test commands.
1	BRK	0	R	**Drawing Break Flag (BRK)** Flag that indicates a drawing break. 0: No drawing break occurs. 1: The GE drawing operation is suspended by setting the RBRK bit in RCR. The BRK flag can be cleared by setting the BRCL bit in SRCR or the SRES bit in RCR.
0	TRA	0	R	**Trap Flag (TRA)** Flag that indicates the end of command execution. 0: Indicates the interval from TRA flag cleared by the SRES bit in RCR or the TRCL bit in SRCR to the end of the next TRAP command execution. 1: Indicates the end of the TRAP command execution. The TRA flag can be cleared by setting the TRCL bit in SRCR or the SRES bit in RCR.

(3) Status Register Clear Register (SRCR)

Register Address (Byte): H'2008

The Status Register Clear Register (SRCR) is a 32-bit write-only register that clears the corresponding flags in the status register (SR). When the SR is cleared, the SRCR is cleared to all-0 internally.

Bit:	31	30	29	. . . . .	3	2	1	0
	—	—	—	. . . . .	—	CECL	BRCL	TRCL
Initial value:	—	—	—	. . . . .	—	*	*	*
R/W:	—	—	—	. . . . .	—	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	—	—	**Reserved** Only 0 should be written to these bits.
2	CECL	Undefined	W	**Command error flag clear (CECL)** Writing 1 to the CECL bit clears the CER flag to 0 in SR.
1	BRCL	Undefined	W	**Drawing break flag clear (BRCL)** Writing 1 to the BRCL bit clears the BRK flag to 0 in SR.
0	TRCL	Undefined	W	**Trap flag clear (TRCL)** Writing 1 to the TRCL bit clears the TRA flag to 0 in SR.

(4) Interrupt Enable Register (IER)

Register Address (Byte): H'200C

The Interrupt Enable Register (IER) is a 32-bit readable/writable register that enables or disables interrupts by the corresponding flags in the Status Register (SR). When a bit in SR is set to 1 and the bit at the corresponding bit position in IER is also 1, ru_irq is driven high.

ru_irq = (TRA & TRE) II (BRK & BRE) II (CER & CEE)

Bit:	31	30	29	. . . . .	3	2	1	0
	—	—	—	. . . . .	—	CEE	BRE	TRE
In itial value:	—	—	—	. . . . .	—	0	0	0
R/W:	—	—	—	. . . . .	—	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	—	—	**Reserved** Only 0 should be written to these bits.
2	CEE	0	R/W	**Command Error Flag Enable (CEE)** Enables or disables interrupt initiated by the CER flag in SR. 0: Interrupts initiated by the CER flag in SR are disabled. 1: Interrupts initiated by the CER flag in SR are enabled. When CER·CEE = 1, an interrupt request is generated.
1	BRE	0	R/W	**Drawing Break Flag Enable (BRE)** Enables or disables interrupt initiated by the BRK flag in SR. 0: Interrupts initiated by the BRK flag in SR are disabled. 1: Interrupts initiated by the BRK flag in SR are enabled. When BRK·BRE = 1, an interrupt request is generated.
0	TRE	0	R/W	**Trap Flag Enable (TRE)** Enables or disables interrupt initiated by the TRA flag in SR. 0: Interrupts initiated by the TRA flag in SR are disabled. 1: Interrupts initiated by the TRA flag in SR are enabled. When TRA·TRE = 1, an interrupt request is generated.

(5) Display List Area Start Address Registers (DLSAR)

Register Address (Byte): H'0030, H'0034

The Display List Area Start Address Registers (DLSAR) are 32-bit readable/writable registers that specify the memory area to be used as the display list.

The upper bits (A22 to A16) of the start address are set in the DLSAH field, and the lower bits (A15 to A5) in the DLSAL field.

Bit:	31	· · · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · · ·	—	—	—	—	—	—	—	—	—	DLSAH (address A22 to A16 setting)						
Initial value:	—	· · · · · · · ·	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
R/W:	—	· · · · · · · ·	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	31	· · · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · · ·	DLSAL (address A15 to A5 setting)											—	—	—	—	—
Initial value:	—	· · · · · · · ·	*	*	*	*	*	*	*	*	*	*	*	—	—	—	—	—
R/W:	—	· · · · · · · ·	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—	—	—	—	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	—	Reserved
6 to 0	DLSAH	Undefined	R/W	Address A22 to A16 Setting

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
15 to 5	DLSAL	Undefined	R/W	Address A15 to A5 Setting
4 to 0	—	—	—	Reserved

(6) Command Status Registers (CSTR)

Register Address (Byte): H'007C, H'0080

The Command Status Registers (CSTR) are 32-bit read-only registers that store the address of the command word (op code word) being executed.

The upper bits (A22 to A16) of the command word address are set in the CSTH field, and the lower bits (A15 to A1) in the CSTL field. The address indicated by the CSTH and CSTL fields is a word address.

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · ·	—	—	—	—	—	—	—	—	—	CSTH (address A22 to A16 setting)						
Initial value:	—	· · · · · ·	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
R/W:	—	· · · · · ·	—	—	—	—	—	—	—	—	—	R	R	R	R	R	R	R

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · ·	CSTL (address A15 to A1 setting)															—
Initial value:	—	· · · · · ·	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	—
R/W:	—	· · · · · ·	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	—	Reserved
6 to 0	CSTH	Undefined	R	Address A22 to A16 Setting

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Reserved
15 to 1	CSTL	Undefined	R	Address A15 to A1 Setting
0	—	—	—	Reserved

(7) Return Address Registers (RTNR)

Register Address (Byte): H'0128, H'012C

The Return Address Registers (RTNR) are 32-bit readable/writable registers that specify the return address.

The upper bits (A22 to A16) of the return address are set in the RTNH field, and the lower bits (A15 to A1) in the RTNL field. The address (A22 to A1) indicated by the RTNH and RTNL fields is a word address.

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · ·	—	—	—	—	—	—	—	—	—	RTNH (address A22 to A16 setting)						
Initial value:	—	· · · · · · ·	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · ·	RTNL (address A15 to A1 setting)															—
Initial value:	—	· · · · · · ·	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	—
R/W:	—	· · · · · · ·	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	—

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	—	Reserved
6 to 0	RTNH	Undefined	R/W	Address A22 to A16 Setting

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Reserved
15 to 1	RTNL	Undefined	R/W	Address A15 to A1 Setting
0	—	—	—	Reserved

11.4.3 Q2SD/RU Registers

Legends for register description:

Initial Value: Register value after hardware reset (rsth_ru) or software reset (SRES in SR)
—: Reserved bit (read: undefined value, write: 0)

(1) Rendering Mode Register (RMR)

Register Address (Byte): H'1018

The Rendering Mode Register (RMR) is a 32-bit readable/writable register that specifies Q2SD/RU rendering operations.

If the register value is changed during a drawing operation, the operation does not work correctly.

Bit:	31	. . . . .	4	3	2	1	0
	—	. . . . .	MWX	—	—	—	GBM
Initial value:	—	. . . . .	*	—	—	—	*
R/W:	—	. . . . .	R/W	—	—	—	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	—	—	**Reserved**
4	MWX	Undefined	R/W	**Memory Width (MWX)** Specifies the X-direction logical coordinate space of the graphics memory. 0: X-direction logical coordinate space is 512 pixels 1: X-direction logical coordinate space is 1024 pixels
3 to 1	—	—	—	**Reserved**
0	GBM	Undefined	R/W	**Graphic Bit Mode (GBM)** Specifies the bit configuration of the drawing data handled by the Q2SD/RU. 0: 8 bits/pixel 1: 16 bits/pixel

(2) Source Area Start Address Register (SSAR)

Register Address (Byte): H'1038

The Source Area Start Address Register (SSAR) is a 32-bit readable/writable register that specifies the memory area to be used as the multi-valued source area. The upper bits (A22 to A16) of the start physical address of the source area are set in the SSAH field, and the lower bits (A15 to A13) in the SSAL field.

The settable bit range depends on the pixel data format and maximum memory width. In 8-bit/pixel mode with a 512-pixel memory width, all bits can be set. In 8-bit/pixel mode with a 1024-pixel memory width, or 16-bit/pixel mode with a 512-pixel memory width, bit 13 should be cleared to 0. In 16-bit/pixel mode with a 1024-pixel memory width, bits 14 and 13 should be cleared to 0.

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · ·	SSAL (address A15 to A13 setting)			—	—	—	—	—	—	SSAH (address A22 to A16 setting)						
Initial value:	—	· · · · · · ·	*	*	*	—	—	—	—	—	—	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	R/W	R/W	R/W	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

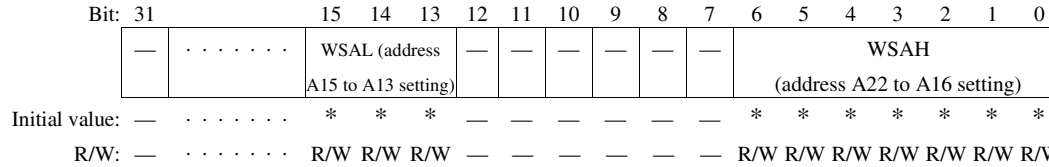
Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Reserved
15 to 13	SSAL	Undefined	R/W	Address A15 to A13 Setting
12 to 7	—	—	—	Reserved
6 to 0	SSAH	Undefined	R/W	Address A22 to A16 Setting

(3) Work Area Start Address Register (WSAR)

Register Address (Byte): H'103C

The Work Area Start Address Register (WSAR) is a 32-bit readable/writable register that specifies the memory area to be used as the work area. The upper bits (A22 to A16) of the start physical address of the work area are set in the WSAH field, and the lower bits (A15 to A13) in the WSAL field.

The settable bit range depends on the pixel data format and maximum memory width. In 8-bit/pixel mode with a 512-pixel memory width, all bits can be set. In 8-bit/pixel mode with a 1024-pixel memory width, or 16-bit/pixel mode with a 512-pixel memory width, bit 13 should be cleared to 0. In 16-bit/pixel mode with a 1024-pixel memory width, bits 14 and 13 should be cleared to 0.



Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	—	Reserved
15 to 13	WSAL	Undefined	R/W	Address A15 to A13 Setting
12 to 7	—	—	—	Reserved
6 to 0	WSAH	Undefined	R/W	Address A22 to A16 Setting

(4) Rendering Start Address Register (RSAR)

Register Address (Byte): H'1130

The Rendering Start Address Register (RSAR) is a 32-bit readable/writable register that specifies the start address of the rendering area.

Only the upper 7 bits (A22 to A16) of the start physical address of the rendering area are set in the RSA field.

Bit:	31	· · · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · · ·	—	—	—	—	—	—	—	—	—	RSA (address A22 to A16 setting)						
Initial value:	—	· · · · · · · ·	—	—	—	—	—	—	—	—	—	*	*	*	*	*	*	*
R/W:	—	· · · · · · · ·	—	—	—	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	—	Reserved
6 to 0	RSA	Undefined	R/W	Address A22 to A16 Setting

(5) Current Pointer Registers (CURR)

The Current Pointer Registers (CURR) are two 32-bit read-only registers that indicate the current pointer coordinates.

Bit 13 is the sign bit.

Register Address (Byte): H'0100

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · · ·	—	—	XC													
Initial value:	—	· · · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	XC	Undefined	R	

Register Address (Byte): H'0104

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · ·	—	—	YC													
Initial value:	—	· · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	YC	Undefined	R	

(6) Local Offset Registers (LCOR)

The Local Offset Registers (LCOR) are two 32-bit read-only registers that indicate the offset coordinates.

Bit 13 is the sign bit.

Register Address (Byte): H'0108

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · ·	—	—	XO													
Initial value:	—	· · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	XO	Undefined	R	

Register Address (Byte): H'010C

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	· · · · · ·	—	—	YO													
Initial value:	—	· · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	YO	Undefined	R	

(7) User Clipping Area Registers (UCLR)

The User Clipping Area Registers (UCLR) are 32-bit read-only registers that indicate the user clipping area.

Bit 13 is the sign bit.

Register Address (Byte): H'0110
Upper-left X

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper-left X	—	· · · · · · ·	—	—	UXMIN													
Initial value:	—	· · · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	UXMIN	Undefined	R	

Register Address (Byte): H'0114
Upper-left Y

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Upper-left Y	—	· · · · · · ·	—	—	UYMIN													
Initial value:	—	· · · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	UYMIN	Undefined	R	

Register Address (Byte): H'0118
Lower-right X

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Lower-right X	—	· · · · · · ·	—	—	UXMAX													
Initial value:	—	· · · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	UXMAX	Undefined	R	

Register Address (Byte): H'011C
Lower-right Y

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Lower-right Y	—	· · · · · ·	—	—	UYMAX													
Initial value:	—	· · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	UYMAX	Undefined	R	

(8) System Clipping Area Registers (SCLR)

The System Clipping Area Registers (SCLR) are two 32-bit read-only registers that indicate the system clipping area.

Bit 13 is the sign bit.

Register Address (Byte): H'0120
Lower-right X

Bit:	31	· · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Lower-right X	—	· · · · · ·	—	—	SXMAX													
Initial value:	—	· · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	SXMAX	Undefined	R	

Register Address (Byte): H'0124
Lower-right Y

Bit:	31	· · · · · · ·	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Lower-right Y	—	· · · · · · ·	—	—	SYMAX													
Initial value:	—	· · · · · · ·	—	—	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W:	—	· · · · · · ·	—	—	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	—	—	Reserved
13 to 0	SYMAX	Undefined	R	

11.4.4 2DGE Registers

Legends for register description:

Initial value: Register value after hardware reset (rsth_ru) or software reset (SRES in SR)

Note: Reserved bits must be set to 0. A register read for reserved bit returns undefined value.

(1) Anti-alias Font Command Registers

Reserved Register

Register Byte Address: H'2100
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Reserved Register

Register Byte Address: H'2104
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Foreground Color Register For Anti-alias Font Command

Register Byte Address: H'2108
Initial value: H'XXXX–XXXX

Bit	Description
31 to 16	Reserved
15 to 0	Foreground Color This parameter defines 16-bpp foreground color value. (Blue: 4 to 0, Green: 10 to 5, Red: 15 to 11)

Destination Position Register For Anti-alias Font Command

Register Byte Address: H'210C
Initial value: H'XXXX–XXXX

Bit	Description
31 to 28	These bits must be set to the value of Bit 27 (2's complement).
27 to 16	Destination Y This parameter defines the Y start position of destination area. (–2048 <= Y <= 2047)
15 to 12	These bits must be set to the value of Bit 11 (2's complement).
11 to 0	Destination X This parameter defines the X start position of destination area. (–2048 <= X <= 2047)

a_value Source Address Register For Anti-alias Font Command

Register Byte Address: H'2110
Initial value: H'XXXX–XXXX

Bit	Description
31 to 23	Reserved
22 to 0	a_value Source Address This parameter defines the start address of the a_value source. a_value source is aligned to 64-bit boundary, so the lower 3 bits must be set to zero.

a_value Source Size Register For Anti-alias Font Command

Register Byte Address: H'2114
Initial value: H'XXXX–XXXX

Bit	Description
31 to 26	Reserved
25 to 16	Height This parameter defines a_value source height specified as the distance between lines (lines -1). (0 <= Height <= 1023)
15 to 10	Reserved
9 to 0	Width This parameter defines a_value source width specified as the distance between pixels (pixels -1). (0 <= Width <= 1023) A multiple of 8 pixels -1 must be set as the Width, so the lower 3 bits must be set to all 1.

Reserved Register

Register Byte Address: H'2118
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Command Register For Anti-alias Font Command

Register Byte Address: H'211C
Initial value: H'0200–0000

Note: Anti-alias Font Command is executed when the value is set to this command register.

Bit	Description
31	Reserved
30	User Clipping Enable Setting this bit enables User Clipping area. User Clipping area is defined as a rectangle. Destination (output) pixels outside the user clipping rectangle are not written to the destination. 0: User Clipping is disabled. 1: User Clipping is enabled.
29 to 26	Reserved
25	1 (Destination addressing is tile.)
24 to 12	Reserved
11	Destination Transparency polarity This bit defines polarity for destination transparency. 0: Destination data is not update (transparent) if this data is equal to Destination Transparent Color register value. 1: Destination data is not update (transparent) if this data is not equal to Destination Transparent Color register value.
10	Destination Transparency enable Setting this bit enables transparency depending on destination color data. 0: Destination transparency is disabled. 1: Destination transparency is enabled. Destination color is compared with Destination Transparent Color register and the transparency depends on bit 11.
9 to 0	Reserved

(2) BitBLT Command Registers

Reserved Register

Register Byte Address: H'2140
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Reserved Register

Register Byte Address: H'2144
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Reserved Register

Register Byte Address: H'2148
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Destination Position Register For BitBLT Command

Register Byte Address: H'214C
Initial value: H'XXXX–XXXX

Bit	Description
31 to 28	These bits must be set to the value of Bit 27 (2's complement).
27 to 16	Destination Y This parameter defines the Y start position of destination area. (–2048 <= Y <= 2047)
15 to 12	These bits must be set to the value of Bit 11 (2's complement).
11 to 0	Destination X This parameter defines the X start position of destination area. (–2048 <= X <= 2047)

Source Position Register For BitBLT Command

Register Byte Address: H'2150
Initial value: H'XXXX–XXXX

Bit	Description
31 to 26	Reserved
25 to 16	Source Y This parameter defines the Y start position of source area. (0 <= Y <= 1023)
15 to 10	Reserved
9 to 0	Source X This parameter defines the X start position of source area. (0 <= X <= 1023)

Note: BitBLT Direction Dependence of Source / Destination Position

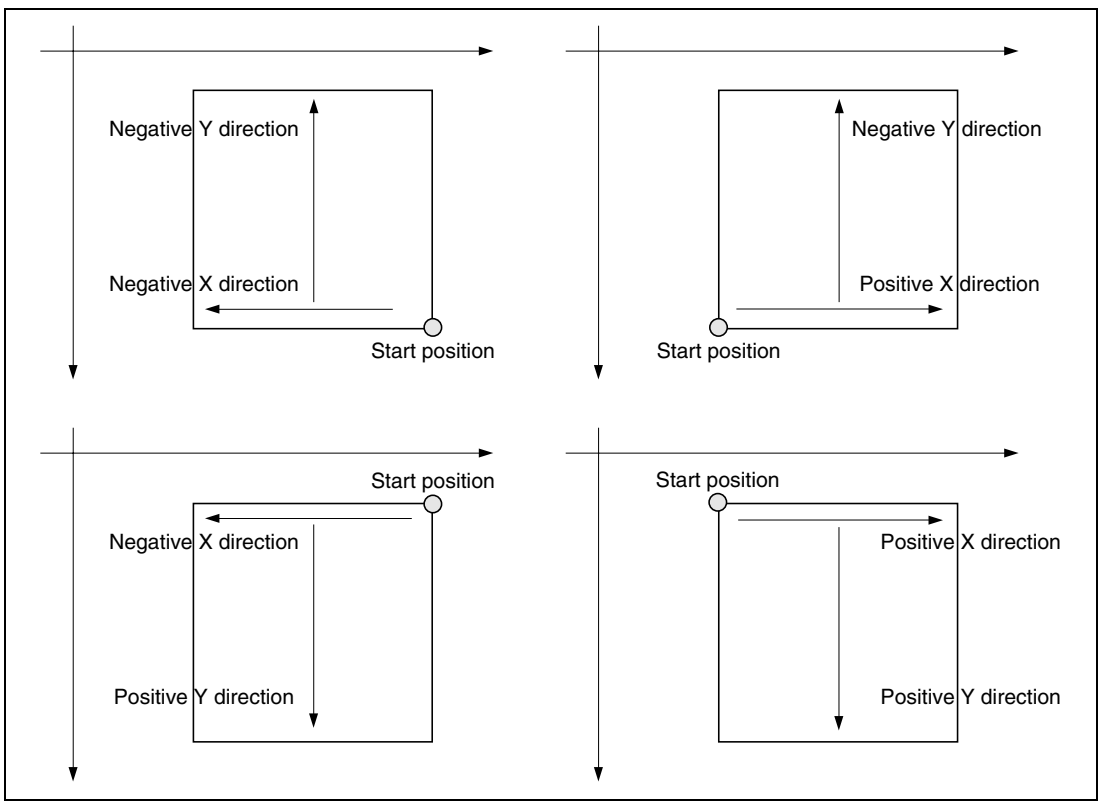


Figure 11.41 Source/Destination Position

Source Size Register For BitBLT Command

Register Byte Address: H'2154
Initial value: H'XXXX–XXXX

Bit	Description
31 to 26	Reserved
25 to 16	Height This parameter defines the source height for BitBLT command. The source height is specified as the distance between lines(lines -1). (0 <= Height <= 1023)
15 to 10	Reserved
9 to 0	Width This parameter defines the source width for BitBLT command. The source width is specified as the distance between pixels(pixels -1). (0 <= Width <= 1023)

Reserved Register

Register Byte Address: H'2158
Initial value: H'XXXX–XXXX

Bit	Description
31 to 0	Reserved

Command Register For BitBLT Command

Register Byte Address: H'215C
Initial value: H'0200–0000

Note: BitBLT Command is executed when the value is set to this command register.

Bit	Description
31	Reserved
30	User Clipping Enable Setting this bit enables User Clipping area. User Clipping area is defined as a rectangle. Destination (output) pixels outside the user clipping rectangle are not written to the destination. 0: User Clipping is disabled. 1: User Clipping is enabled.
29 to 26	Reserved
25	1 (Destination addressing is tile.)
24	Tile addressing source This bit defines addressing mode for source data. 0: Source addressing is linear. 1: Source addressing is tile.
23 to 16	Reserved
15	Y direction This bit defines the direction of transfer for the Y coordinate. 0: Positive Y direction (top-to-bottom drawing direction). Y coordinates for source and destination height counters and the address registers, get incremented after transfer of each line. 1: Negative Y direction (bottom-to-top drawing direction). Y coordinates for source and destination height counters and the address registers, get decrements after transfer of each line.
14	X direction This bit defines the direction of transfer for the X coordinate. 0: Positive X direction (left-to-right drawing direction). X coordinates for source and destination width counters and the address registers within a line, get incremented after transfer of each pixel. 1: Negative X direction (right-to-left drawing direction). X coordinates for source and destination width counters and the address registers within a line, get decrement after transfer of each pixel.
13 to 12	Reserved

Bit	Description
11	Destination Transparency polarity This bit defines polarity for destination transparency. 0: Destination data is not update (transparent) if this data is equal to Destination Transparent Color register value. 1: Destination data is not update (transparent) if this data is not equal to Destination Transparent Color register value.

		Destination Transparency Enable = 1		Destination Transparency Enable = 0	
		Destination Transparency Polarity = 0	Destination Transparency Polarity = 1	Destination Transparency Polarity = 0	Destination Transparency Polarity = 1
Source Transparency Enable = 1	Source Transparency Polarity = 0	A	B	E	
	Source Transparency Polarity = 1	C	D	F	
Source Transparency Enable = 0	Source Transparency Polarity = 0	G	H	I	
	Source Transparency Polarity = 1				

A: If (S == STC II D == DTC) output data (destination') are not written.

B: If (S == STC II D != DTC) output data (destination') are not written.

C: If (S != STC II D = DTC) output data (destination') are not written.

D: If (S != STC II D != DTC) output data (destination') are not written.

E: If (S == STC) output data (destination') are not written.

F: If (S != STC) output data (destination') are not written.

G: If (D == DTC) output data (destination') are not written.

H: If (D != DTC) output data (destination') are not written.

I : does nothing on color transparency

S: Source data, D: Destination data, STC: Source Transparent Color, DTC: Destination Transparent Color

STC and DTC can be set in Source Transparent Color Register and Destination Transparent Color respectively.

Bit	Description	
10	Destination Transparency enable	
	Setting this bit enables transparency depending on destination color data.	
	0: Destination transparency is disabled.	
	1: Destination transparency is enabled. Destination color is compared with Destination Transparent Color register and the transparency depends on bit 11.	
9	Source Transparency polarity	
	This bit defines polarity for source transparency.	
	0: Destination Data is not update (transparent) if source data is equal to Source Transparent Color register value.	
	1: Destination Data is not update (transparent) if source data is not equal to Source Transparent Color register value.	
8	Source Transparency enable	
	Setting this bit enables transparency depending on source color data.	
	0: Source transparency is disabled.	
	1: Source transparency is enabled. Source color is compared with Source Transparent Color register. and the transparency depends on bit 9.	
7 to 0	ROP code	
	2DGE supports 16 raster operations. (S: source data, D: destination data)	
	H'00	0
	H'11	~(S | D)
	H'22	~S & D
	H'33	~S
	H'44	S & ~D
	H'55	~D
	H'66	S ^ D
	H'77	~(S & D)
	H'88	S & D
	H'99	~(S ^ D)
	H'AA	D
	H'BB	~S | D
	H'CC	S
	H'DD	S | ~D
	H'EE	S | D
	H'FF	1

(3) Command Common Registers

User Clip Minimum Register

Register Byte Address: H'2900
Initial value: H'XXXX–XXXX

Bit	Description
31 to 26	Reserved
25 to 16	Minimum Y This parameter specifies the top edge of the user clipping rectangle. All destination (output) pixels with a Y coordinate less than this value are not written. (0 <= Minimum Y <= 1023, Minimum Y < Maximum Y)
15 to 10	Reserved
9 to 0	Minimum X This parameter specifies the left edge of the user clipping rectangle. All destination (output) pixels with a X coordinate less than this value are not written. (0 <= Minimum X <= 1023, Minimum X < Maximum X)

User Clip Maximum Register

Register Byte Address: H'2904
Initial value: H'XXXX–XXXX

Bit	Description
31 to 26	Reserved
25 to 16	Maximum Y This parameter specifies the bottom edge of the user clipping rectangle. All destination (output) pixels with a Y coordinate larger than this value are not written. (0 <= Maximum Y <= 1023, Minimum Y < Maximum Y)
15 to 10	Reserved
9 to 0	Maximum X This parameter specifies the right edge of the user clipping rectangle. All destination (output) pixels with a X coordinate larger than this value are not written. (0 <= Maximum X <= 1023, Minimum X < Maximum X)

System Clip Maximum Register

Register Byte Address: H'2908
Initial value: H'XXXX–XXXX

Top-left coordinates is (0, 0).

Bit	Description
31 to 26	Reserved
25 to 16	Maximum Y This parameter specifies the bottom edge of the system clipping rectangle. All destination (output) pixels with a Y coordinate larger than this value are not written. (0 <= Maximum Y <= 1023)
15 to 10	Reserved
9 to 0	Maximum X This parameter specifies the right edge of the system clipping rectangle. All destination (output) pixels with a X coordinate larger than this value are not written. (0 <= Maximum X <= 1023)

Destination Local Offset Register

Register Byte Address: H'290C
Initial value: H'XXXX–XXXX

Bit	Description
31 to 28	These bits must be set to the value of Bit 27 (2's complement).
27 to 16	Offset Y This parameter is used to add Y coordinate of rendering area. (–2048 <= offset Y <= 2047)
15 to 12	These bits must be set to the value of Bit 11 (2's complement).
11 to 0	Offset X This parameter is used to add X coordinate of rendering area. (–2048 <= offset X <= 2047)

Source/Destination Stride Register

Register Byte Address: H'2910
Initial value: H'XXXX–XXXX

Bit	Description
31, 30	Reserved
29	Color format This parameter defines the number of bits per pixel for destination data and for source data. 0: 8-bit/pixel 1: 16-bit/pixel
28 to 26	Reserved
25 to 16	Source Stride Tile Addressing Source (Bit 24 in Command Register For BitBLT Command) = 0: 0 <= Source Stride <= 1023 When a source pixel data format is 16-bit/pixel, a multiple of 4 pixels -1 must be set as the Source Stride. When a source pixel data format is 8-bit/pixel, a multiple of 8 pixels -1 must be set as the Source Stride. Tile Addressing Source (Bit 24 in Command Register For BitBLT Command)= 1: Source Stride is equal to Destination Stride (Bit 9 to 0).
15 to 10	Reserved
9 to 0	Destination Stride 01 1111 1111: 511 11 1111 1111: 1023

Source Transparent Color Register

Register Byte Address: H'2A00
Initial value: H'XXXX–XXXX

bit	Description
31 to 16	Reserved
15 to 0	Source Transparent Color This is either 8-bit/pixel or 16-bit/pixel color for source color transparency. When a source pixel data format is 8-bit/pixel, Source Transparent Color is set in the lower bits (Bit7 to Bit 0). The upper bits (Bit 15 to Bit 8) must be set to 0.

Destination Transparent Color Register

Register Byte Address: H'2A04
Initial value: H'XXXX–XXXX

Bit	Description
31 to 16	Reserved
15 to 0	Destination Transparent Color This is either 8-bit/pixel or 16-bit/pixel color for destination color transparency. When a destination pixel data format is 8-bit/pixel, Destination Transparent Color is set in the lower bits (Bit7 to Bit 0). The upper bits (Bit 15 to Bit 8) must be set to 0.

Source Base Address Register

Register Byte Address: H'2B00
Initial value: H'XXXX–XXXX

Bit	Description
31 to 23	Reserved
22 to 0	Base address This parameter defines the start address of the source data. The lowest bit must be set to zero if pixel data format is 16-bit/pixel. The lower 13 bits must be set to all zeros if the tile addressing source bit is equal to 1.

Destination Base Address Register

Register Byte Address: H'2B04
Initial value: H'XXXX–XXXX

Bit	Description
31 to 23	Reserved
22 to 16	Base address This parameter defines the start address of the destination data.
15 to 0	Reserved

Section 12 Color Space Converter

12.1 General Description

The Color Space Converter is used to convert YUV data into RGB format line by line.

This function is available only for DMA transfer.

12.2 Features

- Two modes: YUV mode and DELTA YUV mode.
- Primary and secondary DMA channels.

12.3 Block Diagram

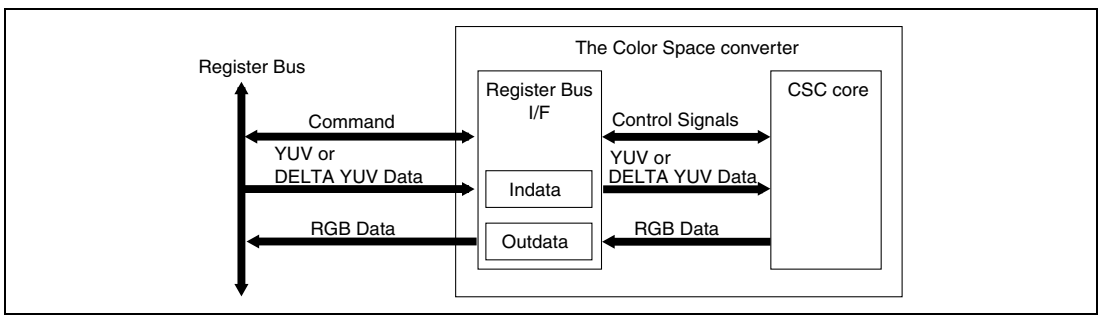


Figure 12.1 Block Diagram

12.4 Data formats

12.4.1 YUV data

YUV data uses a 4:2:2 format. The U and V data is horizontally reduced data.

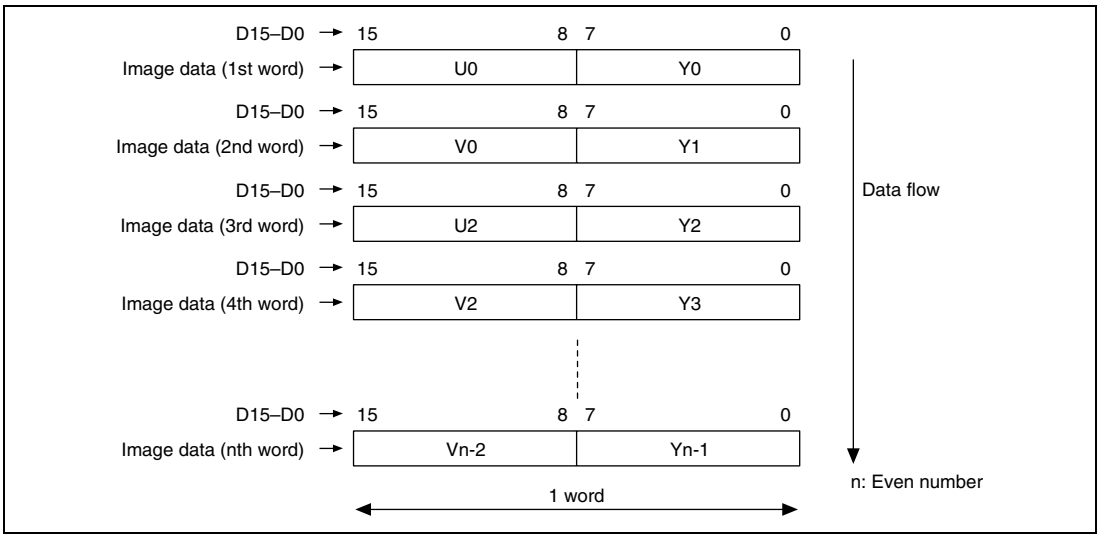


Figure 12.2 YUV Data Format

12.4.2 DELTA YUV data

DELTA YUV data uses a raster as the basic unit. The data configuration for one raster consists of the initial value in the first two words and compressed image data in the remaining words.

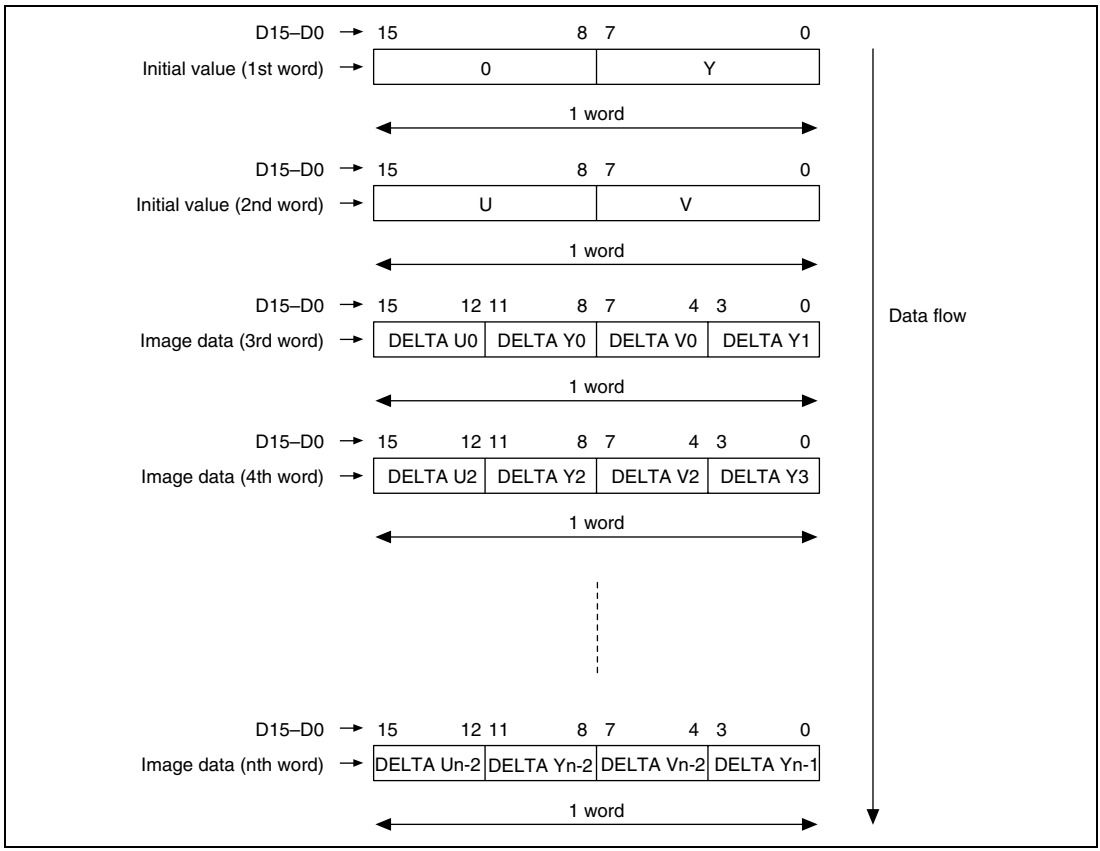


Figure 12.3 DELTA YUV Data Format

12.4.3 RGB data

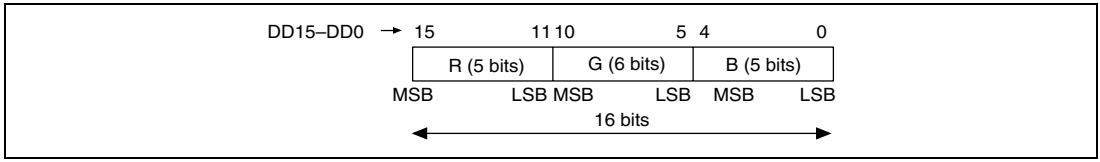


Figure 12.4 RGB Data Format

12.5 Register Description

There is a set of registers which is located in the address space of the PCI or MPX bus and located in the PCI memory window.

12.5.1 CSC Module Registers

Table 12.1 CSC module Register Map

Address (Bytes)	Register Name	Mnemonic or Symbol	R/W	Access Size
H'6920	Stadma	stadma	R/W	32
H'6924	Indata	indata	R/W	32
H'6928	Outdata	outdata	R	32
H'692C	Yuvmod	yuvmod	R/W	32
H'6930	Start_end	start_end	R/W	32
H'6934	Transcount	transcount	R/W	32
H'6938	Interrupt	interrupt	R/W	32

Legends for register description:

Initial value : Register value after reset
— : Undefined value
R/W : Read and Write, write value can be read.
R : Read only, for write always 0 write
R/WC0 : Read and Write, 0 write clear, 1 write is ignored
R/WC1 : Read and Write, 1 write clear, 0 write is ignored
W : Write only, Read prohibited. If reserved, write always 0.
—/W : Write only, Read value undefined.

12.5.2 Stadma Register

This register is read/write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															STOP	START
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	0	R	**Reserved**
1	STOP	0	R/W	**DMA stop (STOP)** In order to stop the DMA transfer of both channels, '1' must be written to the 'STOP' bit. For DMA transfer, the value of the `STOP` bit should always be '0'.
0	START	0	R/W	**DMA start (START)** The bit 'START' is DMA start. If this bit is set to '1' then the DMA request of the primary channel is asserted. After the first DMA acknowledge for the primary channel, this bit is set to '0'.

12.5.3 Indata Register

This register is read/write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PYIDT															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15 to 0	PYIDT	0	R/W	**YUV-DELTA YUV input data (PYIDT)** PYIDT are the input data for the Color Space Converter. PYIDT are arranged as follows

Mode	Initial data		Pixel data			
DELTA YUV	All0	Yabs [7:0]	DELTA Un [15:12]	DELTA Yn [11:8]	DELTA Vn [7:4]	DELTA Yn+1 [3:0]
	Uabs [15:8]	Vabs [7:0]				
YUV	—		Un[15:8]		Yn[7:0]	
			Vn[15:8]		Yn+1[7:0]	

In DELTA YUV mode, PYIDT must be written in the following order:

Initial data: 1. All0, Yabs[7:0]
2. Uabs[15:8], Vabs[7:0]

Pixel data: 3. DELTA Un[15:12], DELTA Yn[11:8], DELTA Vn[7:4], DELTA Yn+1[3:0]

Note: Initial data means the most left in a raster-scan picture data.

In YUV mode, PYIDT must be written in the following order:

Pixel data: 1. Un[15:8], Yn[7:0]
2. Vn[15:8], Yn+1[7:0]

12.5.4 Outdata Register

This register is read only.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																yulineend
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Rdata					Gdata						Bdata				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	yulineend	0	R	**Yuline end** Indicates the end of the conversion of a signal line.
15 to 11	Rdata	0	R	**RGB out data** the RGB data generated by the Color Space Converter.
10 to 5	Gdata	0	R	**RGB out data** the RGB data generated by the Color Space Converter.
4 to 0	Bdata	0	R	**RGB out data** the RGB data generated by the Color Space Converter.

12.5.5 Yuvmod Register

This register is read/write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																YUV MOD
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	0	R	**Reserved**
0	YUVMOD	0	R/W	**YUV MODE (YUVMOD)** Writing '1' to this register sets the Color Space Converter to YUV mode. Writing '0' sets the Color Space Converter to DELTA YUV mode.

12.5.6 Start End Register

This register is read/write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															Line start	Line end
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	0	R	**Reserved**
1	Line start	0	R/W	**Line start** Linestart must be set to '1' before the YUV-RGB conversion of each line, and it is reset after the first DMA acknowledge for the primary channel.
0	Line end	0	R/W	**Line end** Linend is set to '1' automatically before the last PYIDT input of a single line, but must be set to '0' at the beginning of the YUV-RGB conversion of each line.

12.5.7 Transcount Register

This register is read/write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					TRANSCOUNT											
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	0	R	**Reserved**
11 to 0	TRANSCOUNT	0	R/W	**Trans count** The value of TRANSCOUNT should be equal to the PYIDT count, which can be calculated as follows. Mode: PYIDT count DELTA YUV: ((Pixel count)/2) + 2 YUV: Pixel count The pixel count is the number of the converted RGB pixels in one line.

12.5.8 Interrupt Register

This register is read/write.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																interrupt
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	0	R	**Reserved**
0	interrupt	0	R/WC1	**Interrupt** When an interrupt source occurs, the Interrupt Register is set to 1 and an interrupt request is sent to the CPU. Clearing the interrupt is performed by writing 1 to this register. 0 writing is ignored.

The interrupt source of the Color Space Converter occurs after the data conversion of a whole line finishes and the Color Space Converter sends the last DMA request through the secondary channel for RGB data storage of that line.

12.6 Functional Description

The Color Space Converter converts YUV data into RGB format line by line. The conversion formula is:

$$R = (Y + (V - 128) \times 1.37)/8$$
$$G = (Y - 0.698 \times (V - 128) - 0.336 \times (U - 128))/4$$
$$B = (Y + (U - 128) \times 1.73)/8$$

The range of YUV data is 0-255, and the precision of each coefficient in the conversion formula is:

$$1.37 = 1.0101111$$
$$1.73 = 1.1011110$$
$$0.698 = 0.10110010$$
$$0.336 = 0.01010110$$

In DELTA YUV mode, the input data are decompressed firstly, then the YUV-RGB conversion is done according to the formula above. The decompression formula is:

$$Y0 = Yabs + Qf\text{-}1\ (deltaY0)$$
$$U0 = Uabs + Qf\text{-}1\ (deltaU0)$$
$$V0 = Vabs + Qf\text{-}1\ (deltaV0)$$

$$Yi = Yi\text{-}1 + Qf\text{-}1\ (deltaYi) \quad (i = 1, 2, 3 \ldots n+1)$$
$$Ui = Ui\text{-}2 + Qf\text{-}1\ (deltaUi) \quad (i = 2, 4, 6 \ldots n)$$
$$Vi = Vi\text{-}2 + Qf\text{-}1\ (deltaVi) \quad (i = 2, 4, 6 \ldots n)$$

At this point, U and V pixel data with even indices are decompressed. Next, pixel data with odd indices are calculated as follows:

$$Ui+1 = (Ui + Ui+2)/2$$
$$Vi+1 = (Vi + Vi+2)/2$$
$$Un+1 = Un$$
$$Vn+1 = Vn$$

Qf-1 is a parameter, which is coded into 8 bits out of the differences of each 4 bits of (DELTA Yi), (DELTA Ui), and (DELTA Vi). Table 12.2 lists these parameters.

Table 12.2 Coded Parameters

DELTA Y, DELTA U, DELTA V	Qf-1
0	0
1	1
2	4
3	9
4	16
5	27
6	44
7	79
8	128
9	177
10	212
11	229
12	240
13	247
14	252
15	255

In the next section, we describe how to use the Color Space Converter in both PCI and MPX modes.

12.6.1 Utilization Flow

The Color Space Converter has access to the memory through two DMA channels. The primary DMA channel is dedicated to reading YUV data from the main memory in the system or the graphics memory , and the secondary DMA channel is for writing RGB data in the graphic memory. In order to use the Color Space Converter, the user must set properly the registers in the DMAC and in the Color Space Converter .

Table 12.3 Setting Example of CSC Registers

	Value	
Name of register	**DELTA YUV mode**	**YUV mode**
Before the YUV-RGB Conversion of the First Line		
Yuvmod	32'H00000000	32'H00000001
Transcount	PYIDT count	PYIDT count
Start End	32'H00000002	32'H00000002
Stadma	32'H00000001	32'H00000001
Before the YUV-RGB conversion of the next line		
Start End	32'H00000002	32'H00000002
Stadma	32'H00000001	32'H00000001

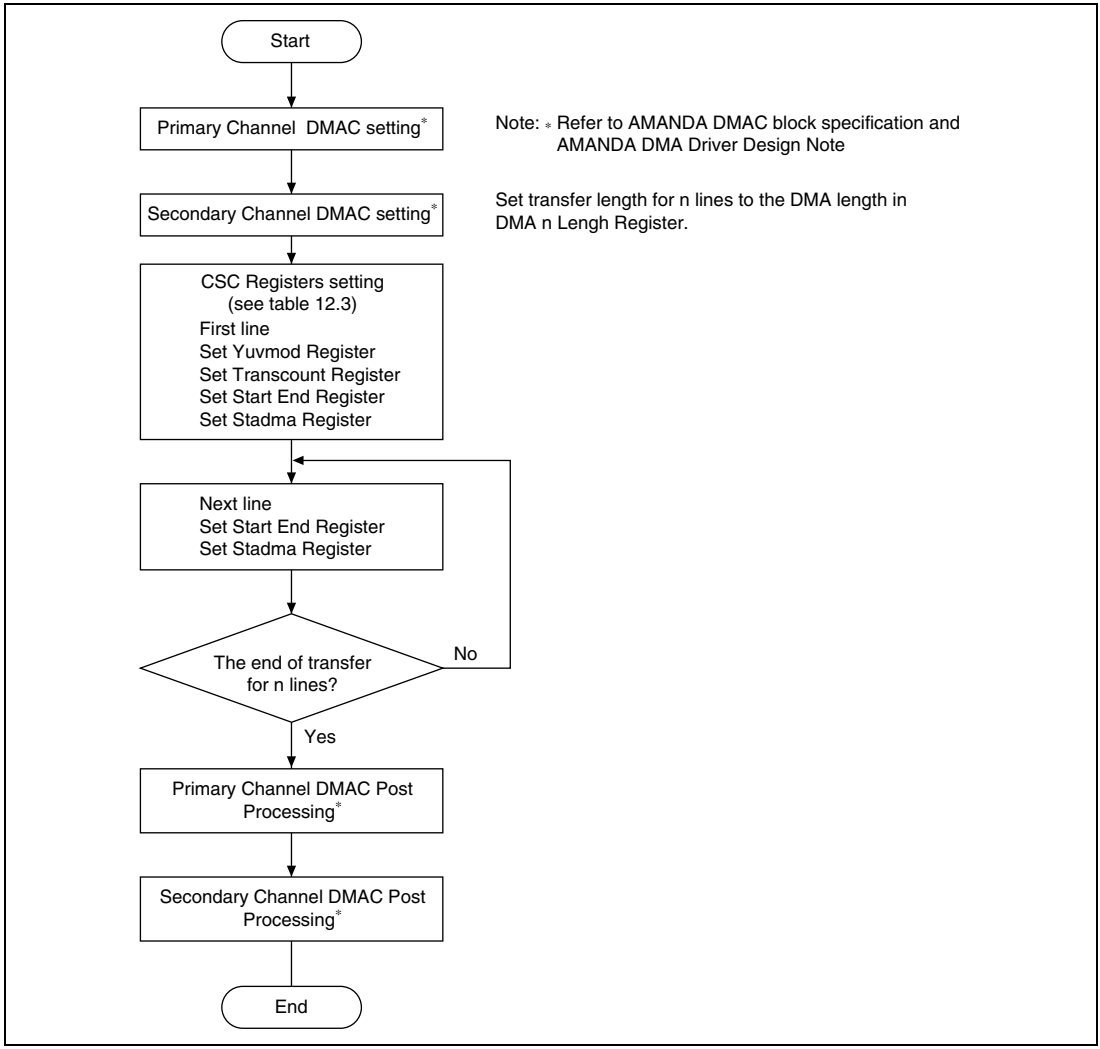
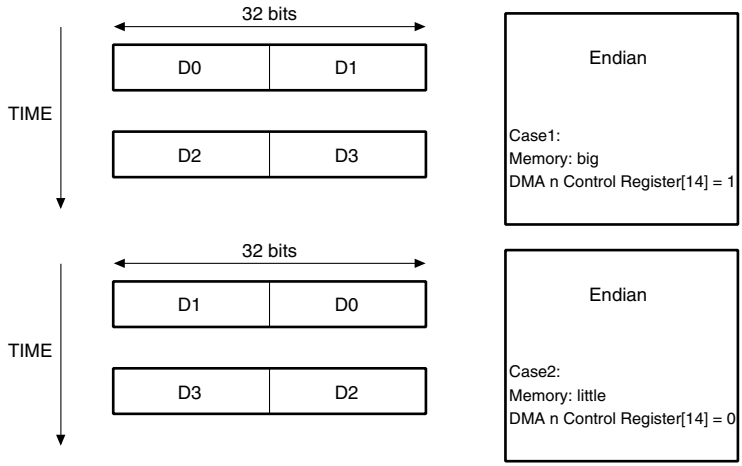


Figure 12.5 Utilization flow of the Color Space Converter in PCI Mode

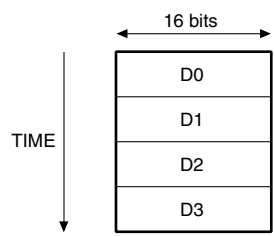
12.6.2 Endian setting

The endian of the data in the memory can be set by writing the appropriate data in the bits '15' and '14' of DMA n Control Register as follows:

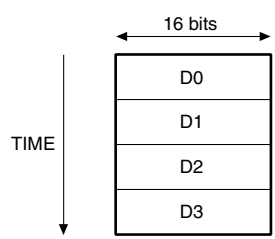
Source: Main Memory (YUV or DELTA YUV Data)



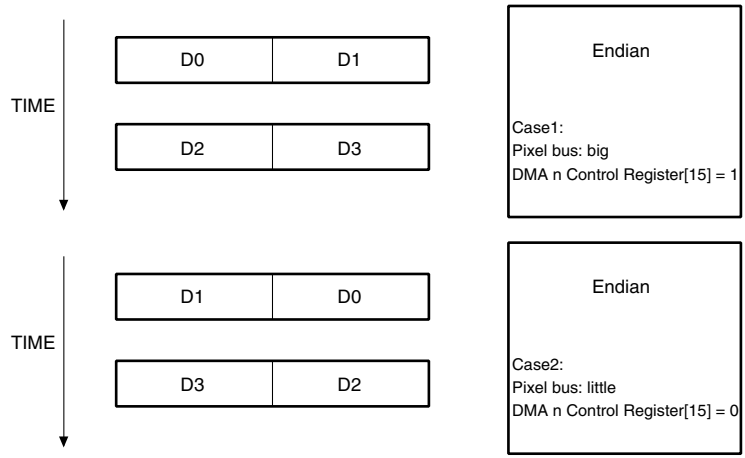
Destination: GM Pixel bus (RGB Data)



Source: CSC Outdata (RGB Data)



Destination: GM Pixel bus (RGB Data)



When the data transfer is from the Color Space Converter to the Graphics memory (GM), the endian setting of the GM should be done in such a way that the RGB data can be read by the Display Out module.

12.6.3 Module Standby Mode

The CSC module allows clock gating to reduce power consumption. Register bus clock can be gated. This module standby mode can be executed by controlling Clock Control 1 Register in Power Control module.

To wake up the module, bit 27 in Clock Control 1 Register must be enabled. After enabling this bit, all access to CSC module can be possible.

To power down the module, the following procedure is required.

1. Stop DMA transfer by enabling STOP bit in Stadma register of CSC module.
2. Disable bit 27 in Clock Control 1 Register.

This will cause CSC module of HD64404 to halt operation.

The register contents are retained.

Section 13 Audio Codec Interface

13.1 General Description

The Audio Codec digital controller interface supports bidirectional data transfer to Audio Codec (AC'97) Version 2.1. Serial data can be received from and transmitted to an appropriate audio codec. Multiple codec implementations are not supported.

The controller will extract or insert data from audio frames and present it as a set of memory mapped registers to the processor via the Register Bus interface. Certain data slots within both the receive and transmit frames will also have the option of DMA transfer.

13.2 Features

- Digital interface to a single AC'97 version 2.1 Audio Codec.
- PIO from status slots 1 and 2 of the Rx frame.
- PIO to command slots 1 and 2 of the Tx frame.
- PIO from data slots 3 and 4 of the Rx frame.
- PIO to data slots 3 and 4 of the Tx frame.
- Selectable 16-or-20 bit DMA from data slots 3 and 4 of the Rx frame.
- Selectable 16-or-20 bit DMA to data slots 3 and 4 of the Tx frame.
- Supports variable sample rates by qualifying slot data with Tag bits and responding to Rx frame slot request bits for the Tx frame.
- Interrupts can be generated for data ready/required and overrun/underrun.
- Supports cold and warm resets, and powerdown.

13.3 Block Diagram

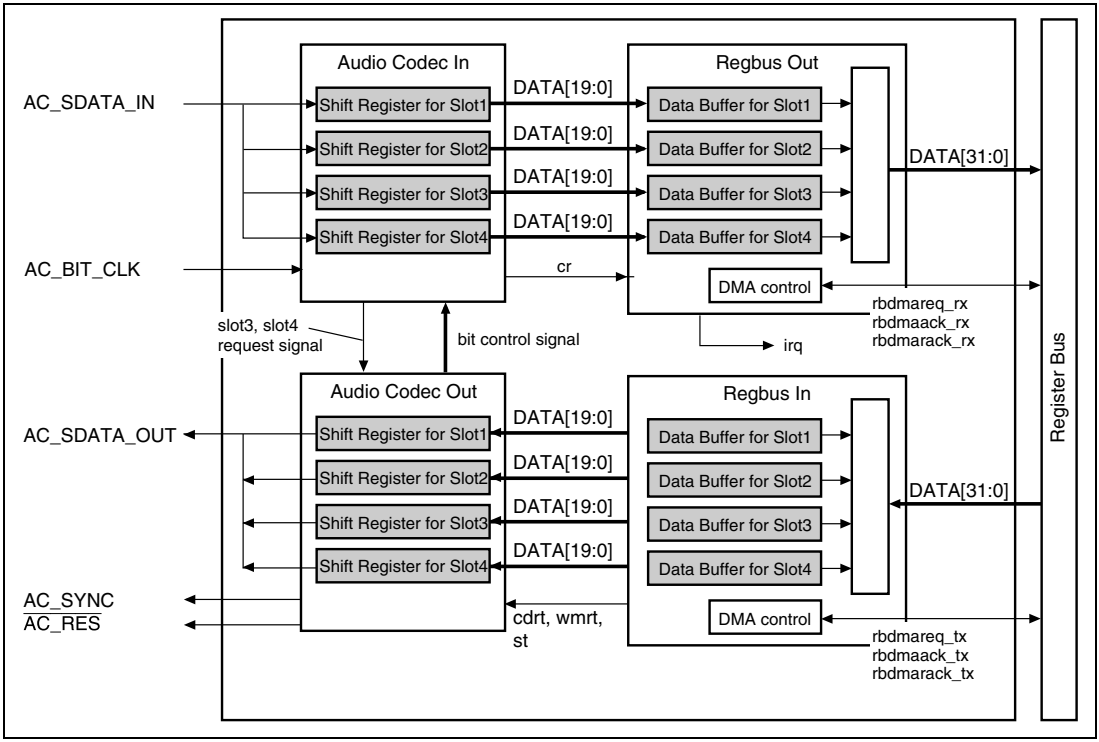


Figure 13.1 Block Diagram

13.4 Pin Description

Table 13.1 Pin configuration

Name	Width	Type	Description
AC_BIT_CLK	1	IN	Audio Codec serial data clock
AC_SDATA_IN	1	IN	Audio Codec serial data in for Rx frame
AC_SDATA_OUT	1	OUT	Audio Codec serial data out for Tx frame
AC_SYNC	1	OUT	Audio Codec frame sync
$\overline{\text{AC\_RES}}$	1	OUT	Audio Codec reset (active low)
rbdmareq_rx	1	OUT	DMA request for Rx data to be read
rbdmareq_tx	1	OUT	DMA request for Tx data to be written
rbdmaack_rx	1	IN	DMA acknowledge for Rx
rbdmaack_tx	1	IN	DMA acknowledge for Tx
rbdmarack_rx	1	IN	DMA return acknowledge for Rx
rbdmarack_tx	1	IN	DMA return acknowledge for Tx
irq	1	OUT	Interrupt request
Register Bus	—	—	System Bus

13.5 Register Description

The interface contains the registers shown in the table below.

Table 13.2 Audio Codec Register map

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6008	Control and Status Register	CR	32
H'6020	Command/Status Address Register	CSAR	32
H'6024	Command/Status Data Register	CSDR	32
H'6028	PCM Playback/Record Left channel	PCML	32
H'602C	PCM Playback/Record Right channel	PCMR	32
H'6050	TX Interrupt Enable Register	TIER	32
H'6054	TX Status Register	TSR	32
H'6058	RX Interrupt Enable Register	RIER	32
H'605C	RX Status Register	RSR	32
H'6060	Audio Codec Control Register	ACR	32
H'6070	TX DMA Register	TXDMA	32
H'6074	RX DMA Register	RXDMA	32

Legends for register description:

Initial Value : Register value after reset
— : Undefined value
R/W : Read and Write, write value can be read.
R : Read only, for write always 0 write
R/WC0 : Read and Write, 0 write clear, 1 write is ignored
R/WC1 : Read and Write, 1 write clear, 0 write is ignored
W : Write only, Read prohibited. If reserved, write always 0.
—/W : Write only, Read value undefined.

13.5.1 Control and Status Register (CR)

CR is a 32-bit Read/Write Register that is used to control the interface, and read status information from it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CR				CDRT	WMRT					ST					
Initial:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	R*	R	R	R	W	R	R	R	R	R

Note: * Read only, for write always 1 write

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	0	R	**Reserved**
15	CR	0	R	**Codec Ready (CR)** 1: The codec connected to the Audio Codec interface is ready. 0: The codec connected to the Audio Codec interface is not ready
14 to 12	—	0	R	**Reserved**
11	CDRT	0	W	**Cold Reset for Audio Codec (CDRT)** Writing 1 will cause a Cold Audio Codec Reset. The bit must be written to 0 again before another 1 is written to cause another Cold Reset. This bit is always read as 0. A Cold Reset should only be performed after power-up, and to wake up the interface after previously issuing a power-down command.

Bit	Bit Name	Initial Value	R/W	Description
10	WMRT	0	W	**Warm Reset for Audio Codec (WMRT)** Writing 1 will cause a Warm Audio Codec Reset. The bit must be written to 0 again before another 1 is written to cause another Warm Reset. This bit is always read as 0. A Warm Reset should only be performed after power-up, and to wake up the interface after previously issuing a power-down command.
9	—	1	R*	**Reserved**
8 to 6	—	0	R	**Reserved**
5	ST	0	W	**Start Transfer (ST)** Writing 1 will start transmitting and receiving data. Writing 0 will stop transmission and reception at the end of a frame, although this method should not be used to stop transmission, during normal operation. This bit is always read as 0.
4 to 0	—	0	R	**Reserved**

Note: * Read only, for write always 1 write.

The codec can be put into power-down mode by writing bit 12 of its register index 26. The codec will respond by removing AC_BIT_CLK, and normal operation of the interface is suspended. This is also the case at power-up. A Cold or Warm reset must be performed to resume normal operation.

13.5.2 Command/Status Address Register (CSAR)

The purpose of the CSAR is to access the address of the register set of the attached codec. Writing to the CSAR will take place when the system needs to write to or request a read from a codec register, and the Command Address will be transmitted to the codec in slot 1. Reading from the CSAR will take place when the codec has responded to a read, and its Status Address will be received in slot 1.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													RW	CA6/ SA6	CA5/ SA5	CA4/ SA4
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CA3/ SA3	CA2/ SA2	CA1/ SA1	CA0/ SA0	SLR EQ3	SLR EQ4	SLR EQ5	SLR EQ6	SLR EQ7	SLR EQ8	SLR EQ9	SLRE Q10	SLRE Q11	SLRE Q12		
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	0	R	**Reserved**
19	RW	0	R/W	**Codec Read/Write Command (RW)** 1: Read. Instructs the codec that the register whose index is in the address field should be read. 0: Write. Instructs the codec that the register whose index is in the address field should be written. The CSDR must have previously been loaded with the write data, because the CSAR and CSDR will then be sent as a pair in the same Tx frame (assuming TX12_ATOMIC is 1 in ACR).
18	CA6/SA6	0	R/W	**Codec Control Register Address 6:0(CA6 to CA0)/Codec Status Register Address 6:0 (SA6 to SA0)** When this register is written, these bits are the index of the Codec Register to be accessed When this register is read, these bits are the index of the codec register for which data is being returned in the CSDR.
17	CA5/SA5	0	R/W	
16	CA4/SA4	0	R/W	
15	CA3/SA3	0	R/W	
14	CA2/SA2	0	R/W	
13	CA1/SA1	0	R/W	
12	CA0/SA0	0	R/W	

Bit	Bit Name	Initial Value	R/W	Description
11	SLREQ3	0	R	**Slot Request 3:12 (SLREQ3 to SLREQ12)**
10	SLREQ4	0	R	These bits are valid in the Rx frame only and indicate that slot data is required by the codec in the next Tx frame. These flags serve no useful purpose as register bits, because they are handled automatically by the hardware. They are included because they are part of slot 1 of the Rx frame. 0: Slot data is required. 1: Slot data is not required.
9	SLREQ5	0	R	
8	SLREQ6	0	R	
7	SLREQ7	0	R	
6	SLREQ8	0	R	
5	SLREQ9	0	R	
4	SLREQ10	0	R	
3	SLREQ11	0	R	
2	SLREQ12	0	R	
1 to 0	—	0	R	**Reserved**

13.5.3 Command/Status Data Register (CSDR)

The purpose of the CSDR is to access the data of the register set of the attached codec. Writing to the CSDR will take place when the system needs to write to a Codec Register, and the Command Data will be transmitted to the codec in slot 2. Reading from the CSDR will take place when the codec has responded to a read, and its Status Data will be received in slot 2. In both cases, the address of the codec register the data corresponds to will be in the CSAR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													CD15 /SD15	CD14 /SD14	CD13 /SD13	CD12 /SD12
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CD11 /SD11	CD10 /SD10	CD9/ SD9	CD8/ SD8	CD7/ SD7	CD6/ SD6	CD5/ SD5	CD4/ SD4	CD3/ SD3	CD2/ SD2	CD1/ SD1	CD0/ SD0				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	0	R	**Reserved**
19	CD15/SD15	0	R/W	**Command Data 15:0 (CD15 to C0)/ Status Data 15:0 (SD15 to SD0)**
18	CD14/SD14	0	R/W	When this register is written, the data will be sent to the codec when its register address is loaded into the CSAR.
17	CD13/SD13	0	R/W	When this register is read, the data will be the contents of the codec register whose address is in the CSAR.
16	CD12/SD12	0	R/W	
15	CD11/SD11	0	R/W	
14	CD10/SD10	0	R/W	
13	CD9/SD9	0	R/W	
12	CD8/SD8	0	R/W	
11	CD7/SD7	0	R/W	
10	CD6/SD6	0	R/W	
9	CD5/SD5	0	R/W	
8	CD4/SD4	0	R/W	
7	CD3/SD3	0	R/W	
6	CD2/SD2	0	R/W	
5	CD1/SD1	0	R/W	
4	CD0/SD0	0	R/W	
3 to 0	—	0	R	**Reserved**

13.5.4 PCM Playback/Record Left Channel (PCML)

The purpose of PCML is to access the left channel digital audio record and playback streams of the codec. When PCML is written to, PCM Playback Left Channel data will be transmitted to the codec. When PCML is read from, PCM Record Left Channel data will be received from the codec. The data is left justified to accommodate codecs whose DACs and ADCs are less than 20 bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													D19	D18	D17	D16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	0	R	**Reserved**
19	D19	0	R/W	**Data 19:0 (D19 to D0)**
18	D18	0	R/W	When this register is written, PCM Playback Left Channel Data will be transmitted to the connected CODEC when requested.
17	D17	0	R/W	
16	D16	0	R/W	When this register is read, PCM Record Left Channel data received from the connected CODEC is stored.
15	D15	0	R/W	
14	D14	0	R/W	
13	D13	0	R/W	
12	D12	0	R/W	
11	D11	0	R/W	
10	D10	0	R/W	
9	D9	0	R/W	
8	D8	0	R/W	
7	D7	0	R/W	
6	D6	0	R/W	
5	D5	0	R/W	
4	D4	0	R/W	
3	D3	0	R/W	
2	D2	0	R/W	
1	D1	0	R/W	
0	D0	0	R/W	

In 16-bit DMA mode, the register has the following definition.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LD15	LD14	LD13	LD12	LD11	LD10	LD9	LD8	LD7	LD6	LD5	LD4	LD3	LD2	LD1	LD0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	LD15	0	R/W	**Left Data 15:0 (LD15 to LD0)**
30	LD14	0	R/W	When these bits are written, PCM Playback Left Channel Data will be transmitted to the connected CODEC when requested.
29	LD13	0	R/W	
28	LD12	0	R/W	When these bits are read, PCM Record Left Channel data received from the connected CODEC is stored.
27	LD11	0	R/W	
26	LD10	0	R/W	
25	LD9	0	R/W	
24	LD8	0	R/W	
23	LD7	0	R/W	
22	LD6	0	R/W	
21	LD5	0	R/W	
20	LD4	0	R/W	
19	LD3	0	R/W	
18	LD2	0	R/W	
17	LD1	0	R/W	
16	LD0	0	R/W	
15	RD15	0	R/W	**Right Data 15:0 (RD15 to RD0)**
14	RD14	0	R/W	When these bits are written, PCM Playback Right Channel Data will be transmitted to the connected CODEC when requested.
13	RD13	0	R/W	
12	RD12	0	R/W	When these bits are read, PCM Record Right Channel data received from the connected CODEC is stored.
11	RD11	0	R/W	
10	RD10	0	R/W	
9	RD9	0	R/W	
8	RD8	0	R/W	
7	RD7	0	R/W	
6	RD6	0	R/W	
5	RD5	0	R/W	
4	RD4	0	R/W	
3	RD3	0	R/W	
2	RD2	0	R/W	
1	RD1	0	R/W	
0	RD0	0	R/W	

13.5.5 PCM Playback/Record Right Channel (PCMR)

The purpose of PCMR is to access the right channel digital audio record and playback streams of the codec. When PCMR is written to, PCM Playback Right Channel data will be transmitted to the codec. When PCMR is read from, PCM Record Right Channel data will be received from the codec. The data is left justified to accommodate codecs whose DACs and ADCs are less than 20 bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													D19	D18	D17	D16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	0	R	**Reserved**
19	D19	0	R/W	**Data 19:0 (D19 to D0)**
18	D18	0	R/W	When this register is written, PCM Playback Right Channel Data will be transmitted to the connected CODEC when requested. When this register is read, PCM Record Right Channel data received from the connected CODEC is stored.
17	D17	0	R/W	
16	D16	0	R/W	
15	D15	0	R/W	
14	D14	0	R/W	
13	D13	0	R/W	
12	D12	0	R/W	
11	D11	0	R/W	
10	D10	0	R/W	
9	D9	0	R/W	
8	D8	0	R/W	
7	D7	0	R/W	
6	D6	0	R/W	
5	D5	0	R/W	
4	D4	0	R/W	
3	D3	0	R/W	
2	D2	0	R/W	
1	D1	0	R/W	
0	D0	0	R/W	

13.5.6 Transmit Interrupt Enable Register (TIER)

TIER, a 32-bit Read/Write register, is used to enable or disable Audio Codec TX Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			PLTFRQIE	PRTFRQIE												
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R	R	R	R/	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PLTFUNIE	PRTFUNIE								
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	0	R	**Reserved**
29	PLTFRQIE	0	R/W	**PCML TX REQUEST Interrupt Enable (PLTFRQIE)** 1: PCML TX request interrupt is enabled. 0: PCML TX request interrupt is disabled.
28	PRTFRQIE	0	R/W	**PCMR TX REQUEST Interrupt Enable (PRTFRQIE)** 1: PCMR TX request interrupt is enabled. 0: PCMR TX request interrupt is disabled.
27 to 10	—	0	R	**Reserved**
9	PLTFUNIE	0	R/W	**PCML TX Underrun Interrupt Enable (PLTFUNIE)** 1: PCML TX underrun interrupt is enabled. 0: PCML TX underrun interrupt is disabled.
8	PRTFUNIE	0	R/W	**PCMR TX Underrun Interrupt Enable (PRTFUNIE)** 1: PCMR TX underrun interrupt is enabled. 0: PCMR TX underrun interrupt is disabled.
7 to 0	—	0	R	**Reserved**

13.5.7 TX Status Register (TSR)

TSR, a 32-bit Read Only register, is used to reflect the status of the Audio Codec TX controller. Each status bit can be cleared by writing 0 to it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD AMT	CMD DMT	PLT FRQ	PRT FRQ												
Initial:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC0	R / WC0	R / WC0	R / WC0	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							PLT FUN	PRT FUN								
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R / WC0	R / WC0	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	CMDAMT	1	R/WC0	**Command Address Empty (CMDAMT)** 1: CSAR Tx buffer is empty and can be filled by the system. (*)
30	CMDDMT	1	R/WC0	**Command Data Empty (CMDDMT)** 1: CSDR Tx buffer is empty and can be filled by the system. (*)
29	PLTFRQ	1	R/WC0	**PCML TX Request (PLTFRQ)** 1: PCML TX buffer is empty and must be filled by the system. This bit is cleared automatically in DMA mode, when PCML is written.
28	PRTFRQ	1	R/WC0	**PCMR TX Request (PRTFRQ)** 1: PCMR TX buffer is empty and must be filled by the system. This bit is cleared automatically in DMA mode, when PCMR is written.
27 to 10	—	0	R	**Reserved**
9	PLTFUN	0	R/WC0	**PCML TX Underrun (PLTFUN)** 1: PCML TX underrun has occurred. This will happen when the codec requests data for slot 3, but no new data is written to PCML.
8	PRTFUN	0	R/WC0	**PCMR TX Underrun (PRTFUN)** 1: PCMR TX underrun has occurred. This will happen when the codec requests data for slot 4, but no new data is written to PCMR.
7 to 0	—	0	R	**Reserved**

CMDAMT and CMDDMT have no associated interrupts. These bits should be polled, and read as 1, before new command data is written to CSAR or CSDR. In case of bit 19 of CSAR is a write and TX12_ATOMIC = 1, the following procedure must be taken.

1) Before the first access to Audio Codec registers after initialization, CMDDMT and CMDAMT must be cleared. 2) After CSDR and CSAR setting, poll CMDDMT and CMDAMT until they become 1 and clear them. 3) Then next register write operation can be started.

13.5.8 Receive Interrupt Enable Register (RIER)

RIER, a 32-bit Read/Write register, is used to enable or disable Audio Codec RX Interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PLRFOVIE	PRRFOVIE												
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	0	R	**Reserved**
22	STARYIE	0	R/W	**Status Address Ready Interrupt Enable (STARYIE)** 1: Status Address Ready Interrupt is enabled. 0: Status Address Ready Interrupt is disabled.
21	STDRYIE	0	R/W	**Status Data Ready Interrupt Enable(STDRYIE)** 1: Status Data Ready Interrupt is enabled. 0: Status Data Ready Interrupt is disabled.
20	PLRFRQIE	0	R/W	**PCML RX Request Interrupt Enable (PLRFRQIE)** 1: PCML RX Request Interrupt is enabled. 0: PCML RX Request Interrupt is disabled.
19	PRRFRQIE	0	R/W	**PCMR RX Request Interrupt Enable (PRRFRQIE)** 1: PCMR RX Request Interrupt is enabled. 0: PCMR RX Request Interrupt is disabled.
18 to 14	—	0	R	**Reserved**
13	PLRFOVIE	0	R/W	**PCML RX Overrun Interrupt Enable (PLRFOVIE)** 1: PCML RX Overrun Interrupt is enabled. 0: PCML RX Overrun Interrupt is disabled.
12	PRRFOVIE	0	R/W	**PCMR RX Overrun Interrupt Enable (PRRFOVIE)** 1: PCMR RX Overrun Interrupt is enabled. 0: PCMR RX Overrun Interrupt is disabled.
11 to 0	—	0	R	**Reserved**

13.5.9 RX Status Register (RSR)

RSR, a 32-bit Read Only register, is used to reflect the status of the Audio Codec RX controller. Each status bit can be cleared by writing 0 to it.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
										STARY	STDRY	PLRFRQ	PRRFRQ			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R / WC0	R / WC0	R / WC0	R / WC0	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PLRFOV	PRRFOV												
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R / WC0	R / WC0	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	0	R	**Reserved**
22	STARY	0	R/WC0	**Status Address Ready (STARY)** 1: Status address is ready.
21	STDRY	0	R/WC0	**Status Data Ready (STDRY)** 1: Status data is ready.
20	PLRFRQ	0	R/WC0	**PCML RX Request (PLRFRQ)** 1: PCML RX data is ready and must be read. This bit is cleared automatically in DMA mode, when PCML is read.
19	PRRFRQ	0	R/WC0	**PCMR RX Request (PRRFRQ)** 1: PCMR RX data is ready and must be read. This bit is cleared automatically in DMA mode, when PCMR is read.
18 to 14	—	0	R	**Reserved**

Bit	Bit Name	Initial Value	R/W	Description
13	PLRFOV	0	R/WC0	**PCML RX Overrun (PLRFOV)** 1: PCML RX data overrun has occurred. This will happen when new data is received from slot 3 before PCML has been read with the previous data.
12	PRRFOV	0	R/WC0	**PCMR RX Overrun (PRRFOV)** 1: PCMR RX data overrun has occurred. This will happen when new data is received from slot 4 before PCMR has been read with the previous data.
11 to 0	—	0	R	**Reserved**

13.5.10 Audio Codec Control Register (ACR)

ACR, a 32-bit Read/Write register, is used to control the Audio Codec interface.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		DMARX16	DMATX16			TX12_ATOMIC		RXDMAL_EN	TXDMAL_EN	RXDMAR_EN	TXDMAR_EN					
Initial:	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R*	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * R: Read only, for write always 1 write

Bit	Bit Name	Initial Value	R/W	Description
31	—	1	R*	**Reserved**
30	DMARX16	0	R/W	**16-bit RX DMA enable (DMARX16)** 1: 16-bit packed RX DMA mode is enabled. When set, this bit will override 20-bit DMA mode settings in bits 22 and 24. 0: 16-bit packed RX DMA mode is disabled. When clear, this bit will allow 20 bit DMA mode settings in bits 22 and 24 to take affect.
29	DMATX16	0	R/W	**16-bit TX DMA enable (DMATX16)** 1: 16-bit packed TX DMA mode is enabled. When set, this bit will override 20-bit DMA mode settings in bits 21 and 23. 0: 16-bit packed TX DMA mode is disabled. When clear, this bit will allow 20-bit DMA mode settings in bits 21 and 23 to take affect.
28, 27	—	0	R	**Reserved**
26	TX12_ATOMIC	1	R/W	**TX slots 1 and 2 atomic control (TX12_ATOMIC)** 1: TX data in CSAR and CSDR will be sent in the same frame if bit 19 of CSAR indicates a write. (CSAR must be written last) 0: TX data in CSAR and CSDR can be sent independently. It is not expected that this setting would be used in normal operation.
25	—	0	R	**Reserved**
24	RXDMAL_EN	0	R/W	**RX DMA Left Enable (RXDMAL_EN)** 1: 20-bit RX DMA is enabled for PCML. 0: 20-bit RX DMA is disabled for PCML.
23	TXDMAL_EN	0	R/W	**TX DMA Left Enable (TXDMAL_EN)** 1: 20-bit TX DMA is enabled for PCML. 0: 20-bit TX DMA is disabled for PCML.
22	RXDMAR_EN	0	R/W	**RX DMA Right Enable (RXDMAR_EN)** 1: 20-bit RX DMA is enabled for PCMR. 0: 20-bit RX DMA is disabled for PCMR.

Bit	Bit Name	Initial Value	R/W	Description
21	TXDMAR_EN	0	R/W	**TX DMA Right Enable (TXDMAR_EN)** 1: 20-bit TX DMA is enabled for PCMR. 0: 20-bit TX DMA is disabled for PCMR.
20 to 0	—	0	R	**Reserved**

Note: R: Read only, for write always 1 write

13.5.11 TX DMA Register (TXDMA)

In DMA operation, the address of this register will be written to the DMA 19 Request Address Register as the destination of DMA data, to be sent to the transmitter. The register itself does not physically exist in the module, and data is actually written to either PCML or PCMR.

13.5.12 RX DMA Register (RXDMA)

In DMA operation, the address of this register will be written to the DMA 19 Request Address Register as the source of DMA data, to be got from the receiver. The register itself does not physically exist in the module, and data is actually read from either PCML or PCMR.

13.6 Functional Description

13.6.1 Receiver

Serial audio data will be input to the module on the AC_SDATA_IN signal, referenced to AC_BIT_CLK. The tag bits are extracted from slot 0 and used to validate the data in the other slots. New data will only be loaded from a slot of a new frame if its tag bit is set.

Support will be for data in slots 1 to 4, tag bits and data corresponding to other slots will be ignored. Valid slot data will be loaded into a shift register and latched into a location accessible from the Register Bus. Each supported data slot will be readable under PIO as a 20 bit quantity, within a 32 bit register. Appropriate status bits will also be generated.

Note: When RX overrun occurs, the current data in RX buffer data of Audio Codec is overwritten by the next incoming data from Audio Codec interface.

13.6.2 Transmitter

Serial audio data will be output from the module on the SDATA_OUT signal, referenced to BIT_CLK. The tag bits of slot 0 are set to indicate which of the slots within the current frame contain valid data. A data slot is loaded into the current TX frame in response to the corresponding SLOTREQ bit from the previous RX frame.

Support will be for data in slots 1 to 4. Data will be latched from a location accessible from the Register Bus into a frame slot. Each supported data slot will be writeable under PIO as a 20 bit quantity, within a 32 bit register. Appropriate status bits will also be generated.

Note: When TX underrun occurs, the current data in TX buffer data of Audio Codec is transmitted until the next data is filled.

13.6.3 DMA

DMA transfer will be supported for the data in slots 3 and 4 of both the RX and TX frame. Bits 29 and 30 of register ACR will determine if the slot data size for DMA operations is 16 or 20 bit.

If the data size is 20 bits, two Register Bus cycles must be performed to transfer both data slots. There is only one DMA request each for the Receiver and Transmitter, and so DMA cycles in stereo mode will therefore be alternate to/from slots 3 and 4. In mono mode, DMA will occur for just one slot.

If the data size is 16 bits, data from slots 3 and 4 will be packed into a single 32 bit quantity (left and right data in PCML), which will take only one Register Bus cycle.

It may be necessary, for some system applications, to halt DMA activity before the terminal count has been reached. This can be done by simply clearing the appropriate DMA enable bits in the ACR. They can be re-enabled, for another transfer, after the DMAC has been re-programmed.

13.6.4 Interrupts

A single interrupt will be provided to flag events from both the Receiver and Transmitter. The sources of interrupt for each can be set in the corresponding Interrupt Enable Register. These will include requests to the processor to read/write slot data and to indicate overrun and underrun conditions. The cause of the interrupt will be determined by reading the appropriate status register. Writing zero to clear a set bit will also clear down the associated interrupt.

13.6.5 Initialized sequence

Figure 13.2 shows the example for initialized sequence

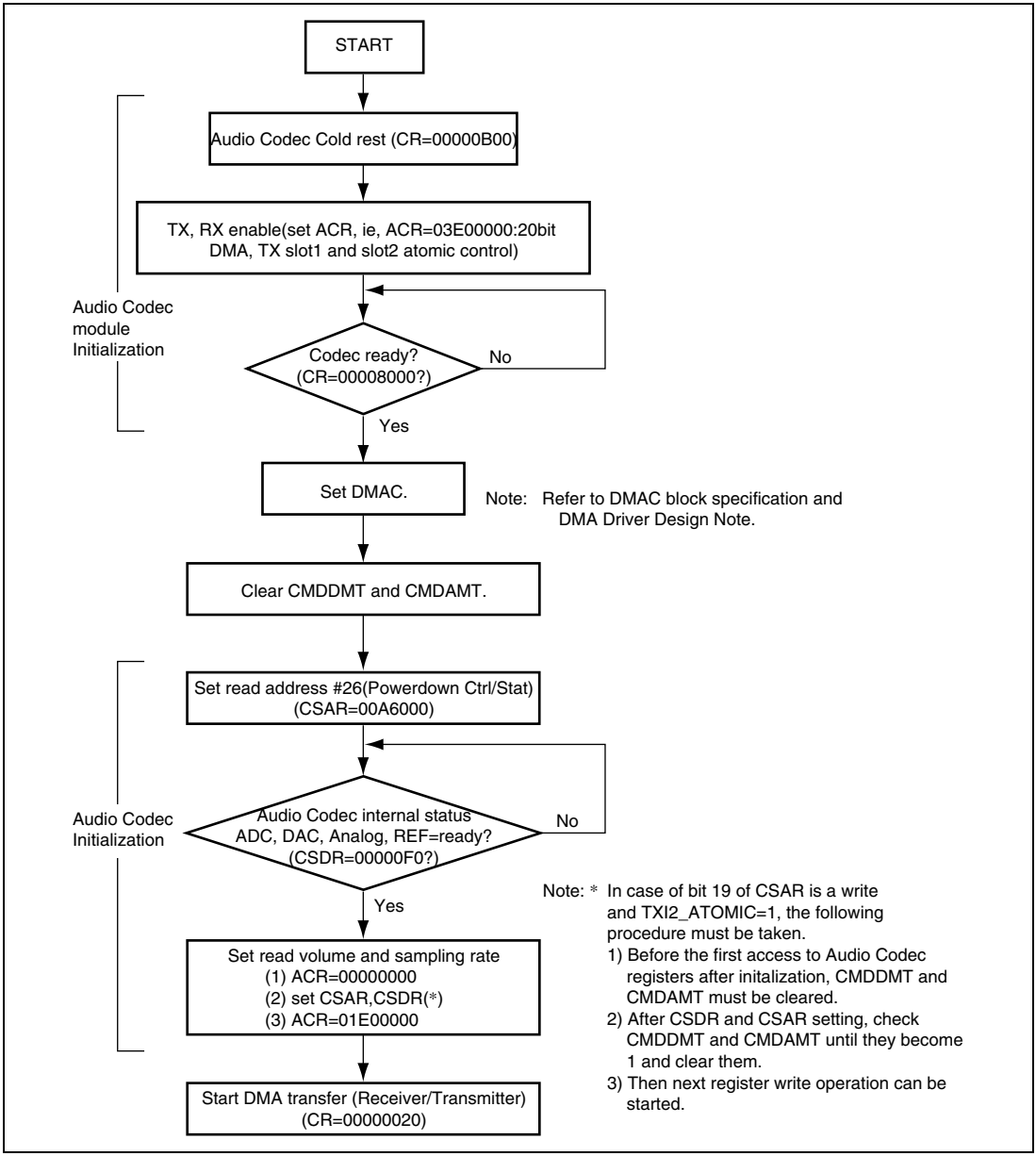


Figure 13.2 Initialized Sequence

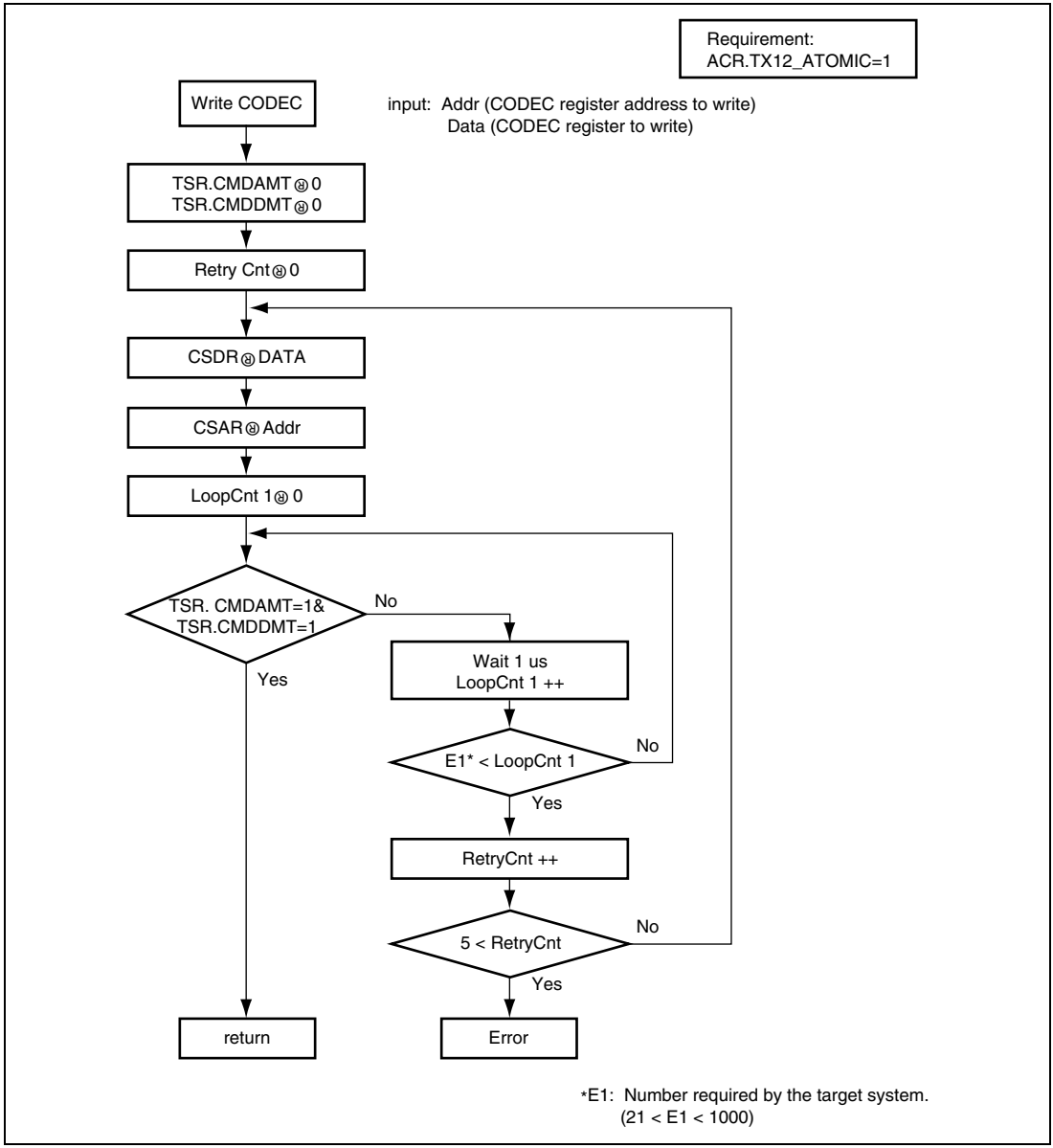


Figure 13.3 Access Flow Chart (1)

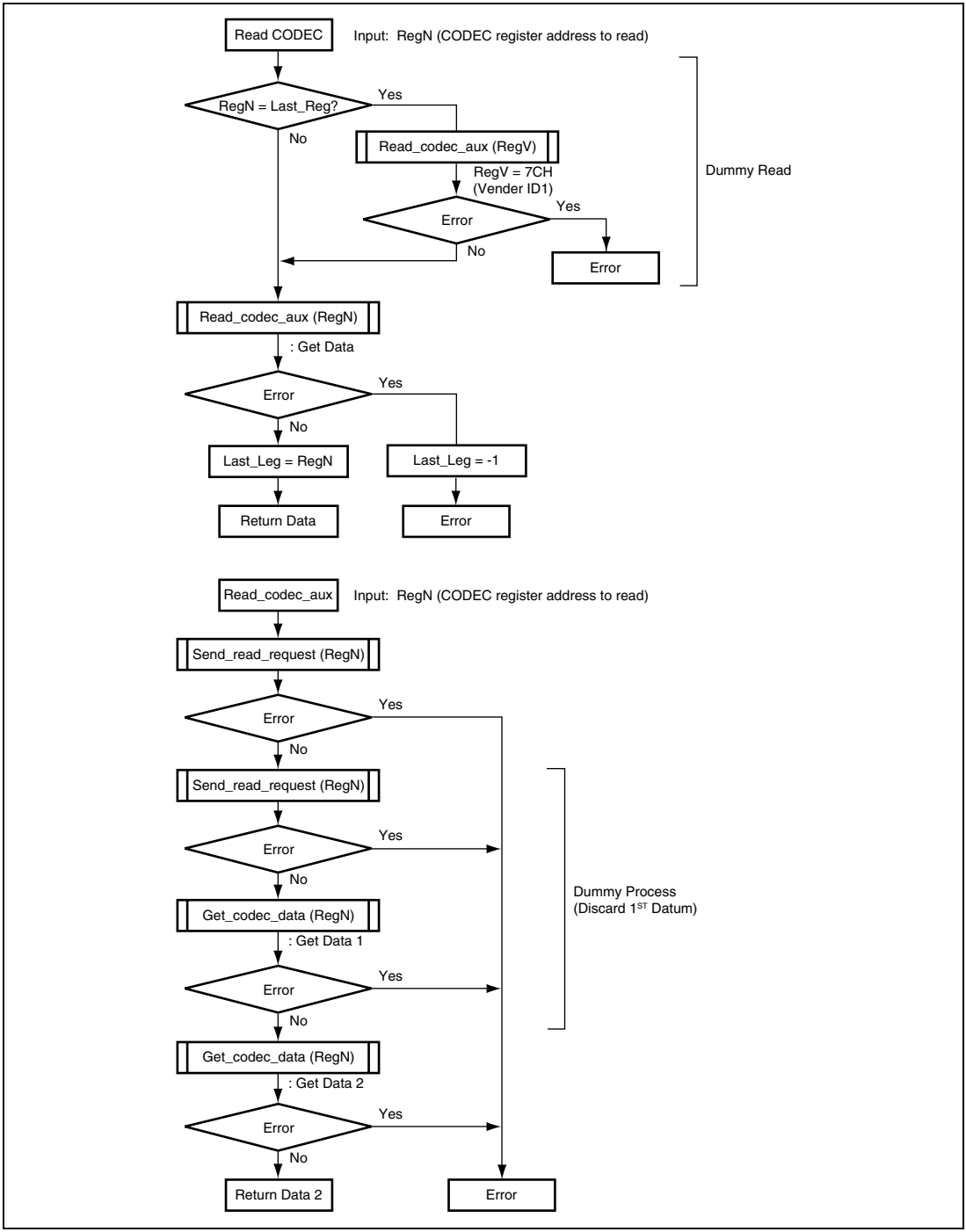


Figure 13.4 Access Flow Chart (2)

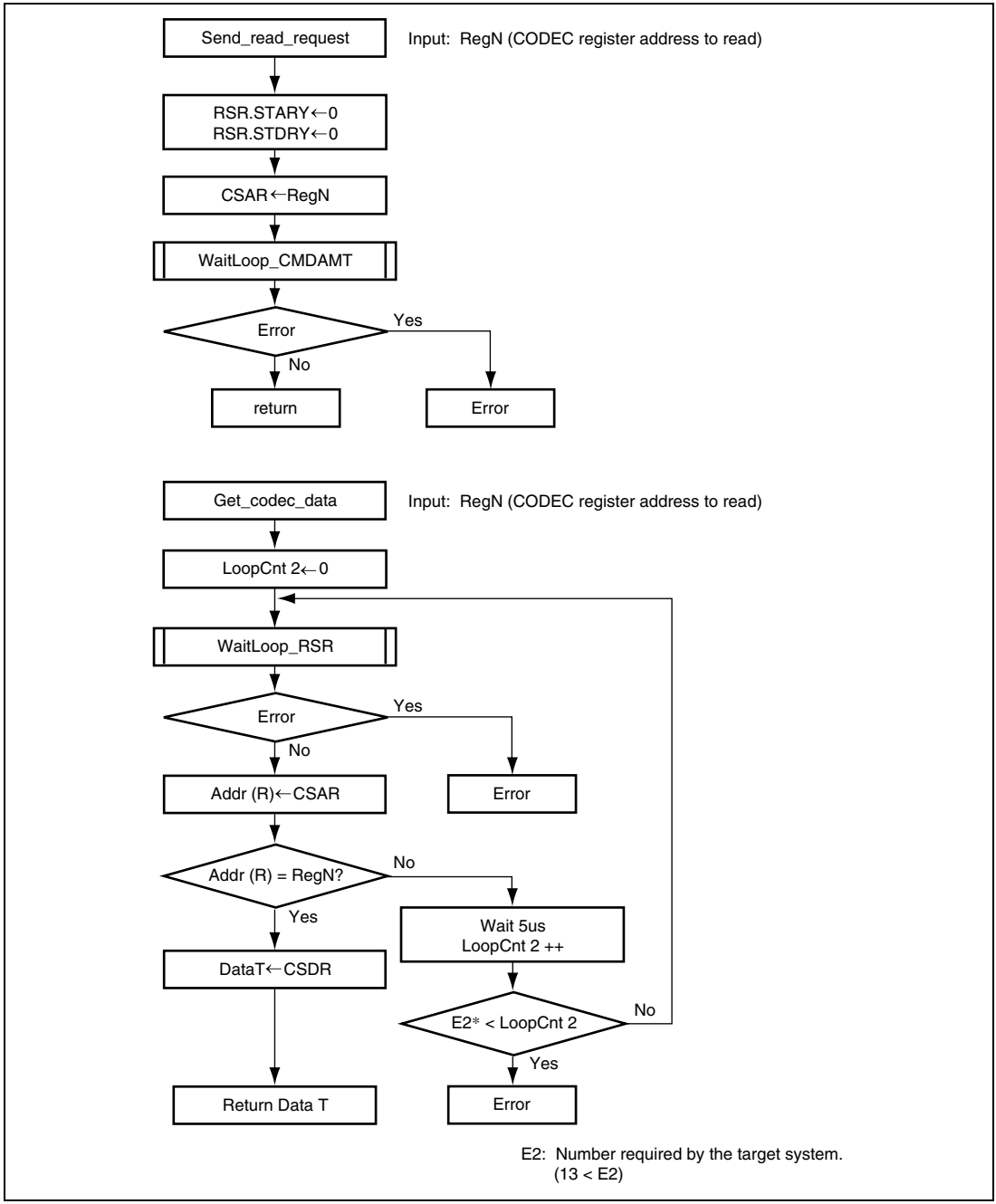


Figure 13.5 Access Flow Chart (3)

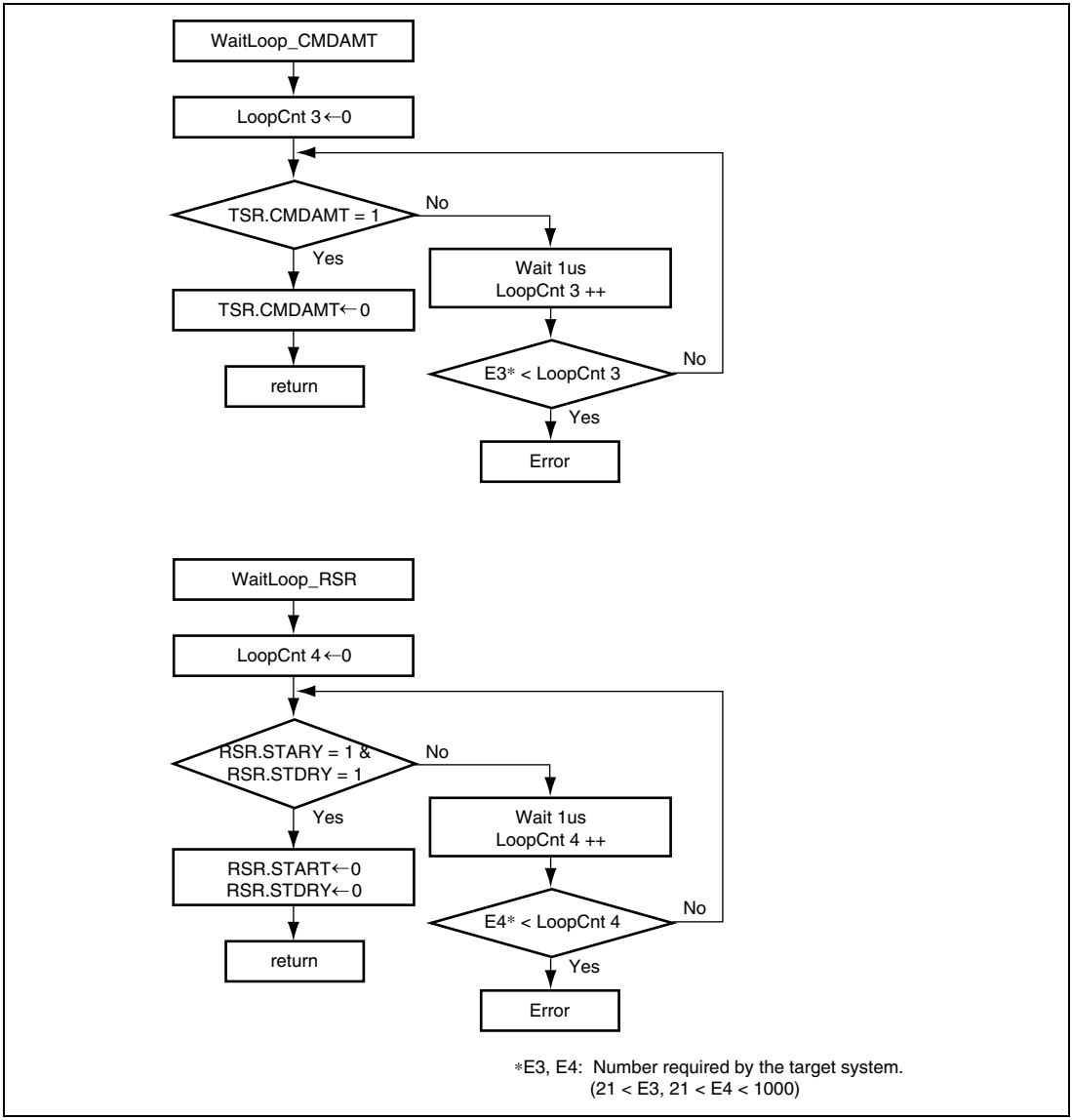


Figure 13.6 Access Flow Chart (4)

13.6.6 Module Standby

Standby mode can be enabled/disabled by controlling the Audio Codec bit in the Clock Control 1 (CC1) Register in the Power and Control module.

To wake up the module, the Audio Codec bit in the Clock Control 1 (CC1) Register should be enabled. After enabling this bit, all accesses to the Audio Codec module are possible.

To power down the module, the following procedure should be followed.

1. Ensure all data transfers have taken place. Ensure that the transmit buffer is empty and the receive buffer has been read until empty.
2. Disable all DMA requests and Interrupt requests.
3. Put the codec into power down mode.
4. Clear the Audio Codec bit in Clock Control 1 (CC1) Register.

13.6.7 General

The module will generate the AC_SYNC signal, which is used to indicate the position of slot 0 within the frame.

13.7 References

AC'97 Component Specification, Revision 2.1.

Section 14 Serial Sound Interface (SSI) Module

14.1 General Description

The Serial Sound Interface (here in after referred SSI) Module is a transceiver module designed to send or receive audio data interface with a variety of devices offering Philips format. It also provides additional modes for other common formats, as well as support for a burst and multi-channel mode.

14.2 Interfaces

The following block diagram shows how the Serial Sound Interface Module should be integrated into a system.

The module is primarily designed for a 32-bit bus system.

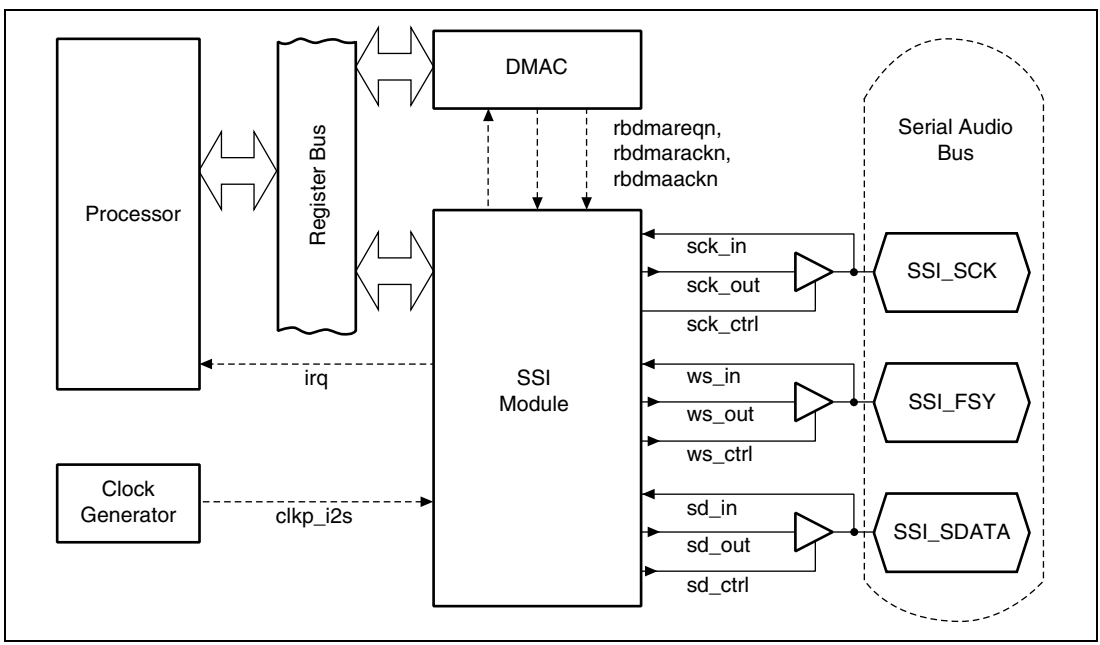


Figure 14.1 Interface Block Diagram

14.2.1 Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 14.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From
Register Bus	—		Access to Registers	
irq	1	out	Interrupt line	Processor
rbdmareqn	1	out	DMA request line	DMAC
rbdmarackn	1	in	DMA request acknowledge	DMAC
rbdmaackn	1	in	DMA cycle acknowledge	DMAC
clkp_i2s	1	in	Serial Oversample Clock (supplied from Audio Clock)	Clock Generator
sck_in	1	in	Serial Clock Input	from IO Buffer
sck_out	1	out	Serial Clock Output	to IO Buffer
sck_ctrl	1	out	Serial Clock Direction	to IO Buffer
ws_in	1	in	Word Select Input	from IO Buffer
ws_out	1	out	Word Select Output	to IO Buffer
ws_ctrl	1	out	Word Select Direction	to IO Buffer
sd_in	1	in	Serial Data In	from IO Buffer
sd_out	1	out	Serial Data Out	to IO Buffer
sd_ctrl	1	out	Serial Data Direction	to IO Buffer

14.2.2 Software Interfaces

The registers accessible by the software are listed in the following table. All registers must be written to and read from 32 bits at a time.

Table 14.2 Register List

Channel	Address (Bytes)	Register Name	Abbreviation	Access Size
0	H'6880	Control Register 0	CR0	32
	H'6884	Status Register 0	SR0	32
	H'6888	Transmit Data Register 0	TDR0	32
	H'688C	Receive Data Register 0	RDR0	32
1	H'68A0	Control Register 1	CR1	32
	H'68A4	Status Register 1	SR1	32
	H'68A8	Transmit Data Register 1	TDR1	32
	H'68AC	Receive Data Register 1	RDR1	32
2	H'68C0	Control Register 2	CR2	32
	H'68C4	Status Register 2	SR2	32
	H'68C8	Transmit Data Register 2	TDR2	32
	H'68CC	Receive Data Register 2	RDR2	32
3	H'68E0	Control Register 3	CR3	32
	H'68E4	Status Register 3	SR3	32
	H'68E8	Transmit Data Register 3	TDR3	32
	H'68EC	Receive Data Register 3	RDR3	32

14.3 Registers

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, read value undefined.

14.3.1 Control Register n (CR n) (n = 0 to 3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				DMEN	UIEN	OIEN	IIEN	DIEN	CHNL		DWL			SWL		
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	BREN	CKDV			MUEN	CPEN	TRMD	EN
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
28	DMEN	0	R/W	**DMA Enable (DMEN)** This enables the DMA Request module pin, dma_req. 0: DMA Request Line disabled, dma_req is permanently LOW. 1: DMA Request Line enabled, level will indicate current data status.

Bit	Bit Name	Initial Value	R/W	Description
27	UIEN	0	R/W	**Underflow IRQ Enable (UIEN)** 0: Underflow IRQ disabled 1: Underflow IRQ enabled
26	OIEN	0	R/W	**Overflow IRQ Enable (OIEN)** 0: Overflow IRQ disabled 1: Overflow IRQ enabled
25	IIEN	0	R/W	**Idle Mode IRQ Enable (IIEN)** 0: Idle IRQ disabled 1: Idle IRQ enabled
24	DIEN	0	R/W	**Data IRQ Enable (DIEN)** 0: Data IRQ disabled 1: Data IRQ enabled
23, 22	CHNL	0	R/W	**Channels (CHNL)** Number of Channels in each System Word. Field ignored if CPEN = 1. 00: 1 Channel Per System Word 01: 2 Channels Per System Word 10: 3 Channels Per System Word 11: 4 Channels Per System Word
21 to 19	DWL	0	R/W	**Data Word Length (DWL)** Encoded number of bits in a Data Word. Field ignored if CPEN = 1. 000: 8 Bits 001: 16 Bits 010: 18 Bits 011: 20 Bits 100: 22 Bits 101: 24 Bits 110: 32 Bits 111: Reserved

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	SWL	0	R/W	**System Word Length (SWL)** Encoded Number of Bits in a System Word. Field ignored if CPEN = 1. 000: 8 Bits 001: 16 Bits 010: 24 Bits 011: 32 Bits 100: 48 Bits 101: 64 Bits 110: 128 Bits 111: 256 Bits
15	SCKD	0	R/W	**Serial Clock Direction (SCKD)** 0: Serial Clock is Input, slave mode 1: Serial Clock is Output, master mode
14	SWSD	0	R/W	**Serial WS Direction (SWSD)** 0: Serial Word Select is Input, slave mode 1: Serial Word Select is Output, master mode
13	SCKP	0	R/W	**Serial Clock Polarity (SCKP)** 0: SSI_FSY and SSI_SDATA change on SSI_SCK falling edge (sampled on SCK rising edge) 1: SSI_FSY and SSI_SDATA change on SSI_SCK rising edge (sampled on SCK falling edge) • SCKP = 0 In Receive Mode (TRMD = 0), pin *sd_in* is sampled on SSI_SCK rising edge. In Transmit Mode (TRMD = 1), pin *sd_out* changes on the SSI_SCK falling edge. In Slave Mode (SCKD = 0), pin *ws_in* is sampled on SSI_SCK rising edge. In Master Mode (SCKD = 1), pin *ws_out* changes on the SSI_SCK falling edge. • SCKP = 1 In Receive Mode (TRMD = 0), pin *sd_in* is sampled on SSI_SCK falling edge. In Transmit Mode (TRMD = 1), pin *sd_out* changes on the SSI_SCK rising edge. In Slave Mode (SCKD = 0), pin *ws_in* is sampled on SSI_SCK falling edge. In Master Mode (SCKD = 1), pin *ws_out* changes on the SSI_SCK rising edge.

Bit	Bit Name	Initial Value	R/W	Description
12	SWSP	0	R/W	**Serial WS Polarity (SWSP)** The function of this bit depends on whether the module is in non-compressed mode or compressed mode. • CPEN = 0 (Non compressed mode) 0: SSI_FSY is LOW for 1st Channel, HIGH for 2nd Channel 1: SSI_FSY is HIGH for 1st Channel, LOW for 2nd Channel • CPEN = 1 (Compressed mode) 0: SSI_FSY is active HIGH Flow Control. WS = HIGH means data should be transferred, LOW means data should not be transferred. 1: SSI_FSY is active LOW Flow Control. WS = LOW means data should be transferred, HIGH means data should not be transferred.
11	SPDP	0	R/W	**Serial Padding Polarity (SPDP)** 0: Padding Bits are LOW 1: Padding Bits are HIGH Field ignored if CPEN = 1.
10	SDTA	0	R/W	**Serial Data Alignment (SDTA)** 0: Serial Data is Left Aligned 1: Serial Data is Right Aligned Field ignored if CPEN = 1.
9	PDTA	0	R/W	**Parallel Data Alignment (PDTA)** 0: Parallel Data (TD or RD) is Left Aligned 1: Parallel Data (TD or RD) is Right Aligned Field ignored if CPEN = 1. If the Data Word Length = 32, Data Word Length = 16 or Data Word Length = 8 then this configuration field has no meaning. This configuration field applies to the Receive Data Register in Receive Mode (TRMD = 0) and to the Transmit Data Register in Transmit Mode (TRMD = 1)

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	• DWL = 000 (decodes to 8 bits), PDTA ignored All Data Bits in RD or TD are as used on the Audio Serial Bus. Four data words are packed into each 32-bit access. The first Data Word is derived from bits 7 down to 0, the second from bits 15 down to 8, the third from bits 23 down to 16 and the last Data Word is derived from bits 31 down to 24. • DWL = 001 (decodes to 16 bits), PDTA ignored • All Data Bits in RD or TD are as used on the Audio Serial Bus. Two data words are packed into each 32-bit access. The first Data Word is derived from bits 15 down to 0 and the second Data Word is derived from bits 31 down to 16. • DWL = 010, 011, 100, 101 (decodes to 18, 20, 22 and 24 bits), PDTA = 0 (left aligned) The Data Bits which are used in RD or TD are the following: Bits 31 down to (32 - decoded DWL). i.e. if DWL = 011 then decoded DWL = 20 and therefore bits 31 down to 12 are used of either RD or TD. All other bits are ignored or reserved. • DWL = 010, 011, 100, 101 (decodes to 18, 20, 22 and 24 bits), PDTA = 1 (right aligned) The Data Bits which are used in RD or TD are the following: Bits (Decoded DWL - 1) down to 0 i.e. if DWL = 011 then decoded DWL = 20 and therefore bits 19 down to 0 are used of either RD or TD. All other bits are ignored or reserved. • DWL = 110 (decodes to 32 bits), PDTA ignored All Data Bits in RD or TD are as used on the Audio Serial Bus.

Bit	Bit Name	Initial Value	R/W	Description
8	DEL	0	R/W	**Serial Data Delay (DEL)** 0: 1 clock cycle delay between SSI_FSY and SSI_SDATA 1: no delay between SSI_FSY and SSI_SDATA Field ignored if CPEN = 1. A 1 clock cycle delay is not supported when the module is configured to be a slave transmitter. In this situation, DEL should be set to 1.
7	BREN	0	R/W	**Burst Mode Enable (BREN)** 0: Burst mode is not enabled 1: Burst mode is enabled. Burst mode is used in conjunction with compressed mode. When Burst mode is enabled the sck_out clock signal is gated. Clock pulses are only output when there is valid serial data being output on sd_out.
6 to 4	CKDV	0	R/W	**Serial Oversample Clock Divide Ratio (CKDV)** Defines the ratio between Oversample Clock, clkp_i2s, and the Serial Bit Clock. This field is ignored if SCKD = 0. The Serial Bit Clock is used for the Shift Register and is provided on the *sck_out* module pin. 000: (Serial Bit Clock Frequency = Oversample Clock Frequency ÷ 1) 001: (Serial Bit Clock Frequency = Oversample Clock Frequency ÷ 2) 010: (Serial Bit Clock Frequency = Oversample Clock Frequency ÷ 4) 011: (Serial Bit Clock Frequency = Oversample Clock Frequency ÷ 8) 100: (Serial Bit Clock Frequency = Oversample Clock Frequency ÷ 16) 101: Reserved 110: Reserved 111: Reserved
3	MUEN	0	R/W	**Mute Enable (MUEN)** 0: Module is not Muted 1: Module is Muted
2	CPEN	0	R/W	**Compressed Mode Enable (CPEN)** 0: Compressed Mode Disabled 1: Compressed Mode Enabled

Bit	Bit Name	Initial Value	R/W	Description
1	TRMD	0	R/W	**Transmit/Receive Mode Select (TRMD)** 0: Module is in Receive Mode 1: Module is in Transmit Mode
0	EN	0	R/W	**SSI Module Enable (EN)** 0: Module is Disabled 1: Module is Enabled

14.3.2 Status Register n (SR n) (n=0-3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
				DMRQ	UIRQ	OIRQ	IIRQ	DIRQ								
Initial:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R / WC0	R / WC0	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CHNO		SWNO	IDST
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
28	DMRQ	0	R	**DMA Request Status Flag (DMRQ)** This status flag allows the CPU to see the value of the dma_req pin on the module. The status bit has two different meaning depending on whether the module is in Transmit or Receive mode. • TRMD = 0 (Receive Mode) If DMRQ = 1 then the module Receive Data register (RD) has unread data. If the RD Register is Read then DMRQ = 0 until there is new unread data. • TRMD = 1 (Transmit Mode) If DMRQ = 1 the module Transmit Data register (TD) requires data to be written to continue the transmission onto the Audio Serial Bus. Once data is written to TD then DMRQ = 0 until it requires further transmit data.

Bit	Bit Name	Initial Value	R/W	Description
27	UIRQ	0	R/WC0	**Underflow Error IRQ Status Flag (UIRQ)** This status flag indicates that data was supplied at a lower rate than was required. The status bit has two different meanings depending on whether the module is in Transmit or Receive mode. In either case this bit is set regardless of the value of Under flow Interrupt Enable Bit (UIEN) and can be cleared by writing 0 to this bit. If UIRQ = 1 and UIEN = 1 then the module pin irq_req = 1. • TRMD = 0 (Receive Mode) If UIRQ = 1 then the module Receive Data register (RD) was read before there was new unread data indicated by DMAREQ or DIRQ status flags. This can lead to the same received sample being stored twice by the host leading to potential corruption of multi-channel data. • TRMD = 1 (Transmit Mode) If UIRQ = 1 then the Transmit Data register (TD) did not have data written to it before it was required for transmission. This will lead to the same sample being transmitted more than once and a potential corruption of multi-channel data. This is more serious error condition than a receive mode underflow as the output SSI data will in error as a consequence. Note: When underflow error occurs, the current data in the data buffer of this module is transmitted until the next data is filled. When the error bit is detected during DMA transfer, the equivalent DMA channel in DMAC must be aborted. After that, module enable and DMA enable bit must be disabled and the error status must be cleared. Then the module enable can be set and DMA can be initiated again.

Bit	Bit Name	Initial Value	R/W	Description
26	OIRQ	0	R/WC0	**Overflow Error IRQ Status Flag (OIRQ)** This status flag indicates that data was supplied at a higher rate than was required. The status bit has two different meanings depending on whether the module is in Transmit or Receive mode. In either case this bit is set regardless of the value of Over flow Interrupt Enable Bit (OIEN) and can be cleared by writing 0 to this bit. If OIRQ = 1 and OIEN = 1 then the module pin irq_req = 1. • TRMD = 0 (Receive Mode) If OIRQ = 1 then the module Receive Data Register (RD) was not read before there was new unread data written to it. This will lead to the loss of a sample and a potential corruption of multi-channel data. • TRMD = 1 (Transmit Mode) If OIRQ = 1 then the Transmit Data Register (TD) had data written to it before it was transferred to the shift register. This will lead to the loss of a sample and a potential corruption of multi-channel data. Note: When overflow error occurs, the current data in the data buffer of this module is overwritten by the next incoming data from SSI interface. When the error bit is detected during DMA transfer, the equivalent DMA channel in DMAC must be aborted. After that, module enable and DMA enable bit must be disabled and the error status must be cleared. Then the module enable can be set and DMA can be initiated again.

Bit	Bit Name	Initial Value	R/W	Description
25	IIRQ	1	R	**Idle Mode IRQ Status Flag (IIRQ)** This IRQ Status flag indicates that the SSI Module is in the idle state. This bit is set regardless of the value of Idle Interrupt Enable Bit (IIEN) to allow polling. The IRQ (module pin irq_req) can be masked by clearing IIEN, but cannot be cleared by writing to this bit. If IIRQ = 1 and IIEN = 1 then the module pin irq_req = 1. 0: Module not in Idle State (IDLE = 0). 1: Module in Idle State (IDLE = 1).
24	DIRQ	0	R	**Data IRQ Status Flag (DIRQ)** This status flag indicates that the module has data to be read or requires data to be written. The status bit has two different meanings depending on whether the module is in Transmit or Receive mode. In either case this bit is set regardless of the value of Data Interrupt Enable Bit (DIEN) to allow polling. The IRQ (module pin *irq_req*) can be masked by clearing DIEN, but cannot be cleared by writing to this bit. If DIRQ = 1 and DIEN = 1 then the module pin irq_req = 1. • TRMD = 0 (Receive Mode) 0: No unread data in the Receive Data Register (RD) 1: Unread data in the Receive Data Register (RD) • TRMD = 1 (Transmit Mode) 0: Transmit Buffer is Full. 1: Transmit Buffer is Empty and requires data to be written to the Transmit Data Register (TD)
23 to 4	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	CHNO	0	R	**Channel Number (CHNO)** This value indicates which channel is currently available to the parallel bus. The value has two different meanings depending on whether the module is in Transmit or Receive mode. • TRMD = 0 (Receive Mode) CHNO indicates which channel the data in the Receive Data Register (RD) currently represents. This value will change as the data in RD is updated from the shift register. • TRMD = 1 (Transmit Mode) CHNO indicates which channel is required to be written to the Transmit Data Register (TD). This value will change as the data is copied to the shift register, regardless of whether or not the data is written to the TD register.
1	SWNO	1	R	**System Word Number (SWNO)** This Status bit indicates which System word is currently available to the parallel bus. The bit has two different meanings depending on whether the module is in Transmit or Receive mode. • TRMD = 0 (Receive Mode) SWNO indicates which system word the data in the Receive Data Register (RD) currently represents. This value will change as the data in RD is updated from the shift register, regardless of whether or not the RD Register has been read. • TRMD = 1 (Transmit Mode) SWNO indicates which system word is required to be written to the Transmit Data Register (TD). This value will change as the data is copied to the shift register, regardless of whether or not the data is written to the TD Register.

Bit	Bit Name	Initial Value	R/W	Description
0	IDST	1	R	**Idle Mode Status Flag (IDMD)** This status flag indicates that the Serial Bus activity has ceased. IDMD = 0 if EN = 1 and the Serial Bus is currently active. This bit can be set under the following conditions. • SSI = Serial Bus Transmit Master IDMD = 1 if no more data has been written to the Transmit Data Register (TD) and the current system word has been completed. It can also be set by clearing the EN bit after sufficient data has been written to TD to finish off the system word currently being output. • SSI = Serial Bus Receive Master IDMD = 1 if the EN bit is cleared and the current system word is completed. SSI = Serial Bus Transmit or Receive Slave IDMD = 1 if the EN bit is cleared and the current system word is completed. Note: If the external master stops the Serial Bus Clock before the current system word is completed then IDLE will never be set.

14.3.3 Transmit Data Register n (TDR n) (n=0-3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TD															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TD															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	TD	0	R/W	**Transmit Data (TD)** Data written to this register is passed to the Shift Register as it is required for transmission. If the Data Word Length < 32 bits then its alignment should be as defined by the PDTA Control Bit. Reading this register will return the data in the Buffer.

14.3.4 Receive Data Register (RDR) (n=0-3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RD															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RD															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RD	0	R	**Received Data (RD)** Data in this register is passed from the Shift Register as each Data Word is received. If the Data Word Length < 32 bits then the data is presented in the format specified by the PDTA Control Bit.

14.4 SSI Module Operation

The module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus formats can be one of eight major modes as shown in the following table.

Description	TXRXMD	CPEN	SCKD	SWSD	EN	MUEN	DIEN	IIEN	OIEN	UIEN	DEL	PDTA	SDTA	SPDP	SWSP	SCKP	SWL(2:0)	DWL(2:0)	CHNL(1:0)
Non-Compressed Slave Receiver	0	0	0	0			Control								Config				
Non-Compressed Slave Transmitter	1	0	0	0															
Non-Compressed Master Receiver	0	0	1	1															
Non-Compressed Master Transmitter	1	0	1	1															
Compressed Slave Receiver	0	1	0/1	1			Control					Ignored			Config Bits		Ignored		
Compressed Slave Transmitter	1	1	0/1	1															
Compressed Master Receiver	0	1	0/1	0															
Compressed Master Transmitter	1	1	0/1	0															

The control bits are valid in every mode, but only some of the configuration bits are valid in the compressed modes.

The module operation is now broken down into:

- A description of the major modes, how the pins are configured.
- How the Configuration Bits affect the major modes.
- How the Control and Status bits should be used to control data flow, in transmit and receive modes.

14.4.1 Non-Compressed Modes

This Major mode is designed to support all Serial Audio Streams which are split into channels. It can support Philips, Sony and Matsushita modes as well as many more variants on these modes.

This Major Mode can be further defined in Receiver/Transmitter mode and Master/Slave mode by the following Configuration bits:

TRMD, CPEN, SCKD, SWSD

The configuration of the serial stream is defined in this Major mode by the following configuration bits:

DEL, PDTA, SDTA, SPDP, SWSP, SCKP, SWL, DWL, CHNL

Slave Receiver

SSI_SDATA Input, SSI_SCK Input, SSI_FSY Input

Required Register Bit Definitions:
TRMD = 0, CPEN = 0, SCKD = 0, SWSD = 0

Description:
This mode allows the module to receive serial data from another device. The clock and word select used for the serial data stream is also supplied from an external device. If these signals do not conform to the format as specified in the configuration fields of the module then operation is not guaranteed.

Slave Transmitter

SSI_SDATA Output, SSI_SCK Input, SSI0_FSY Input

Required Register Bit Definitions:
TRMD = 1, CPEN = 0, SCKD = 0, SWSD = 0

Description:
This mode allows the module to transmit serial data to another device. The clock and word select used for the serial data stream is also supplied from an external device. If these signals do not conform to the format as specified in the configuration fields of the module then operation is not guaranteed.

Master Receiver

SSI_SDATA Input, SSI_SCK Output, SSI0_FSY Output

Required Register Bit Definitions:
TRMD = 0, CPEN = 0, SCKD = 1, SWSD = 1

Description:
This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the *clkp_i2s* input clock. The format of these signals is as defined in the configuration fields of the module. If the incoming data does not follow the configured format then operation is not guaranteed.

Master Transmitter

SSI_SDATA Output, SSI_SCK Output, SSI0_FSY Output

Required Register Bit Definitions:
TRMD = 1, CPEN = 0, SCKD = 1, SWSD = 1

Description:
This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the *clkp_i2s* input clock. The format of these signals is as defined in the configuration fields of the module.

Configuration Fields - Word Length Related

All configuration fields are valid in Non-Compressed Modes (CPEN = 0).

There are many configurations that the SSI Module can support and it is not sensible to show all of the Serial Data formats in this document. Some of the combinations are shown below for the popular formats by Philips, Sony, Matsushita and others.

Philips Format

The following two diagrams demonstrate the supported Philips protocol both with padding and without. Padding occurs when the data word length (decoded from DWL) is smaller than the system word length (decoded from SWL).

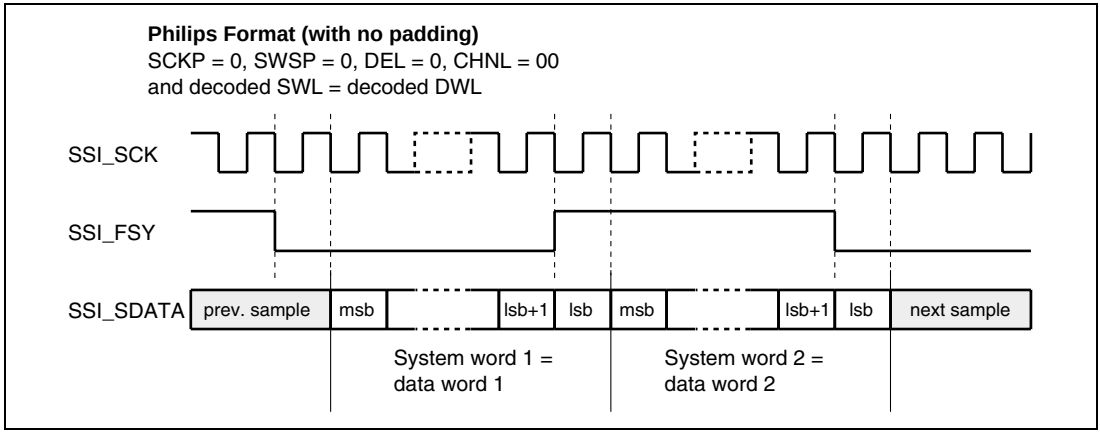


Figure 14.2 Philips Format (with no Padding)

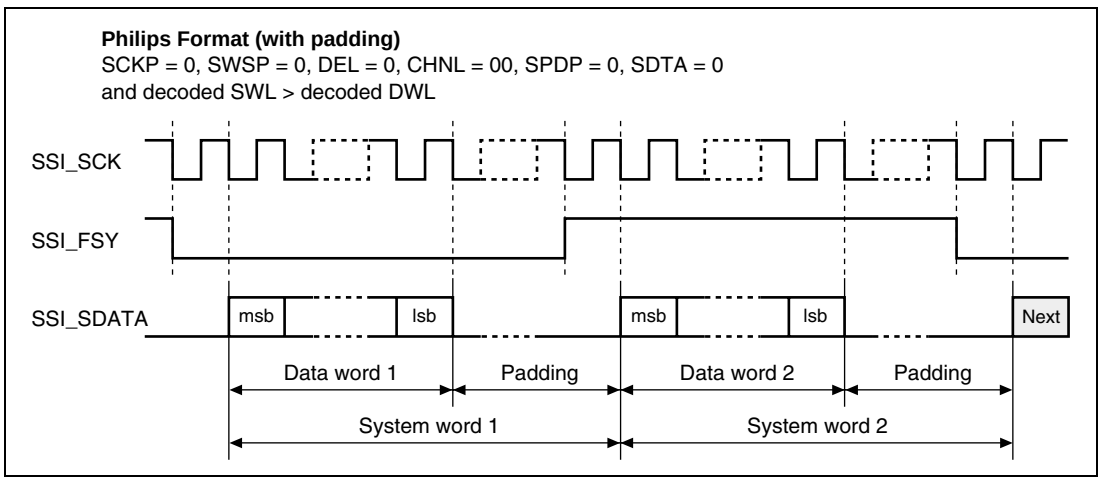


Figure 14.3 Philips Format (with Padding)

The following two diagrams show the formats used by Sony and Matsushita. Padding is assumed in both cases, but may not be present in a final implementation if the decoded System Word Length equals the decoded Data Word Length.

Sony Format

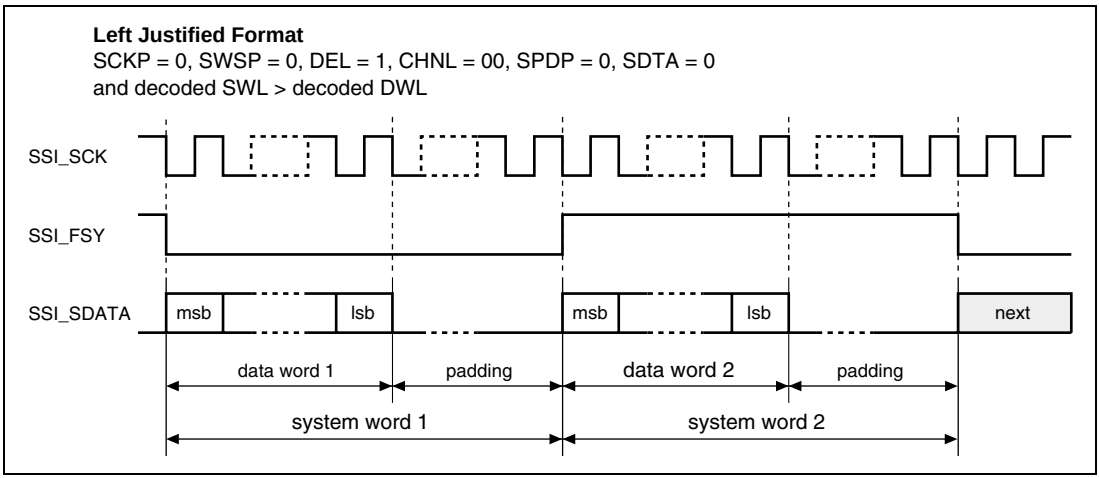


Figure 14.4 Sony Format (Left Justified)

Matsushita Format

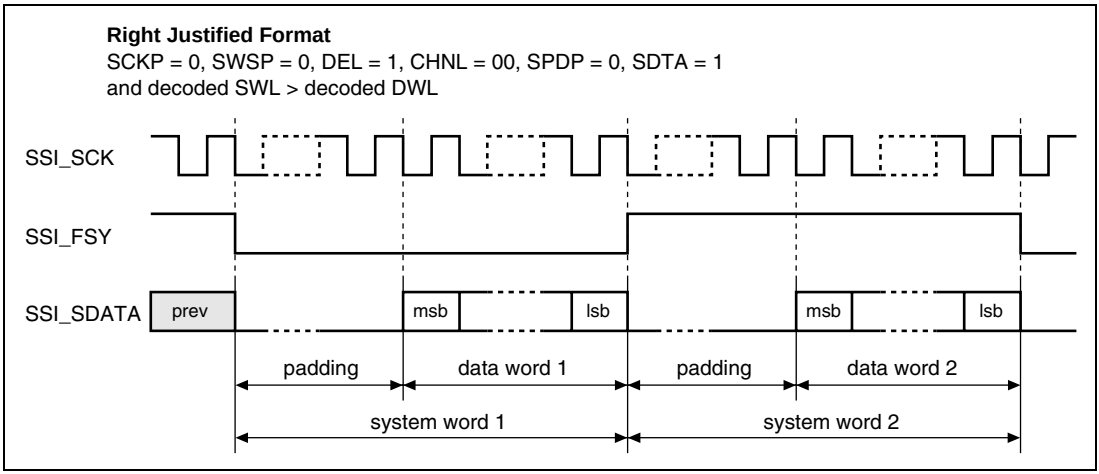


Figure 14.5 Matsushita Format (Right Justified)

HITACHI

Multi-channel Formats

Some devices extend the definition of the specification by Philips and allow more than 2 channels to be transferred within two system words.

The SSI Module supports the transfer of 4, 6 and 8 channels by the use of the CHNL, SWL and DWL configuration fields. It is important that the System Word Length (decoded from SWL) is greater than or equal to the number of Channels (decoded from CHNL) times the Data Word Length (decoded from DWL).

The following table shows the number of padding bits for each of the valid configurations. If a setup is not valid it does not have a number in the following table and has instead a dash.

Table 14.3 The Number of Padding Bits for Each Valid Configuration

Padding Bits Per System Word		DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

In the case of the SSI module configured as a transmitter then each word that is written to the TD register is transmitted in order on the Serial Audio bus.

In the case of the SSI module configured as a receiver each word received on the Serial Audio Bus is presented for reading by the RD register.

The following diagrams show how 4, 6 and 8 channels are transferred on the Serial Audio Bus.

Note that there are no padding bits in the first example, the second example is left aligned and the third is right aligned. This selection is purely arbitrary and is just for demonstration purposes only.

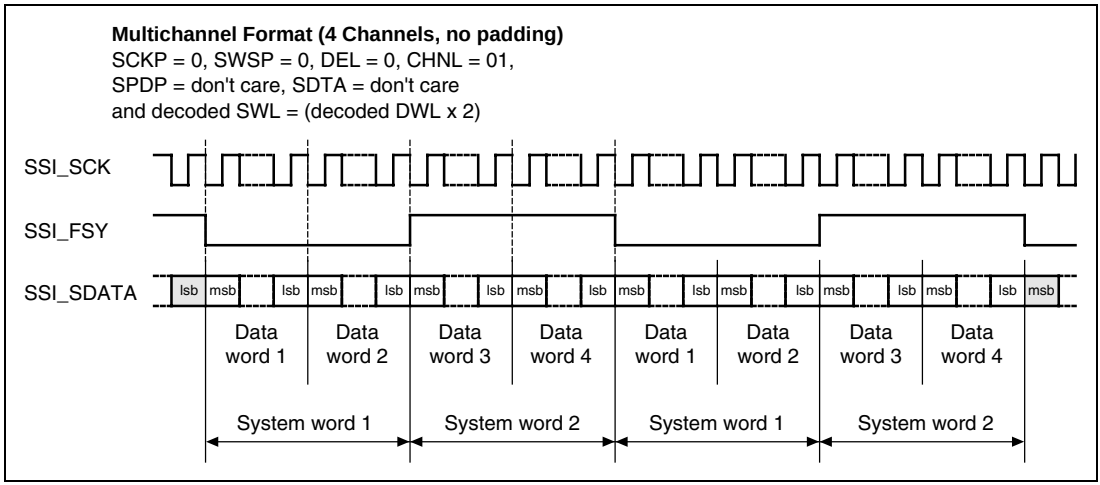


Figure 14.6 Multichannel Format (4 Channels, No Padding)

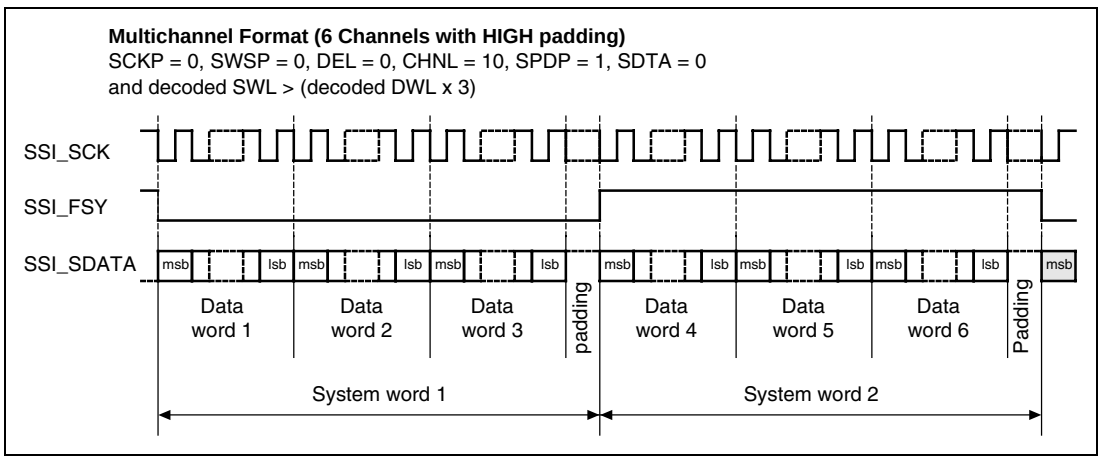


Figure 14.7 Multichannel Format (6 Channels with High Padding)

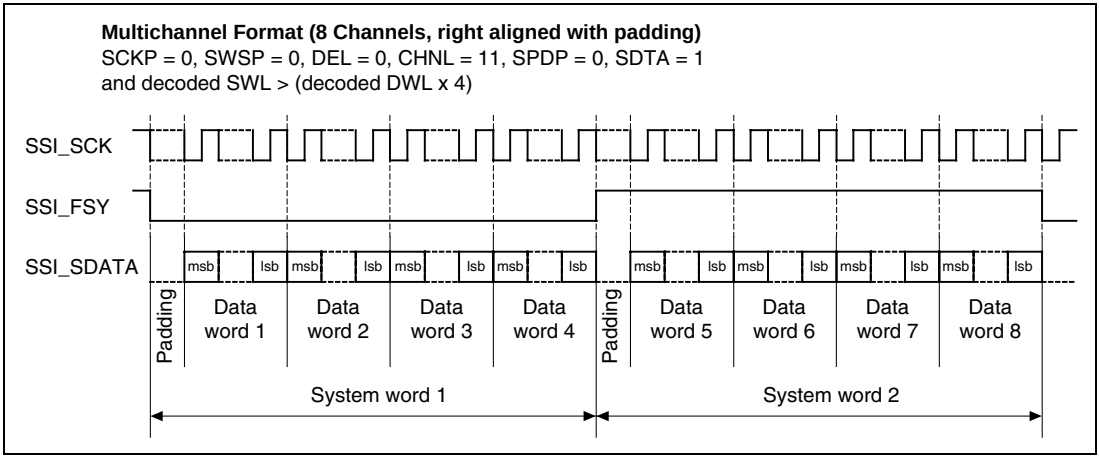


Figure 14.8 Multichannel Format (8 Channels, Right Aligned with Padding)

Configuration Fields - Signal Format Fields

There is several more configuration fields in non-compressed mode which will now be demonstrated. These bits are NOT mutually exclusive, however some configurations will probably not be useful for any other device.

They are demonstrated by referring to the following Basic Sample Format Diagram.

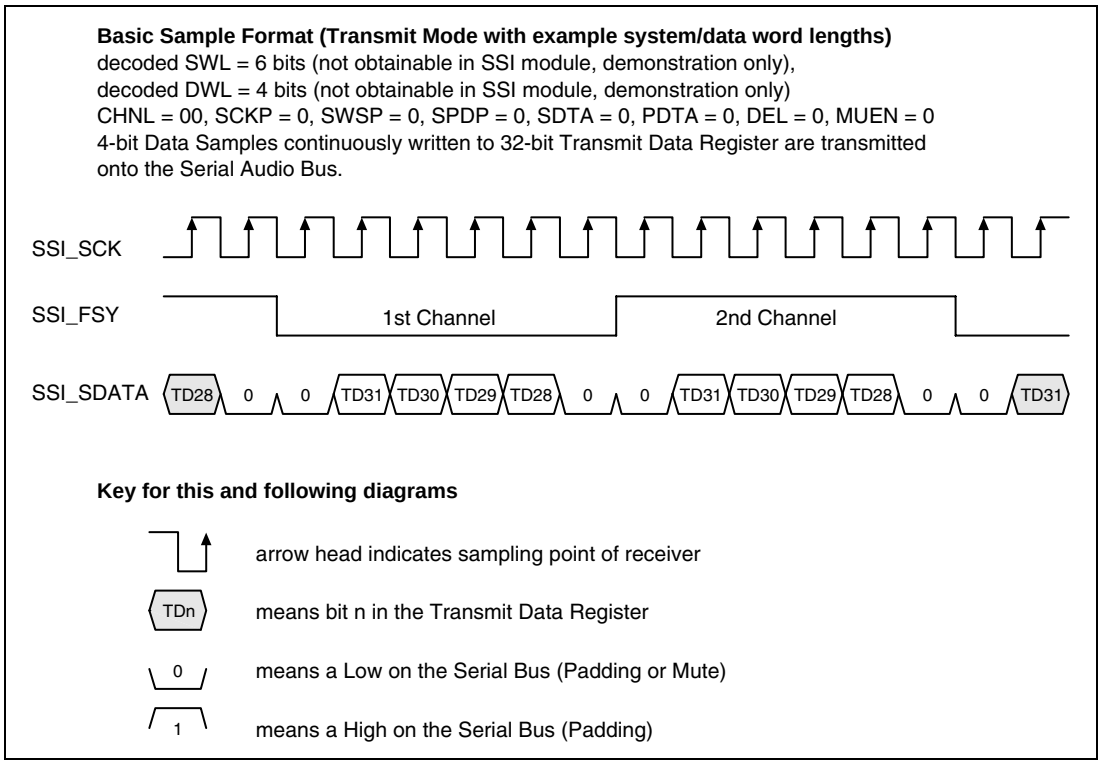


Figure 14.9 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

This diagram uses a decoded System Word Length of 6 bits and a decoded Data Word Length of 4 bits. Neither of these is possible with the SSI module but are used only for clarification of the other configuration bits.

Inverted Clock

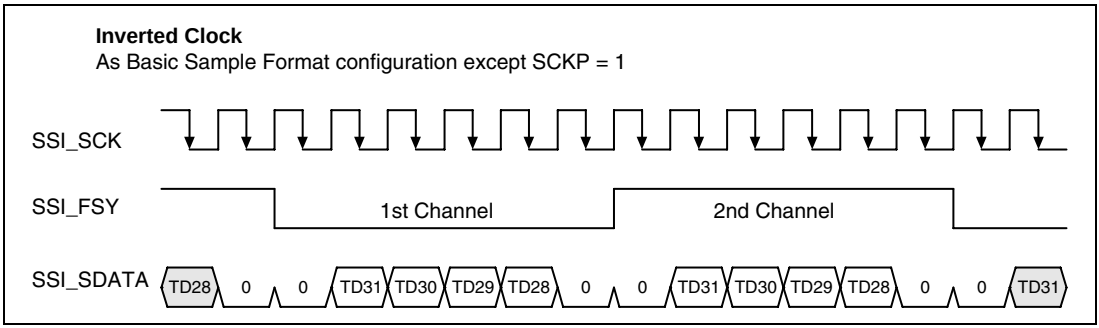


Figure 14.10 Inverted Clock

Inverted Word Select

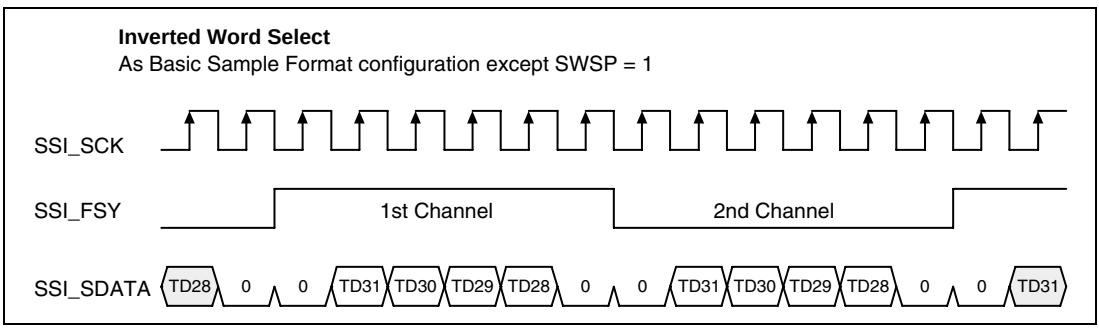


Figure 14.11 Inverted Word Select

Inverted Padding Polarity

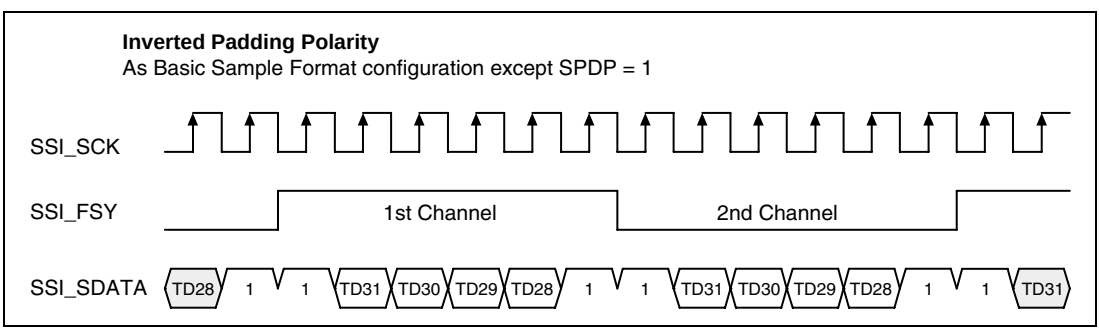


Figure 14.12 Inverted Padding Polarity

Serial Right Aligned with Delay

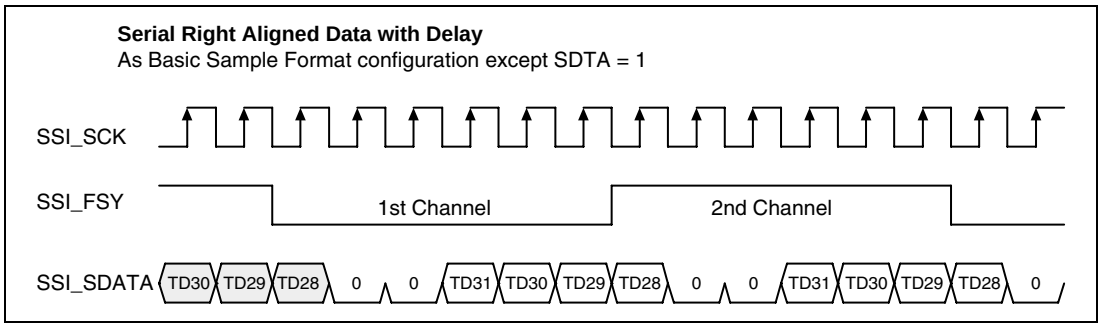


Figure 14.13 Serial Right Aligned with Delay

Serial Right Aligned without Delay

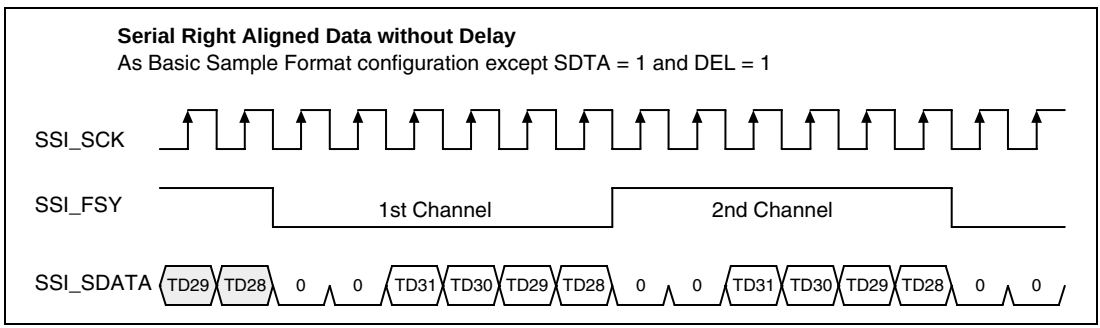


Figure 14.14 Serial Right Aligned without Delay

Serial Left Aligned without Delay

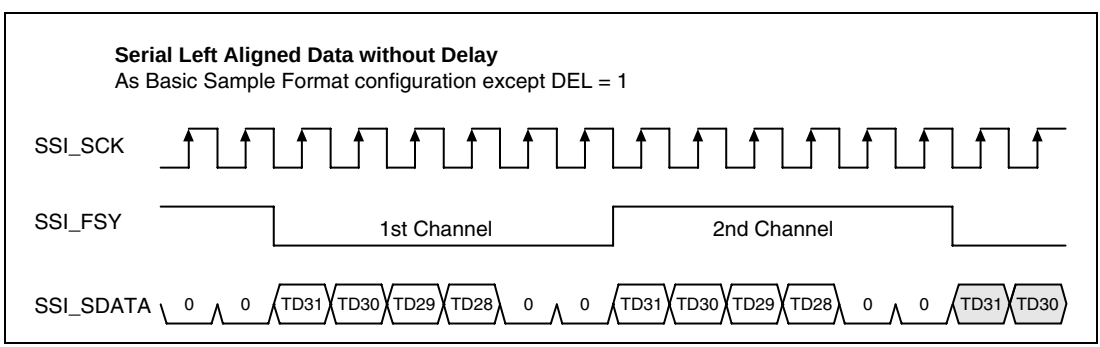


Figure 14.15 Serial Left Aligned without Delay

Parallel Right Aligned with Delay

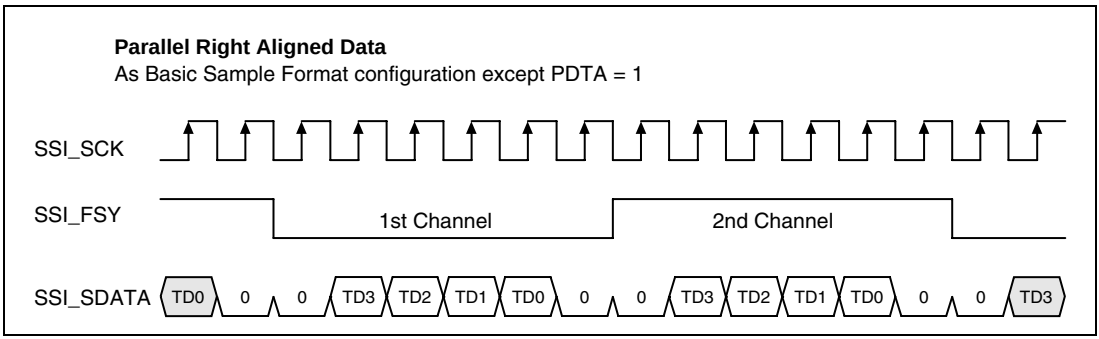


Figure 14.16 Parallel Right Aligned with Delay

Mute Enabled

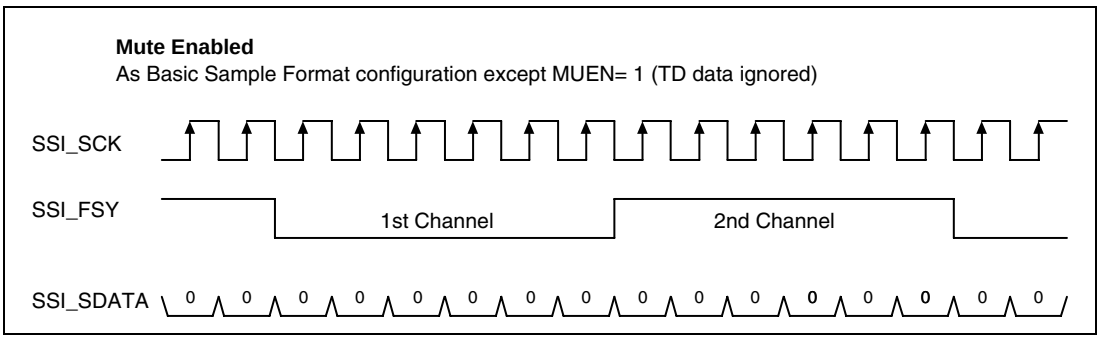


Figure 14.17 Mute Enabled

14.4.2 Compressed Modes

This mode is used to transfer a continuous bit stream. This would typically be a compressed bit stream which requires downstream decoding.

When in streaming mode (burst mode not enabled) there is no concept of a data word. However in order to receive and transmit it is necessary to transfer between the serial bus and word formatted memory. Therefore the word boundary selection is arbitrary during receive/transmit and must be dealt with by another module. When burst mode is enabled then data bits being transmitted can be identified by virtue of the fact that the serial clock output is only activated when there is a word to output and only the required number of clock pulses necessary to clock out each 32-bit word are generated. Note burst mode is only valid in the context of the module being a transmitter of data. Burst mode data cannot be received by this module.

Data is transmitted and received in blocks of 32 bits, and the first bit received/transmitted is bit 31 when stored in memory.

The Word Select Pin in this mode does not act as a System Word start signal as in non-compressed mode, but instead is used to indicate that the receiver can receive another data burst, or the transmitter can transmit another data burst.

This Major Mode can be further defined in Receiver/Transmitter mode and Master/Slave mode by the following Configuration bits:

TRMD, CPEN, SCKD, SWSD, BREN

The configuration of the serial stream is defined in this Major mode by the following configuration bits:

SWSP, SCKP

The sensitivity of the word select pin, which is used for flow control, can be changed by the use of the SWSP configuration field. Likewise the setup/sample point of the data can be changed by the use of the SCKP configuration field.

The following configuration bits have no meaning in compressed mode:

DEL, PDTA, SDTA, SPDP, SWL, DWL, CHNL.

The following diagrams illustrate compressed mode data transfer, with burst mode first disabled and then enabled.

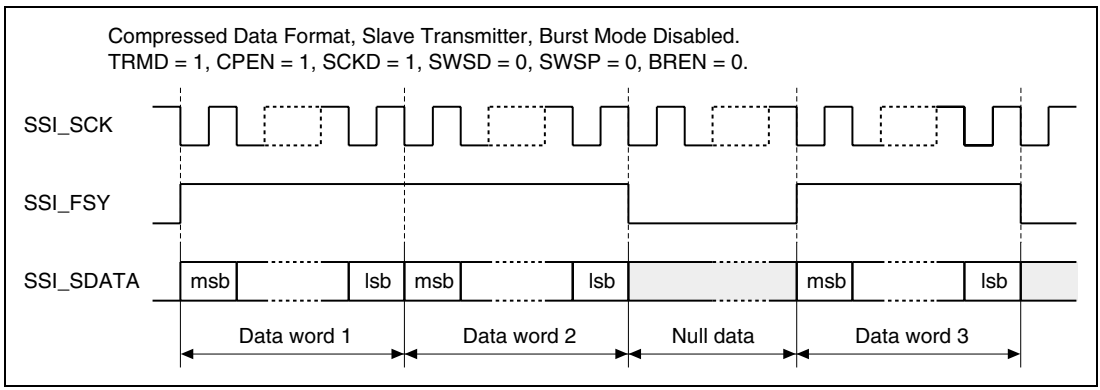


Figure 14.18 Compressed Data Format, Slave Transmitter, Burst Mode Disabled

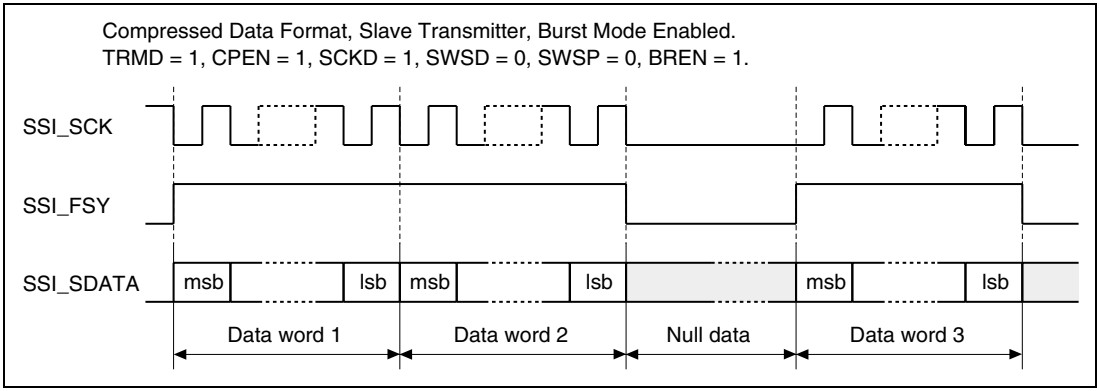


Figure 14.19 Compressed Data Format, Slave Transmitter, and Burst Mode Enabled

Slave Receiver

SSI_SDATA Input, SSI_FSY (flow control) Input

Required Register Bit Definitions:
TRMD = 0, CPEN = 1, SWSD = 0, SCKD = application dependent

Description:
This mode allows the module to receive a serial bit stream from another device and store it in memory, typically under the control of the systems' DMAC.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an input flow control. Assuming that SWSP = 0 if ws_in is HIGH then the module will receive the bit stream in blocks of 32 bits, one data bit per clock. If ws_in goes LOW then the module will complete the current 32-bit block and then stop any further reception, until ws_in goes HIGH again.

Slave Transmitter

SSI_SDATA Output, SSI_FSY (flow control) Input

Required Register Bit Definitions:
TRMD = 1, CPEN = 1, SWSD = 0, SCKD = application dependent

Description:
This mode allows the module to transmit a serial bit stream from local memory to another device, typically under the control of the systems' DMAC.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an input flow control. Assuming that SWSP = 0, if *ws_in* is HIGH then the module will keep transferring the TD Buffer to the shift register and transmitting it in blocks of 32 bits, one data bit per clock. If *ws_in* goes LOW then the module will complete the current 32-bit block and then cease any further transmission, until *ws_in* goes HIGH again.

Master Receiver

SSI_SDATA Input, SSI_FSY (flow control) Output

Required Register Bit Definitions:
TRMD = 0, CPEN = 1, SWSD = 1, SCKD = application dependent

Description:
This mode allows the module to receive a serial bit stream from another device and store it in memory, typically under the control of the systems' DMAC.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an output flow control. The module always asserts word select to indicate it can receive more data. It is the responsibility of the host CPU to ensure it can service the module in time to ensure no data is lost.

Master Transmitter

SSI_SDATA Output, SSI_FSY (flow control) Output

Required Register Bit Definitions:
TRMD = 1, CPEN = 1, SWSD = 1, SCKD = application dependent

Description:
This mode allows the module to transmit a serial bit stream from local memory to another device, typically under the control of the systems' DMAC.

The shift register clock can be supplied from an external device or from an internal clock.

The word select pin is used as an output flow control. The module always asserts word select to indicate it will transmit more data. Word select is not asserted until the first word is ready to transmit however. It is the responsibility of the receiving device to ensure it can receive the serial data in time to ensure no data is lost.

14.5 Module Operation

The SSI module is designed to require minimum interaction from the CPU once it has been configured for a data transfer. The processor is required to configure the SSI and DMAC modules and handle any overflow/underflow IRQs if necessary.

14.5.1 Operation Modes

There are three modes of operation: configuration, enabled and disabled. The following diagram shows how the module enters each of these modes.

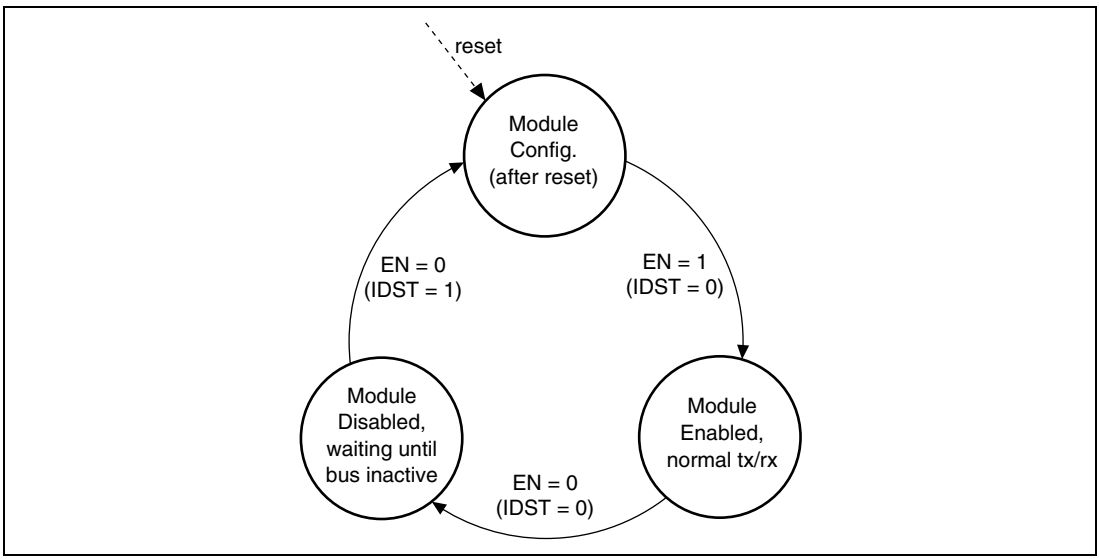


Figure 14.20 Operation Modes

Configuration Mode

This mode is entered after the module is released from reset. All required configuration fields in the Control Register should be defined in this mode, before the module is enabled by setting the EN bit.

Setting the EN bit causes the module to enter the Module Enabled Mode.

Module Enabled Mode

Operation of the module in this mode is dependent on the configuration selected. It is mainly affected by whether the module is in Receive or Transmit Mode as described in the Transmit Operation and Receive Operation Sections.

14.5.2 Transmit Operation

Transmission can be achieved in one of two ways: either DMA or IRQ driven.

DMA driven is preferred to reduce the Processor Load, however this does depend on the availability of a DMA Controller in the system. In this mode the Processor will only receive interrupts if there is an underflow or overflow of data or the DMAC has finished its transfer.

The alternative is using the IRQs that the SSI module generates to supply data as required. This mode has a higher IRQ load as the module is only double buffered and will require data to be written at least every system word period.

One last alternative to control transmission is to poll the status register. This method is not described in this document.

When the SSI module has been enabled for transmission, at least one longword must be written to the transmit register before disabling the transmitter (In 16b mode, two 16b words will be transmitted; in 8b mode, 4 bytes will be transmitted. For all other data sizes, 1 data word will be transmitted, e.g., 18b for 18b mode.)
Failure to do this will result in a lockup, which requires a hard reset.
When disabling the module, the SSI clock must remain present until the module is in the idle state, indicated by the idle interrupt bit.

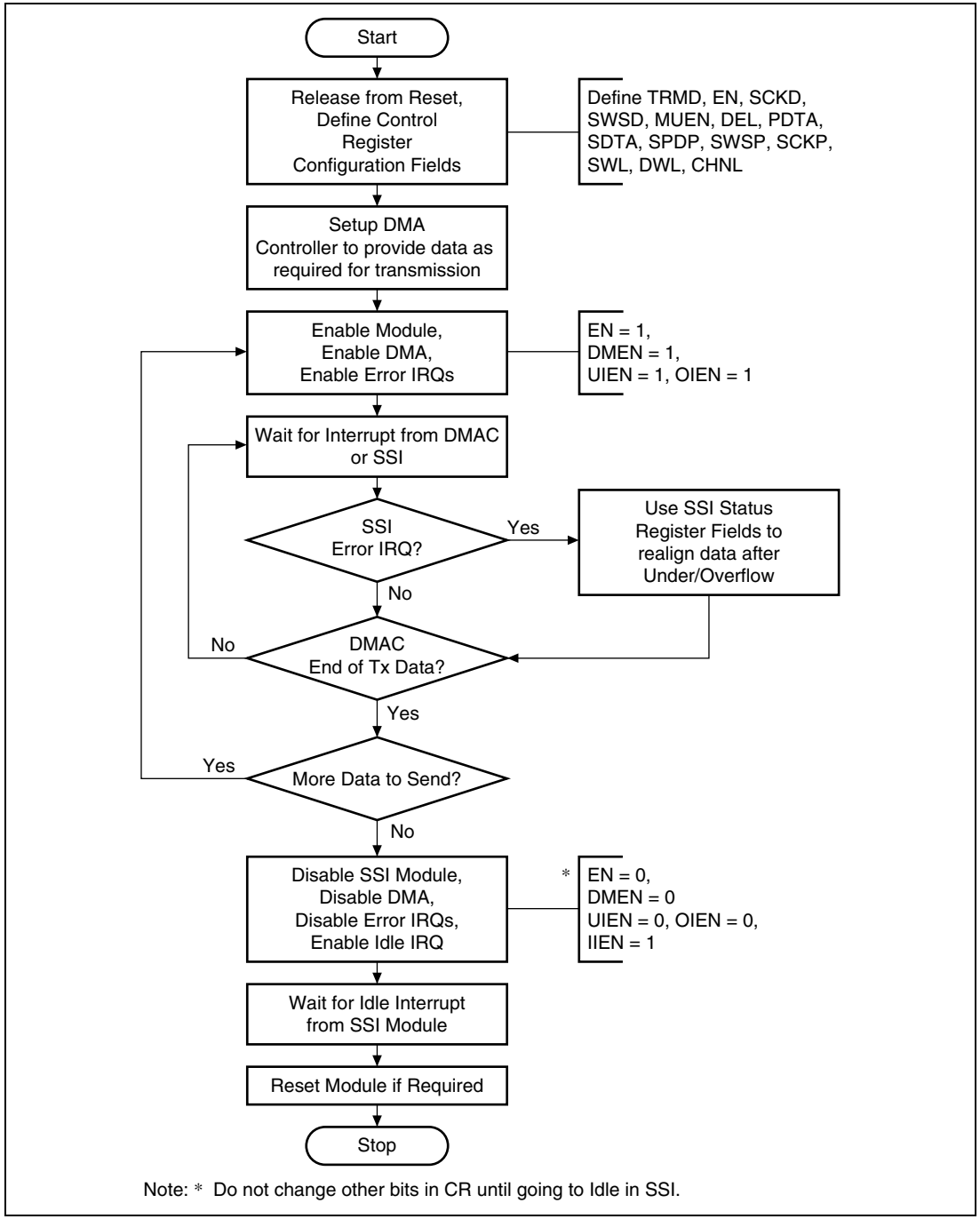


Figure 14.21 Transmission Using DMA Controller

Transmission using IRQ Data Flow Control

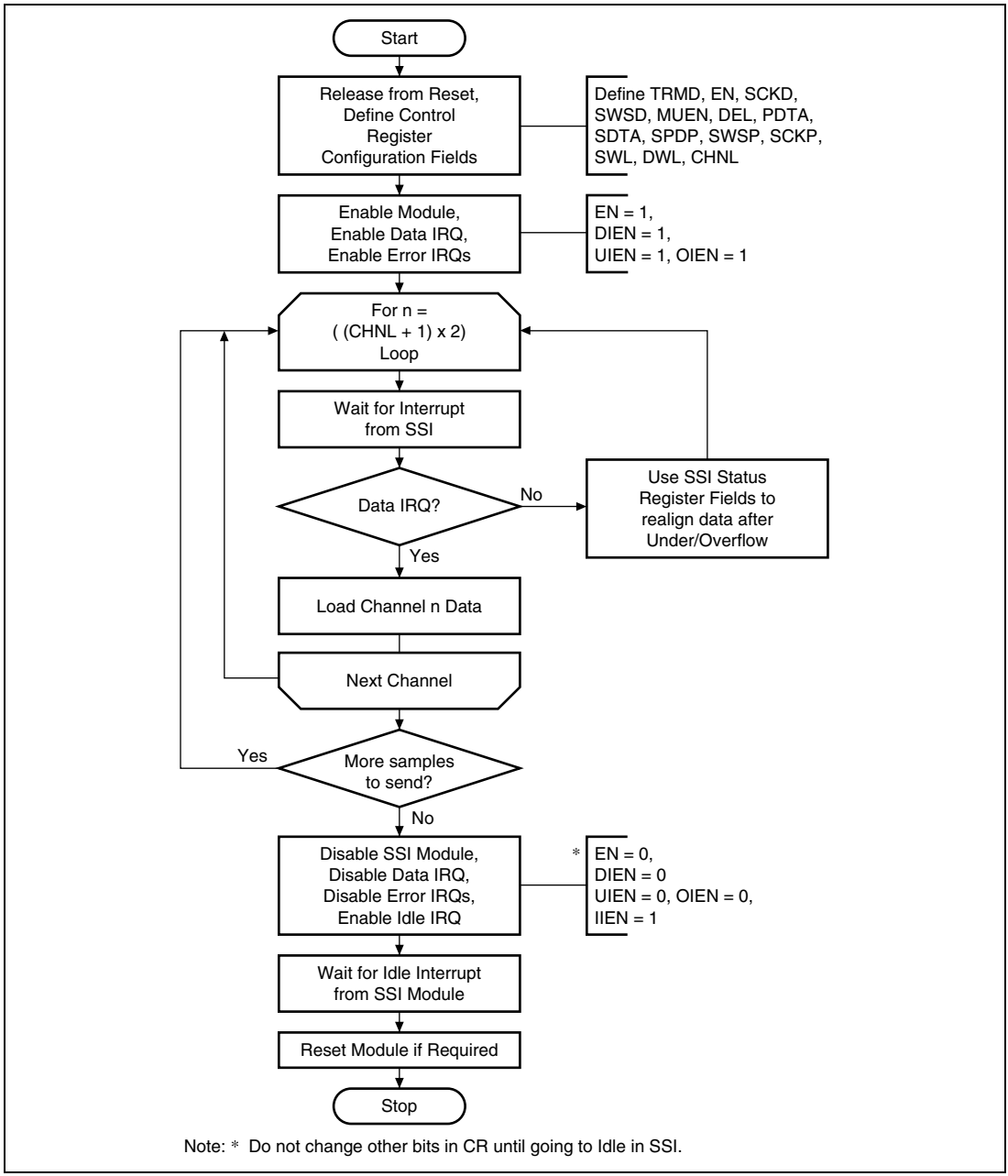


Figure 14.22 Transmission using IRQ Data Flow Control

14.5.3 Receive Operation

As with Transmission the Reception can be achieved in one of two ways: either DMA or IRQ driven.

The following two flowcharts demonstrate the flow of operation.

When disabling the module, the SSI clock must retain present until the module is in the idle state, which is indicated by the idle interrupt bit.

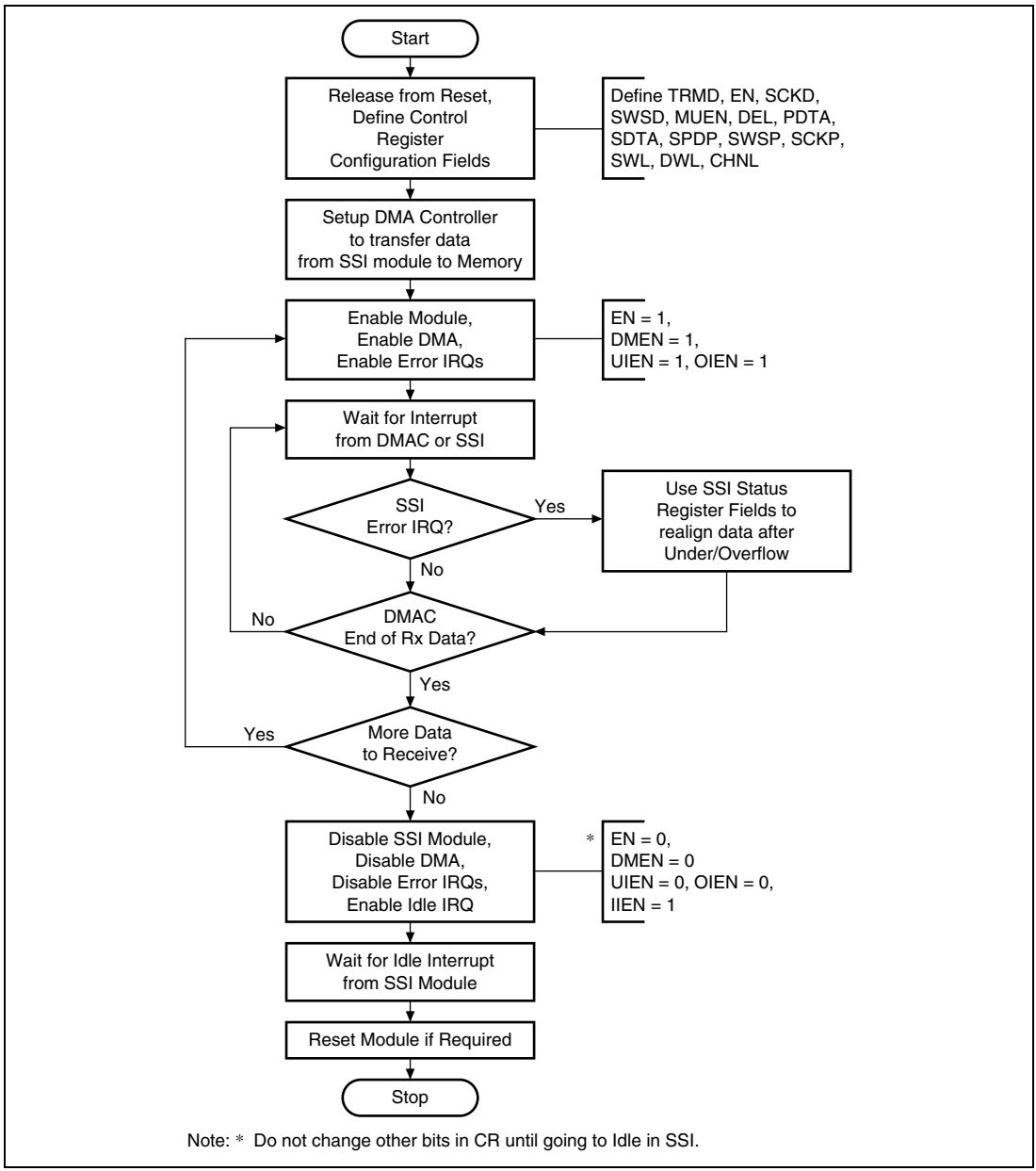


Figure 14.23 Reception using DMA Transfer

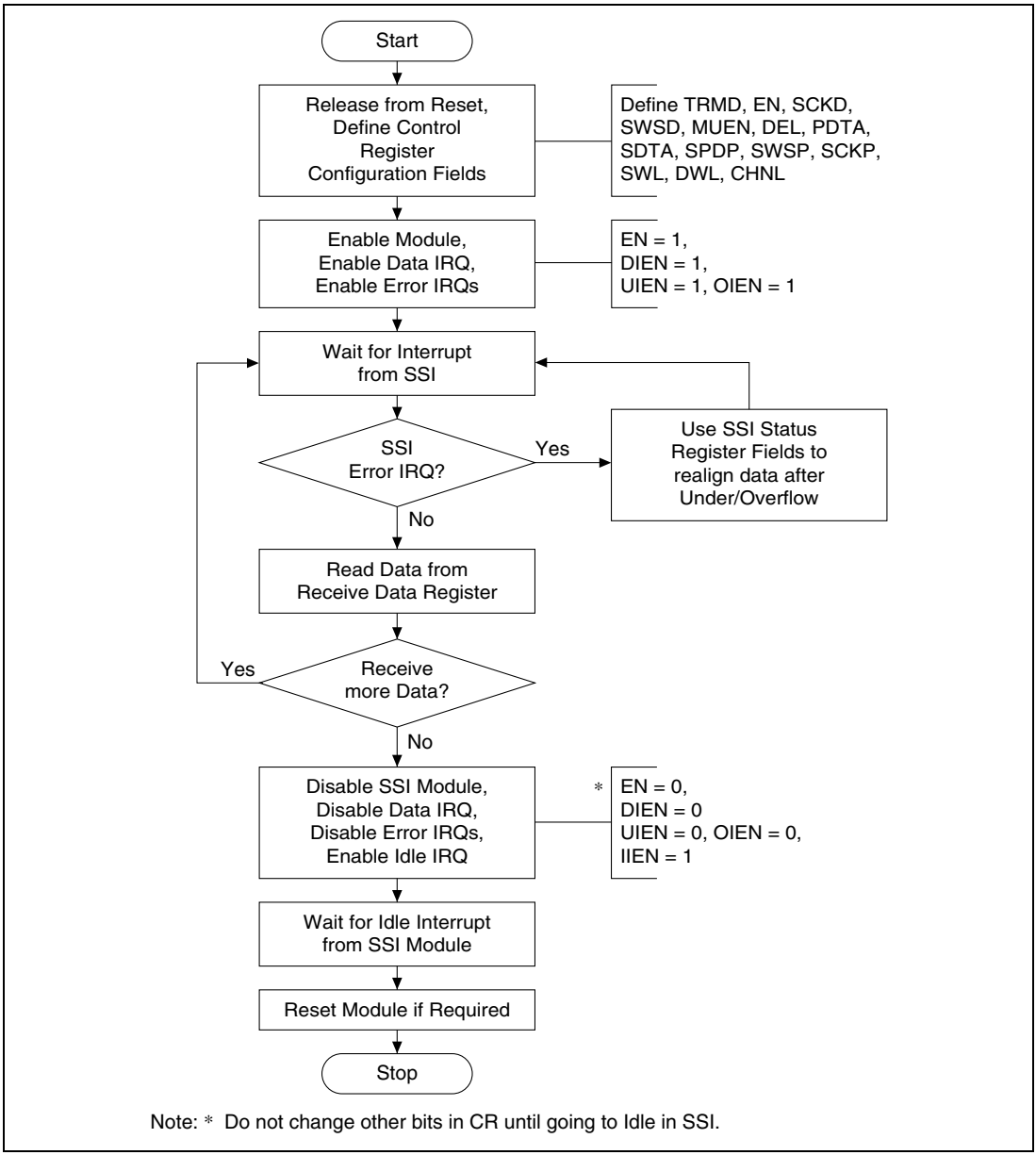


Figure 14.24 Reception using IRQ Flow Control

In the event of an underflow or overflow error condition, the Channel Number (CHNO) and System Word Number (SWNO) can be used to recover to a known status. When an underflow or overflow occurs, the host can read the Channel Number and System Word Number to determine what point the serial audio stream has reached. In the transmitter case, the host can skip forward through the data it wants to transmit until it finds the sample data that matches what the SSI module is expecting to transmit next, and so resynchronize with the audio data stream. In the receiver case the host can skip forward storing null sample data until it is ready to store the sample data that the SSI module is indicating will be received next, and so resynchronize with the audio data stream.

14.6 Functional Description

The complete functionality is described by the following sub-functions:

- Register Bus Interface
- Buffer and Shift Register
- Control (including Bit Counter)
- Serial Clock Control

The relationships between the sub-functions are shown below:

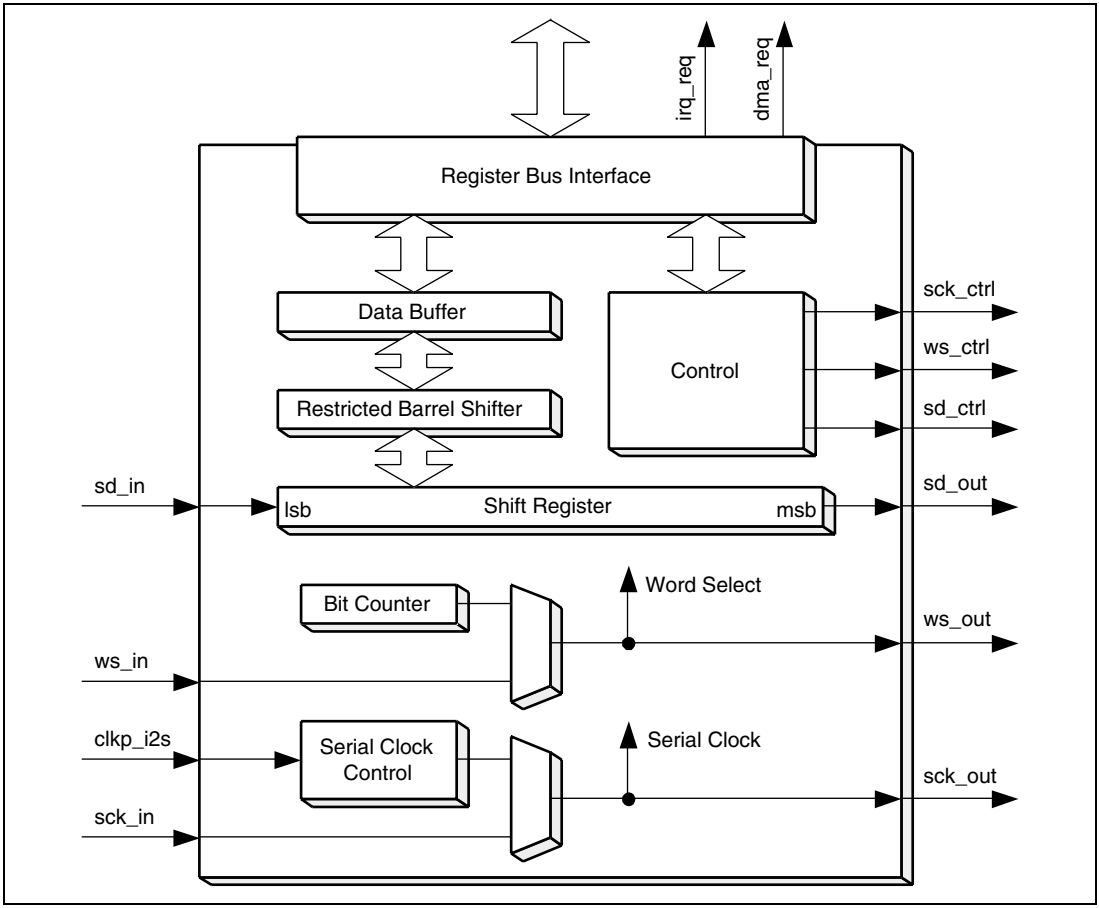


Figure 14.25 Functional Relationships

14.6.1 Register Bus Interface and Control

This Interface is used for control, status and data flow to and from the processor.

It holds the Control and Status Registers accessible by the Register Bus.

It has additional facilities for IRQ and DMA control signals.

14.6.2 Buffer and Shift Register

This function is the main method for transferring from the parallel and serial domains. The Buffer operates from the same clock as the Register Bus Interface. The Shift Register operates from the same clock as derived by the Serial Clock Control Function. These clocks can and will be asynchronous to each other in normal mode so the module will have to deal with asynchronous transfers, between the registers.

14.6.3 Control (including Bit Counter)

The Control Logic has the state machine controlling the Serial Bus transfers.

The Bit Counter is used to feed the state machine with the current Serial Data Bus bit count.

14.6.4 Serial Clock Control

This function is used to control and select which clock is used for the Serial Bus interface.

If the Serial Clock Direction is set to input (SCKD = 0), the SSI Module is in Clock Slave Mode, then the Bit Clock that is used in the Shift Register is the module pin, sck_in.

If the Serial Clock Direction is set to output (SCKD = 1), the SSI Module is in Clock Master Mode, and the bit clock is derived from the Module Input Pin, *clkp_i2s*. This Input Clock is then divided by the ratio in the Oversampling Clock Divide Ratio (CKDV) Configuration Field and used as the bit clock in the Shift Register.

In either case the module pin, sck_out, is the same as the Bit Clock.

14.7 Power Saving and Clocking Strategy

The Register Bus Interface circuitry is clocked from the Register Bus Clock. The Serial Bus Logic including the Shift Register is clocked by either the clock from the Pin or from the module clock pin, *clkp_i2s*.

The SSI module allows clock gating on the register bus clock to reduce power consumption. Standby mode can be enabled/disabled by controlling the SSI0, SSI1, SSI2 and SSI3 bits in the Clock Control 1 (CC1) Register in the Power and Control module.

To wake up the module, the SSI0, SSI1, SSI2 and/or SSI3 bits in the clock_control_1 (CC1) register should be enabled. After enabling this bit, all accesses to the SSI module are possible.

To power down the module, the following procedure should be followed.

1. Ensure all data transfers have taken place. Disable DMA requests and all IRQ event except for the in idle IRQ. Disable the SSI module.
2. Wait for the in idle IRQ event.
3. Disable appropriate SSI bit in Clock Control 1 (CC1) Register.

14.8 References

1. Philips format Specification, Philips Semiconductors, and Revised: June 5, 1996

Section 15 Hitachi SPDIF Interface

15.1 Overview

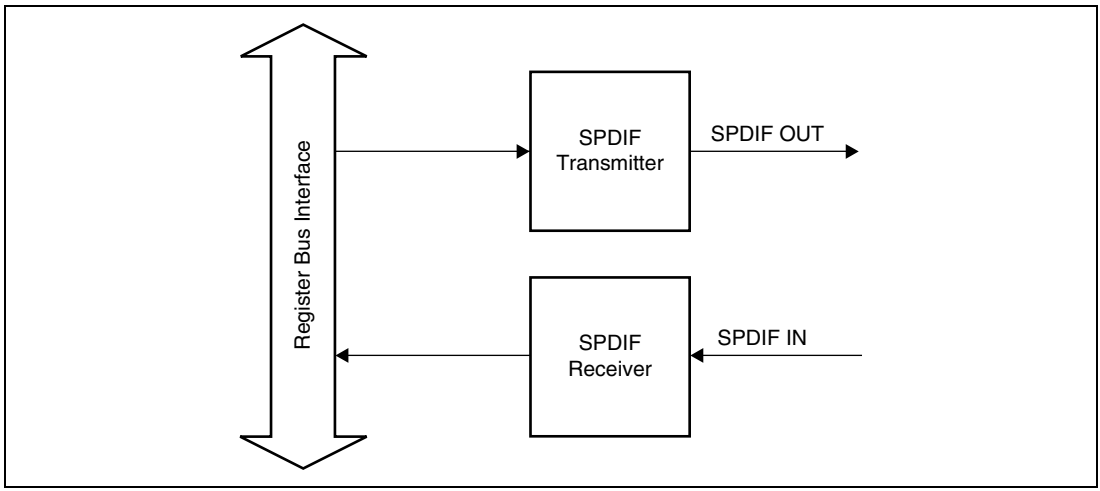


Figure 15.1 Overview Block Diagram

15.2 Features

- Supports the IEC 60958 communications standard (Stereo and Consumer use modes only).
- Supports sampling frequencies of 32kHZ, 44.1kHZ and 48kHz.
- Supports audio word sizes of 16 to 24 bits per sample.
- Biphase mark encoding for zero DC offset.
- Interfaces with Register Bus.
- Double buffered data.
- Parity encoded serial data.
- Simultaneous transmit and receive
- Receiver autodetects IEC 61937 compressed mode data

15.3 Functional Block Diagram

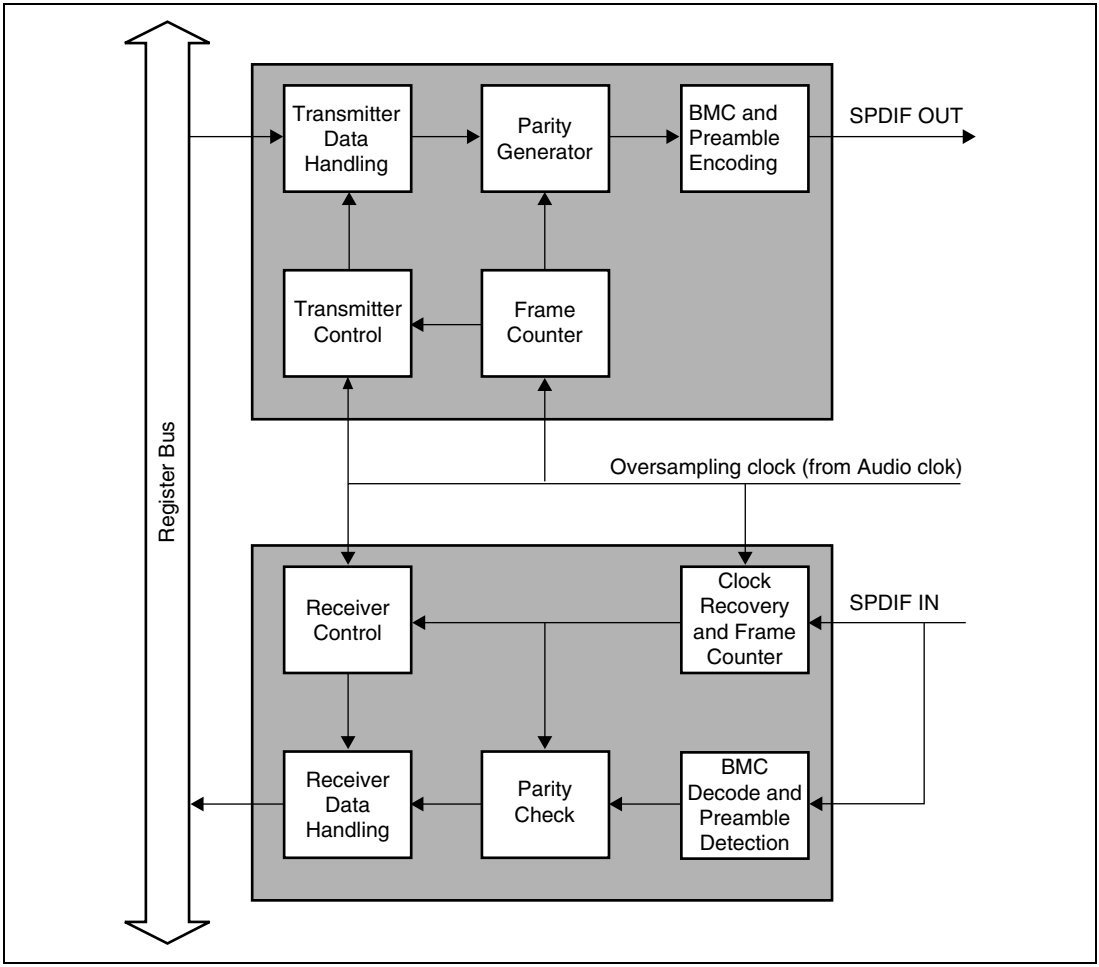


Figure 15.2 Functional Block Diagram

15.4 Pin Description

15.4.1 Processor Interface Pins

Table 15.1 Processor Interface Pins

Pin Name	Direction	Description
SPDIF IN	in	Transmitter BMC encoded spdif bitstream
SPDIF OUT	out	Receiver BMC encoded spdif bitstream

15.5 SPDIF (IEC60958) Block Format

SPDIF blocks contain 192 frames. Each frame contains two subframes – left channel and right channel. The subframe consists of 4 bits preamble, 24 bits data, a validity flag, a user bit, a channel status bit and an even parity bit. Figure 15.3 shows the subframe format. SPDIF is BMC encoded for zero DC offset.

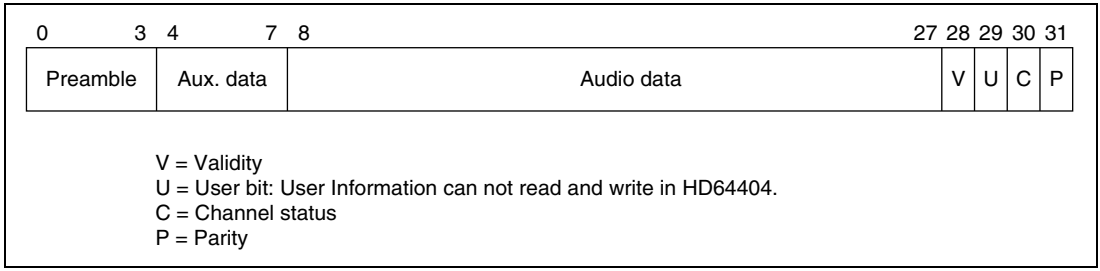


Figure 15.3 Subframe Format

The starting preamble for each subframe depends upon whether it is left or right channel information or the start of a new block. Figure 15.4 shows the block format.

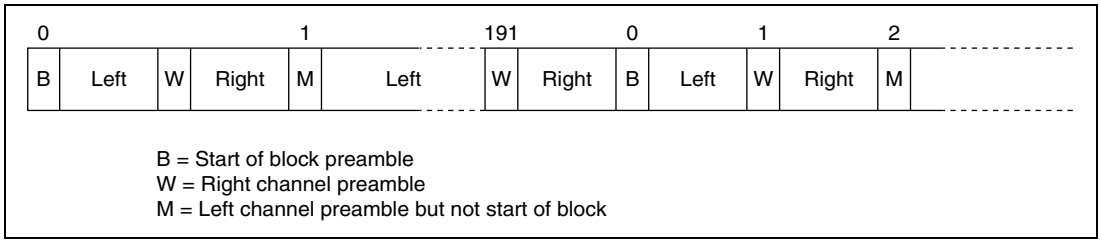


Figure 15.4 Block Format

The table shows the binary preamble values.

Table 15.2 Binary Preamble Values

Preamble	Last BMC State = 0	Last BMC State = 1
B	11101000	00010111
M	11100010	00011101
W	11100100	00011011

Note: Due to the even parity bit only one of the above preamble types is used in any one transmission. However both sets need to be decodable because a polarity reversal may occur in the connection.

Channel status information is encoded at one bit per subframe, this totals 192 bits of channel status information per block for each subframe. For channel status block format please refer to the IEC 60958 standard.

15.6 Register Map

Table 15.3 Register Map

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6240	Transmitter Left Channel Audio	TLCA	32
H'6244	Transmitter Right channel Audio	TRCA	32
H'6248	Transmitter Left Channel Status	TLCS	32
H'624C	Transmitter Right Channel Status	TRCS	32
H'6250	Reserve Register	—	32
H'6254	Receiver Left Channel Audio	RLCA	32
H'6258	Receiver Right Channel Audio	RRCA	32
H'625C	Receiver Left Channel Status	RLCS	32
H'6260	Receiver Right Channel Status	RRCS	32
H'6264	Reserve Register	—	32
H'6268	Control	CTRL	32
H'626C	Status	STAT	32
H'6270	Transmitter DMA Audio Data	TDAD	32
H'6274	Receiver DMA Audio Data	RDAD	32

Note: All registers are longword registers and must be accessed as such.
A register diagram containing a 0 indicates that 0 must be written to this bit (if the register is writeable) and that a 0 will be returned when read (if readable).

15.7 Register Descriptions

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

15.7.1 Control Register (CTRL)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						PB	RASS		TASS		RDE	TDE	NCSI	AOS	RME	TME
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REIE	TEIE	UBOI	UBUI	CREI	PAEI	PREI	CSEI	ABOI	ABUI	RUII	TUII	RCSI	RCBI	TCSI	TCBI
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	0	R	**Reserved**
26	PB	0	R/W	**Pass Back (PB)** Passes transmitter SPDIF OUT into SPDIF Receiver in SPDIF module 0: Pass Back disabled 1: Pass Back enabled
25, 24	RASS	0	R/W	**Receiver Audio Sample Bit Size (RASS)** Indicates the receiver audio sample bit size (16, 20 or 24 bits), for data alignment purposes. 00: 16-bit sample 01: 20-bit sample 10: 24-bit sample 11: Reserved

Bit	Bit Name	Initial Value	R/W	Description
23, 22	TASS	0	R/W	**Transmitter Audio Sample Bit Size (TASS)** Indicates the transmitter audio sample bit size (16, 20 or 24 bits), for data alignment purposes. 00: 16-bit sample 01: 20-bit sample 10: 24-bit sample 11: Reserved
21	RDE	0	R/W	**Receiver DMA Enable (RDE)** Enables DMA requests for the receiver. 0: Receiver DMA disabled 1: Receiver DMA enabled
20	TDE	0	R/W	**Transmitter DMA Enable (TDE)** Enables the DMA requests for the transmitter. 0: Transmitter DMA disabled 1: Transmitter DMA enabled
19	NCSI	0	R/W	**New Channel Status Information (NCSI)** This bit is set when there is new channel status information for the transmitter to process. 0: New channel status information not available 1: New channel status information available
18	AOS	0	R/W	**Audio Only Samples (AOS)** Cleared if audio left and right channel registers contain user information, else all user bits are set to zero. 0: User information present 1: User information not present
17	RME	0	R/W	**Receiver module enable (RME)** Enables the receiver module. 0: Receiver module disabled 1: Receiver module enabled
16	TME	0	R/W	**Transmitter module enable (TME)** Enables the transmitter module. 0: Transmitter module disabled 1: Transmitter module enabled

Bit	Bit Name	Initial Value	R/W	Description
15	REIE	0	R/W	**Receiver error interrupt enable (REIE)** When cleared this bit masks all receiver error interrupts. When set all receiver error interrupts is unmasked. 0: Receiver error interrupts disabled 1: Receiver error interrupts enabled
14	TEIE	0	R/W	**Transmitter error interrupt enable (TEIE)** When cleared this bit masks all transmitter error interrupts. When set all transmitter error interrupts is unmasked. 0: Transmitter error interrupts disabled 1: Transmitter error interrupts enabled
13	UBOI	0	R/W	**User buffer overrun interrupt enable (UBOI)** Enables the user buffers overrun interrupts. 0: User buffer overrun interrupt disabled 1: User buffer overrun interrupt enabled
12	UBUI	0	R/W	**User buffer underrun interrupt enable (UBUI)** Enables the user buffers underrun interrupts. 0: User buffer underrun interrupt disabled 1: User buffer underrun interrupt enabled
11	CREI	0	R/W	**Clock recovery error interrupt enable (CREI)** Enables the clock recovery error interrupt. 0: Clock recovery error interrupt disabled 1: Clock recovery error interrupt enabled
10	PAEI	0	R/W	**Parity error interrupt enable (PAEI)** Enables the parity check error interrupt. 0: Parity check error interrupt disabled 1: Parity check error interrupt enabled
9	PREI	0	R/W	**Preamble error interrupt enable (PREI)** Enables the preamble check error interrupt. 0: Preamble error interrupt disabled 1: Preamble error interrupt enabled
8	CSEI	0	R/W	**Channel status error interrupt enable (CSEI)** Enables the channel status error interrupt. 0: Channel status error interrupt disabled 1: Channel status error interrupt enabled

Bit	Bit Name	Initial Value	R/W	Description
7	ABOI	0	R/W	**Audio Buffer Overrun interrupt enable (ABOI)** Enables the receiver audio buffer overrun interrupt. 0: Audio buffer overrun interrupt disabled 1: Audio buffer overrun interrupt enabled
6	ABUI	0	R/W	**Audio Buffer Underrun interrupt enable (ABUI)** Enables the transmitter audio buffer underrun interrupt. 0: Audio buffer underrun interrupt disabled 1: Audio buffer underrun interrupt enabled
5	RUII	0	R/W	**Receiver user information interrupt enable (RUII)** Enables the receiver user information register full interrupt. 0: Receiver user information interrupt is disabled 1: Receiver user information interrupt is enabled
4	TUII	0	R/W	**Transmitter user information interrupt enable (TUII)** Enables the transmitter user information register empty interrupt. 0: Transmitter user information interrupt is disabled 1: Transmitter user information interrupt is enabled
3	RCSI	0	R/W	**Receiver Channel Status interrupt enable (RCSI)** Enables the receiver channel status register empty interrupt. 0: Receiver channel status interrupt is disabled 1: Receiver channel status interrupt is enabled
2	RCBI	0	R/W	**Receiver Channel Buffer interrupt enable (RCBI)** Enables the receiver audio channel buffer empty interrupt. 0: Receiver audio channel interrupt is disabled 1: Receiver audio channel interrupt is enabled

Bit	Bit Name	Initial Value	R/W	Description
1	TCSI	0	R/W	**Transmitter Channel Status interrupt enable (TCSI)** Enables the transmitter channel status register empty interrupt. 0: Transmitter channel status interrupt is disabled 1: Transmitter channel status interrupt is enabled
0	TCBI	0	R/W	**Transmitter Channel Buffer interrupt enable (TCBI)** Enables the transmitter audio channel buffer empty interrupt. 0: Transmitter audio channel interrupt is disabled 1: Transmitter audio channel interrupt is enabled

15.7.2 Status Register (STAT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
																CMD
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RIS	TIS	UBO	UBU	CE	PARE	PREE	CSE	ABO	ABU	RUIR	TUIR	CSRX	CBRX	CSTX	CBTX
Initial:	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
R/W	R	R	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	0	R	**Reserved**
16	CMD	0	R	**Compressed Mode Data (CMD)** Sets if the data being received is compressed mode data (see IEC61937). 0: Data is not compressed mode 1: Data is compressed mode
15	RIS	1	R	**Receiver idle state (RIS)** Sets if the receiver is in the idle state. 0: Receiver is not in idle state 1: Receiver in idle state
14	TIS	1	R	**Transmitter Idle State (TIS)** Sets if the transmitter is in the idle state. 0: Transmitter is not in idle state 1: Transmitter is in idle state
13	UBO	0	R/WC0	**User Buffer Overrun (UBO)*** Sets if the receiver user buffer overruns. This bit is cleared by writing 0 to the register. If bit REIE and bit UBOI of the Control Register are set this causes an interrupt. 0: User buffer has not overrun 1: User buffer has overrun
12	UBU	0	R/WC0	**User Buffer Underrun (UBU)*** Sets if the transmitter user buffer underruns. This bit is cleared by writing 0 to the register. If bit TEIE and bit UBUI of the Control Register are set this causes an interrupt. 0: User buffer has not underrun 1: User buffer has underrun
11	CE	0	R/WC0	**Clock error (CE)*** Sets when the clock recovery falls out of synchronisation. This is cleared by writing 0 to this bit. If bit REIE and bit CREI of the Control Register are set this causes an interrupt. 0: Clock recovery stable 1: Clock recovery error

Bit	Bit Name	Initial Value	R/W	Description
10	PARE	0	R/WC0	**Parity error (PARE)***
				Sets when the parity checker produces a fail result. This is cleared by writing 0 to this bit. If bit REIE and bit PAEI of the Control Register are set this causes an interrupt.
				0: Parity check correct 1: Parity error
9	PREE	0	R/WC0	**Preamble error (PREE)***
				Sets when the start of word preamble fails to appear in the correct place. This is cleared by writing 0 to this bit. If bit REIE and bit PREI of the control register are set this causes an interrupt.
				Note: Only set after a start of block preamble has occurred.
				0: Preamble present 1: Preamble error
8	CSE	0	R/WC0	**Channel Status Error (CSE)***
				Sets when the channel status information is written before the 32nd frame of the current block. This is cleared by writing 0 to this bit. If bit TEIE and bit CSEI of the Control Register is set this causes an interrupt.
				0: Channel status correct 1: Channel status error
7	ABO	1	R/WC0	**Audio Buffer Overrun (ABO)***
				Indicates that the receiver audio buffer is full in both the first and second stages and that data has been overwritten. This bit is cleared by writing zero to this bit. If bit REIE and bit ABOI of the Control Register is set then this causes an interrupt.
				0: Receiver audio buffer not overrun
				1: Receiver audio buffer overrun
6	ABU	1	R/WC0	**Audio Buffer Underrun (ABU)***
				Indicates that the transmitter audio buffer is empty in both the first and second stages and that the last data transmission has been repeated. This bit is cleared by writing zero to this bit. If bit TEIE and bit ABUI the Control Register is set then this causes an interrupt.
				0: Transmitter audio buffer not underrun 1: Transmitter audio buffer underrun

Bit	Bit Name	Initial Value	R/W	Description
5	RUIR	0	R	**Receiver User Information Register (RUIR)** Indicates the status of the Receiver User Information Register. This bit is cleared by reading from the Receiver User Register. If bit RUII of the Control Register is set then this causes an interrupt. 0: Receiver User Information Register is empty 1: Receiver User Information Register is full
4	TUIR	0	R	**Transmitter User Information Register (TUIR)** Indicates the status of the Transmitter User Information Register. This bit is cleared by writing to the Transmitter User Register. If bit TUIR of the Control Register is set then this causes an interrupt. 0: Transmitter User Information Register is full 1: Transmitter User Information Register is empty
3	CSRX	0	R	**Left and Right Channel Status—Receiver (CSRX)** Indicates the status of the Receiver Channel Status Registers. This bit is cleared by reading from the Receiver Channel Status Registers. If bit RCSI of the Control Register is set this causes an interrupt. 0: Receiver Channel Status Registers are empty 1: Receiver Channel Status Registers are full
2	CBRX	0	R	**Left and Right Channel Buffers – Receiver (CBRX)** Indicates the status of the Receiver Audio Channel Registers. This bit is cleared by reading from the Receiver Audio Channel Registers. If bit RCBI of the Control Register is set this causes an interrupt. This bit is output as rbdmareq_rx. 0: Receiver Audio Channel Registers are empty 1: Receiver Audio Channel Registers are full

Bit	Bit Name	Initial Value	R/W	Description
1	CSTX	0	R	**Left and Right Channel Status—Transmitter (CSTX)** Indicates the status of the Transmitter Channel Status Registers. This bit is cleared by writing to the Transmitter Channel Status Registers. If bit TCSI of the Control Register is set this causes an interrupt. 0: Transmitter Channel Status Register is full 1: Transmitter Channel Status Register is empty
0	CBTX	0	R	**Left and Right Channel Buffers—Transmitter (CBTX)** Indicates the status of the Transmitter Audio Channel Registers. This bit is cleared by writing to the Transmitter Audio Channel Registers. If bit TCBI of the Control Register is set this causes an interrupt. This bit is output as rbdmareq_tx. 0: Transmitter Audio Channel Registers are full 1: Transmitter Audio Channel Registers are empty

Note: * When those error bits are detected during DMA transfer, the equivalent DMA channel in DMAC must be aborted. After that, module Enable bit, either RME or TME, and DMA Enable bit, either RDE or TDE, must be disabled and the error status must be cleared. Then the module enable can be set and DMA can be initiated again.

15.7.3 Transmitter Left Channel Audio Register (TLCA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									Audio PCM data							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Audio PCM data															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	W	**Reserved**
23 to 0	Audio PCM data	0	W	**Audio PCM data** LSB aligned PCM encoded audio data.

15.7.4 Transmitter Right Channel Audio Register (TRCA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									Audio PCM data							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Audio PCM data															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	W	**Reserved**
23 to 0	Audio PCM data	0	W	**Audio PCM data** LSB aligned PCM encoded audio data.

15.7.5 Transmitter DMA Audio Data Register (TDAD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									Audio PCM data							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Audio PCM data															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	W	**Reserved**
23 to 0	Audio PCM data	0	W	**Audio PCM data** LSB aligned PCM encoded audio data.

15.7.6 Reseve Register

For the contents of the user bytes please refer to the appropriate standard for the device in use.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved								Reserved							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								Reserved							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	Reserved	0	W	Write data is unknown
23 to 16	Reserved	0	W	Write data is unknown
15 to 8	Reserved	0	W	Write data is unknown
7to 0	Reserved	0	W	Write data is unknown

15.7.7 Transmitter Left Channel Status Register (TLCS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			Clock Acc		FS				Channel Number				Source Number			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Category code								0	0	Control					0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	W	**Reserved**
29, 28	Clock Acc	0	W	**Clock Accuracy** 00: Level II 01: Level III 10: Level I 11: Reserved
27 to 24	FS	0	W	**Sample Frequency (FS)** 0000: 44.1kHz 0100: 48 kHz 1100: 32 kHz
23 to 20	Channel Number	0	W	**Channel Number** 0000: Don't care 1000: A (left channel) 0100: B (right-channel) 1100: C
19 to 16	Source Number	0	W	**Source Number** 0000: Don't care 1000: 1 0100: 2 1100: 3
15 to 8	Category code	0	W	**Category code** 00000000: 2-channel general format 10000000: 2-channel Compact Disc (IEC 908) 01000000: 2-channel PCM encoder/decoder 11000000: 2-channel Digital Audio Tape Recorder
7	0	0	W	
6	0	0	W	
5 to 1	Control	0	W	**Control** The control bits are copied from the source (see IEC60958 standard)
0	0	0	W	

15.7.8 Transmitter Right Channel Status Register (TRCS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			Clock Acc		FS				Channel Number				Source Number			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Category code								0	0	Control					0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	W	**Reserved**
29, 28	Clock Acc	0	W	**Clock Accuracy** 00: Level II 01: Level III 10: Level I 11: Reserved
27 to 24	FS	0	W	**Sample Frequency (FS)** 0000: 44.1kHz 0100: 48 kHz 1100: 32 kHz
23 to 20	Channel Number	0	W	**Channel Number** 0000: Don't care 1000: A (left-channel) 0100: B (right-channel) 1100: C
19 to 16	Source Number	0	W	**Source Number** 0000: Don't care 1000: 1 0100: 2 1100: 3
15 to 8	Category code	0	W	**Category code** 00000000: 2-channel general format 10000000: 2-channel Compact Disc (IEC 908) 01000000: 2-channel PCM encoder/decoder 11000000: 2-channel Digital Audio Tape Recorder

Bit	Bit Name	Initial Value	R/W	Description
7	0	0	W	
6	0	0	W	
5 to 1	Control	0	W	**Control** The control bits are copied from the source (see IEC60958 standard)
0	0	0	W	

15.7.9 Receiver Left Channel Audio Register (RLCA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									Audio PCM data							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Audio PCM data															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	**Reserved**
23 to 0	Audio PCM data	0	R	**Audio PCM data** LSB aligned PCM encoded audio data.

15.7.10 Receiver Right Channel Audio Register (RRCA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									Audio PCM data							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Audio PCM data															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	**Reserved**
23 to 0	Audio PCM data	0	R	**Audio PCM data** LSB aligned PCM encoded audio data.

15.7.11 Receiver DMA Audio Data (RDAD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									Audio PCM data							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Audio PCM data															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	—	R	**Reserved**
23 to 0	Audio PCM data	0	R	**Audio PCM data** LSB aligned PCM encoded audio data.

15.7.12 Reserve Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Reserved								Reserved							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved								Reserved							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	Reserved	0	R	Read data is unknown
23 to 16	Reserved	0	R	Read data is unknown
15 to 8	Reserved	0	R	Read data is unknown
7to 0	Reserved	0	R	Read data is unknown

15.7.13 Receiver Left Channel Status Register (RLCS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			Clock Acc		FS				Channel Number				Source Number			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Category code								0	0	Control					0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	**Reserved**
29, 28	Clock Acc	0	R	**Clock Accuracy** 00: Level II 01: Level III 10: Level I 11: Reserved
27 to 24	FS	0	R	**Sample Frequency (fs)** 0000: 44.1kHz 0100: 48 kHz 1100: 32 kHz
23 to 20	Channel Number	0	R	**Channel Number** 0000: Don't care 1000: A (left-channel) 0100: B (right-channel) 1100: C

Bit	Bit Name	Initial Value	R/W	Description
19 to 16	Source Number	0	R	**Source Number** 0000: Don't care 1000: 1 0100: 2 1100: 3
15 to 8	Category code	0	R	**Category code** 00000000: 2-channel general format 10000000: 2-channel Compact Disc (IEC 908) 01000000: 2-channel PCM encoder/decoder 11000000: 2-channel Digital Audio Tape Recorder
7	0	0	R	
6	0	0	R	
5 to 1	Control	0	R	**Control** The control bits are copied from the source (see IEC60958 standard)
0	0	0	R	

15.7.14 Receiver Right Channel Status Register (RRCS)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			Clock Acc		FS				Channel Number				Source Number			
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Category code								0	0	Control					0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	**Reserved**
29, 28	Clock Acc	0	R	**Clock Accuracy** 00: Level II 01: Level III 10: Level I 11: Reserved
27 to 24	FS	0	R	**Sample Frequency (fs)** 0000: 44.1kHz 0100: 48 kHz 1100: 32 kHz
23 to 20	Channel Number	0	R	**Channel Number** 0000: Don't care 1000: A (left-channel) 0100: B (right-channel) 1100: C
19 to 16	Source Number	0	R	**Source Number** 0000: Don't care 1000: 1 0100: 2 1100: 3
15 to 8	Category code	0	R	**Category code** 00000000: 2-channel general format 10000000: 2-channel Compact Disc (IEC 908) 01000000: 2-channel PCM encoder/decoder 11000000: 2-channel Digital Audio Tape Recorder
7	0	0	R	
6	0	0	R	
5 to 1	Control	0	R	**Control** The control bits are copied from the source (see IEC60958 standard)
0	0	0	R	

15.8 Functional Description—Transmitter

15.8.1 Module Interface

The SPDIF module interface supports the Hitachi Register Bus protocol for 32-bit interfaces.

15.8.2 Transmitter Module

The transmitter module is designed to produce IEC 60958 (SPDIF) encoded 2-channel PCM data from a variety of sources.

The main clock for the transmitter module is the oversampling clock (supplied from Audio Clock). This clock oversamples at a rate of 4 times the clock frequency required for BMC encoding. The clock frequency required for transmission is 128 times the audio data sample frequency.

Audio data and channel status information are written to the module left channel and then right channel. Channel status need only be written when the information changes, and will only be requested after 30 frames (when all the current channel status data has been transmitted) and must be received before the start of the next block i.e. 192 frames.

The audio data is stored in a double buffer arrangement. Either an interrupt request is sent, or the status register can be polled, to indicate when the first stage buffer is empty. DMA transfers are sent left channel audio data on the first request and right channel data on the second.

The channel status information is stored in a 30-bit register. Channel status information consists of 192 bits per frame, as there are only 30 bits of data after the first 30 bits have been sent zeroes are generated to produce the correct number of bits per frame.

The audio data consists of up to 24 bits audio data. The validity bit is always set to zero for the audio data.

Even parity is generated every 32 bits of the serial stream but does not include the preambles. The serial stream is then BMC encoded at a 128 × fs rate and the preamble is written prior to the start of each word.

Note: When transmitter user buffer underrun occurs, the current data in the buffer data of SPDIF is transmitted until the next data is filled.

15.8.3 Transmitter Module Initialisation

The device defaults to an idle state when it comes out of reset, or can be put into an idle state when 0 is written to the TME bit of the CTRL Register. Whilst idle the transmitter module has the following settings:

- The transmitter idle status bit (TIS) is set to 1, all other status bits are 0.
- Preamble generation is disabled.
- Left-right synchronisation is set to 0 (0 for left channel, 1 for right).
- Word_count and frame_count are both 0.
- The output from the BMC encoder is set to 0.
- Channel status, user and audio data registers will retain its value prior to putting the module into idle.

To exit the idle state the user must write 1 to bit (TME) of the CTRL register.

15.8.4 Transmitter Module Data Transfer

Once the transmitter module has left the idle state, it is ready for data transfer. Data transfer timing can be achieved in three ways. Either the transfer is done by interrupts, DMA requests or by polling the status register. There is a shared interrupt line (for both transmit and receive) and a single transmitter DMA request line.

Figure 15.5 illustrates the transmitter data transfer using interrupts.

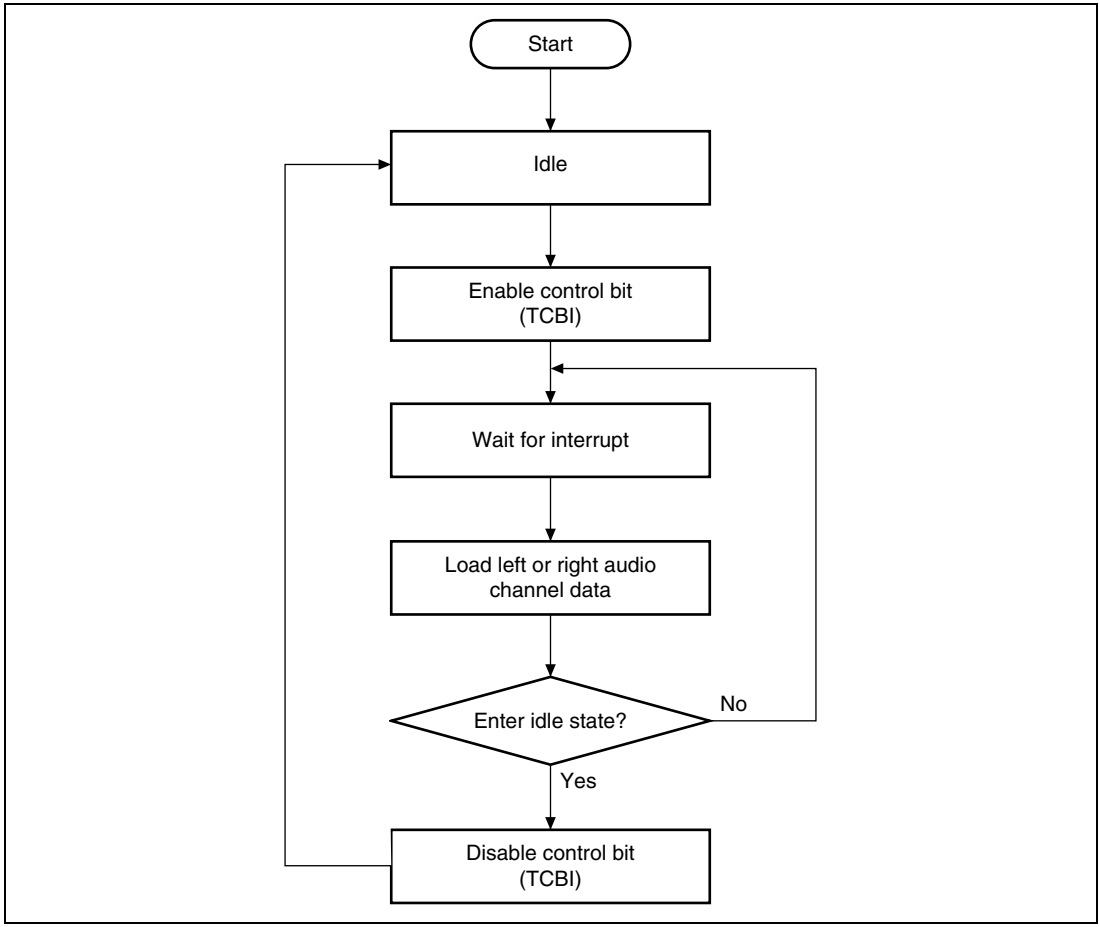


Figure 15.5 Transmitter Data Transfer Flow Diagram - Interrupt Driven

Figure 15.6 illustrates the transmitter data transfer using DMA request.

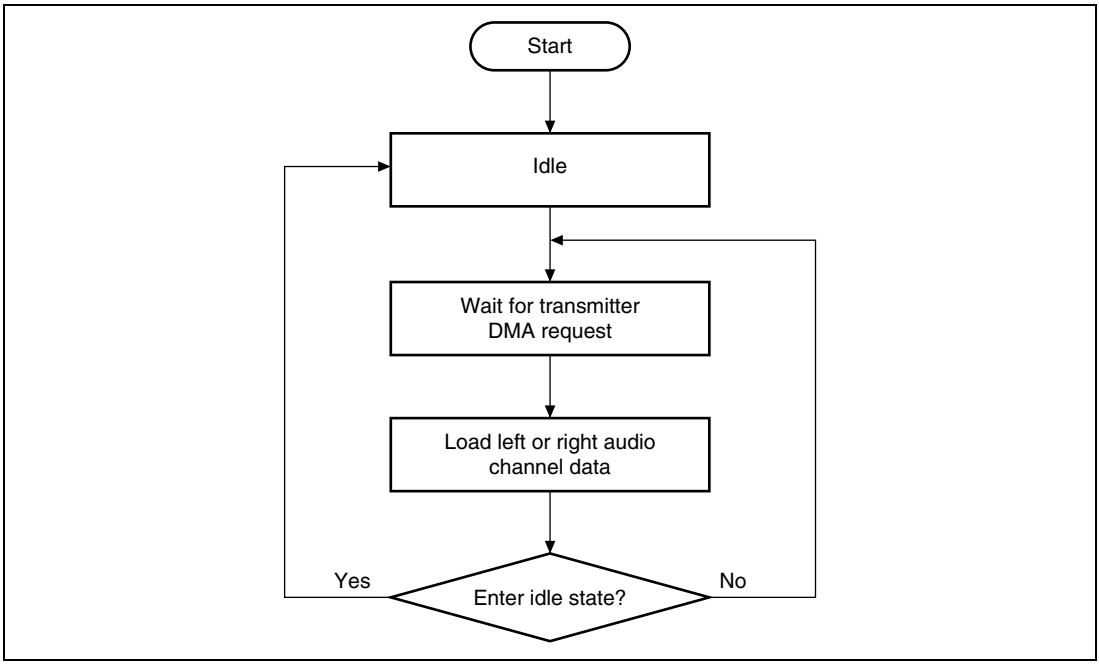


Figure 15.6 Transmitter Data Transfer Flow Diagram—DMA Request Driven

Channel status information is only required to be updated when the information has changed; this needs to be done before the transmission of the next block. New data should be written after 30 frames have been sent; this is indicated either by an interrupt or by polling the status bit. If channel status is written before 30 frames have been sent (whilst current information is being sent) then an interrupt indicates that the channel status error bit (CSE) in the Status Register has been set.

Note: 30 frames contains all the valid information in a single channel status block.

15.9 Functional Description—Receiver

15.9.1 Module Interface

The SPDIF module. The interface supports the Hitachi Register Bus protocol for 32-bit interfaces.

15.9.2 Receiver Module

The receiver module is designed to recover data and clock from an IEC 60958 encoded bitstream. The recovered data is structured as audio PCM data with channel status.

The main clock for the receiver module is the oversampling clock (supplied from Audio Clock). The module runs with a 4 times oversampling clock rate.

Note: The oversampling clock is the same for the transmitter and receiver.

Clock recovery is performed using a pulse width counter and averaging filters to produce a sampling pulse in the middle of each bit in the datastream. A clock error status bit indicates clock synchronisation loss. Synchronisation is achieved when a preamble occurs on the data stream for the first time. Continuous adjustment prevents jitter and/or clock drift from affecting clock recovery, provided that they fall within the IEC 60958 specifications.

Once the clock recovery is successful the BMC decoder initiates its preamble detection. The decoder searches for the start of block preamble (see Table 15.4). A preamble error status bit indicates that following preambles have not appeared at the correct time, such failures are most likely caused by transmission loss or interference.

Even parity checking is performed on the decoded data. A discrepancy will result in the parity error status bit being set.

The data is sorted into audio, user and channel status information. The audio is stored in a double buffer arrangement. Either an interrupt request or polling of the status bit will indicate when the data is ready to be read. DMA transfers receive left channel audio data on the first request and right channel data on the second.

Channel status is stored in a 30-bit register. Channel status information is received at 1-bit per frame, and so the registers will not be full until 30 and 32 frames have been received respectively. New channel status is compared with the current data to see if it has changed and is only read by the processor if it has.

Notes: 1. Channel status data requests do not support DMA.
2. When receiver user buffer overrun occurs, the current data in the buffer data of SPDIF is overwritten by the next incoming data from SPDIF interface.

15.9.3 Receiver Module Initialisation

The device defaults to an idle state when it comes out of reset, or can be put into an idle state by writing 0 to bit RME of the CTRL Register. Whilst idle the module has the following settings:

- The receiver idle status bit is set to 1, all other status bits are 0.
- Left-right synchronisation is set to 0 (0 for left channel, 1 for right).
- Word_count and frame_count are both 0.
- Channel status and audio data registers will retain its value prior to putting the module into idle.

To exit the idle state the user must write 1 to bit (RME) of the CTRL Register.

15.9.4 Receiver Module Data Transfer

Once the module has left the idle state it is ready for data transfer. Data transfer timing can be achieved in three ways. The transfer can be done by interrupts, or by polling the Status Register, or by DMA. There is a shared interrupt line (transmit and receive) and a single receiver DMA request line. Data transfer for the receiver can be interrupted by error signals caused by:

1. Clock recovery failure.
2. Transmission loss or interference – indicated by a preamble error.
3. Parity check failure.

Transmission loss or interference can cause the start of subframe or start of block preamble to be misplaced or not present.

Parity check failure occurs when the parity bit is incorrect, this can be caused by any of the above.

Clock Recovery Deviation

The receive margin for clock recovery is based on the following equation:

Equation 1

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

where M = receive margin
N = oversampling rate
L = frame length = 33
D = duty cycle = 0.6
F = oversampling clock deviation = Level II accuracy = 1000 in $10e^{-6}$

Figure 15.7 indicates what the receive margin M represents

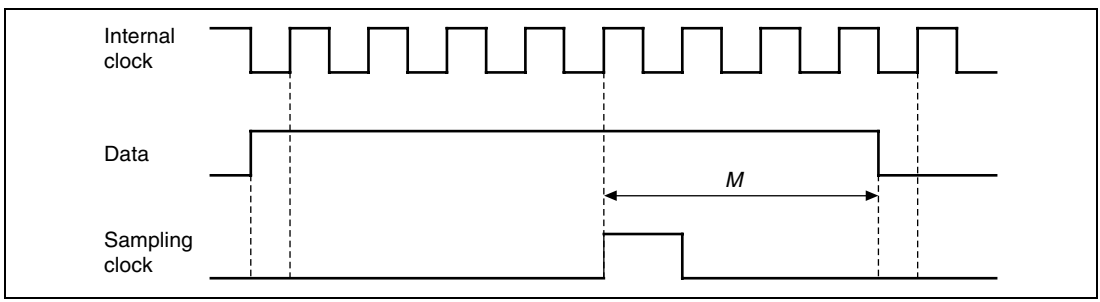


Figure 15.7 Receive Margin

Introducing jitter into the equation gives the following inequality.

Equation 2

$$j \leq \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

J = clock jitter

Eight times oversampling produces a receive margin = 39.25%
Four times oversampling produces a receive margin = 31.75%
Two times oversampling produces a receive margin = 16.75%

The fastest sample frequency is 48kHz. This requires a clock speed of 128 × 48kHz = 6.144MHz. The worst case jitter in one cycle is specified at 40ns = 24.5% of the period. This means that an oversampling rate of 4 or more will satisfy the inequality and therefore be sufficient for clock recovery.

Figure 15.8 illustrates the receiver data transfer using interrupts.

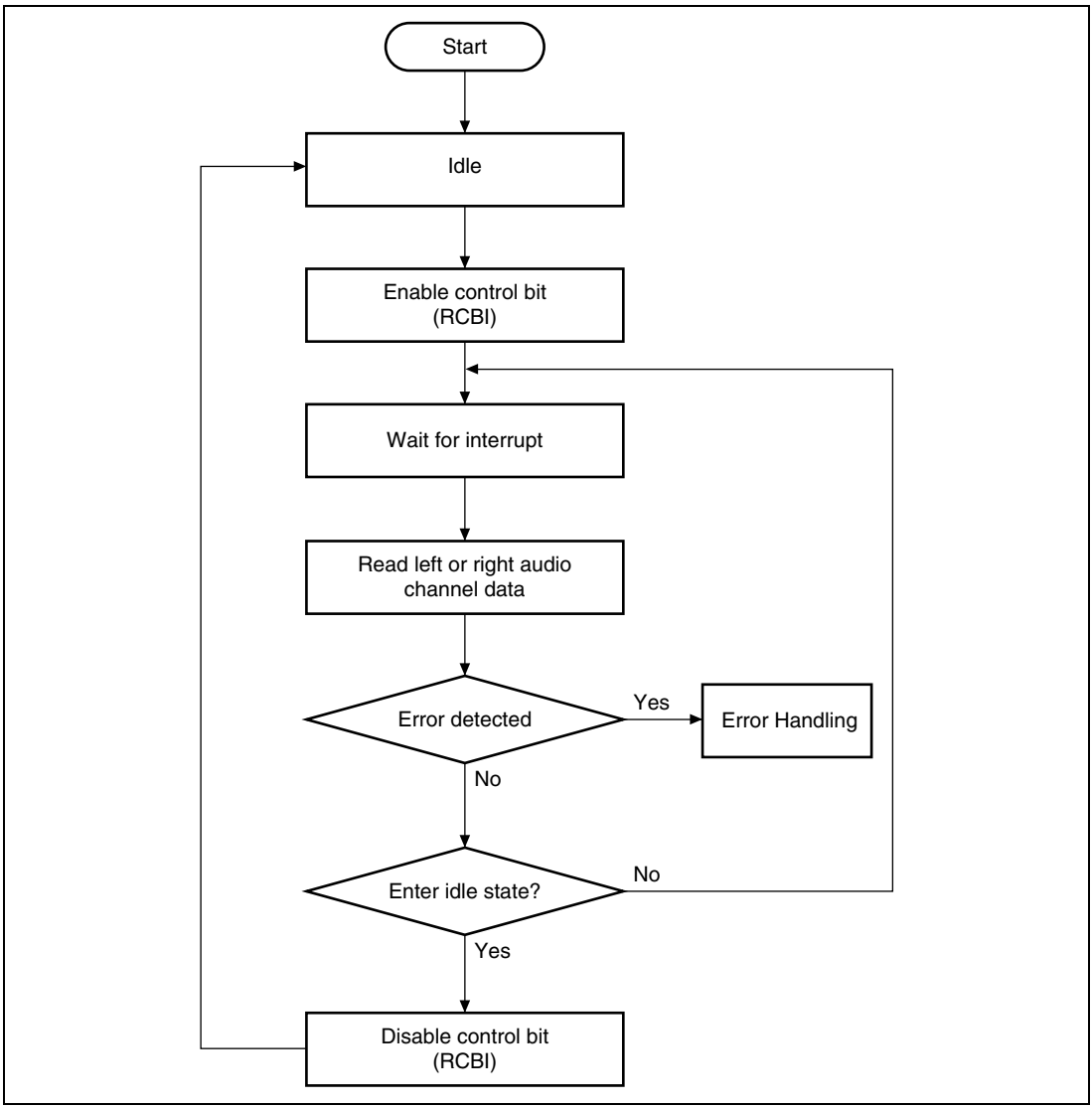


Figure 15.8 Receiver Data transfer Flow Diagram - Interrupt Driven

Interrupts to indicate that the Channel Status Information Register is full occur after frame 30 has been received and only if the information has changed. When the first four bytes have been stored an interrupt occurs.

Figure 15.9 illustrates the transmitter data transfer using DMA request in case rbclk is more than 40MHz.

In the beginning of SPDIF transmit, 0x0000 data must be written to TLCA and TRCA 2 times together with checking CBTX bit of the Status Register. After four Longword 0x0000 data are written, set TDE bit in Control register.

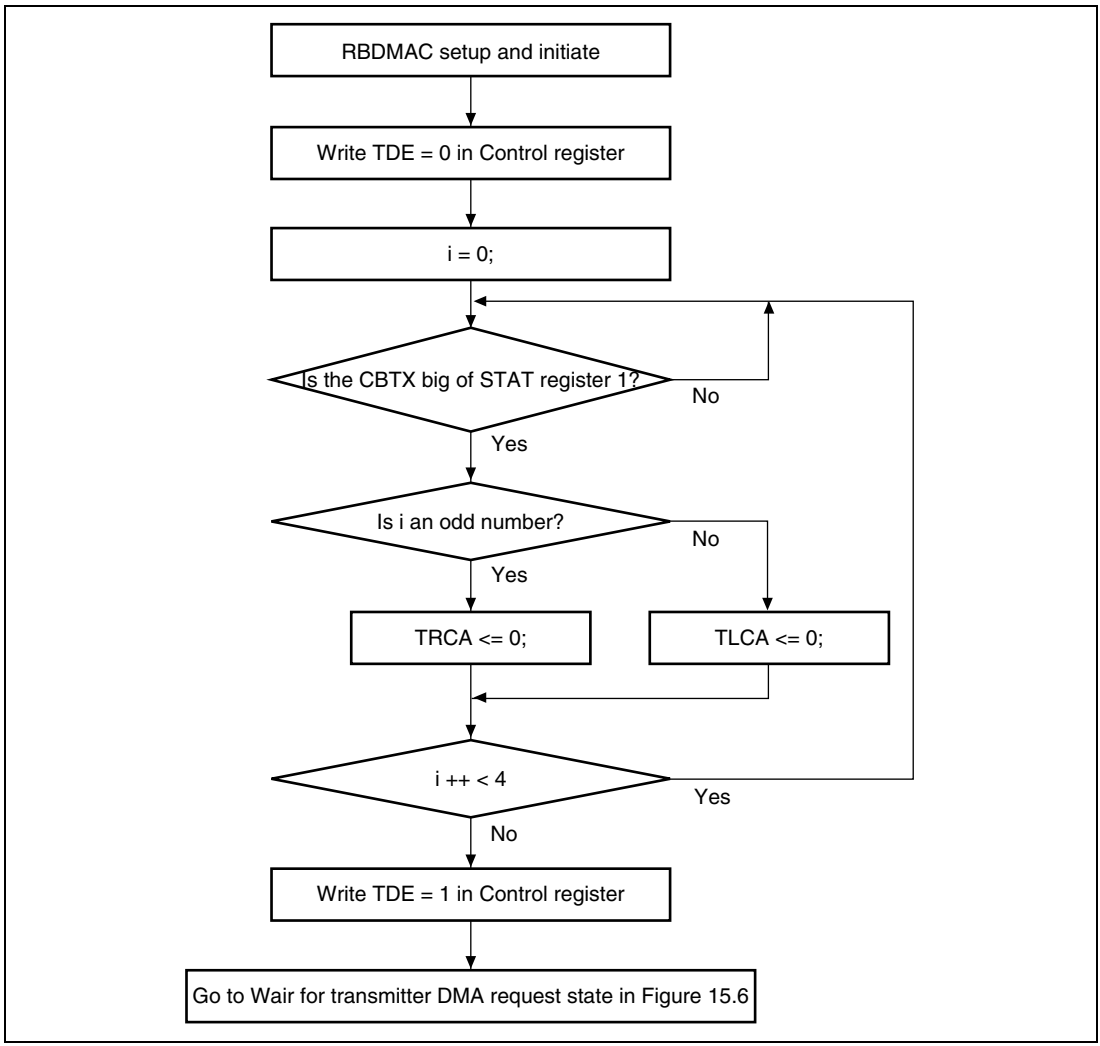


Figure 15.9 Transmitter Data Transfer Flow Diagram – DMA request Driven in case rbclk is more then 40MHz

15.10 Disabling the Module

15.10.1 Transmitter and Receiver Idle

The transmitter or receiver modules can be disabled by writing to the idle bit of the Control Register (TME for the transmitter and RME for the receiver). The idle state can be detected by polling the idle bit of the Status Register (TIS and RIS).

15.10.2 Power Down Mode

The transmitter and receiver modules can be put into a power down mode by using the clock stopping option in the Power Control and Configuration block in the following manner:

1. Disable the SPDIF transmitter or receiver module using the procedure described in the above section.
2. Disable the SPD bit of the Clock Control 1 (CC1) Register of the Power Control and Configuration block of the HD64404 chip.

15.11 Compressed Mode Data

Compressed mode data is defined in the IEC 61937 specification. This module only detects compressed mode data. This is done by checking the validity bit and bit 1 of the channel status data. If both are one then the data is in compressed mode. This is indicated by the setting of the CMD bit in the status register.

Note: Only the receiver detects compressed mode data, as the information is not relevant to the transmitter.

15.12 References

IEC60958 Digital Audio Interface
IEC61937 Compressed Mode Digital Audio Interface

15.13 Glossary

For glossary of terms see IEC60958 and IEC61937.

Section 16 Hitachi I²C Interface

16.1 General Description

This LSI provides an on-chip 2-channel I²C bus interface supporting the Philips (Inter IC Bus) I²C bus interface. It should be noted, however, the register structure used to control the I²C bus differs from that of the Philips implementation.

Figure 16.1 shows a connection example for the I²C bus interface.

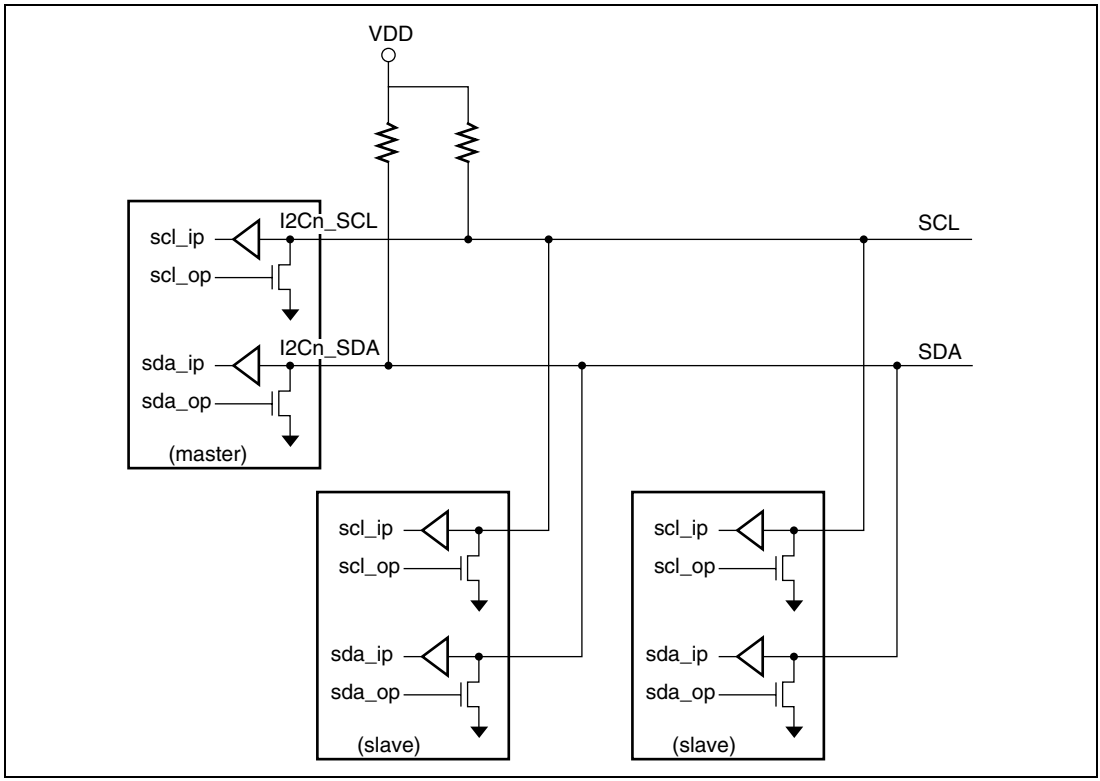


Figure 16.1 I²C Bus Interface Connection Example (When HD64404 is Used as the Master)

16.2 Features

- Supports the Philips I^{2}C bus interface
- Multi-master capability.
- Programmable seven-bit address slave
- Seven or ten bit compatible master.
- Master and slave have clock stalling for data synchronisation.
- Fast mode compatible.
- Fully programmable bus clock periods.
- Adaptable to a wide range of system clock frequencies.
- 3.3V I/O, It does not support 5V tolerant.

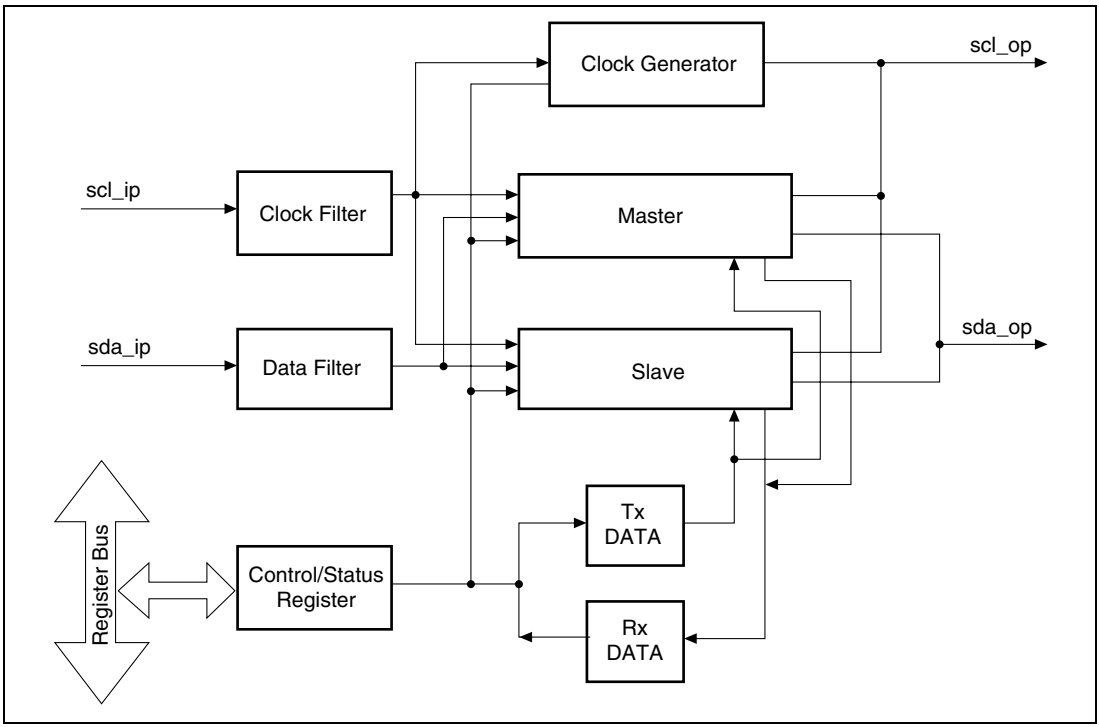


Figure 16.2 Overview Block Diagram

16.3 Pin Descriptions

Table 16.1 lists the pins used in the I^{2}C bus interface. Note that the pins depicted in the I^{2}C Block Diagram shown in Figure 16.2 are described in tables 16.2 and 16.3, respectively.

Table 16.1 I^{2}C Bus Interface

Channel	Pin Name	Direction	Description
0	I2C0_SCL	In/Out	I2C0 Serial clock input output pad*
	I2C0_SDA	In/Out	I2C0 Serial data input output pad*
1	I2C1_SCL	In/Out	I2C1 Serial clock input output pad*
	I2C1_SDA	In/Out	I2C1 Serial data input output pad*

Note: * SCL/SDA output on the I^{2}C bus open drain pad.

Table 16.2 I^{2}C Block Interface

Pin Name	Direction	Description
scl_ip	In	SCL input from the I^{2}C bus
sda_ip	In	SDA input from the I^{2}C bus
scl_op	Out	SCL output on the I^{2}C bus
sda_op	Out	SDA output on the I^{2}C bus

Table 16.3 Register Bus Interface

Pin Name	Direction	Description
Register Bus	—	Register bus signals
Irq	Out	Interrupt request

16.4 Register Map

All registers in the I^{2}C module are mapped to the register bus interface.

Table 16.4 I^{2}C Register Map

Channel	Address (Bytes)	Register Name	Abbreviation	Access Size
0	H'6780	Slave Control Register	SCR0	32
	H'6784	Master Control Register	MCR0	32
	H'6788	Slave Status Register	SSR0	32
	H'678C	Master Status Register	MSR0	32
	H'6790	Slave Interrupt Enable Register	SIER0	32
	H'6794	Master Interrupt Enable Register	MIER0	32
	H'6798	Clock Control Register	CCR0	32
	H'679C	Slave Address Register	SAR0	32
	H'67A0	Master Address Register	MAR0	32
	H'67A4	Receive Data	RXD0	32
	H'67A4	Transmit Data	TXD0	32
1	H'67C0	Slave Control Register	SCR1	32
	H'67C4	Master Control Register	MCR1	32
	H'67C8	Slave Status Register	SSR1	32
	H'67CC	Master Status Register	MSR1	32
	H'67D0	Slave Interrupt Enable Register	SIER1	32
	H'67D4	Master Interrupt Enable Register	MIER1	32
	H'67D8	Clock Control Register	CCR1	32
	H'67DC	Slave Address Register	SAR1	32
	H'67E0	Master Address Register	MAR1	23
	H'67E4	Receive Data	RXD1	32
	H'67E4	Transmit Data	TXD1	32

16.5 Register Descriptions

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, read value undefined.

All bits are active high unless otherwise stated, and are reset to their inactive level.
All accesses via the register bus are longword accesses.

16.5.1 Slave Control Register (SCR n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													SDBS	SIE	GCAE	FNA
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	0	R	**Reserved**
3	SDBS	0	R/W	**Slave Data Buffer Select (SDBS)** This bit is used to select the stage size of the receive data buffer. The SDBS bit selects the stage count of the receive data buffer which comprises two register stages (i.e., receive data register and shift register). When SDBS is set to 0, a 2-stage buffer configuration is selected, which during the period that both buffers are full and the SDR flag has not been cleared, SCL is held low, and then when the SDR flag is cleared, the low level state held on SCL is released. When SDBS is set to 1, a single-stage buffer is selected, at which point SCL will be held low from the moment the receive data register acquires the data packet up until the SDR flag is cleared.
2	SIE	0	R/W	**Slave Interface Enable (SIE)** This bit must be set for the slave to operate. If this bit is low the slave interface is effectively reset. Further, SIE is set when the MIE bit is set.
1	GCAE	0	R/W	**General Call Acknowledgement Enable (GCAE)** If it is required that the slave acknowledges the transmission of a general call address from the master, then this bit must be set.
0	FNA	0	R/W	**Force Non-Acknowledge (FNA)** In the slave receive mode, the level on the FNA bit is sent to the transmitting device as the acknowledge signal. The FNA bit is set to 0 during the period that the data packet is being received, and set to 1 on completion of data reception. Force non acknowledge back to the master during slave receive. When the slave has received the last required byte of data in a data packet, this is communicated to the master by not driving an acknowledge (nack). The master issues a stop on to the bus after receiving a nack. The setting of this bit does not effect the acknowledging of slave addresses.

16.5.2 Slave Status Register (SSR n) (n = 0,1)

The status bits (bit 0 to bit 6) of the Slave Status Register are cleared by writing zeroes to the respective status bit positions. The individual status bits are held at 1 until reset by writing a 0 to the appropriate bit position (with the exception of the GCAR and STM bits).

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										GCAR	STM	SSR	SDE	SDT	SDR	SAR
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6	GCAR	0	R	**General Call Address Received (GCAR) (Read only)** Indicates that the address received from the bus is a general call address (00H). This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the Slave Control Register) is 0 or when the SSR bit (bit 4 in the Slave Status Register) is set to 1.
5	STM	0	R	**Slave Transmit Mode (STM) (read only)** Current slave transmit mode (read/write). When set this bit indicates a write operation, when not set a read. This status bit does not cause an interrupt. This bit is automatically cleared by hardware when the SIE bit (bit 2 in the Slave Control Register) is 0 or when the SSR bit (bit 4 in the Slave Status Register) is set to 1.
4	SSR	0	R/WC0	**Slave Stop Received (SSR)** A stop has been seen on the bus. This status bit becomes active after the rising edge of SDA during the stop bit.

Bit	Bit Name	Initial Value	R/W	Description
3	SDE	0	R/WC0	**Slave Data Empty (SDE)** Transmit data has been loaded into the Shift Register. At the start of data byte transmission, the contents of the TXD Register are loaded into a shift register ready for passing onto the bus. This status bit indicates that this has taken place and that the TXD Register is again available for further data. This status bit becomes active on the falling edge of SCL before the first data bit. This bit must be reset once new data has been written to the TXD Register. The slave holds SCL low to stall the bus, if it reaches the start of a slave transmit cycle and this status bit is still set.
2	SDT	0	R/WC0	**Slave Data Transmitted (SDT)** A byte of data has been transmitted to the master on the bus. This status bit becomes active after the falling edge of SCL during the last data bit.
1	SDR	0	R/WC0	**Slave Data Received (SDR)** A byte of data has been received from the bus and is available in the Receive Data Register. This status bit becomes active after the falling edge of SCL during the last data bit. After data has been read from the RXD Register, this bit must be reset. When the SDBS bit is set to 0, SCL will be held low to stall the bus provided the SDR bit remains set until the reception of the next data packet is complete. When SDBS is set to 1, SCL will be held low from the moment the receive data register acquires the data packet up until the SDR flag is cleared.
0	SAR	0	R/WC0	**Slave Address Received (SAR)** Indicates that the slave has recognised its own address on the bus (defined by the contents of the Slave Address register). If the general call acknowledgement enable bit is enabled in the Slave Control Register, then this status bit could also indicate the reception of a general call address on the bus. In that case, bit GCAR of this register is used to differentiate the receipt of a general call address. Bit STM indicates whether the access is read (high) or write (low). This status becomes active after the falling edge of SCL during the last address bit. The slave holds SCL low at the start of the ACK phase until the software resets this status bit.

16.5.3 Slave Interrupt Enable Register (SIER n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SSRE	SDEE	SDTE	SDRE	SARE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R	**Reserved**
4	SSRE	0	R/W	**Slave Stop Received Interrupt Enable (SSRE)** When set this bit enables the SSR interrupt.
3	SDEE	0	R/W	**Slave Data Empty Interrupt Enable (SDEE)** When set this bit enables the SDE interrupt.
2	SDTE	0	R/W	**Slave Data Transmitted Interrupt Enable (SDTE)** When set this bit enables the SDT interrupt.
1	SDRE	0	R/W	**Slave Data Received Interrupt Enable (SDRE)** When set this bit enables the SDR interrupt.
0	SARE	0	R/W	**Slave Address Received Interrupt Enable (SARE)** When set this bit enables the SAR interrupt.

16.5.4 Slave Address Register (SAR n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										SADD						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6 to 0	SADD	0	R/W	**Slave Address (SADD)** This is the unique seven bit address allocated to the slave on the I^2C bus. The slave interface looks for a match between this address and the first seven bits transmitted as the slave address, at the beginning of a data packet transmission.

16.5.5 Master Control Register (MCR n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MDBS	FSCL	FSDA	OBPC	MIE	TSBE	FSB	ESG
Initial:	0	0	0	0	0	0	0	0	0	-	-	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7	MDBS	0	R/W	**Master Data Buffer Select (MDBS)** This bit is used to select the stage size of the receive data buffer. The MDBS bit selects the stage size of the master data buffer which comprises a 2-stage register configuration (Receive Data Register and Shift Register). When MDBS is set to 0, a 2-stage buffer configuration is selected, and during the period that both buffers are full and the MDR flag has not been cleared, SCL is held low, and when the MDR flag is cleared, the low level state held on SCL is released. When MDBS is set to 1, a single-stage buffer is selected, at which point SCL is held low level from the moment the data packet is applied to the Receive Data Register until the MDR flag is cleared.

Bit	Bit Name	Initial Value	R/W	Description
6	FSCL	—	R/W	**Force SCL (FSCL)** Force SCL pin level (read reflects current level on scl_ip). When the OBPC bit is set then this bit directly controls the SCL line on the bus. The level on this bit (which includes the reset level) during a read cycle, since it reflects the level on scl_ip, will change depending on the level on scl_ip.
5	FSDA	—	R/W	**Force SDA (FSDA)** Force SDA pin level (read reflects current level on sda_ip). When the OBPC bit is set then this bit directly controls the SDA line on the bus. The level on this bit (which includes the reset level) during a read cycle, since it reflects the level on sda_ip, will change depending on the level on sda_ip.
4	OBPC	0	R/W	**Override Bus Pin Control (OBPC)** When set this bit causes FSDA and FSCL, in this register, to control the SDA and SCL lines directly. This mode is used for testing purposes only.
3	MIE	0	R/W	**Master Interface Enable (MIE)** When set this bit enables the master interface. Note also that the simultaneous setting of SIE enables the slave interface.
2	TSBE	0	R/W	**Transmission of Start Byte Enable (TSBE)** When set this bit causes the master to transmit a start byte (01H) onto the bus after each start or restart that it issues. The start byte is used for interfacing to slower microcontroller based I^2C interfaces.

Bit	Bit Name	Initial Value	R/W	Description
1	FSB	0	R/W	**Force Stop onto the Bus (FSB)** When set this bit causes the master to issue a stop onto the bus, at the end of the current transfer. If ESG is also set then the master immediately issues a start and begins transmitting a new data packet. If ESG is not set then the master goes into the idle condition. I^2C module fetches the value of FSB when the last bit transmits or receives of a byte is done and it goes to STOP condition. Therefore in order to stop the communication after the predetermined number of bytes is transferred, FSB bit needs to be set before the final byte transfer is started. For transmission, FSB needs to be set together with the final data, i.e. Nth byte data, before TDE is cleared. For receive, FSB needs to be set when N-1th byte data is received and before TDR is cleared.
0	ESG	0	R/W	**Enable Start Generation (ESG)** When set this bit causes the master to start transmission of a data packet. If the bus is idle when ESG is set, then the master issues a start onto the bus and then issues the slave address. If the master is taking part in a transfer when ESG is set, then at the end of that data byte transfer, the master issues a restart before transmitting the slave address. When transmitting a data packet, the software must reset this bit when the slave address has been transmitted; else a restart is issued after every transmission is completed.

16.5.6 Master Status Register (MSR n) (n = 0,1)

The status bits (bit 0 to bit 6) of the Master Status Register are cleared by writing zeroes to the respective status bit positions. The individual status bits are held at 1 until reset by writing a 0 to the appropriate bit position.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										MNR	MAL	MST	MDE	MDT	MDR	MAT
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6	MNR	0	R/WC0	**Master Nack Received (MNR)** When set this bit indicates that the master has received a nack response (the SDA line was high during the acknowledge cycle on the bus) to either an address or data transmission.
5	MAL	0	R/WC0	**Master Arbitration Lost (MAL)** In a multi-master system, when set this bit indicates that the master has lost arbitration to one of the other masters on the bus. At this point, MIE is reset and the master interface is disabled.
4	MST	0	R/WC0	**Master Stop Transmitted (MST)** When set this bit indicates that the master has sent a stop onto the bus. A stop can be sent either as a result of the setting of the force stop bit in the Control Register, or from a nack being received from a slave during a slave receive data packet.

Bit	Bit Name	Initial Value	R/W	Description
3	MDE	0	R/WC0	**Master Data Empty (MDE)** At the start of a data byte transmission the contents of the transmit data register are loaded into a shift register ready for passing onto the bus. When set this bit indicates that this has taken place and that the Transmit Data Register is available for further data.
2	MDT	0	R/WC0	**Master Data Transmitted (MDT)** A byte of data has been sent to the slave on the bus. This status bit becomes active after the falling edge of SCL during the last data bit.
1	MDR	0	R/WC0	**Master Data Received (MDR)** A byte of data has been received from the bus and is available in the Receive Data Register. This status bit becomes active after the falling edge of SCL during the last data bit. After data has been read from the Receive Data Register this status bit must be reset. When the MDBS bit is set to 0, SCL will be held low to stall the bus provided the MDR bit remains set until the reception of the subsequent data packet is complete. When MDBS is set to 1, SCL will be held low from the moment the Receive Data Register acquires the data packet up until the MDR flag is cleared.
0	MAT	0	R/WC0	**Master Address Transmitted (MAT)** The slave address byte of a data packet has been transmitted by the master. This bit becomes active after the falling edge of SCL during the ack bit of after address.

16.5.7 Master Interrupt Enable Register (MIER n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										MNRE	MALE	MSTE	MDEE	MDTE	MDRE	MATE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6	MNRE	0	R/W	**Master Nack Received Interrupt Enable (MNRE)** When set this bit enables the MNR interrupt.
5	MALE	0	R/W	**Master Arbitration Lost Interrupt Enable (MALE)** When set this bit enables the MAL interrupt.
4	MSTE	0	R/W	**Master Stop Transmitted Interrupt Enable (MSTE)** When set this bit enables the MST interrupt.
3	MDEE	0	R/W	**Master Data Empty Interrupt Enable (MDEE)** When set this bit enables the MDE interrupt.
2	MDTE	0	R/W	**Master Data Transmitted Interrupt Enable (MDTE)** When set this bit enables the MDT interrupt.
1	MDRE	0	R/W	**Master Data Received Interrupt Enable (MDRE)** When set this bit enables the MDR interrupt.
0	MATE	0	R/W	**Master Address Transmitted Interrupt Enable (MATE)** When set this bit enables the MAT interrupt.

16.5.8 Master Address Register (MAR n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SADD1							STM1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 1	SADD1	0	R/W	**Slave Address (SADD1)** This is the address of the slave to which the master intends to communicate with.
0	STM1	0	R/W	**Slave Transfer Mode (STM1)** This bit indicates which mode the slave is to operate in. The STM1 bit sets the operating mode (transmit or receive mode) of the slave to the external slave device that matches the slave address (SADD1) sent from the master. The slave device is automatically set to the transmit/receive mode by hardware on reception of the STM1 signal. When set this bit indicates a read operation, when not set a write operation.

16.5.9 Clock Control Register (CCR n) (n = 0,1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SCGD						CDF	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 2	SCGD	0	R/W	**SCL Clock Generation Divider (SCGD)** When operating in master mode the SCL clock is generated from the internal clock frequency using SCGD as the ratio. The slave will also rely on the clock generation when SCL is held low to hold the bus up during data overrun. For this reason SCGD must be programmed for master and slave modes of operation. The formula expressing the relationship is: Equation 2 SCL rate calculation SCL freq = clock freq / (20 + (SCGD * 8)) Suggested settings for CDF and SCGD for various processor rates and the two I^2C bus speeds are given in table 16.5.
1, 0	CDF	0	R/W	**Clock Division Factor (CDF)** The internal clock for most of the blocks in the I^2C module operate as a divided version of the Register Bus clock. The main system clock is used directly in the processor interface. The internal I^2C clock is generated from the Register Bus clock using the CDF as the divider: Equation 1 Internal clock frequency calculation clock freq = sys clock freq / (1 + CDF) The minimum time to ensure adequate setup and hold times on the SDA line relative to the SCL line on the bus. The clock freq is to ensure that the glitch filtering will operate with glitches of up to 50ns in width (as described in the Fast Mode I^2C specification). Note: CDF have to be set to the value that the clock freq frequency is lower than 20 MHz.

Table 16.5 Suggested Settings for CDF and SCGD

	100 kHz		400 kHz	
Sysclockfreq	CDF	SCGD	CDF	SCGD
50 MHz	2	19	2	3
Error	–3.10%		–5.30%	
33 MHz	3	8	2	1
Error	–1.79%		–1.79%	

16.5.10 Receive/Transmit Data (RXD n/TXD n) (n = 0, 1)

Reading from or writing to this register accesses different physical internal registers. When receiving or transmitting data to or from the master or slave, a double buffered arrangement is used. When data is to be transmitted, a shift register is loaded with TXD. After data has been received into the Shift Register from the I^2C bus, it is then loaded into RXD.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RXD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 0	RXD	0	R	**Read—Receive Data (RXD)** Data received by master or slave.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TXD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	W	**Reserved**
7 to 0	TXD	0	W	**Write—Transmit Data (TXD)** Data transmitted by master or slave.

16.6 Functional Description

16.6.1 Data and Clock Filters

These blocks filter out glitches on signals coming from the I^2C bus. Glitches up to one internal clock period in width are rejected. (See Clock Control Register for details of internal clock frequency). This is specified for the faster I^2C bus rate (400kHz) but does not violate the slower I^2C bus rate specification.

These blocks also perform resynchronisation of the bus signal levels to the internal clock.

16.6.2 Clock Generator

The clock generator has two functions. Firstly, it generates the SCL I^2C bus clock under command of the master or slave interface; the SCL clock must be synchronised to any external master that may be sharing the I^2C bus. Secondly, it controls the internal clock rate, used by filtering blocks and the master and slave interfaces. This operates as a clock enable signal to the registers in these blocks. The registers are actually clocked off the master clock, but their toggle rates are determined by the internal clock rate.

16.6.3 Master/Slave Interfaces

These two interfaces run independently and in parallel. The master interface controls the transmission of address and data on the I^2C bus. The slave interface monitors the I^2C bus and takes part in transmissions if its programmed address is seen on the bus. Both interfaces communicate with the Control/Status Registers independently. Only one interrupt line comes from the module; the source could be either the master or the slave.

16.6.4 Software Interface

The I^2C module communicates with software by means of a set of status bits held in two separate registers; one for the master and one for the slave. These status bits are triggered and held, as the various phases of an I^2C bus access are encountered.

The status bits are detailed in the Register Descriptions section of this document. These status bits can enable the generation of interrupts via two associated Interrupt Enable Registers. Only one set of Transmit/Receive Data Registers are maintained, shared between the master and slave interfaces. This arrangement can function since the master and slave interfaces, although operating independently, are never involved simultaneously in an I^2C bus access.

16.6.5 Software Status Interlocking

In order that the software interface to the I^2C module be as rugged as possible, various status interlocks are built into the operation of the master and slave interfaces. The status bits involved are:

- MDR and SDR
- MDE and SDE
- MAL
- SAR

MDR and SDR: MDR and SDR are set to 1 when data is received. Clear the status after reading the Receive Data Register. If data is received while MDR and SDR are set, hardware recognises that unread data remains in the Receive Data Register and automatically holds SCL at low level and suspends data transmission. In this case, transmission can be resumed by clearing the status after reading the receive data.

Consequently, when receiving data successively, be sure to clear the status of MDR and SDR after reading the Receive Data Register.

MDE and SDE: If the MDE or SDE status bits are still set when the slave or master reaches the stage when data is to be transmitted onto the I^2C bus (using the data from the transmit data register) then the SCL line must be held low until the MDE or SDE status bits are reset. The MDE or SDE status bit being set indicates that the data currently held in the Transmit Data Register has already been transmitted onto the I^2C bus.

The software must clear this status bit when it has written to the Transmit Data Register which allows the module to continue transmitting subsequent data bytes. This is not required for the first byte of data to be transmitted onto the bus.

MAL: When the master loses arbitration, the MAL bit (of the Master Status Register) is set and the MIE bit (of the Master Control Register) is reset. At this point, the master mode is disabled and the I^2C bus interface is set to operate in the slave mode. When master operation is restarted, data transfer from the master begins after the MAL bit has been cleared.

SAR: The SAR status bit is set when the slave has recognised its address from the I^2C bus. At this point the slave interface forces the SCL line low until the SAR status bit is reset.

This is particularly important when a slave transmit is about to take place on the bus, and the slave will transmit the data from the Transmit Data Register. The software responds to the SAR status by writing the required data into the Transmit Data Register and then resetting the SAR status bit. This allows the slave interface to carry on with the access.

If the slave is about to receive data, then the situation could arise where the software has yet to read data loaded from a previous access, from the Receive Data Register. The new access could attempt to overwrite the valid data still held in the Receive Data Register. However, this is avoided using the SAR status bit. After the software has read any data in the Receive Data Register, then by resetting the SAR bit (if it is set) then no problems will arise with the Receive Data Register being overwritten.

16.7 Operation

Figure 16.3 shows the bus timing of the I^{2}C bus interface. Table 16.6 describes the meaning of each symbol in figure 16.3.

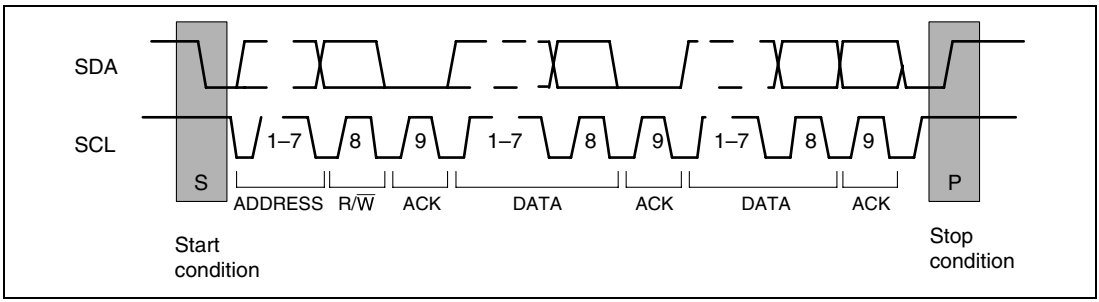


Figure 16.3 I^{2}C Bus Timing

Table 16.6 Description on Symbols of I^{2}C Bus Data Format

Symbol	Description
START CONDTION(S)	A master device changes SDA from high to low level while SCL is high level.
ADDRESS	Indicates a slave address. A slave address is selected by the master device.
R/$\overline{W}$	Indicates data transmission or reception. If the R/ $\overline{W}$ bit is 0, the data transfer direction is from the master to the slave device. If 1, vice versa.
ACK	Indicates data acknowledge. Data receive device makes SDA low level (the slave device returns a data acknowledge signal in master transmission mode, and vice versa).
DATA	Indicates transmit or receive data. The data length is eight bits, which are transferred in the MSB first.
STOP CONDITION(P)	A master device changes SDA from low to high level while SCL is high level.

16.8 I²C Bus Data Format

16.8.1 7-Bit Address Format

Figure 16.4 shows the format of data transfer from the master to the slave device (master data transmit format). Figure 16.5 shows the data transfer format (master data receive format) in which the master device read data on and after the second byte from the slave device.

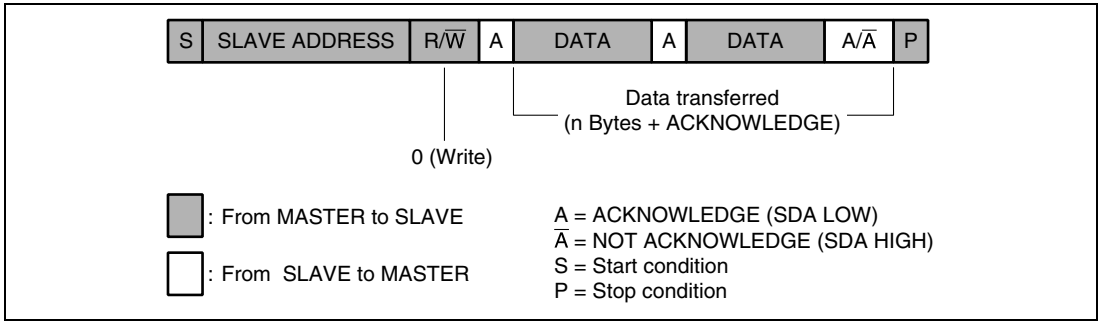


Figure 16.4 Master Data Transmit Format

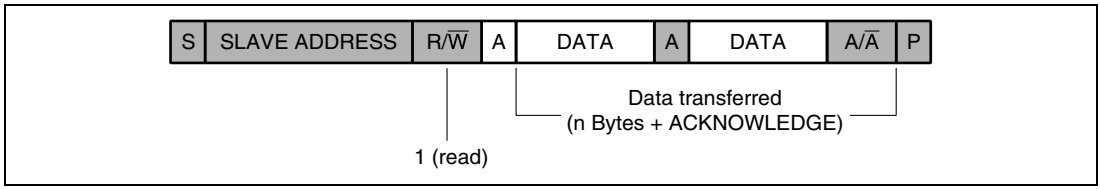


Figure 16.5 Master Data Receive Format

Figure 16.6 shows the combination transfer combination format in which the data transfer direction changes during one transfer.

When changing the direction at the first transfer, retransmit command (Sr), the slave address and the R/$\overline{W}$ signal are transmitted. In this case, the R/$\overline{W}$ signal is set to the direction opposite to the first transfer direction.

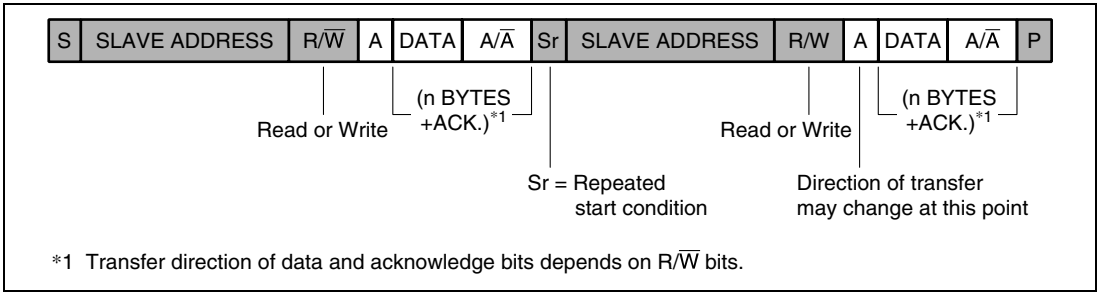


Figure 16.6 Combination Transfer Format of Master Transfer

16.8.2 10-Bit Address Format

Description is given below on the 10-bit address transfer format supported in master mode.

This format has three transfer methods as the 7-bit address transfer format.

Figure 16.7 shows the data transmit format. The set value of the Master Address Register is output in one byte following the first transfer condition (S). The value set in Transmit Data (TXD) is transmitted as a slave address in the second byte. Data transfer on and after the third byte is done in the same way as the 7-bit address data transmit.

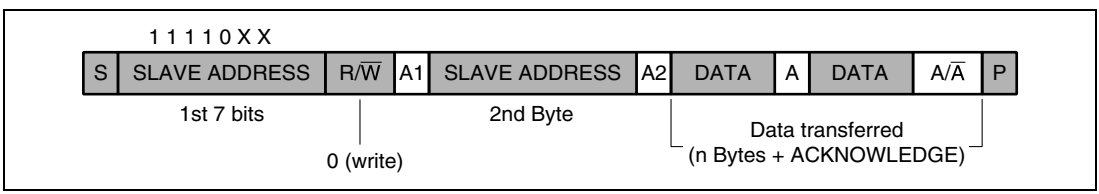


Figure 16.7 10-Bit Address Data Transfer Format

Figure 16.8 show the data receive format. Address transmit of two bytes in the data receive format is done in the same way as in the data transmit format. Then, retransmit condition (Sr) is transmitted and the value set in the Master Address Register is output. At this time, STM1 must be set to 1 (receive mode). Data transfer is done in the same way as in the 7-bit address data receive format.

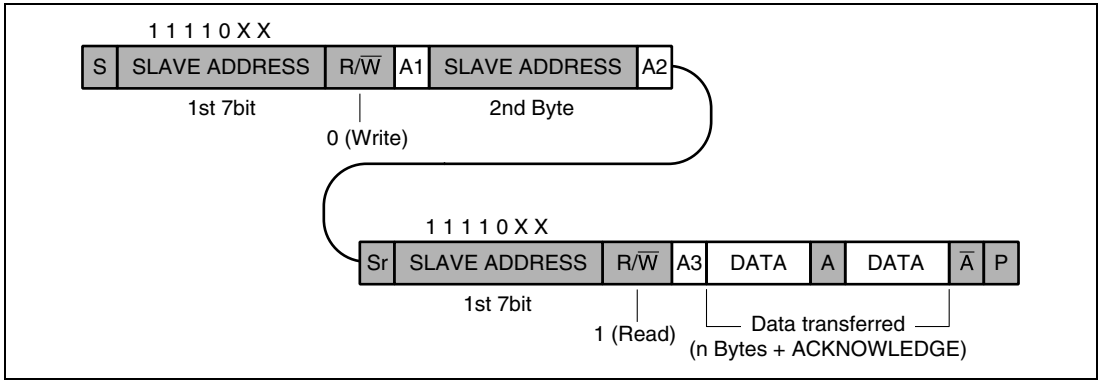


Figure 16.8 10-Bit Address Data Receive Format

Figure 16.9 shows the data transmit/receive combination format

In the data transmit/receive combination format, data is transmitted after an address is transmitted with the first two bytes. Then, retransmit condition (Sr) is transmitted instead of stop condition (P). After Sr is transmitted, the procedure is the same as that in the data receive format.

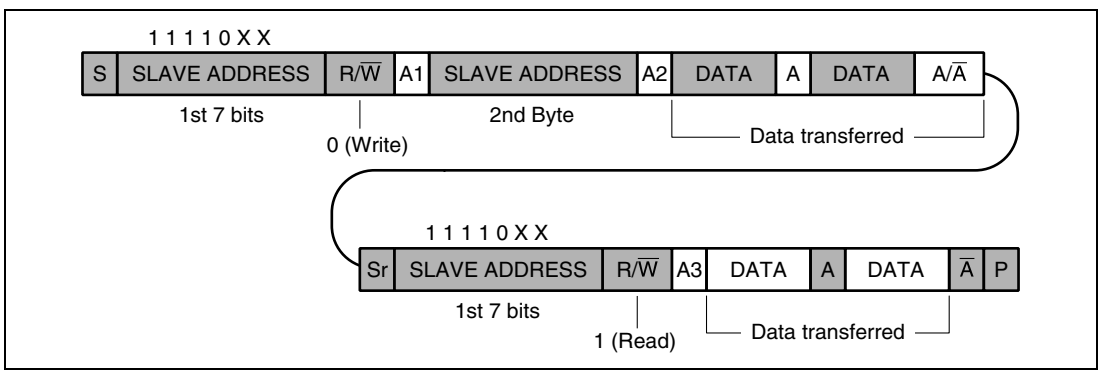


Figure 16.9 10-Bit Address Transmit/Receive Combination Format

16.8.3 Master Transmit Operation

The transmit procedure and operation in master transmit mode is described below. Figure 16.10 shows the timing operation in master transmit mode.

1. For initial setting, set the Clock Control Register, and the IRQ generation status control bits with the Clock Control Register and the Master Interrupt Enable Register according to the slave address, transmit data, and the transmit speed. Since slave mode is also required even when the master mode is used, set the device address to the slave Address Register.
2. Monitor the FSDA (bit 5) and FSCL (bit 6) in the Master Control Register. Confirm that both bits are high, which means that the other I^2C device is not using the bus. After confirmation, set the MIE (bit 3) and ESG (bit 0) in the Master Control Register to 1 to start master transmit.
3. After the signals for indicating the transmit start condition, the slave address, and the data transfer direction are transmitted, IRQ of MAT (bit 0) in the Master Status Register is generated in the timing of (1) in figure 16.10. At this time, clear the ESG bit to 0.
4. IRQ of the SAR (slave address received) is generated in the timing of (3). If the IRQ processing of the slave device is delayed, the slave device extends the SCL period to suspend data transmit (in the timing of (7) in figure 16.10). The slave device makes SDA low level at the ninth clock and returns ACK.
5. Data transmit is done in the unit of eight bits plus one bit of ACK, i.e., in the unit of nine bits. IRQ of MDE (bit 3) is generated at the ninth clock before data transfer (in the timing of (2) in figure 10). IRQ of MDT (bit 2) is generated at the eighth clock after 1-byte data transfer (in the timing of (4) in figure 16.10). Clear MDE to 0 after setting transmit data. IRQ of SDR (slave data receive) of the slave device is generated at the eighth clock (in the timing of (6) in figure 16.10). Clear SDR after the slave device reads the receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmit (in the timing of (8) in figure 16.10).
6. To end data transfer, IRQ of MNR (bit 6) in the Master Status Register is generated at the ninth clock (in the timing of (5) in figure 10) when ACK from the slave device is 1 (Nack). The master device receives this Nack and outputs data transfer end condition. When data transmit ends at the master device set FSB (bit 1) in the Master Control Register to 1 to output suspend condition. I^2C module fetches the value of FSB when the last bit transmits or receives of a byte is done and it goes to STOP condition. Therefore in order to stop the communication after the predetermined number of byte is transferred, FSB bit needs to be set before the final byte transfer is started.
7. FSB bit needs to be set before the final byte transferred. So in master transmit mode, after the last byte is set, IRQ of MST (Master Stop Transmitted) bit is checked by either interrupt or polling. At the same time MNR (Master NACK Received) bit must be checked, If NACK is returned, goes to Error Routine to retransmit the last byte.

Timing from (1) to (6) in figure 16.10 is generated after the falling edge of the clock.

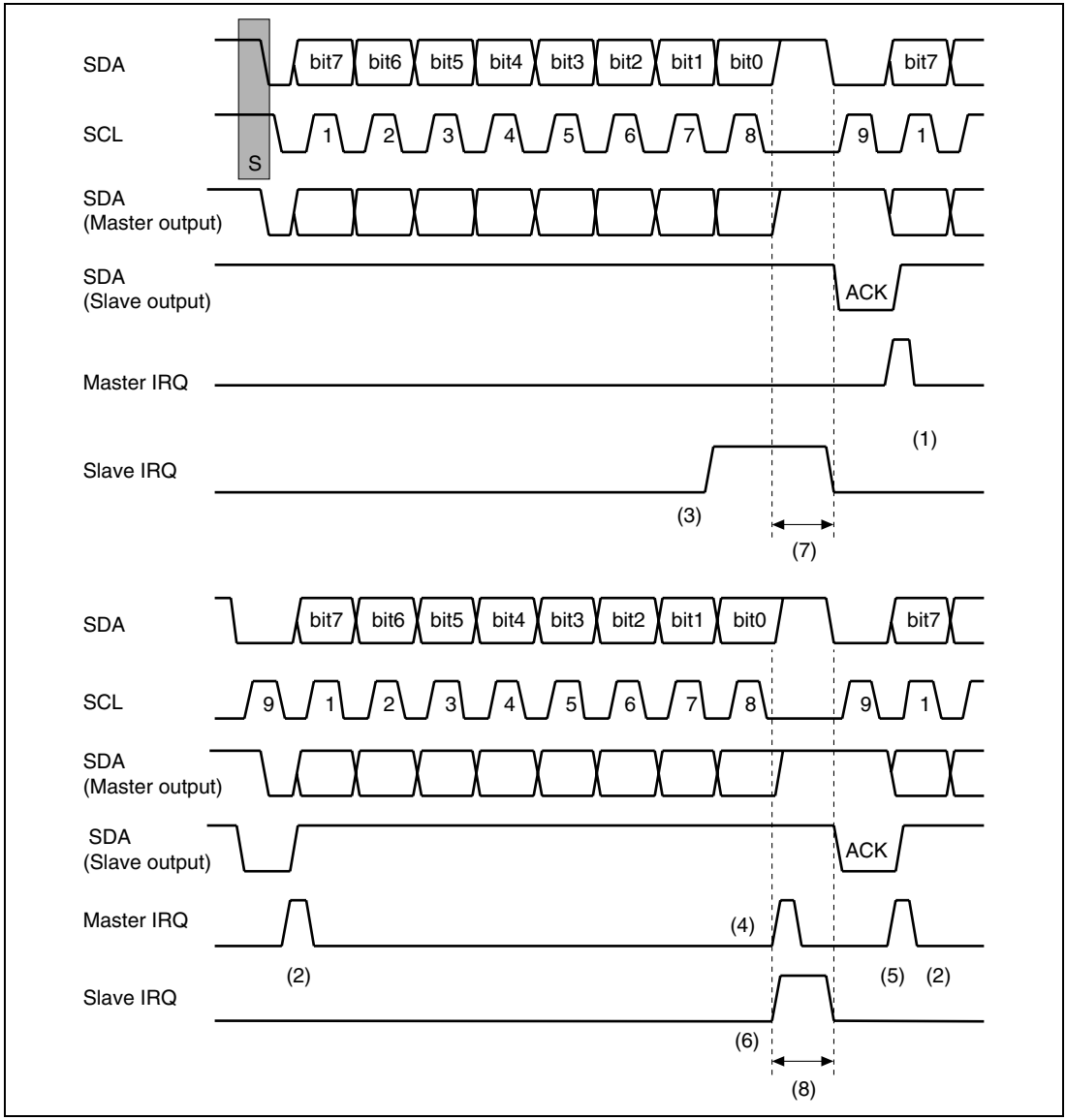


Figure 16.10 Data Transmit Mode Operation Timing

16.8.4 Master Receive Operation

Data receive procedure and operation in master receive mode is described below. Figure 16.11 shows the operation timing in master data receive mode.

1. In master data receive mode, as to transmit of a slave address and a 1-byte signal indicating the data transfer direction, operation is the same as that in master data transmit mode. At this time, however, select 1 (receive) for the data transfer direction.
2. The slave device automatically goes into data transmit mode by the signal that indicates the data transfer direction, and transmits 1-byte data synchronously with the SCL clock output from the master device. The master device generates the IRQ of MDR (bit 1) at the eighth clock (in the timing of (2) in figure 11). Clear the MDR bit after the master device reads receive data. If this processing is delayed, the slave device extends the SCL period to suspend data transmit, as shown in the timing of (3) in figure 16.11.
3. The slave device generates IRQ of the status SDT (bit 2) indicating 1-byte data transfer end at the eighth clock (in the timing of (2) in figure 11) and IRQ of the status SDE (bit 3) indicating data empty at the ninth clock (in the timing of (1) in figure 16.11). Clear SDE after writing slave transmit data to TXD.
4. To end data transfer, set FSB (bit 1) in the Master Control Register of the master device and output suspend condition. I^2C module fetches the value of FSB when the last bit transmits or receives of a byte is done and it goes to STOP condition. Therefore in order to stop the communication after predetermined number of bytes is transferred, FSB bit needs to be set before the final byte transfer is started. After confirmation of the final byte receive, though Master Receiver finishes the receive transaction, the protocol layer will inform Slave Transmitter or retransmission if the last byte is incorrect.

Timing from (1) to (3) in figure 16.11 is generated after the falling edge of the clock.

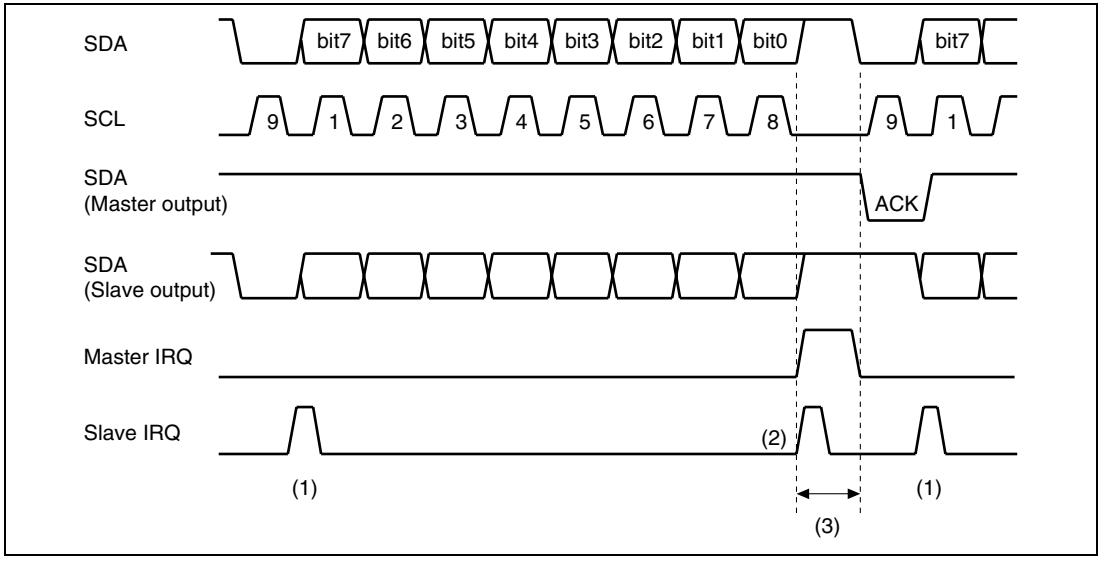


Figure 16.11 Data Receive Mode Operation Timing

16.8.5 Procedure for Entering Standby Mode

Communication cannot be done in standby mode because clock supply stops. When entering standby mode, complete communication and check the status of the registers described below.

Check the followings when communication is being done:

1. When completing communication in I^2C Master Mode, check that MST (bit 4) in the Master Status Register is 1 and clear MIE (bit 3) in the Master Control Register to 0.
2. When completing communication in I^2C Slave Mode, check that SSR (bit 4) in the Slave Status Register is 1 and clear SIE (bit 2) in the Slave Control Register to 0.

Check the followings when communication is not being done:

1. Check that MIE (bit 3) in the Master Control Register is 0.
2. Check that SIE (bit 2) in the Slave Control Register is 0.
3. Monitor the status of FSCL (bit 6) and FSDA (bit 5) in the Master Control Register to check that both of them are 1 (Determine the timing for monitoring according to the SCL frequency to be used). When MIE bit and SIE bit are 1, check that communication is not being done and clear them to 0.

16.9 Programming Examples

16.9.1 Master Transmitter

In order to set up the master interface to transmit a data packet on the I^2C bus, follow the following procedure:

1. Load the Clock Control Register:
   A. SCL clock generation divider (SCGD) <= 01h
      (SCL frequency of 400 kHz).
   B. Clock division factor (CDF) <= 2h
      (Internal frequency (clockfreq) 11MHz with 33MHz external (sysclockfreq)).
2. Load the Master Control Register, first data byte and address:
   A. Master Address Register <= Address of slave being accessed and STM1 bit (write mode: '0').
   B. Transmit Data Register <= first data byte to be transmitted.
   C. Master Control Register <= 09h
      (MIE <= 1, ESG <= 1).
3. Wait for the address to go out:
   A. Wait for master event, MAT in the Master Status Register.
   B. Master Control Register <= 08h (Reset the enable start generation bit to prevent restart. It must be done before the data byte is transmitted).
      If only one data byte is to be transmitted then Master Control Register <= 0Ah. That is: enable the stop generation. This generates a stop on the bus as soon as one byte has been transmitted.
   C. Reset status bit MAT.
4. Monitor the progress of data byte transmission:
   A. Wait for master event, MDE in the Master Status Register.
   B. Transmit Data Register <= subsequent data bytes.
   C. Reset status bit MDE.

   When last data byte is set in Transmit Data register then:
   D. Master Control Register <= 0Ah
      (Set the force stop control bit).
   E. Reset status bit MDE.
5. Wait for the end of transmission:
   A. Wait for master event, MST in the master status register.
   B. Reset status bit MST.

16.9.2 Master Receiver

In order to set up the master interface to receive a data packet on the I^2C bus, follow the following procedure:

1. Load the Clock Control Register:
   A. SCL clock generation divider (SCGD) <= 01h
      (SCL frequency of 400 kHz).
   B. Clock division factor (CDF) <= 2h
      (Internal frequency (clockfreq) 11MHz with 33MHz external (sysclockfreq)).
2. Load the Master Control Register and address:
   A. Master Address Register <= address of slave being accessed and STM1 bit (read mode: '1').
   B. Master Control Register <= 09h
      (MIE <= 1, ESG <= 1).
3. Wait for the address to go out:
   A. Wait for master event, MAT in the Master Status Register.
   B. Master Control Register <= 08h
      (Reset the enable start generation bit to prevent restart. It must be done before the data byte is received).
      If only one data byte is to be received then Master Control Register <= 0Ah. That is: enable the stop generation. This generates a stop on the bus as soon as one byte has been received.
   C. Reset status bit MAT.
4. Monitor the progress of data byte reception:
   A. Wait for master event, MDR in the Master Status Register.
   B. Read data from Received Data Register.
   C. Reset status bit MDR.

   If next data byte received is to be the last data byte transmitted by the slave device then:
   D. Master Control Register <= 0Ah
      (Set the force stop control bit).
   E. Reset status bit MDR.
5. Wait for the end of transmission:
   A. Wait for master event, MST in the Master Status Register.
   B. Reset status bit MST.

16.9.3 Master Transmitter—Restart—Master Receiver

In order to set up the master interface to transmit a byte of data on the I^2C bus, issue a restart, then read data bytes back from the slave, follow the following procedure:

1. Load the Clock Control Register:
   A. SCL clock generation divider (SCGD) <= 01h
   (SCL frequency of 400 kHz).
   B. Clock division factor (CDF) <= 2h
   (Internal frequency (clockfreq) 11MHz with 33MHz external (sysclockfreq)).
2. Load the Master Control Register and address:
   A. Master Address Register <= address of slave being accessed and STM1 bit (writes mode: '0').
   B. Master Control Register <= 09h
   (MIE <= 1, ESG <= 1).
3. Wait for the address to go out:
   A. Wait for master event, MAT in the Master Status Register.
   B. Master Address Register <= address of slave being accessed and STM1 bit (read mode: '1').
   Since the enable start generation bit in the Master Control Register is still set, at the end of the byte transmission the master will issue a restart. Since the new address has been loaded above (to a read) the bus direction will be turned around.
   C. Reset status bit MAT.
4. Wait for the address to go out:
   A. Wait for master event, MAT in the Master Status Register.
   B. Master Control Register <= 08h
   (Reset the enable start generation bit to prevent further restart. It must be done before the data byte is received).
   C. Reset status bit MAT.
5. Monitor the progress of data byte reception:
   A. Wait for master event, MDR in the Master Status Register.
   Read data from Received Data Register and reset status bit MDR.
   If next data byte received is to be the last data byte transmitted by the slave device then:
   B. Master Control Register <= 0Ah
   (set the force stop control bit).
   C. Reset status bit MDR.
6. Wait for the end of transmission.
   A. Wait for the master event MST in the Master Status Register.
   B. Reset status bit MST

16.10 Notice

Purchase of I^2C components of Hitachi, Ltd., or one of its sublicensed Associated Companies conveys as license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

Section 17 Hitachi Serial Peripheral Interface

17.1 General Description

The Hitachi Serial Peripheral Interface Module is a transceiver module designed to send and receive control information. It is designed so that it can be easily connected to peripherals outside the device.

The HSPI module can be configured in either Master or Slave mode and in Master mode can initiate transmissions.

The transmit and receive sections within the module are double buffered to allow duplex communication.

A flexible system clock division strategy allows a wide range of bit rates to be supported.

The programmable clock control logic allows setting for 2 different transmit protocols and accommodates transmit and receive functions on either edge of the serial bit clock.

Error detection logic is provided for warning of read buffer overflow.

The module has a facility to generate the Chip Select to slave modules when configured as a master either automatically as part of the data transfer process, or under the manual control of the host processor.

The module supports DMA transfer of both receive and transmit data independently via two DMA channels if implemented in the system.

17.2 Interfaces

The following block diagram shows how the HSPI Module could be integrated into a system. Implementations can vary depending on whether the module is required to support both Master and Slave modes.

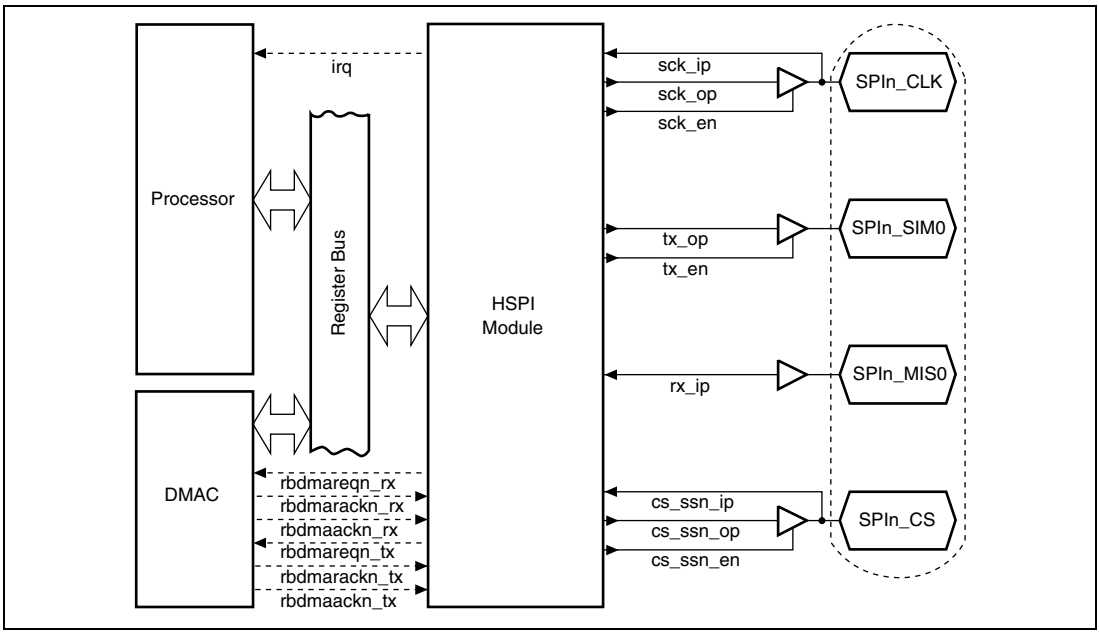


Figure 17.1 Interface Block Diagram

17.2.1 Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 17.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From
Register Bus	—		Access to Registers	
irq	1	Out	Interrupt line	Interrupt Priority
rbdmareqn_rx	1	Out	Receive DMA Request	DMAC
rbdmarackn_rx	1	In	Receive DMA request acknowledge	DMAC
rbdmaackn_rx	1	In	Receive DMA access occurring	DMAC
rbdmareqn_tx	1	Out	Transmit DMA Request	DMAC
rbdmarackn_tx	1	In	Transmit DMA request acknowledge	DMAC
rbdmaackn_tx	1	In	Transmit DMA access occurring	DMAC
sck_ip	1	In	Serial Clock Input	from I/O Buffer
sck_op	1	Out	Serial Clock Output	to I/O Buffer
sck_en	1	Out	Serial Clock output enable	to I/O Buffer
tx_op	1	Out	Serial Transmit Data	to I/O Buffer
rx_ip	1	In	Serial Receive Data	from I/O Buffer
tx_en	1	Out	TX output enable	to I/O Buffer
cs_ssn_ip	1	In	Slave Select Input	from I/O Buffer
cs_ssn_op	1	Out	Chip Select Output	to I/O Buffer
cs_ssn_en	1	Out	Chip Select output enable	to I/O Buffer

17.2.2 Software Interfaces

The registers accessible by the software are listed in the following table. All registers should be read and written to as 32-bit words.

Table 17.2 Register List

Channel	Address (Bytes)	Register name	Abbreviation	Access Size
0	H'66E0	Control Register 0	CR0	32
	H'66E4	Status Register 0	SR0	32
	H'66E8	System Control Register 0	SCR0	32
	H'66EC	Transmit Buffer Register 0	TXBR0	32
	H'66F0	Receive Buffer Register 0	RXBR0	32
1	H'6700	Control Register 1	CR1	32
	H'6704	Status Register 1	SR1	32
	H'6708	System Control Register 1	SCR1	32
	H'670C	Transmit Buffer Register 1	TXBR1	32
	H'6710	Receive Buffer Register 1	RXBR1	32
2	H'6720	Control Register 2	CR2	32
	H'6724	Status Register 2	SR2	32
	H'6728	System Control Register 2	SCR2	32
	H'672C	Transmit Buffer Register 2	TXBR2	32
	H'6730	Receive Buffer Register 2	RXBR2	32

17.3 Register Description

Legends for register description:

Initial Value : Register value after reset
— : Undefined value
R/W : Read and Write, write value can be read.
R : Read only, for write always 0 write
R/WC0 : Read and Write, 0 write clear, 1 write is ignored
R/WC1 : Read and Write, 1 write clear, 0 write is ignored
W : Write only, Read prohibited. If reserved, write always 0.
—/W : Write only, Read value undefined.

17.3.1 Control Register n (CR n) (n = 0 to 2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									FBS	CLKP	IDIV	CLKC				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
7	FBS	0	R/W	**First Bit Start (FBS)** This bit controls the timing relationship between each bit of transferred data and the serial bit clock. 0: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device on the first edge of SCK after SPIn_CS pin goes low. Similarly the first received bit is sampled on the first edge of SCK after SPIn_CS pin goes low. 1: The first bit transmitted from the HSPI module is set up such that it can be sampled by the receiving device on the second edge of SPIN_CLK after SPIn_CS pin goes low. Similarly the first received bit is sampled on the second edge of SPIN_CLK after SPIn_CS pin goes low.
6	CLKP	0	R/W	**Serial Clock Polarity (CLKP)** 0: SPIN_CLK signal is not inverted and so is low when inactive. 1: SPIN_CLK signal is inverted and so is high when inactive.

Bit	Bit Name	Initial Value	R/W	Description
5	IDIV	0	R/W	**Initial Clock Division Ratio (IDIV)** 0: The system clock is divided by a factor of 4 initially to create an intermediate frequency which is further divided to create the serial bit clock when a master. 1: The system clock is divided by a factor of 32 initially to create an intermediate frequency which is further divided to create the serial bit clock when a master.
4 to 0	CLKC	0	R/W	**Clock Division Count (CLKC)** This value determines how many intermediate frequency cycles long both the high and low periods of the serial bit clock will last. 00000: High and Low period = 1 intermediate frequency cycle. Serial bit clock frequency = intermediate frequency/2. 00001: High and Low period = 2 intermediate frequency cycles. Serial bit clock frequency = intermediate frequency/4. 00010: High and Low period = 3 intermediate frequency cycles. Serial bit clock frequency = intermediate frequency/6. : 11111: High and Low period = 32 intermediate frequency cycles. Serial bit clock frequency = intermediate frequency/64.

The serial bit clock frequency can be computed using the following formula:

$$\text{Serial bit clock requency} = \frac{\text{System clock frequency}}{\text{Initial division} \times ((\text{Clock count} + 1) \times 2)}$$

When the module is configured as a slave the IDIV and CLKC fields are ignored and the module synchronises to the externally supplied serial bit clock. The highest external serial bit clock that the module can operate with is system clock frequency/8.

If any of the FBS, CLKP, IDIV or CLKC bit values are changed, then the module will undergo a soft reset.

17.3.2 Status Register n (SRn) (n = 0 to 2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TXFU	TXHA	TXEM	RXFU	RXHA	RXEM	RXOO	RXOW	RXFL	TXFN	TXFL
Initial:	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/WC0	R/WC0	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
10	TXFU	0	R	**Transmit Fifo Full Flag (TXFU)** This status flag applies to the fifo mode of operation only. The flag is set when the transmit fifo is full of bytes for transmission and cannot accept any more. It is cleared when data is taken out of the transmit fifo for transfer on the SPI bus.
9	TXHA	0	R	**Transmit Fifo Halfway Flag (TXHA)** This status flag applies to the fifo mode of operation only. The flag is set when the transmit fifo reaches the halfway point, ie has 4 bytes of data and 4 spaces for more data. It is cleared when more data is written to the transmit fifo. It remain set until cleared regardless of the subsequent fifo levels. If TXHA = 1 and THIE = 1 then module pin *irq* = 1
8	TXEM	1	R	**Transmit Fifo Empty Flag (TXEM)** This status flag applies to the fifo mode of operation only. The flag is set when the transmit fifo is empty of data to transmit. It is cleared when more data is written to the transmit fifo. If TXEM = 1 and TEIE = 1 then module pin *irq* = 1

Bit	Bit Name	Initial Value	R/W	Description
7	RXFU	0	R	**Receive Fifo Full Flag (RXFU)** This status flag applies to the fifo mode of operation only. The flag is set when the receive fifo is full of received bytes and cannot accept any more. It is cleared when data is read out of the receive fifo. If RXFU = 1 and RFIE = 1 then module pin *irq* = 1
6	RXHA	0	R	**Receive Fifo Halfway Flag (RXHA)** This status flag applies to the fifo mode of operation only. The flag is set when the receive fifo reaches the halfway point, ie has 4 bytes of data and 4 spaces for more data. It is cleared when more data is read from the receive fifo. It remain set until cleared regardless of the subsequent fifo levels. If RXHA = 1 and RHIE = 1 then module pin *irq* = 1
5	RXEM	1	R	**Receive Fifo Empty Flag (RXEM)** This status flag applies to the fifo mode of operation only. The flag is set when the receive fifo is empty of received data. It is cleared when more data is received into to the receive fifo. If RXEM = 0 and RNIE = 1 then module pin *irq* = 1
4	RXOO	0	R/WC0	**Receive Buffer Overrun Occurred Flag (RXOO)** This status flag is set when new data has been received but the previous received data has not been read from the HSPI module Receive Buffer Register (RXBR). The previously received data will not be overwritten by the newly received data. The RXOO flag will stay HIGH until reset by writing a 0 to its bit position. If RXOO = 1 and ROIE = 1 then module pin *irq* = 1.
3	RXOW	0	R/WC0	**Receive Buffer Overrun Warning Flag (RXOW)** This status flag is set when a new serial data transfer starts and the previous received data has not been read from the HSPI module Receive Buffer Register (RXBR). The RXOW flag will stay HIGH until reset by writing a 0 to its bit position. If RXOW= 1 and ROIE = 1 then module pin *irq* = 1.

Bit	Bit Name	Initial Value	R/W	Description
2	RXFL	0	R	**Receive Buffer Full Status Flag (RXFL)** This status flag indicates that new data is available in the RXBR Register and has not yet been read. It is set at the completion of a serial bus transfer at the point the shift register contents are loaded into the Receive Buffer. This bit can be reset by reading the RXBR Register. If RXFL = 1 and RXDE = 1 then module pin *rbdmareq_rx* = 1.
1	TXFN	0	R	**Transmit Finish Status Flag (TXFN)** This status flag indicates that the last transmission has completed. It is set as the Transmit Buffer Register is able to accept more data from the Register Bus. This bit can be reset by writing more data to the TXBR Register. If TXFN= 1 and TFIE = 1 then module pin *irq* = 1
0	TXFL	0	R	**Transmit Buffer Full Status Flag (TXFL)** This status flag indicates that the Transmit Buffer Register has unsent data. It is set as the Transmit Buffer Register is written with data from the Register Bus. This bit is reset when the Transmit Buffer Register is able to accept more data from the Register Bus If TXFL= 0 (i.e. the transmit buffer is empty) and TXDE = 1 then module pin *rbdmareq_tx* = 1.

17.3.3 System Control Register n (SCR n) (n = 0 to 2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TE IE	TH IE	RN IE	RH IE	RF IE	FF EN	LM SB	CSV	CSA	TF IE	RO IE	RX DE	TX DE	MA SL
Initial:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W	R	R	W	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
13	TEIE	0	W	**Transmit Fifo Empty Interrupt Enable (TEIE)** 0: Transmit Fifo Empty Interrupt Disabled. 1: Transmit Fifo Empty Interrupt Enabled. This is a write-only register. When read, data is undefined.
12	THIE	0	W	**Transmit Fifo Halfway Interrupt Enable (THIE)** 0: Transmit Fifo Halfway Interrupt Disabled. 1: Transmit Fifo Halfway Interrupt Enabled. This is a write-only register. When read, data is undefined.
11	RNIE	0	R/W	**Receive Fifo Not Empty Interrupt Enable (RNIE)** 0 : Receive Fifo Not Empty Interrupt Disabled. 1 : Receive Fifo Not Empty Interrupt Enabled.
10	RHIE	0	R/W	**Receive Fifo Halfway Interrupt Enable (RHIE)** 0: Receive Fifo Halfway Interrupt Disabled. 1: Receive Fifo Halfway Interrupt Enabled.
9	RFIE	0	R/W	**Receive Fifo Full Interrupt Enable (RFIE)** 0: Receive Fifo Full Interrupt Disabled. 1: Receive Fifo Full Interrupt Enabled.
8	FFEN	0	R/W	**Fifo Mode Enable (FFEN)** This bit controls the enabling of fifo mode. When fifo mode is enabled two 8-entry deep fifos are made available, one for transmit data and one for receive data. These fifos are read and written via the TXBR and RXBR Registers. When fifo mode is disabled the existing TXBR and RXBR Registers are used directly so new data must be written to the TXBR Register and read from the RXBR Register for each and every transfer on the SPI bus. Fifo mode must not be enabled if DMA requests are also going to be used to service the TXBR and RXBR registers. 0: Fifo mode disabled. 1: Fifo mode enabled.

Bit	Bit Name	Initial Value	R/W	Description
7	LMSB	0	R/W	**LSB or MSB first Control (LMSB)** 0: Data is transmitted and received most significant bit (MSB) first. 1: Data is transmitted and received least significant bit (LSB) first.
6	CSV	1	R/W	**Chip Select Value (CSV)** This bit controls the value of the Chip Select output when the module is a master and manual Chip Select generation has been selected. 0: Chip Select output is low. 1: Chip Select output is high.
5	CSA	0	R/W	**Chip Select Automatic or Manual (CSA)** 0: Chip Select output is automatically generated during the transfer of data. 1: Chip Select output is manually controlled, with its value being determined by the CSV bit.
4	TFIE	0	R/W	**Transmission Finished Interrupt Enable (TFIE)** 0: Transmission finished interrupt disabled. 1: Transmission finished interrupt enabled.
3	ROIE	0	R/W	**Receive Overrun Occurred / Warning Interrupt Enable (ROIE)** 0: Receive overrun occurred / warning interrupt disabled. 1: Receive overrun occurred / warning interrupt enabled.
2	RXDE	0	R/W	**Receive DMA Enable (RXDE)** 0: Receive DMA module pin *rbdmareq_rx* disabled. 1: Receive DMA module pin *rbdmareq_rx* enabled.
1	TXDE	0	R/W	**Transmit DMA Enable (TXDE)** 0: Transmit DMA module pin *rbdmareq_tx* disabled. 1: Transmit DMA module pin *rbdmareq_tx* enabled.
0	MASL	0	R/W	**Master/Slave Select Bit (MASL)** 0: HSPI module configured as Slave. 1: HSPI module configured as Master.

If any of the FFEN, LMSB, CSA or MASL bit values are changed, then the module will undergo a soft reset.

17.3.4 Transmit Buffer Register n (TXBR n) (n = 0 to 2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
7 to 0	TD	0	R/W	**Transmit Data (TD)** Data written to this register is passed to the Shift Register as it is required for transmission. Reading this register will return the data in the Transmit Buffer.

17.3.5 Receive Buffer Register n (RXBR n) (n = 0 to 2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved** The returned value is not guaranteed.
7 to 0	RD	0	R	**Received Data (RD)** Data in this field is passed from the Shift Register as each byte is received, unless the previous received data has not been read.

17.4 HSPI Module Operation

17.4.1 Operation Overview without DMA (Fifo Mode Disabled)

The operation of the transmit/receive function is best described by considering the flow events required for the process. The Flow Chart in Figure 17.2 below describes the procedural flow of a transmit/receive operation.

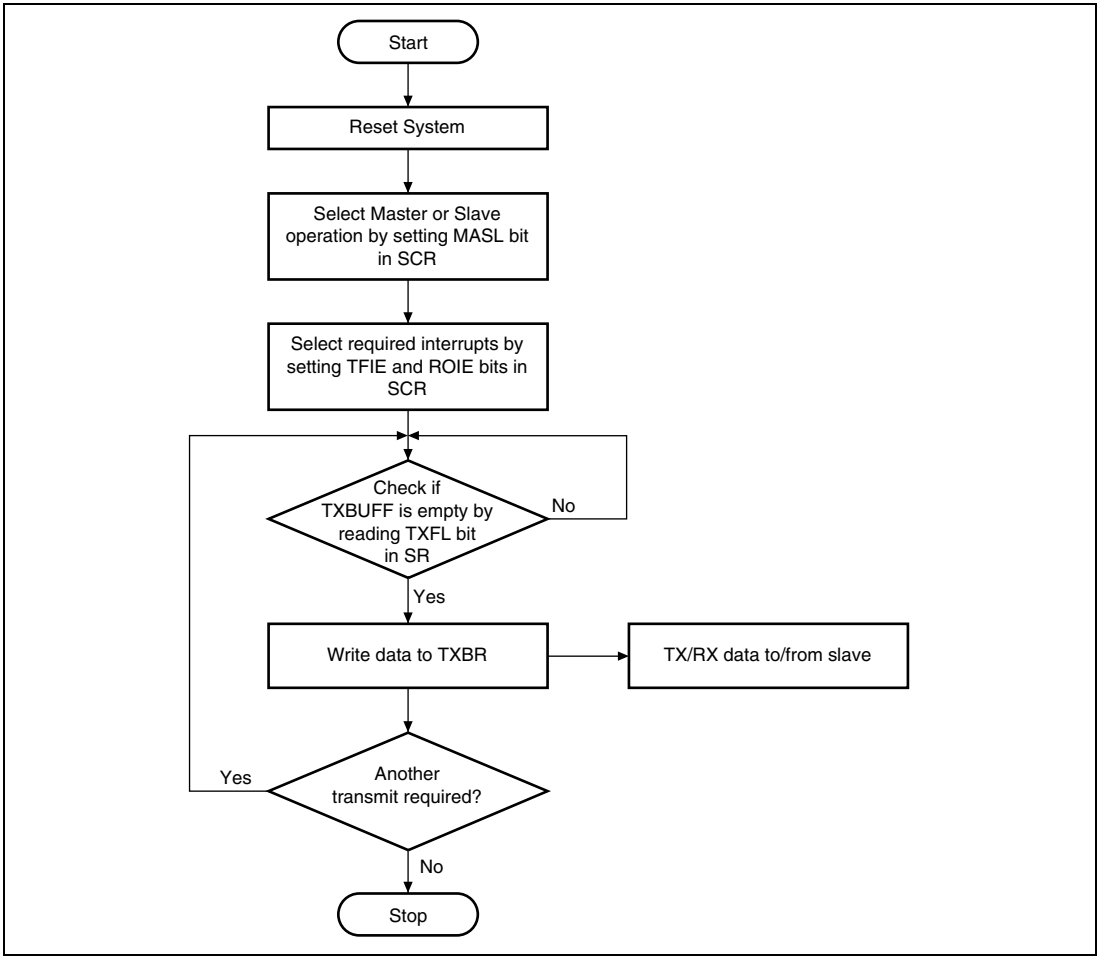


Figure 17.2 Operational Flowchart

Depending on the settings of the Control Register (CR) the Master will transmit data to the Slave on either the falling or rising edge of SPIN_CLK and sample data from the Slave on the opposite edge. The transmit function between the master and slave is complete when the Transmit Finish (TXFN) bit in the Status Register (SR) is set. This bit should be used to identify when an SPI transfer event (byte transmitted and byte received) has occurred, even in the case where the SPI module is being used to receive data only (null data being transmitted). By default data is transmitted MSB first, but LSB first is also possible depending on how the LMSB bit in the System Control Register (SCR) is configured.

During the transmit function the Slave responds by sending data to the Master synchronised with the SPIN_CLK from the master. Data from the Slave is sampled and shifted into the Shift Register in the module and on completion of the transmit function, is transferred into the Receive Buffer Register (RXBR).

The SPIN_CS pin is used to select the HSPI module when configured as a slave, and prepare it to receive data from an external Master. When the FBS bit in the CR is 0, the SPIN_CS pin must be driven high between successive bytes. When FBS = 1, the SPIN_CS pin can stay low for several byte transmissions. In this case, if the system is configured such that FBS is always 1, then the SPIN_CS line can be tied to ground (if the module will only be used as a slave).

17.4.2 Operation Overview with DMA

The operation of the module when DMA is used to perform transmit and receive data transfers is simpler than when DMA is not used. The module must be configured as in the case for transfers without DMA. Fifo mode must not be enabled. The DMA controller should then be configured to transfer the required amount of data. DMA requests can then be enabled in the HSPI module and the transfers will then take place without further processor intervention.

When the DMA controller indicates that all transfers have finished then the DMA request signals in the HSPI module should be disabled to remove any remaining DMA requests. This is necessary as the HSPI module will always request data to transmit.

17.4.3 Operation Overview with Fifo Mode Enabled

In order to reduce the interrupt overhead on the processor when operation in DMA mode is not an option a fifo mode has been provided. When fifo mode is enabled up to 8 bytes can be written in advance for transmission and 8 bytes can be received before the receive fifo needs to be read. If a known amount of data is to be transferred between the SPI module and an external device then the following procedure can be followed:

- Set up the module for the required SPI transfer characteristics (Master/Slave , Clock polarity etc) and enable fifo mode.
- Write bytes into the transmit fifo via the TXBR Register. If more than 8 bytes are to be transmitted then enable the Transmit Fifo Halfway Interrupt to keep track of the fifo level as data is transmitted.
- Respond to the Transmit Fifo Halfway interrupt when it occurs by writing more data to the transmit fifo and reading data from the receive fifo via the RXBR Register.
- When all of the transmit data has been written into the transmit fifo, disable the Transmit Fifo Halfway Interrupt and read the contents of the receive fifo until it is empty. Enable the Receive Fifo Not Empty Interrupt to keep track of when the final bytes of the transfer are received.
- Respond to the Receive Fifo Not Empty Interrupt until all the expected data has been received.
- Disable the module until it is required again.

In some applications it is necessary to receive an unknown quantity of data from an external SPI device. If this is the case the following procedure can be used:

- Set up the module for the required SPI transfer characteristics (Master/Slave , Clock polarity etc) and enable fifo mode.
- Fill the transmit fifo with the data to transmit. Enable the Receive Fifo Not Empty Interrupt.
- Respond to the Receive Fifo Not Empty Interrupt and read data from the receive fifo until empty. Write more data to the transmit fifo if required.
- Disable the module when the transfer is to stop.

17.4.4 Timing Diagrams

The following diagrams explain the timing relationship of all shift and sample processes in the HSPI. Figure 17.3 shows the conditions when FBS = 0, whilst Figure 17.4 shows the conditions when FBS = 1. It can be seen that if CLKP = 0 then transmit data is shifted on the falling edge of SPIN_CLK and receive data is sampled on the rising edge. The opposite is true when CLKP = 1.

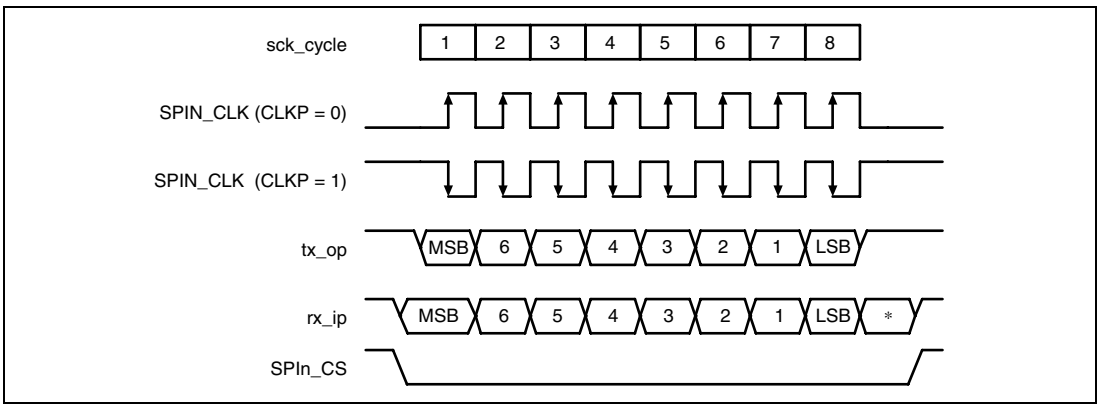


Figure 17.3 Timing Conditions when FBS = 0

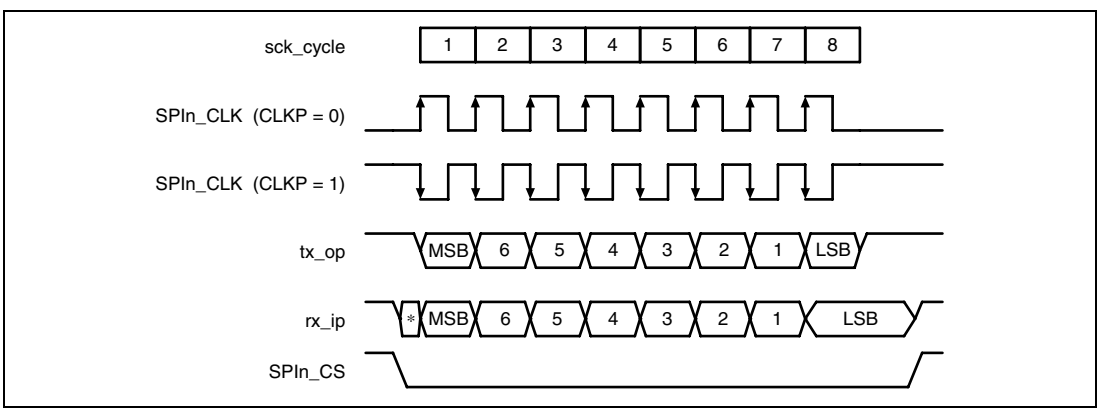


Figure 17.4 Timing Conditions when FBS = 1

17.4.5 Error Handling

To make sure that the system is running correctly, the following status flags are implemented in the Status Register (SR):

- receive buffer overrun occurred
- receive buffer overrun warning
- receive buffer full
- transmission finished
- transmit buffer full
- receive fifo empty
- receive fifo halfway
- receive fifo full
- transmit fifo empty
- transmit fifo halfway
- transmit fifo full

The receive buffer overrun occurred and receive buffer overrun warning can be output to the IRQ pin if the overrun occurred / warning interrupt enable bit has been set in the System Control Register (SCR). The transmission finished flag can be output to the IRQ pin if the transmission finished interrupt enable bit has been set in the System Control Register (SCR). The receive fifo halfway, receive fifo full, transmit fifo empty and transmit fifo halfway flags can be output to the IRQ pin if the appropriate interrupt enable bits in the System Control Register (SCR) are set. The receive fifo empty flag can generate an IRQ when showing not empty if the receive fifo not empty interrupt enable bit is set in the System Control Register (SCR).

Receive buffer overrun occurred: This occurs when the receive buffer has not been read and the next data item has been received. The previously received data will not be overwritten, but the newly received data will be lost.

Receive buffer overrun warning: This occurs when the receive buffer has not been read and the next data transfer has started. In this case there is a risk that the data currently being received will be lost unless the previously received data is read before the end of the current transfer.

Receive buffer full: This status flag is set when the receive buffer register contains received data that has not yet been read.

Transmission finished: This status flag is set when the current transfer has finished and another can take place.

Transmit buffer full: This status flag is set when the transmit buffer register contains data that has not yet been transmitted.

Receive fifo empty: This status flag is set when the receive fifo is empty of received data.

Receive fifo halfway: This status flag is set when the receive fifo is half full of received data.

Receive fifo full: This status flag is set when the receive fifo is full of received data.

Transmit fifo empty: This status flag is set when the transmit fifo has no data left to transmit.

Transmit fifo halfway: This status flag is set when the transmit fifo is half full of data to transmit.

Transmit fifo full: This status flag is set when the transmit fifo is full of data to transmit.

17.4.6 Soft Reset

To ensure the module can be returned to a known state and to flush the receive and transmit fifo pointers a soft reset of the module can be performed. A soft reset occurs whenever control bits change excluding interrupt/dma enable bits and the Chip Select Value bit. Hence to cause a soft reset all that needs to be done is to change one of the following control bits:

- First Bit Start (FBS)
- Serial Clock Polarity (CLKP)
- Initial Clock Division Ratio (IDIV)
- Clock Division Count (CLKC)
- Fifo Mode Enable (FFEN)
- LSB or MSB first Control (LMSB)
- Chip Select Automatic or Manual (CSA)
- Master/Slave Select Bit (MASL)

If the master device sets CS SSN low except in data transfer when this module is in slave mode, set the Control Register(CR) register again after software reset. This is to prevent data from being erroneously received.

17.5 Functional Description

The below diagram shows the main internal blocks of the HSPI and their interconnection.

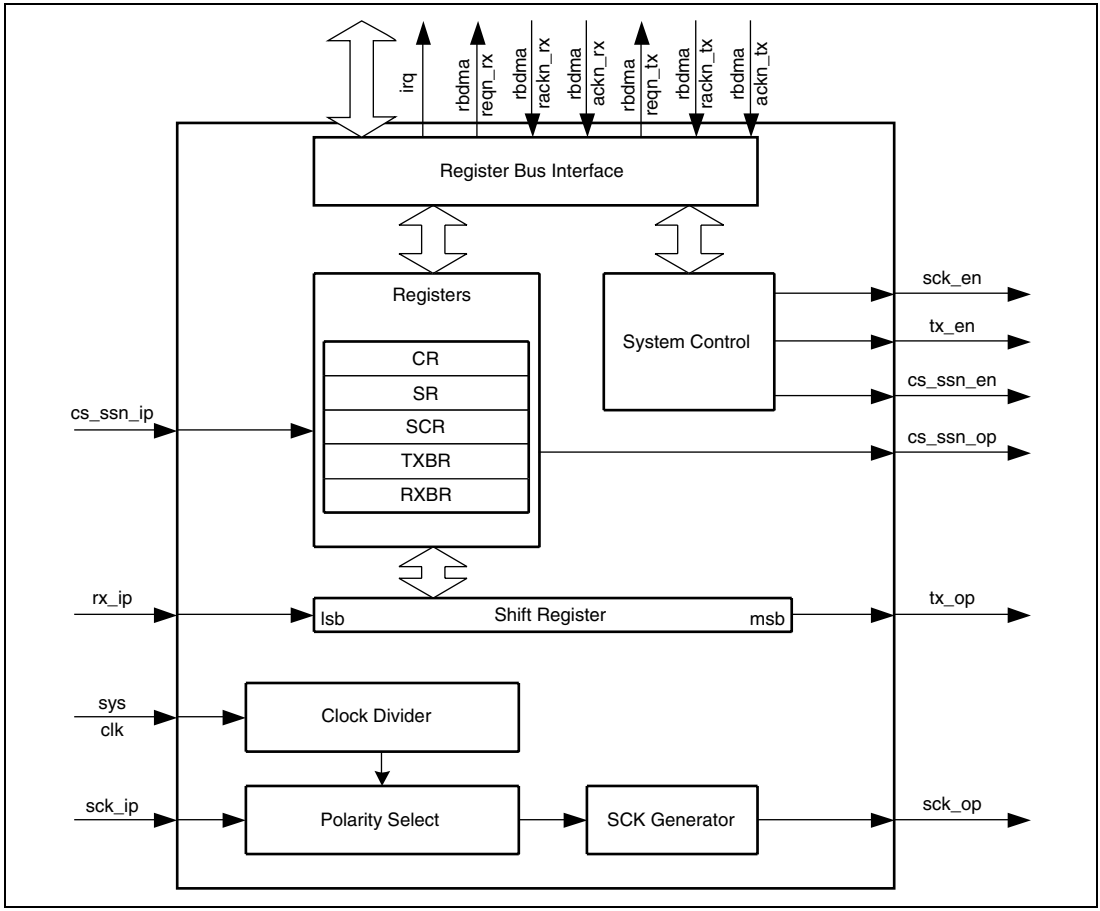


Figure 17.5 Functional Diagram

17.5.1 Clock Selection

The Clock Control bits in the Control Register (CR) allows setting of different bit rates for data transmission. In both master and slave modes the HSPI can transmit/receive data at a maximum frequency of system clock/8.

17.5.2 Clock Polarity and Transmit Control

The Control Register (CR) also allows the user to define when the First Bit (FBS) of transmit data will be shifted and the polarity required for transmission. The FBS bit in the Control Register (CR) allows selection between 2 different transfer formats. The MSB or LSB is valid on the falling edge of SPIN_CS. The Serial Clock Polarity (CLKP) bit in the Control Register (CR) allows for control of the Polarity Select block which controls which edges of SPIN_CLK shift and sample data in the Master and Slave.

17.5.3 Transmit and Receive Routines

The Master and Slave can be considered linked together as a circular shift register synchronised with SPIN_CLK. The transmit byte from the Master is replaced with the receive byte from the Slave in 8 SPIN_CLK cycles. Both the transmit and receive functions are double buffered to allow for continuous reads and writes. When fifo mode is enabled 8 entry fifos are available for both transmit and receive data.

17.6 Power Saving and Clocking Strategy

The module is a synchronous design clocked throughout by the register bus clock.

The SPI module allows clock gating on the register bus clock to reduce power consumption. Standby mode can be enabled/disabled by controlling the SPI0,SPI1 and SPI2 bits in the Clock Control 1 (CC1) Register in the Power and Control module.

To wake up the module, the SPI0, SPI1 and/or SPI2 bits in the Clock Control 1 (CC1) Register should be enabled. After enabling this bit, all accesses to the SPI module are possible.

To power down the module, the following procedure should be followed.

1. Ensure all data transfers have taken place. I.e.the transmit buffer (or fifos) should be empty and the receive buffer (or fifos) should have been read until they are empty.
2. Disable all DMA requests and Interrupt requests. Disable fifo mode.
3. Disable appropriate SPI bit in Clock Control 1 (CC1) Register.

Section 18 ATAPI

18.1 General Description

ATAPI i/f provides both ATA and ATAPI physical interface. This unit also supports both ATA task command and ATAPI packet command.

18.2 Features

- Primary channel support
- Master/slave support
- 3.3V I/O interface
- PIO mode 0 to 4, Multiword DMA mode 0 to 2, UltraDMA mode 0 to 2 support

18.3 External Interface

Table 18.1 Pin Description

Signal	(ATAPI specification)	Function	Direction
AT_DSD[15:0]	(DD(15:0))	Bi-directional data bus	IN/OUT
AT_DSA[2:0]	(DA(2:0))	Address Bus	OUT
AT_$\overline{\text{DMACK0}}$	(DMACK)	Primary Channel DMA acknowledge	OUT
AT_DMARQ0	(DMARQ)	Primary channel DMA request	IN (schmitt)
AT_$\overline{\text{DCS}}$[1:0]	(CS0-, CS1-)	Primary channel chip select	OUT
AT_$\overline{\text{DIOW}}$	(DIOW-)	Primary channel disk write	OUT
AT_$\overline{\text{DIOR}}$	(DIOR-, HDMARDY-, HSTROBE)	Primary channel disk read	OUT
AT_DCHRDY0	(IORDY, DDMARDY-, DSTROBE)	Primary channel ready signal	IN (schmitt)
AT_DIRQ1	(INTRQ)	Primary channel interrupt request *	IN (schmitt)
AT_$\overline{\text{RESET}}$	(RESET-)	Primary channel ATAPI device reset (active low)	OUT

Note: * ATAPI i/f treats the interrupt signal from the ATAPI device as a level-triggered input.

18.4 Block Diagram

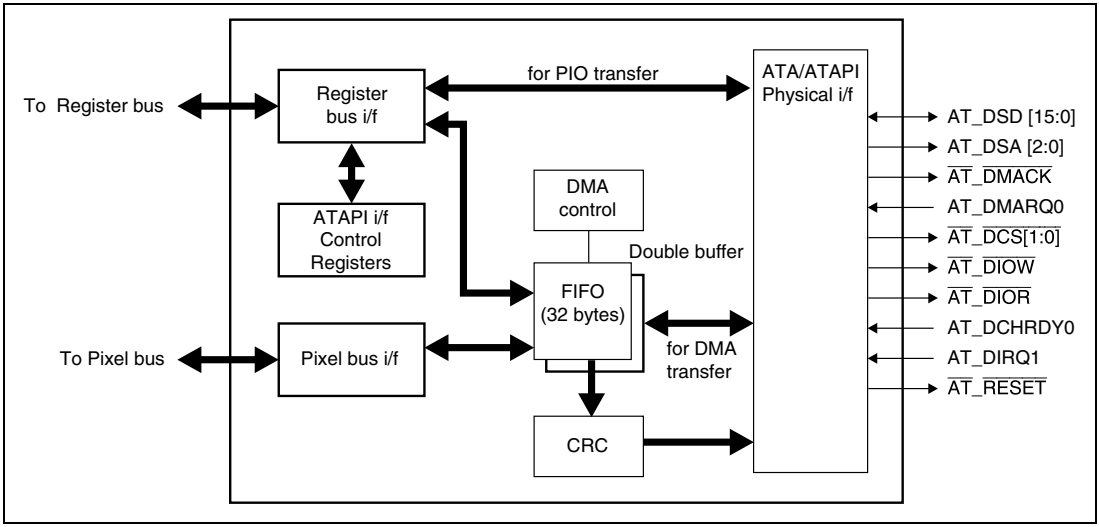


Figure 18.1 ATAPI Block Diagram

18.5 Register Description

There will be a set of registers which will be located in the address space of the PCI or MPX bus and will be located in the PCI memory space.

18.5.1 ATAPI Interface Registers

Table 18.2 ATA Task File Register Map

(These resisters are located in the ATAPI /ATA device , are not located in HD64404 ATAPI module.)

Offset	Read Register	Write Register	Pin address (AT_DCS[1:0], AT_DSA[2:0]) H: High Level L: Low Level @3.3V I/O	Access Size*[1] (Available Bit Size)	Register Location
H'00	Data	Data	HL-LLL/HH-XXX (X: don't care)	32 (16)*[2]	Drive
H'04	Error	Features	HL-LLH	32 (8)*[3]	Drive
H'08	Sector Count	Sector Count	HL-LHL	32 (8)*[3]	Drive
H'0C	Sector Number	Sector Number	HL-LHH	32 (8)*[3]	Drive
H'10	Cylinder Low	Cylinder Low	HL-HLL	32 (8)*[3]	Drive
H'14	Cylinder High	Cylinder High	HL-HLH	32 (8)*[3]	Drive
H'18	Device/Head	Device/Head	HL-HHL	32 (8)*[3]	Drive
H'1C	Status	Command	HL-HHH	32 (8)*[3]	Drive
H'38	Alternate Status	Device Control	LH-HHL	32 (8)*[3]	Drive

Notes: *1 CPU must access those registers by 32 bits (longword) access. Byte or word access is prohibited.

*2 Bits 15 down to 0 on the external 32-bit PCI/MPX data bus are used.

*3 Bits 7 down to 0 on the external 32-bit PCI/MPX data bus are used.

Table 18.3 ATAPI Packet Command Task File Register Map

(These resisters are located in the ATAPI /ATA device , are not located in HD64404 ATAPI module.)

Offset	Read Register	Write Register	Pin address (AT_DCS[1:0], AT_DSA[2:0])	Access Size*[1] (Available Bit Size)	Register Location
H'00	Data	Data	HL-LLL	32 (16)*[2]	Drive
H'04	Error	Features	HL-LLH	32 (8)*[3]	Drive
H'08	Interrupt Reason	—	HL-LHL	32 (8)*[3]	Drive
H'0C	—	—	HL-LHH	32 (8)*[3]	Drive
H'10	Byte Count Low	Byte Count Low	HL-HLL	32 (8)*[3]	Drive
H'14	Byte Count High	Byte Count High	HL-HLH	32 (8)*[3]	Drive
H'18	Device select	Device select	HL-HHL	32 (8)*[3]	Drive
H'1C	Status	Command	HL-HHH	32 (8)*[3]	Drive
H'38	Alternate Status	Device Control	LH-HHL	32 (8)*[3]	Drive

Notes: *1 CPU must access those registers by 32bits (longword) access. Byte or word access is prohibited.

*2 Bits 15 down to 0 on the external 32-bit PCI/MPX data bus are used.

*3 Bits 7 down to 0 on the external 32-bit PCI/MPX data bus are used.

Table 18.4 ATAPI i/f Control Register Map

(These resisters are located in HD64404 ATAPI module.)

Offset	Register Name	Access Type	Register Access Size (*)	Register Location
H'80	ATAPI Control	Read/Write	32	HD64404
H'84	ATAPI Status	Read/Write	32	HD64404
H'88	Interrupt Enable	Read/Write	32	HD64404
H'8C	PIO Timing	Read/Write	32	HD64404
H'90	Multiword DMA Timing	Read/Write	32	HD64404
H'94	Ultra DMA Timing	Read/Write	32	HD64404
H'98	Reserved	Read	32	HD64404
H'9C	DMA Start Address	Read/Write	32	HD64404
H'A0	DMA Transfer Count	Read/Write	32	HD64404
H'A4	ATAPI Control 2	Read/Write	32	HD64404
H'A8	Reserved	Read	32	HD64404
H'AC	Reserved	Read	32	HD64404
H'B0	ATAPI Signal Status	Read	32	HD64404
H'B4	Data Transfer Mode	Read/Write	32	HD64404
H'BC	Byte swap	Read/Write	32	HD64404
H'C0-FC	FIFO data	Read	32	HD64404

Note: * CPU must access those registers by 32 bits (longword) access. Byte or word access is prohibited.

18.5.2 ATA Task File Register

Note that these registers are located in ATAPI/ATA device side and not all of ATAPI/ATA devices support these registers.

This reflects ATAPI 5 specification.

(1) Data Register

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	L	L

This register is used for host PIO data transfer only.

Bit	Bit Name	Description
15 to 0	Data[15:0]	

(2) Data Port

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	H	X	X	X

When DMACK- is asserted, $\overline{DCS[1]}$ and $\overline{DCS[0]}$ is deasserted.

This register is used for host DMA data transfers.

Bit	Bit Name	Description
15 to 0	Data[15:0]	

(3) ERROR Register

This register is readable only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	L	H

Bit	Bit Name	Description
7 to 3	—	The content of the register depends on specific command
2	ABRT	ABRT is Command Aborted.
1, 0	—	The content of the register depends on specific command

(4) Features Register

This register is writable only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	L	H

Bit	Bit Name	Description
7 to 0	—	All bits are command dependent.

(5) Sector Count Register

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	H	L

Bit	Bit Name	Description
7 to 0	—	All bits are command dependent.

(6) Sector Number Register

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	H	H

Bit	Bit Name	Description
7 to 0	—	All bits are command dependent.

(7) Cylinder Low Register

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	L	L

This register depends on the command.

Bit	Bit Name	Description
7 to 0	—	All bits are command dependent.

(8) Cylinder High Register

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	L	H

This register depends on the command.

Bit	Bit Name	Description
7 to 0	—	All bits are command dependent

(9) Device/Head Register

This register is readable/writable.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	H	L

Bit	Bit Name	Description
7	obsolete	These bits are obsolete.
6	—	
5	obsolete	These bits are obsolete.
4	DEV	DEV is Device select.
3 to 0	—	

(10) Status Register

This register is readable only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	H	H

Bit	Bit Name	Description
7	BSY	BSY indicates the device is busy.
6	DRDY	DRDY is Device Ready.
5, 4	—	
3	DRQ	DRQ is Data request
2, 1	obsolete	
0	ERR	ERR is that an error occurred during execution.

(11) Command Register

This register is writable only.

- Address pins

Pin	AT_$\overline{CS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	H	H

This register has a command code which sends to the device.

Bit	Bit Name	Description
7 to 0	Command Code	

(12) Alternate Status Register

This register is readable only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	L	H	H	H	L

This register has the same value as the Status Register in the command block.

Bit	Bit Name	Description
7 to 0	Status[7:0]	

(13) Device Control Register

This register is only writable when $\overline{DMACK}$ is not asserted.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	L	H	H	H	L

Bit	Bit Name	Description
7 to 3	R	R indicates Reserved.
2	SRST	SRST is for software reset.
1	nIEN	nIEN is for enable or disable the assertion of the INTRQ.
0	0	

18.5.3 ATAPI Packet Command Task File Register

Note that these registers are located in ATAPI device side and not all of ATAPI/ATA devices support these registers.

This reflects ATAPI 5 specification.

(1) ERROR Register

This register is readable only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	L	H

Bit	Bit Name	Description
7 to 4	Sense Key	Sense Key is a specific error indication.
3	na	"na" indicates the content of a bit or field is not a applicable to the particular command.
2	ABRT	ABRT is Abort.
1	EOM	EOM indicates to detect the end of the media.
0	ILI	ILI is illegal length indication.

(2) Features Register

This register is write only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	L	H

Bit	Bit Name	Description
7 to 2	na	
1	OVL	OVL is to indicate that Packet command is to be overlapped.
0	DMA	DMA indicates that all data transfer is done by DMA except Command Packet transfer.

(3) Interrupt Reason Register

This register is read/write.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	L	H	L

Bit	Bit Name	Description
7 to 3	na	
2	REL	REL is to indicate that device released ATA bus before the command is finished.
1	I/O	I/O is the direction of the data transfer. 1: from device to host 0: from host to device
0	C/D	C/D is Command or Data. 1: Command 0: Data

(4) Byte Count Low Register

This register is read/write.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	L	L

This register is used for only PIO mode.

Bit	Bit Name	Description
7 to 0	Byte Count[7:0]	

(5) Byte Count High Register

This register is read/write.

- Address pins

Pin	AT_$\overline{\text{DCS}}$[1]	AT_$\overline{\text{DCS}}$[0]	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	L	H

This register is used for only PIO mode.

Bit	Bit Name	Description
7 to 0	Byte Count[15:8]	

(6) Device Select Register

This register is read/write.

- Address pins

Pin	AT_$\overline{\text{DCS}}$[1]	AT_$\overline{\text{DCS}}$[0]	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	H	L

Bit	Bit Name	Description
7	obsolete	These bits are obsolete.
6	—	The content of the register depends on specific command
5	obsolete	These bits are obsolete.
4	DEV	DEV is Device select.
3 to 0	—	The content of the register depends on specific command

(7) Status Register

This register is read only.

- Address pins

Pin	AT_$\overline{\text{DCS}}$[1]	AT_$\overline{\text{DCS}}$[0]	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	H	H

Bit	Bit Name	Description
7	BSY	BSY indicates the device is busy.
6	DRDY	DRDY is Device Ready.
5	DMRD	DMRD is DMA ready.
4	SERV	SERV is Service request.
3	DRQ	DRQ is Data request
2, 1	na	
0	CHK	CHECK is that an error occurred during execution.

(8) Command Register

This register is write only.

- Address pins

Pin	AT_$\overline{\text{DCS}}$[1]	AT_$\overline{\text{DCS}}$[0]	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	H	L	H	H	H

This register has a command code which sends to the device.

Bit	Bit Name	Description
7 to 0	Command Code	

(9) Alternate Status Register

This register is read only.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	L	H	H	H	L

This register has the same value as the Status register in the command block .

Bit	Bit Name	Description
7 to 0	Status[7:0]	

(10) Device Control Register

This register is only write when DMACK- is not asserted.

- Address pins

Pin	AT_$\overline{DCS[1]}$	AT_$\overline{DCS[0]}$	AT_DSA[2]	AT_DSA[1]	AT_DSA[0]
Level	L	H	H	H	L

Bit	Bit Name	Description
7 to 3	R	R indicates Reserved.
2	SRST	SRST is for software reset.
1	nIEN	nIEN is for enable or disable the assertion of the INTRQ.
0	0	

18.5.4 ATAPI I/F Control Register Map

Legends for register description:

Initial value : Register value after reset
— : Read: undefined value, Write: always 0 write
R/W : Read and write register
R/WC0 : Read and write register, 0 write clear the register, 1 write is ignored.
R : Read only register, for write always 0 write

All control/status registers are active high.

(1) ATAPI Control

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIO FIFO	RESET	M/S	BUS SEL	UDM AEN		R/W	STOP	START
Initial:	-	-	-	-	-	-	-	0	0	0	0	0	-	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	—	R	Reserved
8	PIO FIFO	0	R/W	PIO FIFO is PIO transfer enable where 32bytes × 2 FIFO buffer is used. When the register bus DMA of ATAPI interface module is executed, this bit is effective. But if UDMAEN = 1 then PIOFIFO value is ignored.
7	RESET	0	R/W	RESET is ATAPI device reset. If this bit is set to '1' then ATAPI reset signal is asserted. AT_$\overline{\text{RESET}}$ is active low. When this bit is set to '1' then AT_$\overline{\text{RESET}}$ is low. When this bit is set to '0' then AT_$\overline{\text{RESET}}$ is high.
6	M/S	0	R/W	M/S is ATAPI device MASTER or SLAVE selection. 1 = MASTER, 0 = SLAVE.
5	BUSSEL	0	R/W	BUSSEL is Pixel bus or REGISTER bus selection when DMA. 1 = Pixel bus, 0 = REGISTER bus.
4	UDMAEN	0	R/W	UDMAEN is Ultra DMA enable. When Ultra DMA is used, set this bit to '1'. Set it to '0' when Multiword DMA or PIO FIFO mode.
3	—	—	R	Reserved
2	R/W	0	R/W	R/W is FIFO read/write. 1 = read, 0 = write. Set this bit to '1' when reading data from ATAPI device . Set it to '0' when writing data to ATAPI device.

Bit	Bit Name	Initial Value	R/W	Description
1	STOP	0	R/W	STOP is DMA forced stop. • When writing 0: Ignored 1: Forced termination of data transfer • When reading 0: The forced termination command is not issued. 1: Forced termination of data transfer command is issued. It will become '0' when the next DMA starts. Note: Transfer cannot be restarted from the address at which DMA transfer has been forcibly stopped.
0	START	0	R/W	START is DMA start. If this bit set to '1' then the DMA transfer is started. '0' writing is ignored. • When writing 0: Ignored 1: DMA transfer start • When reading 0: DMA transfer is not active 1: Busy in transfer Note: Must not access Task File Register while DMA is active.

(2) ATAPI Status

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								SWERR	IFERR		DEVTRM	DEVINT	TOUT	ERR	NEND	ACT
Initial:	-	-	-	-	-	-	-	0	0	-	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/WC0	R/WC0	R	R/WC0	R	R/WC0	R/WC0	R/WC0	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	—	R	Reserved
8	SWERR	0	R/WC0	SWERR is software error. It indicates that Task File register access is detected while DMA is active. It is prohibited. Writing 0 resets this register.
7	IFERR	0	R/WC0	IFERR indicates that ATAPI interface protocol error is detected. In other words, 1. (AT_DMARQ0 = 1) or (AT_DCHRDY0 = 0) when the ULTRA DMA data-in burst is in the host termination. 2. AT_DCHRDY0 = 0 when the ULTRA DMA data-out burst is in the device termination. 3. AT_DCHRDY0 = 0 when the ULTRA DMA data-out burst is initiated. 4. (AT_DMARQ0 = 1) or (AT_DCHRDY0 = 0) when the ULTRA DMA data-out burst in the host termination. Writing 0 resets this register.
6	—	—	R	Reserved
5	DEVTRM	0	R/WC0	DEVTRM is set to 1 when the ATAPI device is terminated in ULTRA DMA mode while the number of DMA transfer bytes does not reach the value set in this ATAPI module. Writing 0 resets this register.
4	DEVINT	0	R	DEVINT is ATAPI device interrupt AT_DIRQ1 status. This register is read only. Since this register doesn't hold its status in HD64404 chip, if AT_DIRQ 1 becomes 0, this register will also become 0. ATAPI i/f treats the interrupt signal from the ATAPI device as a level-triggered input. According to ATAPI standard, AT-DIRQ1 will be negated by the ATAPI device within 400 ns of the negation of AT_$\overline{\text{DIOR}}$ that reads the Status register to clear interrupt pending.
3	TOUT	0	R/WC0	TOUT indicates that IORDY timeout is detected. Timeout is detected if no response is returned in 150 cycles or longer of Pixel Bus clock cycle time. Writing 0 resets this register.
2	ERR	0	R/WC0	DMA abort occurs, ERR is set to '1' by writing '1' to STOP bit in the ATAPI Control register. Writing 0 resets this register.

Bit	Bit Name	Initial Value	R/W	Description
1	NEND	0	R/WC0	NEND is DMA normal end. Writing 0 resets this register.
0	ACT	0	R	ACT is DMA active. This register is read only. This register is cleared when DMA is completed. It is not recommended to use it as a source of interrupt.

(3) Interrupt Enable

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								iSWERR	IIFERR		iDEVTRM	iDEVINT	iTOUT	iERR	iNEND	iACT
Initial:	-	-	-	-	-	-	-	0	0	-	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	—	R	Reserved
8	iSWERR	0	R/W	iSWERR is SWERR interrupt enable.
7	iIFERR	0	R/W	iIFERR is IFERR interrupt enable.
6	—	—	R	Reserved
5	iDEVTRM	0	R/W	iDEVTRM is DEVTRM interrupt enable
4	iDEVINT	0	R/W	iDEVINT is DEVINT interrupt enable.
3	iTOUT	0	R/W	iTOUT is TOUT interrupt enable.
2	iERR	0	R/W	iERR is ERR interrupt enable.
1	iNEND	0	R/W	iNEND is NEND interrupt enable.
0	iACT	0	R/W	iACT is ACT interrupt enable. Since ACT is cleared automatically when DMA is completed, it is not recommended to set 1.

Note: Write 1 to each bit is to enable the interrupt signal of the related ATAPI status register bit.

(4) PIO Timing

Set the machine cycle numbers to the following bits before the access to ATAPI device.

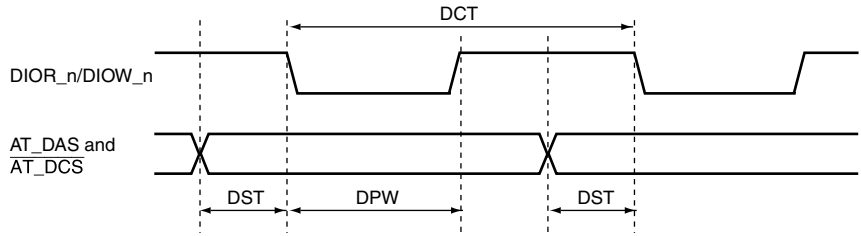
The machine cycle is a pixel bus clk.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			pSDCT						pSDPW					pSDST		
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			pMDCT						pMDPW					pMDST		
Initial:	-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	—	R	Reserved
29 to 24	pSDCT	—	R/W	pSDCT is the cycle time of Slave ATAPI device.
23 to 19	pSDPW	0	R/W	pSDPW is the AT_$\overline{\text{DIOR}}$/AT_$\overline{\text{DIOW}}$ pulse width of Slave ATAPI device.
18 to 16	pSDST	0	R/W	pSDST is the address setup time to AT_$\overline{\text{DIOR}}$/ AT_$\overline{\text{DIOW}}$ for slave ATAPI device in PIO mode.
15, 14	—	—	R	Reserved
13 to 8	pMDCT	0	R/W	pMDCT is the cycle time of Master ATAPI device.
7 to 3	pMDPW	0	R/W	pMDPW is the AT_$\overline{\text{DIOR}}$/AT_$\overline{\text{DIOW}}$ pulse width of Master ATAPI device.
2 to 0	pMDST	0	R/W	pMDST is the address setup time to AT_$\overline{\text{DIOR}}$/ AT_$\overline{\text{DIOW}}$ for master ATAPI device in PIO mode.

The timing values for DCT, DST and DPW are determined by multiplying the value in each register by the pixel clock cycle time.



DCT : data transfer cycle time
DPW : Low pulse width of DIOR_n/DIOW_n
DST : Setup time to ATA address and DIOR_n/DIOW_n

Note: The prefix pS is for setting the slave side while pM is for the master.
The timing values for DCT, DST and DPW are determined by multiplying the value in each register by the pixel clock cycle time.

Figure 18.2 PIO timing register

- PIO timing value table (Master / Slave)

Pixel bus clk	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4
66 MHz	H'29AE	H'1AAC	H'17AB	H'0D3B	H'0933
78 MHz	H'30C7	H'1FC5	H'1BC4	H'0F44	H'0B3B
83 MHz	H'33CF	H'21CD	H'1DCC	H'1044	H'0B3B
88 MHz	H'36D8	H'23DE	H'1EDC	H'114C	H'0C44
92 MHz	H'39E0	H'25E6	H'20E4	H'124C	H'0C44
96 MHz	H'3BE8	H'26EE	H'21EC	H'134C	H'0D44
99 MHz	H'3DF0	H'27F6	H'22F4	H'134C	H'0D44
100 MHz	H'3DF0	H'28F6	H'22F4	H'134C	H'0D44

Ex.) If (pixel bus clk=99MHz) and (Slave = mode 0) and (Master = mode 1) then PIO timing = '3DF027F6'

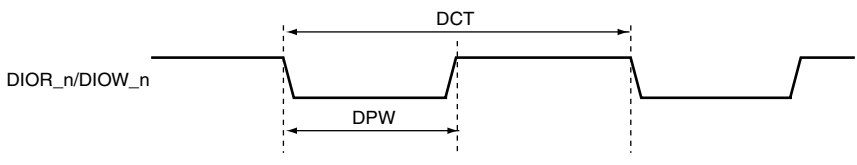
(5) Multiword DMA Timing

Set the machine cycle numbers to the following bits before the access to ATAPI device.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						mSDCT						mSDPW				
Initial:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						mMDCT						mMDPW				
Initial:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	Reserved
26 to 21	mSDCT	0	R/W	mSDCT is the cycle time of Slave ATAPI device.
20 to 16	mSDPW	0	R/W	mSDPW is the AT_$\overline{\text{DIOR}}$/AT_$\overline{\text{DIOW}}$ pulse width of Slave ATAPI device.
15 to 11	—	—	R	Reserved
10 to 5	mMDCT	0	R/W	mMDCT is the cycle time of Master ATAPI device.
4 to 0	mMDPW	0	R/W	mMDPW is the AT_$\overline{\text{DIOR}}$/AT_$\overline{\text{DIOW}}$ pulse width of Master ATAPI device.



DCT : data transfer cycle time
DPW : Low pulse width of DIOR_n/DIOW_n

Note: The prefix mS is for slave while mM is for the master.
The timing values for DCT and DPW are determined by multiplying the value in each register by the pixel clock cycle time.

Figure 18.3 Multiword DMA timing register

- Multi word DMA timing value table

Pixel bus clk	Mode 0	Mode 1	Mode 2
66 MHz	H'042F	H'0166	H'0126
78 MHz	H'04F2	H'01A8	H'0167
83 MHz	H'0533	H'01C8	H'0167
88 MHz	H'0594	H'01E8	H'0188
92 MHz	H'05D5	H'01E9	H'0188
96 MHz	H'0616	H'0209	H'01A8
99 MHz	H'0637	H'0209	H'01A8
100 MHz	H'0637	H'0209	H'01A8

Ex.) If (pixel bus clk=99MHz) and (Slave = mode 0) and (Master = mode 1) then Multi word DMA timing = '06370209'

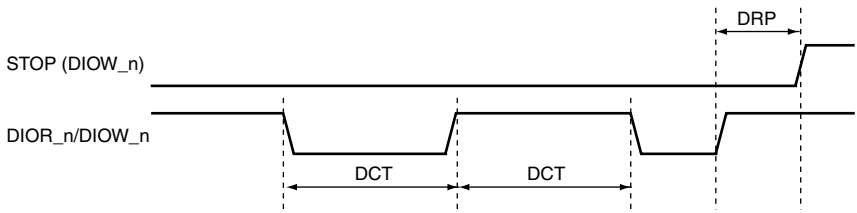
(6) Ultra DMA Timing

Set the machine cycle numbers to the following bits before the access to ATAPI device.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								uSDCT				uSDRP				
Initial:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								UMDCT				uMDRP				
Initial:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	—	R	Reserved
24 to 21	uSDCT	—	R/W	uSDCT is the cycle time of Slave ATAPI device.
20 to 16	uSDRP	0	R/W	uSDRP is the time from negating DMARDY (Not AT_DCHRDY0) until pause by slave ATAPI device.
15 to 9	—	—	R	Reserved
8 to 5	uMDCT	0	R/W	uMDCT is the cycle time of Master ATAPI device.
4 to 0	uMDRP	0	R/W	uMDRP is the time from negating DMARDY (Not AT_DCHRDY0) until pause by master ATAPI device.



DCT : data transfer cycle time
DRP : Time from negating DMARDY (DIOR_n) to asserting STOP (DIOW_n) used in data_in_burst mode

Note: The prefix uS is for slave while uM is for the master.
The timing values for DCT and DRP are determined by multiplying the value in each register by the pixel clock cycle time.

Figure 18.4 Ultra DMA timing register

- Ultra DMA timing value table

Pixel bus clk	Mode 0	Mode 1	Mode 2
66 MHz	H'010C	H'00C9	Not Available(*)
78 MHz	H'014E	H'00EB	H'00A9
83 MHz	H'014F	H'00EC	H'00AA
88 MHz	H'016F	H'010C	H'00CA
92 MHz	H'0190	H'010D	H'00CB
96 MHz	H'0191	H'010D	H'00CB
99 MHz	H'0191	H'010E	H'00CB
100 MHz	H'0191	H'010E	H'00CB

Ex.) If (pixel bus clk=99MHz) and (Slave = mode 0) and (Master = mode 1) then Ultra DMA timing = '0191010E'

Note: * Minimum pixel bus frequency for Ultra DMA mode 2 is 75MHz.

(7) DMA Start Address

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						DSTA[26:16]										
Initial:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DSTA[15:2]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	Reserved
26 to 2	DSTA[26:2]	0	R/W	DSTA is DMA start address that indicates the data transfer start address in Graphic Memory. Bits 26 to 0 are used to specify DMA start address in byte. Since 32-bit address boundary must be kept for DMA start address, bit 1 and 0 are ignored.
1, 0	—	—	R	Reserved

Notes: 1. This register is valid only when bit 5 (BUSSEL) of the ATAPI Control Register is 1.
2. This address does not change and the set value is retained even after DMA activation.

(8) DMA Transfer Count

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
						DTRC [26:16]										
Initial:	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTRC [15:1]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	—	R	Reserved
26 to 1	DTRC[26:1]	0	R/W	DTRC is DMA transfer count. Bits 26 to 1 are used to specify the number of transfer words.
0	—	—	R	Reserved

Notes: 1. If bit 5 (BUSSEL) of the ATAPI Control Register is 0, the set value of this register must be the same as the value set to DMA Length of the DMA n Length register in the DMAC Block.

2. This count value does not change and the set value is retained even after DMA activation.

(9) ATAPI Control 2

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															WORDSWAP	IFEN
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	—	R	Reserved
1	WORDSWAP	0	R/W	WORDSWAP is to swap the upper 16-bit data and the lower 16-bit data when 32-bit data bus is enable in pixel bus or register bus. 0: Word swap is not executed. 32-bit Data on the pixel/register bus appears in a big endian format. 1: Word swap is executed between ATAPI interface and register/pixel bus interface. 32 bit Data on the pixel/register bus appears in a little endian format. Note that wordswap is only available on Data transfer when ATAPI Control Register[0]=1: DMA mode start. Other than DMA, all register accesses are LW access.
0	IFEN	0	R/W	IFEN is ATAPI interface enable. 0: ATAPI interface disable 1: ATAPI interface enable Note: ATAPI interface I/O pins function as input, and output pins goes Hi-Z.

(10) ATAPI Signal Status

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															DDMARDY	DMARQ
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	—	R	Reserved
1	DDMARDY	—	R	DDMARDY is ATAPI DDMARDY (Not AT_DCHRDY0) signal status.
0	DMARQ	—	R	DMARQ is ATAPI DMARQ (AT_DMARQ0) signal status.

(11) Data Transfer Mode

This register is only available for the pixel bus DMA transfer.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											MWX1	MWX0			GBM	TileEN
Initial:	-	-	-	-	-	-	-	-	-	-	0	0	-	-	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	—	R	Reserved
5	MWX1	0	R/W	**Memory width for Data transfer write (MWX)**
4	MWX0	0	R/W	The Memory width for image data in case of DMA to Graphic Memory. 00: 512 pixels. 01: 1024 pixels 10: 2048 pixels 11: 4096 pixels
3, 2	—	—	R	Reserved
1	GBM	0	R/W	**Data transfer graphic bit mode (GBM)** 0: 16 bits/pixel 1: 8 bits/pixel
0	TileEN	0	R/W	**Data transfer to Tile Space or Linear Space (TileEn)** 0: Linear Space 1: Tiled Space In order to use Tiled space as Graphic Memory, set TileEn = 1 and DMA start address as 32-byte address boundary.

Note: If TileEN bit zero, the value of MWX1 and MWX0 bits is ignored.

Correspondence between Graphic Memory Physical Addresses (bytes) and Rendering Coordinates and Multi-valued Source Coordinates when TileEN = 1.

8 bits/pixel (GBM=1), 512 pixels (MWX = 0) Y (vertical) address = A[26:9], X(horizontal) address = A[8:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:13]														A[8:5]				A[12:9]				A[4:2]			0	0

8 bits/pixel (GBM=1) , 1024 pixels (MWX = 1) Y(vertical) address = A[26:10], X(horizontal) address = A[9:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:14]														A[9:5]				A[13:10]				A[4:2]			0	0

8 bits/pixel (GBM=1), 2048 pixels (MWX = 2) Y(vertical) address = A[26:11], X(horizontal) address = A[10:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:15]														A[10:5]				A[14:11]				A[4:2]			0	0

8 bits/pixel (GBM=1), 4096 pixels (MWX = 3) Y (vertical) address = A[26:12], X(horizontal) address = A[11:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:16]														A[11:5]				A[15:12]				A[4:2]			0	0

16 bits/pixel (GBM=0), 512 pixels (MWX = 0) Y (vertical) address = A[26:10], X(horizontal) address = A[9:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:14]														A[9:5]				A[13:10]				A[4:2]			0	0

16 bits/pixel (GBM=0), 1024 pixels (MWX = 1) Y (vertical) address = A[26:11], X(horizontal) address = A[10:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:15]														A[10:5]				A[14:11]				A[4:2]			0	0

16 bits/pixel (GBM=0), 2048 pixels (MWX = 2) Y (vertical) address = A[26:12], X(horizontal) address = A[11:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:16]											A[11:5]							A[15:12]				A[4:2]			0	0

16 bits/pixel (GBM=0), 4096 pixels (MWX = 3) Y (vertical) address = A[26:13], X(horizontal) address = A[12:0]

A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
A[26:17]										A[12:5]								A[16:13]				A[4:2]			0	0

Upper line: Memory physical addresses (bytes)
Lower line: Logical coordinates (X, Y)

(12) Byteswap

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																BYTE SWAP
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	—	R	Reserved
0	BYTESWAP	0	R/W	BYTESWAP is to swap the upper 8 bit data and the lower 8 bit data in ATAPI i/f. BYTESWAP = 1: Byte swap is executed between ATAPI interface and register bus/pixel bus interface. Note that byteswap is only available on Data transfer when ATAPI control register[0]=1: DMA mode start.

(13) FIFO data

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFODATA[31:16]															
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFODATA[15:0]															
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	FIFODATA [31:0]	—	R	FIFODATA is FIFO data when DMA is stopped. Must not read while DMA is active.

18.6 Functional Description

ATAPI interface supports a primary channel as a host. Master/slave configuration is also supported as defined in ATAPI interface specification. Read/write FIFO buffers in ATAPI interface make up to 16Mbyte/sec data transfer for PIO and Multiword DMA mode possible in a normal operation but it also depends on the bus traffic of register bus or Pixel bus. It supports 3.3V I/O interface.

ATA Task files register and ATAPI Packet Command Task registers are mapped in Super H register map space. Therefore accessing those addresses by Super H is to access registers located in the device such as DVD ROM drive by addressing $\overline{\text{DCS}}$[1:0] and DSA[2:0] pins.

Data Transfer Mode

ATAPI i/f control register supports PIO transfer, Multiword DMA transfer, and Ultra DMA transfer mode. It initiates transfer modes and sets a specific ATAPI interface timing which is different from each mode.

PIO mode 0,1,2,3,4, Multiword DMA mode 0,1,2 and Ultra DMA mode 0,1,2 (Up to 33MB/s) are supported.

For both Multiword DMA and Ultra DMA data transfers, register bus or pixel bus can be used while register bus can be only used for PIO transfer.

Table 18.5 Data Transfer Mode

Bus selection	Source or destination for the pixel bus or register bus DMA	PIO Without FIFO	PIO with FIFO*[4]	Multiword DMA	Ultra DMA
Register bus	Graphic Memory/PCI external Device	A*[3]	A	A	A*[1]
Pixel bus	Graphic Memory	NA	NA	A	A
Data packing on the register bus *[2]		NA	A	A	A
PIOFIFO bit in ATAPI Control		Don't Care	1	0	Don't Care
UDMAEN bit in ATAPI Control		Don't Care	0	0	1
START/STOP bit in ATAPI Control		Not used	Used	Used	Used

Legend: A: Available
NA: Not Available

Notes: *1 Up to 16MB/s data transfer is achievable.
*2 Available case packs two 16bit data, AT_DSD[15:0], on 32bit register bus. For pixel bus, these two 16bit data are always packed. Data alignment can be changed by WORDSWAP and BYTESWAP register. Not available case put one 16bit data on 32bit register bus, at bit 15 down to 0 position.
*3 Only CPU PIO access is available. Register bus DMA can not be used .
*4 External ATAPI interface is in PIO mode and internally the register bus DMA is used .

For register bus DMA, one of the DMA channels in DMAC is initiated. For this mode, DMA transfer count in ATAPI Register that should be the same value as DMA transfer count in DMAC Register has to be set. (see DMAC specification) DMA Start Address Register in ATAPI i/f is ignored.

For pixel bus DMA, the data is transferred between ATAPI device and Graphic Memory.

Standby Mode

When entering standby mode, do not stop clock supply to the ATAPI module until DMA transfer ends between the ATAPI i/f and the Graphic Memory/PCI external device.

18.7 Required Termination

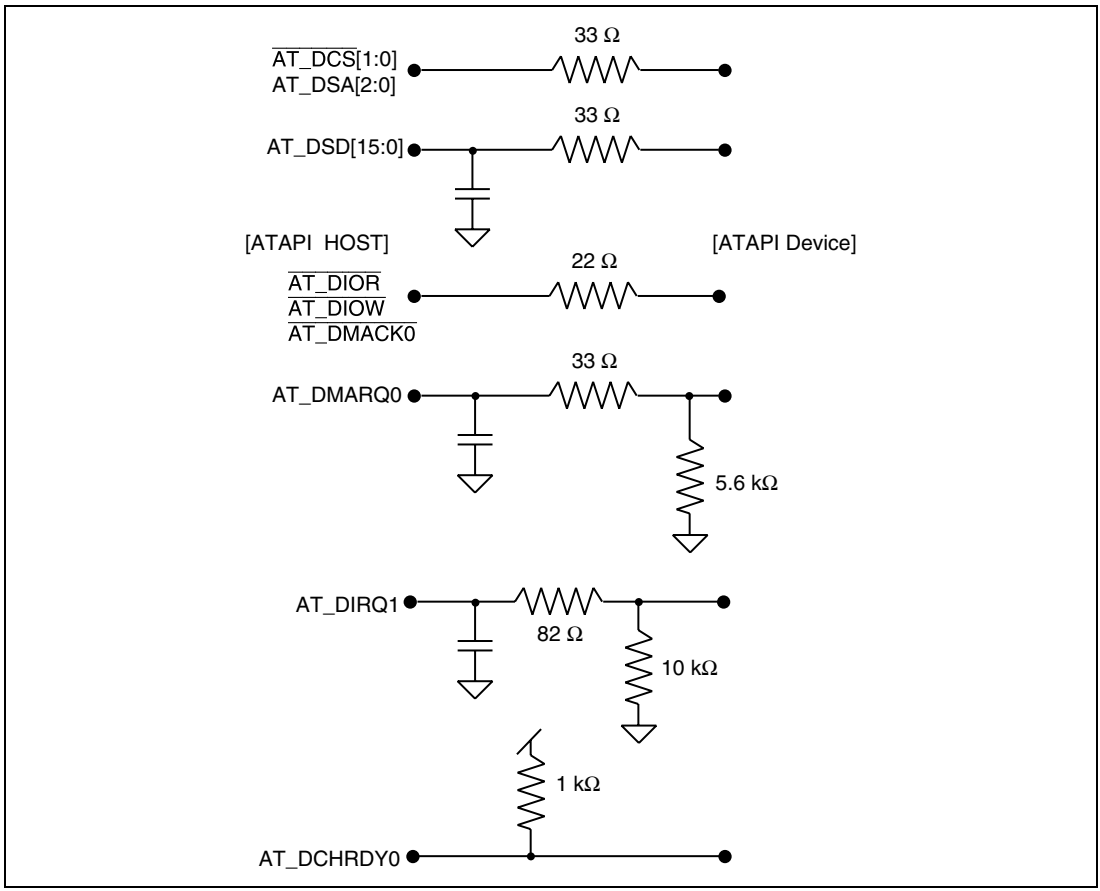


Figure 18.5 Required Termination

ATAPI i/f treats the interrupt signal from the ATAPI device as a level-triggered input

References

ATA/ATAPI-5 specification rev 2.0

18.8 Operating Procedure

18.8.1 Initialization

Setting of Interface Enable Bit

Write 1 to IFEN bit of ATAPI Control 2 Register.

Setting of Timing Registers

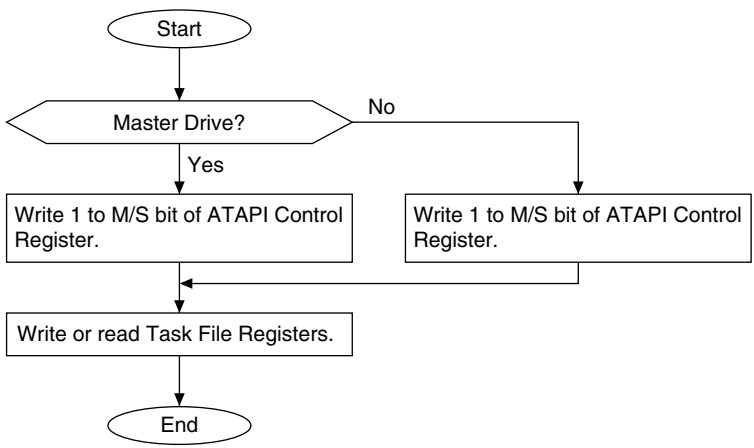
Write the appropriate value to the following registers.

Refer to Register Description for the value.

- PIO timing register
- Multiword DMA timing register
- Ultra DMA timing register

18.8.2 Procedure in PIO Transfer Mode

Case Not Using FIFO



Note: Must not execute PIO transfer while ACT bit of ATAPI Status Register is 1.

Figure 18.6 Procedure in PIO Transfer Mode (Not Using FIFO)

Case using FIFO by polling

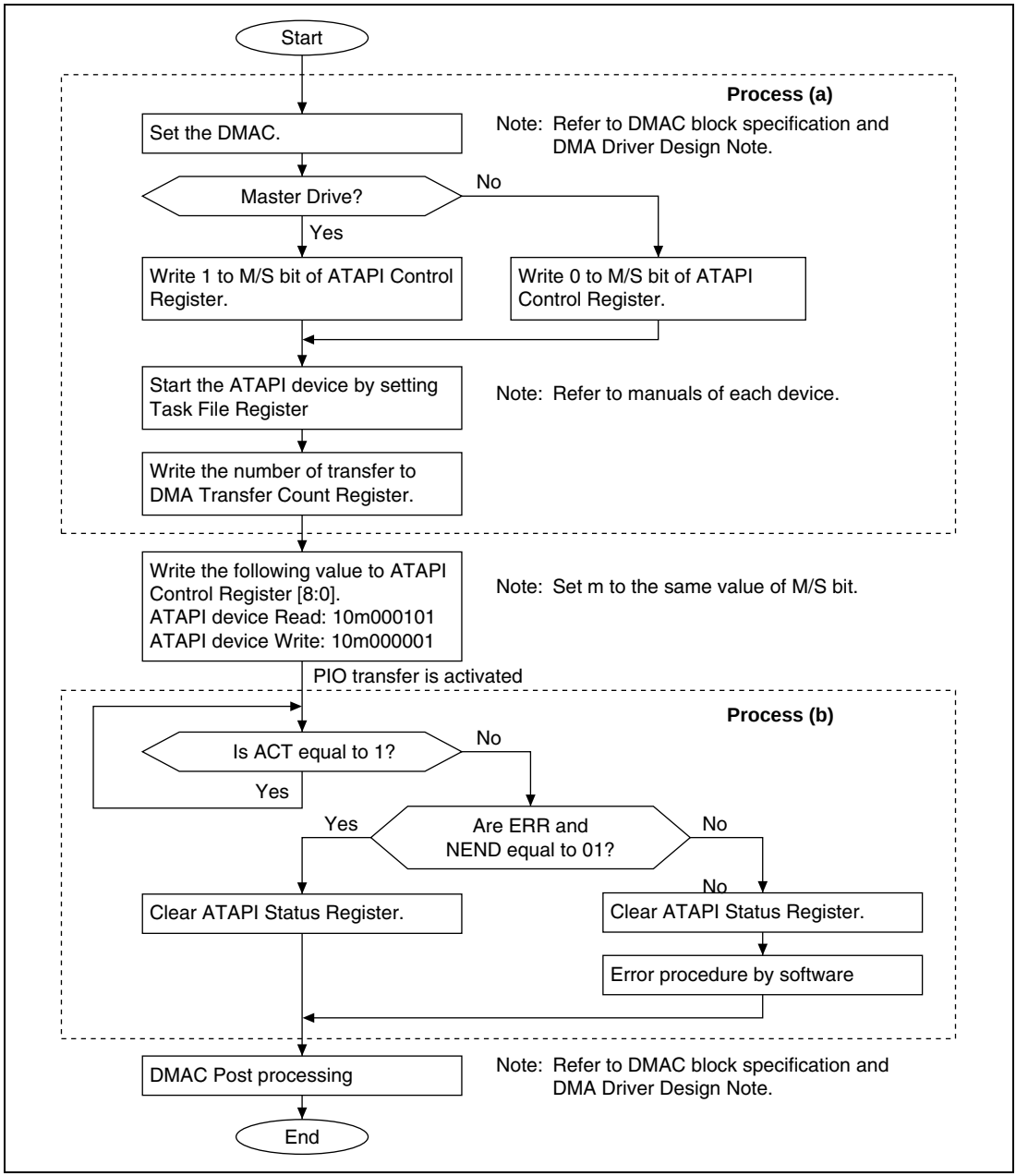


Figure 18.7 Procedure in PIO Transfer Mode (Using FIFO by Polling)

Case Using FIFO by Interrupt

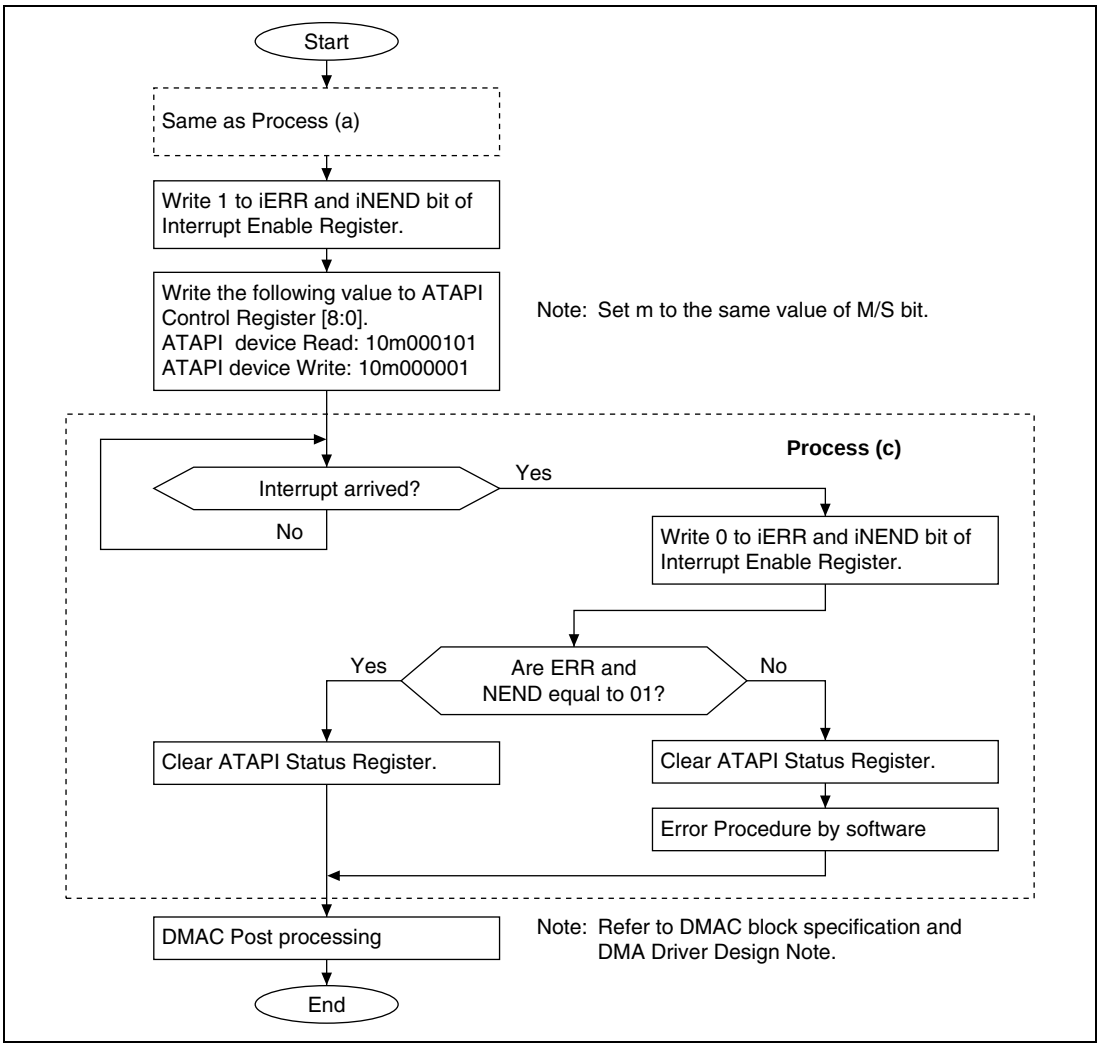


Figure 18.8 Procedure in PIO Transfer Mode (Using FIFO by Interrupt)

18.8.3 Procedure in Multi Word DMA Transfer Mode

Transfer from/to Peripherals on Register Bus by Polling

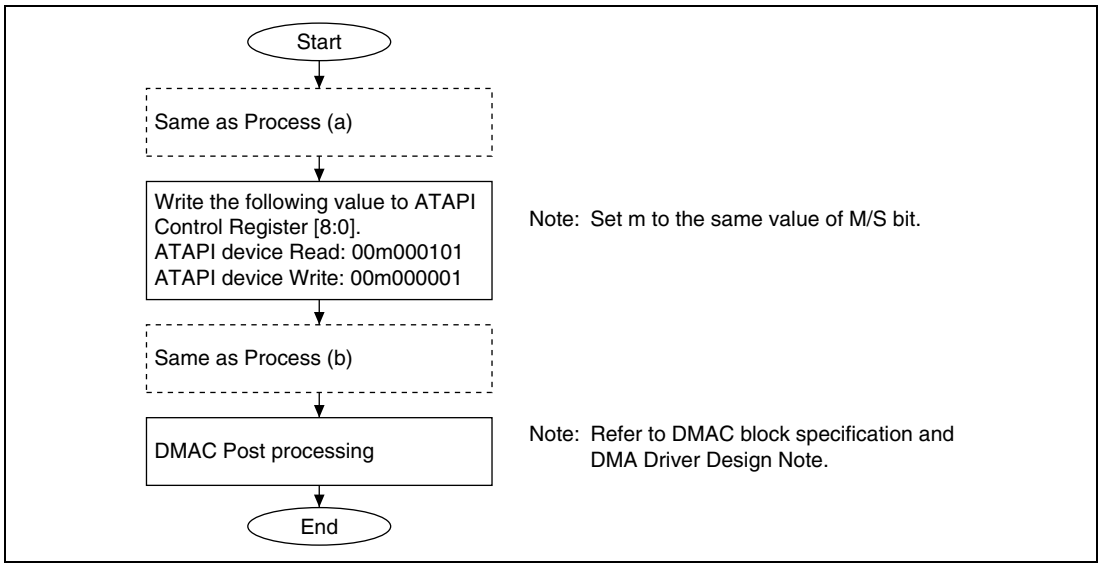


Figure 18.9 Transfer from/to Peripherals on Register Bus by Polling

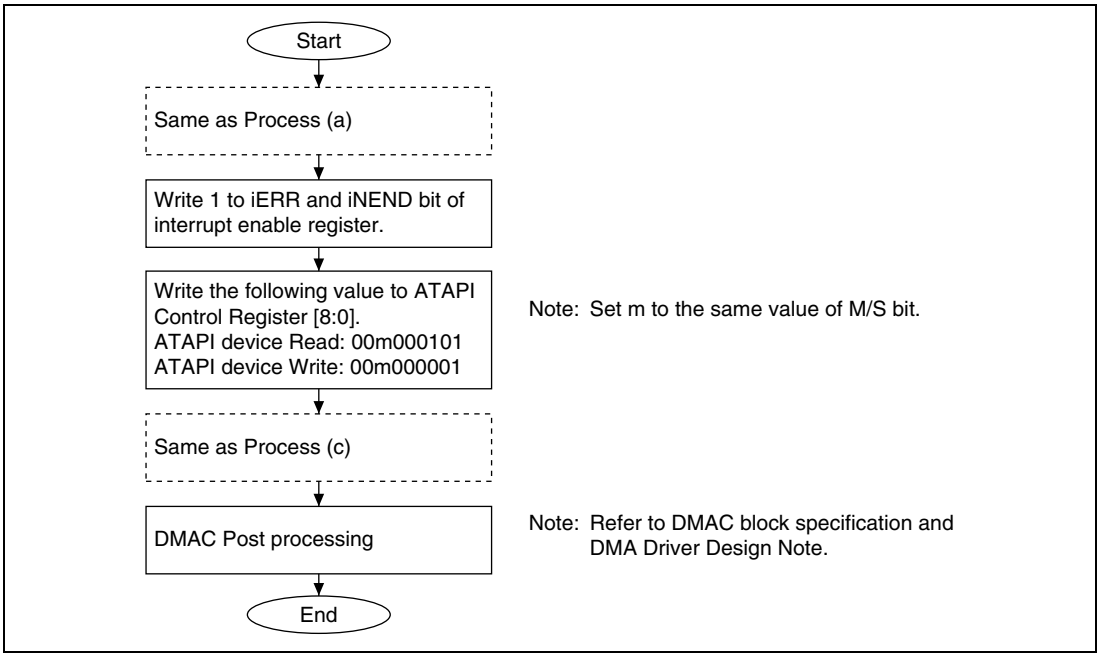


Figure 18.10 Transfer from/to Peripherals on Register Bus by Interrupt

Transfer from/to Graphic Memory through Pixel Bus by Polling

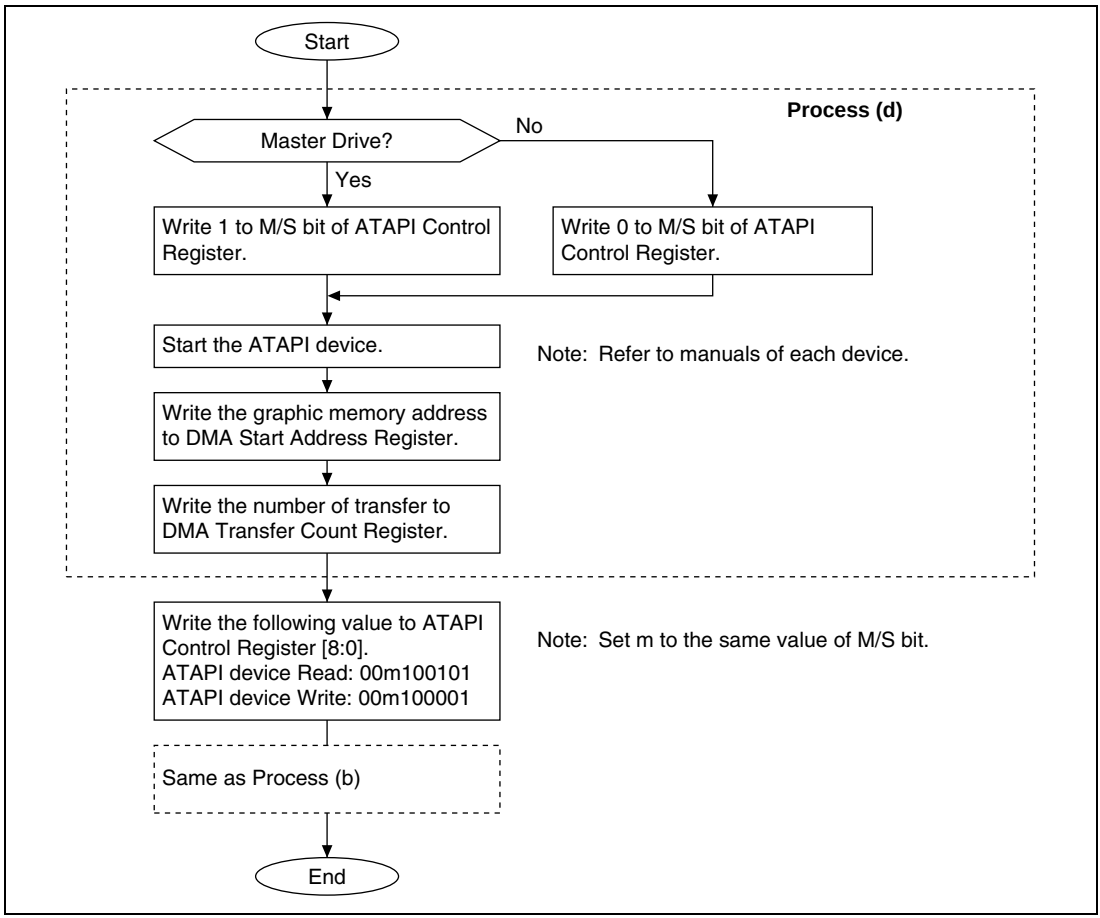


Figure 18.11 Transfer from/to Graphic Memory through Pixel Bus by Polling

Transfer from/to Graphic Memory through Pixel Bus by Interrupt

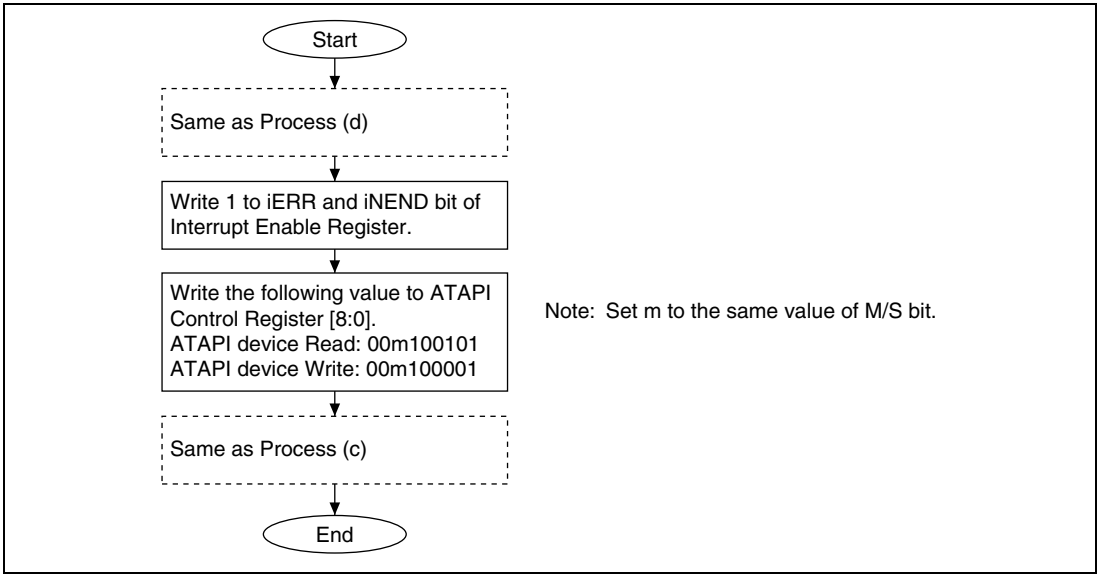


Figure 18.12 Transfer from/to Graphic Memory through Pixel Bus by Interrupt

18.8.4 Procedure in Ultra DMA Transfer Mode

Transfer from/to Peripherals on Register Bus by Polling

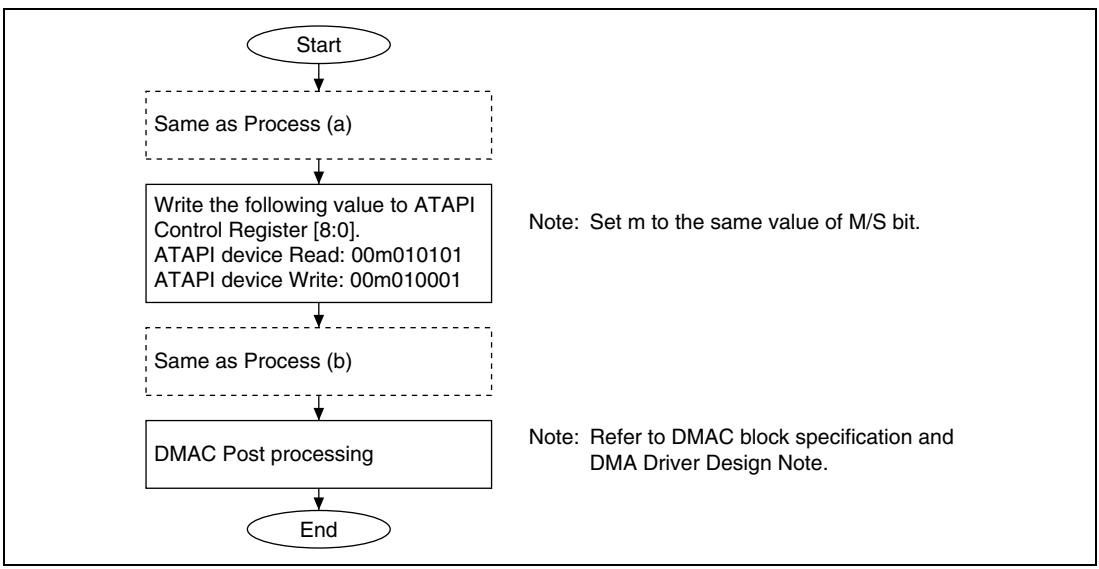


Figure 18.13 Transfer from/to Peripherals on Register Bus by Polling

Transfer from/to Peripherals on Register Bus by Interrupt

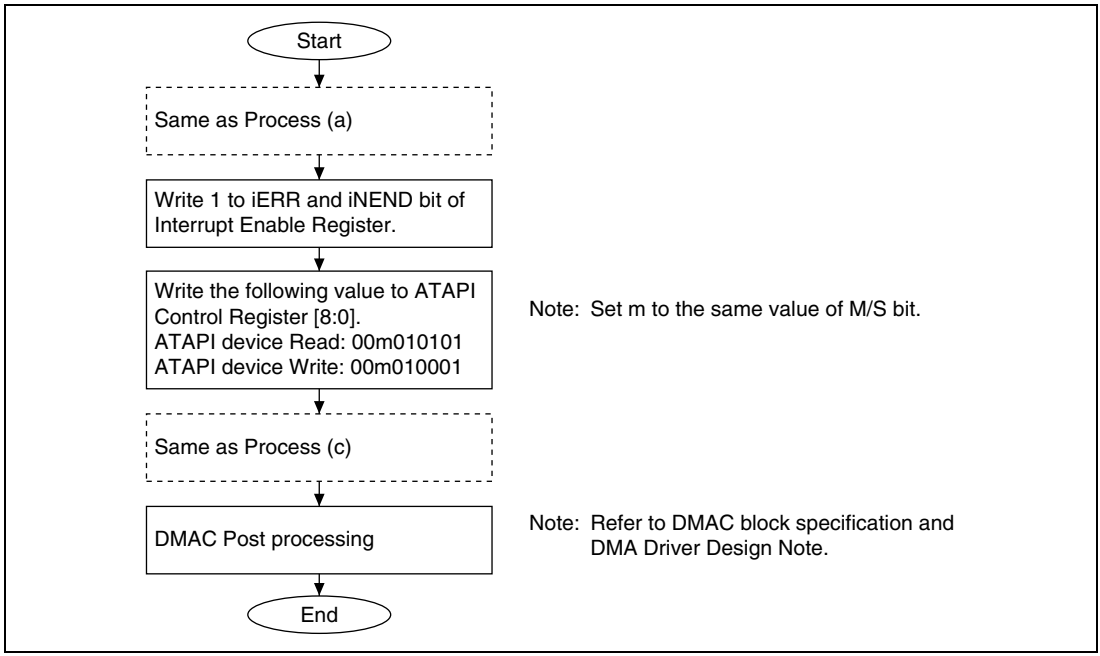


Figure 18.14 Transfer from/to Peripherals on Register Bus by Interrupt

Transfer from/to Graphic Memory through Pixel Bus by Polling

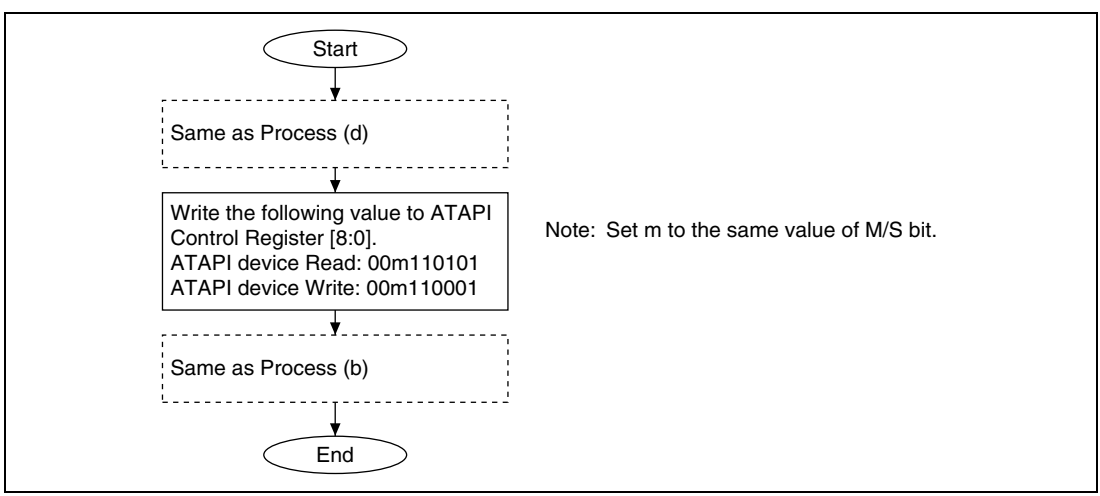


Figure 18.15 Transfer from/to Graphic Memory through Pixel Bus by Polling

Transfer from/to Graphic Memory through Pixel Bus by Interrupt

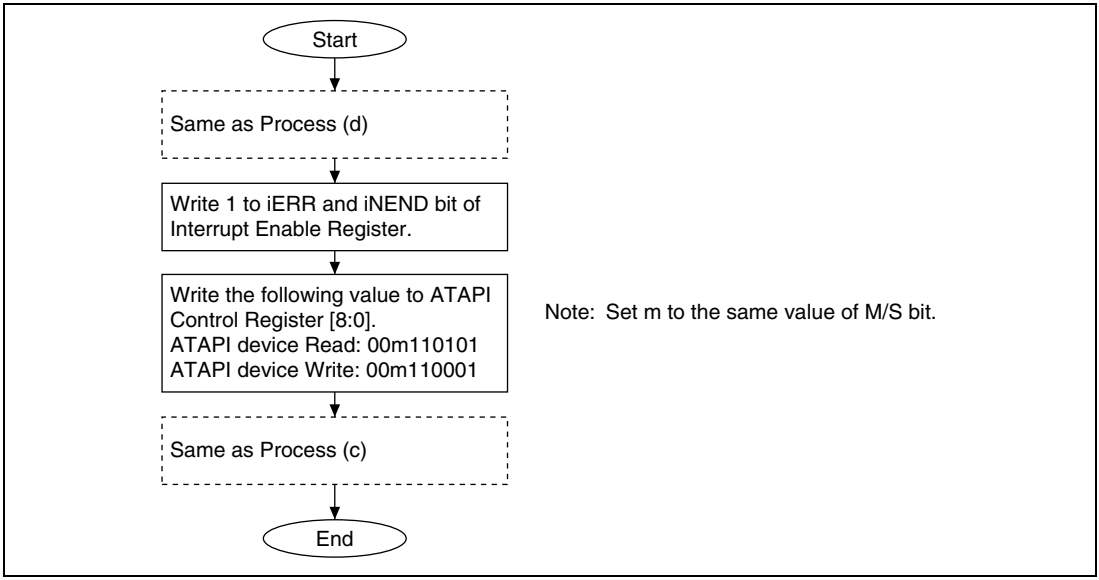


Figure 18.16 Transfer from/to Graphic Memory through Pixel Bus by Interrupt

18.8.5 Procedure in Hardware Reset for ATAPI Device

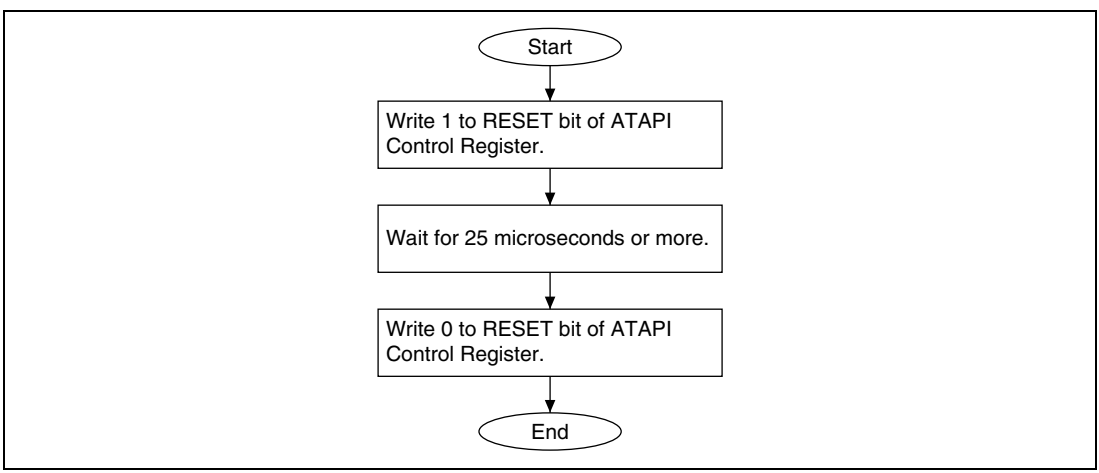


Figure 18.17 Procedure in Hardware Reset for ATAPI Device

HITACHI

Section 19 HCAN-2 Module

19.1 Summary

19.1.1 Overview

This document primarily describes the programming interface for the HCAN-2 (Hitachi CAN Version 2) module, that is significantly improved from the previous HCAN-1 (Hitachi CAN Version 1). It serves to facilitate the hardware/software interface so that engineers involved in the HCAN implementation can ensure the design is successful.

19.1.2 Scope

The CAN Data Link Controller function is not described in this document. It is the responsibility of the reader to investigate the CAN Specification Document (see references). The interfaces from the CAN Controller are described, in so far as they pertain to the connection with the User Interface.

The programming model is described in some detail. It is not the intention of this document to describe the implementation of the programming interface, but to simply present the interface to the underlying CAN functionality.

The document places no constraints upon the implementation of the HCAN module in terms of process, packaging or power supply criteria. These issues are resolved where appropriate in implementation specifications.

19.1.3 Audience

In particular this document provides the design reference for software authors who are responsible for creating a CAN application using this module.

In the creation of the HCAN user interface LSI engineers must use this document to understand the hardware requirements.

19.1.4 References

1. CAN License Specification, Robert Bosch GmbH, 1992
2. CAN Specification Version 2.0, Robert Bosch GmbH, 1991
3. Implementation Guide for the CAN Protocol, CAN Specification 2.0 Addendum, CAN In Automation, Erlangen, Germany
4. OSEK Communication Specification, Version 2.1 revision 1, OSEK /VDX, 17th June 1998

19.1.5 Features

- supports CAN specification 2.0A/2.0B and ISO-11898:
- support ISO-WD-11898-4 for CAN Timer Triggered Communication on level 1 (TTCAN Level 1)
- 31 programmable Mailboxes for transmit / receive + 1 receive-only mailbox
- sleep mode for low power consumption and automatic recovery from sleep mode by detecting CAN bus activity
- programmable receive filter mask (standard and extended identifier) supported by all Mailboxes
- programmable CAN data rate up to 1MBit/s
- transmit message queuing with internal priority sorting mechanism against the problem of priority inversion for real-time applications
- data buffer access without handshake requirement
- flexible micro-controller interface
- flexible interrupt structure
- 16-bit free running timer with pre-scalar, 2 Timer Compare Match Registers, CAN-ID Compare Match, 2 Input Capture Registers, Drift Correction Register, Local Offset Register
- 4-bit Basic Cycle Counter for Time Triggered Transmission
- Timer Compare Match Registers with interrupt generation + Timer counter clear/set capability to support schedule-monitoring of transmit/receive, one-shot transmission at a specific time, etc
- CAN-ID Compare Match with Timer Clear/Set + Input Capture Register Disable when received a specific CAN Frame
- Input Capture Registers used for TimeStamp and Global Synchronization on a CAN system, interacting with SOF/EOF of CAN Frame and CAN-ID Compare Match
- Flexible TimeStamp for both transmission and reception (stamp-timing programmable) supported
- Time-Triggered Transmission, Periodic Transmission supported on top of Event Triggered Transmission
- Timer Counter and Basic Cycle value can be embedded into a CAN frame and transmitted

19.1.6 HCAN-2 Differences from HCAN-1

CAN Core Interface

HCAN-2 obtains the new CAN Core Version that achieves the same baud rate as the previous version with half a clock speed and features some useful test modes. The power consumption will consequently be halved. Also, the change eliminates the existing limitations with HCAN-1C

Features Added

The following features have been added in HCAN-2.

- Timer Function added supporting 2 Input Capture Registers and 3 Timer Compare Match interrupts/timer-clear-set and CAN-ID Compare Match timer-clear-set/ICR-freeze
- Timestamp support of all incoming messages and outgoing messages
- LAFM (Local Acceptance Filter Mask) support of all incoming messages
- Global Synchronization using one of the 2 input capture register
- Time Triggered Transmission and Periodic Transmission supported on top of Event Triggered Transmission
- Extended/Enhanced Address Map
- IRR0 function to notify S/W reset and Halt
- Halt Mode status bit and Error Passive status bit added to GSR
- Various Test Modes supported
- IRR2 does not duplicate IRR1 and RXPR does not duplicate RFPR any more – Data Frame and Remote Frame are separated
- When transmitting, the highest priority search is scanned from Mailbox-31 down to Mailbox-1
- When receiving, the matching ID search is scanned from Mailbox-31 down to Mailbox-1, and 1 received message is only stored into 1 Mailbox
- More flexible BCR
- Bus Off Interrupt (IRR6) is generated at the end of Bus Off period, as well as at the beginning

19.2 Architecture

The HCAN-2 device offers a flexible and sophisticated way to organise and control CAN frames, supporting CAN2.0B Active and ISO-11898. The module is formed from 5 different functional entities. These are the Micro Processor Interface (MPI), Mailbox, Mailbox Control, Timer, and CAN Interface. The figure below shows the block diagram of the HCAN-2 Module. The bus interface timing is designed based on the SuperH peripheral bus interface (PP-Bus).

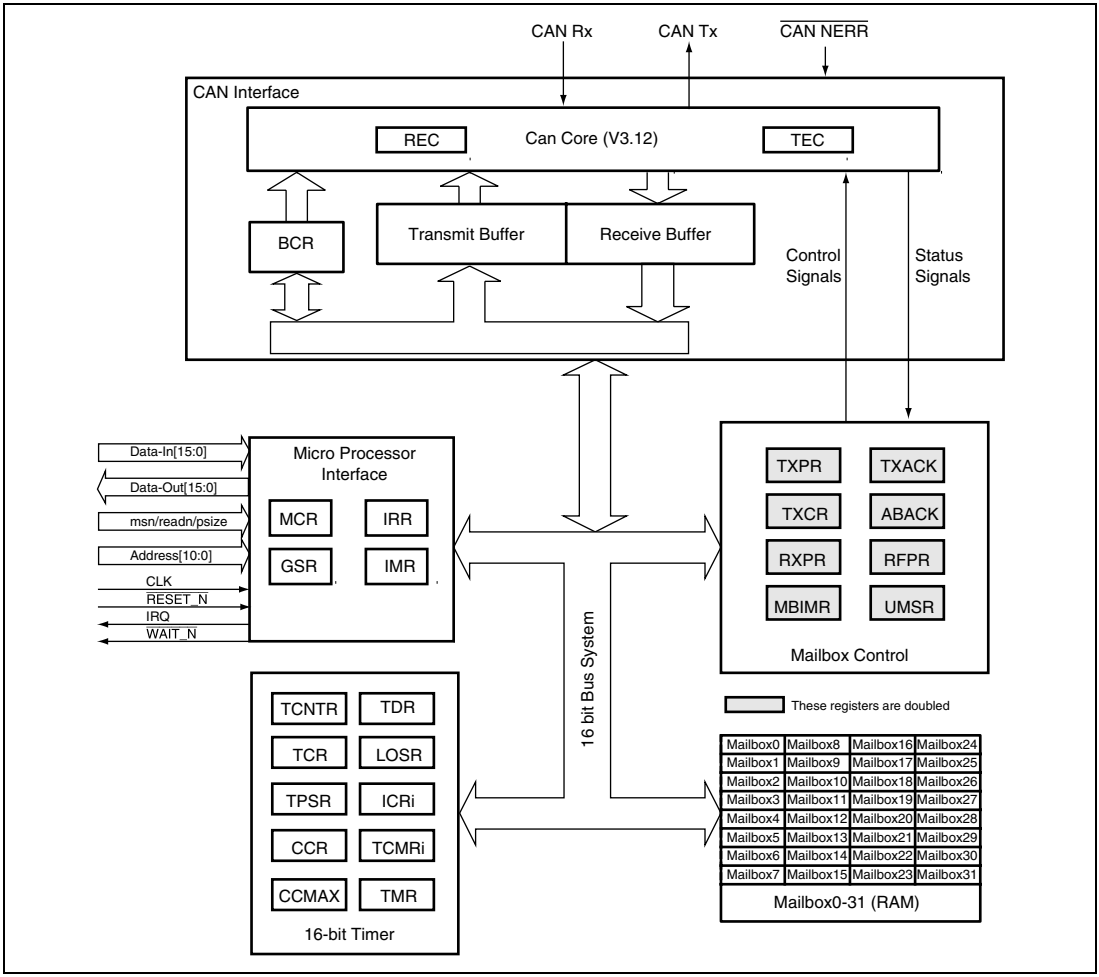


Figure 19.1 Block Diagram of HCAN-2 Module

Important: Since HCAN-2 is designed based on a 16-bit bus system, LongWord (32-bit) access is prohibited when the module is connected to a 32-bit bus. In this case, Word Access must be used for all the registers, and Word or Byte access must be used for the Mailboxes.

Micro Processor Interface (MPI)

The MPI allows communication between the Hitachi host processor and the HCAN's registers/mailboxes to control the timer unit, the memory interface and the data controller etc. It also contains the Wakeup Control logic that detects the CAN bus activities and notifies the MPI and the other parts of HCAN so that the HCAN can automatically exit the Sleep mode.

Contains registers such as MCR, IRR, GSR and IMR.

Mailbox

The Mailboxes are essentially RAM configured as message buffers. There are 32 Mailboxes, and each mailbox has the following information.

- CAN message control (identifier, dlc, rtr, ide, etc)
- CAN message data (for CAN Data frames)
- Time Stamp for message receive/transmit
- Local Acceptance Filter Mask to receive or Transmission Trigger Time to transmit
- 3-bit wide Mailbox Configuration, Disable Automatic Re-Transmission bit, Auto-Transmission for Remote Request bit, New Message Control bit, Time Trigger Enable bit, Periodic Transmission Enable bit, Timer Counter Transmit

Mailbox Control

The Mailbox Control handles the following functions.

For received messages, compare the IDs and generate appropriate RAM addresses/data to store messages from the CAN Interface into the Mailbox and set/clear appropriate registers accordingly.

To transmit messages, run the internal arbitration to pick the correct priority message regardless of whether it is event-triggered or time-triggered, and load the message from the Mailbox into the Tx-buffer of the CAN Interface and set/clear appropriate registers accordingly.

Arbitrates Mailbox accesses between the host CPU and the Mailbox Control.

Contains registers such as TXPR, TXCR, TXACK, ABACK, RXPR, RFPR, and MBIMR.

Timer

The Timer function is the functional entity which provides HCAN with support for transmitting and receiving messages at a specific time frame and recording the result.

The Timer is a 16-bit free running up counter which can be controlled by the host CPU. It provides 3 16-bit Compare Match Registers presenting several features. They can generate interrupt signals, clear and set to the Local Offset value the counter value or clear the transmission of the messages in the transmission queue. 2 16-bit Input Capture Registers are included to record time-stamp on CAN messages and synchronize the Timer value globally within a CAN system.

The clock period of this Timer offers a wide selection derived from the system clock.

Contains registers such as TCNTR, TCR, TSR, TDCR, LOSR, ICR0_tm, ICR0_cc, ICR1, TCMR0, TCMR1 and TCMR2, TMR, CCR and CMAX.

CAN Interface

This block supports the requirements for a CAN Bus Data Link Controller which is specified in Ref. [2]. It fulfils all the functions of a standard DLC as specified by the OSI 7 Layer Reference model. This functional entity also provides the registers and the logic which are specific to a given CAN bus, which includes the Receive Error Counter, Transmit Error Counter, the Bit Timing Configuration Registers and various useful Test Modes. This block also contains functional entities to hold the data received and the data to be transmitted for the CAN Data Link Controller.

Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 19.1 Digital Block Interface Signals and Pin List

Signal or Pin name	In/Out	Function
CAN0_RX	In	CAN bus receive signal of channel 0
CAN0_TX	Out	CAN bus transmit signal of channel 0
$\overline{\text{CAN0\_NERR}}$	In	CAN Bus Error of channel 0
CAN1_RX	In	CAN bus receive signal of channel 1
CAN1_TX	Out	CAN bus transmit signal of channel 1
$\overline{\text{CAN1\_NERR}}$	In	CAN Bus Error of channel 1

19.3 Programming Model - Overview

The purpose of this programming interface is to allow convenient, effective access to the CAN bus for efficient message transfer.

19.3.1 Memory Map

The diagram of the memory map is shown below.

Base address: channel 0 → H'8000 channel 1 → H'8800

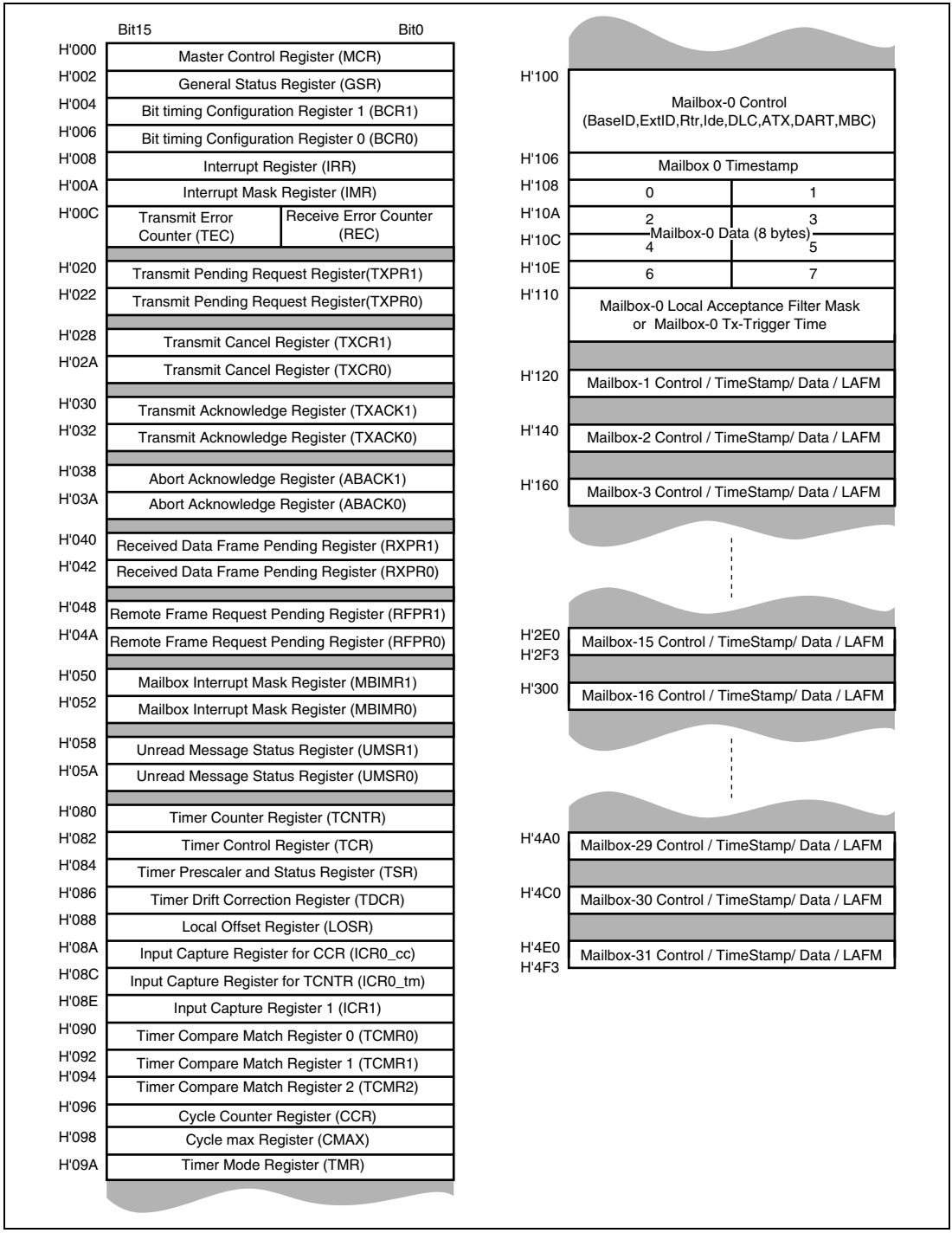


Figure 19.2 HCAN-2 Memory Map

19.3.2 Mailbox Structure

Mailboxes play a role as message buffers to transmit/receive CAN frames. Each Mailbox is comprised of 4 identical storage fields that are 1): Message Control, 2): Message Data, 3): Timestamp and 4): Local Acceptance Filter Mask/Transmission Trigger Time. The following table shows the address map for the control, data, timestamp and LAFM addresses for each mailbox.

Important: The Control/Timestamp/LAFM-Trigger Time fields can only be accessed in Word size (16-bit), whereas the message data area can be accessed in Word (16-bit) or Byte (8-bit) size. Also, unused parts of Mailboxes must be initialized during the configuration to their inactive state as they are in effect configured of RAMs. When the LAFM is not used to receive messages, it must be cleared.

Important: Unused Mailboxes can be used as extra memory. However, it is important in such case to disable the related mailbox (setting MBC to '111' (Bin).) in order to avoid that the mailbox joins the search for a matching identifier during the reception of messages, and even store a wrong message in the worst case.

Table 19.2 Mailbox Structure

	Address			
	Control	Time Stamp	Data	LAFM/Trigger Time
Mailbox	6 Bytes	2 Bytes	8 Bytes	4 Bytes
0 (Receive Only)	100 – 105	106 – 107	108 – 10F	110 – 113
1	120 – 125	126 – 127	128 – 12F	130 – 133
2	140 – 145	146 – 147	148 – 14F	150 – 153
3	160 – 165	166 – 167	168 – 16F	170 – 173
4	180 – 185	186 – 187	188 – 18F	190 – 193
5	1A0 – 1A5	1A6 – 1A7	1A8 – 1AF	1B0 – 1B3
6	1C0 – 1C5	1C6 – 1C7	1C8 – 1CF	1D0 – 1D3
7	1E0 – IE5	1E6 – 1E7	1E8 – 1EF	1F0 – 1F3
8	200 – 205	206 – 207	208 – 20F	210 – 213
9	220 – 225	226 – 227	228 – 22F	230 – 233
10	240 – 245	246 – 247	248 – 24F	250 – 253
11	260 – 265	266 – 267	268 – 26F	270 – 273
12	280 – 285	286 – 287	288 – 28F	290 – 293
13	2A0 – 2A5	2A6 – 2A7	2A8 – 2AF	2B0 – 2B3
14	2C0 – 2C5	2C6 – 2C7	2C8 – 2CF	2D0 – 2D3
15	2E0 – 2E5	2E6 – 2E7	2E8 – 2EF	2F0 – 2F3
16	300 – 305	306 – 307	308 – 30F	310 – 313
17	320 – 325	326 – 327	328 – 32F	330 – 333
18	340 – 345	346 – 347	348 – 34F	350 –353
19	360 – 365	366 – 367	368 – 36F	370 – 373
20	380 – 385	386 – 387	388 – 38F	390 – 393
21	3A0 – 3A5	3A6 – 3A7	3A8 – 3AF	3B0 – 3B3
22	3C0 – 3C5	3C6 – 3C7	3C8 – 3CF	3D0 – 3D3
23	3E0 – 3E5	3E6 – 3E7	3E8 – 3EF	3F0 – 3F3
24	400 – 405	406 – 407	408 – 40F	410 – 413
25	420 – 425	426 – 427	428 – 42F	430 – 433
26	440 – 445	446 – 447	448 – 44F	450 – 453
27	460 – 465	466 – 467	468 – 46F	470 – 473
28	480 – 485	486 – 487	488 – 48F	490 – 493
29	4A0 – 4A5	4A6 – 4A7	4A8 – 4AF	4B0 – 4B3
30	4C0 – 4C5	4C6 – 4C7	4C8 – 4CF	4D0 – 4D3
31	4E0 – 4E5	4E6 – 4E7	4E8 – 4EF	4F0 – 4F3

Mailbox-0 is a receive-only box, and all the rest of Mailbox-1 to 31 can operate as both receive and transmit boxes, dependant upon the MBC (Mailbox Configuration) bits in the Message Control. The following diagram shows the structure of a Mailbox in detail.

Address	Data Bus																Access Size	Field Name
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
H'100 + N*32	0	STDID[10:0]											RTR	IDE	EXTID [17:16]		Word (16-bit)	Control
H'102 + N*32	EXTID[15:0]																Word (16-bit)	
H'104 + N*32	CCM	TTE	NMC	ATX	DART	MBC[2:0]			0	TCT	CBE	CLE	DLC[3:0]				Byte (8-bit) or Word (16-bit)	
H'106 + N*32	TimeStamp[15:0]																Word (16-bit)	TimeStamp
H'108 + N*32	MSG_DATA_0 (first Rx/Tx Byte)								MSG_DATA_1								Byte (8-bit) or Word (16-bit)	Data
H'10A + N*32	MSG_DATA_2								MSG_DATA_3								Byte (8-bit) or Word (16-bit)	
H'10C + N*32	MSG_DATA_4								MSG_DATA_5								Byte (8-bit) or Word (16-bit)	
H'10E + N*32	MSG_DATA_6								MSG_DATA_7								Byte (8-bit) or Word (16-bit)	
H'110 + N*32	Local Acceptance filter Mask 0 (LAFM0) / Tx-Trigger Time 0 (TTT)																Word (16-bit)	LAFM/ Tx Trigger Control
H'112 + N*32	Local Acceptance filter Mask 1 (LAFM1) / Tx-Trigger Time 1 (TTT)																Word (16-bit)	

Figure 19.3 Mailbox-N Structure

Note: 1. All bits shadowed in gray are reserved and should be written LOW. The value returned by a read may not always be '0' and should not be relied upon.

2. ATX and DART are not supported by Mailbox-0, and the MBC setting of Mailbox-0 is limited.

3. If the CAN Bus is configured in Little Endian (MCR.4 = 1) the transfer is started from MSG_DATA_1 instead of MSG_DATA_0 (i.e. the sequence becomes: MSG_DATA_1, MSG_DATA_0, MSG_DATA_3, MSG_DATA_2, MSG_DATA_5, MSG_DATA_4, MSG_DATA_7, MSG_DATA_6).

Message Control Field

STD_ID[10:0]: These bits set the identifier (standard identifier) of data frames and remote frames.

EXT_ID[17:0]: These bits set the identifier (extended identifier) of data frames and remote frames.

RTR (Remote Transmission Request bit): Used to distinguish between data frames and remote frames. This bit is overwritten by received CAN Frames depending on Data Frames or Remote Frames.

Important: Please note that, when ATX bit is set with the setting MBC = 001(bin), the RTR bit will never be set. When a Remote Frame is received, the host processor can be notified by the corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt), however, as HCAN needs to transmit the current message as a Data Frame, the RTR bit remains '0'.

RTR	Description
0	Data frame
1	Remote frame

IDE (Identifier Extension bit): Used to distinguish between the standard format and extended format of CAN data frames and remote frames.

IDE	Description
0	Standard format
1	Extended format

DLC[3:0] (Data Length Code): These bits encode the number of data bytes from 0, 1, 2, … 8 that will be transmitted in a data frame.

DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
0	0	0	0	Data Length = 0 bytes
0	0	0	1	Data Length = 1 byte
0	0	1	0	Data Length = 2 bytes
0	0	1	1	Data Length = 3 bytes
0	1	0	0	Data Length = 4 bytes
0	1	0	1	Data Length = 5 bytes
0	1	1	0	Data Length = 6 bytes
0	1	1	1	Data Length = 7 bytes
1	X	X	X	Data Length = 8 bytes

MBC[2:0] (Mailbox Configuration): These bits configure the nature of each Mailbox as follows. When MBC = 111 (Bin), the Mailbox is inactive, i.e., it does not receive or transmit a message regardless of TXPR or other settings. When the MBC is set to '000' (Bin) and the TTE bit is set, the Tx-Trigger Time field becomes available. The MBC = '110' setting is prohibited. When the MBC is set to any other value, the LAFM field becomes available.

MBC[2]	MBC[1]	MBC[0]	Data Frame Transmit	Remote Frame Transmit	Data Frame Receive	Remote Frame Receive	Remarks
0	0	0	Yes	Yes	No	No	• Not allowed for Mailbox-0 • Time-Trigger can be used
0	0	1	Yes	Yes	No	Yes	• Can be used with ATX • Not allowed for Mailbox-0 • LAFM can be used
0	1	0	No	No	Yes	Yes	• Allowed for Mailbox-0 • LAFM can be used
0	1	1	No	No	Yes	No	• Allowed for Mailbox-0 • LAFM can be used
1	0	0	No	Yes	Yes	Yes	• Not allowed for Mailbox-0 • LAFM can be used
1	0	1	No	Yes	Yes	No	• Not allowed for Mailbox-0 • LAFM can be used
1	1	0	Setting prohibited				
1	1	1	Mailbox inactive				

TCT (Timer Counter Transfer): When this bit is set and a Mailbox is configured as a transmit box, depending from its DLC (set to 1 to perform TTCAN Level 1), the TCNTR value, at the SOF, is embedded in the second and third bytes of the message data, instead of MSG_DATA_2 and 3, and the Cycle_count in the first byte instead of MSG_DATA_0[3:0] when this Mailbox starts transmission. This function will be useful when HCAN performs a Time Master role to send the Time reference message. For example, considering that two HCAN controllers are connected in the same network and that the receiver stores the message in Mailbox N, the data format is as follow depending of the endian configuration set for the CAN Bus (MCR.4).

H'108 + N*32	Cycle_Counter (first Rx/Tx Byte)	MSG_DATA_1	Byte (8-bit) or Word (16-bit)	Data
H'10A + N*32	TCNTR[7:0]	TCNTR[15:8]	Byte (8-bit) or Word (16-bit)	
H'10C + N*32	MSG_DATA_4	MSG_DATA_5	Byte (8-bit) or Word (16-bit)	
H'10E + N*32	MSG_DATA_6	MSG_DATA_7	Byte (8-bit) or Word (16-bit)	
Big Endian				

Data	MSG_DATA_1	Cycle_Counter (first Rx/Tx Byte)	Byte (8-bit) or Word (16-bit)	H'108 + N*32
	TCNTR[15:8]	TCNTR[7:0]	Byte (8-bit) or Word (16-bit)	H'10A + N*32
	MSG_DATA_5	MSG_DATA_4	Byte (8-bit) or Word (16-bit)	H'10C + N*32
	MSG_DATA_7	MSG_DATA_6	Byte (8-bit) or Word (16-bit)	H'10E + N*32
Little Endian				

Figure 19.4 Message Data field

CBE: CAN Bus Error. An external fault-tolerant CAN transceiver can be used together with the HCAN module. In such case the error output pin of the transceiver (normally active low) must be connected to the $\overline{\text{CAN\_NERR}}$ pin of the Companion Chip of the interested HCAN channel. The value of the $\overline{\text{CAN\_NERR}}$ pin is stored into the bit CBE at the end of each Transmission/Reception (if the message is stored). This bit reports the inverted value of the $\overline{\text{CAN\_NERR}}$ pin. Then, using a transceiver with the error pin active low, CBE shows a potential physical error with the CAN Bus when set to '1'. If a transceiver with error pin active high is used the notation must be inverted. As the $\overline{\text{CAN\_NERR}}$ value is updated after the transmission or reception in the correspondent Mailbox non-interrupt is dedicated to this function but instead the interrupt for successful transmission (IRR.8) or reception (IRR.2/IRR.1) should be considered.

DART (Disable Automatic Re-Transmission): When this bit is set, it disables the automatic re-transmission of a message in the event of an error on the CAN bus or an arbitration lost on the CAN bus. In effect, when this function is used, the corresponding TXCR bit is automatically set at the start of transmission. When this bit is set to '0', HCAN tries to transmit the message as many times as required until it is successfully transmitted or it is cancelled by the TXCR.

ATX (Automatic Transmission of Data Frame): When this bit is set to '1' and a Remote Frame is received into the Mailbox, a Data Frame is transmitted from the same Mailbox using the current contents of the message data by setting the corresponding TXPR automatically. The scheduling of transmission is still governed by CAN identifier. In order to use this function, MBC[2:0] needs to be programmed to be '001' (Bin). When a transmission is performed by this function, the DLC (Data Length Code) to be used is the one that has been received.

Important: Please note that, when this function is used, the RTR bit will never be set despite receiving a Remote Frame. When a Remote Frame is received, the host processor will be notified by the corresponding RFPR set or IRR[2] (Remote Frame Request Interrupt) if it is enabled through IMR[2] and the related MBIMR, however, as HCAN needs to transmit the current message as a Data Frame, the RTR bit remains '0'. If a mailbox is configured to receive Data Frames and Remote Frames, then the RTR bit is overwritten by the received CAN Frames.

NMC (New Message Control): When this bit is set to '0', the Mailbox of which the RXPR bit is already set does not store the new message but maintains the old one and sets the UMSR correspondent bit. When this bit is set to '1', the Mailbox of which the RXPR bit is already set overwrites with the new message and sets the UMSR correspondent bit.
If a message is received in a mailbox configured in the overwrite mode (NMC = 1), the CPU must perform an additional check at the end of the data dumping from the mailbox in order to guarantee that the mailbox data have not been corrupted during such operation by another incoming message. The check, to be performed at the end of the mailbox access, consists in verifying that the associated bit of UMSR has not been set and so no overwrite has occurred; in case such bit is set data have been corrupted and so the message must be discarded.

TTE (Time Trigger Enable): When this bit is set to '1', the Mailbox of which the TXPR bit is set will transmit the message at the specified time in the Tx Trigger Time (TTT) field. Please refer to the Appendix 19.4.3 for further details.

Please note that if the time triggered mode is used, Mailbox No. 1 should be used only for reception.

CCM (CAN-ID Compare Match): When this bit is set, a reception of a message into the corresponding Mailbox can trigger two actions. If TCR9 is set to '1', the reception of the message will automatically clear TCR14 to freeze the ICR0 Register. If TCR10 is set to '1', the reception of the message will automatically clear the TCNTR (Timer Counter Register) and set it to LOSR (Local Offset Register) value.

CLE (Transmission Clear Enable): When this bit is set, a reception of a message into the corresponding Mailbox produces the cancellation of the pending messages in the transmission queue. This action is notified by IRR.8 and ABACK.

Important: It's important, for the proper operation of this feature, that the configuration of the Mailbox is not changed while receiving a message.

Timestamp Fields

Storage for the Timestamp recorded on messages for transmit/receive. The Timestamp will be a useful function to monitor if messages are received/transmitted within expected schedule or to schedule messages for transmission in the appropriate order.

Message Receive: For received messages, the ICR1 (Input Capture Register 1) always captures the TCNTR (Timer Counter Register) value or the concatenation of Cycle_Counter + TCNTR[15:4], depending on the programmed value in the bit 3 of TMR (Timer Mode Register) at either SOF or EOF depending on the programmed value in the TCR13 (Timer Control Register), and stores the ICR1 value into this Timestamp field of the corresponding Mailbox.

Message Transmit: For transmitted messages, the TCNTR (Timer Counter Register) value or the concatenation of Cycle_Counter + TCNTR[15:4], depending on the programmed value in the bit 3 of TMR (Timer Mode Register) is captured when either a TXPR bit or a TXACK bit is set depending on the programmed value in the TCR12, and the captured value is stored into this Timestamp field of the corresponding Mailbox.

Message Data Fields

Storage for the CAN message data that is transmitted or received. MSG_DATA[0] corresponds to the first data byte that is transmitted or received. The bit order on the bus is bit 7 through to bit 0.

Local Acceptance Filter Mask (LAFM)/Tx-Trigger Time (TTT)

This area can be used as Local Acceptance Filter Mask (LAFM) for receive boxes or as Tx-Trigger Time (TTT) for transmit boxes.

LAFM: When MBC is set to 001, 010, 011, 100, 101 (Bin), this field becomes a LAFM Field. The LAFM is comprised of two 16-bit read/write areas as follows. It allows a Mailbox to accept more than one identifier for receives.

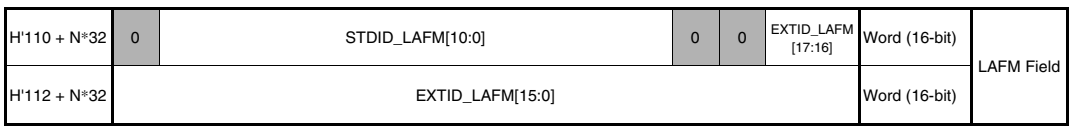


Figure 19.5 Acceptance filter

If a bit is set in the LAFM then the corresponding bit of a received CAN identifier is ignored when the HCAN searches a Mailbox with the matching CAN identifier. If the bit is clear then the corresponding bit of a received CAN identifier must match to the STD_ID/EXT_ID set in the mailbox to be stored. The structure of the LAFM is same as the message control in a Mailbox. If this function is not required, it must be filled with '0'.

Important: When LAFM is used, HCAN starts to find a matching identifier from Mailbox-31 down to Mailbox-0. As soon as HCAN finds one, it stops the search and stores the message into the Mailbox. This means that a received message can only be stored into one Mailbox.

Important: When a message is received and a matching Mailbox is found, the whole message is stored into the Mailbox. This means that, if the LAFM is used, the STD_ID, RTR, IDE, and EXT_ID may differ to the ones originally set as they are updated with the STD_ID, RTR, IDE, and EXT_ID of the received message.

STD_LAFM[10:0]—Filter mask bits for the CAN base identifier [10:0] bits.

STD_LAFM[10:0]	Description
0	Corresponding CAN base ID bit set in Mailbox0 is cared
1	Corresponding CAN base ID bit set in Mailbox0 is not cared

EXT_LAFM[17:0]—Filter mask bits for the CAN Extended identifier [17:0] bits.

EXT_LAFM[17:0]	Description
0	Corresponding CAN Extended ID bit is cared
1	Corresponding CAN Extended ID bit is not cared

TTT: When MBC is set to 000 (Bin), this field becomes a Tx-Trigger Time (TTT) Field. The TTT is comprised of two 16-bit read/write areas as follows.

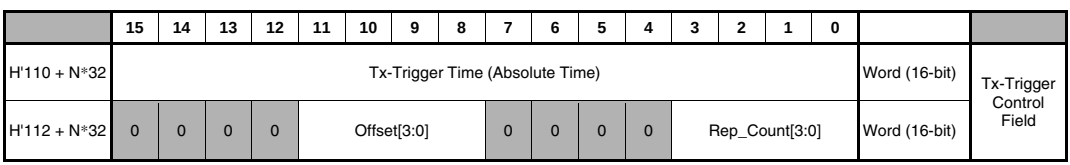


Figure 19.6 Tx-Trigger control field

The first 16-bit area specifies the time that triggers the transmission of the message in absolute time. The second 16-bit area specifies the basic cycle in the system matrix where the transmission must start (Offset) and the frequency for periodic transmission. When a TXPR is set, the corresponding Tx-Trigger Time (TTT), the Repeat Counter and the Offset are downloaded into an internal register. When the internal TTT register matches to the TCNTR value, and the internal Offset matches to CCR value the corresponding Mailbox automatically starts transmission. In order to enable this function, the TTE (Time Trigger Enable) bit must be enabled (set to '1') and the Timer (TCNTR) must be running (TCR15 = 1). When TTE is set to '0' and the corresponding TXPR bit is set, it joins the queue for transmission immediately. If Repeat Counter is different from zero the transmission occurs periodically every Rep_Count's basic cycle from CCR = Offset to CCR = MAX_CYCLE. In such case once TXPR is set by S/W, HCAN does not clear the corresponding TXPR bit to carry on performing the periodic transmission. In order to stop the

periodic transmission, TXPR must be cleared by TXCR or the Rep_Count field must be cleared. If Repeat Counter is zero the transmission occurs only once in correspondence of the programmed Basic Cycle (i.e. CCR = Offset and TCNTR = TTT).

The Tx-Trigger Time must not be set outside the TCNTR cycle if Compare Match Timer Clear-Set function is used (by TCMR0 or CCM). Please bear in mind that during a time triggered transmission only another one time triggered transmission can be triggered and that a minimum difference of 200 System Clock cycles between them is allowed. Please refer to the Appendix for further details.

19.3.3 HCAN Control Registers

The following sections describe HCAN control registers. The address is mapped as follow.

Important: These registers can only be accessed in Word size (16-bit).

Table 19.3 HCAN control registers

Channel	Address (Bytes)	Register Name	Abbreviation	Access Size (Bits)
0	H'8000	Master Control Register 0	MCR0	16
	H'8002	General Status Register 0	GSR0	16
	H'8004	Bit timing Configuration Register 1_0	BCR1_0	16
	H'8006	Bit timing Configuration Register 0_0	BCR0_0	16
	H'8008	Interrupt Request Register 0	IRR0	16
	H'800A	Interrupt Mask Register 0	IMR0	16
	H'800C	Transmit Error Counter Register 0/ Receive Error Counter Register 0	TEC0/REC0	16
1	H'8800	Master Control Register 1	MCR1	16
	H'8802	General Status Register 1	GSR1	16
	H'8804	Bit timing Configuration Register 1_1	BCR1_1	16
	H'8806	Bit timing Configuration Register 0_1	BCR0_1	16
	H'8808	Interrupt Request Register 1	IRR1	16
	H'880A	Interrupt Mask Register 1	IMR1	16
	H'880C	Transmit Error Counter Register 1/ Receive Error Counter Register 1	TEC1/REC1	16

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

Master Control Register n (MCR n) (n = 0, 1)

The Master Control Register (MCR) is a 16-bit read/write register that controls HCAN.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TST7	TST6	TST5	TST4	TST3	TST2	TST1	TST0	MCR7		MCR5	MCR4	MCR3	MCR2	MCR1	MCR0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TST7	0	R/W	**Test Mode (TST7)** This bit enables/disables the test modes settable by TST[6:0]. When this bit is set, the following TST[6:0] become effective. 0: HCAN is in normal mode 1: HCAN is in test mode
14	TST6	0	R/W	**Write CAN Error Counters (TST6)** This bit enables the TEC (Transmit Error Counter) and REC (Receive Error Counter) to be writable. The same value can only be written into the TEC/REC at the same time. The maximum value that can be written into the TEC/REC is D'255 (H'FF). This means that the HCAN cannot be forced into the bus off state. Before writing into the TEC/REC, HCAN needs to be put into Halt Mode, and when writing into the TEC/REC, the TST7 (MCR15) needs to be '1'. Only the same value can be set between TEC/REC, and the value written into TEC is used to write REC. 0: TEC/REC is not writable but read-only 1: TEC/REC is writable with the same value at the same time

Bit	Bit Name	Initial Value	R/W	Description
13	TST5	0	R/W	**Force to Error Passive (TST5)** This bit can force HCAN to become Error Passive. When this bit is set, HCAN behaves as an Error Passive node, regardless of the Error Counters. 0: State of HCAN depends on the Error Counters 1: HCAN behaves as an Error Passive node regardless of the Error Counters
12	TST4	0	R/W	**Auto Acknowledge Mode (TST4)** Allows HCAN to generate its own Acknowledge bit in order to enable Self-Test. In order to achieve the Self-Test mode, the message transmitted needs to be read back, and there are 2 settings for this. One is to set (Enable Internal Loop = 1 & Disable Tx Output = 1 & Disable Rx Input = 1), so that the Tx value can be internally provided to the Rx. The other way is to set (Enable Internal Loop = 0 & Disable Tx Output = 0 & Disable Rx Input = 0) and connect the Tx and Rx onto the CAN bus so that the transmitted data can be received via the CAN bus. 0: HCAN does not generate its own Acknowledge bit 1: HCAN generates its own Acknowledge bit
11	TST3	0	R/W	**Disable Error Counters (TST3)** Enable/disable the Error Counters (TEC/REC) to be functional. When this bit is disabled, the Error Counters (TEC/REC) remain unchanged and holds the current value. When this bit is enabled, the Error Counters (TEC/REC) function according to the CAN specification. 0: Error Counters (TEC/REC) function according to the CAN specification 1: Error Counters (TEC/REC) remain unchanged and holds the current value

Bit	Bit Name	Initial Value	R/W	Description
10	TST2	0	R/W	**Disable Rx Input (TST2)** Control the Rx to be supplied into the CAN Interface block. When this bit is enabled, the Rx pin value is supplied into the CAN Interface block. When this bit is disabled, the Rx value for the CAN block always remains recessive or the Tx value internally connected if Enable Internal Loop = 1. 0: External Rx pin is supplied for the CAN Interface block 1: Enable Internal Loop = 0: Rx always remain recessive for the CAN Interface block Enable Internal Loop = 1: Tx is internally supplied for the CAN Interface block
9	TST1	0	R/W	**Disable Tx Output (TST1)** Controls the Tx Output pin to output transmit data or recessive bits. If this bit is enabled, the internal transmit output value appears on the Tx pin. If this bit is disabled, the Tx Output pin always remains recessive. 0: External Tx pin is supplied for the CAN Interface block 1: Enable Internal Loop = 0: Tx is always recessive on the Tx pin Enable Internal Loop = 1: Tx is internally looped backed the internal Rx
8	TST0	0	R/W	**Enable Internal Loop (TST0)** Enable/disable the internal TX looped back to the internal Rx. For details, please refer to the Application Note. 0: Rx is fed from the Rx Pin 1: Rx is fed back from the internal Tx signal
7	MCR7	0	R/W	**Auto-wake Mode (MCR7)** MCR7 enables or disables the Auto-wake mode. If this bit set, the HCAN automatically cancels the sleep mode (MCR5) by detecting CAN bus activity (dominant bit). If MCR7 is cleared the HCAN does not automatically cancel the sleep mode. 0: Auto-wake by CAN bus activity disabled 1: Auto-wake by CAN bus activity enabled

Bit	Bit Name	Initial Value	R/W	Description
6	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
5	MCR5	0	R/W	**Sleep Mode (MCR5)** Enables or disables Sleep mode transition. If this bit is set the sleep mode is enabled. The HCAN waits for the completion of the current bus activity before shutting down. Until this mode is terminated the HCAN will ignore all CAN bus activities. The two Error Counters (REC, TEC) will remain the same during and after the Sleep mode. This mode will be exited in two ways: By writing a '0' to this bit position, or, if MCR[7] is enabled, after detecting a dominant bit on the CAN bus. When leaving this mode the HCAN will synchronize to the CAN bus (by checking for 11 recessive bits) before re-initialising. This means that, when the No. 2 method is used, HCAN will miss the first message to receive, however, CAN transceivers have the same feature, and the S/W needs to be designed in this manner. **Important:** In effect, this mode is same as setting the module to the Halt mode and stopping the clock. This means that, the interrupt is generated from IRR0 when entering the Sleep mode. During the Sleep mode, only the MPI block is accessible, i.e., MCR/GSR/IRR/IMR are accessible. However, for example, IRR1 cannot be cleared as it is an OR'ed signal of RXPR that cannot be cleared during the sleep mode, therefore, it is recommended to set the Halt mode first and then transit to the Sleep mode. 0: HCAN sleep mode released 1: Transition to HCAN sleep mode enabled

Bit	Bit Name	Initial Value	R/W	Description
4	MCR4	0	R/W	**CAN Endian Mode (MCR4):** This bit controls whether the HCAN should transmit the Messages in Little Endian Mode or Big Endian Mode. By using this bit, in other words, it is possible to perform an Endian conversion from the HCAN and the external network. Please note that this bit has effect only on the order of which the Data Field is Transmitted/Received. 0: Data Field Transmitted/Received in Big Endian mode 1: Data Field Transmitted/Received in Little Endian mode
3	MCR3	0	R/W	**Reserved.** This bit needs to be kept to its reset value
2	MCR2	0	R/W	**Message Transmission Priority (MCR2)** MCR2 selects the order of transmission for pending transmit data. If this bit is set pending transmit data are sent in order of the bit position in the Transmission Pending Register (TXPR). The order of transmission starts from Mailbox-31 as the highest priority, and then down to Mailbox-1 (if those mailboxes are configured for transmission). Please note that this feature cannot be used for time triggered transmission. If MCR2 is cleared, all messages for transmission are queued with respect to their priority (by running internal arbitration). The highest priority message has the Arbitration Field with the lowest digital value and is transmitted first. The internal arbitration includes the RTR bit and the IDE bit. 0: Transmission order determined by message identifier priority 1: Transmission order determined by mailbox number priority (Mailbox-31 → Mailbox-1)

Bit	Bit Name	Initial Value	R/W	Description
1	MCR1	0	R/W	**Halt Request (MCR1)** Setting the MCR1 bit causes the CAN controller to complete its current operation and then to be cut off the CAN bus. The HCAN remains in this Halt Mode until the MCR1 is cleared. During the Halt mode, the CAN Interface does not join the CAN bus activity or does not store messages nor transmit messages. All of the registers and Mailbox contents remain. The HCAN will complete the current operation if it is a transmitter or a receiver, and then enter the Halt Mode. If the CAN bus is in idle or intermission state, HCAN will enter the Halt Mode immediately. Entering the Halt Mode can be notified by IRR0 and GSR4. If a Halt request is made during bus off, HCAN remains bus off even after 128 × 11 recessive bits. In order to exit this state, the Halt condition needs to be recovered by SW. In the Halt mode, the HCAN configuration can be modified as it does not join the bus activity. This bit has to be cleared by writing a '0' to re-join the CAN bus. After this bit is cleared, the CAN Interface waits until it detects 11 recessive bits, and then joins the CAN bus. 0: Normal operating mode 1: Halt mode transition request

Bit	Bit Name	Initial Value	R/W	Description
0	MCR0	1	R/W	**Reset Request (MCR0)** Controls resetting of the HCAN module. After detecting a reset request the HCAN controller enters its reset routine, re-initialising the internal logic, and then set GSR3 and IRR0 to notify the reset mode. During a re-initialisation, all the registers are cleared. This bit has to be cleared by writing a '0' to join the CAN bus. After this bit is cleared, the HCAN module needs to be re-configured, waits until it detects 11 recessive bits, ant then joins the CAN bus. After Power On Reset, this bit and GSR3 are always set. This means that a reset request has been made and HCAN is in configuration mode. 0: CAN Interface normal operating mode (MCR0 = 0 and GSR3 = 0) Setting condition: When 0 is written after an HCAN reset 1: CAN Interface reset mode transition request

General Status Register n (GSR n) (n = 0, 1)

The General Status Register (GSR) is a 16-bit read-only register that indicates the status of HCAN.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											GSR5	GSR4	GSR3	GSR2	GSR1	GSR0
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 6	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
5	GSR5	0	R	**Error Passive Status Bit (GSR5)** Indicates whether the CAN Interface is Error Passive or not. This bit will be set high as soon as the HCAN enters the Error Passive state and is cleared when the module enters again the Error Active state (This means the GSR.5 will stay high during Error Passive and during Bus Off). Consequently to find out the correct state both GSR.5 and GRS.0 must be considered. 0: HCAN is not Error Passive Setting condition: HCAN is in Error Active state 1: HCAN is Error Passive (if GSR.0 = 0) Setting condition: When TEC ≥ 128 or REC ≥ 128
4	GSR4	0	R	**Halt/Sleep Status Bit (GSR4)** Indicates whether the CAN Interface is in the halt/sleep state or not. 0: HCAN is not in the Halt state or Sleep state 1: Halt mode (if MCR1 = 1) or Sleep mode (if MCR5 = 1) Setting condition: If MCR1 is set and the CAN bus is either in intermission or idle

Bit	Bit Name	Initial Value	R/W	Description
3	GSR3	1	R	**Reset Status Bit (GSR3)** Indicates whether the CAN Interface is in the reset state (Configuration mode) or not. 0: Normal operating state Setting condition: After an HCAN internal reset 1: Reset state (Configuration mode)
2	GSR2	1	R	**Message Transmission Complete Flag (GSR2)** Flag that indicates to the host processor if the HCAN is processing transmission requests or if a transmission is completed. In effect, this bit is an OR'ed signal of all the TXPR bits. Please note the difference to the meaning of IRR8 (Slot Empty) that is an OR'ed signal of all the TXACK/ABACK bits. 0: Transmission in progress 1: There is no message requested for transmission
1	GSR1	0	R	**Transmit/Receive Warning Flag (GSR1)** Flag that indicates an error warning. 0: Reset condition: When TEC < 96 or REC < 96 or TEC ≥ 256 1: When 96 ≤ TEC < 256 or 96 ≤ REC < 256
0	GSR0	0	R	**Bus Off Flag (GSR0)** Flag that indicates that HCAN is in the bus off state. 0: Reset condition: Recovery from bus off state 1: When TEC ≥ 256 (bus off state)

Bit Timing Configuration Register n (BCR0 n, BCR1 n) (n = 0, 1)

The Bit Timing Configuration Registers (BCR0 and BCR1) are 2 × 16-bit read/write register that is used to set CAN bit timing parameters and the baud rate pre-scaler for the CAN Interface.

For the following description the following definition is used:

$$\text{Timequanta} = \frac{BRP}{f_{clk}}$$

Where: BRP (Baud Rate Predivider) is stored in BCR0 and fclk is the used external frequency.

- BCR1

  Please refer to the table 19.4 for TSG1 and TSG2 setting.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSG1[3:0]					TSG2[2:0]					SJW[1:0]				EG	BSP
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TSG1[3]	0	R/W	**Time Segment 1 (TSG1[3:0] = BCR1[15:12])**
14	TSG1[2]	0	R/W	These bits are used to set the segment for absorbing output buffer, CAN bus, and input buffer delay. A value from 4 to 16 can be set.
13	TSG1[1]	0	R/W	
12	TSG1[0]	0	R/W	0000: Setting prohibited 0001: Setting prohibited 0010: Setting prohibited 0011: PRSEG + PHSEG1 = 4 time quanta 0100: PRSEG + PHSEG1 = 5 time quanta : 1111: PRSEG + PHSEG1 = 16 time quanta
11	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
10	TSG2[2]	0	R/W	**Time Segment 2 (TSG2[2:0] = BCR1[10:8])**
9	TSG2[1]	0	R/W	These bits are used to set the segment for correcting 1-bit time error. A value from 2 to 8 time quanta can be set as shown below
8	TSG2[0]	0	R/W	000: Setting prohibited 001: PHSEG2 = 2 time quanta (conditionally prohibited) See table 19.4 010: PHSEG2 = 3 time quanta 011: PHSEG2 = 4 time quanta 100: PHSEG2 = 5 time quanta 101: PHSEG2 = 6 time quanta 110: PHSEG2 = 7 time quanta 111: PHSEG2 = 8 time quanta
7, 6	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.

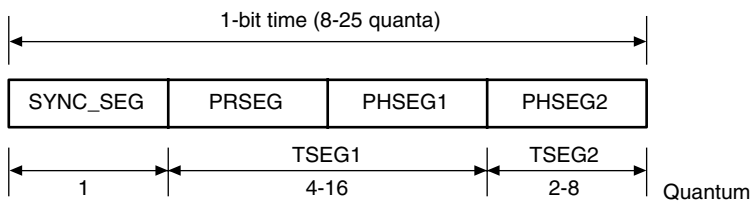
Bit	Bit Name	Initial Value	R/W	Description
5	SJW[1]	0	R/W	**ReSynchronization Jump Width (SJW[1:0] = BCR0[5:4])**
4	SJW[0]	0	R/W	These bits set the synchronization jump width. 00: Synchronization Jump width = 1 time quantum 01: Synchronization Jump width = 2 time quanta 10: Synchronization Jump width = 3 time quanta 11: Synchronization Jump width = 4 time quanta
3, 2	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
1	EG	0	R/W	**Edge Select (EG = BCR1[1])** Selects at which edge is to be used for re-synchronization. In order to comply to the standard CAN, '0' should be set. 0: Re-synchronization is performed at falling edge of Rx 1: Re-synchronization is performed at both rising + falling edge of Rx
0	BSP	0	R/W	**Bit Sample Point (BSP = BCR1[0])** Sets the point at which data is sampled. Three-time sampling is only available when the BRP is programmed to be less than 4. 0: Bit sampling at one point (end of time segment 1) 1: Bit sampling at three points (end of time segment 1, and 1 time quantum before and after)

- BCR0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BRP[7]	BRP[6]	BRP[5]	BRP[4]	BRP[3]	BRP[2]	BRP[1]	BRP[0]
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
7	BRP[7]	0	R/W	**Baud Rate Pre-scale (BRP[7:0] = BCR0 [7:0])**
6	BRP[6]	0	R/W	These bits are used to set the clock used for the Time Quantum.
5	BRP[5]	0	R/W	00000000: 1 × system clock
4	BRP[4]	0	R/W	00000001: 2 × system clock
3	BRP[3]	0	R/W	00000010: 3 × system clock
2	BRP[2]	0	R/W	: (BRP+1) X system clock
1	BRP[1]	0	R/W	11111111: 256 × system clock
0	BRP[0]	0	R/W	

Requirements of Bit Configuration Register



SYNC_SEG: Segment for establishing synchronization of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)

PRSEG: Segment for compensating for physical delay between networks.

PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronization (resynchronization) is established.)

PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronization (resynchronization) is established.)

The HCAN Bit Rate Calculation is:

$$\text{Bit rate} = \frac{f_{clk}}{BRP \times (\text{TSG1} + \text{TSG2} + 1)}$$

Where BRP, TSG1 and TSG2 are derived values from the descriptions of the tables above, but not the actual programmed values. The '+ 1' is for the Sync-Seg and fixed to 1 time quanta.

f_{CLK} = Pϕ (Peripheral Clock (ϕ/2 or ϕ/3))

BCR Setting Constraints

TSG1 > TSG2 ≥ SJW (SJW = 1 to 4)
TSG + TSG2 + 1 = 8 to 25 time quanta

These constraints allow the setting range shown in the table below for TSG1 and TSG2 in the Bit Timing Configuration Register.

Table 19.4 TSG1 and TSG2 setting.

			TSG2 (BCR[10:8])						
			001	010	011	100	101	110	111
			2	3	4	5	6	7	8
TSG1 (BCR [15:12])	0011	4	No	Yes	No	No	No	No	No
	0100	5	Yes	Yes	Yes	No	No	No	No
	0101	6	Yes	Yes	Yes	Yes	No	No	No
	0110	7	Yes	Yes	Yes	Yes	Yes	No	No
	0111	8	Yes	Yes	Yes	Yes	Yes	Yes	No
	1000	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1001	10	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1010	11	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1011	12	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1100	13	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1101	14	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1110	15	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	1111	16	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Examples:

1. To have a Bit rate of 1Mbps with a frequency of fclk = 40MHz it is possible to set: BRP = 4, TSG1 = 6, TSG2 = 3.

   Then the configuration to write is BCR1 = 5200 and BCR0 = 0003.
2. To have a Bit rate of 500Kbps with a frequency of 35MHz it is possible to set: BPR = 5, TSG1 = 8, TSG2 = 5. Then the configuration to write is BCR1 = 7400 and BCR0 = 0004.

Interrupt Request Register n (IRR n) (n = 0, 1)

The Interrupt Register (IRR) is a 16-bit read/write-clearable register containing status flags for the various interrupt sources.

- IRR (Address = H'008)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRR1	IRR1	IRR1	IRR1	IRR1	IRR1	IRR9	IRR8	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IRR15	0	R/W	**Timer Compare Match Interrupt 1 (IRR15)** Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to the Timer value (TCMR1 = TCNTR), this bit is set. Please note that this bit is not set if the TCMR1 value is H'0000. 0: Timer Compare Match has not occurred to the TCMR1 Clearing condition: Writing 1 1: Timer Compare Match has occurred to the TCMR1 Setting condition: TCMR1 matches to the Timer value (TCMR1 = TCNTR) if TMR.1 = 0 or to Cycle_Count + TCNTR[15:4] if TMR.1 = 1.
14	IRR14	0	R/W	**Timer Compare Match Interrupt 0 (IRR14)** Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR) or to Cycle_Count + TCNTR[15:4] depending of the configuration set in TMR.1 (Timer Mode Register), this bit is set. Please note that this bit is not set if the TCMR0 value is H'0000. 0: Timer Compare Match has not occurred to the TCMR0 Clearing condition: Writing 1 1: Timer Compare Match has occurred to the TCMR0 Setting condition: TCMR0 matches to the Timer value (TCMR0 = TCNTR).

Bit	Bit Name	Initial Value	R/W	Description
13	IRR13	0	R/W	**Timer Overrun Interrupt (IRR13)** Indicates that the Timer has overrun and is reset to the LOSR (Local Offset Register) value. Please note that this bit is set even when the TCMR0 is enabled to clear-set the Timer value and its value is set to H'FFFF. 0: Timer has not overrun Clearing condition: Writing 1 1: Timer has overrun Setting condition: When the Timer (TCNTR) changes from H'FFFF to H'0000
12	IRR12	0	R/W	**Wake-up on Bus Activity (IRR12)** IRR12 indicates that a CAN bus activity is present. While the HCAN is in sleep mode and a recessive to dominant bit transition takes place on the CAN bus, this bit is set. The operation of this interrupt is configured in the Master Control Register (MCR7 - Auto-wake Mode). This interrupt is cleared by writing a '1' to this bit position. Writing a '0' has no effect. 0: Bus idle state Clearing condition: Writing 1 1: CAN bus activity detected in HCAN sleep mode Setting condition: Recessive → dominant bit transition detection while in sleep mode
11	IRR11	0	R/W	**Timer Compare Match Interrupt 2 (IRR11)** Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to the Timer value (TCMR2 = TCNTR) or to Cycle_Count + TCNTR[15:4] depending of the configuration of TMR.2 (Timer Mode Register), this bit is set. Please note that this bit is not set if the TCMR2 value is H'0000. 0: Timer Compare Match has not occurred to the TCMR2 Clearing condition: Writing 1 1: Timer Compare Match has occurred to the TCMR2 Setting condition: TCMR2 matches to the Timer value (TCMR2 = TCNTR) if TMR.2 = 0 or to Cycle_Count + TCNTR[15:4] if TMR.2 = 1

Bit	Bit Name	Initial Value	R/W	Description
10	IRR10	0	R/W	**Cycle Counter overflow Interrupt 2 (IRR10)** Indicates that the Cycle_Counter has reached the maximum value (CMAX). When the CCR counter matches to the CMAX value (CCR = CMAX), this bit is set and the CCR is cleared. Please note that setting CMAX = 0 produces the Cycle_Counter to be disabled and consequently no interrupt are generated. 0: Cycle Counter has not reached CMAX or CMAX = 0 Clearing condition: Writing 1 1: Cycle Counter has reached CMAX and CMAX ≠ 0 Setting condition: CCR matches to the CMAX value (CCR = CMAX)
9	IRR9	0	R	**Message Overrun/Overwrite Interrupt Flag (IRR9)** Status flag indicating that a message has been received but the existing message in the matching Mailbox has not been read due to the corresponding RXPR or RFPR set to '1'. The received message is either abandoned (overrun) or overwritten dependant upon the NMC (New Message Control) bit. This bit is cleared by writing a '1' to the correspondent bit position in UMRS (Unread Message Status Register). Writing a '0' has no effect. 0: No message overrun/overwrite Clearing condition: Clearing of all bit in UMSR 1: Receive message overrun and its storage has been rejected or message overwrite Setting condition: Message is received while the corresponding RXPR or RFPR = 1 and MBIMR = 0

Bit	Bit Name	Initial Value	R/W	Description
8	IRR8	0	R	**Mailbox Empty Interrupt Flag (IRR8)** This bit is set when at least one TXPR bit is cleared, indicating that one of the messages for transmission or transmission cancellation has been successfully made and this mailbox is now ready to accept a new message data for the next transmission. In effect, this bit is set by an OR'ed signal of the TXACK and ABACK bits, therefore, this bit is automatically cleared when all the TXACK and ABACK bits are cleared. Writing '0' takes no effect. Please note that this bit does not represent that all TXPR bits are reset, whereas GSR.2 does. 0: Messages set for transmission or transmission cancellation NOT progressed. Clearing condition: All the TXACK and ABACK bits are cleared. 1: Message has been transmitted or aborted, and new message can be stored Setting condition: When one of the TXPR bits is cleared by completion of transmission or completion of transmission abort, i.e., when a TXACK or ABACK bit is set (if MBIM = 0).
7	IRR7	0	R/W	**Overload Frame (IRR7)** Status flag indicating that the HCAN has transmitted an overload frame. It remains latched until reset by writing a '1' to this bit position, writing a '0' has no effect. 0: Clearing condition: Writing 1 1: Setting condition: Overload frame transmitted

Bit	Bit Name	Initial Value	R/W	Description
6	IRR6	0	R/W	**Bus Off Interrupt Flag (IRR6)** This bit is set when HCAN enters the Bus-off state or when HCAN leaves Bus-off and returns to Error-Active. The cause therefore is the existing condition TEC ≥ 256 at the node or the end of Bus-off 128 × 11bits. This bit remains latched even the HCAN node leaves the bus-off condition, and needs to be explicitly cleared by S/W. The S/W is expected to read the GSR.0 to judge whether HCAN has become bus-off or error active. It is cleared by writing a '1' to this bit position even if the node is still bus-off. Writing a '0' has no effect. 0: Clearing condition: Writing 1 1: Bus off state caused by transmit error or Error Active state returning from Bus-off Setting condition: When TEC ≥ 256 or End of Bus-off after 128 × 11bits
5	IRR5	0	R/W	**Error Passive Interrupt Flag (IRR5)** Status flag indicating the error passive state caused by the transmit or receive error counter. This bit is reset by writing a '1' to this bit position, writing a '0' has no effect. If this bit is cleared the node may still be error passive. 0: Clearing condition: Writing 1 1: Error passive state caused by transmit/receive error Setting condition: When TEC ≥ 128 or REC ≥ 128
4	IRR4	0	R/W	**Receive Overload Warning Interrupt Flag (IRR4)** This bit becomes set and latches if the Receive Error Counter (REC) reaches a value greater than 96. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect. When the interrupt is cleared the REC still holds its value greater than 96. 0: Clearing condition: Writing 1 1: Error warning state caused by receive error Setting condition: When REC ≥ 96

Bit	Bit Name	Initial Value	R/W	Description
3	IRR3	0	R/W	**Transmit Overload Warning Interrupt Flag (IRR3)** This bit becomes set and latches if the Transmit Error Counter (TEC) reaches a value greater than 96. The interrupt is reset by writing a '1' to this bit position, writing '0' has no effect. When the interrupt is reset the TEC still holds a value greater than 96. 0: Clearing condition: Writing 1 1: Error warning state caused by transmit error Setting condition: When TEC ≥ 96
2	IRR2	0	R	**Remote Frame Request Interrupt Flag (IRR2)** Status flag indicating that a remote frame has been received in a mailbox. This bit is set if at least one receive mailbox contains a remote frame transmission request. This bit is cleared by ensuring all bits in the Remote Request Pending Register (RFPR) are cleared. Writing to this bit has no effect. 0: Clearing condition: Clearing of all bits in RFPR 1: At least one remote request is pending Setting conditions: When remote frame is received and the corresponding MBIMR = 0
1	IRR1	0	R	**Data Frame Received Interrupt Flag (IRR1)** IRR1 indicates that there are pending Data Frames received. If this bit is set at least one receive mailbox contains a pending message. This bit is cleared when all bits in the Receive Message Pending Register (RXPR) are cleared, i.e. there is no pending message in any receiving mailbox. It is in effect a logical OR from each configured receive mailbox. Writing to this bit has no effect. 0: Clearing condition: Clearing of all bits in RXPR 1: Data frame received and stored in Mailbox Setting conditions: When data is received and the corresponding MBIMR = 0

Bit	Bit Name	Initial Value	R/W	Description
0	IRR0	1	R/W	**Reset/Halt/Sleep Interrupt Flag (IRR0)** Status flag indicating that the CAN Interface has been reset or halted and HCAN is now in Configuration mode or the HCAN is asleep. An interrupt signal will be generated through this bit to notify the change of the HCAN's state to the host processor if a MCR0 (S/W reset) or MCR1 (Halt) or MCR5 (Sleep) request is made. The GSR may be read after this bit is set to figure out which state HCAN is in. **Important:** When a Sleep mode request needs to be made, the Halt mode should be used beforehand. Please refer to the MCR5 description. 0: Clearing condition: Writing 1 1: Transition to S/W reset mode or Transition to halt mode or transition to sleep mode without halt mode Setting condition: When reset/halt processing is completed after S/W reset (MCR0) or Halt mode (MCR1) or Sleep mode (MCR5) is requested

Interrupt Mask Register n (IMR n) (n = 0, 1)

The Interrupt Mask Register is a 16-bit register that protects all corresponding interrupts in the Interrupt Request Register (IRR) from generating an output signal on the IRQ. An interrupt request is masked if the corresponding bit position is set to '1'. This register can be read or written at any time. The IMR directly controls the generation of IRQ, but does not prevent the setting of the corresponding bit in the IRR.

- IMR (Address = H'00A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMR15	IMR14	IMR13	IMR12	IMR11	IMR10	IMR9	IMR8	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0
Init value:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	IMR15	1	R/W	Maskable interrupts sources corresponding to IRR[15:0] respectively. When a bit is set, the interrupt signal is not generated, although setting the corresponding IRR bit is still performed. 0: Corresponding IRR is not masked (IRQ is generated for interrupt conditions) 1: Corresponding interrupt of IRR is masked
14	IMR14	1	R/W	
13	IMR13	1	R/W	
12	IMR12	1	R/W	
11	IMR11	1	R/W	
10	IMR10	1	R/W	
9	IMR9	1	R/W	
8	IMR8	1	R/W	
7	IMR7	1	R/W	
6	IMR6	1	R/W	
5	IMR5	1	R/W	
4	IMR4	1	R/W	
3	IMR3	1	R/W	
2	IMR2	1	R/W	
1	IMR1	1	R/W	
0	IMR0	1	R/W	

Transmit Error Counter n (TEC n) and Receive Error Counter n (REC n) (n = 0, 1)

The Transmit Error Counter (TEC) and Receive Error Counter (REC) is a 16-bit read/(write) register that functions as a counter indicating the number of transmit/receive message errors on the CAN Interface. The count value is stipulated in the CAN protocol specification Refs. [2] and [3]. In the normal mode this register is read only, and can only be modified by the CAN Interface. This register can be cleared by a Reset request (MCR0) or bus off.

In a test mode (i.e. MCR[15] = MCR[14] = 1) it is possible to write to this register. A same value can only be written to TEC/REC, and the value written into TEC is set to TEC and REC. When writing to this register, HCAN needs to be put into Halt Mode. This feature is only intended for test purposes.

- TEC/REC (Address = H'00C)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0	REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15	TEC7	0	R/W*	Transmission Error Counter. This register is incremented if an error is detected during transmission as specified on the CAN specification (see section 19.1.4 References).
14	TEC6	0	R/W*	
13	TEC5	0	R/W*	
12	TEC4	0	R/W*	
11	TEC3	0	R/W*	
10	TEC2	0	R/W*	
9	TEC1	0	R/W*	
8	TEC0	0	R/W*	
7	REC7	0	R/W*	Reception Error Counter. This register is incremented if an error is detected during reception as specified on the CAN specification (see section 19.1.4 References).
6	REC6	0	R/W*	
5	REC5	0	R/W*	
4	REC4	0	R/W*	
3	REC3	0	R/W*	
2	REC2	0	R/W*	
1	REC1	0	R/W*	
0	REC0	0	R/W*	

Note: * It is only possible to write the value in test mode when MCR15 = MCR14 = 1

19.3.4 HCAN Mailbox Registers

The following sections describe HCAN Mailbox registers that control/flag individual Mailboxes. The address is mapped as follows.

Important: These registers can only be accessed in Word size (16-bit).

Table 19.5 HCAN Mailbox Registers

Channel	Address (Bytes)	Register Name	Mnemonic or Symbol	R/W	Access Size (Bits)
0	H'8020	Transmit Pending Register 1_0	TXPR1_0	R/W	16
	H'8022	Transmit Pending Register 0_0	TXPR0_0	R/W	
	H'8024				
	H'8026				
	H'8028	Transmit Cancel Register 1_0	TXCR1_0	R/W	16
	H'802A	Transmit Cancel Register 0_0	TXCR0_0	R/W	
	H'802C				
	H'802E				
	H'8030	Transmit Acknowledge Register 1_0	TXACK1_0	R/W	16
	H'8032	Transmit Acknowledge Register 0_0	TXACK0_0	R/W	
	H'8034				
	H'8036				
	H'8038	Abort Acknowledge Register 1_0	ABACK1_0	R/W	16
	H'803A	Abort Acknowledge Register 0_0	ABACK0_0	R/W	
	H'803C				
	H'803E				
	H'8040	Received Data Frame Pending Register 1_0	RXPR1_0	R/W	16
	H'8042	Received Data Frame Pending Register 0_0	RXPR0_0	R/W	
	H'8044				
	H'8046				
	H'8048	Remote Frame Request Pending Register 1_0	RFPR1_0	R/W	16
	H'804A	Remote Frame Request Pending Register 0_0	RFPR0_0	R/W	
	H'804C				
	H'804E				
	H'8050	Mailbox Interrupt Mask Register 1_0	MBIMR1_0	R/W	16
	H'8052	Mailbox Interrupt Mask Register 0_0	MBIMR0_0	R/W	
	H'8054				
	H'8056				
	H'8058	Unread message Status Register 1_0	UMSR1_0	R/W	16
	H'805A	Unread message Status Register 0_0	UMSR0_0	R/W	
	H'805C				
	H'805E				

Channel	Address (Bytes)	Register Name	Mnemonic or Symbol	R/W	Access Size (Bits)
1	H'8820	Transmit Pending Register 1_1	TXPR1_1	R/W	16
	H'8822	Transmit Pending Register 0_1	TXPR0_1	R/W	
	H'8824				
	H'8826				
	H'8828	Transmit Cancel Register 1_1	TXCR1_1	R/W	16
	H'882A	Transmit Cancel Register 0_1	TXCR0_1	R/W	
	H'882C				
	H'882E				
	H'8830	Transmit Acknowledge Register 1_1	TXACK1_1	R/W	16
	H'8832	Transmit Acknowledge Register 0_1	TXACK0_1	R/W	
	H'8834				
	H'8836				
	H'8838	Abort Acknowledge Register 1_1	ABACK1_1	R/W	16
	H'883A	Abort Acknowledge Register 0_1	ABACK0_1	R/W	
	H'883C				
	H'883E				
	H'8840	Data Frame Receive Pending Register 1_1	RXPR1_1	R/W	16
	H'8842	Data Frame Receive Pending Register 0_1	RXPR0_1	R/W	
	H'8844				
	H'8846				
	H'8848	Remote Frame Receive Pending Register 1_1	RFPR1_1	R/W	16
	H'884A	Remote Frame Receive Pending Register 0_1	RFPR0_1	R/W	
	H'884C				
	H'884E				
	H'8850	Mailbox Interrupt Mask Register 1_1	MBIMR1_1	R/W	16
	H'8852	Mailbox Interrupt Mask Register 0_1	MBIMR0_1	R/W	
	H'8854				
	H'8856				
	H'8858	Unread message Status Register 1_1	UMSR1_1	R/W	16
	H'885A	Unread message Status Register 0_1	UMSR0_1	R/W	
	H'885C				
	H'885E				

Transmit Pending Register n (TXPR1 n, TXPR0 n) (n = 0, 1)

The TXPR1 and TXPR0 are 16-bit read/conditionally-write registers that contain any transmit pending flags for the CAN module. The TXPR1 controls Mailbox-31 to Mailbox-16, and the TXPR0 controls Mailbox-15 to Mailbox-1. The host CPU may set the TXPR bits to affect any message being considered for transmission by writing a '1' to the corresponding bit location. Writing a '0' has no effect, and TXPR cannot be cleared by writing a '0' and must be cleared by setting the corresponding TXCR bits. TXPR may be read by the host CPU to determine which, if any, transmissions are pending. In effect there is a transmit pending bit for all Mailboxes except for the Mailbox-0. Writing a '1' to a bit location when the mailbox is configured to receive will have no effect, and will be automatically cleared when an internal arbitration for transmission runs.

The HCAN will clear a transmit pending flag after successful transmission of its corresponding message or when a transmission abort is requested successfully from the TXCR. The TXPR flag is not cleared if the message is not transmitted due to the CAN node losing the arbitration process or due to errors on the CAN bus, and HCAN automatically tries to transmit it again unless its DART bit (Disable Automatic Re-Transmission) is set in the Message-Control of the corresponding Mailbox. In such case (DART set) the transmission is cleared and notified through Mailbox Empty Interrupt Flag (IRR.8) and the correspondent bit within the Abort Acknowledgement Register (ABACK).

If the status of the TXPR changes, the HCAN shall ensure that in the identifier priority scheme (MCR[2] = 0), the highest priority message is always presented for transmission in an intelligent way even under circumstances such as bus arbitration losses or errors on the CAN bus. Please refer to the Application Note for details.

When the HCAN changes the state of any TXPR bit position to a '0', an empty slot interrupt (IRR[8]) may be generated. This indicates that either a successful or an aborted mailbox transmission has just been made. If a message transmission is successful it is signaled in the TXACK Register, and if a message transmission abortion is successful it is signaled in the ABACK Register. By checking these registers, the contents of the Message-Data of the corresponding Mailbox may be modified to prepare for the next transmission.

- TXPR1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TXPR1[15:0]	0	R/W*	Requests the corresponding Mailbox to transmit a CAN Frame. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number. 0: Transmit message idle state in corresponding mailbox Clearing condition: Completion of message transmission or message transmission abortion (automatically cleared) 1: Transmission request made for corresponding mailbox

Note: * Only when writing a '1' to a Mailbox configured as transmit.

- TXPR0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXPR0[15:1]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXPR0[15:1]	0	R/W*	**I**ndicates that the corresponding Mailbox is requested to transmit a CAN Frame. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. When multiple bits are set, the order of the transmissions is governed by the MCR2 – CAN-ID or Mailbox number. 0: Transmit message idle state in corresponding mailbox Clearing condition: Completion of message transmission or message transmission abortion (automatically cleared) 1: Transmission request made for corresponding mailbox
0	—	0	R	**Reserved** This bit is always '0' as this is a receive-only Mailbox. Writing a '1' to this bit position has no effect. The returned value is not guaranteed.

Note: * Only when writing a '1' to a Mailbox configured as transmit.

Transmit Cancel Register n (TXCR1 n, TXCR0 n) (n = 0, 1)

The TXCR1 and TXCR0 are 16-bit read/conditionally-write registers. The TXCR1 controls Mailbox-31 to Mailbox-16, and the TXCR0 controls Mailbox-15 to Mailbox-1.This register is used by the microprocessor to request the pending transmission requests in the TXPR to be cancelled. To clear the corresponding bit in the TXPR the host processor must write a '1' to the bit position in the TXCR. Writing a '0' has no effect.

When an abort has succeeded the CAN controller clears the corresponding TXPR + TXCR bits, and sets the corresponding ABACK bit. However, once a Mailbox has started a transmission, it cannot be cancelled by this bit. In such a case, if the transmission finishes in success, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding TXACK bit, however, if the transmission fails due to a bus arbitration loss or an error on the bus, the CAN controller clears the corresponding TXPR + TXCR bit, and sets the corresponding ABACK bit. If an attempt is made by the host processor to clear a mailbox transmission that is not transmit-pending it shall have no effect, and will be automatically cleared when an internal arbitration for transmission runs.

- TXCR1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TXCR1[15:0]	0	R/W*	Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 0 corresponds to Mailbox-31 to 16 (and TXPR1[15:0]) respectively. 0: Transmit message cancellation idle state in corresponding mailbox Clearing condition: Completion of transmit message cancellation (automatically cleared) 1: Transmission cancellation request made for corresponding mailbox

Note: * Only when writing a '1' to a Mailbox configured as transmit.

- TXCR0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXCR0[15:1]															0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXCR0[15:1]	0	R/W*	Requests the corresponding Mailbox, that is in the queue for transmission, to cancel its transmission. The bit 15 to 1 corresponds to Mailbox-15 to 1 (and TXPR0[15:1]) respectively. 0: Transmit message cancellation idle state in corresponding mailbox Clearing condition: Completion of transmit message cancellation (automatically cleared) 1: Transmission cancellation request made for corresponding mailbox
0	0	0	R	This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

Note: * Only writing a '1' to a Mailbox that is requested for transmission and is configured as transmit.
Only when writing a '1' to a Mailbox configured as transmit.

Transmit Acknowledge Register n (TXACK1 n, TXACK0 n) (n = 0, 1)

The TXACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the microprocessor that a mailbox transmission has been successfully made. When a transmission has succeeded the HCAN sets the corresponding bit in the TXACK Register. The host processor may clear a TXACK bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect.

- TXACK1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TXACK1[15:0]	0	R/WC1	Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) Setting condition: Completion of message transmission for corresponding mailbox

- TXACK0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXACK0[15:1]															0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	TXACK0[15:1]	0	R/WC1	Notifies that the requested transmission of the corresponding Mailbox has been finished successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox has successfully transmitted message (Data or Remote Frame) Setting condition: Completion of message transmission for corresponding mailbox
0	TXACK0[0]	0	R	This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

Abort Acknowledge Register n (ABACK1n, ABACK0n) (n = 0, 1)

The ABACK1 and TXACK0 are 16-bit read/conditionally-write registers. These registers are used to signal to the microprocessor that a mailbox transmission has been aborted as per its request. When an abort has succeeded the HCAN sets the corresponding bit in the ABACK register. The host processor may clear the Abort Acknowledge bit by writing a '1' to the corresponding bit location. Writing a '0' has no effect. An ABACK bit position is set by the HCAN to acknowledge that a TXPR bit has been cleared by the corresponding TXCR bit.

- ABACK1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ABACK1[15:0]	0	R/WC1	Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 0 corresponds to Mailbox-31 to 16 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) Setting condition: Completion of transmission cancellation for corresponding mailbox

- ABACK0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ABACK0[15:1]															0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 1	ABACK0[15:1]	0	R/WC1	Notifies that the requested transmission cancellation of the corresponding Mailbox has been performed successfully. The bit 15 to 1 corresponds to Mailbox-15 to 1 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox has cancelled transmission of message (Data or Remote Frame) Setting condition: Completion of transmission cancellation for corresponding mailbox
0	0	0	R	This bit is always '0' as this is a receive-only mailbox. Writing a '1' to this bit position has no effect and always read back as a '0'.

Received Data Frame Pending Register n (RXPR1 n, RXPR0 n) (n = 0, 1)

The RXPR1 and RXPR0 are 16-bit read/conditionally-write registers. The RXPR is a register that contains the received Data Frames pending flags associated with the configured Receive Mailboxes. When a CAN Data Frame is successfully stored in a receive mailbox the corresponding bit is set in the RXPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Data Frames. When a RXPR bit is set, it also sets IRR1 (Data Frame Received Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR1 is not set. Please note that these bits are only set by receiving Data Frames and not by receiving Remote frames. Please note that during overrun/overwrite conditions if a Data Frame is overwritten/overrun with a Remote Frame of vice versa both UMSR, RXPR and RFPR will be set for the same Mailbox. In this case the application needs to check the RTR bit within the Mailbox Control Field to understand the nature of the message on the Mailbox. Consequently when UMSR is set both RXPR and RFPR should be checked and, if necessary, cleared.

- RXPR1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RXPR1[15:0]	0	R/WC1	Configurable receives mailbox locations corresponding to each mailbox position from 31 to 16 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox received a CAN Data Frame Setting condition: Completion of Data Frame receive on corresponding mailbox

- RXPR0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXPR0[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RXPR0[15:0]	0	R/WC1	Configurable receives mailbox locations corresponding to each mailbox position from 15 to 0 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox received a CAN Data Frame Setting condition: Completion of Data Frame receive on corresponding mailbox

Remote Frame Request Pending Register n (RFPR1 n, RFPR0 n) (n = 0, 1)

The RFPR1 and RFPR0 are 16-bit read/conditionally-write registers. The RFPR is a register that contains the received Remote Frame pending flags associated with the configured Receive Mailboxes. When a CAN Remote Frame is successfully stored in a receive mailbox the corresponding bit is set in the RFPR. The bit may be cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. In effect there is a bit position for all mailboxes. However, the bit may only be set if the mailbox is configured by its MBC (Mailbox Configuration) to receive Remote Frames. When a RFPR bit is set, it also sets IRR2 (Remote Frame Request Interrupt Flag) if its MBIMR (Mailbox Interrupt Mask Register) is not set, and the interrupt signal is generated if IMR2 is not set. Please note that these bits are only set by receiving Remote Frames and not by receiving Data frames. Please note that during overrun/overwrite conditions if a Data Frame is overwritten/overrun with a Remote Frame of vice versa both UMSR, RXPR and RFPR will be set for the same Mailbox. In this case the application needs to check the RTR bit within the Mailbox Control Field to understand the nature of the message on the Mailbox. Consequently when UMSR is set both RXPR and RFPR should be checked and, if necessary, cleared.

- RFPR1n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RFPR1[15:0]	0	R/WC1	Remote Request pending flags for mailboxes 31 to 16 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox received Remote Frame Setting condition: Completion of remote frame receive in corresponding mailbox

- RFPR0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFPR0[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	RFPR0[15:0]	0	R/WC1	Remote Request pending flags for mailboxes 15 to 0 respectively. 0: Clearing condition: Writing '1' 1: Corresponding Mailbox received Remote Frame Setting condition: Completion of remote frame receive in corresponding mailbox

Mailbox Interrupt Mask Register n (MBIMR1 n, MBIMR0 n) (n = 0, 1)

The MBIMR1 and MBIMR0 are 16-bit read/write registers. The MBIMR only prevents the setting of IRR related to the Mailbox activities, that are IRR[1] – Data Frame Received Interrupt, IRR[2] – Remote Frame Request Interrupt, IRR[8] – Mailbox Empty Interrupt, and IRR[9] – Message OverRun Interrupt). If a mailbox is configured as receive, a mask at the corresponding bit position prevents the generation of a receive interrupt (IRR[1] and IRR[2] and IRR[9]) but does not prevent the setting of the corresponding bit in the RXPR or RFPR or MOR. Similarly when a mailbox has been configured for transmission, a mask prevents the generation of an Interrupt signal and setting of an Mailbox Empty Interrupt due to successful transmission or abortion of transmission (IRR[8]), however, it does not prevent the HCAN from clearing the corresponding TXPR/TXCR bit + setting the TXACK bit for successful transmission, or it does not prevent the HCAN from clearing the corresponding TXPR/TXCR bit + setting the ABACK bit for abortion of the transmission.

A mask is set by writing a '1' to the corresponding bit position for the mailbox activity to be masked. At reset all mailbox interrupts are masked.

- MBIMR 1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR1[15:0]															
Initial:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	MBIMR1[15:0]	1	R/W	Enable or disable interrupts requests from individual Mailbox-31 to Mailbox-16 respectively. 0: Interrupt Request from IRR1/IRR2/IRR8/ IRR9 enabled 1: Interrupt Request from IRR1/IRR2/IRR8/ IRR9 disabled

- MBIMR 0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBIMR0[15:0]															
Initial:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	MBIMR0[15:0]	1	R/W	Enable or disable interrupts requests from individual Mailbox-15 to Mailbox-0 respectively. 0: Interrupt Request from IRR1/IRR2/IRR8/ IRR9 enabled 1: Interrupt Request from IRR1/IRR2/IRR8/ IRR9 disabled

Unread Message Status Register n (UMSR1 n, UMSR0 n) (n = 0, 1)

UMSR1 and UMSR0 is a 16-bit read/write register and records the mailboxes whose contain has not been accessed by the CPU prior to a new message being received. If the host processor has not cleared the corresponding bit in the RXPR/RFPR when a new message for that mailbox is received, the corresponding UMSR bit is set to ′1′. This bit may be cleared by writing a ′1′ to the corresponding bit located in the UMSR. Writing a ′0′ has no effect.

If a mailbox is configured as transmit box, the corresponding UMSR will not be set.

- UMSR 1 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	UMSR1[15:0]	0	R/WC1	Indicate that an unread received message has been overwritten/overrun for Mailboxes 31 to 16. 0: Clearing condition: Writing '1' 1: Unread received message is overwritten by a new message or overrun condition Setting Condition: When a new message is received before RXPR/RFPR is cleared.

- UMSR 0 n (n = 0, 1)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UMSR0[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	UMSR0[15:0]	0	R/WC1	Indicate that an unread received message has been overwritten/overrun from Mailboxes 15 to 0. 0: Clearing condition: Writing '1' 1: Unread received message is overwritten by a new message Setting condition: When a new message is received before RXPR/RFPR is cleared

19.3.5 Timer Registers

The timer is a new feature within the HCAN-2 design. The Timer is 16 bits and supports several source clocks. These can all be divided by a pre-scale counter to reduce the speed of the clock. It also supports two Input Capture Registers (ICR1, ICR0) and two Compare Match Registers (CMR1, CMR0). The address map is as follows.

Important: These registers can only be accessed in Word size (16-bit).

Table 19.6 HCAN Timer registers

Channel	Address (Bytes)	Register Name	Abbreviation	Access Size (Bits)
0	H'8080	Timer Counter Register 0	TCNTR0	16
	H'8082	Timer Control Register 0	TCR0	16
	H'8084	Timer Status Register 0	TSR0	16
	H'8086	Timer Drift Correction Register 0	TDCR0	16
	H'8088	Local Offset Register 0	LOSR0	16
	H'808A	Input Capture Register for Basic Cycle 0	ICR0-cc0	16
	H'808C	Input Capture Register for Timer 0	ICR0-tm0	16
	H'808E	Input Capture Register 1_0	ICR1_0	16
	H'8090	Timer Compare Match Register 0_0	TCMR0_0	16
	H'8092	Timer Compare Match Register 1_0	TCMR1_0	16
	H'8094	Timer Compare Match Register 2_0	TCMR2_0	16
	H'8096	Cycle Counter Register 0	CCR0	16
	H'8098	Cycle Maximum Register 0	CMAX0	16
	H'809A	Timer Mode Register 0	TMR0	16
1	H'8880	Timer Counter Register 1	TCNTR1	16
	H'8882	Timer Control Register 1	TCR1	16
	H'8884	Timer Status Register 1	TSR1	16
	H'8886	Timer Drift Correction Register 1	TDCR1	16
	H'8888	Local Offset Register 1	LOSR1	16
	H'888A	Input Capture Register for Basic Cycle 1	ICR0-cc1	16
	H'888C	Input Capture Register for Timer 1	ICR0-tm1	16
	H'888E	Input Capture Register 1_1	ICR1_1	16
	H'8890	Timer Compare Match Register 0_1	TCMR0_1	16
	H'8892	Timer Compare Match Register 1_1	TCMR1_1	16
	H'8894	Timer Compare Match Register 2_1	TCMR2_1	16
	H'8896	Cycle Counter Register 1	CCR1	16
	H'8898	Cycle Maximum Register 1	CMAX1	16
	H'889A	Timer Mode Register 1	TMR1	16

Important: It is suggested to have the Timer disabled (TCR15 = 0) to set or change the configurations of the registers related to the Timer.

Timer Counter Register n (TCNTR n) (n = 0, 1)

This is a 16-bit read/write register that allows the CPU to monitor and modify the value of the Free Running Timer Counter. When the Timer rolls over or meets TCMR0 (Timer Compare Match Register 0) + TCR11 is set to '1', the TCNTR is set to LOSR (Local Offset Register) and starts running again.

Important: Please note that the timer and the cycle counter are managed as two independent registers and no double buffers are included to the read-write operation. Consequently if the timer is cleared (notified by Interrupt request) between the two read operations the read cycle counter and timer value could be related to a different time windows. In order to avoid this problem a double read operation is suggested (read CCR1 = CCR, TCNTR1 = TCNTR and again CCR2 = CCR. If CCR2 ≠ CCR1 read a second time TCNTR).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCNTR[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCNTR[15:0]	0	R/W*	Indicate the value of the Free Running Timer.

Note: * The register can be cleared by the Compare Match condition.

Timer Control Register n (TCR n) (n = 0, 1)

The Timer Control Register is a 16-bit read/write register and provides functions to control the operation of the Timer. This read/write register should be configured with the desired setting before to enable the timer through the bit TCR[15]. It is suggested to disable the timer before to modify the value of the pre-scaler.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCR15	TCR14	TCR13	TCR12	TCR11	TCR10	TCR9		TCR7		TPSC5	TPSC4	TPSC3	TPSC2	TPSC1	TPSC0
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15	TCR15	0	R/W	**Enable Timer** When this bit is set, the timer is running. When this bit is cleared the timer completes the current cycle (notified by Timer overrun or a compare match condition on TCMR0) and is cleared to zero. 0: Timer stop running and is cleared at the end of current cycle 1: Timer is running
14	TCR14	0	R/W	**Disable ICR0** Controls whether to enable or disable the Input Capture Register 0 (ICR0). When this bit is enabled, the Timer value is always captured everytime a Start Of Frame (SOF) appears on the CAN bus, whether HCAN is a transmitter or receiver. When this bit is disabled, the value of ICR0 remains latched. This function offers the Global Synchronization methodology. Please refer to the application note for details. 0: ICR0 is disabled and holds the current value Clearing condition:TCR9 = 1 when CAN-ID of received message is equal to the ID of a Mailbox with CCM set 1: CR0 is enabled and captures the Timer value at every SOF

Bit	Bit Name	Initial Value	R/W	Description
13	TCR13	0	R/W	**TimeStamp Control for Reception** Specifies if the Timestamp in the message control of each Mailbox is recorded at the Start Of Frame (SOF) or End Of Frame (EOF) when a message is received. In effect, this bit selects the trigger for the Input Capture Register 1 (ICR1) that is used to timestamp for transmission Mailboxes. 0: Timestamp is recorded at the SOF of every message received 1: Timestamp is recorded at the EOF of every message received
12	TCR12	0	R/W	**TimeStamp Control for Transmission** Specifies if the Timestamp of each transmit Mailbox is recorded at the point that the corresponding TXPR bit is set or the corresponding TXACK is set when a transmission request is made. This bit selects the trigger for the Input Capture Register 1 (ICR1) that is used to timestamp for receiving Mailboxes. The Input Capture Register 1 (ICR1) is used to timestamp, as the ICR0 can be enabled or disabled. 0: Timestamp is recorded at the point that the TXPR bit is set for message transmission 1: Timestamp is recorded at the point that the TXACK bit is set for message transmission
11	TCR11	0	R/W	**Timer Clear-Set Control by TCMR0** Specifies if the Timer is to be cleared and set to the LOSR when the TCMR0 matches to the TCNTR. Please note that the TCMR0 is also capable to generate an interrupt signal to the host processor via IRR15. 0: Timer is not cleared by the TCMR0 1: Timer is cleared by the TCMR0

Bit	Bit Name	Initial Value	R/W	Description
10	TCR10	0	R/W	**Timer Clear-Set Control by CCM** Specifies if the Timer is to be cleared and set to the LOSR by the CAN-ID Compare Match featured for receive Mailboxes. When a Mailbox stores a received message, the Timer counter (TCNTR) is automatically cleared and set to the LOSR, if the CCM bit of the corresponding Mailbox and this bit is set. Please note that the CCM is NOT capable of generating an interrupt signal since this can be performed by the Message Receive Interrupt (IRR1) or Remote Frame Request Interrupt (IRR2). 0: Timer is not cleared-set by CCM 1: Timer is cleared and set to LOSR by the CCM
9	TCR9	0	R/W	**ICR0 Automatic Disable by CCM:** specifies if the ICR0 is to be disabled by the CAN-ID Compare Match (CCM) featured for receive Mailboxes. When a Mailbox stores a received message, the Bit14 of this register (TCR14) is automatically cleared and the value of ICR0 is maintained, if the CCM bit of the corresponding Mailbox and this bit is set. 0: TCR14 is not cleared by CCM 1: TCR14 is automatically cleared by CCM
8	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
7	TCR7	0	R/W	**Drift Correction Control** Specifies if the TCNTR is to be incremented by +2 or +0 everytime the TCNTR reaches the cycle specified by the DCR. If this function is not required, the TDCR must be set to '0000' (hex). 0: Timer is incremented by +0 (i.e. stays the same value for one source clock cycle) every cycle specified by TDCR. 1: Timer is incremented by +2 within the cycle specified by TDCR (please refer to TDCR section).
6	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.

Bit	Bit Name	Initial Value	R/W	Description
5	TPSC5	0	R/W	**HCAN Timer Prescaler (TPSC[5:0])**
4	TPSC4	0	R/W	This control field allows the timer source clock (2*[HCAN system clock]) to be divided before it is used for the timer. The following relationship exists between source clock period and the timer period
3	TPSC3	0	R/W	
2	TPSC2	0	R/W	
1	TPSC1	0	R/W	
0	TPSC0	0	R/W	000000: 1 × Source Clock 000001: 2 × Source Clock 000010: 4 × Source Clock 000011: 6 × Source Clock 000100: 8 × Source Clock : 111111: 126 × Source Clock

Timer Status Register n (TSR n) (n = 0, 1)

This register is a 16 bit read-only register, and allows the CPU to monitor the Timer Compare Match status and the Timer Overrun Status.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												TSR4	TSR3	TSR2	TSR1	TSR0
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 5	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
4	TSR4	0	R	**HCAN Timer Status (TSR[4:0])**
3	TSR3	0	R	This read-only field allows the CPU to monitor the status of the Cycle Counter, the Timer and the Compare Match registers. Writing to this field has no effect.
2	TSR2	0	R	
1	TSR1	0	R	
0	TSR0	0	R	**Bit 4: Cycle Counter Overflow (TSR4)** Indicates that the Cycle Counter has reached its maximum value and is reset to H'0. Please note that setting CMAX = 0 produces the Cycle Counter to be disabled and TSR.4 to be always cleared to '0'. 0: Cycle Counter has not overflow Clearing condition: Writing '1' to IRR10 (Cycle Counter Overflow Interrupt) 1: Cycle Counter has overflow Setting condition: When the Cycle Counter value changes from the maximum value (CMAX) to H'0 **Bit 3: Timer Compare Match Flag 2 (TSR3)** Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 2 (TCMR2). When the value set in the TCMR2 matches to the Timer value (TCMR2 = TCNTR), this bit is set. Please note that this bit is not set if the TCMR2 value is H'0000. Also, please note that this bit is read-only and is cleared when IRR11 (Timer Compare Match Interrupt 2) is cleared. 0: Timer Compare Match has not occurred to the TCMR2 Clearing condition: Writing '1' to IRR11 (Timer Compare Match Interrupt 2) 1: Timer Compare Match has occurred to the TCMR2 Setting condition: TCMR2 matches to the Timer value (TCMR2 = TCNTR)

Bit	Bit Name	Initial Value	R/W	Description
0				**Bit 2: Timer Compare Match Flag 1 (TSR2)** Indicates that a Compare-Match condition occurred to the Timer Compare Match Register 1 (TCMR1). When the value set in the TCMR1 matches to the Timer value (TCMR1 = TCNTR), this bit is set. Please note that this bit is not set if the TCMR1 value is H'0000. Also, please note that this bit is read-only and is cleared when IRR15 (Timer Compare Match Interrupt 1) is cleared. 0: Timer Compare Match has not occurred to the TCMR1 Clearing condition: Writing '1' to IRR15 (Timer Compare Match Interrupt 1) 1: Timer Compare Match has occurred to the TCMR1 Setting condition: TCMR1 matches to the Timer value (TCMR1 = TCNT) **Bit 1: Timer Compare Match Flag 0 (TSR1)** Indicates that a Compare-Match condition occurred to the Compare Match Register 0 (TCMR0). When the value set in the TCMR0 matches to the Timer value (TCMR0 = TCNTR), this bit is set. Please note that this bit is not set if the TCMR0 value is H'0000. Also, please note that this bit is read-only and is cleared when IRR14 (Timer Compare Match Interrupt 0) is cleared. 0: Compare Match has not occurred to the TCMR0 Clearing condition: Writing '1' to IRR14 (Timer Compare Match Interrupt 0) 1: Compare Match has occurred to the TCMR0 Setting condition: TCMR0 matches to the Timer value (TCMR0 = TCNTR) **Bit 0: Timer Overrun (TSR0)** Indicates that the Timer has overrun and is reset to H'0000. Please note that this bit is set even when the TCMR0 is set to H'FFFF and is enabled to clear the Timer value. 0: Timer has not overrun Clearing condition: Writing '1' to IRR13 (Timer Overrun Interrupt) 1: Timer has overrun Setting condition: When the Timer value changes the value from H'FFFF to H'0000

Timer Mode Register (TMR)

This register is a 16-bit read/write register. It is used to specify the value to be used for the timer functions.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													TMR3	TRM2	TMR1	
Init value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.
3	TMR3	0	R/W	**TimeStamp value** Specifies if the Timestamp for transmission and reception must contain the Timer value (TCNTR) or the concatenation of Cycle_Counter + TCNTR[15:4]. This feature is very useful for time triggered transmission. 0: TCNTR[15:0] is used for the TimeStamp 1: Cycle_Counter + TCNTR[15:4] is used for the TimeStamp
2	TRM2	0	R/W	**TCMR2 control** Specifies if the Timer Compare Match 2 must be compared with the Timer value (TCNTR) or with Cycle_Counter + TCNTR[15:4]. 0: TCMR2 = TCNTR[15:0] is used for TCMR2 1: TCMR2[15:12] = Cycle_Counter AND TCMR2[11:0] = TCNTR[15:4] is used for TCMR2
1	TMR1	0	R/W	**TCMR1 control** Specifies if the Timer Compare Match 1 must be compared with the Timer value (TCNTR) or with Cycle_Counter + TCNTR[15:4]. 0: TCMR1 = TCNTR[15:0] is used for TCMR1 1: TCMR1[15:12] = Cycle_Counter AND TCMR1[11:0] = TCNTR[15:4] is used for TCMR1
0	—	0	R	**Reserved** The written value should always be '0' and the returned value is not guaranteed.

Timer Drift Correction Register (TDCR)

This register is a 16-bit read/write register. The purpose of this register is to compensate the drift of the Timer caused by a different clock running at other CAN nodes on the same system. When the TCNTR reaches to the cycle specified by this register, the Timer value is incremented by +2 or +0 (i.e. stays at the same value). Please note that this register does not point at a specific time but a specific cycle. This means, if TCNTR/2 > TDCR, then the drift correction will be performed more than twice (unless the TCMR0 is used to clear the TCNTR before it reaches the second cycle). When this TDCR register is set to '0000' (hex), the drift correction will not be performed at all.

Please note that for a proper operation of the timer the maximum programmable value must be TDCR <= 8000 (hex).

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TDCR [15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TDCR [15:0]	0	R/W	**Timer Drift Correction Register (TDCR)** Indicates the value of the cycle to compensate the drift of the Timer (TCNTR).

Local Offset Register (LOSR)

This register is a 16-bit read/write register. The purpose of this register is to set a local offset to the Timer TCNTR. Whenever the TCNTR is cleared by Overflow or Timer Compare Match or CAN-ID Compare Match, the TCNTR starts running at the value set in this register.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOSR[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	LOSR[15:0]	0	R/W	**Local Offset Register (LOSR)** Indicates the value of the local offset for the Timer (TCNTR) to start with.

Cycle Counter Register (CCR)

This register is a 4-bit read/write register. Its purpose is to store the number of the base cycle for TT Transmissions. Its value is incremented by one every time the free running counter (TCNTR) is cleared to zero by a Compare Match condition on TCMR0.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CCR[3:0]			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	**Reserved**
3 to 0	CCR[3:0]	0	R/W	**Cycle Counter Register (CCR)** Indicates the number of the current Base Cycle of the matrix cycle for Timer Triggered transmission.

Cycle Maximum Register (CMAX)

This register is a 4-bit read/write register. Its purpose is to store the maximum value for the cycle counter (CCR) for TT Transmissions to set the number of basic cycles in the matrix system. When the Cycle Counter reaches the maximum value (CCR = CMAX) the Cycle Counter is cleared to zero and an interrupt is generated on IRR.10.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													CMAX[3:0]			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	**Reserved**
3 to 0	CMAX[3:0]	0	R/W	**Cycle Maximum Register (CMAX)** Indicates the number of basic cycles available in the matrix cycle for Timer Triggered transmission. The initial value of CMAX is '0' producing the Cycle Counter to be disabled. During the configuration the requested value must be programmed.

Input Capture Registers n (ICR0-cc n, ICR0-tm n, ICR1 n) (n = 0, 1)

The Input Capture registers are composed of one 4-bit read/write register (ICR0_cc) and two 16-bit read/write registers (ICR0_tm and ICR1).

ICR0_cc: The ICR0_cc can be used for Global Synchronization purpose, when used with ICR0_tm. The current Basic Cycle value (Cycle_Counter) is captured at the SOF if it is enabled by the Bit14 in the TCR, regardless whether the received message matches to the identifiers set in the receive Mailboxes or not. If it is disabled by the Bit14 in the TCR, the ICR0_cc holds the current value.

ICR0_tm: The ICR0_tm can be used for Global Synchronization purpose, when used with ICR0_cc. The Timer value is captured at the SOF if it is enabled by the Bit14 in the TCR, regardless whether the received message matches to the identifiers set in the receive Mailboxes or not. If it is disabled by the Bit14 in the TCR, the ICR0_tm holds the current value.

ICR1: The ICR1 is used to record the timestamp for messages to be transmitted and received. The Bit13 (for receive) and Bit12 (for transmit) in the TCR control register at which point the timestamp should be recorded. The difference to the ICR0 is that the ICR1 cannot be disabled so that the timestamps recorded on messages are always accurate.

- ICR0-cc (Address = H'08A)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													ICR0-cc[3:0]			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*	R/W*	R/W*	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 4	—	0	R	
3	ICR0-cc[3:0]	0	R/W*	This register samples the value of the cycle counter register (CCR) at every SOF on the CAN Bus when enabled by TCR[14].
2		0	R/W*	
1		0	R/W*	
0		0	R/W	

Note: * This registers can be written, however, has no effect.

- ICR0-tm (Address = H'08C)/ICR1 (Address = H'08E)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ICR0-tm[15:0] , ICR1[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*	R/W*

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ICR0-tm[15:0]	0	R/W*	This register samples the value of the timer (TCNTR) at every SOF on the CAN Bus when enabled by TCR[14].

Note: * This registers can be written, however, has no effect.

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	ICR1[15:0]	0	R/W*	This register samples the value of the timer (TCNTR) at the condition specified by the Bit13 (for reception) and Bit12 (for transmission) in the TCR control register.

Note: * This registers can be written, however, has no effect.

Timer Compare Match Registers n (TCMR0 n, TCMR1 n, TCMR2 n) (n = 0, 1)

TCMR0 (Address = H'090)/TCMR1 (Address = H'092)/TCMR2 (Address = H'094)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TCMR0[15:0] , TCMR1[15:0], TCMR2[15:0]															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 0	TCMR0[15:0] , TCMR1[15:0], TCMR2[15:0]	0	R/W	These registers are used to set up three values to be compared with the rimer (TCNTR) to generate specific actions (see following explanation).

These three registers are 16 bit read/write registers and are capable of generating interrupt signals, clearing-setting the Timer value (only supported by TCMR0) or clear the transmission messages in the queue (only supported by TCMR2). Both registers offer exactly the same function except for the clear of the Timer and the clear of the transmission. The value used for the compare can be configured independently for each register, using the bits 1, 2 and 3 of TMR (Timer Mode Register), to be the timer value (TCNTR[15:0]) or the concatenation of Cycle_Count + TCNTR[15:4] value.

Interrupt: The interrupts are flagged by the Bit15, Bit14 and 11 in the IRR accordingly when a Compare Match occurs, and these bits cannot be prevented from being set in the IRR except the TCMR value is H'0000. The generation of interrupt signals itself can be prevented by the Bit15, Bit14 and Bit11 in the IMR. When a Compare Match occurs and the IRR15 (or IRR14 or IRR11) is set, the Bit2 or Bit1 or Bit3 in the TSR (HCAN Timer Status Register) is also set. Clearing the IRR bit also clears the corresponding bit of TSR.

Timer Clear-Set: The Timer value can only be cleared and set to the LOSR by the TCMR0 when a Compare Match occurs if it is enabled by the Bit11 in the TCR. TCMR1 and TCMR2 do not have this function.

Cancellation of the messages in the transmission queue: The messages in the transmission queue can only be cleared by the TCMR2 when a Compare Match occurs. TCMR1 and TCMR0 do not have this function.

When HCAN2 is used in TTCAN mode these registers can be used as follow.

TCMR0: To specify the length of the basic cycle

TCMR1: To generate Interrupt on specified time (i.e. when a reception/transmission is due or at the beginning of a arbitrating time window) to monitor sheduled reception/transmission or to trigger the application to set transmission for event triggered messages.

TCMR2: To abort all transmission pending on specified time (i.e. when a watch trigger is reached).

19.4 Application Note

19.4.1 Test Mode Settings

The HCAN has various test modes. The register TST[7:0] (MCR[15:8] is used to select the HCAN test mode. The default (initialized) settings allow HCAN to operate in Normal mode. The following table is examples for test modes.

Bit15: TST7	Bit14: TST6	Bit13: TST5	Bit12: TST4	Bit11: TST3	Bit10: TST2	Bit9: TST1	Bit8: TST0	Description
0	0	0	0	0	0	0	0	Normal Mode (initial value)
1	0	0	0	1	0	1	0	Listen-Only Mode (Receive-Only Mode)
1	0	0	1	—	0	0	0	Self Test Mode 1 (External)
1	0	0	1	—	1	1	1	Self Test Mode 2 (Internal)
1	1	0	—	—	—	—	—	Error Passive Mode 1
1	—	1	—	—	—	—	—	Error Passive Mode 2

Normal Mode: HCAN operates in the normal mode.

Listen-Only Mode: ISO-11898 requires this mode for baud rate detection etc. The Error Counters are disabled so that the TEC/REC does not increase the values, and the Tx Output is disabled so that HCAN does not generate error frames.

Self Test Mode 1: HCAN generates its own Acknowledge bit. The Rx/Tx pins must be connected to the CAN bus.

Self Test Mode 2: HCAN generates its own Acknowledge bit. The Rx/Tx pins do not need to be connected to the CAN bus or any external devices, as the internal Tx is looped back to the internal Rx.

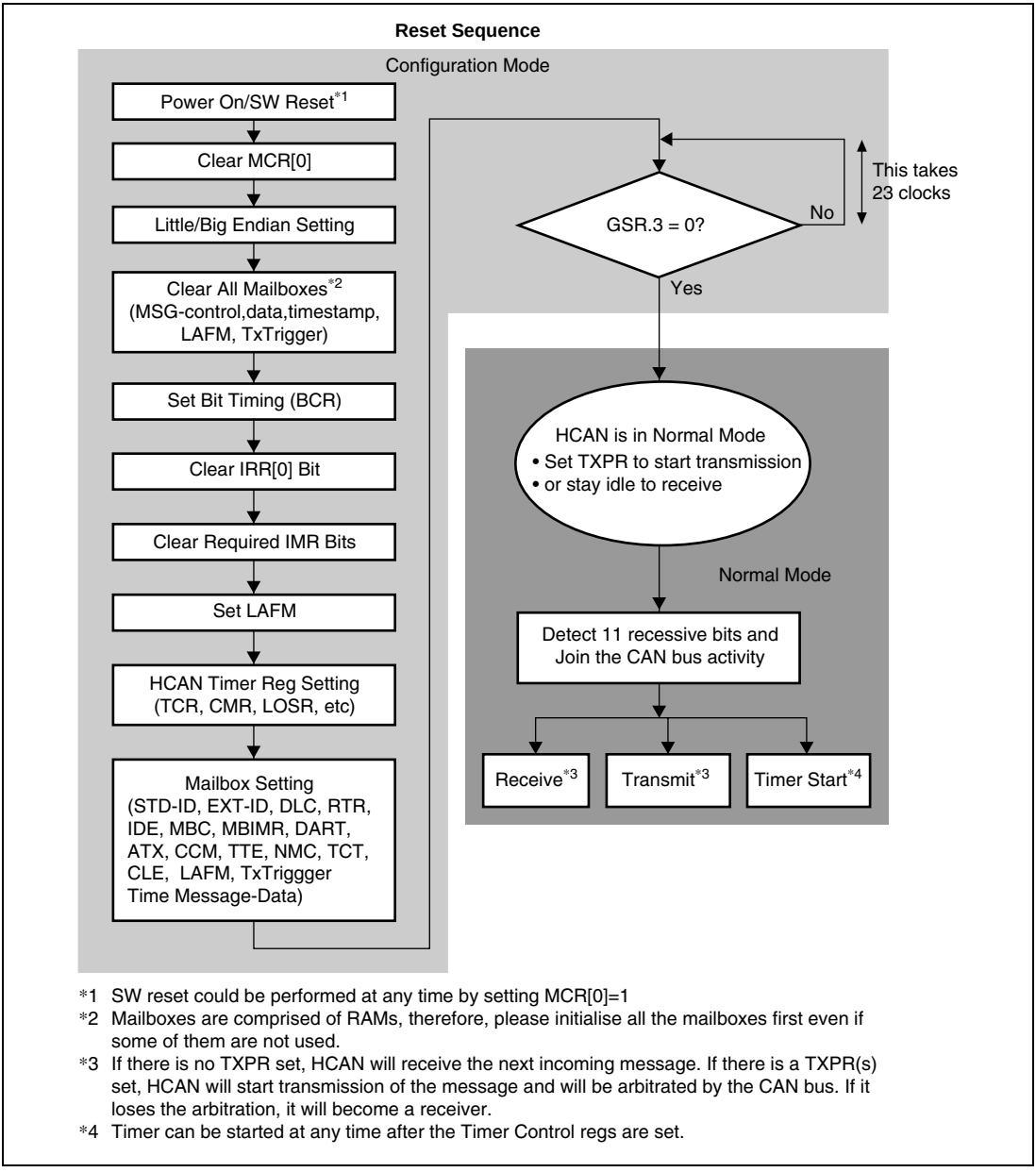
Error Passive 1: HCAN can be forced to become an Error Passive node by writing a value (greater than 127) into the Error Counters. (MCR1 must be '1' when writing to the Error Counter). The value written into TEC is used to write into REC, so only the same value can be set to these registers. Also, HCAN needs to be put into Halt Mode when writing into TEC/REC.

Error Passive 2: HCAN can be forced to become an Error Passive node by setting the TST5.

19.4.2 Configuration of HCAN

Reset Sequence

The following sequence is an example to configure the HCAN after (S/W or H/W) reset. After reset, all the registers are initialized, therefore, HCAN needs to be configured before joining the CAN bus activity. Please read the notes carefully.



*1 SW reset could be performed at any time by setting MCR[0]=1
*2 Mailboxes are comprised of RAMs, therefore, please initialise all the mailboxes first even if some of them are not used.
*3 If there is no TXPR set, HCAN will receive the next incoming message. If there is a TXPR(s) set, HCAN will start transmission of the message and will be arbitrated by the CAN bus. If it loses the arbitration, it will become a receiver.
*4 Timer can be started at any time after the Timer Control regs are set.

Figure 19.7 Reset Sequence

19.4.3 Message Transmission Sequence

Event Triggered Transmission

Message Transmission Request: The following sequence is an example to transmit a CAN frame onto the bus. As described in the previous register section, please note that IRR[8] is set when one of the TXACK or ABACK bits is set, meaning one of the Mailboxes has completed its transmission or transmission abortion and is now ready to be updated for the next transmission, whereas, the GSR[3] means that there is currently no transmission request made (TXPR = H'0000).

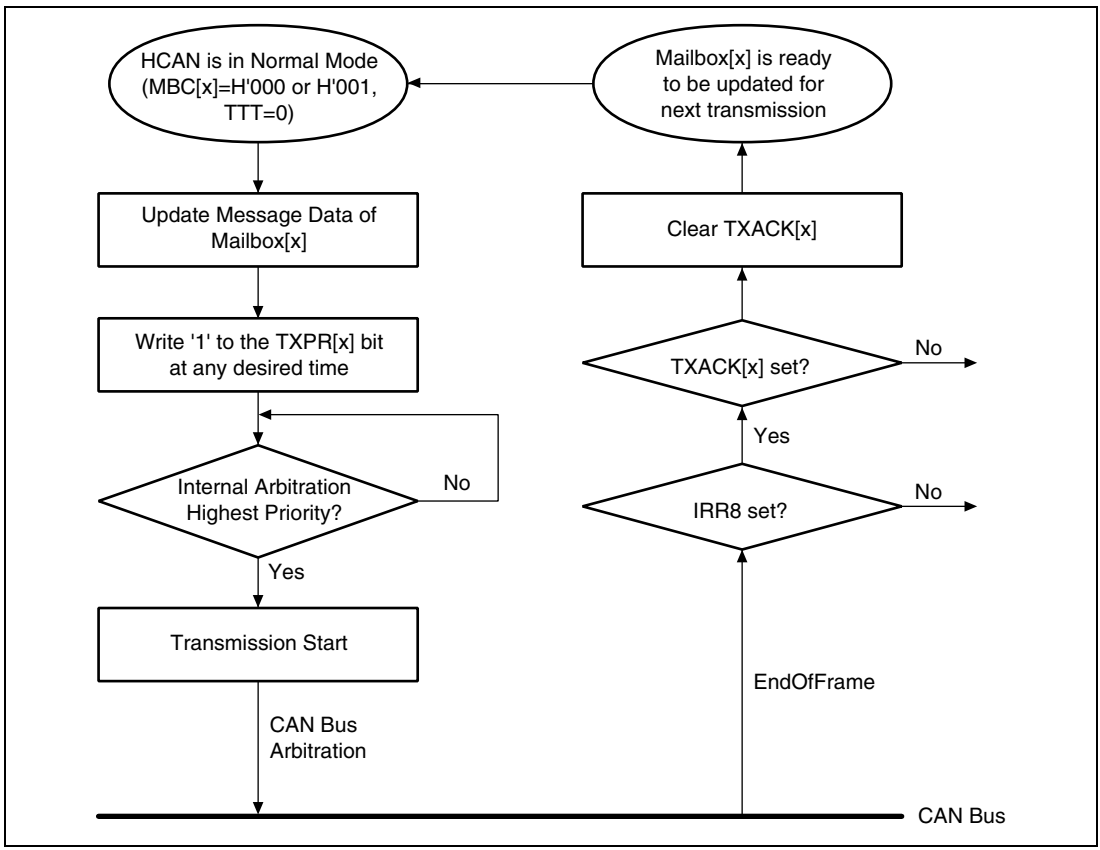


Figure 19.8 Transmission request

Internal Arbitration for transmission: The following diagram explains how HCAN manages to schedule transmission-requested messages in the correct order based on the CAN identifier. 'Internal arbitration' picks up the highest priority message amongst transmit-requested messages.

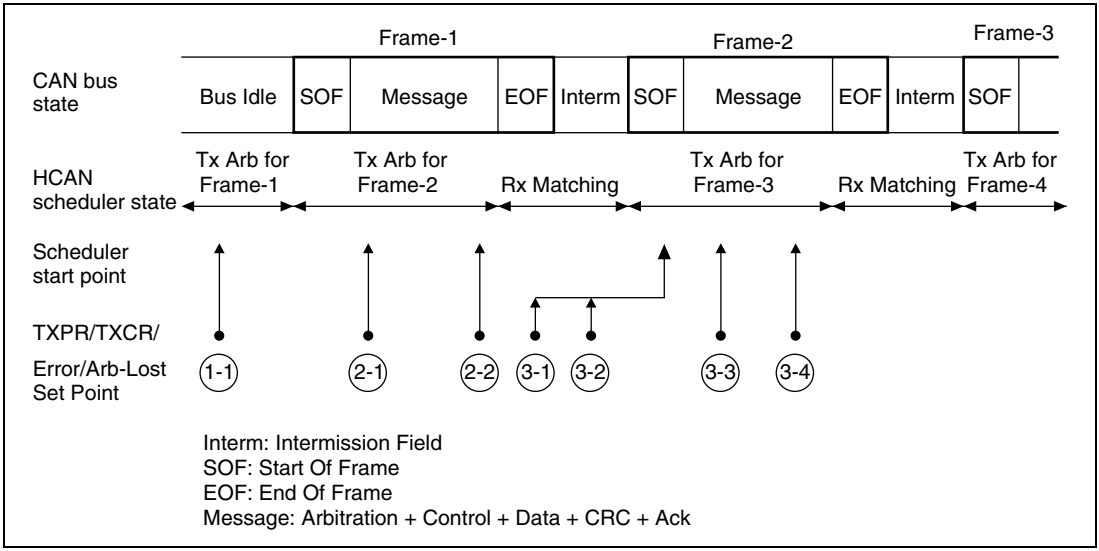


Figure 19.9 Internal Arbitration for transmission

The HCAN Scheduler, which runs internal arbitration, has 2 states – Tx Arbitration State and Rx Matching State. The HCAN Scheduler is in the Rx Matching State if the CAN bus is in the EOF or Intermission cycles, or otherwise is in the Tx Arbitration State. When a transmission (or transmission abortion) request is made in the Tx Arbitration State, the internal arbitration starts running immediately. When a transmission (or transmission abortion) request is made in the Rx Matching State, the internal arbitration waits until the Rx Matching State (i.e. Intermission field) is finished, and then starts running as soon as the HCAN scheduler state becomes 'Tx Arbitration'.

There are 4 factors that can run internal arbitration, which are:

- TXPR is set
- TXCR is set (please note that, if TXCR is set for the message currently under transmission, HCAN does not stop the transmission but completes. If the message loses the bus arbitration or causes an error on the bus, HCAN will cancel the transmission request.)
- Error occurs on the CAN bus
- Message under transmission loses the arbitration on the CAN bus
- Mailbox with the setting MBC = B'001 receives a Remote Frame

Whenever these factors happen, the internal arbitration starts running to ensure that the highest priority message is always transmitted first. The followings are examples set in the diagram.

1-1: When a TXPR bit(s) is set while the CAN bus is idle, the internal arbitration starts running immediately and the transmission is started.

2-1, 2-2: During this period (Tx Arb for Frame-2), whenever or however many times any of the 4 factors occurs, the internal arbitration starts running and scheduled for the next frame (Frame-2) to be transmitted.

3-1, 3-2: During this period (Rx Matching), any internal arbitration is not allowed to run, but scheduled later at the SOF of the next frame (Frame-2). If the transmit-requested message has the highest priority, the transmission will be set for the Frame-3.

3-3, 3-4: This is the same case as 2-1, 2-2.

Time Triggered Transmission

Message Transmission Request: The following sequence is an example of how to transmit a CAN frame onto the bus.

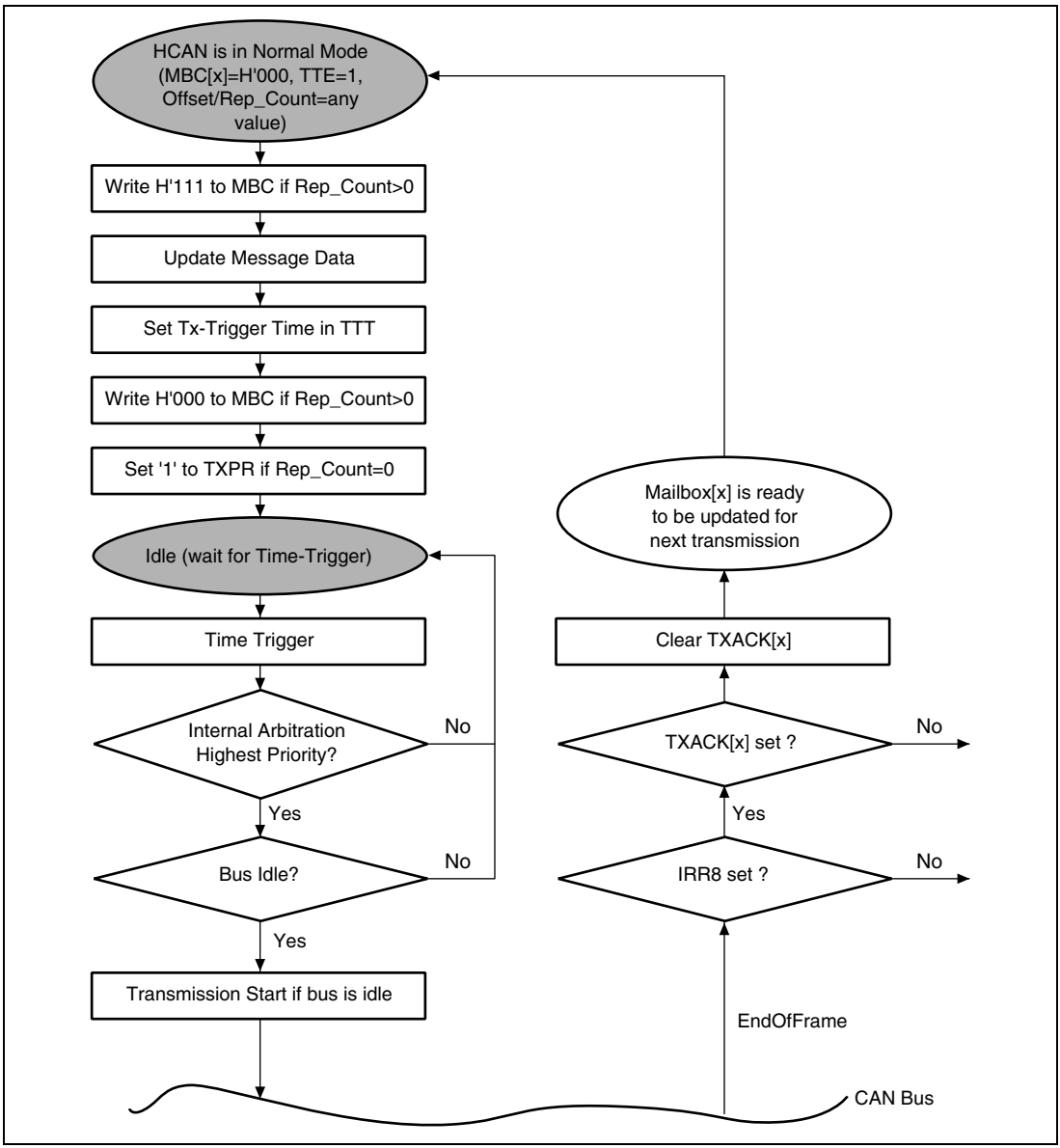


Figure 19.10 Time Triggered Transmission

When the Periodic Transmission is set (Rep_Count > 0) the MBC needs to be set to H'111 to make the Mailbox inactive before updating the message data. This is to prevent HCAN from transferring the message data into the Tx buffer in the middle of the update of the message data. This may not be required if the S/W can ensure that a message is updated before a Tx trigger occurs.

When the TCNTR reaches to TTT (Tx Trigger Time) of a Mailbox, HCAN immediately transfers the message into the Tx buffer. At this point, the bus must be idle (or Intermission) for HCAN to enable the transmission.

The TXPR can be modified at any time. HCAN ensures the transmission of Time Triggered messages is always scheduled correctly. However, in order to guarantee the correct schedule, there are some important rules that are:

- TTT (Tx Trigger Time) cannot be modified once the TXPR bit has been set. If the TTT needs to be modified, the TXPR has to be cleared by setting the corresponding TXCR bit.
- TTT cannot be set outside the range of TCNTR if TCMR0 is used to clear-set the TCNTR. This could cause a scheduling problem.
- TXPR is not cleared for periodic transmission (Rep_Count > 0). If a periodic transmission needs to be cancelled, the corresponding TXCR bit needs to be set to clear the TXPR bit.
- During a Time Triggered Transmission only another one message can be triggered and a time difference of 200 system clock cycles must be inserted between them.
- If HCAN is not the time master of the communication it is necessary to clear all transmissions at the end of each basic cycle and set them back again after the synchronization sequence. This is to guarantee that a transmission is not attempted when the node is not synchronized (reference message not received).

Automatic re-transmission of Time Triggered messages: Within a time triggered system the re-transmission of a message on the CAN Bus must be enabled/disabled depending of the type of message.

A short description is presented hereafter.

Reference message**:** The reference message must be re-transmitted on the CAN Bus by a (potential) time master if one of the following conditions occurs:

1. It is disturbed by an error on the CAN Bus
2. It looses the arbitration on the CAN Bus
3. When it is scheduled to start but the CAN Bus is busy with an error
4. When it is scheduled to start but the CAN Bus is busy with another message that is not a valid reference message with higher priority
5. A reference message is received with a wrong DLC field
6. It is not acknowledged on the CAN Bus and the watch trigger is not reached

Messages scheduled to be transmitted on merged arbitrating time windows**:** A message programmed to be transmitted on a merged arbitrating time window must be retransmitted on the CAN Bus if one of the following conditions occurs when the transmission enable windows is not close:

7. It is disturbed by an error on the CAN Bus
8. It looses the arbitration on the CAN Bus
9. When it is scheduled to start but the CAN Bus is busy with an error
10. When it is scheduled to start but the CAN Bus is busy with another message
11. It is not acknowledged on the CAN Bus

All other messages: All other messages on a TTCAN system must have their automatic retransmission disabled.

HCAN-2 is re-transmitting a time triggered message when the corespondent DART bit (Disable Automatic Re-Transmission) is not set (DART = '0') only on the cases d) and j) above. On all the other cases the automatic retransmission is not performed. This does not cause any problem for a reference message. In fact for the nature of a time triggered network there must be at least another potential time master and, if the reference message from the current time master is missing, the other one(s) must send its own reference message. Then no specific actions are requested to the application.
With regards to the messages to be transmitted on merged arbitrating time windows it is suggested to configure the related Mailbox on event triggered mode (TTE = ′0′) and use the timer compare match register 1 (TCMR1) to trigger the application to set the transmission for the message. For event triggered transmission the message is always re-transmitted if one of the above condition (a to k) occurs on the CAN Bus.

The transmission enable window: The TTCAN ISO working draft, ISO-WD-11898-4, specifies a trigger signaling the beginning of each interested time slot and an enable window (Tx_Enable_Window) where the TTCAN node must be enabled to start the transmission within the time slot itself. This transmission window can be from 1 to 16 nominal CAN bit time.

Then, for example, if when the time slot begins (Tx_start) the CAN Bus is busy the transmission can be delayed up to Tx_start+ Tx_Enable_Window.

HCAN-2 is working in a slight different way. The application cannot define on H/W the beginning of the time slot and the length of the enable window but the exact point in time where the transmission needs to start. Then the transmission will be performed only at the scheduled transmission time. It is suggested to define the TTCAN network and the TTCAN message scheduling to set the transmission time on the related Mailbox considering the Tx_Enable_Window.

Example of Time Triggered System: The following diagram shows a simple example of how time trigger system works using HCAN.

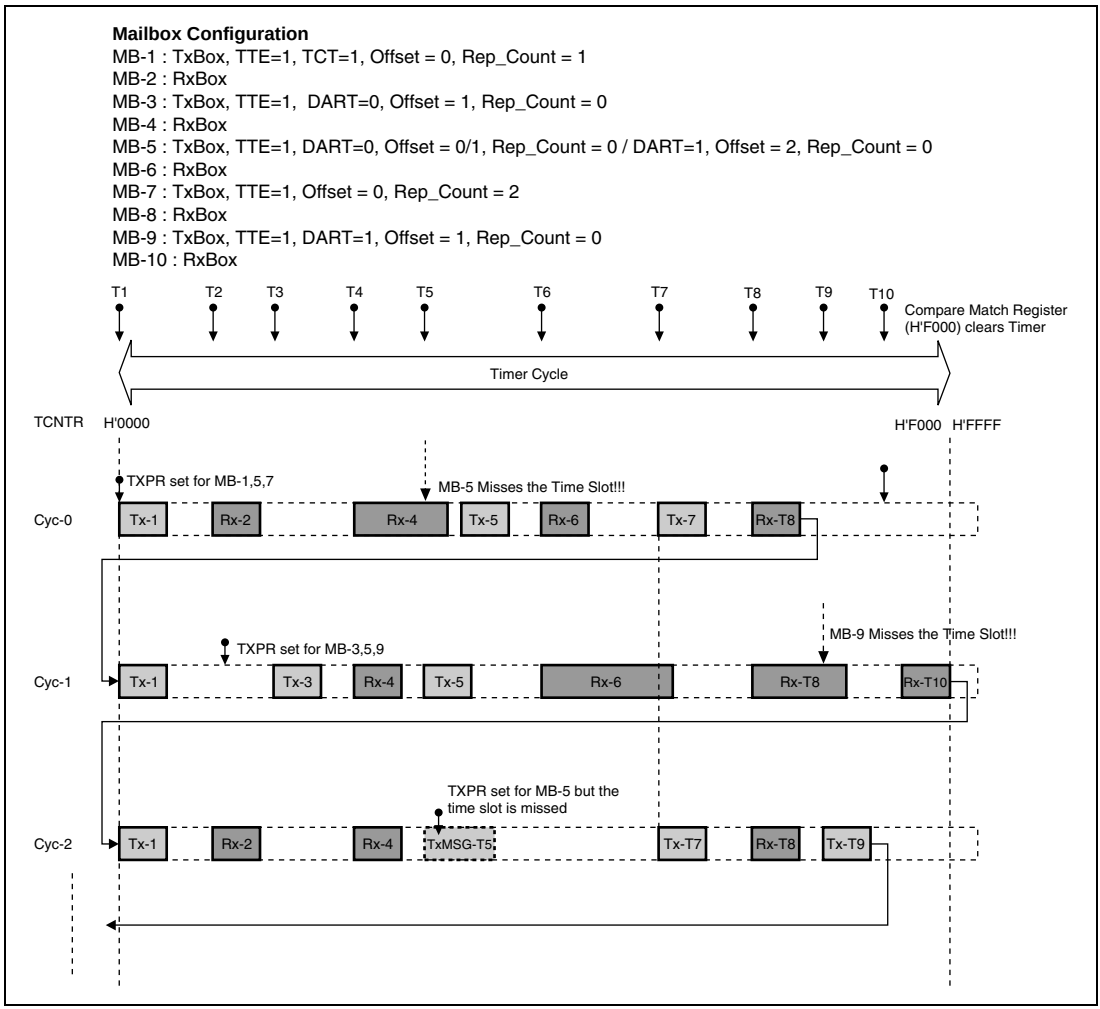


Figure 19.11 Example of Time Triggered System

The Mailbox-1, 3, 5, 7 and 9 are configured as transmit boxes with their TTE (Time Trigger Enable) bits set, and the Mailbox-2, 4, 6, 8 and 10 are configured as receive boxes. The transmission from Mailbox-1 and Mailbox-7 are periodic and respectively at each Cycle (Rep_Count = 1) and each two Cycle (Rep_Count = 2). Mailbox-5 is used with different settings (please refer to following description).

The system has 10 Time Windows – T1 to T10, 10 basic cycles (CMAX = 9), and the TCNTR is cleared (LOSR = H'0000) by TCMR0 that is set to H'F000.

Cycle-0: TXPR is set for Mailbox-1, 5 and 7 to transmit. The Mailbox-1 fails to transmit the message as it takes for a while for HCAN to complete the internal arbitration. The Mailbox-5 fails to transmit at the scheduled time as the message coming in T4 timing exceeds T5 timing but as DART is not set, the transmission occurs as soon as the bus becomes free. The Mailbox-7 transmits the message at the scheduled time (T7).

Cycle-1: The Mailbox-1 transmits the message again at T1 timing as its transmission is set for every Base Cycle. After this, TXPR is set for Mailbox-3, 5 and 9. HCAN transmits both messages at T3 and T5 according to the schedule. A message comes in at T8 timing, exceeding the T9 timing. The Mailbox-9, with the DART bit set, cannot transmit the message because of the bus occupied by another CAN node, and misses the time slot. The Mailbox-9 waits for the next time slot – T9.

Cycle-2: The Mailbox-1 (periodic every base cycle) and Mailbox-7 (periodic every 2 base cycle) transmit at the scheduled timing as the bus is free. The TXPR is set for Mailbox-5 just after the T5 timing, and misses the time slot. The TXPR will be kept and transmitted at the 2nd cycle of the next Matrix cycle as DART is set.

Apart from the settings above, the following options may be useful, too.

- Transmit the TCNTR and Cycle_count value at the SOF in the TxMSG-T1 by setting the TCT bit, acting as Time Master
- Disable ICR0 by setting the CCM bit to compare the HCAN's local time against the global time received
- Adjust Timer value by setting LOSR (Local Offset Register) and TDCR (Timer Drift Correction Register)
- Use TCMR1 to generate interrupt signals to monitor if messages are received/transmitted on schedule or to trigger transmission of event triggered messages.
- Use TCMR2 to abort all pending messages after a certain time (i.e. after a watch trigger condition is reached).

Mixed Mode Transmission

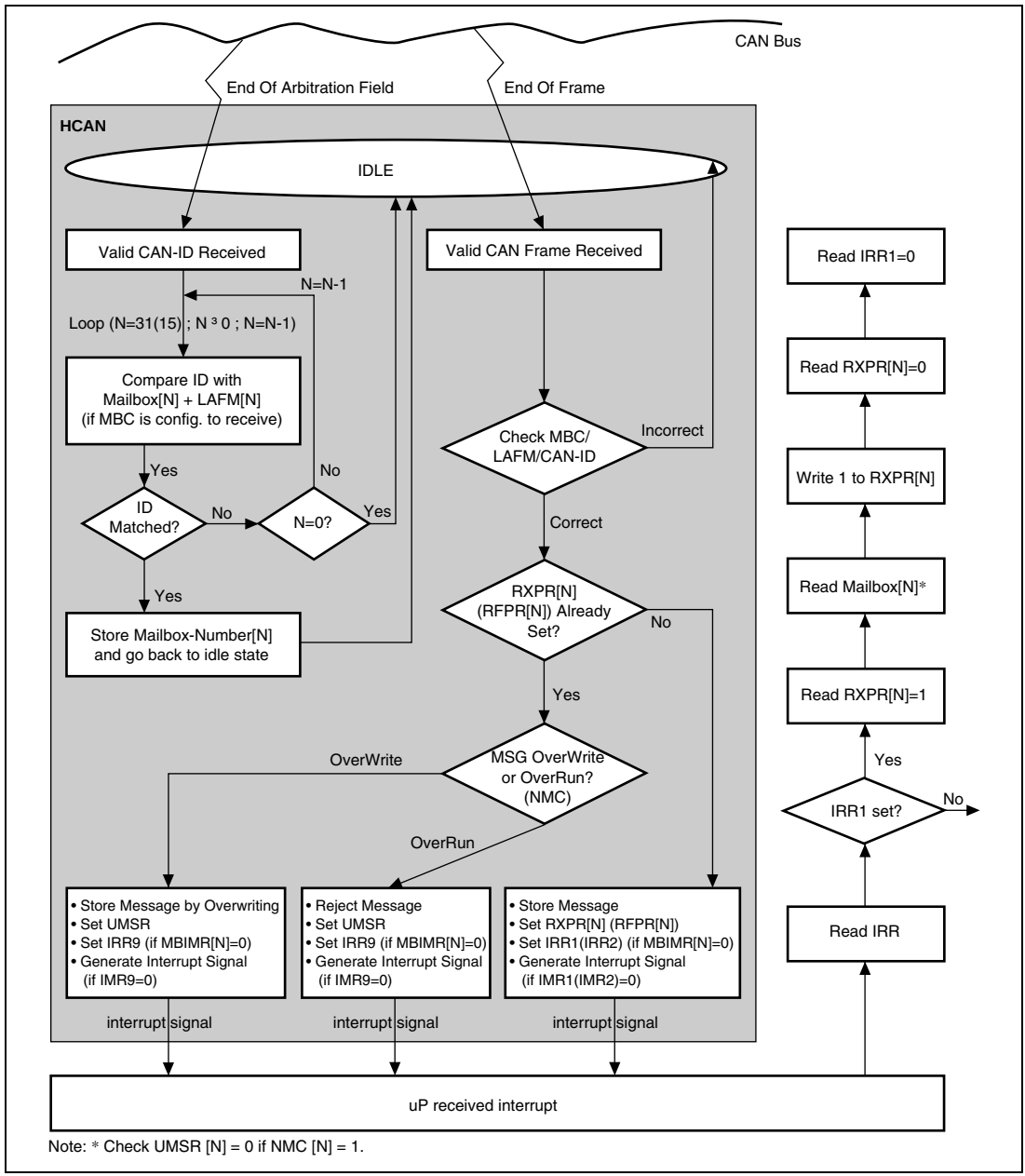
Event Triggered Transmission and Time Trigger Transmission can be mixed. HCAN will still ensure the correct operation. When an event triggered transmission is requested while receiving a message or transmitting a message, internal arbitration will run to pick up the highest priority message and, if it is, it will be transmitted after the current CAN frame.

This is important, for example, in the case that some urgent messages need to be transmitted on a Time-Trigger CAN system, the time schedule can be violated and those urgent messages can be present on the CAN bus immediately.

Please bear in mind that in a real TTCAN system Event Triggered messages can be used only in merged arbitration windows. The application needs to assure that Event Triggered mesaages do not occupy windows reserved to other messages (reference messages, exclusive time windows and arbitrating time windows not merged). This is necessary to guarantee the transmission of all scheduled messages.

19.4.4 Message Receive Sequence

The diagram below shows the message receive sequence.



Note: * Check UMSR [N] = 0 if NMC [N] = 1.

Figure 19.12 Message Receive Sequence

When HCAN recognises the end of the Arbitration field during receiving of a message, it starts comparing the received identifier to the identifiers set in the Mailboxes, starting from Mailbox-31 down to Mailbox-0. It first checks the MBC if it is configured as a receive box, and reads LAFM, and reads the CAN-ID of Mailbox-31 (if configured as receive) to finally compare them to the received ID. If it does not match, the same check takes place at Mailbox-30 (if configured as receive). Once HCAN finds a matching identifier, it stores the number of Mailbox-[N] into an internal buffer, stops the search, and goes back to idle state, waiting for the End Of Frame (EOF) to come. When an EOF is notified by the CAN Interface logic, HCAN this time only reads the MBC, LAFM and CAN-ID of Mailbox-[N] to confirm the matching condition again (i.e., there has been no modification to the configuration of Mailbox-[N]). This re-confirmation guarantees the data consistency even when a Mailbox is re-configured during receiving a message. If it still matches, then the message is written or abandoned, depending on the NMC bit. If it is written into the corresponding Mailbox, including the CAN-ID, i.e., there is a possibility that the CAN-ID is overwritten by a different CAN-ID of the received message due to the LAFM used. This also implies that, if the identifier of a received message matches to ID + LAFM of 2 or more Mailboxes, the higher numbered Mailbox will always store the relevant messages and the lower numbered Mailbox will never receive messages. Therefore, the settings of the identifiers and LAFMs need to be carefully selected.

19.4.5 Reconfiguration of Mailbox

When re-configuration of Mailboxes is required, the following procedures should be taken.

Change ID of transmit box or Change transmit box to receive box: Confirm that the corresponding TXPR is not set. The identifier or the corresponding MBCR bit can be changed at any time. When both need to be changed, please change the identifier first and then the corresponding MBCR bit.

Change ID of receive box or Change receive box to transmit box:
Method-1: Using Halt Mode
The advantage of this method is that HCAN will not lose a message if the message is currently on the CAN bus and HCAN is a receiver. HCAN will be in Halt Mode after completing the reception. The disadvantage is that it might take long if HCAN is receiving a message (as the transmission to the halt state is delayed until the end of the reception), and also HCAN will not be able to receive/transmit messages during the Halt mode.

Method-2: Without Using Halt Mode
The advantage of this method is that the re-configuration is done instantly, and the S/W overhead will be less as there is no interruption. The reason that RXPR needs to be read before and after the re-configuration is to check if a message is received or not during this period. Please note that the MBIMR does not prevent the RXPR bit or the IRR1 from being set but simply prevents the interrupt signal from being generated. If a message is received, it is unknown if the received

message is for the previous ID or for the new ID. Therefore, if a message is received during this period, it is better to abandon this message, and this is the disadvantage of this method.

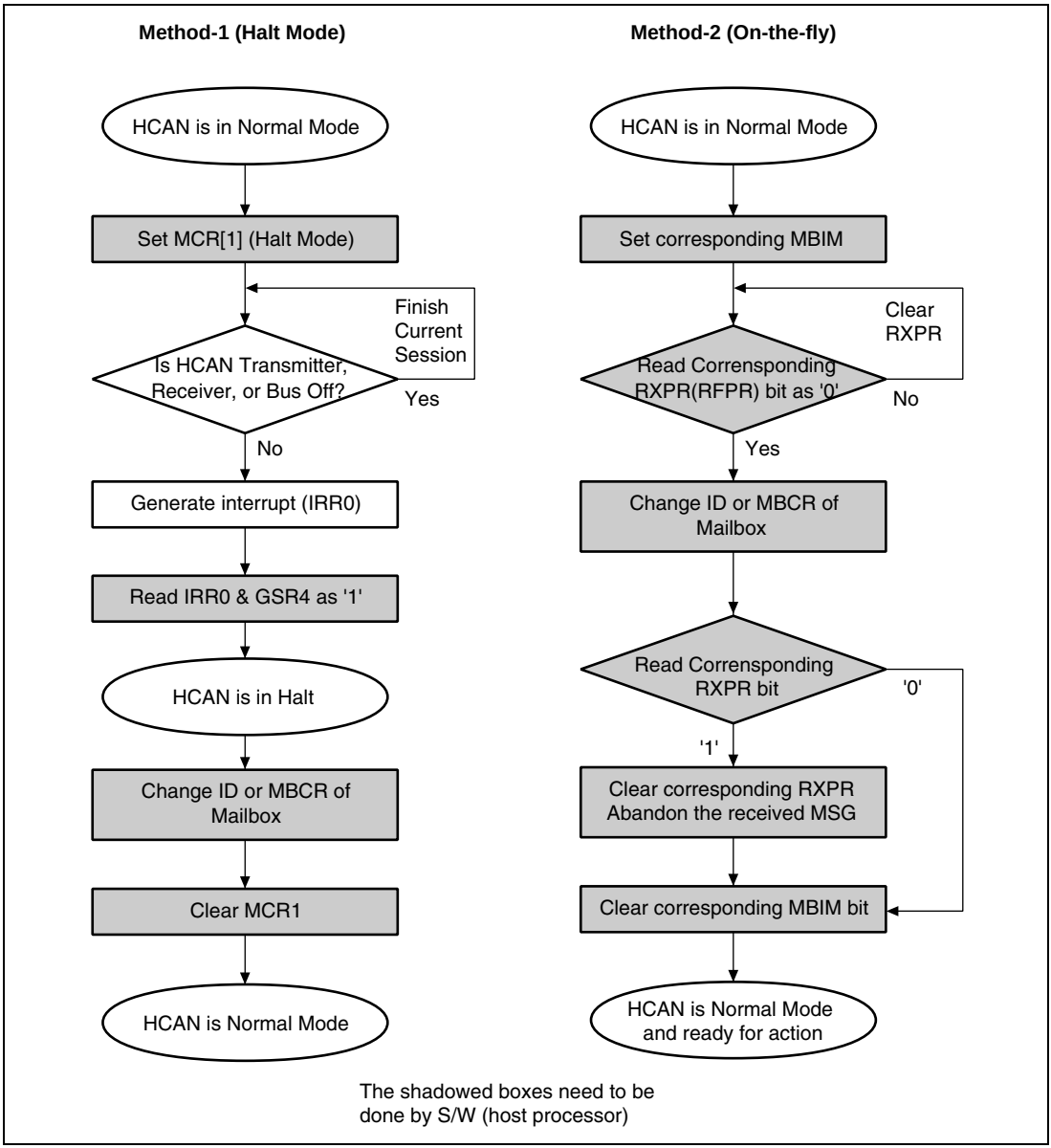


Figure 19.13 Change ID of Receive Box or Change Receive Box to Transmit Box

19.4.6 Global Synchronization

When HCAN is used in time triggered mode the application must assure that it is synchronized with other TTCAN nodes on the CAN Bus and all transmissions are performed only on the allowed time slots. Then internal timer TCNTR can be used as a Local Timer. TCNTR and CCR give the reference point within the programmed system matrix.

In a strict time triggered network working on Level1 the timer must be incremented every Network Time Unit (NTU = 1/Baud Rate). It must be related to the used Baud Rate. In order to fulfil this requirement the frequency used for the HCAN-2 IP must be chosen accordingly checking that the following relation is valid:

$$1 \leq T_{presc} = \frac{f_{clk}}{2 * BitRate} \leq 126$$

Where: T_{presc}: is the prescaler to be used for the embedded timer,

f_{clk}: is the external clock source,

BitRate is the requested Baud Rate
The formula for the HCAN-2 bit rate calculation must also be borne in mind.

If the network is composed only by HCAN2 IPs working in TTCAN mode the timer can be incremented on a different way as long as the same source clock (obtained changing the timer prescaler and the peripheral clock) is the same on all the nodes. In such case the above formula could be omitted.

The ICR0-tm Registers can be used to achieve Global Synchronization on a TTCAN system.

The Global Synchronization is different depending if the HCAN is acting like the time master or time slave of the TTCAN system. Please refer to the ISO spec for the failure handling to switch between time master and potential time master.

Synchronization as a Time master: When HCAN is acting as the time master it is the responsible for the global time in the system. The SW needs to set a periodic transmission at every cycle to send the reference message (tct = '1' in Mailbox Configuration field). The timer and cycle counter values need to be preloaded before enabling the timer (TCR[15]).

Synchronization as a Time slave: When HCAN is acting as a time slave the SW needs to check the reference message to synchronize to the system. It cannot start transmission before the synchronization is completed. TCMR0 must not be set to clear the timer and the length of each cycle must be controlled only with the reception of a reference message.

It is important that a node not synchronized does not start transmission on the CAN Bus. Then all pending transmission (if any) should be cancelled when a reference message is received and set again once the following synchronization sequence is completed.

(1) The ICR0-tm is enabled and capturing the TCNTR at every SOF of CAN frame. The TCR9 (ICR0 automatic disable by CCM) is set to freeze ICR0-tm and ICR0-cc when a reference message is received.

(2) The CAN identifier of the received message matches to the Mailbox dedicated to the reception of the reference message, i.e., receives a reference message, therefore, the TCR14 (ICR0-tm enable) is automatically cleared and the ICR0-tm holds the last value captured at the SOF.

(3) Once the reception is completed an interrupt is generated kicking off the Interrupt Service Routine (ISR).

(4) The ISR must check if the interrupt is related to the reception of a valid reference message.

(5) If a valid reference message has been received the ISR needs to load the timer with the following value:

TCNTR = TCNTR - ICR0_tm if TCNTR>ICR0_tm

TCNTR = ICR0_tm - (16'hffff-TCNTR) if TCNTR<ICR0_tm

Please note that on the above the latency of the SW is not considered. This must be checked on the development phase and, in case, added to the formula.

(6) Enable ICR0 again, to capture the TCNTR at every SOF.

(7) Finally the value of the cycle counter needs to be synchronized with the one of the reference message. For this the ISR needs to copy the value of the cycle counter embedded on the reference message into CCR.

Notice that at this point, as the node is not yet synchronized it should only receive without transmitting any message. A Time Triggered node needs to receive at least two reference message to join transmission on the CAN Bus.

(8) HCAN receives another reference message and freezes the ICR0 again.

(9) Points (3) to (7) above are repeated.

After (9) the HCAN node is synchronized with the master node(s), so transmission can be started.

The points from (3) to (7) should be repeated for every reference message on the CAN Bus as, due to clock drift, the values of the timer can change during the activity.

19.4.7 HCAN module Standby-mode

This HCAN module allows clock gating to reduce power consumption. The module standby mode can be controlled by the Clock Control 1 (CC1) Register in Power Control module with the bit 18 for channel 0 and the bit 19 for channel 1.

To power down one of the two HCAN channels, the following procedure is required.

1. Send HCAN in halt mode (MCR.1 = '1');
2. Wait for the halt mode interrupt (IRR.0)
3. Clear all pending interrupt request;
4. Disable the requested channel by clearing the related bit in controlling Clock Control 1 (CC1) Register in Power Control module (clear the bit 18 and/or 19 to '0').

To wake up the module the following procedure is required:

1. Enable the requested channel by setting the related bit in controlling Clock Control 1 (CC1) Register in Power Control module (set the bit 18 and/or 19 to '1').
2. Modify HCAN configuration if necessary
3. Recover the halt mode of the HCAN by clearing MCR.1.
4. After detecting 11 recessive bit on the CAN Bus HCAN is able to join the communication. If used with TTCAN system a synchronization is necessary before start transmission.

19.4.8 Registers Index

Table 19.7 Register Index

Symbol	Register's Name	Brief description	Pag.
MCR	Master Control Register	General configurations for HCAN and test mode setting	818
GSR	General Status Register	Status register for HCAN	825
BCRi	Bit Configuration Registers	Timing configurations for Baud Rate setting	826
IRR	Interrupt Request Register	Interrupt Request status	831
IMR	Interrupt Mask Register	Mask for Interrupt Request	837
TXPR	Transmission Pending Register	Transmission request	842
TXCR	Transmission Cancel Register	Abort transmission request	845
TXACK	Transmission Acknowledge Register	Transmission successful Flag	847
ABACK	Abort Acknowledge Register	Transmission abort flag	849
RXPR	Received Data Frame Pending Register	Data Frame reception flag	850
RFPR	Remote Frame Request Pending	Remote Frame reception flag	852
MBIMR	Mailbox Interrupt Mask Register	Mask for Mailbox related interrupt	853
UMSR	Unread Message Status Register	Overwrite Message Flag	855
TCNTR	Timer Counter Register	Current Timer value	858
TCR	Timer Control Register	General Timer Configuration	859
TPSR	Status Register	Status flags for Timer	862
TMR	Timer Mode Register	Value to be used for TimeStamp and TCMRi registers	865
TDCR	Timer Drift Correction Register	Timer correction for synchronization within the network	866
LOSR	Local Offset Register	Offset for Timer	866
CCR	Cycle Counter Register	Current Cycle Counter Value for TT transmission	867
CMAX	Cycle Maximum Register	Number of Basic Cycles	867
ICRi	Input Capture Registers	Input capture value	868
TCMRi	Timer Compare Match Registers	Compare value for Timer	869

Section 20 Most Interface Module

20.1 General Description

The Most Interface Module (MIM) performs all the necessary interfacing functions required by the external OS8104 MOST transceiver chip. It organises the transfer of real-time, control and/or packet data between the transceiver and the processor or Register Bus DMA Controller.

Packet data is not constrained in length to that which can fit into a single MOST frame, but can instead be split up and sent over several frames. To minimise the burden on the host software, the MIM itself performs the repackaging of data necessary to send data over multiple frames.

20.1.1 Features

- Support for up to 4 streaming real-time channels (up to 4 for transmit, up to 2 for receive)
- Each streaming channel's bandwidth can be configured from zero to eight 32-bit words
- Variable frame rate up to 50 kHz, subject to OS8104 transceiver specification
- Standard 32-bit Register Bus DMA/processor interface to the host system
- Automatic repackaging for long data packets to minimise software overhead
- Data can be sent as high bandwidth packets, to transfer bursts of information
- Data can also be sent as low bandwidth control packets

20.1.2 Terminology

Quadlet	A group of eight bytes, i.e. 64 bits of information.
Longword	A group of four bytes, i.e. 32 bits of information.
MIM	MOST Interface Module
MOST	MOST transceiver OS8104

20.2 Architectural Overview

20.2.1 Block Diagram

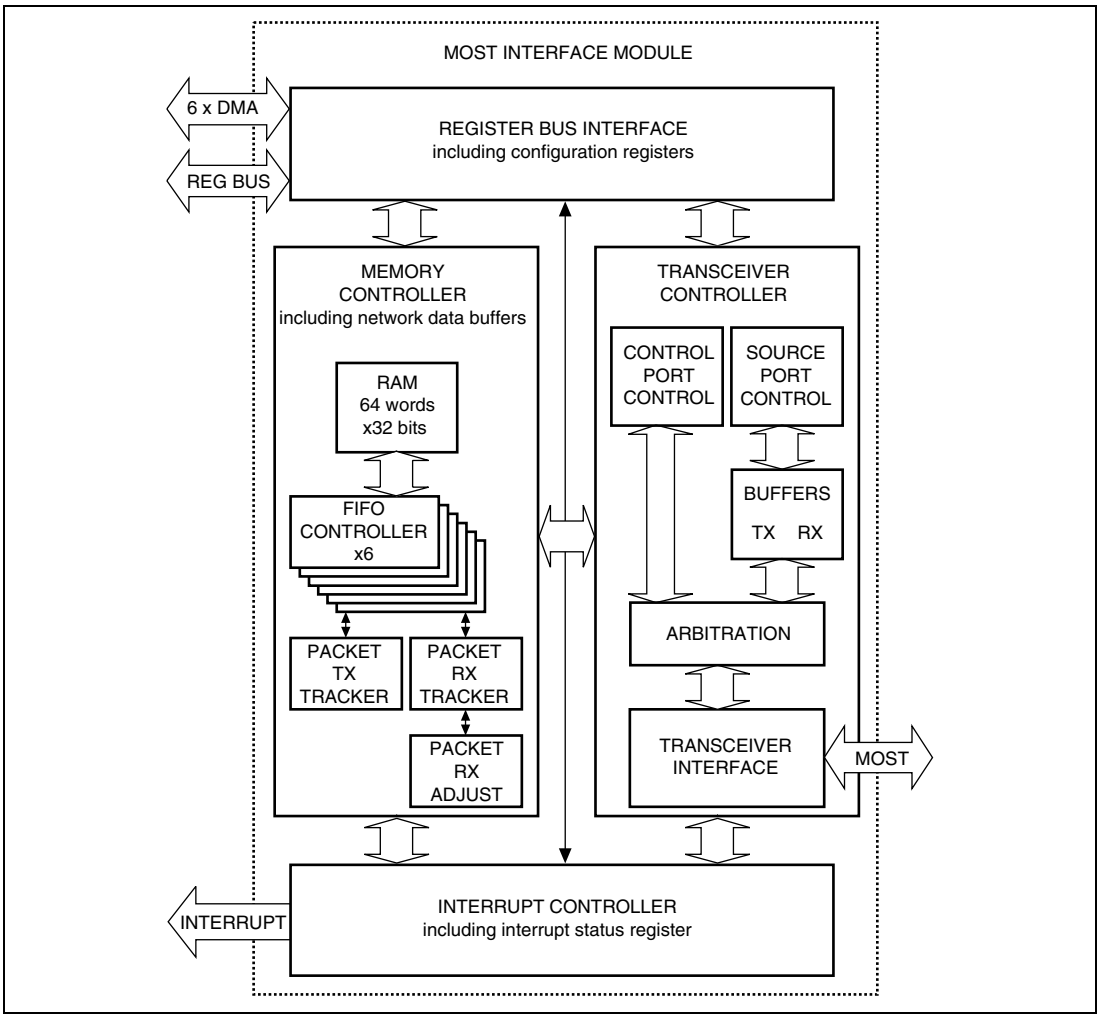


Figure 20.1 Block Diagram of the MOST Interface Module

20.3 Pin Descriptions

Table 20.1 MOST Interface Module Port Connections

Signal or Pin Name	Bits	I/O	Function	To/From	Clock
Register Bus	—	IO	System register bus interface	DMAC	rbclk
irq	1	O	Active high interrupt for errors, etc	Interrupt	rbclk
rbdmareqn	6	O	Active low DMA request signals	DMAC	rbclk
rbdmarackn	6	I	Active low DMA request acknowledge signals	DMAC	rbclk
rbdmaackn	6	I	Active low DMA acknowledge signals	DMAC	rbclk
MPAD(1:0)	2	O	Parallel address select	MOST	rbclk
MDATA(7:0)	8	IO	Data bus	MOST	rbclk
$\overline{\text{MRD}}$	1	O	Active low read control	MOST	rbclk
$\overline{\text{MWR}}$	1	O	Active low write control	MOST	rbclk
$\overline{\text{MAINT}}$	1	I	Active low asynchronous message interrupt	MOST	async
$\overline{\text{MINT}}$	1	I	Active low power on interrupt	MOST	async
$\overline{\text{MRESET}}$	1	O	Active low software reset for transceiver	MOST	rbclk
MERROR	1	I	Active high MOST transceiver error status	MOST	async
MFRAME_SYNC	1	I	Frame Sync I/O	MOST	async
MSRC_FLOW	1	I	Parallel Flow control	MOST	async
MCP_FROW	1	I	Control port flow control	MOST	async

20.4 Register Description

Note that all registers are 32-bit, and must be read from, or written to, as long words. Access to single bytes within each 32-bit register is not possible. MIM address locations not in the lists below are reserved and must not be accessed. Fields marked H'00 or similar are reserved. When writing to such fields, the bits must be set to 0. When reading, the values are not guaranteed.

20.4.1 Data Registers

With the exception of MIM Control Msg Register, all of the data registers can be programmed for automatic DMA transfer. Data written to or read from these registers is directly transmitted to or received from the MOST transceiver.

If the MOST Interface Module's MIM Module Config Register is programmed to support the correct endian scheme for the host system, then the ordering of data within these seven registers remains the same for both big-endian and little-endian systems.

Table 20.2 MOST Interface Module FIFO Buffer Registers

Address (Bytes)	Register Name	Access Size
H'6800	MIM Stream1	32
H'6804	MIM Stream2	32
H'6808	MIM Stream3	32
H'680C	MIM Stream4	32
H'6810	MIM Control Msg	32
H'6814	MIM PacketTx	32
H'6818	MIM PacketRx	32

20.4.2 Configuration Registers

The registers listed below are used to configure the MOST Interface Module or the MOST transceiver. The data order within these registers is not affected when the MOST Interface Module's MIM Module Config Register is programmed for different endian schemes.

Table 20.3 MOST Interface Module Register List

Address (Bytes)	Register Name	Access Size
H'6840	MIM Module Config	32
H'6844	MIM Buffer Ready	32
H'6848	MIM Interrupt Status	32
H'684C	MIM Interrupt Enable	32
H'6850	MIM Stream1 Config	32
H'6854	MIM Stream2 Config	32
H'6858	MIM Stream3 Config	32
H'685C	MIM Stream4 Config	32
H'6860	MIM Control Config	32
H'6864	MIM PacketTx Config	32
H'6868	MIM PacketRx Config	32
H'6870	MIM MOST Reg Wr	32
H'6874	MIM MOST Reg Rd	32
H'6878	MIM_Status	32

Legends for register description:

Initial Value : Register value after reset
— : Undefined value
R/W : Read and Write, write value can be read.
R : Read only, for write always 0 write
R/WC0 : Read and Write, 0 write clear, 1 write is ignored
R/WC1 : Read and Write, 1 write clear, 0 write is ignored
W : Write only, Read prohibited. If reserved, write always 0.
—/W : Write only, Read value undefined.

Note: Always write 0s to the bits reserved in the registers. If these bits are read, the value may not always be 0. It would be a good idea to mask the reserved bits when reading the register values in the software program.

20.5 Module Register Descriptions

20.5.1 MIM Stream1, MIM Stream2, MIM Stream3, MIM Stream4 Registers

Reset value: H'00000000

A set of 4 buffered registers via which real-time information is read or written. Each of these registers expects between 0 and 15 accesses per frame, depending on the set-up in the corresponding MIM StreamX Config Register. In the event that less than 4 bytes need to be used in any given word, then the leftmost bytes contain valid data, and the rightmost bytes of data must be ignored. The format is the same for big-endian and little-endian systems.

MIM Stream1, MIM Stream2 Registers

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SD0								SD1							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SD2								SD3							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	SD0	0	W	**SD0** The first of the 4 bytes in this word to be sent in the frame
23 to 16	SD1	0	W	**SD1** The second byte
15 to 8	SD2	0	W	**SD2** The third byte
7 to 0	SD3	0	W	**SD3** The last byte of the current word

MIM Stream3, MIM Stream4 Registers

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SD0								SD1							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SD2								SD3							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	SD0	0	R/W	**SD0** The first of the 4 bytes in this word to be sent or received in the frame
23 to 16	SD1	0	R/W	**SD1** The second byte
15 to 8	SD2	0	R/W	**SD2** The third byte
7 to 0	SD3	0	R/W	**SD3** The last byte of the current word

20.5.2 MIM_Stream1_Config, MIM_Stream2_Config, MIM_Stream3_Config, MIM_Stream4_Config Registers

Reset value: H'00000000

These 4 registers each configure their corresponding streaming channel. MIM Module Config Registers must also be configured to enable the system features required for a given application.

MIM Stream1 Config, MIM Stream2 Config,

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLR	DM														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		QA														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	CLR	0	R/WC1	**CLR** Setting this bit clears this channel's FIFO. The bit clears automatically
30	DMA	0	R/W	**DMA** Writing 1 to this bit disables this channel's DMA request ability
29 to 15	—	0	R	**Reserved**
14 to 0	QA	0	R/W	**QA** Quadlet allocation (bit 0 = quadlet 0, bit 14 = quadlet 14). Write '1' to allocate and '0' to de-allocate the Quadlets.

MIM Stream3 Config, MIM Stream4 Config

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLR	DM														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RX	QA														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	CLR	0	R/WC1	**CLR** Setting this bit clears this channel's FIFO. The bit clears automatically
30	DMA	0	R/W	**DMA** Writing 1 to this bit disables this channel's DMA request ability
29 to 16	—	0	R	**Reserved**
15	RX	0	R/W	**RX** Receiver. For streams 3 and 4 only, select Rx (1) or Tx (0)
14 to 0	QA	0	R/W	**QA** Quadlet allocation (bit 0 = quadlet 0, bit 14 = quadlet 14)

20.5.3 MIM_PacketTx (W)

Reset value: H'00000000

The format of this register is the same for big-endian and little-endian systems. Outgoing packets of length L 32-bit words are written to this buffered register in this sequence:

Longword:	1	2	3	4	5	6	7	8
	PH	PW1	PW2	PW3	—	—	—	PWL

Word 1	PH	Packet Header (format described above) for outgoing packets
Word 2	PW1	The 1st data word (format described above)
Words 3+	PW2.L	The nth Packet Data Word (format described above), from 2 to L

20.5.4 MIM_PacketRx (R)

Reset value: H'00000000

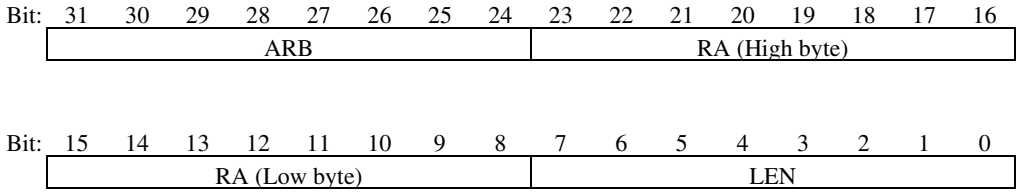
The format of this register is the same for big-endian and little-endian systems. Incoming packets of length L 32-bit words are read from this register in the sequence below:

Longword:	1	2	3	4	5	6	7	8
	PH	PW1	PW2	PW3	—	—	—	PWL

Word 1	PH	Packet Header (format described below) for incoming packets
Word 2	PW1	The 1st data word (format described below)
Words 3+	PW2.L	The nth Packet Data Word (format described below), from 2 to L

Packet Header Word Format (PH)

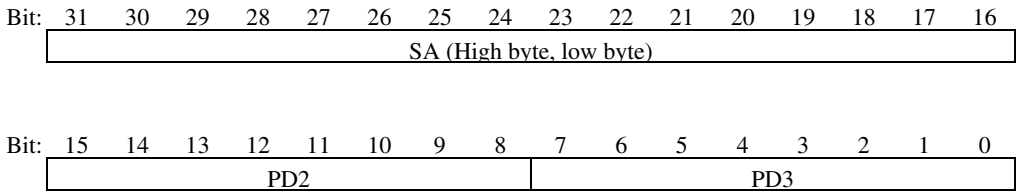
Note that the Asynchronous Packet Header format is the same as in the MOST specification. For packet transmission, the software must correctly generate the header, this task is not performed by the MIM.



Bit	Bit Name	Description
31 to 24	ARB	**ARB** Arbitration, equal to {(sender's bNPR node position *2) + 1}
23 to 16	RA (High byte)	**RA**
15 to 8	RA (Low byte)	Remote address where the packet is sent to (TX) or comes from (RX)
7 to 0	LEN	**LEN** Length, excluding header, in 32-bit words, from 0x01 to H'FE

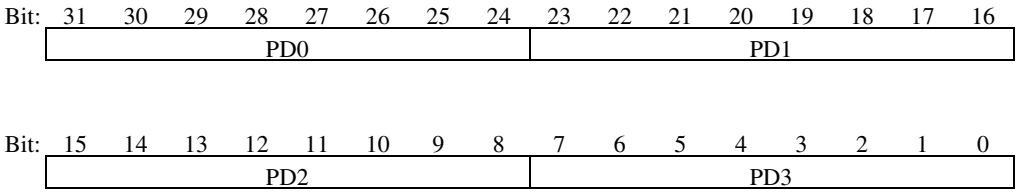
Packet Data Word format- First word (PW1)

Note the first two data bytes of word 1 of the message must be the source address of the local MOST node.



Bit	Bit Name	Description
31 to 16	SA (High byte, low byte)	**SA** Source address (sent first)
15 to 8	PD2	**PD2** 3rd data byte in this word
7 to 0	PD3	**PD3** 4th data byte in this word (sent last)

Packet Data Word format – Subsequent words (PW2.PWL)

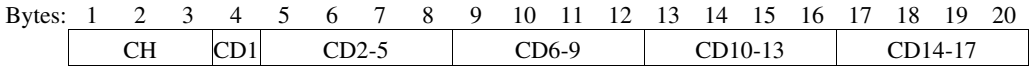


Bit	Bit Name	Description
31 to 24	PD0	**PD0** 1st data byte in this word (sent first)
23 to 16	PD1	**PD1** 2nd data byte in this word
15 to 8	PD2	**PD2** 3rd data byte in this word
7 to 0	PD3	**PD3** 4th data byte in this word (sent last)

20.5.5 MIN Control Msg (RW)

Reset value: H'00000000

A control message sequence must comprise exactly five 32-bit words, including the header. It should be written or read, 32 bits at a time, via MIM_Control_Msg in the sequence shown below. There is sufficient buffering to allow access to the whole message without pauses.



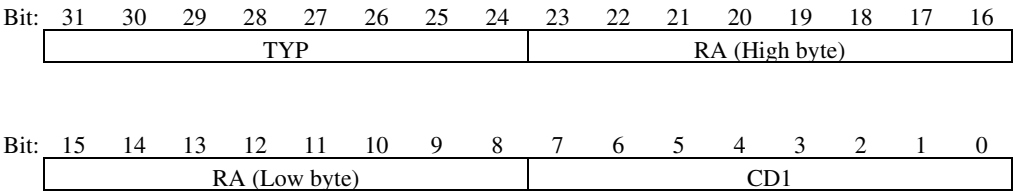
Bytes 1 to 3 CH Transmit Control Message Header (format described below)

Bytes 4 to 20 CDn The nth Control Message Data Byte, from 1 to 17.

Description of the Control Message Header Format for RCH and TCH (CH, CD1)

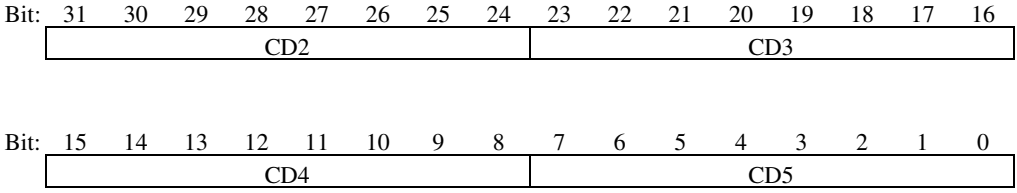
The format used for control messages is the same as that described in the MOST specification, except that

the PRIORITY byte of the message is specified in MIM Control Config Register below. This allows the rest of the control message to be specified efficiently, as five 32-bit words, and does in fact mirror the format of mRCMB, the transceiver's Receive Control Message Buffer. For more information on each field, see the MOST transceiver specification.



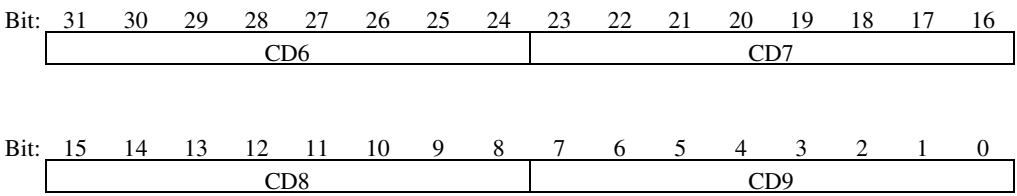
Bit	Bit Name	Description
31 to 24	TYP	**TYP** Control message Type
23 to 16	RA (High byte)	**RA**
15 to 8	RA (Low byte)	Remote address where the packet is sent to (TX) or comes from (RX)
7 to 0	CD1	**CD1** Control Message Data byte 1 (not part of header, shown for clarity)

Control Message Data Format – CD2 to 5



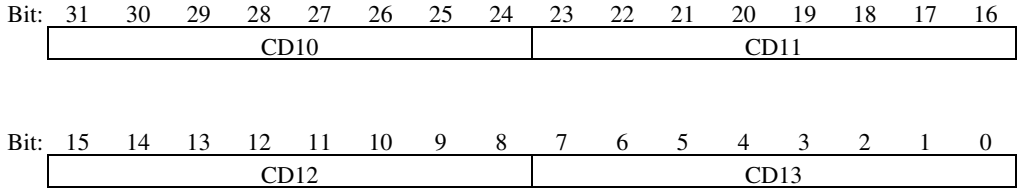
Bit	Bit Name	Description
31 to 24	CD2	**CD2** Control Message Data byte 2
23 to 16	CD3	**CD3** Control Message Data byte 3
15 to 8	CD4	**CD4** Control Message Data byte 4
7 to 0	CD5	**CD5** Control Message Data byte 5

Control Message Data Format – CD6 to 9



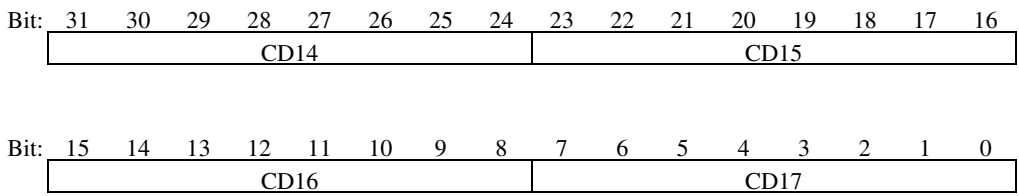
Bit	Bit Name	Description
31 to 24	CD6	**CD6** Control Message Data byte 6
23 to 16	CD7	**CD7** Control Message Data byte 7
15 to 8	CD8	**CD8** Control Message Data byte 8
7 to 0	CD9	**CD9** Control Message Data byte 9

Control Message Data Format – CD10 to 13



Bit	Bit Name	Description
31 to 24	CD10	**CD10** Control Message Data byte 10
23 to 16	CD11	**CD11** Control Message Data byte 11
15 to 8	CD12	**CD12** Control Message Data byte 12
7 to 0	CD13	**CD13** Control Message Data byte 13

Control Message Data Format – CD14 to 17



Bit	Bit Name	Description
31 to 24	CD14	**CD14** Control Message Data byte 14
23 to 16	CD15	**CD15** Control Message Data byte 15
15 to 8	CD16	**CD16** Control Message Data byte 16
7 to 0	CD17	**CD17** Control Message Data byte 17

20.5.6 MIM Control Config Register

Reset Value: H'00000000

This register is used to control and monitor the control message buffer. See the section 20.7.3, Control Messages, on control messages, for more details.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLR												PRI			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERR								EMT	RM	CM	CMT			LRT	LMB
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/ WC0	R/W	R/ WC1	R/ WC1	R	R	R/W	R

Bit	Bit Name	Initial Value	R/W	Description
31	CLR	0	R/WC1	**CLR** Setting this bit clears this channel's FIFO. The bit clears automatically
30 to 20	—	0	R	**Reserved**
19 to 16	PRI	0	R/W	**PRI** Priority of the control message being transmitted from register bus
15 to 8	ERR	0	R/WC0	**ERR** Error details, valid if EMT is set (reports Transceiver bXTS Register)
7	EMT	0	R/WC0	**EMT** If set an error has been reported by MOST transmit, see ERR field
6	RMR	0	R/W	**RMR** Register Bus Message Request, 1 = start (write) or still sending (read)
5	CMR	0	R/WC1	**CMR** Completed (1) transfer from MOST Receive Ctrl buffer, write 1 to clear

Bit	Bit Name	Initial Value	R/W	Description
4	CMT	0	R/WC1	**CMT** Completed (1) transfer from MOST Transmit Ctrl buffer, write 1 to clear
3, 2	—	0	R	**Reserved**
1	LRT	0	R/W	**LRT** If set, request (w) or confirm ® lock by Register Bus Transmit process
0	LMB	0	R	**LMB** Read only, if set it is locked by Most Transmit or Receive buffer

20.5.7 MIM Interrupt Status Register

Reset value: H'00600000

This register report which events have been active by returning '1' for the event's bit. An interrupt occurs if the corresponding bit in MIM Interrupt Enable Register is set. Write '0' to clear a bit. Please reter section ″20.12 Interrupt sources″ for more information about interrupts.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XSE	XLR	XLE	RPU		TCE	TPL	FE	CE	XW	XRR	FPR	FPT	FCM	FS4	FS3
Initial:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W	R/	R/	R/	R/	R	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/
	WC0	WC0	WC0	WC0		WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FS2	FS1	TPT	RPM	TCT	RC	TPM	RPT	TCM	RCT	XIC	CSB	CGA	CLA	CPA	ALC
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/	R/
	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0	WC0

Bit	Bit Name	Initial Value	R/W	Description
31	XSE	0	R/WC0	**XSE** Transceiver system error, or initialised after power-up
30	XLR	0	R/WC0	**XLR** Transceiver lock error removed and stable for 128 frame periods
29	XLE	0	R/WC0	**XLE** Transceiver lock error
28	RPU	0	R/WC0	**RPU** Receive Packet Unpacking error – if set, MIM detected error
27	—	0	R	**Reserved**
26	TCE	0	R/WC0	**TCE** Transmit Control message Error discovered during transmission
25	TPL	0	R/WC0	**TPL** Transmit Packet Length mismatch discovered during transmission
24	FE	0	R/WC0	**FE** FIFO Error – write to a full FIFO, or read from an empty FIFO. This bit is set if an underflow or overflow error occurs during read/write to the FIFOs in MIM. An error for any FIFO will result in FE bit being set.
23	CE	0	R/WC0	**CE** Conflict – process violated for Ctrl Lock, MOST Reg Wr Register or Packet Tx Register
22	XWR	1	R/WC0	**XWR** Transceiver Write Ready, see Accessing transceiver registers
21	XRR	1	R/WC0	**XRR** Transceiver Read Ready, see Accessing transceiver registers
20	FPR	0	R/WC0	**FPR** FIFO Ready for MIM PacketRx Register (data available)

Bit	Bit Name	Initial Value	R/W	Description
19	FPT	0	R/WC0	**FPT** FIFO Ready for MIM PacketTx Register (if space available, enabled)
18	FCM	0	R/WC0	**FCM** FIFO Ready for MIM Control Msg Register buffer within the MIM
17	FS4	0	R/WC0	**FS4** FIFO Ready for MIM Stream4 Register (if space available, and enabled)
16	FS3	0	R/WC0	**FS3** FIFO Ready for MIM Stream3 Register (if space available, and enabled)
15	FS2	0	R/WC0	**FS2** FIFO Ready for MIM Stream2 Register (if space available, and enabled)
14	FS1	0	R/WC0	**FS1** FIFO Ready for MIM Stream1 Register (if space available, and enabled)
13	TPT	0	R/WC0	**TPT** Packet data transfer from MIM to transceiver complete
12	RPM	0	R/WC0	**RPM** Received packet(s) in memory (mim PacketRx Config's RPC field > 0)
11	TCT	0	R/WC0	**TCT** Control message transfer from MIM to transceiver complete
10	RCM	0	R/WC0	**RCM** Control message reception (transfer from transceiver to MIM) complete
9	TPM	0	R/WC0	**TPM** Packet data transmission by MOST complete
8	RPT	0	R/WC0	**RPT** Packet data received by MOST transceiver

Bit	Bit Name	Initial Value	R/W	Description
7	TCM	0	R/WC0	**TCM** Control message transmission by MOST complete. See TCE for errors
6	RCT	0	R/WC0	**RCT** Control message received by MOST transceiver
5	XIC	0	R/WC0	**XIC** Transceiver Initialisation Complete
4	CSB	0	R/WC0	**CSB** Changed Synchronous Bandwidth – change detected in the SBC register
3	CGA	0	R/WC0	**CGA** Changed Group Address – change detected in MOST's bGA register
2	CLA	0	R/WC0	**CLA** Changed Logical Address –bNAH/bNAL or bAPAH/bAPAL changed
1	CPA	0	R/WC0	**CPA** Changed Position Address – change detected in MOST's bNPR register
0	ALC	0	R/WC0	**ALC** Network configuration changed, e.g. total number of nodes or delays

20.5.8 MIM Interrupt Enable Register

Reset value: H'00000000

Setting a bit to '1' enables interrupts from the corresponding source, and '0' masks the interrupts. See MIM Interrupt Status Register for descriptions of each bit field.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	XSE	XLR	XLE	RPU		TCE	TPL	FE	CE	XW	XRR	FPR	FPT	FCM	FS4	FS3
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FS2	FS1	TPT	RPM	TCT	RC	TPM	RPT	TCM	RCT	XIC	CSB	CGA	CLA	CPA	ALC
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	XSE	0	R/W	See MIM Interrupt Status Register for description
30	XLR	0	R/W	
29	XLE	0	R/W	
28	RPU	0	R/W	
27	—	0	R	Reserved
26	TCE	0	R/W	See MIM Interrupt Status Register for description
25	TPL	0	R/W	
24	FE	0	R/W	
23	CE	0	R/W	
22	XWR	0	R/W	
21	XRR	0	R/W	
20	FPR	0	R/W	
19	FPT	0	R/W	
18	FCM	0	R/W	
17	FS4	0	R/W	
16	FS3	0	R/W	
15	FS2	0	R/W	
14	FS1	0	R/W	
13	TPT	0	R/W	
12	RPM	0	R/W	
11	TCT	0	R/W	
10	RCM	0	R/W	
9	TPM	0	R/W	
8	RPT	0	R/W	
7	TCM	0	R/W	
6	RCT	0	R/W	
5	XIC	0	R/W	
4	CSB	0	R/W	See MIM Interrupt Status Register for description
3	CGA	0	R/W	
2	CLA	0	R/W	See MIM Interrupt Status Register for description
1	CPA	0	R/W	
0	ALC	0	R/W	

20.5.9 MIM Buffer Ready Register

Reset value: H'00000000

This register returns the status of each of the internal FIFO buffers. It is useful in system configurations when the MIM Registers are not to be accessed via DMA. In the case of a receive buffer, it indicates that one longword of data is available. In the case of a transmit buffer, it indicates that one longword of space is available. Note that if the corresponding feature is not enabled in MIM Module Config Register then the buffer describes itself as "not ready".

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RCT	RCR	RMT	RPT	RPR	RS4	RS3	RS2	RS1
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	0	R	**Reserved**
8	RCT	0	R	**RCT** FIFO Buffer Ready for Control Message Read from MOST Transmit
7	RCR	0	R	**RCR** FIFO Buffer Ready for Control Message Read from MOST Receive
6	RMT	0	R	**RMT** FIFO Buffer Ready for Message Transmit through control port
5	RPT	0	R	**RPT** FIFO Buffer Ready for Packet Tx
4	RPR	0	R	**RPR** FIFO Buffer Ready for Packet Rx
3	RS4	0	R	**RS4** FIFO Buffer Ready for Stream 4
2	RS3	0	R	**RS3** FIFO Buffer Ready for Stream 3
1	RS2	0	R	**RS2** FIFO Buffer Ready for Stream 2
0	RS1	0	R	**RS1** FIFO Buffer Ready for Stream 1

20.5.10 MIM PacketRx Config Register

Reset value: H'000000FF

This register configures and monitors the reception of data packets. The RPR counter must be updated by software as packets are read, because this in turn updates RPC, which is used by the MIN Interrupt Status Register to determine when packets are available for reading.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLR	DMA							RPR							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W C1	R/W	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RPC								RPL							
Initial:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	CLR	0	R/WC1	**CLR** Setting this bit clears this channel's FIFO. The bit clears automatically
30	DMA	0	R/W	**DMA** Writing 1 to this bit disables the Packet Rx DMA request ability
29 to 24	—	0	R	**Reserved**
23 to 16	RPR	0	W	**RPR** Received Packet Read (write only, reading returns H'00)—used to inform the MIM how many have been dealt with
15 to 8	RPC	0	R	**RPC** Received Packet Count (read only, writes are ignored)—returns a count of how many packets still need to be read
7 to 0	RPL	H'FF	R	**RPL** Received Packet Length remaining—Indication of how many words of an incoming packet need to be transferred from the MIM to the DMAC after the current transfer. Value of H'FF indicates no transfers are outstanding

20.5.11 MIM PacketTx Config Register

Reset value: H'000000FF

This register configures and monitors the transmission of data packets. Setting the TPR bit indicates to the MIM that a packet is in memory, ready for transmission. Reading it returns the status in the MIM. If high, the packet is still spooling into the transceiver.

Note that as a precaution, if the length of packet described in this register (TPL) does not match the length of packet described in the header, transferred from memory, then an interrupt is generated, the FIFO is cleared and packet transfer is aborted. The interrupt must be cleared and, if DMA is being used to transfer packet data to the MIM, the DMA controller needs to be reprogrammed before subsequent packets can be sent.

This register must be written to just once per packet transmission.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CLR	DMA														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R
	C1															

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								TPR	TPL							
Initial:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	CLR	0	R/WC1	**CLR** Setting this bit clears this channel's FIFO. The bit clears automatically
30	DMA	0	R/W	**DMA** Writing 1 to this bit disables the Packet Tx DMA request ability
29 to 9	—	0	R	Reserved
8	TPR	0	R/W	**TPR** Transmit Packet Request – High means start (W) or still (R) sending. If '1' is written to this bit, then the length specified in TPL must be correct. No further writes to this register can take place until the packet transmission has been finished, signified by TPR going low.
7 to 0	TPL	1	R/W	**TPL** Transmit Packet Length, in quadlets, excluding the header. Write the correct length when initiating a transfer. Read the number of words which still need to be transferred to the MOST after the next transfer. Value of H'FF indicates no transfers are outstanding and the whole packet has been sent to the MOST

20.5.12 MIM Module Config Register

Reset value: H'00000014

This register is used to configure the system architecture of the MOST Interface Module.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
							DCP	END	DPC	DPS	DPA	RCT	RCR	ESP	RES	ECT
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECR	EPR	EPT	EMT	ES4	ES3	ES2	ES1	ME	POE	CLK					
Initial:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	0	R	**Reserved**
25	DCP	0	R/W	**DCP** Disable Control Port usage (If set, MIM only uses Source Port)
24	END	0	R/W	**END** Endian format. If set, Data, that is received from or transferred to the peripheral bus, is swapped around. Otherwise Data is stored as it is. This swapping only affects the MIM_StreamX, MIM_PacketRx, MIM_PacketTx and MIM_Control_Msg registers. For example, when data is 0x01020304 and END bit is set, the data stored in the register is 0x04030201.
23	DPC	0	R/W	**DPC** Disable Polling of Control Message registers in MOST. If set, the MIM does not periodically poll registers such as bMSGS in the transceiver.
22	DPS	0	R/W	**DPS** Disable Polling of System registers in MOST. If set, the MIM does not periodically poll registers such as bSBC in the MOST transceiver.
21	DPA	0	R/W	**DPA** Disable Polling of Addressing registers in MOST. If set, the MIM does not periodically poll registers such as bNAH, bNAL in the transceiver.
20	RCT	0	R/W	**RCT** Reset MTX in MOST MSGS register after auto-reading control message reply (auto-read only occurs when ECT bit in this register is set)
19	RCR	0	R/W	**RCR** Release Control Message Rx Buffer in MOST after auto-reading message (auto-read only occurs when ECR bit in this register is set)
18	ESP	0	R/W	**ESP** Enable Source Port Read accesses. When this bit is clear, the MIM can only transmit streaming or packet data. When set, the MIM can also receive such data. Control Port operation is unaffected.

Bit	Bit Name	Initial Value	R/W	Description
17	RES	0	R/W	**RES** RESET transceiver (1 = Activate RESET). The MOST transceiver is reset whenever the MIM is reset. The MOST is automatically released from hardware resets, but software resets must write '0' to RES to end a reset.
16	ECT	0	R/W	**ECT** Enable Control Message auto-read from MOST Transmit Buffer feature
15	ECR	0	R/W	**ECR** Enable Control Message auto-read from MOST Receive Buffer feature
14	EPR	0	R/W	**EPR** Enable Packet Rx feature
13	EPT	0	R/W	**EPT** Enable Packet Tx feature
12	EMT	0	R/W	**EMT** Enable Control Message auto-write to MOST Transmit Buffer feature
11	ES4	0	R/W	**ES4** Enable Stream 4 feature
10	ES3	0	R/W	**ES3** Enable Stream 3 feature
9	ES2	0	R/W	**ES2** Enable Stream 2 feature
8	ES1	0	R/W	**ES1** Enable Stream 1 feature
7	ME	0	R/W	**ME** MIM Enable—if low, MIM does not communicate with the transceiver
6	POE	0	R/W	**POE** Pin Output Enable – if low, external MOST pins should be tristated

Bit	Bit Name	Initial Value	R/W	Description
5	CLK	0	R/W	**CLK**
4	CLK	1	R/W	Rbclk period, in nanoseconds, rounded down to the nearest nanosecond. Periods of between 17 ns and 34 ns (29MHz and 59MHz) are supported. If a value of H'3F is written to this field, then worst case clock frequency is assumed by the MIM for Rbclk.
3	CLK	0	R/W	
2	CLK	1	R/W	
1	CLK	0	R/W	
0	CLK	0	R/W	

20.5.13 MIM MOST Reg Wr Register

Reset value: H'00040000

This memory location is used for writing data into any of the transceiver's memory locations.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													XAI	XWR	XA	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XA								XD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	0	R	**Reserved**
19	XAI	0	R/W	**XAI** Transceiver address increment, '1' means ignore field XA and instead automatically use the value of XA from the previous MIM MOST Reg Wr plus one
18	XWR	1	R	**XWR** Transceiver write ready, '1' means more data can be sent, This operation is explained later in this document.
17 to 8	XA	0	R/W	**XA** Transceiver address location to be written to. Bits 17, 16 contain the page number (normally 0) and bits 15 to 8 contain the address location within that page
7 to 0	XD	0	R/W	**XD** Transceiver data to be stored in location XAL

20.5.14 MIM MOST Reg Rd Register

Reset value: H'00040000

This is used for reading the MOST transceiver's registers. The 11-bit source register location, is written into XA. Next, XRR is polled – or an interrupt is awaited – until the "Transceiver read ready" bit is set, whereupon the data can be used. This is discussed in detail later in this document.

Note that this register must be used for reads from all transceiver registers, with no exceptions. The MIM does have the option to poll certain transceiver registers (bGA, bMSGS, bSBC, bNPR, bNAH/L and bAPAH/L) but the results of this polling is private to the MIM and the results are not accessible to the software. Reading from any of these register locations results in the MIM refreshing its local values from the MOST transceiver.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
													XAI	XRR	XA	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	XA								XD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 20	—	0	R	**Reserved**
19	XAI	0	R/W	**XAI** Transceiver address increment, '1' means ignore field XA and instead automatically use the value of XA from the previous MIM MOST Reg Rd Register plus one
18	XRR	0	R	**XRR** Transceiver read ready, '1' means the data in XD is valid for address XA
17 to 8	XA	0	R/W	**XA** Transceiver address location to be read from. Bits 17, 16 contain the page number (normally 0) and bits 15 to 8 contain the address location within that page
7 to 0	XD	0	R	**XD** Transceiver data from location XAL (only valid when XRR bit is set). This field is read only, any data written to these bits is ignored.

20.5.15 MIM_Status

Reset value : H'00400080

Safest value for worst case conditions : H'00D00080

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
									CPA	CPW	CPE	CP1		FER	FET	FEC
Initial:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FE4	FE3	FE2	FE1					XLRC[7:0]							
Initial:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	0	R	**Reserved**
23	CPA	0	R/W	When set to 1, always wait for 325 ns after falling edge of CP_flow. This period is taken from the MOST Transceiver data sheet.
22	CPW	1	R/W	When set to 1, wait for 325 ns after falling edge of CP_flow if MOST network is out of lock. This period is taken from the MOST Transceiver data sheet.
21	CPE	0	R/W	When set to 1, one Control Port access takes place per sf-interval if ERROR pin is 1 and MOST network is unlocked.
20	CP1	0	R/W	When set to 1, one Control Port access always takes place per sf-interval
19	—	0	R	**Reserved**
18	FER	0	R/W	When 1 and IRQ_FE is set, Packet RX FIFO error. Write '0' to clear. Clear Packet RX FIFO first to correct the error condition.
17	FET	0	R/W	When 1 and IRQ_FE is set, Packet TX FIFO error. Write 0 to clear. Clear Packet TX FIFO first to correct the error condition.
16	FEC	0	R/W	When 1 and IRQ_FE is set, Control Message FIFO error. Write 0 to clear. Clear Control Message FIFO first to correct the error condition.

Bit	Bit Name	Initial Value	R/W	Description
15	FE4	0	R/W	When 1 and IRQ_FE is set, Stream4 FIFO error. Write 0 to clear. Clear Stream4 FIFO first to correct the error condition.
14	FE3	0	R/W	When 1 and IRQ_FE is set, Stream3 FIFO error. Write 0 to clear. Clear Stream3 FIFO first to correct the error condition.
13	FE2	0	R/W	When 1 and IRQ_FE is set, Stream2 FIFO error. Write '0' to clear. Clear Stream2 FIFO first to correct the error condition.
12	FE1	0	R/W	When 1 and IRQ_FE is set, Stream1 FIFO error. Write 0 to clear. Clear Stream1 FIFO first to correct the error condition.
11 to 8	—	All 0	R	**Reserved**
7	XLRC7	1	R/W	This is an 8-bit count. When Transceiver lock is lost and then the transceiver lock is regained, this count determines the number of frames the Transceiver should stay stable and locked before the XLR bit is set in the MIM_Interrupt_status register. Default value is H'80.
6	XLRC6	0	R/W	
5	XLRC5	0	R/W	
4	XLRC4	0	R/W	
3	XLRC3	0	R/W	
2	XLRC2	0	R/W	
1	XLRC1	0	R/W	
0	XLRC0	0	R/W	

20.6 Functional overview

20.6.1 General Functionality

The MOST Interface Module (MIM) controls the transfer of data between the host system, including software and peripherals, and the MOST OS 8104 Transceiver chip.

Packet data can be used either to send or receive packets of information to other nodes on the network. Although packet data can be up to 254 longwords in length, it is only possible for between 1 and 9 longwords of data to be transmitted or received in each MOST frame. The MIM partitions the data between subsequent frames, so that the host processor just needs to write or read the entire packet, to or from memory. An interrupt can inform the processor when transmission or reception has occurred.

Streaming data can also be passed to and from the network via the MOST-MIM interface, to convey real-time data such as audio channels.

The transfer of streaming and packet data can be arranged via the DMA controller, and a unique DMA channel can be allocated for each of these six possible data channels. Data can then be transferred automatically from a source such as S/PDIF to the MIM.

The final data passing mechanism is for control messages. These can be used to send low-bandwidth fixed-size packets, of 17 bytes, or to remotely configure or interrogate any other node on the network. Only 2 bytes per frame are allocated to this mechanism. As the data rate is so low, and the intended use is purely for remote configuration of dumb nodes after power-up, it is not possible to allocate a DMA channel. Sufficient buffering has been incorporated to allow an entire control message to be stored or retrieved with just five successive processor accesses.

20.7 Data Handling Methods

20.7.1 Streaming real-time data

For real-time information, the DMA Controller must service the incoming and outgoing streaming MIM buffers directly, on a regular basis. By directly transferring data from a source module, such as S/PDIF, to the MIM, the software overhead is kept to a minimum.

There are four streaming channels available. Any of these may be configured for transmission of real-time data, and up to two may be configured for reception of real-time data. Each channel can be programmed to expect up to 32 bytes of data in each frame, in four-byte increments. There is no real concept of "messages" with streaming data, a stream should be viewed as a continuous data flow, which can be "listened to" or "written to" as required.

Streaming data can be transferred either by setting up a DMA channel, or by manually transferring longwords of data to / from the MIM StreamX FIFO Buffer Registers. If the manual method is chosen, then prior to each MIM StreamX FIFO Register access, the corresponding FS1, FS2, FS3 or FS4 bit of MIM Buffer Ready Register should be checked to ensure that there is space available for a write, or data available for a read. Data must be transferred quickly enough to ensure that the small local FIFOs in the MIM do not become completely full or empty, otherwise receive data frames may be overwritten, or transmit data frames may be repeated.

Should less longwords be required for a stream than are programmed – for instance, near the end of a stream configured for 5 longwords per frame, only 3 longwords remain – then it is important to transfer the full 5 longwords per frame, subject to MIM Buffer Ready Register indicating space available. In this way, the MIM FIFO Buffer does not think the software has simply failed to respond quickly enough and starts to repeat data to allow the software to catch up. Unused bytes should be padded out with H'00. In summary, if a stream is configured for Y words per frame, then the total data transferred by the software should be a multiple of Y.

It is not necessary to transfer an entire frame's worth of data for each stream at once. The only requirement is that the average throughput of the data transfers is sufficient for the bandwidth

required. The MIM Buffer Ready Register signal indicates that just one longword of data can be transferred so, if DMA is not used, a repeating cycle of poll-transfer-poll-transfer is required.

See the section "Configuration of the MOST transceiver" for more details on how to set up streaming channels.

20.7.2 High Bandwidth packet data

The format of asynchronous packet data is identical to that described in the OS8104 MOST transceiver specification. For transmitting packet information via DMA, the software must:-

(a) Ensure packet transmission was enabled at powerup (set EPT in MIM Module Config Register)
(b) Write to MIM PacketTx Config Register, setting the DMA and CLR bits only to purge any previous, unwanted packet transmissions from the system.
(c) Write the header – priority, destination, and length L - into the first 32-bit memory location.
(d) Write the entire packet into the subsequent L 32-bit memory locations (eg in DMA FIFO)
(e) Initialise a transmission by writing to fields TPR and TPL in MIM PacketTx Config Register. Ensure that the CLR bit is clear, and if DMA transfer is required, the DMA bit must also be clear. These fields must all be written to simultaneously using just one write, and the TPL field must contain the length L of the packet excluding the header. No further writes may take place until the MIM reports clears the TPR bit to signal that packet transmission has finished, either naturally or through an error condition. If the TPL field does not match the length stored in the packet header, an error will be flagged via the TPL bit in the MIM Interrupt Status Register and packet transmission will be aborted.
(f) Await a TPM interrupt or poll the TPR bit in MIM PacketTx Config Register signifying successful transmission, before sending another packet or writing to MIM PacketTx Config Register again.

If DMA transfer is not used, the packet transmission must be initialised by first configuring MIM PacketTx Config Register, and then the packet itself must be stored in MIM PacketTx Register by writing first the header, then the L words of packet data. Before each write to MIM PacketTx Register, the RPT field of MIM Buffer Ready Register should be checked to ensure that space is available.

Note that if the length described in the packet itself does not match the length expected, as programmed in the MIM PacketTx Config Register, then the TPL bit is set in the MIM Interrupt Status Register, and packet transmission is aborted. To re-start packet transmission it is necessary to clear the interrupt, reprogram the MIM PacketTx Config Register and reprogram the DMA controller.

For receiving packet information, the software must use the DMA controller. Packets will automatically be queued up, one immediately after the other, in DMA memory as they arrive.

DMA must be used to receive packets because it is necessary to remove all data from the MIM PacketRx Register buffer within precisely one audio frame period of receipt. Interrupt latencies and the bandwidth of packet data (up to 9 longwords per audio frame) are likely to make this task extremely difficult to perform without DMA. Should the buffer not be emptied quickly enough, then data will be lost.

The whole procedure is:

(a) Ensure packet reception is enabled (set EPR in MIM Module Config Register)
(b) Await a RPM interrupt or poll MIM PacketRx Config Register, to see if an entire packet has arrived
(c) Check the RPC field to see how many packets are in memory
(d) Read the packet header from the first 32-bit location and the entire packet from the subsequent L 32-bit memory locations. Repeat steps for other outstanding packets
(e) Update the RPR field in MIM PacketRx Config Register so that the MIM can continue to track how many packets are outstanding, and notify the software accordingly.

In the event of a packet reception error at any point in the packet transfer process, spooling of the packet to memory halts, and no further packets are received until the error, logged in the interrupt status register, has been acknowledged/cleared down by the software. Clearing the interrupts in MIM_Interrupt_Status register does not affect the registers of the MOST Transceiver packet data registers. The software program will have to write the correct values into the registers as and when required.

20.7.3 Control Messages

There are two techniques for passing control messages between the MIM and the MOST transceiver. The process can be handled manually by means of the MIM MOST Reg Wr Register and MIM MOST Reg Rd Registers, reading from / writing to the MOST transceiver registers in accordance with the MOST specification. Alternatively, the MIM offers a facility to speed up the process, by automating the transfer and corresponding register programming associated with control messages.

To automatically send a control message, use of the control port by the MIM must be enabled. The software needs to pass five 32-bit words to the MIM. Sufficient buffering is maintained in the MIM to allow these words to be written one after the other, without interruption. However, permission to use the buffer must first be granted via the MIM Control Config Register on section 20.5.6, MIM Control Config Register. This is necessary because of the limited space within the MIM for buffering control messages, necessitating the sharing of a common buffer.

The format of control messages is fully described within the Transceiver specification. The only difference for the MIM is that the priority must not be stored in the message buffer, but must instead be written to MIM Control Config Register when the message transfer is initiated.

If less than 5 words of data are required for the control message, it is still necessary for exactly 5 words to be written to, or read from, the control message buffer. In the case of a write, unused bytes should be set to H'00. In the case of a read, unused bytes should be ignored.

20.7.4 Automatically Sending Control Messages to the MOST transceiver

(a) Ensure that the facility to automatically send control messages is enabled by setting EMT, and clearing DCP, in MIM Module Config Register

(b) Request use of the message buffer by writing '1' to the LRT bit in MIM Control Config Register

(c) Read back MIM Control Config Register and check LRT to see if permission has been granted. If not, then the buffer is being used to store an incoming message – follow the "Automatically Reading Control Messages from the MOST transceiver" procedure below, then return to (a). (The software will have to write '1' to the LRT bit again in the MIM_Control_Config register in step (b), if in step (c) LRT is not '1'. This must be done till permission is obtained.)

(d) If permission has been granted, store the message in MIM Control Msg Register. Exactly five longword writes must occur.

(e) Perform a single write to MIM Control Config Register of '1' to the RMR (Register Bus Message Request) bit, with the correct priority in the PRI field, to initiate transfer of the message.

(f) Once the message has been transferred to the transceiver, the MIM clears the RMR bit automatically, and releases the control message buffer lock (LRT = '0'). Optionally, an interrupt can be generated. Interrupt bit TCT reports when the MIM has successfully transferred the control message to the MOST transceiver.

(g) Subsequently, if polling of the MOST MSGS register is enabled (when the MIM_Module_Config register's DPC bit is clear), interrupt bit TCM reports when the destination node has physically received the message. (The interrupt bits set conditions have been explained later in this document.)

20.7.5 Automatically Reading Control Messages from the MOST Transceiver

This procedure is used to read responses to a previous transmission in the Tx Control Buffer, or new messages in the Rx Control Buffer.

There are two modes in addition to the procedure outlined below. It is possible to configure the MIM either to automatically tell the MOST transceiver that the control message has been retrieved, or to let the software inform the MOST transceiver in the manner described in the MOST transceiver specification. Bits RCT and RCR in MIM Module Config Register, when set, will enable the automatic procedure for replies to previous transmissions, and brand new incoming messages, respectively. For simplicity, it is recommended that these bits are enabled.

(a) Ensure that the facility to automatically retrieve control messages is enabled by clearing the DPC and DCP bits in MIM_Module_Config. To retrieve replies from the MOST's transmit buffer (replies are relevant for specific types of Control messages only), the ECT bit must also be set. To retrieve new incoming messages from the MOST's receive buffer, the ECR bit must also be set.

(b) Read MIM Control Config Register and poll bits CMR or CMT to see when a message has been retrieved from the MOST transceiver's Receive or Transmit Control buffer respectively – or, optionally, await an interrupt from the RCM bit of MIM Interrupt Status Register. Note that prior to this, the RCT interrupt will flag when the MOST transceiver has received a new control message. However, unlike RCM, the RCT bit does not indicate that the MIM has retrieved the message ready for reading. (The interrupt bits set conditions have been explained later in this document.)

(c) Read the message or reply to a message from MIM Control Msg Register – exactly five longwords reads must occur.

(d) If bit CMT is set, and the message is a reply to an earlier transmission, check bit EMT in MIM Control Config Register. If this bit is set, then an error occurred during transmission, and the details can be retrieved from field ERR in the register. The error codes used are identical to those specified in the bXTS register of the MOST transceiver specification. Bit CMT reflects the TXR (unsuccessful transmission) field of the MOST bMSGS register.

(e) Write '1' to CMR or CMT in MIM Control Config Register, as appropriate, to acknowledge receipt.

(f) The MIM then releases the lock on the control message buffer. (LMB = '0')

Clearing of the Interrupt bits does not change the contents of MOST Transceiver registers bMSGS and bMSGC automatically, unless the corresponding bits in the MIM_Module_Config register are set (RCT and RCR)

Special Case - Large amounts of Control Messages received and high system latency.

If system latency is such that it is necessary, when the MIM is configured form automatic control message retrieval, to stop the MIM from perpetually reserving the control message buffer for incoming messages – ie to give local software a chance to reserve the control message buffer lock for an outgoing message – then it should be noted that the transmit lock request bit in MIM Control Config Register, LRT, cannot be set before writing '1' to CMR. This means that, if system latency is high, the control message buffer may be re-allocated to another incoming MOST messages. To prevent this happening, the following workaround sequence may be used:

1. Retrieve the Rx message
2. Disable ECT and ECR bits in MIM Module Config Register - this stops the MIM grabbing the lock
3. Write 1 to the CMR bit of MIM Control Config Register
4. Write 1 to LRT - note this must be a separate write to step 3
5. Re-Enable the ECT and ECR bits in MIM Module Config Register

20.7.6 Addressing formats

There are a variety of ways of addressing each node on the MOST networks, and Figure 20.2 below gives a general overview. Deciding whether control messages are addressed to the local node is done by the MOST transceiver, in accordance with the MOST specification. For flexibility, incoming packets are accepted if any of the address types described below are matched.

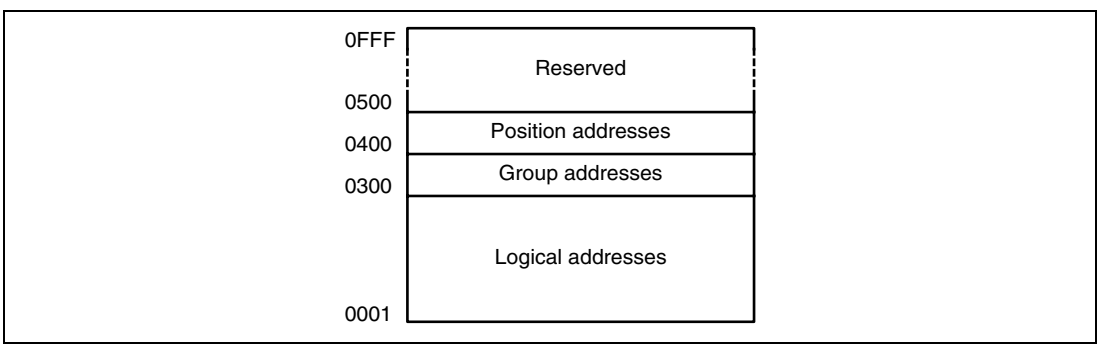


Figure 20.2 Address Map

Logical addresses are defined in the transceiver's bNAH/L registers. Packets can also use an alternative logical address defined in bAPAH/L. Valid values are between H'0001 and H'02FF.

The MOST specification states that control messages may use any of the above addressing formats except for alternative logical addresses. Packet messages may only use logical addresses as defined in bNAH/L and bAPAH/L. For added flexibility, the MIM will accept incoming packets

whose addresses match any of the formats described here – group, positional, broadcast or logical. For control messages, however, the decision to accept or reject is taken by the MOST transceiver.

Group addresses are formed by prefixing a group ID with H'03. Therefore, to send messages to a group of nodes whose transceiver bGA registers are set to H'57, the address used would be H'0357. Note that 0H'03C8 is a special address, reserved for broadcasts to the entire network.

Position addresses are formed by prefixing a node's position in the network with H'04. Therefore, to send messages to a nodes whose transceiver bNPR register is set to H'01, the address used would be H'0401. Note that position addresses are determined by the network configuration, and the master node is always H'0400. Therefore, when addressing messages to the master node, H'0400 should always be used as the address to guarantee receipt.

20.8 Configuration of the MOST transceiver

It is the responsibility of the software to put the MOST transceiver into Parallel Combined mode, and to configure the Routing Engine correctly as described in the MOST specification.

The MIM StreamX Config Registers must be configured in the following way. The procedure to change the configuration of a stream is outlined in the following two sections:-

20.8.1 Canceling a stream currently in use

(a) Re-configure the MOST incoming or outgoing routing engine, as appropriate, so that the MOST transceiver ignores any data presented by the MIM for the stream being cancelled. This is achieved by setting the relevant locations, for the stream being cancelled, to their default values.
(b) Send a Control Message to the Master MOST node, requesting that the previous channel allocations for the stream are deallocated.
(c) Await confirmation of the deallocation.
(d) Disable the DMA transfer to this stream, if applicable.
(e) Disable the corresponding ES1, ES2, ES3 or ES4 bit in MIM Module Config Register.
(f) Write to the corresponding MIM StreamX Config Register, setting the DMA and CLR bits to purge any old data or DMA activity.

20.8.2 Setting up a new data stream

(a) Ensure that the ES1/2/3/4 bit in MIM Module Config Register has been disabled and that the MIM StreamX Config Register has been written to, as described in part (f) above, with a view to purging old data or DMA activity.

(b) Send a Control Message to the Master MOST node, requesting for the required bandwidth (in multiples of four bytes) to be reserved. See the MOST transceiver specification for information on Control Message types.

(c) Await a reply to the Control Message from the Master MOST node. This reply will describe which channels in the MOST frames have been reserved.

(d) Configure the incoming and/or outgoing routing engine of the MOST transceiver in accordance with the MOST transceiver specification, using MIM MOST Reg Wr Register.

(e) Configure the DMAC for DMA transfer, if required. Refer to HD64404 DMAC Block specification.

(f) Write to the appropriate MIM StreamX Config configuration Register with new configuration information. Details of how to configure this register are below. The CLR bit should be set to '0', and if DMA is required, then the DMA bit should also be set to '0'.

(g) Enable the corresponding ES1, ES2, ES3 or ES4 bit in MIM Module Config Register.

Note: If DMA is used for transferring Streaming data through the MIM, then, the data stream should be continuous without having any limits on the number of bytes to be transferred. If a Stream is enabled in MIM, then MIM expects a continuous stream of data till the application disables the Stream. Example of Streaming data is output of a CD player, which will always transfer data, till the CD player is in ″play″ mode. If the number of bytes to be sent or received is limited and if DMA is used, then, after the last Quadlet is written to or read from the MIM FIFOs, FE bit is ″set″ in the MIM_Intterrupt_Status register as MIM expects continuous stream of data. To avoid this, we have to disable DMA as soon as the last byte is received or sent to the MIM FIFO. This may not be possible, since, it is very difficult to detect the exact time, when the last byte has been sent or received by the MIM FIFOs.

20.8.3 Programming MIM Module Config'

The QA field has 15 bits, each of which corresponds to one of the usable quadlets in the data sequence written to or read from the MOST transceiver. For example, bit 0 corresponds to quadlet 0 (the first quadlet of the SF0), and bit 7 corresponds to quadlet 7 (the last quadlet in SF3). For clarity, refer to Figure 20.3 below. The last quadlet can never be used for streaming data, even when the maximum permissible streaming bandwidth has been allocated. By default the MOST only allocates the first 6 quadlets to streaming data, but appropriate programming of the MOST bSBC register can allow quadlets 6 - 14 to be used for streaming channels.

One MOST frame period															
SF0		SF1		SF2		SF3		SF4		SF5		SF6		SF7	
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
Bit 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	STAT

Figure 20.3 MOST Interface Format

Any further re-ordering of data can be achieved by programming the MOST transceiver's routing engine to achieve the desired effect.

For each of the MIM StreamX Config Registers, if a QA bit is set then one quadlet of information is routed from that stream into the corresponding quadlet of the MOST frames. This allows each stream to be reprogrammed independently of the others. It is only possible to configure quadlets in this way, streams can only be allocated by the MIM in multiples of 4 bytes.

As an example, suppose the configuration registers are programmed as illustrated in Figure 20.4. Note it is possible for two streams to occupy the same Quadlet as long as one is receive and the other is transmit. This is illustrated for streams 3 and 4, bit position 4. The resulting data sequences sent to and received from the MOST transceiver are shown in Figure 20.5 and Figure 20.6.

REG	BIT	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Stream 1 Tx		0	0	0	0	0	0	0	0	**1**	0	0	0	0	**1**	**1**
Stream 2 Tx		0	0	0	0	0	**1**	0	0	0	0	0	0	**1**	0	0
Stream 3 Rx		0	0	0	0	0	0	0	0	0	0	**1**	0	0	**1**	**1**
Stream 4 Tx		0	0	0	0	**1**	0	**1**	**1**	0	**1**	**1**	**1**	0	0	0

Figure 20.4 Example Configuration

One MOST frame period – **OUTGOING** (TX)															
SF0		SF1		SF2		SF3		SF4		SF5		SF6		SF7	
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
STR 1	STR 1	STR 2	STR 4	STR 4	STR 4	STR 1	STR 4	STR 4	STR 2	STR 4	n/a	n/a	n/a	n/a	

Figure 20.5 Example of Outgoing Data Sequence

One MOST frame period – **INCOMING** (RX)															
SF0		SF1		SF2		SF3		SF4		SF5		SF6		SF7	
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
STR 3	STR 3	n/a	n/a	STR 3	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	

Figure 20.6 Example of Incoming Data Sequence

Within a quadlet, bits 31 to 24 of the stream data are used by the first of the four bytes, bits 23 to 16 are used by the second byte, bits 15 to 8 are used by the third byte and bits 7 to 0 are used by the fourth and final byte.

20.8.4 Example streaming application

As a further example, consider the following case, using the procedure outlined at the start of this section. The desire is to send one quadlet of streaming information per frame using MIM Streaming channel number 1.

(a) Ensure that ES1 in MIM Module Config Register is disabled, and then write to MIM Stream1 Config Register, setting the DMA and CLR bits. This ensures that the Stream 1 buffers and DMA activity are purged.

(b) Send a Control Message to the Master MOST node, requesting 4 bytes of bandwidth.

(c) The reply indicates that, for example, MOST byte channels 0, 6, 9 and 20 are allocated.

(d) Configure the outgoing routing engine of the MOST transceiver. The desire in this example is for the MIM to present the streaming data in quadlet 8 (software chooses which quadlet the MIM uses to present data), and for the MOST to route those four bytes to byte channels 0, 6, 9 and 20, as dictated by the Master node in step 3.

For outgoing data, the MOST routing engine addresses are based on the byte allocations that the Master node has granted within the MOST frames – this is different to the addressing method for incoming data. The MOST routing engine address for outgoing bytes is (0x00 + destination byte number allocated by timing Master), so in this case the relevant routing engine addresses for byte channels 0, 6, 9 and 20 are H'00, H'06, H'09 and H'14. These must be allocated to quadlet 8 of the MIM's transfer, ie to outgoing bytes H'20, H'21, H'22, H'23. To allocate MIM bytes to the outgoing frame, the routing engine address must be written to with

(H'40 + MIM's outgoing byte number), in this case 0 × 60, 0 × 61, 0 × 62 and 0 × 63. Therefore for outgoing data:

1. Write H'60 to MOST location H'00
2. Write H'61 to MOST location H'06
3. Write H'62 to MOST location H'09
4. Write H'63 to MOST location H'14

As an aside, suppose the stream is being configured for incoming data during quadlet 8, rather than outgoing data. For incoming data, the MOST routing engine addresses are based on the quadlets during which the data is transferred by the MIM – this is different to the addressing method for outgoing data. The MOST routing engine address for incoming bytes is (H'40 + byte number that the MIM looks at), so if the MIM expects incoming bytes during quadlet 8 (bytes H'20 to H'23) the relevant routing engine addresses are H'60, H'61, H'62 and H'63. These bytes must be routed from MOST frame channels 0, 6, 9 and 20 respectively. Therefore for incoming data:

5. Write H'00 to MOST location H'60
6. Write H'06 to MOST location H'61
7. Write H'09 to MOST location H'62
8. Write H'14 to MOST location H'63

(e) Configure the DMAC for DMA transfer.

(f) Configure MIM Stream1 Config Register to use, as chosen by software in step (d), quadlet 8 of the outgoing frame. (Suppose that, in this example, bSBC in the MOST transceiver is set to H'0B, which allows quadlets 0 to 10 to be used by the MIM for streaming data). As DMA is being used, both the DMA and CLR bits are set to '0'. Therefore H'00000100 is written to MIM Stream1 Config Register.

(g) Enable ES1 in MIM Module Config Register.

Refer to the MOST transceiver specification for more detailed information on the programming of the routing engine for Parallel Combined / Parallel Synchronous mode. At the time of writing, this information can be found both in the Routing Engine chapter (but avoid the "serial" subsections) and also the "Configuring Parallel Combined Mode" section.

20.9 Accessing Transceiver Registers

It is possible to access the registers of any MOST transceiver connected to the local network. Remote nodes must be accessed using control messages as described in the MOST specification, but registers on the local node can be accessed using the procedure shown in Figure 20.7.

Although it is possible to read transceiver registers by using the MIM MOST Reg Rd Register described on page 857, data is not valid immediately as it must be read from the transceiver. Bit XRR must be polled, or the MIM can be programmed to issue an interrupt when ready. Bit XRR is cleared automatically by the MIM when a new address is written into the address filed of MIM_MOST_Reg_Rd register.

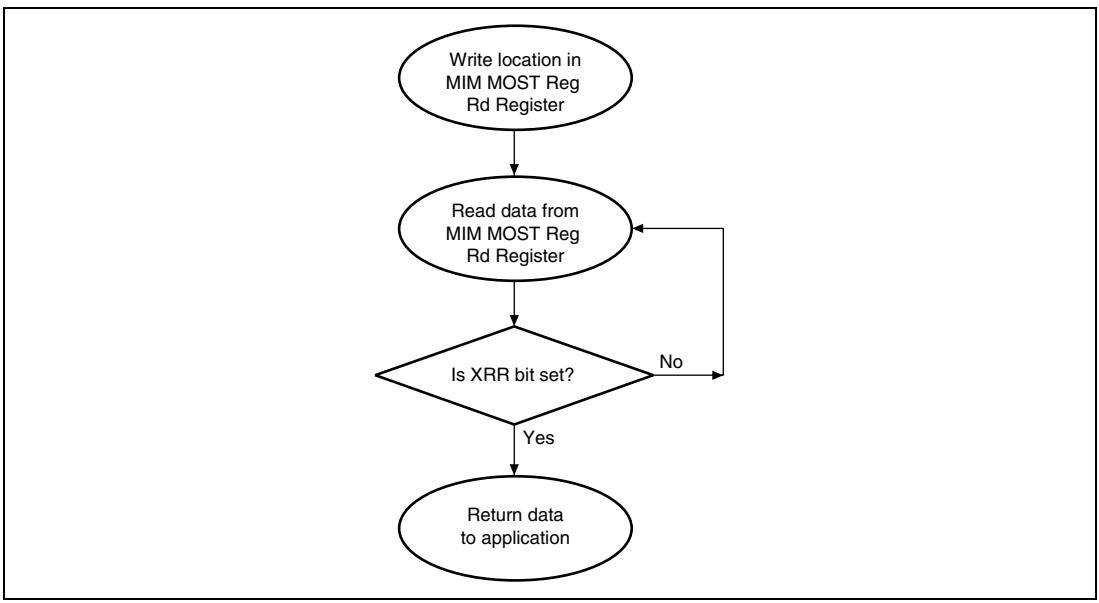


Figure 20.7 Procedure for Reading Transceiver Registers

Transceiver registers can be written to using a single access to the MIM MOST Reg Wr Register described earlier . However, if successive writes are required, then the processor must wait for any backlog to clear by following the procedure in Figure 20.8 below. As an alternative to polling bit XWR, the MIM can be programmed to issue an interrupt when ready.

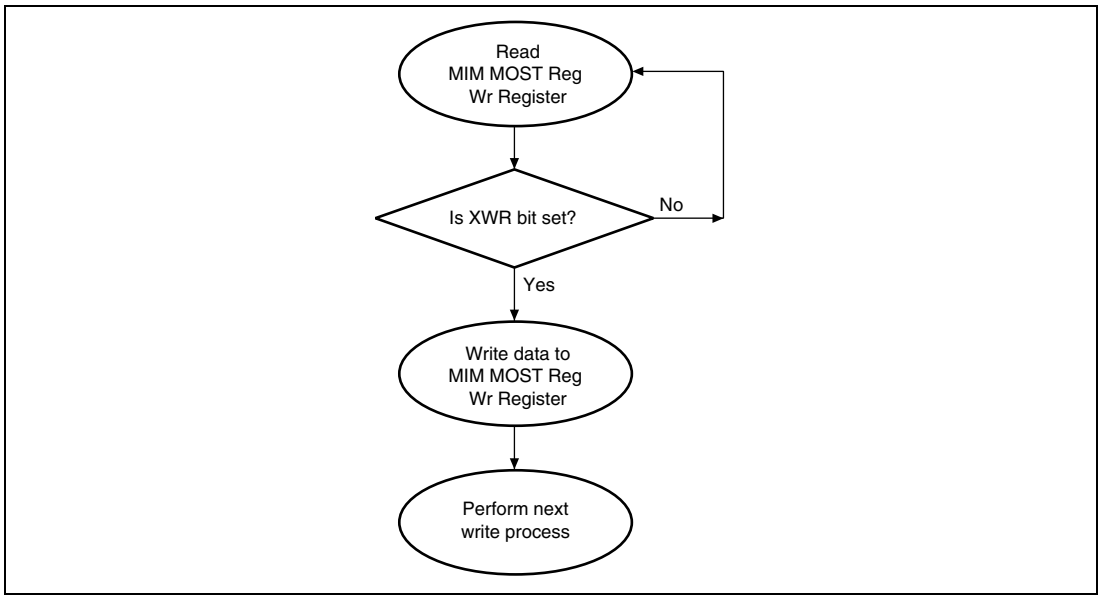


Figure 20.8 Procedure for Writing to Multiple Transceiver Registers

The slow nature of accesses to and from the MOST transceiver means that, if the processor were to write to many transceiver registers sequentially, without a break, then data would be lost. Not only are transceiver accesses slow, but they must also be scheduled for suitable gaps in the incoming and outgoing frame data accesses, so that reception and transmission of data cannot be affected.

Note that if a large block of reads or writes is to occur from consecutive transceiver locations, the transfer of data is most efficient if the DPC, DPS and DPA bits of MIM Module Config Register are set prior to the block transfer, and cleared after the transfer. The setting of these bits inhibits the MIM from periodically polling transceiver registers such as bMSGS, bSBC, bNAH, bNAL, bGA, etc. If this periodic polling were allowed to continue, then the transceiver would see accesses from these registers in between the block accesses, meaning that the MAP (address of transfer) would continually be changing, slowing down the accesses. Of course, if polling of these registers is never a requirement, then the feature can be disabled at power-up and no special action needs to be taken during block transfers.

The MIM Interrupt Status Register on section 20.5.7, MIM Interrupt Status Register, also incorporates two bits to help monitor the status of these XRR and XWR ready signals. Once set,

the XRR and XWR bits remain set until the respective registers are written to again with a new read/write request.

If using the XAI bit of MIM MOST Reg Wr Register or MIM MOST Reg Rd Register to achieve block reads or block writes with automatically incrementing addresses, then the first time the read/write is programmed, the XAI bit should be clear and the XA field should contain the start address. For subsequent accesses, the XAI bit can be set to automatically increment the address, in which case the XA address field will be ignored. Whenever XAI is clear, then XA must contain a valid address.

20.10 Transceiver Power Up Procedure

Page 160 of the MOST transceiver specification version 1.1-00 describes the "Read/Write set-up" time which must be allowed between the last source port access of an SF interval, and the start of the next SF period. It comments, in the paragraph describing how to calculate this set-up time, that "Bits SPR2. 0 in bSDC2 must always be set to '101'" when using the formula in the specification.

It is therefore required that after reset, register bSDC2 is written to with a value of H'A0 to comply with that paragraph in the MOST specification. This is in addition to the normal power-up procedure described in section 16 of the MOST specification.

In order for the MIM to quickly detect transceiver loss of lock situations, the MOST's error pin must be programmed to go active when lock is lost. This means that the MOST's Transceiver Status register, bXSR, needs to be set up so as to mask all types of error apart from transceiver lock errors. In the current version of the specification, the default value of H'50 is correct and currently no action needs to be taken.

In summary, the procedures required after power-up, in addition to those described explicitly in the transceiver specification, include (but are not necessarily limited to):

(a) Ensure the board design configures the MOST transceiver for parallel synchronous operation by pulling its PAR SRC pin high, and its ASYNC pin low.
(b) If the MIM is to be allowed access to the control port of the MOST transceiver, ensure the board design configures the MOST transceiver for parallel control port operation by pulling its PAR CP pin high.
(c) Enable the MIM by writing to MIM Module Config Register, setting bits ME, ESP and POE, and programming the CLK field with the correct clock period for the target system. If the MIM is to be allowed access to the MOST registers or control messages via the control port, then clear bit DCP, otherwise set it to disable control port accesses.
(d) Poll MIM Interrupt Status Register until bit 31 (XSE, Transceiver Status Error) is set, indicating that the transceiver has completed its power-up initialisation process
(e) Write H'A0 to MOST register bSDC2 in accordance with the MOST specification
(f) Ensure that MOST Register bXSR is set to H'50 in accordance with the MOST specification

(g) Configure the incoming routing engine in accordance with the MOST specification

(h) Configure the outgoing routing engine in accordance with the MOST specification

(i) Put the MOST transceiver into parallel combined mode by writing H'01 to bPCMA

(j) Consider updating the MIM's copies of critical internal transceiver registers, if register polling or accesses to the control port are disabled.

(k) Determine the allocation of bandwidth for streaming data, via bSBC in the MOST transceiver.

20.11 Automatic polling of transceiver registers

If Control Port accesses are completely disabled by setting bit DCP in MIM Module Config Register or if automatic polling of certain MOST Registers is disabled, then the software must at some stage allow the MIM to find out the values of certain key MOST registers. For instance, the correct value of SBC must always be known, and for incoming packets the node address registers must also be known.

The mechanism for achieving this is:

1. If manual CP accesses are allowed and only the MOST transceiver knows the correct value, read the value from the transceiver using MIM MOST Reg Rd Register
2. If no CP accesses are allowed at all, or only the software knows the correct value, write the value to the transceiver using MIM MOST Reg Wr Register. If CP accesses are completely disabled, then this action lets the MIM know what the value should be, but the MIM does not attempt to write it to the transceiver.

There are two tiers of automatic polling. Every frame, the MIM tries to perform the following sequence:

(a) Poll SBC if DPS in MIM Module Config Register is clear. If DPS is not clear, software must manually tell the MIM what SBC value to use, via one of the mechanisms described above.

(b) Read or write to control port if MIM MOST Reg Wr/Rd Register requested

(c) Send / receive a control message byte if enabled and required

(d) Poll MSGS if DPC in MIM Module Config Register is clear.

(e) Continue the round-robin polling sequence of addressing registers from where it last left off, if the DPA bit in MIM Module Config Register is clear. The contents of these registers are private to the MIM, and can only be accessed by software if it manually tells the MIM to read from the corresponding MOST Register location. If DPA is set and polling is disabled, then the MIM should be manually configured with correct values for the following registers:
   - bNPR Node Position register
   - bGA Group Address register
   - bNAH Node Address High
   - bNAL Node Address Low

- bAPAH Packet Address High
- bAPAL Packet Address Low

20.12 Interrupt sources

Most of the possible interrupt sources in the MIM Interrupt Status Register are described in the corresponding functional sections. This section offers general notes on interrupt handling.

For interrupts related to flags available from the MOST transceiver's interrupt register, it should be noted that the MIM polls MOST Registers for the respective errors, rather than monitoring the MOST $\overline{\text{MINT}}$ pin and relying on the MOST transceiver itself being programmed to flag these interrupts. However, the XSE interrupt bit can be programmed to flag active whenever it detects a falling edge on the $\overline{\text{MINT}}$ pin, if it is required that the MOST transceiver interrupt processing capabilities are used.

Interrupt sources which depend on information from the MOST transceiver will only work if both polling of the respective MOST Registers, and control port accesses, are enabled. If such an interrupt is received, then the interrupt service routine must deal with the source of the interrupt in the MOST transceiver, by writing to a MOST Register appropriate to the interrupt, before clearing the MIM's interrupt. The MIM itself does not automatically clear down MOST interrupts at source. Examples of such interrupt source include the ALC bit of MSGS, and the XSE interrupt.

The "FIFO Ready" interrupt sources mirror the corresponding bits in the MIM Buffer Ready Register, except that the FCM interrupt is active whenever any of MIM Buffer Ready's Register Control Message bits (RCT, RCR and RMT) are active. Similarly, XWR and XRR reflect the status of the corresponding bits in MIM MOST Reg Wr Register and MIM MOST Reg Rd Register.

The RPU "Receive Packet Unpacking" interrupt can be set after one of the following events:-

(a) MOST indicates a packet reception error when the MIM is receiving a packet.
(b) MOST indicates a new packet reception when the MIM is still receiving a previous packet.
(c) Software flushes the MIM PacketRx FIFO while a packet is being received.

The TCE "Transmit Control Message error" interrupt is active when a control message reply is received, with an error – ie in the MOST MSGS register, bits MTX and TXR are both set.

The FE "FIFO Error" interrupt is active when software tries to read from an empty buffer or write to a full buffer. To clear this interrupt, the FIFOs have to be cleared by setting the CLR bit in the corresponding configuration registers (MIM_StreamX_Config, MIM_Control_Config, MIM_PacketTx_Config and MIM_PacketRx_Config) for the FIFOs. After clearing the FIFOs, the FE bit can be cleared in the MIM_Interrupt_Status register. If the FIFOs are not cleared earlier, then, the FE bit will be set again for every read or write operation from or to the FIFO, which overflows or under flows.

The CE "Conflict Error" interrupt becomes active after one of the following events:-

(a) A second packet transmission is initiated before the first one has finished

(b) A second write via MIM MOST Reg Wr Register is initiated before the first one has finished

(c) The Control Message buffer is written to without the lock being obtained beforehand

(d) The Control Message buffer is read from before a message has been stored

The TPT interrupt indicates when the MIM has physically transferred an outgoing packet to the MOST transceiver. However, this interrupt should be ignored because TPM, which is active when the transceiver has finished putting the packet onto the network, is far more useful. The TPM interrupt signifies when the MIM itself is ready for a new packet transmission, and it mirrors the clearing of the TPR bit in MIM PacketTx Config Register.

Similarly, the RPT interrupt indicates when the M IM has started to receive a correctly addressed incoming packet from the MOST transceiver. Because DMA must now be used to handle incoming packets, this interrupt should be ignored, as the alternative RPM interrupt will go active when an entire packet has been fully received and stored via DMA.

The XIC "Transceiver Initialisation Complete" interrupt is active whenever the transceiver signals, via its $\overline{\text{MINT}}$ pin, that it is initialised after a hardware or software reset.

The CSB, CGA, CLA and CPA interrupts report when the MIM has detected a change in certain MOST transceiver registers. These interrupts will only function correctly if control port accesses are enabled, along with polling of the appropriate transceiver registers. These interrupts report changes in the MOST's bSBC (Synchronous Bandwidth), bGA (Group Address), bNAH / bNAL / bAPAH / bAPAL (Logical addresses) or bNPR (Node Position register).

The XLR interrupt is an indicator of the health of the MOST network, via the MOST transceiver's "error" pin. If the network has locked correctly and has been stable for a few milliseconds, then XLR will be set, and it will be impossible to clear the XLR bit until something goes wrong and the MOST networks loses its lock. On the contrary, the XLE bit provides an immediate indicator when the MOST network loses its lock (see "Transceiver loss of lock procedure" below). The sequence as lock is regained is that first XLE will disappear (perhaps sporadically), and then after a few milliseconds of stability XLR will be set. A typical procedure for using these bits, after network lock has first been achieved, is:

(a) Unmask XLE and mask XLR, and clear both the XLE and XLR interrupt flags.

(b) Process data normally unless the XLE interrupt occurs.

(c) Mask XLE, unmask XLR, and clear both the XLE and XLR interrupt flags.

(d) At this point, no streaming or packet data can be transmitted via the MOST networks. Existing packet transfers will be aborted. Tidy-up the packet transmission configuration registers, and the corresponding DMAC channels, purging the MIM's and DMAC's FIFO Buffers to remove traces of the last aborted transmission. Disable packet transmission and reception – any attempt to initiate a packet transmission will be ignored. Streaming configurations can be left enabled, and control port functions (MOST register accesses and Control Messages) are unaffected.

(e) Wait until the XLR interrupt occurs.

(f) Packet and streaming data can now be used again. Repeat the process from step 1.

Description of MIM Interrupt status registers bits.

Bit 31 : XSE
Set condition: Falling edge of $\overline{\text{MINT}}$ input.

Bit 30: XLR
Set condition: MOST Transceiver lock error removed and MOST network stable for the number of frames set in the MIM_Status register (bits [7:0]).

Bit 29: XLE
Set condition: ERROR pin of MOST Transceiver is '1'.

Bit 28: RPU
Set condition: Error during packet reception. Incorrect flushing of FIFO.

Error indicated by MOST transceiver in the status bytes.

Bit 26: TCE
Set condition: MTX bit is '1' and the TXR bit is '0' in bMSGS register of MOST Transceiver

Bit 25: TPL
Set condition: Length in header doesn't match the value set in the packet TXconfig register.

Bit 24: FE
Set condition: Error during reading or writing of FIFOs of MIM.

Bit 23: CE
Set condition: Second packet transmission is initiated before the first one has finished.
Second write via MIM_MOST_reg_wr is initiated before first write has finished.
Control message buffer is written to without obtaining lock.
Control message buffer is read before a message has been stored.

Bit 22: XWR
Set condition: MIM_MOST_Reg_Wr register is ready to write data.

Bit 21: XRR
Set condition: Data is available in the MIM_MOST_Reg_rd register for reading.

Bit 20: FPR
Set condition: Mirrors the corresponding bit in MIM_Buffer_Ready register.

Bit 19: FPT
Set condition: Mirrors the corresponding bit in MIM_Buffer_Ready register

Bit 18: FCM
Set condition: When RCT or RCR or RMT bits set in MIM_Buffer_Ready register.

Bit 17: FS4
Set condition: Mirrors the corresponding bit in MIM_Buffer_Ready register

Bit 16: FS3
Set condition: Mirrors the corresponding bit in MIM_Buffer_Ready register

Bit 15: FS2
Set condition: Mirrors the corresponding bit in MIM_Buffer_Ready register

Bit 14: FS1
Set condition: Mirrors the corresponding bit in MIM_Buffer_Ready register

Bit 13: TPT
Set condition: when new packet is transferred from MIM to MOST transceiver.

Bit 12: RPM
Set condition: when new packet is received by MIM.

Bit 11: TCT
Set condition: when Control message is transmitted to the MOST transceiver.

Bit 10 : RCM
Set condition: when control message or a reply is received from MOST transceiver.
Received control message has priority over

Bit 9: TPM
Set condition: when AINTN signal goes to '1' at the end of packet transmission.

Bit 8: RPT
Set condition: when the header bytes of a new packet data are received by MIM with start field set to '1'.

Bit 7: TCM
Set condition: when bit MTX is '1' in the bMSGS register of the MOST Transceiver.

Bit 6: RCT
Set condition: when bit MRX is '1' in the bMSGS register of the MOST Transceiver.

Bit 5: XIC
Set condition: when $\overline{\text{MINT}}$ pin goes to '0' after power-on reset.

Bit 4: CSB
Set condition: change in SBC for packet TX in MOST.

Bit 3: CGA
Set condition: change in group address

Bit 2: CLA
Set condition: change in most node address or most alternative packet address.

Bit 1: CPA
Set condition: change in the node position address

Bit 0: ALC
Set condition: bit 3 (ALC) is '1' in bMSGS register in MOST Transceiver.

20.13 Transceiver loss of lock procedure

In the event of the lock in the MOST optical network being lost, there is severe disruption to network services and source port data (streaming and packet) is unreliable. Furthermore, the PLL on the local MOST transceiver will not be locked, and the frame rate of the local MOST transceiver will drift down to a much lower frequency than usual. Therefore, as a precautionary measure, the MIM performs the following actions:

(a) Flag an interrupt status bit to say that lock has been lost.

(b) Discard incoming stream data, and when lock resumes, resynchronise the companion chip data stream with the MOST data stream, and start receiving again

(c) Stop sending outgoing stream data, and instead send zeros. When lock resumes, resynchronise the companion chip data stream with the MOST data stream, and start sending data again

(d) Stop sending packet data, and abort the current packet data Tx request. Software must re-send the last packet after lock is regained

(e) Stop receiving incoming packet data, and flag a packet reception error. This, and the "loss of lock" interrupt status bits in both the MIM and the MOST transceiver, must be cleared by software before new packets are received.

20.14 Interfaces to the MOST transceiver

Describing the precise timing and interfacing between the MOST Interface Module and the MOST transceiver is beyond the scope of this specification. For further information, see the MOST Transceiver datasheet OS8104, which specifies the interface to which the MOST Interface Module supports.

Figure 20.9 below gives an overview of the data communications between the MIM and the MOST transceiver. For more details on this interface, please refer to the MOST transceiver specification.

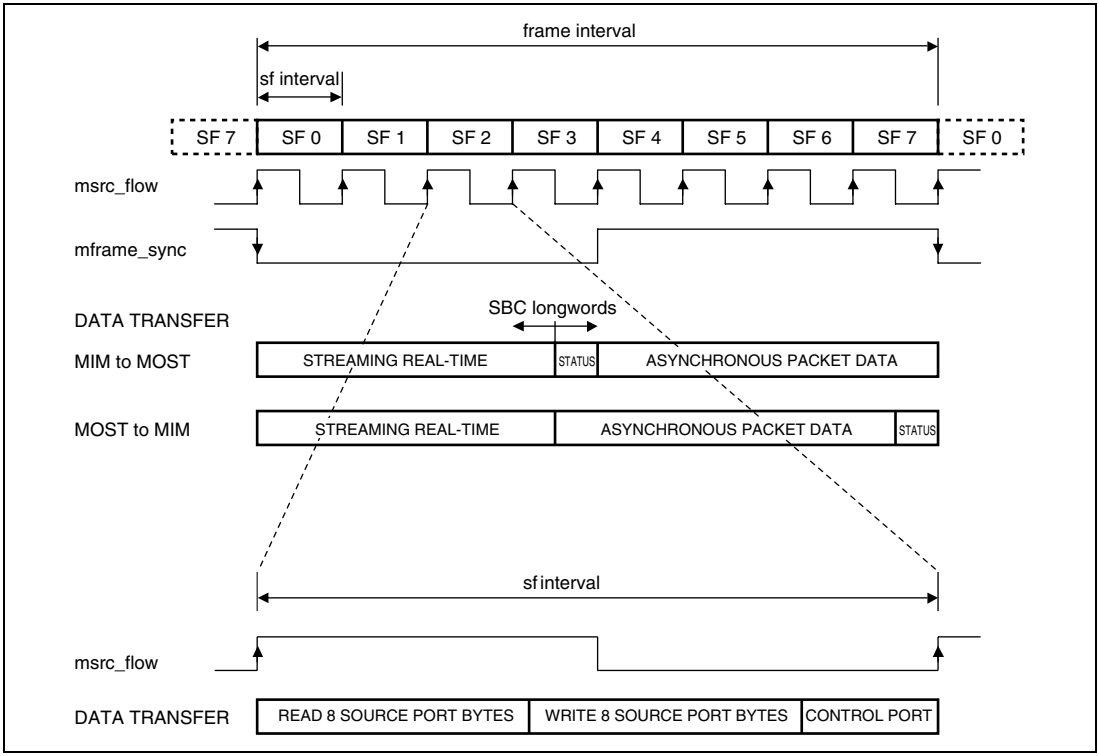


Figure 20.9 Communications between the MIM and the MOST Transceiver

20.15 MOST Interface Module Standby Mode

The MOST Interface module allows clock gating to reduce power consumption.

To power down the module, the following procedure is required:

1. Write to MIM Module Config Register, clearing the ECT, ECR and EMT bits and setting the DPC, DPS and DPA bits.
2. Wait for all outstanding MOST control port accesses to finish, by monitoring the XRR and XWR bits in MIM MOST Reg Rd Register and MIM MOST Reg Wr Register respectively if necessary.
3. Wait for any Packet Transmissions to finish.
4. Wait for any Control Message Transmissions to finish.
5. Read MIM Control Config Register. If the LMB bit is set, wait until the CMR bit is set and process the incoming control message as normal. It will no longer be possible to send outgoing replies.
6. If DMA is used, program the DMAC to cancel all DMA activity via the MOST Interface Module

7. Program the MOST transceiver's routing engine so that it does not accept any outgoing bytes from the MOST Interface Module. Details of how to achieve this are in the MOST Transceiver specification.
8. Write to the four MIM StreamX Config Registers, setting the DMA bit and clearing the QA bits.
9. Write to MIM Module Config Register, clearing the ME, ES1, ES2, ES3, ES4, EPT and EPR bits.
10. Write to the Clock Control 1 Register in the Power Control module, clearing the MOST bit
11. Write to MIM Control Config Register, MIM StreamX Config Register, MIM PacketTx Config Register and MIM PacketRx Config Register, setting the CLR bit in each register.
12. Write to MIM Interrupt Enable Register, clearing all bits.
13. Write to the Clock Control 1 Register in the Power Control module, clearing the MOST bit

To wake up the module, the following procedure is required:-

1. Write to the Clock Control 1 Register in the Power Control module, setting the MOST bit
2. Write to MIM Interrupt Status Register, clearing all bits.
3. Configure the MOST Interface module in the usual way.

All MOST Interface Module registers except MIM Interrupt Status Register will have retained their pre-powerdown settings.

20.16 References

MOST Transceiver datasheet OS8104, by Oasis SiliconSystems (Version 1.1-00 used)
Hitachi Register Bus DMA Controller Specification

21.1 General Description

The UART communicates in asynchronous mode.

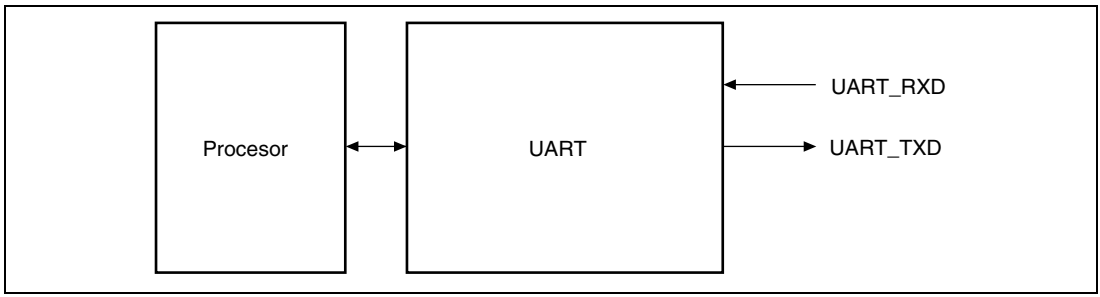


Figure 21.1 Overview Block Diagram

21.2 Features

Asynchronous mode for serial communication

21.2.1 Asynchronous Mode

Serial data communication is synchronised one character at a time. The UART can communicate with a universal asynchronous receiver/transmitter (UART), asynchronous communication interface adapter (ACIA), or other chip that employs standard asynchronous serial communication. There are different selectable serial data communication formats.

- Data length: 7 or 8 bits
- Stop bit length: 1 or 2 bits
- Parity bit: even, odd, or none
- Receive error detection: parity, overrun, and framing errors
- Break detection: by reading the UART_RXD level directly when a framing error occurs

21.3 Block Diagram

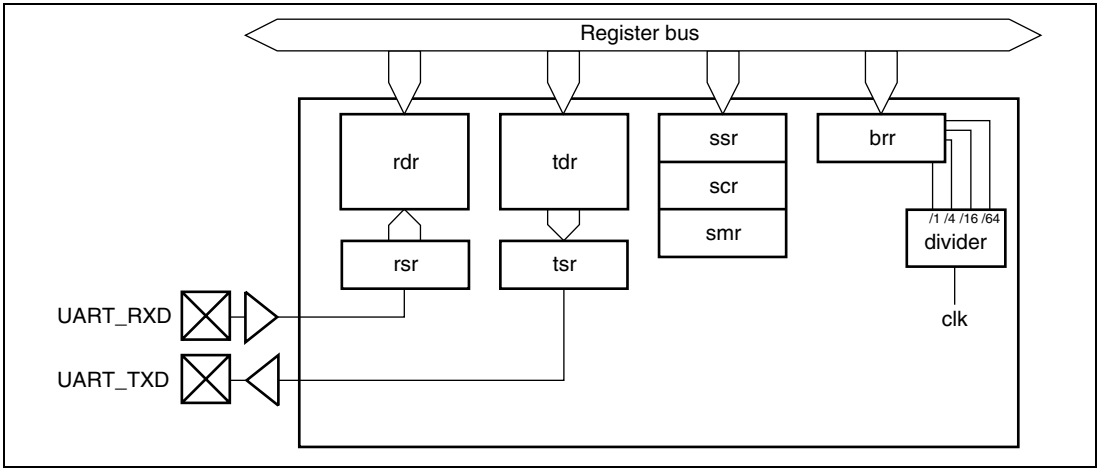


Figure 21.2 Functional Block Diagram

21.4 Interfaces

21.4.1 Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 21.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From	Synchronisation to Clocks
UART_RXD	1	In	Receive Data	External UART	rbclk after baud rate divider
UART_TXD	1	Out	Transmit Data	External UART	rbclk after baud rate divider

21.4.2 Software Interfaces

The registers accessible by the software are listed in the following table:

All registers addresses below are long word addresses only.

Table 21.2 Register List

Channel	Address (Bytes)	Register Name	Abbreviation	Access Size
0	H'6620	Serial Mode Register 0	SMR0	32
	H'6624	Bit Rate Register 0	BRR0	32
	H'6628	Serial Control Register 0	SCR0	32
	H'662C	Transmit Data Register 0	TDR0	32
	H'6630	Serial Status Register 0	SSR0	32
	H'6634	Receive Data Register 0	RDR0	32
	H'6638	Reserved		
	H'663C	IrDA Control Register 0	ICR0	32
1	H'6640	Serial Mode Register 1	SMR1	32
	H'6644	Bit Rate Register 1	BRR1	32
	H'6648	Serial Control Register 1	SCR1	32
	H'664C	Transmit Data Register 1	TDR1	32
	H'6650	Serial Status Register 1	SSR1	32
	H'6654	Receive Data Register 1	RDR1	32
	H'6658 to H'665C	Reserved		
2	H'6660	Serial Mode Register 2	SMR2	32
	H'6664	Bit Rate Register 2	BRR2	32
	H'6668	Serial Control Register 2	SCR2	32
	H'666C	Transmit Data Register 2	TDR2	32
	H'6670	Serial Status Register 2	SSR2	32
	H'6674	Receive Data Register 2	RDR2	32
	H'6678 to H'667C	Reserved		
3	H'6680	Serial Mode Register 3	SMR3	32
	H'6684	Bit Rate Register 3	BRR3	32
	H'6688	Serial Control Register 3	SCR3	32
	H'668C	Transmit Data Register 3	TDR3	32
	H'6690	Serial Status Register 3	SSR3	32
	H'6694	Receive Data Register 3	RDR3	32
	H'6698 to H'669C	Reserved		

All reserved and unused bits do not have a guaranteed value when read.

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

Receive Shift Register (RSR)

RSR is the register that receives serial data.

The UART loads serial data input at the UART_RXD pin into RSR in the order received, LSB (bit 0) first, there by converting the data to parallel data. When 1 byte has been received, it is automatically transferred to RDR. The CPU cannot read or write RSR directly.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RSR							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	—	**Reserved**
7 to 0	RSR	0	—	

Receive Data Register (RDR)

RDR is the register that stores received serial data. Bits 31 to 8 are reserved.

When the UART finishes receiving 1 byte of serial data, it transfers the received data from RSR into RDR for storage. RSR is then ready to receive the next data. This double buffering allows data to be received continuously.

RDR is a read-only register. The CPU cannot modify its contents. RDR is initialised to H'00 by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									RDR							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 0	RDR	0	R	

Transmit Shift Register (TSR)

TSR is the register that transmits serial data.

The UART loads transmit data from TDR into TSR, then transmits the data serially from the UART_TXD pin, LSB (bit 0) first. After transmitting one data byte, the UART automatically loads the next transmit data from TDR into TSR and starts transmitting it. If the TDRE (transmit data register empty) flag is set to 1 in SSR,, the UART does not load the TDR contents into TSR. The CPU cannot read or write TSR directly.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TSR							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	—	**Reserved**
7 to 0	TSR	0	—	

Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission. Bits 31 to 8 are reserved.

When the UART detects that TSR is empty, it moves transmit data written in TDR from TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR. The CPU can always read and write TDR. TDR is initialised to H'FF by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TDR							
Initial:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7 to 0	TDR	1	R/W	

Serial Mode Register (SMR)

SMR specifies the UART serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write SMR. SMR is initialised to H'00 by a reset and in standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						TDMA	OSM	RDMA		CHR	PE	OE	STOP		CKS1	CKS0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	0	R	**Reserved**
10	TDMA	0	R/W	**Transmit DMA Enable (TDMA)** Selects whether the DMA function to the DMAC module is enabled. 0: Transmit DMA function is disabled and the UART will not make any Transmit DMA requests to the DMAC module. 1: Transmit DMA function is enabled and Transmit DMA requests to the DMAC module will be made when either the TDR register is empty. The SCR Transmit Enable (TE) bit should be cleared to '0' prior to this bit being enabled. Once the Transmit DMA function is enabled, Transmit is enabled be setting the TE bit to '1'.
9	OSM	0	R/W	**Over-sample Mode (OSM)** Selects the number of samples per bit received in asynchronous UART mode 0: 16 samples per bit is selected, this is standard for UART operation. 1: 8 samples per bit is selected to enable bit rates up to 460800 bit/s.
8	RDMA	0	R/W	**Receive DMA Enable (RDMA)** Selects whether the DMA function to the DMAC module is enabled. 0: Receive DMA function is disabled and the UART will not make any Receive DMA requests to the DMAC module. 1: Receive DMA function is enabled and Receive DMA requests to the DMAC module will be made when the RDR register is full.

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	**Reserved**
6	CHR	0	R/W	**Character Length (CHR)** Selects 7-bit or 8-bit data length in asynchronous mode. 0: 8-bit data 1: 7-bit data* Note: * When 7-bit data is selected, the MSB (bit 7) in TDR is not transmitted.
5	PE	0	R/W	**Parity Enable (PE)** This bit enables or disables the addition of a parity bit to transmit data, and the checking of the parity bit in receive data. 0: Parity bit not added or checked 1: Parity bit added and checked* Note: * When PE is set to 1, an even or odd parity bit is added to transmit data according to the even or odd parity mode selected by the OE bit, and the parity bit in receive data is checked to see that it matches the even or odd mode selected by the OE bit.
4	OE	0	R/W	**Parity Mode (OE)** Selects even or odd parity. The OE bit setting is valid when the PE bit is set to 1 to enable the adding and checking of a parity bit. The OE setting is ignored when parity adding and checking is disabled in asynchronous mode. 0: Even parity*[1] 1: Odd parity*[2] Notes: *1 When even parity is selected, the parity bit added to transmit data makes an even number of 1s in the transmitted character and parity bit combined. Receive data must have an even number of 1s in the received character and parity bit combined. *2 When odd parity is selected, the parity bit added to transmit data makes an odd number of 1s in the transmitted character and parity bit combined. Receive data must have an odd number of 1s in the received character and parity bit combined.

Bit	Bit Name	Initial Value	R/W	Description
3	STOP	0	R/W	**Stop Bit Length (STOP)** Selects one or two stop bits. 0: One stop bit*[1] 1: Two stop bits*[2] Notes: *1 One stop bit (with value 1) is added at the end of each transmitted character. *2 Two stop bits (with value 1) are added at the end of each transmitted character. In receiving, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1 it is treated as a stop bit. If the second stop bit is 0 it is treated as the start bit of the next incoming character.
2	—	0	R	**Reserved** Keep to '0'
1	CKS1	0	R/W	**Clock Select 1 and 0 (CKS1, CKS0)**
0	CKS0	0	R/W	These bits select the clock source of the on-chip baud rate generator. Four clock sources are available: ϕ, ϕ/4, ϕ/16, and ϕ/64. For the relationship between the clock source, bit rate register setting, and baud rate, see Bit Rate Register (BRR). 00: ϕ 01: ϕ/4 10: ϕ/16 11: ϕ/64

Serial Control Register (SCR)

SCR enables the UART transmitter and receiver, and enables or disables interrupts.

The CPU can always read and write SCR. SCR is initialised to H'00 by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								ME	TIE	RIE	TE	RE		TEIE		
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	0	R	**Reserved**
8	ME	0	R/W	**Module Enable (ME)** Selects whether the module is enabled 0: The module is disabled and all ports are set to input or tri-state. 1: The module is enabled and all ports are set as configured with other register bits.
7	TIE	0	R/W	**Transmit Interrupt Enable (TIE)** Enables or disables the transmit-data-empty interrupt (TXI) requested when the TDRE flag in SSR is set to 1 due to transfer of serial transmit data from TDR to TSR. 0: Transmit-data-empty interrupt request (TXI) is disabled* 1: Transmit-data-empty interrupt request (TXI) is enabled Note: * TXI interrupt requests can be cleared by reading the value 1 from the TDRE flag, then clearing it to 0; or by clearing the TIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
6	RIE	0	R/W	**Receive Interrupt Enable (RIE)** Enables or disables the receive-data-full interrupt (RXI) requested when the RDRF flag is set to 1 in SSR due to transfer of serial receive data from RSR to RDR; also enables or disables the receive-error interrupt (ERI). 0: Receive-data-full (RXI) and receive-error (ERI) interrupt requests are disabled 1: Receive-data-full (RXI) and receive-error (ERI) interrupt requests are enabled (When RDMA in TDR register = 0) Receive-error (ERI) interrupt request is enabled (When RDMA in TDR register = 1) Note: RXI and ERI interrupt requests can be cleared by reading the value 1 from the RDRF, FER, PER, or ORER flag, then clearing it to 0; or by clearing the RIE bit to 0.
5	TE	0	R/W	**Transmit Enable (TE)** Enables or disables the start of UART serial transmitting operations. 0: Transmitting disabled*1 1: Transmitting enabled*2 Notes: *1 The TDRE bit is locked at 1 in SSR. *2 In the enabled state, serial transmitting starts when the TDRE bit in SSR is cleared to 0 after writing of transmit data into TDR. Select the transmission format in SMR before setting the TE bit to '1'.
4	RE	0	R/W	**Receive Enable (RE)** Enables or disables the start of UART serial receiving operations. 0: Receiving disabled*1 1: Receiving enabled*2 Notes: *1 Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags. These flags retain their previous values. *2 In the enabled state, serial receiving starts when a start bit is detected in asynchronous mode. The reception format needs selecting in SMR before setting the RE bit to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	—	0	R	**Reserved** Keep to '0'
2	TEIE	0	R/W	**Transmit-End Interrupt Enable (TEIE)** Enables or disables the transmit-end interrupt (TEI) requested if TDR does not contain new transmit data when the MSB is transmitted. 0: Transmit-end interrupt requests (TEI) are disabled* 1: Transmit-end interrupt requests (TEI) are enabled* Note: * TEI interrupt requests can be cleared by reading the value 1 from the TDRE flag in SSR, then clearing the TDRE flag to 0, thereby also clearing the TEND flag to 0; or by clearing the TEIE bit to 0.
1, 0	—	0	R	**Reserved** Keep to '0'

Serial Status Register (SSR)

SSR is the register containing status flags that indicate UART operating status.

The CPU can always read and write SSR, but cannot write 1 in the TDRE, RDRF, ORER, PER, and FER flags. These flags can be cleared to 0 only if they have first been read while set to 1. The TEND is a read-only bit that cannot be written. SSR is initialised to H'84 by a reset and in standby mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									TDRE	RDRF	ORER	FER	PER	TEND		
Initial:	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R/WC0	R/WC0	R/WC0	R/WC0	R/WC0	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	0	R	**Reserved**
7	TDRE	1	R/WC0	**Transmit Data Register Empty (TDRE)** Indicates that the UART has loaded transmit data from TDR into TSR and the next serial transmit data can be written in TDR. 0: TDR contains valid transmit data. This bit can be cleared by software by writing '0' to this bit, but only after it has previously been read while set to '1'. This bit it is cleared automatically by a DMA write to the TDR register. 1: TDR does not contain valid transmit data. This bit is set by a chip reset, TE in the SCR being cleared to 0 or TDR contents are loaded into TSR, so new data can be written in TDR.
6	RDRF	0	R/WC0	**Receive Data Register Full (RDRF)** Indicates that RDR contains new receive data. 0: RDR does not contain new receive data. 1: RDR contains new receive data. This bit can be cleared by software by writing '0' to this bit, but only after it has previously been read while set to '1'. This bit it is cleared automatically by a DMA read from the RDR register. Note: The RDR contents and RDRF flag are not affected by detection of receive errors or by clearing of the RE bit to 0 in SCR. They retain their previous values. If the RDRF flag is still set to 1 when reception of the next data ends, an overrun error occurs and the data contained in the RSR register is not loaded into the RDR register. If further data is received the data held in the RSR will be overwritten.

Bit	Bit Name	Initial Value	R/W	Description
5	ORER	0	R/WC0	**Overrun Error (ORER)** Indicates that data reception ended abnormally due to an overrun error. 0: Receiving is in progress or has ended normally*1 1: A receive overrun error occurred*2 This bit is set by Reception of the next serial data ending when RDRF = 1. Notes: *1 Clearing the RE bit to 0 in SCR does not affect the ORER flag, which retains its previous value. *2 RDR continues to hold the receive data before the overrun error, so subsequent receive data is lost. Serial receiving cannot continue while the ORER flag is set to 1.
4	FER	0	R/WC0	**Framing Error (FER)** Indicates that data reception ended abnormally due to a framing error. 0: Receiving is in progress or has ended normally*1 1: A receive framing error occurred*2 This bit is set if the stop bit at the end of receive data is checked and found to be 0. Notes: *1 Clearing the RE bit to 0 in SCR does not affect the FER flag, which retains its previous value. *2 When the stop bit length is 2 bits, only the first bit is checked. The second stop bit is not checked. When a framing error occurs the UART transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the FER flag is set to 1.

Bit	Bit Name	Initial Value	R/W	Description
3	PER	0	R/WC0	**Parity Error (PER)** Indicates that data reception ended abnormally due to a parity error. 0: Receiving is in progress or has ended normally*[1] 1: A receive parity error occurred*[2] This bit is set by the number of '1's in receive data, including the parity bit, does not match the even or odd parity setting of OE in SMR. Notes: *1 Clearing the RE bit to 0 in SCR does not affect the PER flag, which retains its previous value. *2 When a parity error occurs the UART transfers the receive data into RDR but does not set the RDRF flag. Serial receiving cannot continue while the PER flag is set to 1.
2	TEND	1	R	**Transmit End (TEND)** Indicates that when the last bit of a serial character was transmitted TDR did not contain new transmit data, so transmission has ended. The TEND flag is a read-only bit and cannot be written. 0: Transmission is in progress. This bit is cleared automatically when data is written to the TDR, or when the TDRE flag is cleared. 1: End of transmission. The bit is set by a chip reset, or by the TE bit being cleared to 0 in SCR, or the TDRE is 1 when the last bit of a serial character is transmitted.
1, 0	—	0	R	**Reserved**

Bit Rate Register (BRR)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						BRR										
Initial:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	0	R	**Reserved**
10 to 0	BRR	1	R/W	

BRR is an 11-bit register that, together with the CKS1 and CKS0 bits in SMR, select the baud rate generator clock source, determines the serial communication bit rate. Bits 31 to 12 are reserved.

The CPU can always read and write BRR. BRR is initialised to H'7FF by a reset and in standby mode.

The BRR setting is calculated as follows:

$$BRR = \frac{\phi}{2 \times 2^{2n-1} \times S \times B} - 1$$

Where: B = Bit rate (bit/s)
BRR = BRR setting for baud rate generator ($0 \leq N \leq 2048$)
ϕ = Register bus clock frequency (Hz).
S = Bit Sample Rate.

Sample rate is set for the following modes to:

OSM bit in SMR = '0', S = 16.
OSM bit in SMR = '1', S = 8.

n = Baud rate generator clock source (n = 0, 1, 2, 3).

For the clock sources and values for n, see following table.

		SMR Settings	
n	Clock Source	CKS1	CKS0
0	φ	0	0
1	φ/4	0	1
2	φ/16	1	0
3	φ/64	1	1

The bit rate error is calculated as follows:

$$\text{Error}\ (\%) = \left(\frac{\phi}{(BRR + 1) \times B \times S \times 2 \times 2^{2n-1}} - 1 \right) \times 100$$

21.5 Functional Description

21.5.1 Overview

The UART has an asynchronous mode in which characters are synchronised individually.

21.5.2 Asynchronous Mode

- Data length is selectable: 7 or 8 bits.
- Parity and stop bits length (1 or 2 bits), are selectable. These selections determine the communication format and character length.
- In receiving, it is possible to detect framing errors, parity errors, overrun errors, and the break state.
- The UART operates using the on-chip baud rate generator.

Table 21.3 SMR Settings and Serial Communication Formats

SMR Settings			UART Communication Format		
Bit 6 CHR	**Bit 5 PE**	**Bit 3 STOP**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	8-bit data	Absent	1 bit
0	0	1			2 bits
0	1	0		Present	1 bit
0	1	1			2 bits
1	0	0	7-bit data	Absent	1 bit
1	0	1			2 bits
1	1	0		Present	1 bit
1	1	1			2 bits

21.5.3 Operation

Each transmitted or received character begins with a start bit and ends with a stop bit. Serial communication is synchronised one character at a time.

The transmitting and receiving sections of the UART are independent, so full duplex communication is possible. The transmitter and receiver are both double buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmitting and receiving.

Figure 21.3 shows the general format of asynchronous serial communication. In asynchronous serial communication the communication line is normally held in the mark (high) state. The UART monitors the line and starts serial communication when the line goes to the space (low) state, indicating a start bit. One serial character consists of a start bit (low), data (LSB first), parity bit (high or low), and stop bit (high), in that order.

When receiving, the UART synchronises at the falling edge of the start bit. The UART samples each data bit on the eighth pulse of a clock with a frequency 16 times the bit rate. If OSM bit in the Serial Mode Register is set to '1', the UART will samples each data bit on the fourth pulse of a clock with a frequency 8 times the bit rate. Receive data is therefore latched at the centre of each bit.

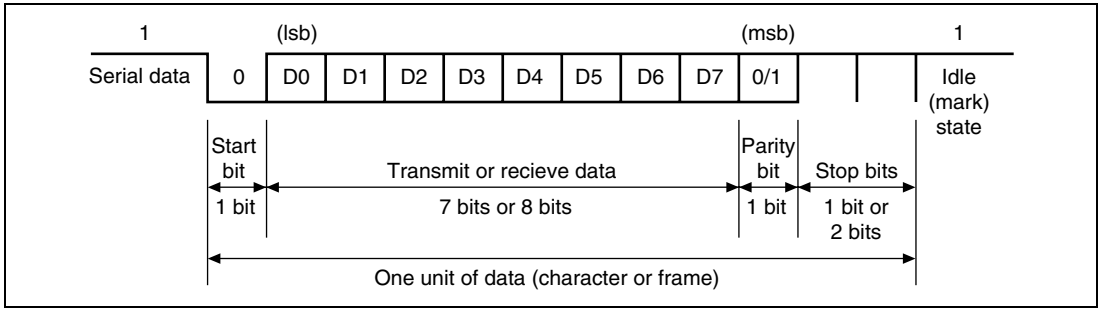


Figure 21.3 Data Format in Asynchronous Communication Serial Data.

21.5.4 Transmitting and Receiving Data

UART Initialisation

Before transmitting or receiving, clear the TE and RE bits to 0 in SCR, then initialise the UART as follows. When changing the communication mode or format, always clear the TE and RE bits to 0 before following the procedure given below. Clearing TE to 0 sets the TDRE flag to 1 and initialises TSR. Clearing RE to 0, however, does not initialise the RDRF, PER, FER, and ORER flags and RDR, which retain their previous contents.

Figure 21.4 is a sample flow chart for initialising the SCR

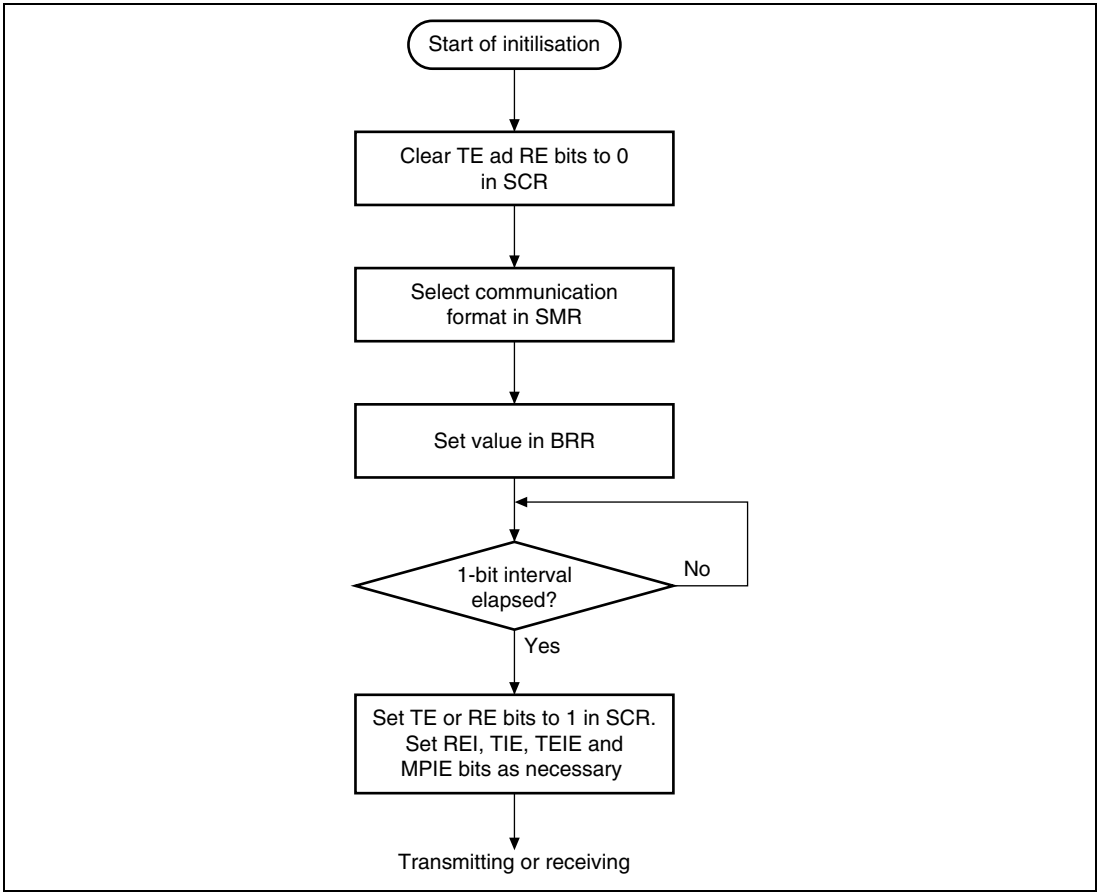


Figure 21.4 Sample Flowchart for UART Initialisation.

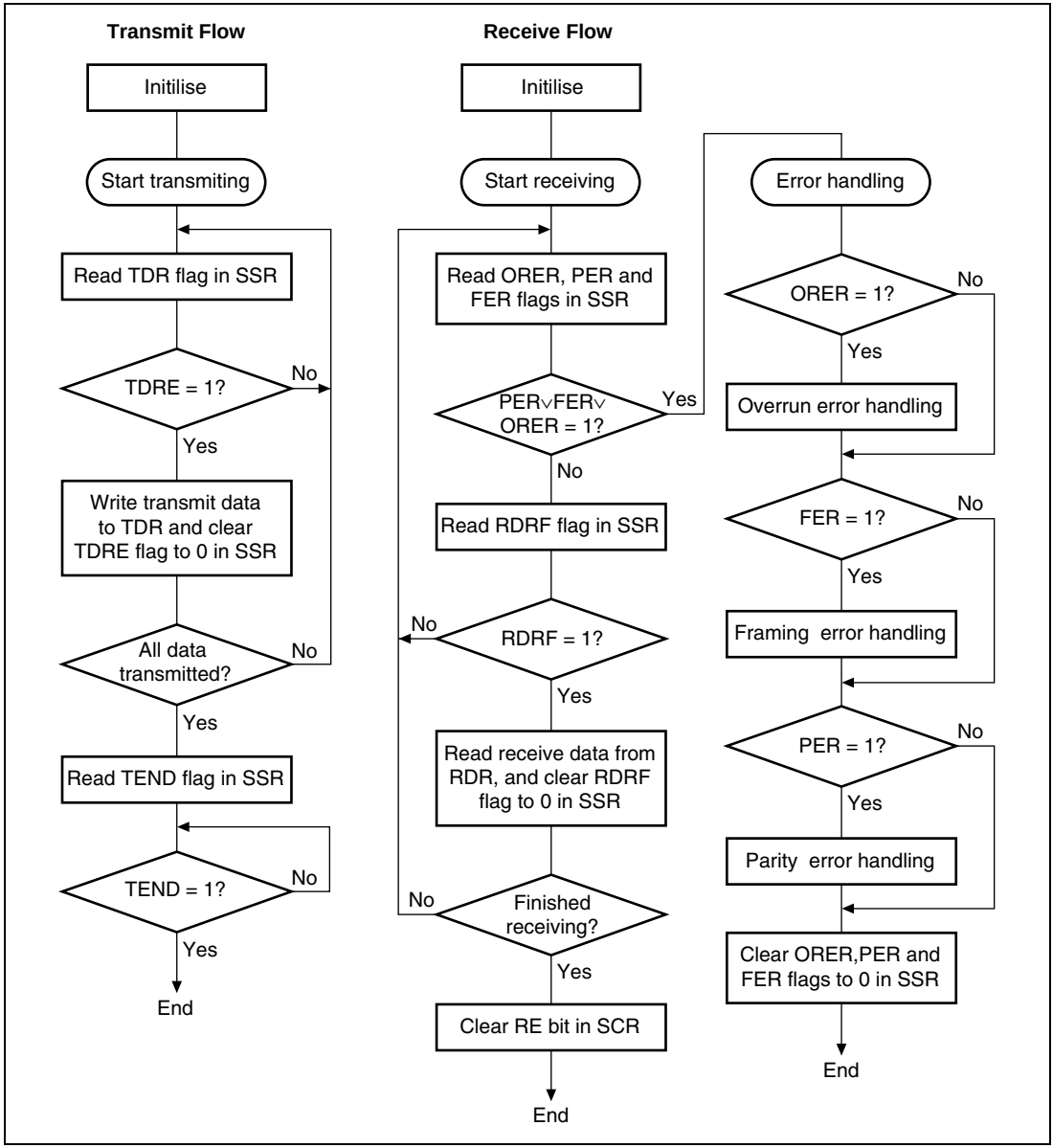


Figure 21.5 Sample Flowchart for Transmitting and Receiving Serial Data.

Transmitting Serial Data

Figure 21.5 shows a sample flowchart for transmitting serial data and indicates the procedure to follow.

In transmitting serial data, the UART operates as follows.

- The UART monitors the TDRE flag in SSR. When the TDRE flag is cleared to 0 the UART recognises that TDR contains new data, and loads this data from TDR into TSR.
- After loading the data from TDR into TSR, the UART sets the TDRE flag to 1 and starts transmitting. If the TIE bit is set to 1 in SCR, the UART requests a transmit-data-empty interrupt

  (TXI) at this time.

  Serial transmit data is transmitted in the following order from the UART_TXD pin:

  — Start bit: One 0 bit is output.

  — Transmit data: 7 or 8 bits are output, LSB first.

  — Parity bit: One parity bit (even or odd parity) is output. Formats in which a parity bit is not output can also be selected.

  — Stop bit: One or two 1 bits (stop bits) are output.

  — Mark state: Output of 1 bits continues until the start bit of the next transmit data.
- The UART checks the TDRE flag when it outputs the stop bit. If the TDRE flag is 0, the UART loads new data from TDR into TSR, outputs the stop bit, then begins serial transmission of the next frame. If the TDRE flag is 1, the UART sets the TEND flag to 1 in SSR, outputs the stop bit, then continues output of 1 bits in the mark state. If the TEIE bit is set to 1 in SCR, a transmit-end interrupt (TEI) is requested at this time.

Receiving Serial Data

Figure 21.5 shows a sample flowchart for receiving serial data and indicates the procedure to follow.

In receiving, the UART operates as follows.

- The UART monitors the receive data line. When it detects a start bit, the UART synchronises internally and starts receiving.
- Receive data is stored in RSR in order from LSB to MSB.
- The parity bit and stop bit are received.

  After receiving, the UART makes the following checks:

  — Parity check: The number of 1s in the receive data must match the even or odd parity setting of the OE bit in SMR.

  — Stop bit check: The stop bit value must be 1. If there are two stop bits, only the first stop bit is checked.

  — Status check: The RDRF flag must be 0 so that receive data can be transferred from RSR into RDR.

  If these checks all pass, the RDRF flag is set to 1 and the received data is stored in RDR. Check the RDRF flag is set to 1, then read the receive data from RDR and clear the RDRF flag to 0. To continue to receive data, the RDR data must be read and the RDRF flag cleared before the stop bit of the current frame is received.

  If one of the checks fails (receive error), the UART operates as indicated in Table 21.4.

Note: When a receive error occurs, further receiving is disabled. In receiving, the RDRF flag is not set to 1. Be sure to clear the error flags to 0.

- When the RDRF flag is set to 1, if the RIE bit is set to 1 in SCR, a receive-data-full interrupt (RXI) is requested. If the ORER, PER, or FER flag is set to 1 and the RIE bit in SCR is also set to 1, a receive-error interrupt (ERI) is requested. If the DMAC is activated by an RXI interrupt to read the RDR value, the RDRF flag is cleared automatically.

Table 21.4 Receive Error Conditions.

Receive Error	Abbreviation	Condition	Data transfer
Overrun error	ORER	Receiving of next data ends while RDRF flag is still set to 1 in SSR	Receive data not transferred from RSR to RDR
Framing Error	FER	Stop bit is 0	Receive data transferred from RSR to RDR
Parity Error	PER	Parity of receive data differs from odd/even parity setting in SMR	Receive data transferred from RSR to RDR

Use a byte in longword format, I described below, and extract bytes from longwords in memory.

Byte data extraction should be done in driver level so that DMAC always transfer longword for income data. i.e. only the least significant byte is always valid in each longword data.

	1st byte	2nd byte	3rd byte
Serial	0xAA	0xBB	0xCC
RB Bus	0x000000AA	0x000000BB	0x000000CC
FIFO	0x000000AA	0x000000BB	0x000000CC
Memory	0x000000AA	0x000000BB	0x000000CC
User *	0xAA	0xBB	0xCC

Note: * valid data extracted from driver buffer to user space by driver.

21.5.5 Reset Strategy

All registers will be equipped with an asynchronous reset.

21.5.6 Standby mode

The UART module allows clock gating to reduce power consumption. This module standby mode can be executed by controlling Clock Control1 (CC1) Register in Power Control module. The associated bits in cc1 register for UART channel 0, 1, 2, and 3, are bit 9, bit 10, bit 11and bit 12 respectively.

To wake up one of the UART channels, the associated bit in cc1 register must be enabled. After enabling this bit, all access to that channel can be possible.

To power down one of the UART channels, the following procedure is required using the associated register for that channel:

1. Wait until Transmit Data Register Empty (TDRE) and Transmit End (TEND) are '1' to warranty full transmission of pending bytes in Transmit Data Register (TDR) and Transmit Shift Register (TSR).
2. Disable Module Enable (ME) bit in Serial Control Register.
3. Disable the associated bit in Clock Control 1 (CC1) Register in Power Control Module.

22.1 General Description

The IRDA interface module is a small extension to the UART module specification to enable RXD and TXD pins to connect to a 115.2 Kbp/s IrDA transceiver device. This specification will only describe the UART's additional features that are required to connect it to a IrDA device, and will describe how to set up the UART to use this mode. All details on how to use the UART should be gained from the UART Module Block Specification.

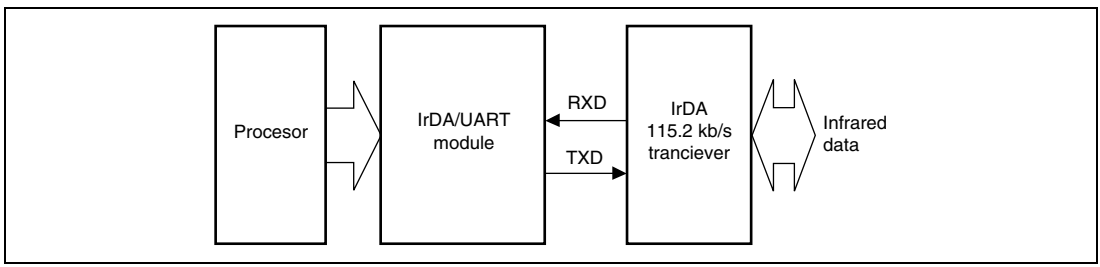


Figure 22.1 Overview Block Diagram

22.2 Features

This module enables simultaneous transmit and receive of data from an IrDA transceiver. Receive and Transmit Registers are double buffered to enable continuous transmission. All features of the UART device are also available within this module.

22.3 Block Diagram

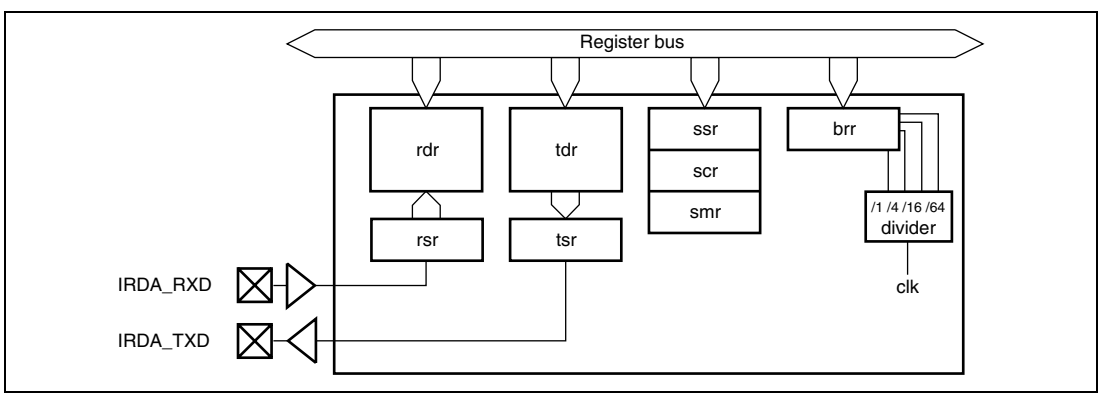


Figure 22.2 Functional Block Diagram

22.4 Interfaces

22.4.1 Digital Inputs/Outputs

The following table lists the digital interface pins and their functions:

Table 22.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From	Synchronization to Clocks
IRDA_RXD	1	In	Receive data	External Irda	rbclk after baud rate divider
IRDA_TXD	1	Out	Transmit data	External Irda	rbclk after baud rate divider

22.4.2 Software Interfaces

The registers accessible by the software are listed in the following table:

All register addresses below are long word addresses only.

Table 22.2 Register List

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6620	Serial Mode Register 0	SMR0	32
H'6624	Bit Rate Register 0	BRR0	32
H'6628	Serial Control Register 0	SCR0	32
H'662C	Transmit Data Register 0	TDR0	32
H'6630	Serial Status Register 0	SSR0	32
H'6634	Receive Data Register 0	RDR0	32
H'6638	Reserved		
H'663C	IrDA Control Register 0	ICR	32

For Details on all registers except IrDA Mode register please reference the UART Module Block Specification.

All Reserved or unused bits do not have a guaranteed value when read.

Legends for register description:

Initial value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

22.4.3 IrDA Control Register 0 (ICR0)

ICR is a 4-bit register that specifies the IrDA receive and transmit configuration.

The CPU can always read and write to ICR. ICR is initialised to H'00 by a reset.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													IBR			IME
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	0	R	**Reserved**
3 to 1	IBR	0	R/W	**IrDA Baud Rate (IBR)** These bits select the signaling rate to be used for IrDA transmission. This register is used to divide down the BRR clock to use the lower IrDA bit rates. Set the BRR clock to 115.2 Kbit/s or 57.6 Kbit/s. Refer to the clock table for more details. IrDA Signaling Rate 000: IrDA Baud Rate = BRR 001: = BRR/2 010: = BRR/3 011: = BRR/6 100: = BRR/12 101: = BRR/48
0	IME	0	R/W	**IrDA Mode Enable (IME)** Selects whether IrDA mode is enabled. 0: IrDA mode disabled. 1: IrDA mode is enabled.

Note: When IrDA Baud Rate is changed, please set the IME bit to 0 (IrDA mode disabled).

22.5 Functional Description

22.5.1 Overview

The IrDA module can be used to transmit and receive data from an IrDA transceiver at a range of frequencies up to 115.2 Kbit/s. For further details on how to configure this device for UART transmissions, please refer to the UART module block specification.

22.5.2 IrDA Mode Register Settings.

For the module to operate correctly with the IrDA transceiver it is essential that the SMR register outlined in the UART module block specification is set correctly.

IrDA data is transmitted only in asynchronous mode, with 8 data bits and 1 stop bits. No parity bits are transmitted. See Figure 22.3.

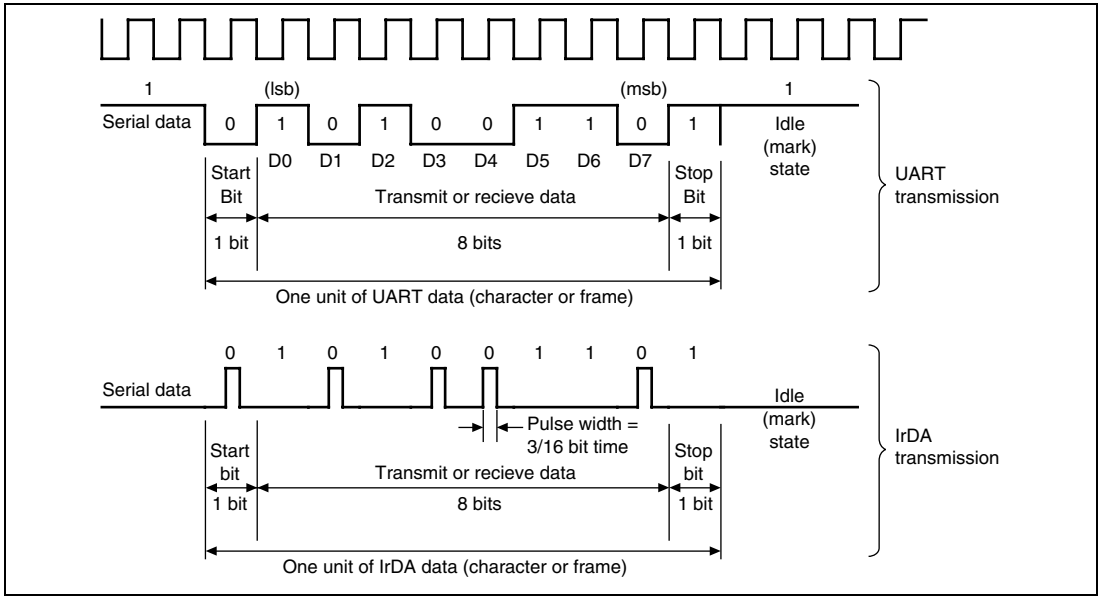


Figure 22.3 IrDA Transmit/Receive Timing Diagram

Register configurations to achieve the above timing can be found in Table 22.3 and this register should always be configured in this way when the ICR bit is set to 1 (IrDA mode enabled).

Table 22.3 SMR Settings and Serial Communication Formats

SMR Settings				SCI Communication Format		
Bit 9 OSM	**Bit 6 CHR**	**Bit 5 PE**	**Bit 3 STOP**	**Data Length**	**Parity Bit**	**Stop Bit Length**
0	0	0	0	8-bit data	Absent	1 bit

All other registers can be used as outlined in the UART module block specification.

22.5.3 BRR Setting

The Baud Rate Register should be set according to the equation below. For IrDA mode operations, the BRR should be calculated for a bit rates of 115.2 Kbit/s or 57.6 Kbit/s, regardless of whether 115.2 Kbit/s or 57.6 Kbit/s, or a lower frequency is used. To select a lower frequency the setting of IrDA Register, IrDA Baud Rate bits are used.

The BRR setting is calculated as follows:

$$BRR = \frac{\phi}{S \times 2 \times 2^{2n-1} \times B} - 1$$

Where: B = Bit rate (bits/s) = 115.2×10^3 or 57.6 Kbit/s in IrDA mode.
BRR = BRR setting for baud rate generator ($0 \leq N \leq 2048$)
ϕ = System clock frequency (Hz).
S = over-sample rate = 32 for IrDA mode.
n = Baud rate generator clock source (n = 0, 1, 2, 3).

For the clock sources and values for *n*, see following table

		SMR Settings	
n	**Clock Source**	**CKS1**	**CKS0**
0	ϕ	0	0
1	$\phi/4$	0	1
2	$\phi/16$	1	0
3	$\phi/64$	1	1

For example using the Register bus clock = 33MHz = ϕ

IrDA baud rate = 115.2 Kbit/s, $\therefore S = 32$

Use $n = 0$ $\therefore 2^{2n-1} = 0.5$.

$$BRR = \frac{33 \times 10^6}{32 \times 2 \times 0.5 \times 115.2 \times 10^3} - 1 = 7.95$$

BRR setting = 8, error = 0.54%

Table 22.4 IrDA BRR setting value and Error

ϕ Register bus clock (MHz)	n	B	Calculated BRR value	BRR setting value of Bit Rate Register	Freq (Hz)	Error (%)	IBR of ICR0	
							B = 115.2k	B = 57.6k
31	0	57600	15.8	16	56985	-1.0672	—	1
31	0	28800		16		-1.0672	—	2
31	0	19200		16		-1.0672	—	3
31	0	9600		16		-1.0672	—	6
31	0	4800		16		-1.0672	—	12
33	0	115200	7.95	8	114583	-0.5353	1	—
33	0	57600		8		-0.5353	2	—
33	0	38400		8		-0.5353	3	—
33	0	19200		8		-0.5353	6	—
33	0	9600		8		-0.5353	12	—
33	0	2400		8		-0.5353	48	—
33	0	57600	16.9	17	57291	-0.5353	—	1
33	0	28800		17		-0.5353	—	2
33	0	19200		17		-0.5353	—	3
33	0	9600		17		-0.5353	—	6
33	0	4800		17		-0.5353	—	12
35	0	57600	18	18	57565	-0.0594	—	1
35	0	28800		18		-0.0594	—	2
35	0	19200		18		-0.0594	—	3
35	0	9600		18		-0.0594	—	6
35	0	4800		18		-0.0594	—	12
37	0	115200	9.04	9	115625	0.3689	1	—
37	0	57600		9		0.3689	2	—
37	0	38400		9		0.3689	3	—
37	0	19200		9		0.3689	6	—
37	0	9600		9		0.3689	12	—
37	0	2400		9		0.3689	48	—
39	0	57600	20.2	20	58035	0.7564	—	1
39	0	28800		20		0.7564	—	2
39	0	19200		20		0.7564	—	3
39	0	9600		20		0.7564	—	6
39	0	4800		20		0.7564	—	12
42	0	57600	21.8	22	57065	-0.9284	—	1
42	0	28800		22		-0.9284	—	2
42	0	19200		22		-0.9284	—	3
42	0	9600		22		-0.9284	—	6

ϕ Register bus clock (MHz)	n	B	Calculated BRR value	BRR setting value of Bit Rate Register	Freq (Hz)	Error (%)	IBR of ICR0	
							B = 115.2k	B = 57.6k
42	0	4800		22		-0.9284	—	12
44	0	115200	11.0	11	114583	-0.5353	1	—
44	0	57600		11		-0.5353	2	—
44	0	38400		11		-0.5353	3	—
44	0	19200		11		-0.5353	6	—
44	0	9600		11		-0.5353	12	—
44	0	2400		11		-0.5353	48	—
44	0	57600	22.9	23	57291	-0.5353	—	1
44	0	28800		23		-0.5353	—	2
44	0	19200		23		-0.5353	—	3
44	0	9600		23		-0.5353	—	6
44	0	4800		23		-0.5353	—	12
46	0	57600	24.0	24	57500	-0.1736	—	1
46	0	28800		24		-0.1736	—	2
46	0	19200		24		-0.1736	—	3
46	0	9600		24		-0.1736	—	6
46	0	4800		24		-0.1736	—	12
48	0	115200	12.0	12	115384	0.1603	1	—
48	0	57600		12		0.1603	2	—
48	0	38400		12		0.1603	3	—
48	0	19200		12		0.1603	6	—
48	0	9600		12		0.1603	12	—
48	0	2400		12		0.1603	48	—
48	0	57600	25.0	25	57692	0.1603	—	1
48	0	28800		25		0.1603	—	2
48	0	19200		25		0.1603	—	3
48	0	9600		25		0.1603	—	6
48	0	4800		25		0.1603	—	12
50	0	57600	26.1	26	57870	0.4694	—	1
50	0	28800		26		0.4694	—	2
50	0	19200		26		0.4694	—	3
50	0	9600		26		0.4694	—	6
50	0	4800		26		0.4694	—	12
	: IrDA available setting (within standard spec.)							

Note: Baud rate tolerance must be less than +/- 0.87%. Please refer to Infrared Data Association Serial Infrared Physical Layer Specification Version 1.3

22.5.4 Reset Strategy

All registers will be equipped with an asynchronous reset.

22.5.5 Standby mode

The standby mode does not vary respect to the normal operation with the UART module.

Section 23 USB Function

23.1 Features

- Incorporates UDC (USB device controller) supporting USB1.1

  Automatic processing of USB protocol

  Automatic processing of USB standard commands for endpoint 0 (some commands and class/vendor commands require decoding and processing by firmware)
- Transfer speed: Full-speed
- Endpoint configuration

Endpoint Name	Abbreviation	Transfer Type	Maximum Packet Size	FIFO Buffer Capacity (Byte)	DMA Transfer
Endpoint 0	EP0s	Setup	8	8	—
	EP0i	Control-in	8	8	—
	EP0o	Control-out	8	8	—
Endpoint 1	EP1	Bulk-out	64	128	Possible
Endpoint 2	EP2	Bulk-in	64	128	Possible
Endpoint 3	EP3	Interrupt-in	8	8	—

- Interrupt requests: generates various interrupt signals necessary for USB transmission/reception
- Power-down mode

  Power consumption can be reduced by stopping internal clock when UDC cable is disconnected

  Automatic transition to/recovery from suspend state
- In case the Function Module is used, the Host's ConfigurationControl.Port2Switch should be set to 1.

Notes: 1. The USB function includes only a register bus inside HD64404. This bus is used to access or transfer/receive data to/from USB Function Control Registers.

2. When the function module and the host module are switched by Port2Switch, if USB2PENC pin is used, please keep the procedure below.

Situation	Procedure
Switch to Function Module	Two cases can be selected as follows.
(This module is selected in USB Host as default value after power on sequence)	Set Port2Switch to "1" after that set USB Function Module registers
	Set Port2Switch to "1" after that set PULLUPE bit in USBDMAR to "1". Next, set PULLUPE bit to "0" after that set USB Function Module registers.

23.2 Block Diagram

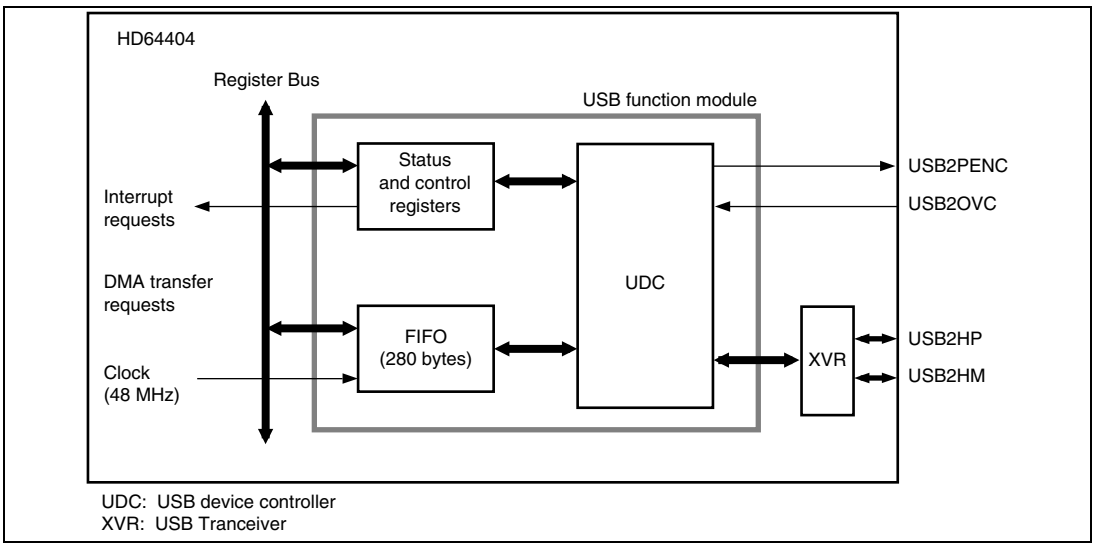


Figure 23.1 Block Diagram of UBC

23.3 Pin Description

Table 23.1 Pin Configuration and Functions

Pin Name	I/O	Function	Polarity
USB2HP	Bidirect	Input pin for D+ signal from receiver	—
USB2HM	Bidirect	Input pin for D– signal from receiver	—
USB2OVC	Input	USB cable connection monitor pin	High Active
USB2PENC	Output	USB D+ PullUp Enable	Low Active

23.4 Register Configuration

Table 23.2 USB Function Module Registers

Name	Abbreviation	R/W	Initial Value	Address	Access Size (Bits Size)
Interrupt Flag Register 0	USBIFR0	R/W	H'10	H'5C00	32
Interrupt Flag Register 1	USBIFR1	R/W	H'00	H'5C04	32
EP0i Data Register	USBEPDR0I	W	—	H'5C08	32
EP0o Data Register	USBEPDR0O	R	H'00	H'5C0C	32
Trigger Register	USBTRG	W	—	H'5C10	32
FIFO Clear Register	USBFCLR	W	—	H'5C14	32
USBEP0o Receive Data Size Register	USBEPSZ0O	R	H'00	H'5C18	32
EP0s Data Register	USBEPDR0S	R	H'00	H'5C1C	32
Data Status Register	USBDASTS	R	H'00	H'5C20	32
EP2 Data Register	USBEPDR2	W	—	H'5C24	32
Reserved	—	—	—	H'5C28	32
Endpoint Stall Register	USBEPSTL	R/W	H'00	H'5C2C	32
Interrupt Enable Register 0	USBIER0	R/W	H'00	H'5C30	32
Interrupt Enable Register 1	USBIER1	R/W	H'00	H'5C34	32
EP1 Data Register	USBEPDR1	R	H'00	H'5C38	32
EP1 Receive Data Size Register	USBEPSZ1	R	H'00	H'5C3C	32
Reserved	—	—	—	H'5C40	32
DMA Setting Resister	USBDMA	R/W	H'00	H'5C44	32
EP3 Data Register	USBEPDR3	W	—	H'5C48	32

Notes: These registers can be set up when 48MHz clock is supplied. It needs to wait for USB_Xtal to be oscilated for the time that is specified in Electrical Specification. Additionally these registers cannot be set up by a maximum of 6 microseconds from setting up the Function bit of XTAL_Control[0] register in the Power Control & Configuration module, 48MHz-clock supply is controllable.

23.5 Register Descriptions

Legends for register description:

Initial Value	: Register value after reset
—	: Read → undefined value, Write → always "0" write
– –	: Read → undefined value
*	: Value is retained
R/W	: Read and Write register
R	: Read only register, for write always 0 write
W	: Write only resister

23.5.1 EP0i Data Register (USBEPDR0I)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPDR0I[7:0]							
Initial:	-	-	-	-	-	-	-	-	*	*	*	*	*	*	*	*
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	W	Reserved
7 to 0	USBEPDR0I[7:0]	*	W	USBEPDR0I is an 8-byte FIFO buffer for endpoint 0, holding one packet of transmit data for control-in. Transmit data is fixed by writing one packet of data and setting bit 0 in the USB Trigger Register. When an ACK handshake is returned from the host after the data has been transmitted, bit 0 in USB interrupt Flag Register 0 is set. This FIFO buffer can be initialized by means of bit 0 in the USBFIFO Clear Register.

23.5.2 EP0o Data Register (USBEPDR0O)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPDR0O[7:0]							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	Reserved
7 to 0	UUSBEPDR0O[7:0]	0	R	USBEPDR0O is an 8-byte receive FIFO buffer for endpoint 0. USBEPDR0O holds endpoint 0 receive data other than setup commands. When data is received normally, bit 2 in USB interrupt flag register 0 is set, and the number of receive bytes is indicated in the EP0o Receive Data Size Register. After the data has been read, setting bit 1 in the USB Trigger Register enables the next packet to be received. This FIFO buffer can be initialized by means of bit 1 in the USBFIFO Clear Register.

23.5.3 EP0s Data Register (USBEPDR0S)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPDR0S [7:0]							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	Reserved
7 to 0	USBEPDR0S[7:0]	0	R	USBEPDR0S is an 8-byte FIFO buffer specifically for endpoint 0 setup command reception. USBEPDR0S receives only setup commands requiring processing on the application side. When command data is received normally, bit 3 in USB Interrupt Flag Register 0 is set. As a setup command must be received without fail, if data is left in this buffer, it will be overwritten with new data. If reception of the next command is started while the current command is being read, command reception has priority, the read by the application is forcibly terminated, and the read data is invalid.

23.5.4 EP1 Data Register (USBEPDR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPDR1 [7:0]							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	Reserved
7 to 0	USBEPDR1[7:0]	0	R	USBEPDR1 is a 128-byte receive FIFO buffer for endpoint 1. USBEPDR1 has a dual-FIFO configuration, and has a capacity of twice the maximum packet size. When one packet of data is received normally from the host, bit 6 in USB Interrupt Flag Register 0 is set. The number of receive bytes is indicated in the USBEP1 Receive Data Size Register. After the data has been read, the buffer that was read is enabled to receive again by writing 1 to bit 5 in the USB Trigger Register. The receive data in this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of bit 1 in the USBFIFO Clear Register.

23.5.5 EP2 Data Register (USBEPDR2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPDR2 [7:0]							
Initial:	-	-	-	-	-	-	-	-	*	*	*	*	*	*	*	*
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	W	Reserved
7 to 0	USBEPDR2[7:0]	*	W	USBEPDR2 is a 128-byte transmit FIFO buffer for endpoint 2. USBEPDR2 has a dual-buffer configuration, and has a capacity of twice the maximum packet size. When transmit data is written to this FIFO buffer and bit 4 in the USB Trigger Register is set, one packet of transmit data is fixed, and the dual-FIFO buffer is switched over. Transmit data for this FIFO buffer can be transferred by DMA. This FIFO buffer can be initialized by means of bit 4 in the USBFIFO Clear Register.

23.5.6 EP3 Data Register (USBEPDR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPDR3 [7:0]							
Initial:	-	-	-	-	-	-	-	-	*	*	*	*	*	*	*	*
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	W	Reserved
7 to 0	USBEPDR3[7:0]	*	W	USBEPDR3 is an 8-byte transmit FIFO buffer for endpoint 3, holding one packet of transmit data in endpoint 3 interrupt transfer. Transmit data is fixed by writing one packet of data and setting bit 6 in the USB Trigger Register. When an ACK handshake is received from the host after one packet of data has been transmitted normally, bit 1 in the USB Interrupt Flag Register is set. This FIFO buffer can be initialized by means of bit 6 in the USBFCLR Register.

23.5.7 Interrupt Flag Register 0 (USBIFR0)

Together with USB Interrupt Flag Register 1, USBIFR0 indicates interrupt status information required by the application. When an interrupt source occurs, the corresponding bit is set to 1 and an interrupt request is sent to the CPU according to the combination with USB Interrupt Enable Register 0. Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits. However, bits 6 and 4 are status bits, and cannot be cleared.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BRST	EP1 FULL	EP2 TR	EP2 EMPTY	SETUP TS	EP0o TS	EP0i TR	EP0i TS
Initial:	-	-	-	-	-	-	-	-	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/ WC0	R	R/ WC0	R	R/ WC0	R/ WC0	R/ WC0	R/ WC0

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	**Reserved**
7	BRST	0	R/WC0	**Bus Reset (BRST)** Set to 1 when the bus reset signal is detected on the USB bus.
6	EP1FULL	0	R	**EP1 FIFO Full (EP1 FULL)** This bit is set when endpoint 1 receives one packet of data normally from the host, and holds a value of 1 as long as there is valid data in the FIFO buffer. EP1 FULL is a status bit, and cannot be cleared.
5	EP2TR	0	R/WC0	**EP2 Transfer Request (EP2 TR)** This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 2 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.

Bit	Bit Name	Initial Value	R/W	Description
4	EP2EMPTY	1	R	**EP2 FIFO Empty (EP2 EMPTY)** This bit is set when at least one of the dual endpoint 2 transmit FIFO buffers is ready for transmit data to be written. EP2 EMPTY is a status bit, and cannot be cleared.
3	SETUPTS	0	R/WC0	**Setup Command Receive Complete (SETUP TS)** This bit is set to 1 when endpoint 0 receives normally a setup command requiring decoding on the application side, and returns an ACK handshake to the host.
2	EP0oTS	0	R/WC0	**EP0o Receive Complete (EP0o TS)** This bit is set to 1 when endpoint 0 receives data from the host normally, stores the data in the FIFO buffer, and returns an ACK handshake to the host.
1	EP0iTR	0	R/WC0	**EP0i Transfer Request (EP0i TR)** This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 0 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
0	EP0iTS	0	R/WC0	**EP0i Transmit Complete (EP0i TS)** This bit is set when data is transmitted to the host from endpoint 0 and an ACK handshake is returned.

23.5.8 Interrupt Flag Register 1 (USBIFR1)

Together with USB Interrupt Flag Register 0, USBIFR1 indicates interrupt status information required by the application. When an interrupt source occurs, the corresponding bit(EP3TR or EP3TS or VBUSF) is set to 1 and an interrupt request is sent to the CPU according to the combination with USB Interrupt Enable Register 1. Clearing is performed by writing 0 to the bit to be cleared, and 1 to the other bits.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													VBUSMN	EP3TR	EP3TS	VBUSF
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC0	R/WC0	R/WC0

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	—	R	**Reserved**
3	VBUSMN	0	R	**USB Bus Connect Status (VBUSMN)** This bit has the same value of USB2OVC pin.
2	EP3TR	0	R/WC0	**EP3 Transfer Request (EP3 TR)** This bit is set if there is no valid transmit data in the FIFO buffer when an IN token for endpoint 3 is received from the host. A NACK handshake is returned to the host until data is written to the FIFO buffer and packet transmission is enabled.
1	EP3TS	0	R/WC0	**EP3 Transmit Complete (EP3 TS)** This bit is set when data is transmitted to the host from endpoint 3 and an ACK handshake is returned.
0	VBUSF	0	R/WC0	**USB Bus Connect (VBUSF)** This bit is set by a rising and falling edge at the USB2OVC. By connecting the VBUS monitor signal to the USB2OVC, an interrupt request can be sent to the CPU when power is supplied to the VBUS. The USB2OVC must be connected, as it is needed inside the module.

23.5.9 Trigger Register (USBTRG)

USBTRG generates one-shot triggers to control the transmit/receive sequence for each endpoint.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										EP3 PKTE	EP1 RDFN	EP2 PKTE		PE0s RDFN	EP0o RDFN	EP0i PKTE
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	W	**Reserved**
6	EP3PKTE	—	W	**EP3 Packet Enable (EP3 PKTE)** After one packet of data has been written to the endpoint 3 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.
5	EP1RDFN	—	W	**EP1 Read Complete (EP1 RDFN)** Write 1 to this bit after one packet of data has been read from the endpoint 1 FIFO buffer. The endpoint 1 receive FIFO buffer has a dual-FIFO configuration. Writing 1 to this bit initializes the FIFO that was read, enabling the next packet to be received.
4	EP2PKTE	—	W	**Endpoint 2 Packet Enable (EP2 PKTE)** After one packet of data has been written to the endpoint 2 FIFO buffer, the transmit data is fixed by writing 1 to this bit.
3	—	—	W	**Reserved**
2	EP0sRDFN	—	W	**EP0s Read Complete (EP0s RDFN)** Write 1 to this bit after EP0s command FIFO data has been read. Writing 1 to this bit enables transmission/reception of data in the following data stage. A NACK handshake is returned in response to transmit/receive requests from the host in the data stage until 1 is written to this bit.

Bit	Bit Name	Initial Value	R/W	Description
1	EP0oRDFN	—	W	**EP0o Read Complete (EP0o RDFN)** Writing 1 to this bit after one packet of data has been read from the endpoint 0 transmit FIFO buffer initializes the FIFO buffer, enabling the next packet to be received.
0	EP0iPKTE	—	W	**EP0i Packet Enable (EP0i PKTE)** After one packet of data has been written to the endpoint 0 transmit FIFO buffer, the transmit data is fixed by writing 1 to this bit.

23.5.10 FIFO Clear Register (USBFCLR)

USBFCLR is provided to initialize the FIFO buffers for each endpoint. Writing 1 to a bit clears all the data in the corresponding FIFO buffer. The corresponding interrupt flag is not cleared. Do not clear a FIFO buffer during transmission/reception.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										EP3 CLR	EP1 CLR	EP2 CLR			EP0o CLR	EP0i CLR
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	—	W	**Reserved**
6	EP3CLR	—	W	**EP3 Clear (EP3 CLR)** When 1 is written to this bit, the endpoint 3 transmit FIFO buffer is initialized.
5	EP1CLR	—	W	**EP1 Clear (EP1 CLR)** When 1 is written to this bit, both FIFOs in the endpoint 1 receive FIFO buffer are initialized.
4	EP2CLR	—	W	**EP2 Clear (EP2 CLR)** When 1 is written to this bit, both FIFOs in the endpoint 2 transmit FIFO buffer are initialized.

Bit	Bit Name	Initial Value	R/W	Description
3, 2	—	—	W	**Reserved**
1	EP0oCLR	—	W	**EP0o Clear (EP0o CLR)** When 1 is written to this bit, the endpoint 0 receive FIFO buffer is initialized.
0	EP0iCLR	—	W	**EP0i Clear (EP0i CLR)** When 1 is written to this bit, the endpoint 0 transmit FIFO buffer is initialized.

23.5.11 EP0o Receive Data Size Register (USBEPSZ0O)

USBEPSZ0O indicates, in bytes, the amount of data received from the host by endpoint 0.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPSZ0O [7:0]							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	**Reserved**
7 to 0	USBEPSZ0O[7:0]	0	R	

23.5.12 Data Status Register (USBDASTS)

USBDASTS indicates whether the transmit FIFO buffers contain valid data. A bit is set when data is written to the corresponding FIFO buffer and the packet enable state is set, and cleared when all data has been transmitted to the host.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											EP3 DE	EP2 DE				EP0i DE
Initial:	-	-	-	-	-	-	-	-	-	-	0	0	-	-	-	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 6	—	—	R	**Reserved**
5	EP3DE	0	R	**EP3 Data Present (EP3 DE)** This bit is set when the endpoint 3 FIFO buffer contains valid data.
4	EP2DE	0	R	**EP2 Data Present (EP2 DE)** This bit is set when the endpoint 2 FIFO buffer contains valid data.
3 to 1	—	—	R	**Reserved**
0	EP0iDE	0	R	**EP0i Data Present (EP0i DE)** This bit is set when the endpoint 0 FIFO buffer contains valid data.

23.5.13 Endpoint Stall Register (USBEPSTL)

The bits in USBEPSTL are used to forcibly stall the endpoints on the application side. While a bit is set to 1, the corresponding endpoint returns a stall handshake to the host. The stall bit for endpoint 0 (EP0 STL) is cleared automatically on reception of 8-bit command data for which decoding is performed by the function. When the SETUPTS flag in IFR0 is set, a write of 1 to the EP0 STL bit is ignored. For details see section 23.8, Stall Operations.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													EP3 STL	EP2 STL	EP1 STL	EP0 STL
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 4	—	—	R	**Reserved**
7 to 4	—	0	R	**Reserved**
3	EP3STL	0	R/W	**EP3 Stall (EP3 STL)** When this bit is set to 1, endpoint 3 is placed in the stall state.
2	EP2STL	0	R/W	**EP2 Stall (EP2 STL)** When this bit is set to 1, endpoint 2 is placed in the stall state.
1	EP1STL	0	R/W	**EP1 Stall (EP1 STL)** When this bit is set to 1, endpoint 1 is placed in the stall state.
0	EP0STL	0	R/W	**EP0 Stall (EP0 STL)** When this bit is set to 1, endpoint 0 is placed in the stall state.

23.5.14 Interrupt Enable Register 0 (USBIER0)

USBIER0 enables the interrupt requests indicated in Interrupt Flag Register 0 (USBIFR0). When an interrupt flag is set while the corresponding bit in USBIER0 is set to 1, an interrupt request is sent to the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									BRST	EP1 FULL	EP2 TR	EP2 TMPTY	SETUP TS	EP0o TS	EP0i TS	EP0i TS
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	**Reserved**
7	BRST	0	R/W	
6	EP1FULL	0	R/W	
5	EP2TR	0	R/W	
4	EP2EMPTY	0	R/W	
3	SETUPTS	0	R/W	
2	EP0oTS	0	R/W	
1	EP0iTS	0	R/W	
0	EP0iTS	0	R/W	

23.5.15 Interrupt Enable Register 1 (USBIER1)

USBIER1 enables the interrupt requests indicated in Interrupt Flag Register 1 (USBIFR1). When an interrupt flag is set while the corresponding bit in USBIER1 is set to 1, an interrupt request is sent to the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														EP3TR	EP3TS	VBUS
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	**Reserved**
7 to 3	—	0	R	**Reserved**
2	EP3TR	0	R/W	
1	EP3TS	0	R/W	
0	VBUS	0	R/W	

23.5.16 EP1 Receive Data Size Register (USBEPSZ1)

USBEPSZ1 is the endpoint 1 receive Data Size Register, indicating the amount of data received from the host. The endpoint 1 FIFO buffer has a dual-FIFO configuration; the receive data size indicated by this register refers to the currently selected FIFO.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									USBEPSZ1 [7:0]							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	**Reserved**
7 to 0	USBEPSZ1[7:0]	0	R	**Reserved**

23.5.17 DMA Setting Register (USBDMAR)

DMA transfer can be carried out between the Endpoint 1 and Endpoint 2 Data Registers by means of the on-chip DMA controller. Dual address transfer is performed, using byte transfer units. In order to start DMA transfer, DMA control settings must be made in addition to the settings in this register.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														PULL UPE	EP2 DMAE	EP1 DMAE
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	—	R	**Reserved**
2	PULLUPE	0	R	**D+ Pull Up Enable (PULLUPE):** When this bit is set to "1", USB2PENC is "HIGH" output. When this bit is set to "0", USB2PENC is "LOW" output. This bit is used to pull up control for D+ signal. 0: Power on (default) 1: Power off
1	EP2DMAE	0	R/W	**Endpoint 2 DMA Transfer Enable (EP2 DMAE):** When this bit is set, DMA transfer is enabled from memory to the endpoint 2 transmit FIFO buffer. If there is at least one byte of space in the FIFO buffer, set the number of transfer bytes in the DMA controller and specify endpoint 2 packet transmission enabling by a DMA transfer end interrupt. Since interrupt requests to the CPU are not masked automatically, interrupt requests must also be masked as necessary in the Interrupt Enable Register.

Bit	Bit Name	Initial Value	R/W	Description
0	EP1DMAE	0	R/W	**Endpoint 1 DMA Transfer Enable (EP1 DMAE):** When this bit is set, DMA transfer of receive data can be performed from the endpoint 1 transmit FIFO buffer to memory. If there is at least one byte of space in the FIFO buffer, the transfer request signal to the DMA controller is asserted. The number of receive bytes must therefore be set in the DMA controller by the endpoint 1 transfer normal end interrupt routine, and the endpoint 1 receive complete flag must be set by the DMA transfer end interrupt. Since interrupt requests to the CPU are not masked automatically, interrupt requests must also be masked as necessary in the Interrupt Enable Register.

23.6 Operation

23.6.1 Cable Connection

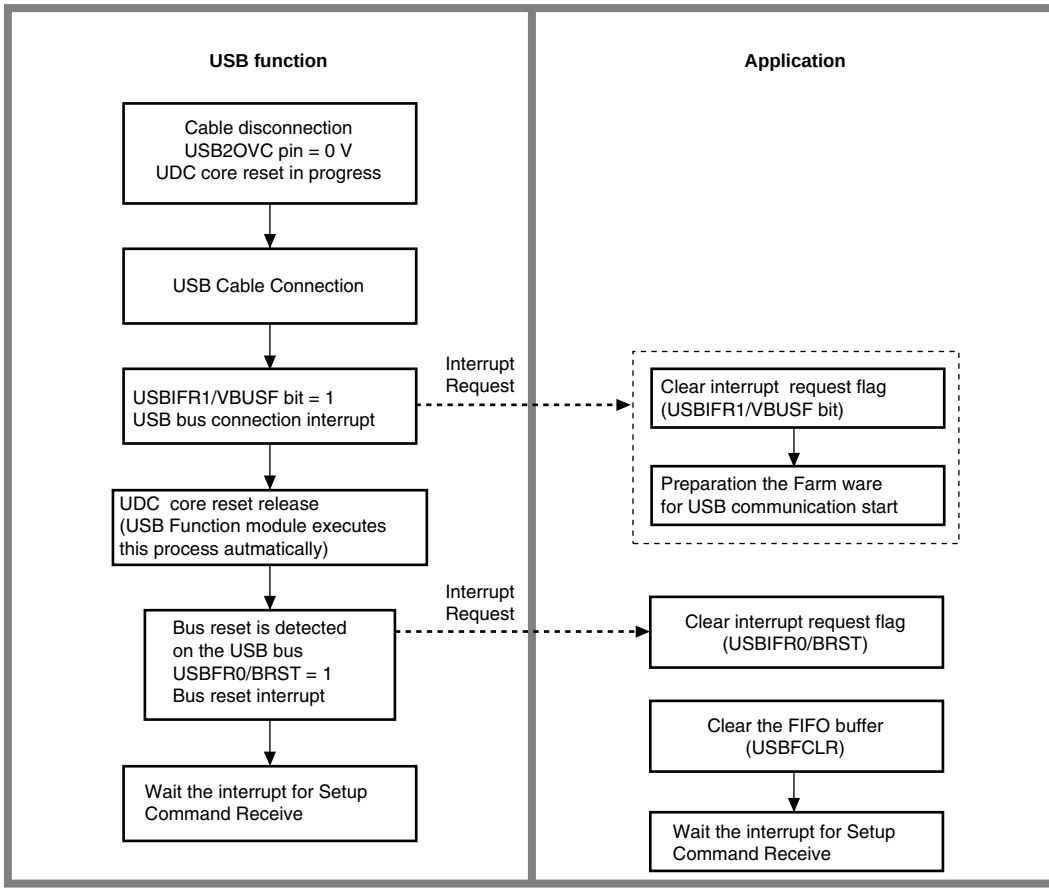


Figure 23.2 Cable Connection Operation

23.6.2 Cable Disconnection

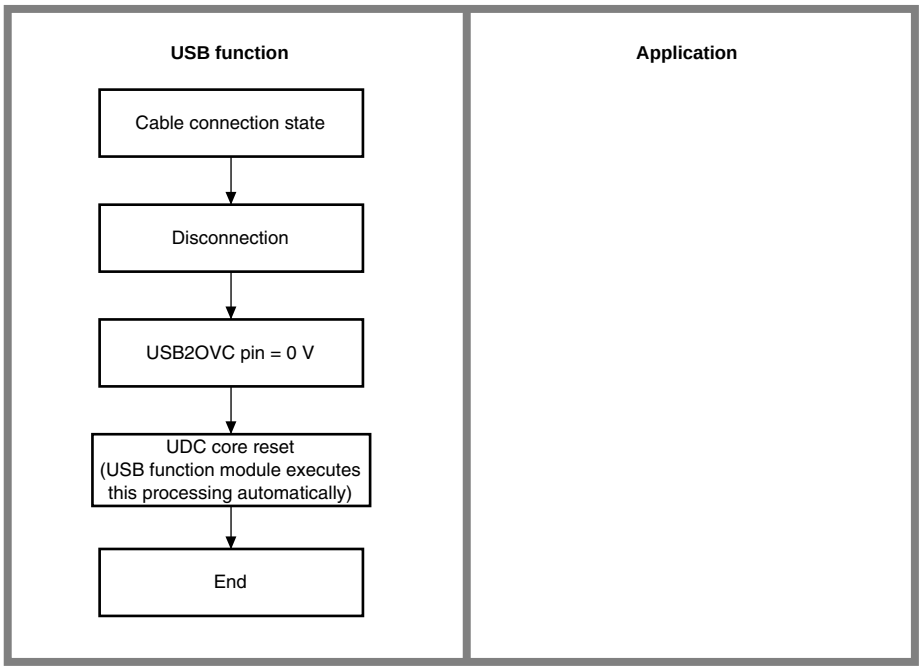


Figure 23.3 Cable Disconnection Operation

23.6.3 Control Transfer

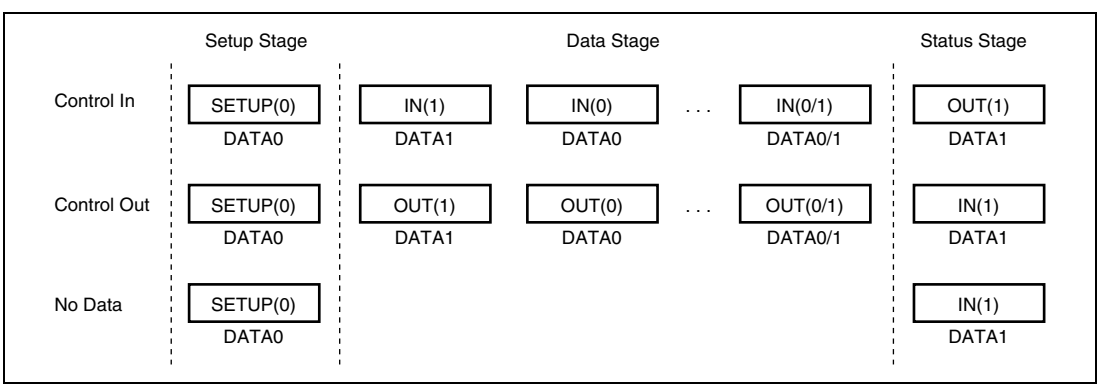
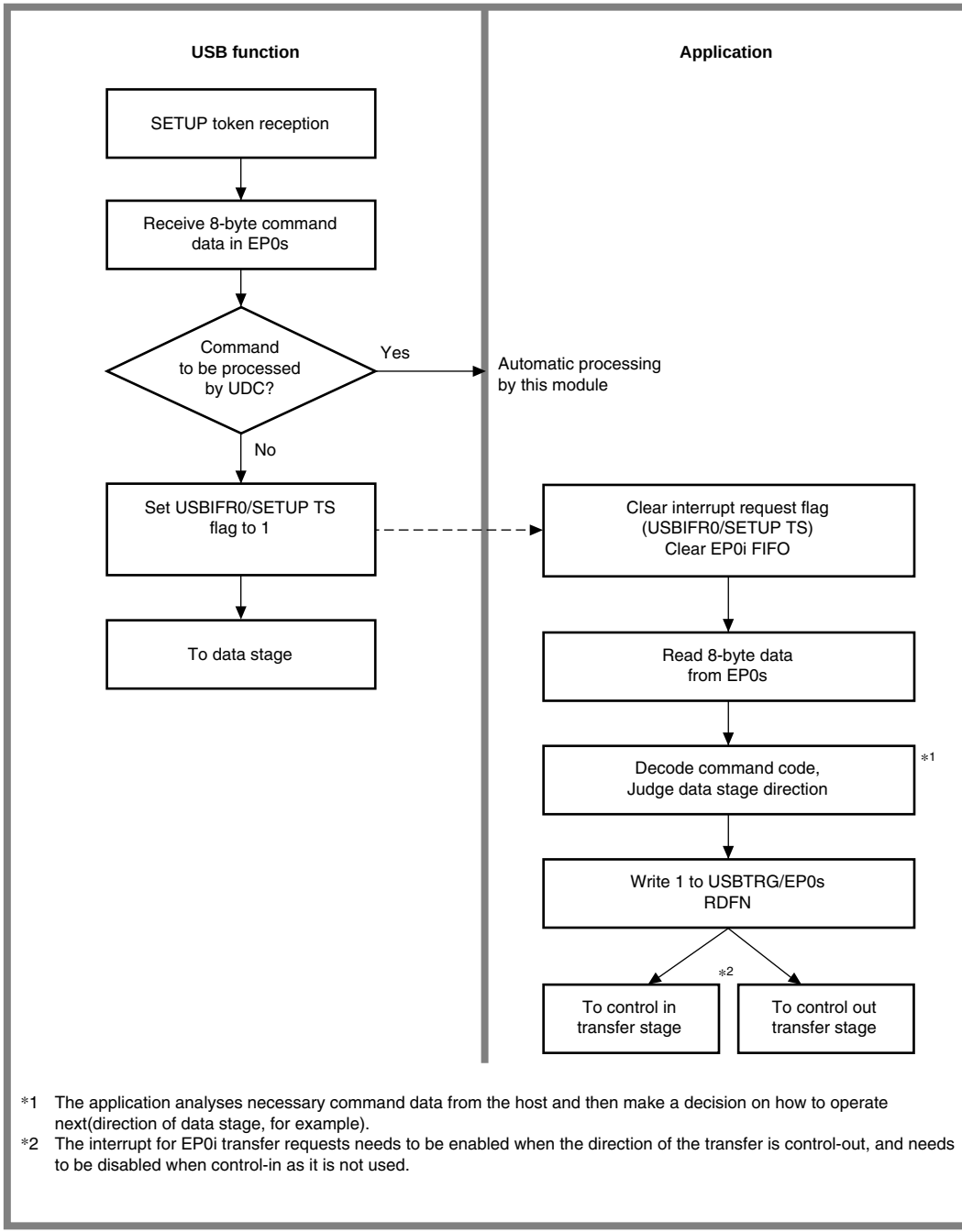


Figure 23.4 Each Transfer Stage in Control Stage

The control transfer consists of three stages that are setup, data (if present), and status (see the diagram of Control Transfer for Stages). The data stage consists of multiple bus transactions. The operation flow of each stage is shown below.

23.6.4 Setup Stage



*1 The application analyses necessary command data from the host and then make a decision on how to operate next(direction of data stage, for example).
*2 The interrupt for EP0i transfer requests needs to be enabled when the direction of the transfer is control-out, and needs to be disabled when control-in as it is not used.

Figure 23.5 Setup Transfer Operation

23.6.5 Data Stage (Control-Out Transfer)

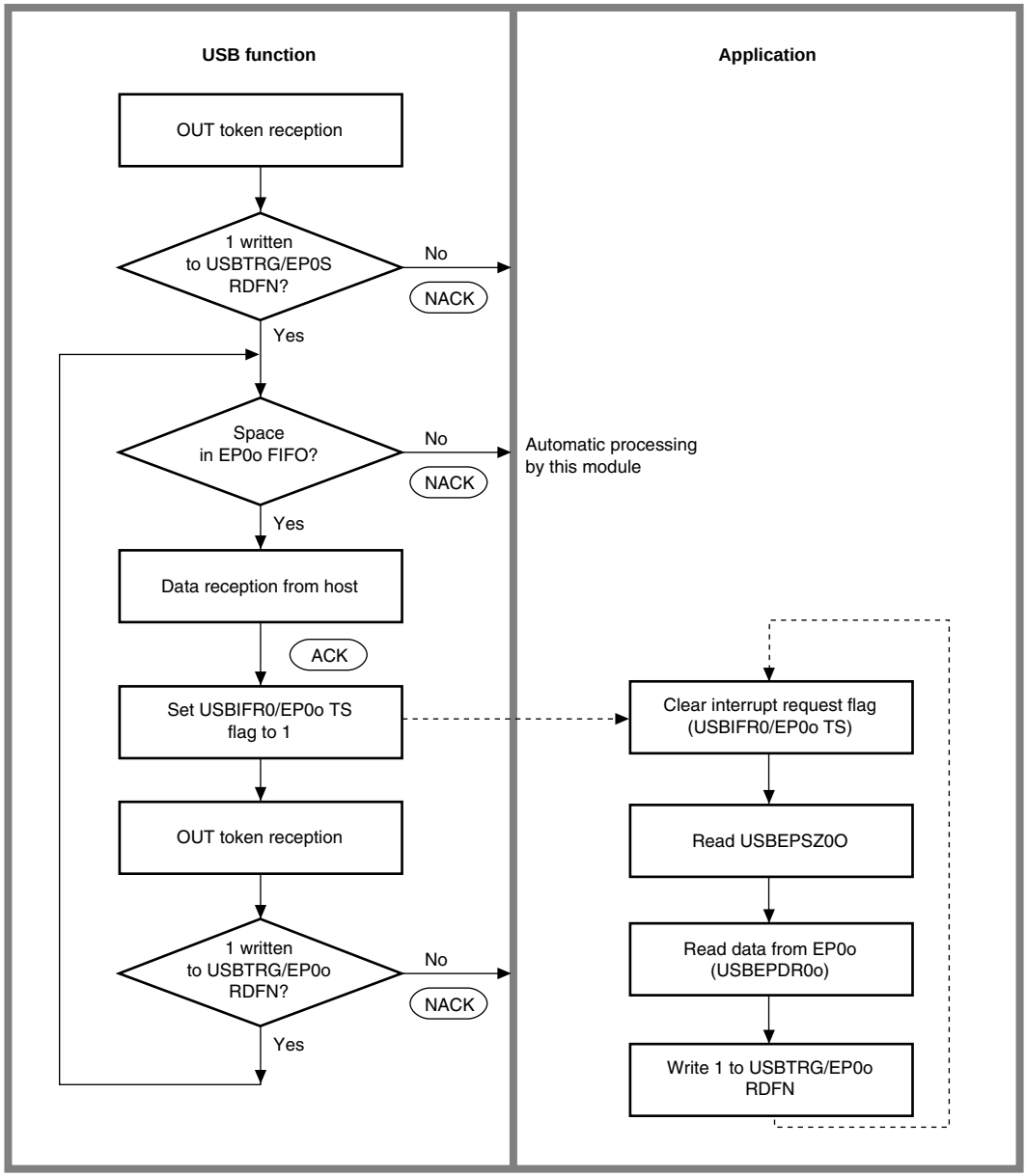


Figure 23.6 Data Stage (Control-Out Transfer Operation)

The application analyses necessary command data from the host and then make a decision on how to operate next. If the data stage is out-transfer by the result of the analysis, it waits for the data from the host and then reads the data from the FIFO after receiving it (IFRO/EP0o TS = 1). The

application may write a '1' into the EP0oo read complete bit and make the FIFO empty, and then wait for the next data to be transmitted.

The data stage is ended by the host transmitting the out token and entering the status stage.

23.6.6 Data Stage (Control-In Transfer)

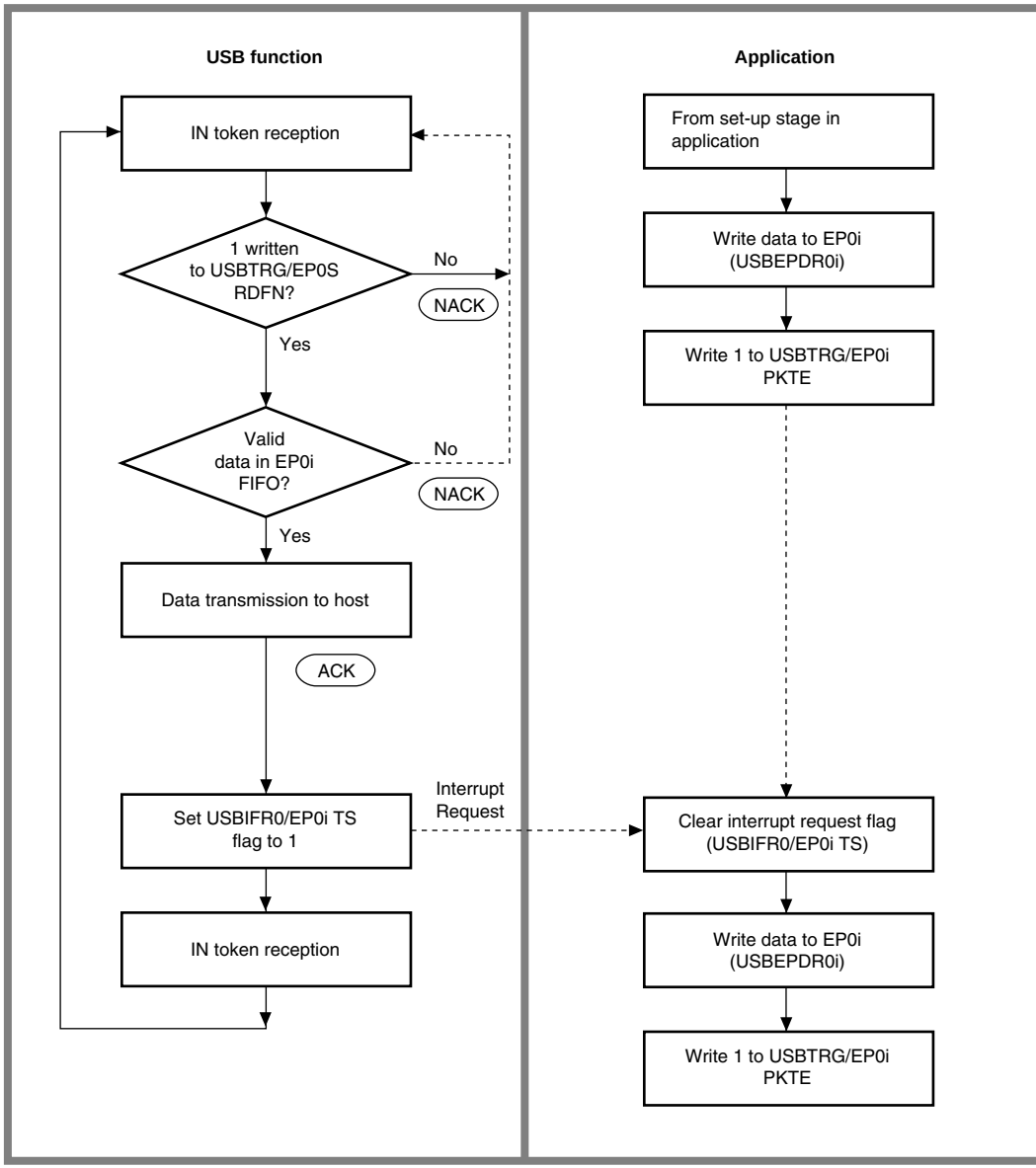


Figure 23.7 Control-In Transfer Operation

The application analyses necessary command data from the host and then make a decision on how to operate next. If the data stage is in-transfer by the result of the analysis, one packet of data to be transferred to the host may be written into the FIFO. If there is another data to be transferred, it may be written into the FIFO after transferring the first written data (IFRO/EP0i TS = 1).

The data stage is ended by the host transmitting the out token and entering the status stage.

Note: If the data size transferred by the function is smaller than that required by the host, the function notifies the end of data stage by transmitting a smaller number of packets to the host than the maximum packet size. If the data size transferred by the function is (the maximum packet size) X N, the function notifies the end of data stage by transmitting a '0' length packet (N: integer).

23.6.7 Status Stage(Control-In Transfer)

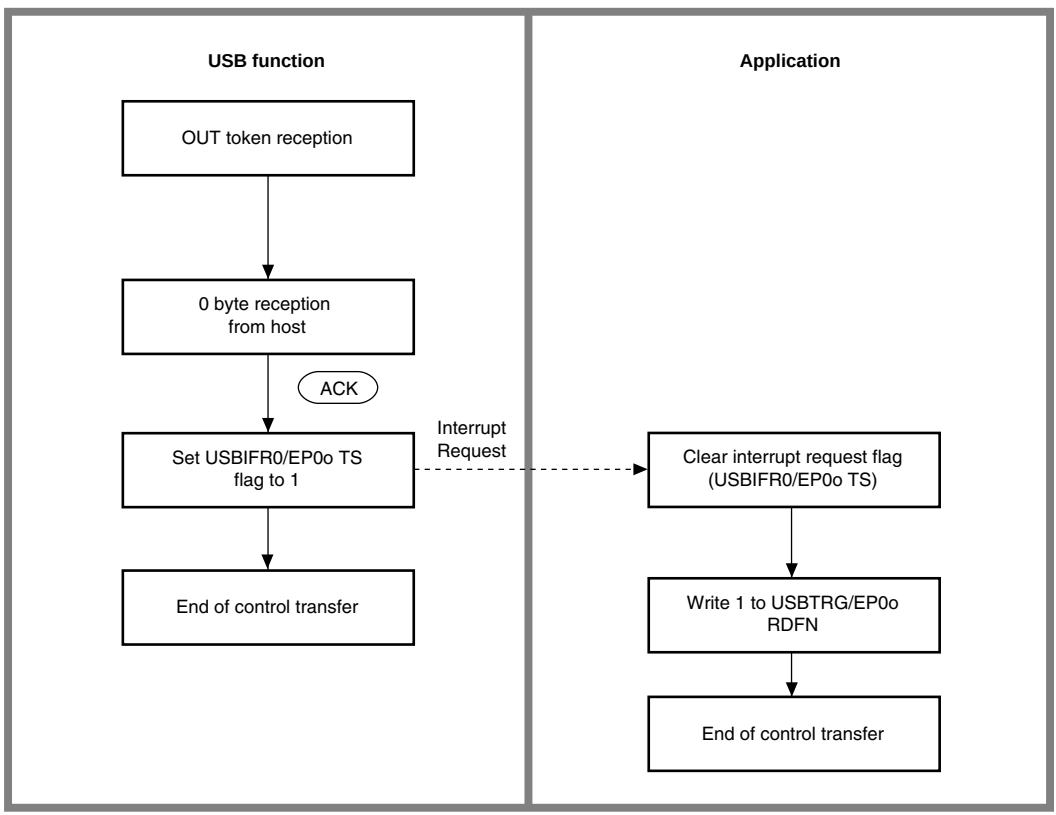


Figure 23.8 Status Stage (Control-In Transfer Operation)

The status stage at control-in is started by the out token from the host. The application completes the control transfer by receiving '0' byte data from the host.

23.6.8 Status Stage (Control-Out Transfer)

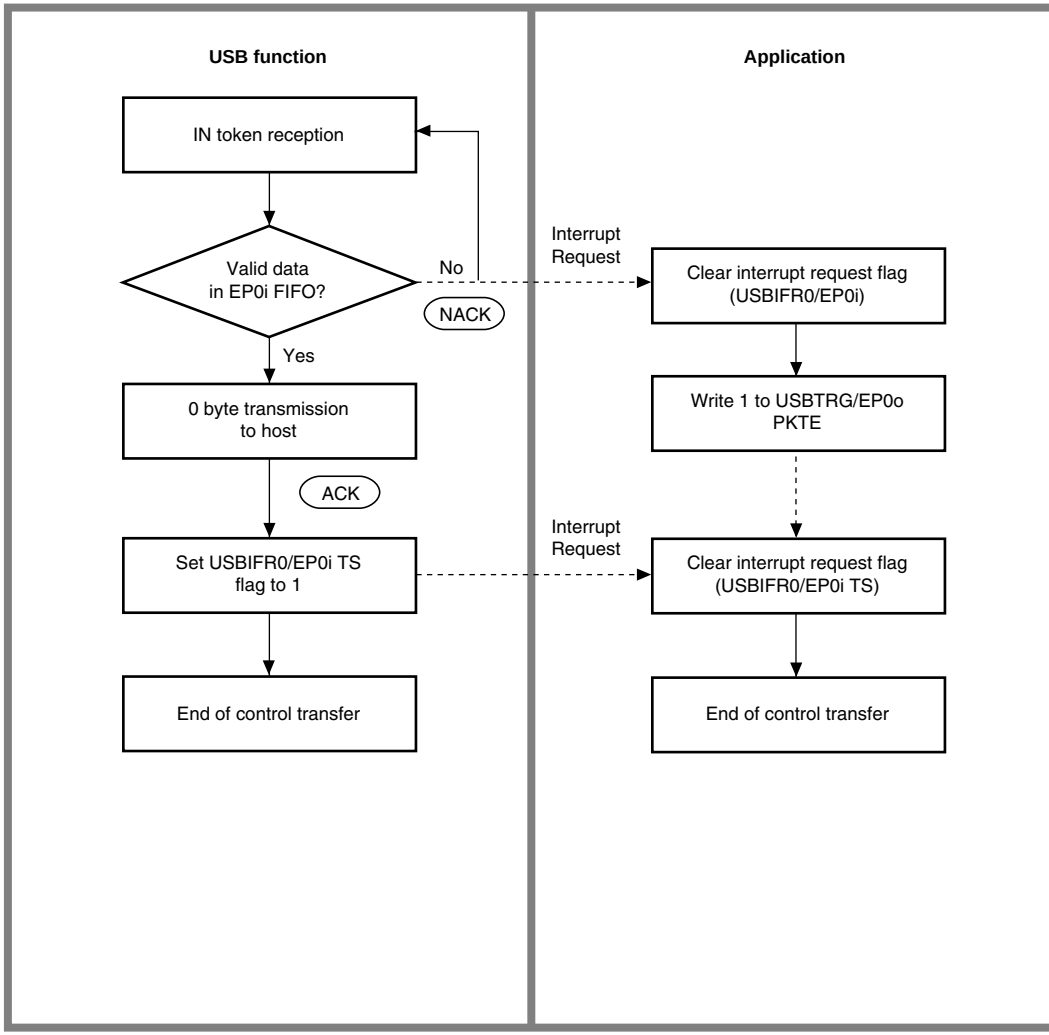


Figure 23.9 Status Stage (Control-Out Transfer Operation)

23.6.9 EP1 Bulk-Out Transfer (Dual FIFOs)

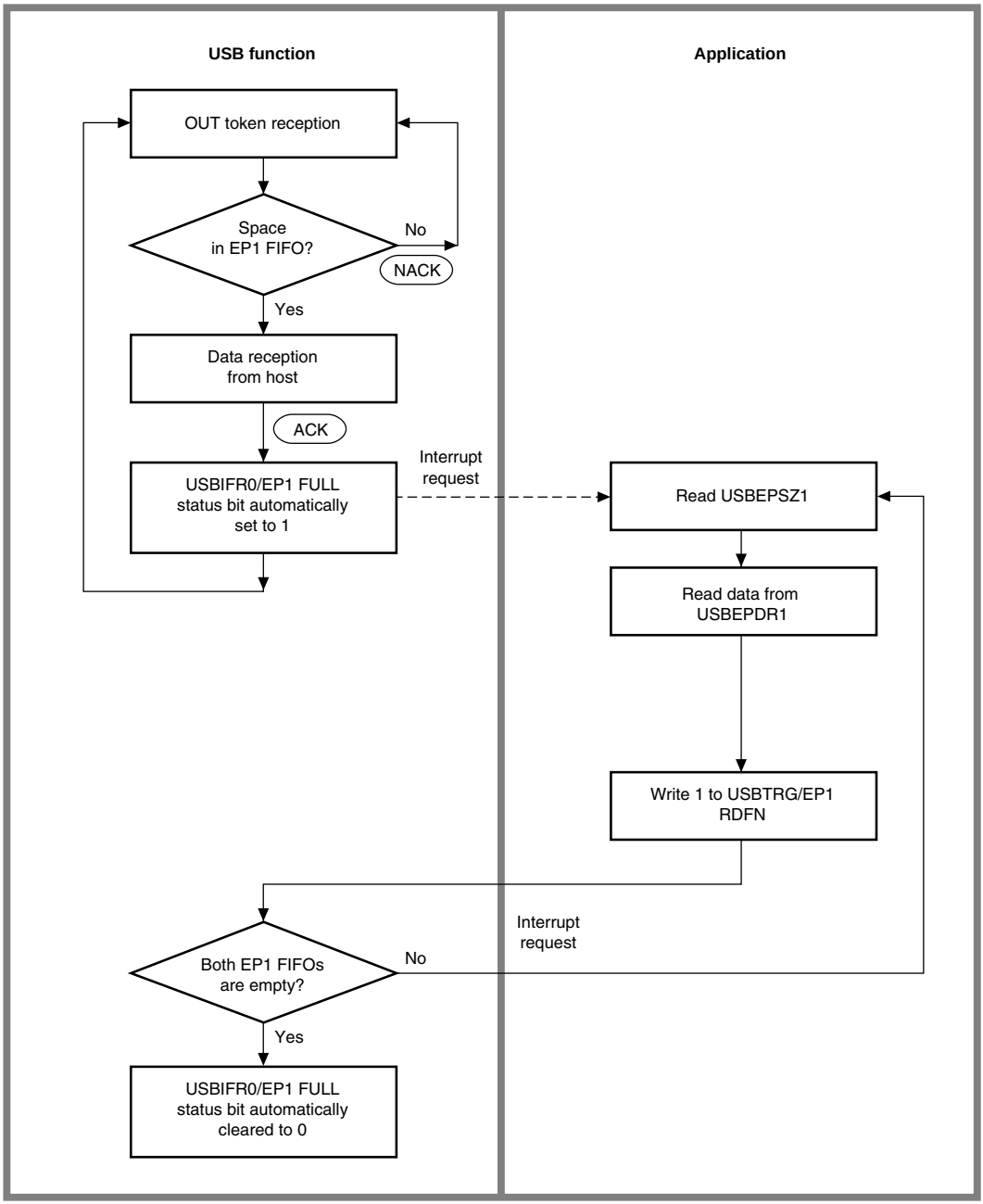


Figure 23.10 EP1 Bulk-Out Transfer Operation

EP1 has two 64-byte FIFOs, but the user can perform data reception and receive data reads without being aware of this dual-FIFO configuration.

When one FIFO is full after reception is completed, the USBIFR0/EP1 FULL bit is set. After the first receive operation into one of the FIFOs when both FIFOs are empty, the other FIFO is empty, and so the next packet can be received immediately. When both FIFOs are full, NACK is returned to the host automatically. When reading of the receive data is completed following data reception, 1 is written to the USBTRG/EP1 RDFN bit. This operation empties the FIFO that has just been read, and makes it ready to receive the next packet.

23.6.10 EP2 Bulk-In Transfer (Dual FIFOs)

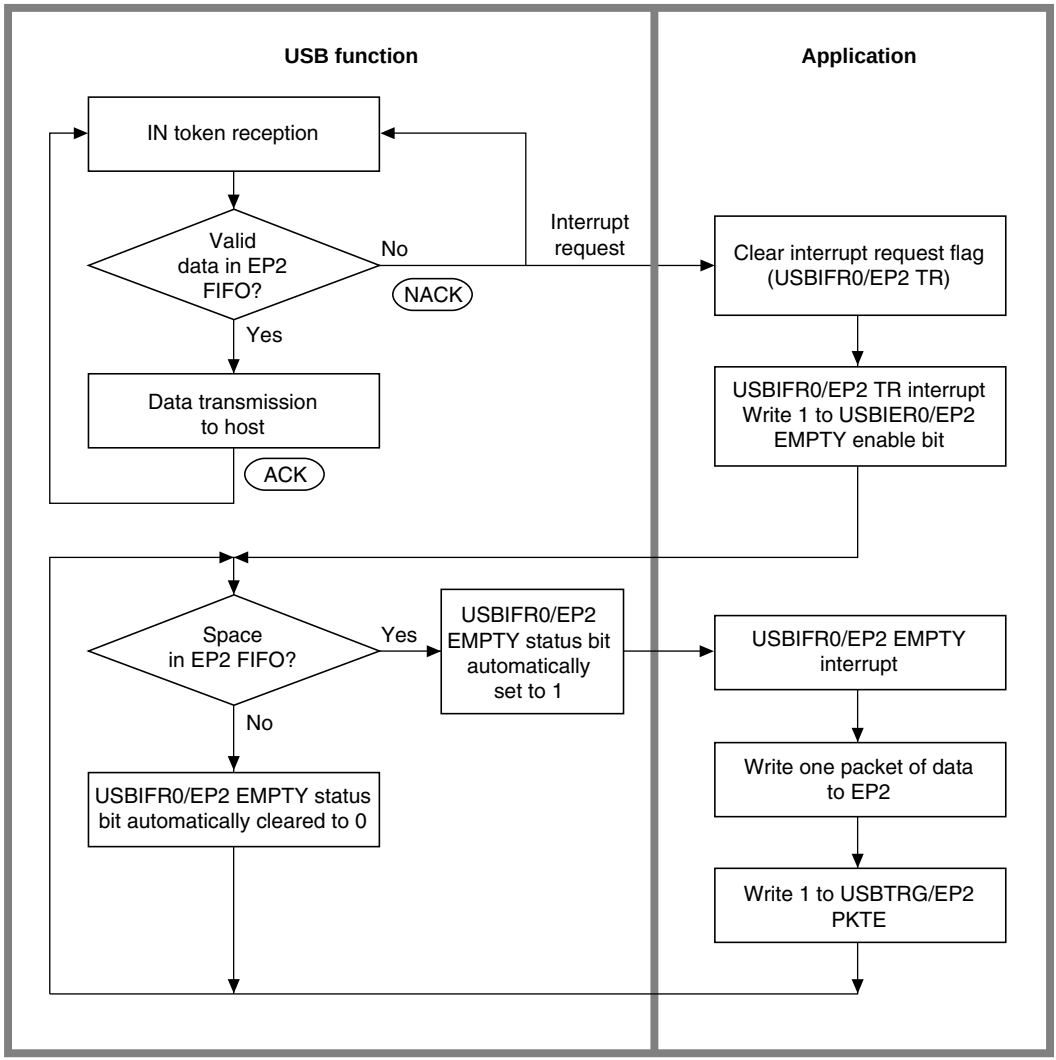


Figure 23.11 EP2 Bulk-In Transfer Operation

EP2 has two 64-byte FIFOs, but the user can perform data transmission and transmit data writes without being aware of this dual-FIFO configuration.

To perform bulk-in transfer, first write 1 to the USBIER0/EP2 EMPTY bit and enable the EP2 EMPTY interrupt. Immediately after a reset the two EP2 FIFOs are empty, and so USBIER0/EP2 EMPTY is already set to 1. An interrupt request is therefore generated immediately when the EP2 EMPTY interrupt is enabled.

The data to be transmitted is written to the data register using this interrupt. After the first transmit data write, the other FIFO is empty, and so the next transmit data can be written immediately. When both FIFOs are full, EP2 EMPTY is cleared to 0. (The status of both FIFOs is sampled at each CK clock, and the result is indicated in USBIFR0/EP2 EMPTY. If at least one FIFO is empty, USBIFR0/EP2 EMPTY is set to 1.) When ACK is returned from the host after data transmission is completed, the FIFO used in the data transmission becomes empty. If the other FIFO contains valid transmit data at this time, transmission is continued.

When transmission of all data has been completed, write 0 to USBIFR0/EP2 EMPTY and disable interrupt requests.

23.6.11 EP3 Interrupt-In Transfer

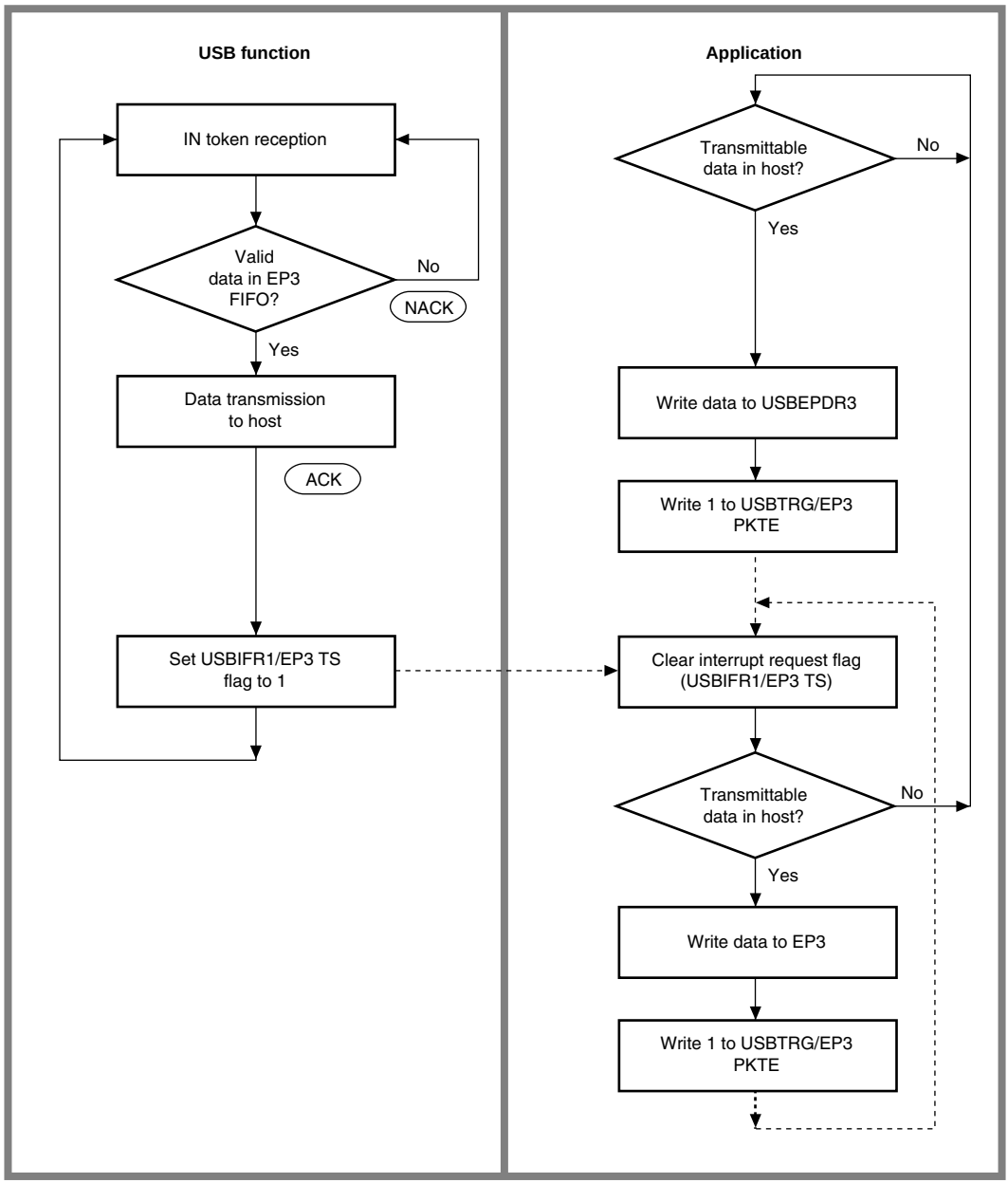


Figure 23.12 EP3 Interrupt-In Transfer Operation

23.7 Processing of USB Standard Commands and Class/Vendor Commands

23.7.1 Processing of Commands Transmitted by Control Transfer

A command transmitted from the host by control transfer may require decoding and execution of command processing on the application side. Whether command decoding is required on the application side is indicated in table 23.3 below.

Table 23.3 Command Decoding on Application Side

Decoding not Necessary on Application Side	Decoding Necessary on Application Side
Clear feature	Get descriptor
Get configuration	Synch frame
Get interface	Set descriptor
Get status	Class/Vendor command
Set address	
Set configuration	
Set feature	
Set interface	

If decoding is not necessary on the application side, command decoding and data stage and status stage processing are performed automatically. No processing is necessary by the user. An interrupt is not generated in this case.

If decoding is necessary on the application side, the USB function module stores the command in the EP0s FIFO. After normal reception is completed, the USBIER0/SETUP TS flag is set and an interrupt request is generated. In the interrupt routine, 8 bytes of data must be read from the EP0s Data Register (USBEPDR0S) and decoded by firmware. The necessary data stage and status stage processing should then be carried out according to the result of the decoding operation.

23.8 Stall Operations

23.8.1 Overview

This section describes stall operations in the USB function module. There are two cases in which the USB function module stall function is used:

- When the application forcibly stalls an endpoint for some reason
- When a stall is performed automatically within the USB function module due to a USB specification violation

The USB function module has internal status bits that hold the status (stall or non-stall) of each endpoint. When a transaction is sent from the host, the module references these internal status bits and determines whether to return a stall to the host. These bits cannot be cleared by the application; they must be cleared with a Clear Feature command from the host.

23.8.2 Forcible Stall by Application

The application uses the USBEPSTL Register to issue a stall request for the USB function module. When the application wishes to stall a specific endpoint, it sets the corresponding bit in USBEPSTL (1-1 in figure 23.13). The internal status bits are not changed. When a transaction is sent from the host for the endpoint for which the USBEPSTL bit was set, the USB function module references the internal status bit, and if this is not set, references the corresponding bit in USBEPSTL (1-2 in figure 23.13). If the corresponding bit in USBEPSTL is set, the USB function module sets the internal status bit and returns a stall handshake to the host (1-3 in figure 23.13). If the corresponding bit in USBEPSTL is not set, the internal status bit is not changed and the transaction is accepted.

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the USBEPSTL Register. Even after a bit is cleared by the Clear Feature command (3-1 in figure 23.13), the USB function module continues to return a stall handshake while the bit in USBEPSTL is set, since the internal status bit is set each time a transaction is executed for the corresponding endpoint (1-2 in figure 23.13). To clear a stall, therefore, it is necessary for the corresponding bit in USBEPSTL to be cleared by the application, and also for the internal status bit to be cleared with a Clear Feature command (2-1, 2-2, and 2-3 in figure 23.13).

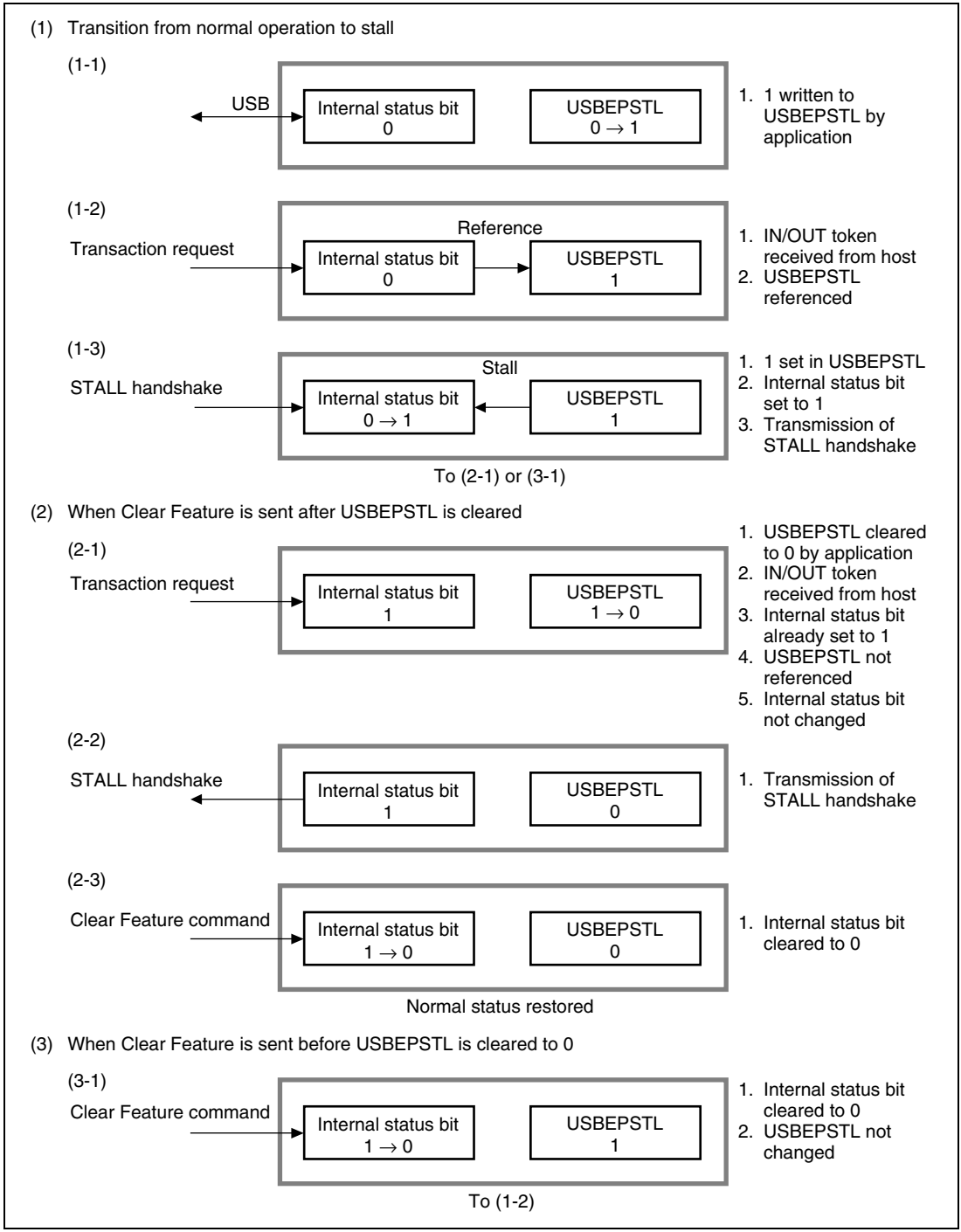


Figure 23.13 Forcible Stall by Application

23.8.3 Automatic Stall by USB Function Module

When a stall setting is made with the Set Feature command, or in the event of a USB specification violation, the USB function module automatically sets the internal status bit for the relevant endpoint without regard to the USBEPSTL register, and returns a stall handshake (1-1 in figure 23.14).

Once an internal status bit is set, it remains set until cleared by a Clear Feature command from the host, without regard to the USBEPSTL register. After a bit is cleared by the Clear Feature command, USBEPSTL is referenced (3-1 in figure 23.14). The USB function module continues to return a stall handshake while the internal status bit is set, since the internal status bit is set even if a transaction is executed for the corresponding endpoint (2-1 and 2-2 in figure 23.14). To clear a stall, therefore, the internal status bit must be cleared with a Clear Feature command (3-1 in figure 23.14). If set by the application, USBEPSTL should also be cleared (2-1 in figure 23.14).

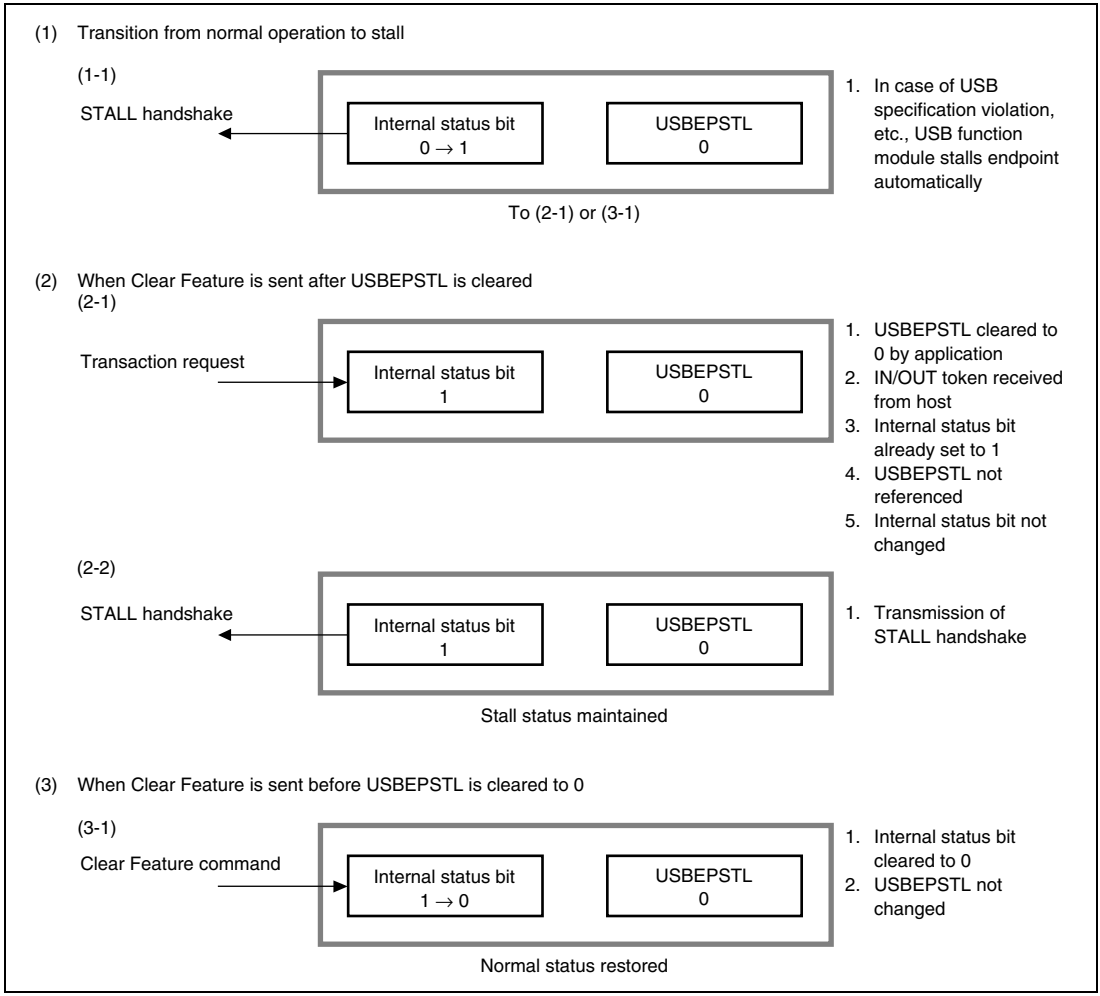


Figure 23.14 Automatic Stall by USB Function Module

23.9 Connection example of an external circuit

23.9.1 Example 1 (When Using USB2PENC)

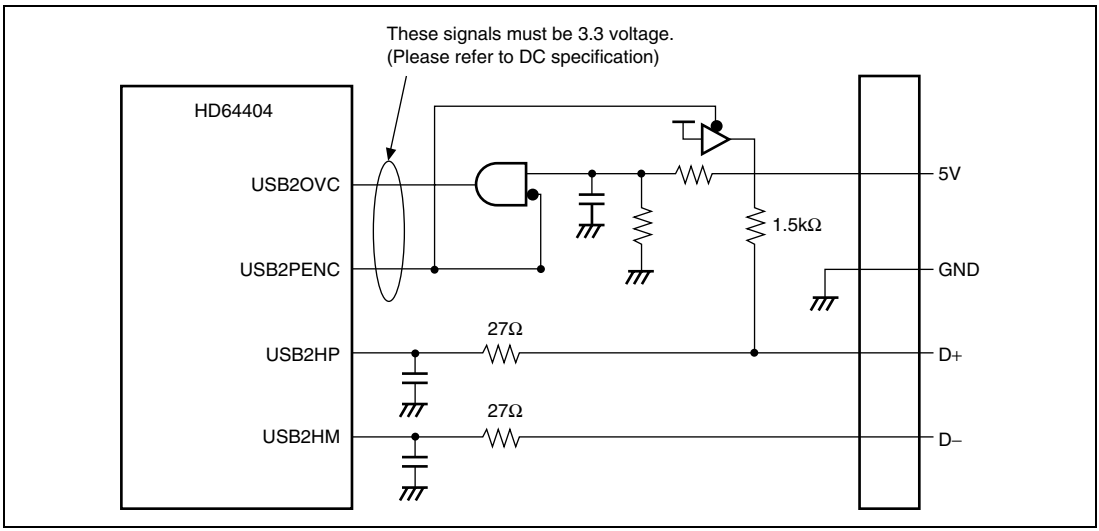


Figure 23.15 Connection Example 1 (When Using USB2PENC)

In case of above connection, The power control and cable connection /disconnection by USB2PENC is possible. When USB2PENC is low level, USB2OVC and Pull-up resistor is off states. Therefore its cable disconnection state. As for this connection, the control of cable connection and power is possible by oneself.

23.9.2 Example 2 (When Using USB2PENC)

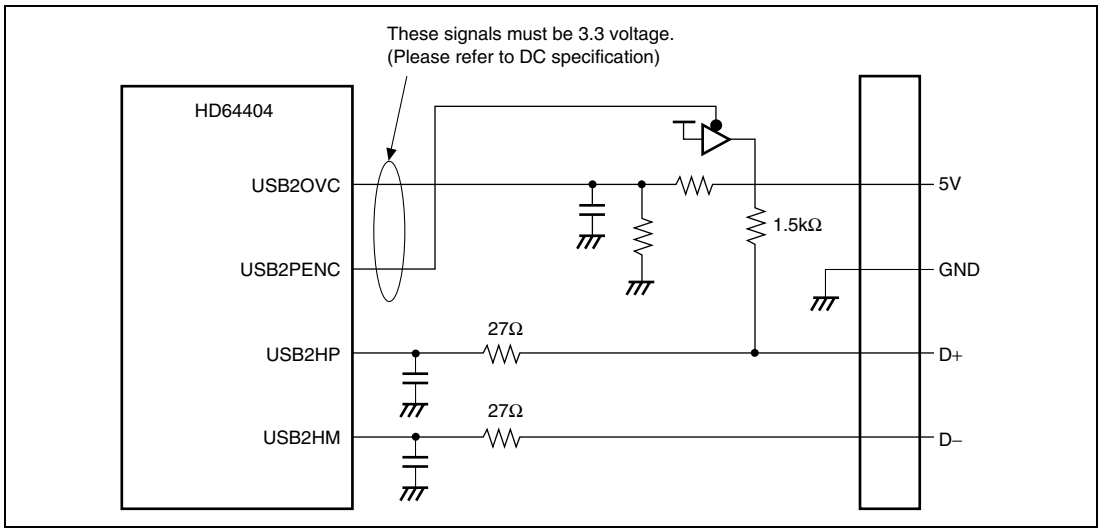


Figure 23.16 Connection Example 2 (When Using USB2PENC)

In case of above connection, The cable connection/disconnection by USB2PENC is possible. When USB2PENC is low level, Pull-up resistor is off states. Therefore its cable is in disconnection state. But so that Host control power, don't stop quickly. As for this connection, Only the control of cable connection is possible by oneself.

23.9.3 Example 3 (When Not Using USB2PENC)

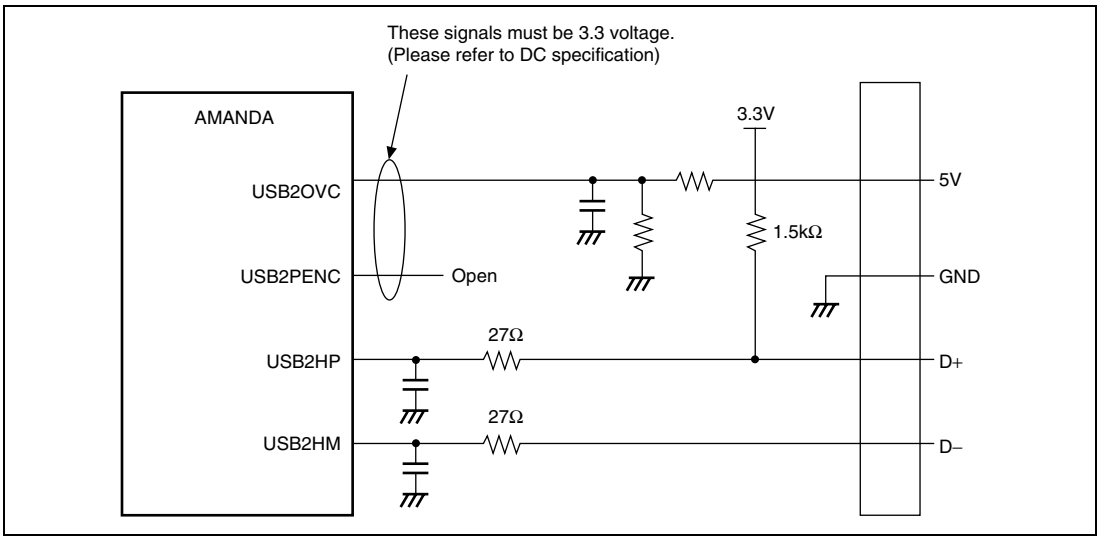


Figure 23.17 Example 3 (When Not Using USB2PENC)

In case of above connection, function control is possible.

23.10 Module Standby Mode

This USB Function module allows clock gating to reduce power consumption.

Both pixel bus clock and register bus clock can be gated. This module standby mode can be executed by controlling Clock Control 2 Register in Power Control & Configuration module.

To wake up the module, USB1 bit of Clock Control 2 Register must be enabled. After enabling this bit, initialisation of this module must be executed.

Section 24 USB HOST

24.1 General Description

USB Host i/f provides an integrated RootHub and 2 USB transceiver ports which support both low speed and high speed. Also it supports Open HCI i/f and operational registers.

In addition, in a software plan, please refer to OpenHCI specifications.

24.2 Features

- Open HCI i/f support
- USB Host i/f support
- RootHub function
- Support Low speed (1.5 Mbps), and High speed (12 Mbps) modes
- Support Overcurrent detection and power enable control

Note: The USB host module connects a pixel bus and a register bus inside HD64404. The register bus is used to access the registers of USB host module . The pixel bus is to transfer/receive data, Endpoint Descriptor (ED), and Transfer Descriptor (TD), to/from Graphic Memory (GM). USB transfer/receive data, ED, and TD are all transferred/received via GM.

Table 24.1 External interface

Signal	Function	Polarity	Direction
USB1HP	USB port1 D+	—	IN/OUT
USB1HM	USB port1 D-	—	IN/OUT
USB1PENC	USB port1 power enable control	High Active	OUT
USB1OVC	USB port1 Over-current detect	Low Active	IN
USB2HP	USB port2 D+	—	IN/OUT
USB2HM	USB port2 D-	—	IN/OUT
USB2PENC	USB port2 power enable control	High Active	OUT
USB2OVC	USB port2 Over-current detect	Low Active	IN

24.3 Block Diagram

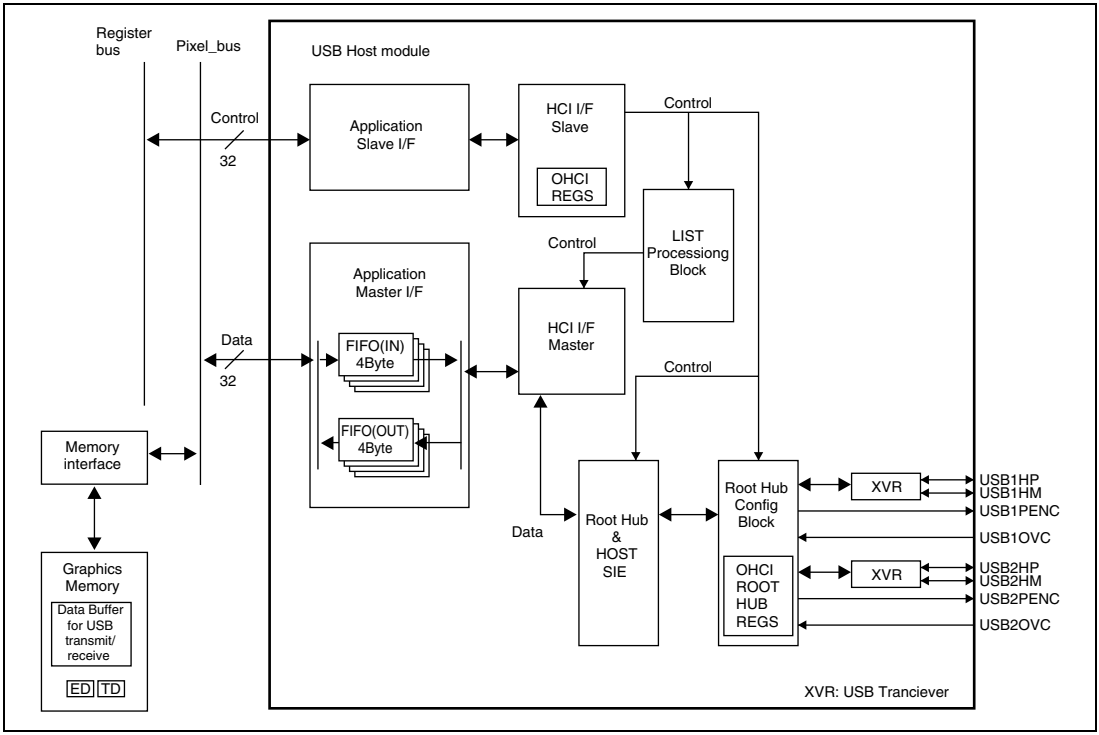


Figure 24.1 Block Diagram of USB Host

24.4 Register Description

There will be a set of registers which will be located in the address space of the PCI or MPX bus and will be located in the PCI memory window.

24.4.1 OpenHCI Registers

Table 24.2 OpenHCI Register Summary

Offset	Register Name	R/W	Initial Value	Access Size
0	HcRevision	R	H'00000010	32
04	HcControl	R/W	H'00000000	32
08	HcCommandStatus	R/W	H'00000000	32
0C	HcInterruptStatus	R/W	H'00000000	32
10	HcInterruptEnable	R/W	H'00000000	32
14	HcInterruptDisable	R/W	H'00000000	32
18	HcHCCA	R/W	H'00000000	32
1C	HcPeriodCurrentED	R/W	H'00000000	32
20	HcControlHeadED	R/W	H'00000000	32
24	HcControlCurrentED	R/W	H'00000000	32
28	HcBulkHeadED	R/W	H'00000000	32
2C	HcBulkCurrentED	R/W	H'00000000	32
30	HcDoneHead	R/W	H'00000000	32
34	HcFmInterval	R/W	H'00002EDF	32
38	HcFmRemaining	R	H'00000000	32
3C	HcFmNumber	R	H'00000000	32
40	HcPeriodicStart	R/W	H'00000000	32
44	HcLSThreshold	R/W	H'00000628	32
48	HcRhDescriptorA	R/W	H'02001202	32
4C	HcRhDescriptorB	R/W	H'00000000	32
50	HcRhStatus	R/W	H'00000000	32
54	HcRhPortStatus[1]	R/W	H'00000100*	32
58	HcRhPortStatus[2]	R/W	H'00000100*	32
F0	ConfigurationControl	R/W	H'00000000	32

Note: * This means an initial value in full speed mode. As for the initial value low speed mode, bit 9 changes to “1”.

Legends for register description:

Initial value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, read value undefined.

Notes: These registers can be set up when 48MHz clock is supplied. It needs to wait for USB XTAL to be oscillated for the time that is specified in Electrical Specification. Additionally these registers cannot be set up by a maximum of 6 microseconds from setting up the Host bit of XTAL Control [1] register in the Power Control & Configration module, 48MHz-clock supply is controllable. Regisers with offsets 0 to 58 support Open HCI Specifications. A regiser with offset FO is specific to HD64404. And, these registers cannot be set up by a maximum of 6 microseconds, since 48MHz clock begins to be supplied. By setting up the Host bit of XTAL Control Register in the Power Control & Configuration module, 48MHz clock supply is controllable.

Registers with offsets 0 to 58 support Open HCI Specifications.

A Register with offset F0 is specific to HD64404.

HcRevision Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									Revision							
Initial:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Register: HcRevision			Offset: 00 to 03
Bits	**Initial Value**	**R/W**	**Description**
31 to 8	0	R	Reserved. Read/Write 0's
7 to 0	H'10	R	**Revision** Indicates the OpenHCI Specification revision number implemented by the Hardware. (X.Y = XYh) USB Host Controller supports the 1.0 specification.

HcControl Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						RWCE	RWC	IR	HCFS		BLE	CLE	IE	PLE	CBSR	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register: HcControl			Offset: 04 to 07
Bits	**Initial Value**	**R/W**	**Description**
31 to 11	0	R	Reserved. Read/Write 0's
10	0	R/W	**RemoteWakeupConnectedEnable** If a remote wakeup signal is supported, this bit is used to enable that operation. Since there is no remote wakeup signal supported, this bit is ignored.
9	0	R/W	**RemoteWakeupConnected** This bit indicated whether the HC supports a remote wakeup signal.

Register: HcControl			Offset: 04 to 07
Bits	**Initial Value**	**R/W**	**Description**
8	0	R/W	**InterruptRouting** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
7 to 6	0	R/W	**HostControllerFunctionalState** This field is used to set the Host Controller state. The state encodings are: 00: USBRESET 01: USBRESUME 10: USBOPERATIONAL 11: USBSUSPEND The Host Controller may force a state change from USB SUSPEND to USB RESUME after detecting resume signaling from a downstream port.
5	0	R/W	**BulkListEnable** When set this bit enables processing of the Bulk list.
4	0	R/W	**ControlListEnable** When set this bit enables processing of the Control list.
3	0	R/W	**IsochronousEnable** When clear, this bit disables the Isochronous List when the Periodic List is enabled (so Interrupt EDs may be serviced). While processing the Periodic List, the Host Controller will check this bit when it finds an isochronous ED.
2	0	R/W	**PeriodicListEnable** When set, this bit enables processing of the Periodic (interrupt and isochronous) list. The Host Controller checks this bit prior to attempting any periodic transfers in a frame.
1, 0	0	R/W	**ControlBulkServiceRatio** Specifies the number of Control Endpoints serviced for every Bulk Endpoint. Encoding is N-1 where N is the number of Control Endpoints (i.e. '00' = 1 Control Endpoint; '11' = 4 Control Endpoints)

HcCommandStatus Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															SOC	
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													OCR	BLF	CLF	HCR
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Register: HcCommandStatus **Offset: 08 to 0B**

Bits	Initial Value	R/W	Description
31 to 18	0	R	Reserved. Read/Write 0's
17 to 16	0	R	**ScheduleOverrunCount** This field increments every time the **SchedulingOverrun** bit in HcInterruptStatus is set. The count wraps from '11' to '00. '
15 to 4	0	R	Reserved. Read/Write 0's
3	0	R/W	**OwnershipChangeRequest** When set by software, this bit sets the **OwnershipChange** field in HcInterruptStatus. The bit is cleared by software.
2	0	R/W	**BulkListFilled** When set, this bit indicates there is an active ED on the Bulk List. The bit may be set by either software or the Host Controller. The bit is cleared by the Host Controller each time it begins processing the head of the Bulk List.
1	0	R/W	**ControlListFilled** When set, this bit indicates there is an active ED on the Control List. The bit may be set by either software or the Host Controller. The bit is cleared by the Host Controller each time it begins processing the head of the Control List.
0	0	R/W	**HostControllerReset** This bit is set to initiate a software reset. This bit is cleared by the Host Controller upon completion of the reset operation.

HcInterruptStatus Register

All bits are set by hardware and cleared by software.

These bits in this register can be cleared by writing 1 to bit positions to be cleared.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		OC														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R / WC1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										RHSC	FNO	UE	RD	SF	WDH	SO
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1

Register: HcInterruptStatus **Offset: 0C to 0F**

Bits	Initial Value	R/W	Description
31	0	R	Reserved. Read/Write 0's
30	0	R/WC1	**OwnershipChange** This bit is set when the **OwnershipChangeRequest** bit of HcCommandStatus is set.
29 to 7	0	R	Reserved. Read/Write 0's
6	0	R/WC1	**RootHubStatusChange** This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus register has changed.
5	0	R/WC1	**FrameNumberOverflow** This bit is set when bit 15 of **FrameNumber** changes value from '0' to '1' or from '1' to '0'.
4	0	R/WC1	**UnrecoverableError** This bit is set when HC detects a system error that is not USB related.
3	0	R/WC1	**ResumeDetected** This bit is set when the Host Controller detects resume signaling on a downstream port.
2	0	R/WC1	**StartofFrame** This bit is set when the Frame Management block signals a start of Frame' event.
1	0	R/WC1	**WritebackDoneHead** This bit is set after the Host Controller has written HcDoneHead to HccaDoneHead.
0	0	R/WC1	**SchedulingOverrun** This bit is set when the List Processor determines a Schedule Overrun has occurred.

HcInterruptEnable Register

Writing a '1' to a bit in this register sets the corresponding bit, while writing a '0' to a bit leaves the bit unchanged.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ME	OCE														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										RHSCE	FNOE	UEE	RDE	SOFE	WDHE	SOE
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register: HcInterruptEnable			Offset: 10 to 13
Bits	**Initial Value**	**R/W**	**Description**
31	0	R/W	**MasterInterruptEnable** This bit is a global interrupt enable. A write of '1' allows interrupts to be enabled via the specific enable bits listed below.
30	0	R/W	**OwnershipChangeEnable** 0: Ignore 1: Enable interrupt generation due to Ownership Change.
29 to 7	0	R	Reserved. Read/Write 0's
6	0	R/W	**RootHubStatusChangeEnable** 0: Ignore 1: Enable interrupt generation due to Root Hub Status Change.
5	0	R/W	**FrameNumberOverflowEnable** 0: Ignore 1: Enable interrupt generation due to Frame Number Overflow.
4	0	R/W	**UnrecoverableErrorEnable** This event is not implemented. All writes to this bit will be ignored.
3	0	R/W	**ResumeDetectedEnable** 0: Ignore 1: Enable interrupt generation due to Resume Detected.
2	0	R/W	**StartOfFrameEnable** 0: Ignore 1: Enable interrupt generation due to Start of Frame.
1	0	R/W	**WritebackDoneHeadEnable** 0: Ignore 1: Enable interrupt generation due to Writeback Done Head.
0	0	R/W	**SchedulingOverrunEnable** 0: Ignore 1: Enable interrupt generation due to Scheduling Overrun.

HcInteruptDisable Register

Writing a '1' to a bit in this register clears the corresponding bit, while writing a '0' to a bit leaves the bit unchanged.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MID	OCD														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R / WC1	R / WC1	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										RHSCD	FNOD	UED	RDD	SOFD	WDHD	SOD
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1	R / WC1

Register: HcInterruptDisable			Offset: 14 to 17
Bits	**Initial Value**	**R/W**	**Description**
31	0	R/W	**MasterInterruptDisable** This bit is a global interrupt disable. A write of '1' disables all interrupts.
30	0	R/W	**OwnershipChangeDisable** 0: Ignore 1: Disable interrupt generation due to Ownership Change.
29 to 7	0	R	Reserved. Read/Write 0's
6	0	R/W	**RootHubStatusChangeDisable** 0: Ignore 1: Disable interrupt generation due to Root Hub Status Change.
5	0	R/W	**FrameNumberOverflowDisable** 0: Ignore 1: Disable interrupt generation due to Frame Number Overflow.
4	0	R/W	**UnrecoverableErrorDisable** This event is not implemented. All writes to this bit will be ignored.
3	0	R/W	**ResumeDetectedDisable** 0: Ignore 1: Disable interrupt generation due to Resume Detected.
2	0	R/W	**StartOfFrameDisable** 0: Ignore 1: Disable interrupt generation due to Start of Frame.
1	0	R/W	**WritebackDoneHeadDisable** 0: Ignore 1: Disable interrupt generation due to Writeback Done Head.
0	0	R/W	**SchedulingOverrunDisable** 0: Ignore 1: Disable interrupt generation due to Scheduling Overrun.

HcHCCA Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	HCCA															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	HCCA															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Register: HcHCCA **Offset: 18 to 1B**

Bits	Initial Value	R/W	Description
31 to 8	0	R/W	HCCA Pointer to HCCA base address. (Within Graphics Memory space)
7 to 0	0	R	Reserved. Read/Write 0's

HcPeriodCurrentED Register

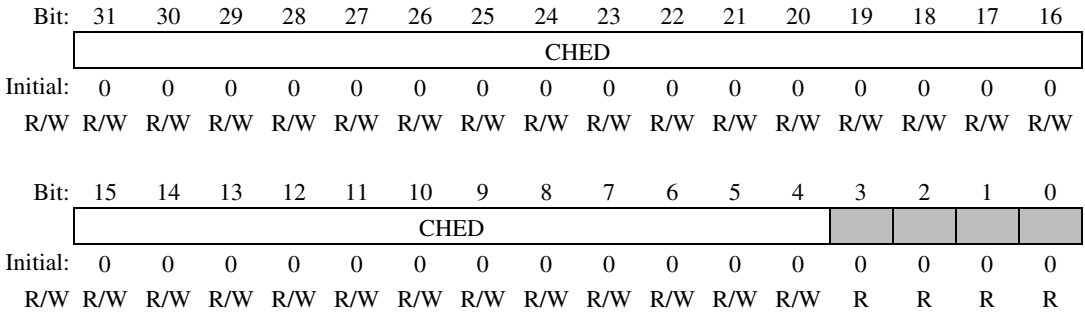
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PCED															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PCED															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Register: HcPeriodCurrentED **Offset: 1C to 1F**

Bits	Initial Value	R/W	Description
31 to 4	0	R	**PeriodCurrentED** Pointer to the current Periodic List ED. (Within Graphics Memory space)
3 to 0	0	R	Reserved. Read/Write 0's

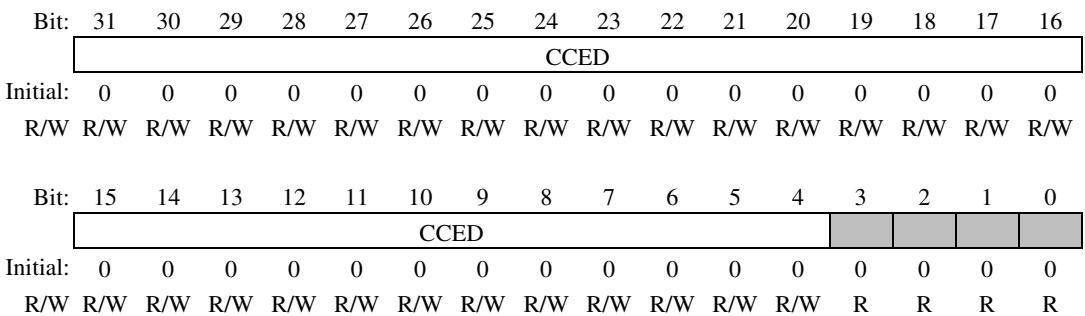
HcControlHeadED Register



Register: HcControlHeadED Offset: 20 to 23

Bits	Initial Value	R/W	Description
31 to 4	0	R/W	**ControlHeadED** Pointer to the Control List Head ED. (Within Graphics Memory space)
3 to 0	0	R	Reserved. Read/Write 0's

HcControlCurrentED Register



Register: HcControlCurrentED Offset: 24 to 27

Bits	Initial Value	R/W	Description
31 to 4	0	R/W	**ControlCurrentED** Pointer to the current Control List ED. (Within Graphics Memory space)
3 to 0	0	R	Reserved. Read/Write 0's

HcBulkHeadED Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BHED															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BHED															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Register: HcBulkHeadED **Offset: 28 to 2B**

Bits	Initial Value	R/W	Description
31 to 4	0	R/W	**BulkHeadED** Pointer to the Bulk List Head ED. (Within Graphics Memory space)
3 to 0	0	R	Reserved. Read/Write 0's

HcBulkCurrentED Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BCED															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCED															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Register: HcBulkCurrentED **Offset: 2C to 2F**

Bits	Initial Value	R/W	Description
31 to 4	0	R/W	**BulkCurrentED** Pointer to the current Bulk List ED. (Within Graphics Memory space)
3 to 0	0	R	Reserved. Read/Write 0's

HcDoneHead Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DH															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DH															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Register: HcDoneHead **Offset: 30 to 33**

Bits	Initial Value	R/W	Description
31 to 4	0	R	**DoneHead** Pointer to the current Done List Head ED. (Within Graphics Memory space)
3 to 0	0	R	Reserved. Read/Write 0's

HcFmInterval Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIT	FSMPS														
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			FI													
Initial:	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register: HcFmInterval			Offset: 34 to 37
Bits	**Initial Value**	**R/W**	**Description**
31	0	R/W	**FrameIntervalToggle** This bit is toggled by HCD whenever it loads a new value into **FrameInterval**.
30 to 16	0	R/W	**FSLargestDataPacket** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame.
15 to 14	0	R	Reserved. Read/Write 0's
13 to 0	H'2EDF	R/W	**FrameInterval** This field specifies the length of a frame as (bit times - 1). For 12,000 bit times in a frame, a value of 11,999 is stored here.

HcFrameRemaining Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FRT															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			FR													
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Register: HcFrameRemaining			Offset: 38 to 3B
Bits	**Initial Value**	**R/W**	**Description**
31	0	R	**FrameRemainingToggle** This bit is loaded with **FrameIntervalToggle** when **FrameRemaining** is loaded.
30 to 14	0	R	Reserved. Read/Write 0's
13 to 0	0	R	**FrameRemaining** This field is a 14 bit decrementing counter used to time a frame. When the Host Controller is in the USB OPERATIONAL state the counter decrements each 12 MHz clock period. When the count reaches 0, the end of a frame has been reached. The counter reloads with **FrameInterval** at that time. In addition, the counter loads when the Host Controller transitions into USB OPERATIONAL.

HcFmNumber Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FN															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Register: HcFmNumber **Offset: 3C to 3F**

Bits	Initial Value	R/W	Description
31 to 16	0	R	Reserved. Read/Write 0's
15 to 0	0	R	**FrameNumber** This field is a 16 bit incrementing counter. The count is incremented coincident with the loading of **FrameRemaining**. The count will roll over from ?FFFFh' to '0h. '

HcPeriodicStart Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			PS													
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register: HcPeriodicStart **Offset: 40 to 43**

Bits	Initial Value	R/W	Description
31 to 14	0	R	Reserved. Read/Write 0's
13 to 0	0	R/W	**PeriodicStart** This field contains a value used by the List Processor to determine where in a frame the Periodic List processing must begin.

HcLSThreshold Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					LST											
Initial:	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register: HcLSThreshold			Offset: 44 to 47
Bits	**Initial Value**	**R/W**	**Description**
31 to 12	0	R	Reserved. Read/Write 0's
11 to 0	H'628	R/W	**LSThreshold** This field contains a value used by the Frame Management block to determine whether or not a low speed transaction can be started in the current frame.

HcRhDescriptorA Register

This register is only reset by a power-on reset ($\overline{\text{PCIRST}}$). It is written during system initialization to configure the Root Hub. These bits should not be written during normal operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	POTPGT															
Initial:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				NOCP	OCPM	DT	NPS	PSM	NDP							
Initial:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R/W	R/W	R	R/W	R/W	R	R	R	R	R	R	R	R

Register: HcRhDescriptorA			Offset: 48 to 4B
Bits	**Initial Value**	**R/W**	**Description**
31 to 24	H'02	R/W	**PowerOnToPowerGoodTime** USB Host Controller power switching is effective within 2 ms. The field value is represented as the number of 2 ms intervals. Only bits [25:24] are implemented as R/W. The remaining bits are read only as '0'. It is not expected that these bits be written to anything other than 1h, but limited adjustment is provided. This field should be written to support the system implementation. This field should always be written to a non-zero value.

Register: HcRhDescriptorA			Offset: 48 to 4B
Bits	**Initial Value**	**R/W**	**Description**
23 to 13	0	R	Reserved. Read/Write 0's
12	1	R/W	**NoOverCurrentProtection** USB Host Controller implements global over-current reporting 0: Over-current status is reported 1: Over-current status is not reported This bit should be written to support the external system port over-current implementation.
11	0	R/W	**OverCurrentProtectionMode** USB Host Controller implements global over-current reporting 0: Global Over-Current 1: Individual Over-Current This bit is only valid when **NoOverCurrentProtection** is cleared. This bit should be written '0'.
10	0	R	**DeviceType** USB Host Controller is not a compound device.
9	1	R/W	**NoPowerSwitching** USB Host Controller implements global power switching. 0: Ports are power switched. 1: Ports are always powered on. This bit should be written to support the external system port power switching implementation.
8	0	R/W	**PowerSwitchingMode** USB Host Controller implements a global power switching mode. 0: Global Switching 1: Individual Switching This bit is only valid when **NoPowerSwitching** is cleared. This bit should be written '0'.
7 to 0	H'02	R	**NumberDownstreamPorts** USB Host Controller supports two downstream ports.

HcRhDescriptorB Register

This register is only reset by a power-on reset ($\overline{\text{PCIRST}}$). It is written during system initialization to configure the Root Hub. These bits should not be written during normal operation.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PPCM															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register: HcRhDescriptorB **Offset: 4C to 4F**

Bits	Initial Value	R/W	Description
31 to 16	0	R/W	**PortPowerControlMask** USB Host Controller implements global-power switching. This field is only valid if **NoPowerSwitching** is cleared and **PowerSwitchingMode** is set (individual port switching). When set, the port only responds to individual port power switching commands (**Set**/**ClearPortPower**). When cleared, the port only responds to global power switching commands (**Set**/**ClearGlobalPower**). 0: Device not removable 1: Global-power mask Port Bit relationship 0: Reserved 1: Port 1 2: Port 2 : 15: Port 15 Unimplemented ports are reserved, read/write '0'.

Register: HcRhDescriptorB			Offset: 4C to 4F
Bits	**Initial Value**	**R/W**	**Description**
15 to 0	0	R/W	**DeviceRemovable** USB Host Controller ports default to removable devices. 0: Device removable 1: Device not removable Port Bit relationship 0: Reserved 1: Port 1 2: Port 2 : 15: Port 15 Unimplemented ports are reserved, read/write '0'.

HcRhStatus Register

This register is reset by the USB RESET state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CRWE														OCIC	LPSC / SGP
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRWE/SRWE														OCI	LPS / CGP
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Register: HcRhStatus			Offset: 50 to 53
Bits	**Initial value**	**R/W**	**Description**
31	—	W	(write) **ClearRemoteWakeupEnable** Writing a '1' to this bit clears **DeviceRemoteWakeupEnable**. Writing a '0' has no effect.
30 to 18	0	R	Reserved. Read/Write 0's
17	0	R/W	**OverCurrentIndicatorChange** This bit is set when **OverCurrentIndicator** changes. Writing a '1' clears this bit. Writing a '0' has no effect.
16	0	R/W	(read) **LocalPowerStatusChange** Not supported. Always read '0'. (write) **SetGlobalPower** Write a '1' issues a **SetGlobalPower** command to the ports. Writing a '0' has no effect.
15	0	R/W	(read) **DeviceRemoteWakeupEnable** This bit enables ports' **ConnectStatusChange** as a remote wakeup event. 0: disabled 1: enabled (write) **SetRemoteWakeupEnable** Writing a '1' sets **DeviceRemoteWakeupEnable**. Writing a '0' has no effect.
14 to 2	0	R	Reserved. Read/Write 0's
1	0	R	**OverCurrentIndicator** This bit reflects the state of the OVRCUR pin. This field is only valid if **NoOverCurrentProtection** and **OverCurrentProtectionMode** are cleared. 0: No over-current condition 1: Over-current condition
0	0	R/W	(read) **LocalPowerStatus** Not Supported. Always read as '0'. (write) **ClearGlobalPower** Writing a '1' issues a **ClearGlobalPower** command to the ports. Writing a '0' has no effect.

HcRhPortStatus[1:2] Register

This register is reset by the USB RESET state.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
												PRSC	OCIC	PSSC	PESC	CSC
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							LSDA /CPP	PPS / SPP				PRS / SPR	POCI / CSS	PSS / SPS	PES / SPE	CCS/ CPE
Initial:	0	0	0	0	0	0	1/0*	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Note: * This bit indicates the speed of the attached device.

Register: HcRhPortStatus[1:2] Offset: 54 to 57, 58 to 5B

Bits	Initial Value	R/W	Description
31 to 21	0	R	Reserved. Read/Write 0's
20	0	R/W	**PortResetStatusChange** This bit indicates that the port reset signal has completed. 0: Port reset is not complete. 1: Port reset is complete.
19	0	R/W	**PortOverCurrentIndicatorChange** This bit is set when **OverCurrentIndicator** changes. Writing a '1' clears this bit. Writing a '0' has no effect.
18	0	R/W	**PortSuspendStatusChange** This bit indicates the completion of the selective resume sequence for the port. 0: Port is not resumed. 1: Port resume is complete.
17	0	R/W	**PortEnableStatusChange** This bit indicates that the port has been disabled due to a hardware event (cleared **PortEnableStatus**). 0: Port has not been disabled. 1: PortEnableStatus has been cleared.

Register: HcRhPortStatus[1:2] Offset: 54 to 57, 58 to 5B

Bits	Initial Value	R/W	Description
16	0	R/W	**ConnectStatusChange** This bit indicates a connection or disconnection event has been detected. Writing a '1' clears this bit. Writing a '0' has no effect. 0: No connect/disconnect event. 1: Hardware detection of connect/disconnect event. Note: If DeviceRemoveable is set, this bit resets to '1'.
15 to 10	0	R	Reserved. Read/Write 0's
9	1/0	R/W	(read) **LowSpeedDeviceAttached** This bit defines the speed (and bud idle) of the attached device. It is only valid when **CurrentConnectStatus** is set. 0: Full Speed device 1: Low Speed device (write) **ClearPortPower** Writing a '1' clears **PortPowerStatus**. Writing a '0' has no effect
8	1	R/W	(read) **PortPowerStatus** This bit reflects the power state of the port regardless of the power switching mode. 0: Port power is off. 1: Port power is on. Note: If **NoPowerSwitching** is set, this bit is always read as '1'. (write) **SetPortPower** Writing a '1' sets **PortPowerStatus**. Writing a '0' has no effect.
7 to 5	0	R	Reserved. Read/Write 0's
4	0	R/W	(read) **PortResetStatus** 0: Port reset signal is not active. 1: Port reset signal is active. (write) **SetPortReset** Writing a '1' sets PortResetStatus. Writing a '0' has no effect.

Register: HcRhPortStatus[1:2] Offset: 54 to 57, 58 to 5B

Bits	Initial Value	R/W	Description
3	0	R/W	(read) **PortOverCurrentIndicator** USB Host Controller supports global over-current reporting. This bit reflects the state of the OVRCUR pin dedicated to this port. This field is only valid if NoOverCurrentProtection is cleared and OverCurrentProtectionMode is set. 0: No over-current condition 1: Over-current condition (write) **ClearSuspendStatus** Writing a '1' initiates the selective resume sequence for the port. Writing a '0' has no effect.
2	0	R/W	(read) **PortSuspendStatus** 0: Port is not suspended 1: Port is selectively suspended (write) **SetPortSuspend** Writing a '1' sets PortSuspendStatus. Writing a '0' has no effect.
1	0	R/W	(read) **PortEnableStatus** 0: Port disabled. 1: Port enabled. (write) **SetPortEnable** Writing a '1' sets PortEnableStatus. Writing a '0' has no effect.
0	0	R/W	(read) **CurrentConnectStatus** 0: No device connected. 1: Device connected. Note: If DeviceRemoveable is set (not removable) this bit is always '1'. (write) **ClearPortEnable** Writing a '1' clears PortEnableStatus. Writing a '0' has no effect.

ConfigurationControl Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BA															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																P2S
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Register: ConfigurationControl Offset: F0 to F3

Bits	Initial Value	R/W	Description
31	0	R/W	**Bus Alignment** 0: Little Endian 1: Big Endian In Little Endian, data is inverted in a byte unit. In Big Endian, data is not inverted. Refer to Fig.24.2
30 to 1	0	R	Reserved. Read/Write 0's
0	0	R/W	**Port2Switch** 0: Host 1: Function

Notes: When the function module and the host module are switched by Port2Switch, if USB2PENC pin is used, please keep the procedure below.

Situation	Procedure
Switch to function module (This module is selected in USB Host as default value after power-on sequence)	Two cases can be seleced as follows: 1. Set Port2Switch to 1 after that set USB function module registers. 2. Set Port2Switch to 1 after that set PULLUPE bit to 1. Next, set PULLUPE bit to 0 after that set USB function module registers.

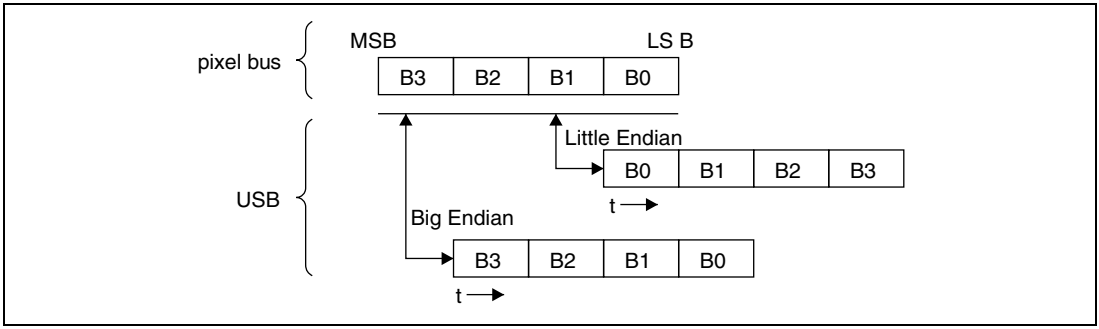


Figure 24.2 Endian Data Flow

24.5 Functional Description

24.5.1 General Functionality

USB Host

The USB Host includes the integrated Root Hub with two external ports, Port 1 and Port 2 as well as the List Processing (LP), the Serial Interface Engine (SIE) and USB clock generator. The interface combines responsibility for executing bus transactions requested by the HC as well as the hub and port management specified by USB. Application i/f converts HCI i/f to register_bus i/f and Pixel_bus i/f. USB Host supports OHCI Operational Registers. Data transfer appears on pixel_bus i/f between Graphics Memory and USB host. Registers in USB host is controlled via register_bus i/f. Endpoint Descriptor(ED) and Transfer Descriptor(TD) need to be stored in Graphic Memory before the data transaction begins.

List Processing: The List Processor consists of four main blocks. The four blocks are the List Control block, the ED block, the TD block, and the Request block. The first three blocks operate in a lock step fashion with the List Control block triggering the ED block, which in turn triggers the TD block. These blocks are responsible for issuing their own bus master requests to the Request block which interfaces to the Host Controller Bus Master.

Serial Interface Engine (SIE): The SIE is responsible for managing all transactions to the USB. It controls the bus protocol, packet generation/extraction, data parallel-to-serial conversion, CRC coding, bit stuffing, and NRZI encoding.

All transactions on the USB are requested by the List Processor and Frame Manager. After the List Processor retrieves all information necessary to initiate communication to a USB device, it generates a request to the SIE accompanied by endpoint specific control information required to generate proper protocol and packet formats to establish the desired communication pipe. The data buffer provides a data path for the data packets and controls the number of bytes transferred.

The FM generates SOF events each millisecond for which the SIE generates an SOF token. The List Processor requests are ignored to allow the SOF to be serviced with the highest priority and no delay.

Root Hub: The Root Hub is a collection of ports which are individually controlled and a hub which maintains control/status over functions common to all ports. The typical command request interface to the hub is emulated by the HCD which communicates directly through the system bus (PCI) to the hub and port controls. The remainder of this section will divide the discussion into hub and port design responsibilities.

The Root Hub descriptor registers, HcRhDescriptorA and HcRhDescriptorB, are implemented R/W to allow multiple configuration with minimal changes to the current implementation.

Hub and port control and status are implemented through the HcRhStatus and HcRhPortStatus Registers. Each port has its own HcRhPortStatus Registers. A command structure is defined through these registers which software uses to control the hub and ports. By writing a '1' to bit locations specified in register description Sections. The following commands summarized in can be executed. The command? behavior is discussed in the sections below

Hub Control: The HC states also reflect the hub state. For example, when the HC is suspended, USB SUSPEND, the Root Hub is suspended. When the HC is in USB RESUME, the hub generates the appropriate bus signaling. USB RESET resets the Root Hub. The following sections describe hub and bus related controls and status.

Port Control: The Port is responsible for all activity associated with driving and monitoring bus states. The HCD controls this behavior through the register command interface.

Clock Generation: The USB interface is sourced by a 48 MHz clock which allows for a 4x data rate oversampling to maintain the receiver phase lock. This clock also sources all USB related clock rates (12 MHz and 1.5 MHz).

- Static SOF Clock

  As the USB system host, the system frame counter is maintained at a constant 1 ms interval. This requires a static 12 MHz clock. This is created by dividing down the 48 MHz internal clock source. The clock is enabled when the HC is not in the USB SUSPEND state.
- Data Rate Clock

  The SIE requires that the transmit and receive clocks operate at 12 and 1.5 MHz. During FS transmissions, the data rate clock is equivalent to the static 12 MHz SOF clock. When the SIE has a LS packet the data rate clock must be changed to 1.5 MHz following the preamble token. The 1.5 MHz data rate is maintained until the response packet is concluded, if expected.

When receiving data, the data rate clock must match that of the source. Working in conjunction with the phase lock circuitry, the data rate clock is adjusted to maintain a 1 to 1 ratio of data bits and data clocks. This will result in periodic adjustment of the internal 48 MHz clock periods to maintain synchronization with the data source. When the packet is complete the data rate clock is re-linked to the static 12 MHz clock discussed above.

Module Standby Mode: This USB HOST module allows clock gating to reduce power consumption.

Both pixel bus clock and register bus clock can be gated. This module standby mode can be executed by controlling Clock Control 2 Register in Power Control & Configuration module.

To wake up the module, USB0 bit of Clock Control 2 Register must be enabled. After enabling this bit , initialisation of this module must be executed .

24.6 Connection Example of an External Circuit

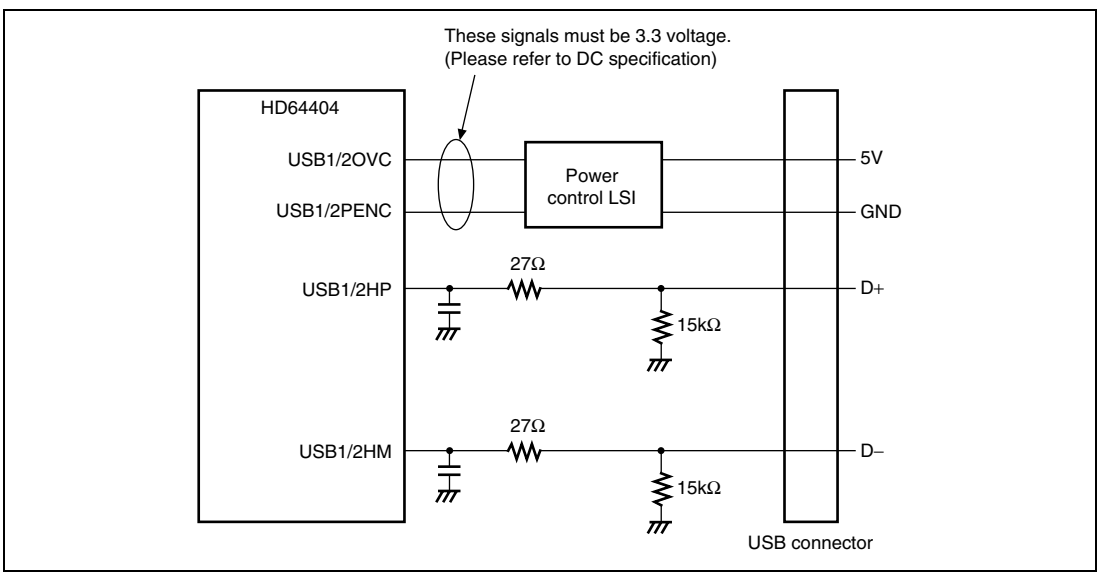


Figure 24.3 Connection Example of External Circuit

Section 25 Interrupt Input

25.1 General Description

The unit is 8 input interrupt controller for funnelling interrupts to the central interrupt controller. This allows interrupts generated outside HD64404 to be included in the HD64404 interrupt scheme. Each interrupt can be detected by edge or level. Each interrupt cause is stored in a status register. Each interrupt can be disabled by setting the control register.

25.2 Features

- Interrupt enable for each input
- Positive or negative edge detect
- Level or edge sensitive

25.3 Interface

The following table lists the digital interface pins and their functions:

Table 25.1 Digital Block Interface Signals and Pin List

Signal or Pin Name	No. of Bits	In/Out	Function	To/From	Synchronization to Clocks
irq	1	Out	Interrupt active	Interrupt Priority	rbclk
INT(7:0)	8	In	Input interrupts	External	—

25.4 Block Diagram

Figure 25.1 shows a Block Diagram of Interrupt Input.

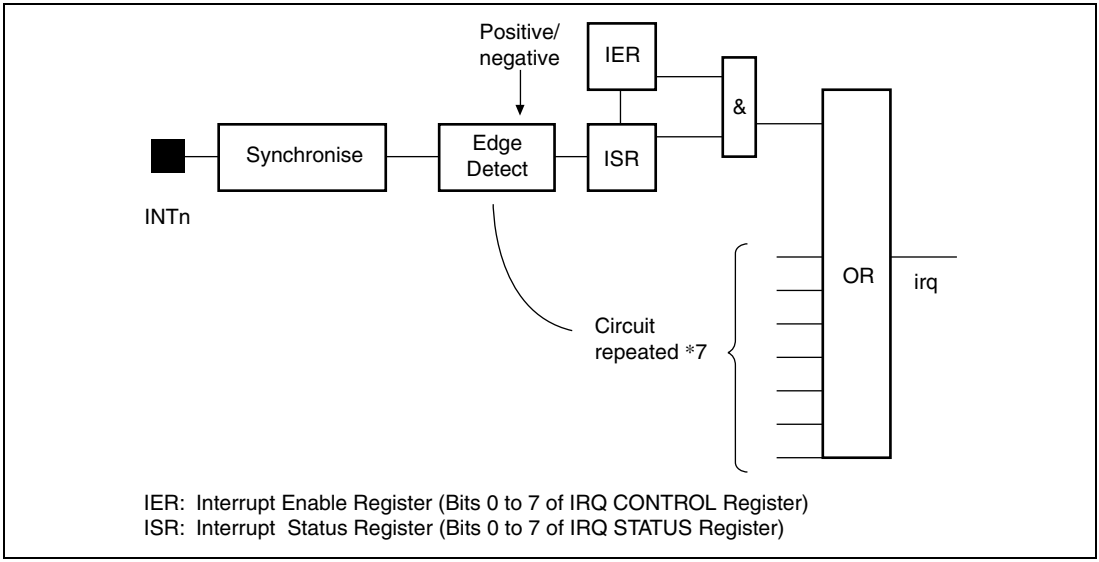


Figure 25.1 Interrupt Input Block Diagram

25.5 Register Description

There are set of registers which are located in the address space of the PCI or MPX bus and are located in the PCI memory window.

Table 25.2 Register List

Address (Bytes)	Register Name	Mnemonic or Symbol	R/W	Access Size
H'6184	IRQ STATUS	IRQS	R/WC0	32
H'6188	IRQ CONTROL	IRQC	R/W	32

Legends for register description:

Initial Value : Register value after reset
— : Undefined value
R/W : Read and Write, write value can be read.
R : Read only, for write always 0 write
R/WC0 : Read and Write, 0 write clear, 1 write is ignored
R/WC1 : Read and Write, 1 write clear, 0 write is ignored

W : Write only, Read prohibited. If reserved, write always 0.
—/W : Write only, Read value undefined.

R/(W)* : (writing 0's clears status bits, writing 1's does not change status bits)

25.5.1 IRQ Status Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									IRQ_ST							
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	—	R	**Reserved**
7 to 0	IRQ_ST	0	R/WC0	**IRQ_ST** Indicates whether an interrupt has been received on that channel. The status bit is set regardless of the state of the interrupt_enable.

25.5.2 IRQ Control

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
								STBY	IRQ_POLARITY							
Initial:	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IRQ_EDGE								IRQ_EN							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	—	R	**Reserved**
24	STBY	0	R/W	**IRQ_STANDBY (STBY)** When the mode is set to standby, the clock to the module can be stopped. For those interrupts set to level sensitive, an input of the active sense generates an interrupt on its output. Stopping the clocks does not affect any of the register settings. 0: Standby disabled 1: Standby enabled.
23 to 16	IRQ_POLARITY	0	R/W	**IRQ_POLARITY** Each bit corresponds to the appropriate input pin and controls whether in level sensitive mode a high or low value sets the status bit and in edge sensitive mode whether a positive or negative transition on the input pin causes the interrupt status bit to be set. 0: High level/Positive edge sets the IRQ_ST bit 1: Low level/Negative edge sets the IRQ_ST bit
15 to 8	IRQ_EDGE	0	R/W	**IRQ_LEVEL_EDGE** Each bit corresponds to the appropriate input pin and controls whether the interrupt status is set if the input is at the active level or an active edge is detected. 0: Status bit is set if active level is detected 1: Status bit is set if active edge is detected
7 to 0	IRQ_EN	0	R/W	**IRQ_EN** Each bit corresponds to the appropriate input pin and allows the interrupt on that channel to cause an interrupt to the central interrupt controller. 0: Interrupt disabled 1: Interrupt enabled

Note: When changing attributes described above, software should do this change atomically, i.e. not allowing another interrupt on this module during the change in order to keep the same IRQ_EN flag values. Utilize some sort of mutual exclusion primitive prepared in OS for this purpose.

25.6 Functional Description

25.6.1 General Functionality

The interrupt input module responds to 8 input pins and is responsible for detecting an edge on the input, which indicates an interrupt. The polarity of the edge is programmable through the IRQ Control Register.

The input pin is synchronised to the register bus clock to remove meta-stable conditions before the edge is detected. As this is a synchronous detect, the minimum pulse to detect an edge is 2 clocks cycles of the register bus, i.e. 60 ns at 33 MHz.

The circuitry can be programmed to be sensitive to either an input level or an input edge. This is programmed through the IRQ_LEVEL_EDGE bits. If an active level or edge is detected then the corresponding bit in the status register is set. If the corresponding bit in the irq_en field in the IRQ Control Register is set then the output interrupt line to the central interrupt_controller sets.

Once the bit in status register is set by detecting the level or edge if the input source is then removed, status is latched till it is cleared.

Once the bit in Status Register is set by detecting the level if the input source is still at the active level then status register can not be cleared.

Writing a 0 to the correct bit clears the interrupt_status, writing a 1 has no effect.

Note once the detection recognition is defined that should not be changed in normal operation, which can cause duplicate interrupt.

25.6.2 Register Bus

The unit is programmable through the register bus and all accesses are 32 bits.

25.6.3 Standby mode

This module allows clock gating to reduce power consumption. The module standby mode can be executed by controlling bit 15 in the Clock Control 1 (CC1) Register.

To wake up the module, bit 15 in the Clock Control 1 (CC1) Register must be enabled and bit 24 in the IRQ Control Register disabled. After enabling this bit all access to the timer module can be possible.

To power down the module, the following procedure is required.

1. Set the IRQ_LEVEL_EDGE bits in the IRQ Control Register to be active edge.
2. Set the standby bit in the IRQ Control Register.
3. Disable bit 15 in the Clock Control 1 (CC1) Register.

Section 26 Timer/Counter

26.1 General Description

The unit has 2 major modes of operation. It can be configured as a four-channel timer/counter unit that contains a 32 bit free running timer as a common time-stamp for four 32-bit capture/compare registers.

Alternatively it can be configured as four 16-bit timer/counters with count enables. In this mode there are four independent 16-bit incrementing/decrementing blocks. Each channel can then be set up to output a signal when the compare time is reached or to store the timer value when an input edge is received. This latter case also supports up/down counting on some channels.

26.2 Features

- 32-bit free running timer
- 4 channels of Output compare or Input capture
- 4-channel 16-bit counters/timers
- Interrupt on capture, compare and overflow
- Programmable pin/edge polarity
- Programmable timer clock
- Independent clocks for 16 bit timers
- Support for rotary switches

26.3 Timer/Counter Interface

Table 26.1 Timer/Counter Interface

Signal or Pin	No. of Pin	Function	Direction
pin_ip	4	Multi-function timer/counter pin input	IN
pin_op	4	Multi-function timer/counter pin output	OUT
pin_en	4	Timer/counter pin enables	OUT
Register Bus		Access to registers	
irq	1	Interrupt line	OUT

26.4 Address Map

The module byte base address is H'06100.

Table 26.2 Address Map

Address (Bytes)	Register Name	Access Size
H'6100	Config	32
H'6104	Free Running Timer	32
H'6108	Control	32
H'610C	IRQ Status	32
H'6110	Channel0 Time	32
H'6114	Channel1 Time	32
H'6118	Channel2 Time	32
H'611C	Channel3 Time	32
H'6120	Channel0 Stop Time	32
H'6124	Channel1 Stop Time	32
H'6128	Channel2 Stop Time	32
H'612C	Channel3 Stop Time	32
H'6130	Channel0 Counter	32
H'6134	Channel1 Counter	32
H'6138	Channel2 Counter	32
H'613C	Channel3 Counter	32

26.5 Block Diagram

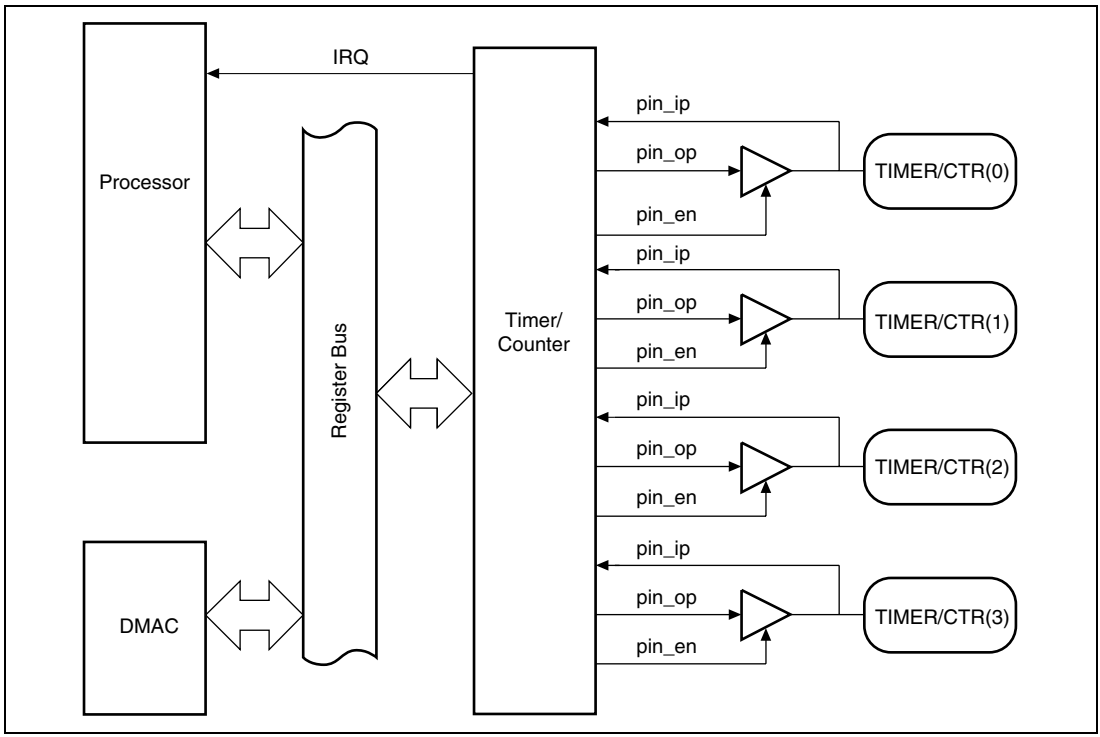


Figure 26.1 Timer Interface Block Diagram.

26.6 Register Description

The registers are located in the address space of the PCI or MPX bus, in the memory window.

Note: Where bits are defined as reserved then only 0's should be written and the value read is indeterminate.

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

26.6.1 Config Register

The possible operations for a pin are timer compare, timer input capture, Up or down count and capture input, where one pin is used for the capture while a second is used to enable the count.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
															ROT2	ROT0
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ED3		ED2		ED1		ED0			FRCM	FRTM	T23			T01	
Initial:	0	0	0	0	0	0	0	0	-	0	0		0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	—	R	**Reserved**
17	ROT2	0	R/W	**Channel 2, 3 rotation enable (ROT2)** Only set when operating in updowncounter mode (FRTM = 0, T01 = 11 and T23 = 011), otherwise this bit is disabled (ROT2 = 0). When set this indicates channel2 and channel3 pins are operating in rotary mode. This is an encoding of the pin pair to generate up and down signals to the counter. Counter 3 needs to be disabled (TE3 = 0, in the Control Register).
16	ROT0	0	R/W	**Channel 0,1 rotation enable (ROT0)** Only set when operating in up down counter mode (FRTM = 0, T01 = 11 and T23 = 011), otherwise this bit is disabled (ROT0 = 0). When set this indicates channel0 and channel1 pins are operating in rotary mode. This is an encoding of the pin pair to generate up and down signals to the counter. Counter 1 needs to be disabled (TE1 = 0, in the Control Register).

Bit	Bit Name	Initial Value	R/W	Description
15, 14	ED3	0	R/W	**Channel 3 pin active control (ED3)** In input mode these bits indicates the following 00: Edge detect on channel3 disabled 01: Edge detect on rising edge of channel3 input 10: Edge detect on falling edge of channel3 input 11: Edge detect on either edge on channel3 input In output mode this bit indicates the following 00: Reserved 01: Output for channel3 is 1 for active period 10: Output for channel3 is 0 for active period 11: Reserved
13, 12	ED2	0	R/W	**Channel 2 pin active control (ED2)** In input mode these bits indicates the following 00: Edge detect on channel2 disabled 01: Edge detect on rising edge of channel2 input 10: Edge detect on falling edge of channel2 input 11: Edge detect on either edge on channel2 input In output mode this bit indicates the following 00: Reserved 01: Output for channel2 is 1 for active period 10: Output for channel2 is 0 for active period 11: Reserved
11, 10	ED1	0	R/W	**Channel 1 pin active control (ED1)** In input mode these bits indicates the following 00: Edge detect on channel1 disabled 01: Edge detect on rising edge of channel1 input 10: Edge detect on falling edge of channel1 input 11: Edge detect on either edge on channel1 input In output mode this bit indicates the following 00: Reserved 01: Output for channel1 is 1 for active period 10: Output for channel1 is 0 for active period 11: Reserved

Bit	Bit Name	Initial Value	R/W	Description
9, 8	ED0	0	R/W	**Channel 0 pin active control (ED0)** In input mode these bits indicates the following 00: Edge detect on channel0 disabled 01: Edge detect on rising edge of channel0 input 10: Edge detect on falling edge of channel0 input 11: Edge detect on either edge on channel0 input In output mode this bit indicates the following 00: Reserved 01: Output for channel0 is 1 for active period 10: Output for channel0 is 0 for active period 11: Reserved
7	—	—	R	**Reserved**
6	FRCM	0	R/W	**Free running control mode (FRCM)** When bits T23, in 16 bit mode, are set to 100 (Capture input with upcounter), this bit determines whether the up counter use a Free Running Counter or input capture on channel 3. 0: External clock (Upcounter with input capture) 1: Internal clock (Free Running Upcounter)
5	FRTM	0	R/W	**Free running timer mode (FRTM)** Determines whether the timer works as a common 32 bit free running timer or four independent 16-bit timer/counters. Setting this bit to '1' will override the setting of bits 4 to 0 in the Config Register. 0: 16 bit mode 1: 32 bit FRT mode

Bit	Bit Name	Initial Value	R/W	Description
4 to 2	T23	0	R/W	**Timer 2, 3 configuration (T23)** These bits are only used in 16 bit mode (i.e. FRTM = 0). These bits control the use of pins 2 and 3 Configuration modes for Channel 2 & 3. 000: Timer2 and Timer3 001: Upcounter2 and Timer3 010: Upcounter2 and Upcounter3 011: Updowncounter2 100: Capture input with upcounter 101: Reserved 110: Reserved 111: Reserved Note: Upcounter2 is a sub-set of Updowncounter2
1, 0	T01	0	R/W	**Timer 0, 1 configuration (T01)** These bits are only used in 16 bit mode (i.e. FRTM = 0). These bits control the use of pins 0 and 1. Pin0 is mapped to timer0/counter 0 and pin1 is mapped to timer1/counter1 Configuration modes for Channel 0 & 1. 00: Timer0 and Timer1 01: Upcounter0 and Timer1 10: Upcounter0 and Upcounter1 11: Updowncounter0 Note: Upcounter0 is a sub-set of Updowncounter0

26.6.2 Free Running Timer

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Free running timer															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Free running timer															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	Free running timer	0	R	**Free running timer (FRT)** Returns the current value of the Free Running Timer (FRT).

26.6.3 Control Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TE3	TE2	TE1	TE0	IOE3	IOE2	IOE1	IOE0	ICE3	ICE2	ICE1	ICE0	IEE3	IEE2	IEE1	IEE0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CC3		CC2		CC1		CC0		SI3	SI2	SI1	SI0	OP3	OP2	OP1	OP0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	TE3	0	R/W	**Timer enable (TE3 to 0)**
30	TE2	0	R/W	Enables the counting of each of the 16 bit counters. When these bits are inactive the counters are reset to 0 when operating in timer mode or in counter mode.
29	TE1	0	R/W	
28	TE0	0	R/W	In updowncounter mode a counter for each pair needs to be disabled, respectively Counter 1 and Counter 3 (TE1 = 0 and TE3 = 0).
				0: Counting disabled, counter will be reset to H'000 1: Counter will increment.
27	IOE3	0	R/W	**Channel 3 to 0 interrupt overflow enable (IOE3 to 0)**
26	IOE2	0	R/W	Enables an interrupt to be generated when the relevant IOx bit is set in the IRQ Status Register.
25	IOE1	0	R/W	
24	IOE0	0	R/W	0: disabled 1: enabled
23	ICE3	0	R/W	**Channel 3 to 0 interrupt compare enable (ICE3 to 0)**
22	ICE2	0	R/W	Enables an interrupt to be generated when the relevant ICx bit is set in the IRQ Status Register.
21	ICE1	0	R/W	
20	ICE0	0	R/W	0: disabled 1: enabled
19	IEE3	0	R/W	**Channel 3 to 0 interrupt edge enable (IEE3 to 0)**
18	IEE2	0	R/W	Enables an interrupt to be generated when the relevant IEx bit is set in the IRQ Status register.
17	IEE1	0	R/W	
16	IEE0	0	R/W	0: disabled 1: enabled Note: When a channel is in output mode, the corresponding IEE has to be set to "0".
15, 14	CC3	0	R/W	**Timer Clock control-channel3 (CC3)**
				These bits specify the clock input for the channel 3 16-bit counter/timer. 00: Clock for timer3 is 1/32 of source clock. 01: Clock for timer3 is 1/128 of source clock. 10: Clock for timer3 is 1/512 of source clock. 11: Clock for timer3 is 1/1024 of source clock. Set the same value as CCO when using 16-bit input capture mode.

Bit	Bit Name	Initial Value	R/W	Description
13, 12	CC2	0	R/W	**Timer Clock control-channel2 (CC2)** These bits specify the clock input for the channel 2 16-bit counter/timer. 00: Clock for timer2 is 1/32 of source clock. 01: Clock for timer2 is 1/128 of source clock. 10: Clock for timer2 is 1/512 of source clock. 11: Clock for timer2 is 1/1024 of source clock. Set the same value as CCO when using 16-bit input capture mode.
11, 10	CC1	0	R/W	**Timer Clock control-channel1 (CC1)** These bits specify the clock input for the channel 1 16-bit counter/timer. 00: Clock for timer1 is 1/32 of source clock. 01: Clock for timer1 is 1/128 of source clock. 10: Clock for timer1 is 1/512 of source clock. 11: Clock for timer1 is 1/1024 of source clock. Set the same value as CCO when using 16-bit input capture mode.
9, 8	CC0	0	R/W	**Free Running Timer Clock control (CC0)** This clock resolution for the timer/counters is derived from the register bus clock. The clock is pre-divided by 32 to generate approximately 1 MHz clock if the register bus clock is 33 MHz or another corresponding frequency for the register bus clock of other than 33 MHz. It can then be controlled to produce the following clocks. This clock is used for the free running timer and also for the channel0 16-bit counter/timer. 00: Clock for FRT and timer0 is 1/32 of source clock. 01: Clock for FRT and timer0 is 1/128 of source clock. 10: Clock for FRT and timer0 is 1/512 of source clock. 11: Clock for FRT and timer0 is 1/1024 of source clock.

Bit	Bit Name	Initial Value	R/W	Description
7	SI3	0	R/W	**Channel 3 to 0 stop ignore (SI3 to 0)**
6	SI2	0	R/W	For each channel, determines whether in output compare mode with 32 bit timer mode, the output remains active for half the maximum time or until the stop value is reached.
5	SI1	0	R/W	
4	SI0	0	R/W	0: Output remains active until the ChannelX Stop Time value is reached. 1: Output remains active for ½ the total duration of the FRT.
3	OP3	0	R/W	**Channel 3 to 0 operation. (OP3 to 0)**
2	OP2	0	R/W	For each channel, if in timer mode, whether the timer is used in output compare or input capture mode.
1	OP1	0	R/W	
0	OP0	0	R/W	0: input capture mode 1: output compare mode Note: When a channel is in output mode, the corresponding IEE has to be set to "0".

26.6.4 IRQ Status Register

These bits, once set, can only be cleared by a write. Only 0 can be written to these bits to clear the interrupt status bits. These conditions only create an interrupt if the relevant interrupt enable bit is set.

Reset Value: H'00000000

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					IO3	IO2	IO1	IO0	IC3	IC2	IC1	IC0	IE3	IE2	IE1	IE0
Initial:	-	-	-	-	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0	R / WC0

Bit	Bit Name	Initial Value	R/W	Description
31 to 12	—	—	R	**Reserved**
11	IO3	0	R/WC0	**Channel 3 to 0 interrupt overflow (IO3 to 0)**
10	IO2	0	R/WC0	A bit for each channel indicates if the upcounters or updowncounters have wrapped i.e. overflowed from H'FFFF to H'0000 or underflowed from H'0000 to H'FFFF 0: The count has not overflowed or underflowed. 1: The count has overflowed or underflowed.
9	IO1	0	R/WC0	
8	IO0	0	R/WC0	
7	IC3	0	R/WC0	**Channel 3 to 0 interrupt compare (IC3 to 0)**
6	IC2	0	R/WC0	A bit for each channel indicates whether in timer mode, the Free Running Timer has become equal to the channel times. 0: The timer has not become equal to the channel time value. 1: The timer has become equal to the channel time value.
5	IC1	0	R/WC0	
4	IC0	0	R/WC0	
3	IE3	0	R/WC0	**Channel 3 to 0 interrupt edge (IE3 to 0)**
2	IE2	0	R/WC0	A bit for each channel indicates whether an edge that will cause an action (active edge) has been detected. 0: Channel 3 to 0 has not received an active edge. 1: Channel 3 to 0 has received an active edge.
1	IE1	0	R/WC0	
0	IE0	0	R/WC0	

26.6.5 Channel 0 Time, Channel 1 Time, Channel 2 Time, Channel 3 Time Registers

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel X time															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel X time															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	Channel X time	0	R/W	**Channel X time** In output compare mode this register specifies the value to compare with the Free Running Timer. In input capture mode this register stores the free running timer value or the 16-bit timer values on the active edge of the input. Every time an edge is detected, the registers are updated and the new captured value will override the existing data value.

26.6.6 Channel 0 Stop Time, Channel 1 Stop Time, Channel 2 Stop Time, Channel 3 Stop Time Registers

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	Channel X stop time															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel X stop time															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	Channel X stop time	0	R/W	**Channel X stop time** In output compare mode this register specifies the value to compare with the free running timer. When this value is reached the TimerX output is reset to the inactive state.

26.6.7 Channel 0 Counter, Channel 1 Counter, Channel 2 Counter, Channel 3 Counter

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Channel X counter															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	—	R	**Reserved**
15 to 0	Channel X counter	0	R/W	**Channel X counter** A register for each channel indicates the current value of the counters. This register can be written to preload the counter. Reading these registers does not affect the count value in any way.

26.7 Functional Description

26.7.1 General Functionality

The timer unit consists of two sections of which the first is a central 32 bit free running timer for which the clock can be set to operate between approximately 1 MHz and 31.25 kHz. The second section consists of four sixteen bit counters with the ability to count up and for some the ability to count down. This counting is controlled through edge detection on the input pins. These can be used as counters which count based on inputs or can operate as timers with input capture/output compare. They differ from the FRT 32 bit timer in that they will reset to H'0000 when a capture or compare occurs on that channel.

The module consists of four channels, each of which can be configured as a timer or a counter. In timer mode, each of the four channels has two modes of operation. These are input capture and output compare.

26.7.2 Edge Detection

The timers and counters are based on edge detecting on the input pins. An active edge can be programmed to be either a rising, falling or both. In addition the edge detection logic can operate in rotary switch mode where the combination of two inputs indicate whether the switch is turning right or left, which will relate to an increment or decrement of the updowncounter. There is a pair of edge detectors that work on two inputs. The outputs can either work independently for the timers or the upcounters or can work as pairs to indicate up and down to the updowncounters.

In order for an edge to be detected the pulse must last for a minimum of 2 periods of the timer resolution for that channel, as shown in figure 26.2.

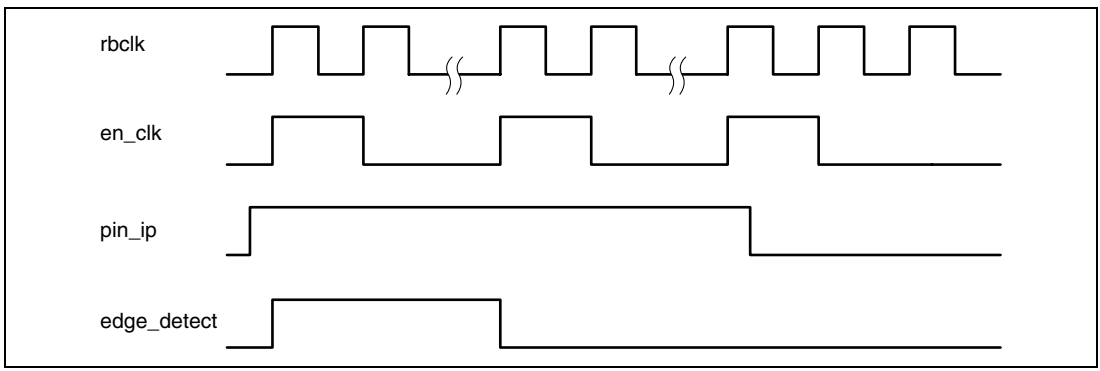


Figure 26.2 Edge Detection

Fig.26.2 – The above timing diagram shows an edge detection (rising edge). The input pin must be asserted for at least two clock resolution cycles and the edge_detect signal stays active till next en_clk pulse.

26.7.3 Timer 32 bit: Input Capture

When operating in input capture mode the channel will detect an edge on the input signal. The option of rising or falling edge is programmable. When this edge is detected the current value of the Free Running Timer (FRT) is captured in to the Channel Time Register for that channel. In addition the interrupt edge bit for that channel will be set and if the interrupt enable for that channel is set then an interrupt will be generated.

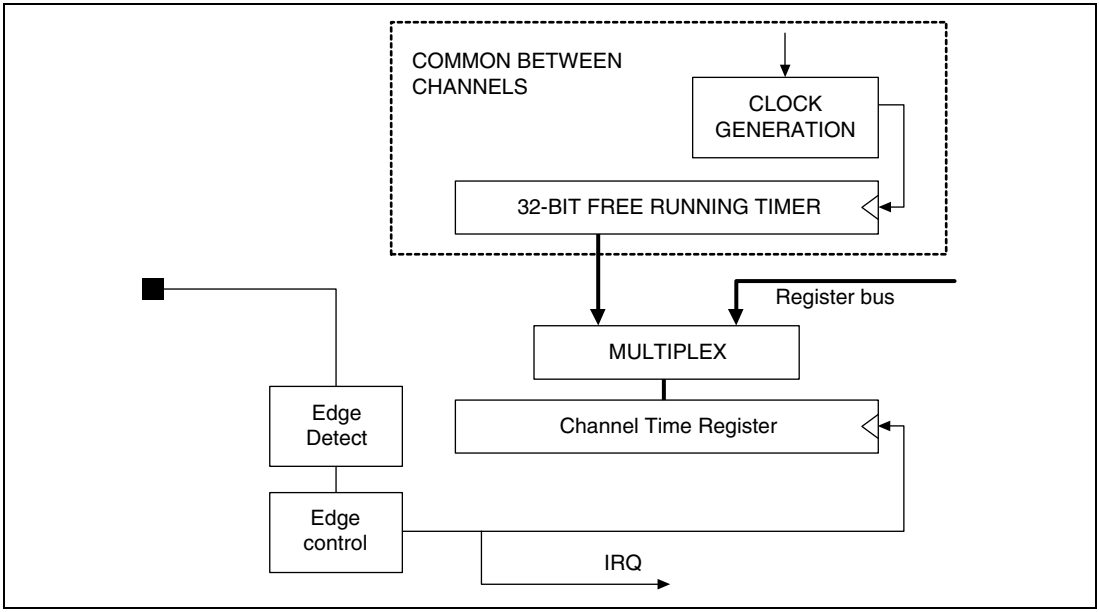


Figure 26.3 32-bit Timer Mode: Input Capture

26.7.4 Timer 32 bit: Output Compare

When operating in output compare the channel time register is compared to the FRT. When the free running timer becomes equal to the channel time register then the output will be set to the active state as defined in the pin active control bit. The output will remain in this state until the time is separated by half the maximum time from the channel time, as shown in figure 26.4, or it reaches the channelX Stop Time value dependent on the setting of the stop_ignore bits in the Control Register. At the point where the channel time register is equal to the FRT the interrupt compare bit will be set and an interrupt generated if the interrupt enable bit is set. When a channel is in output mode, the corresponding IEE has to be set to "0".

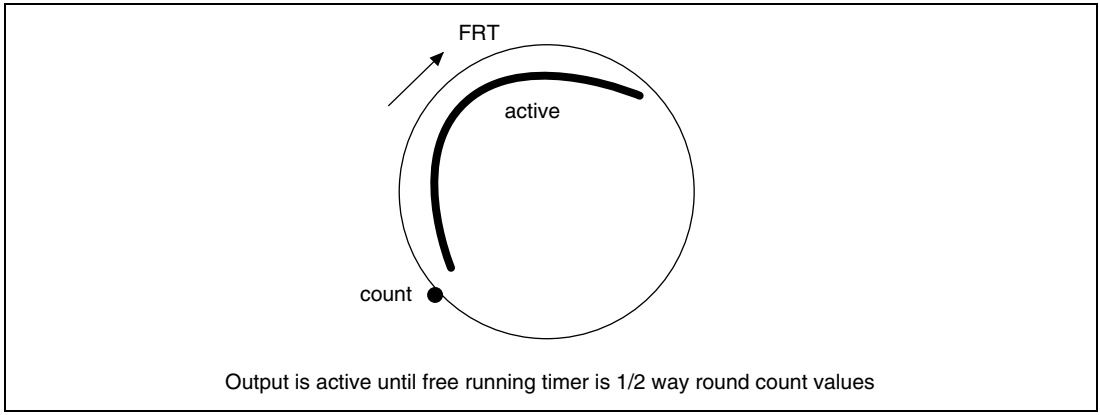


Figure 26.4 Output Pin Assertion Period

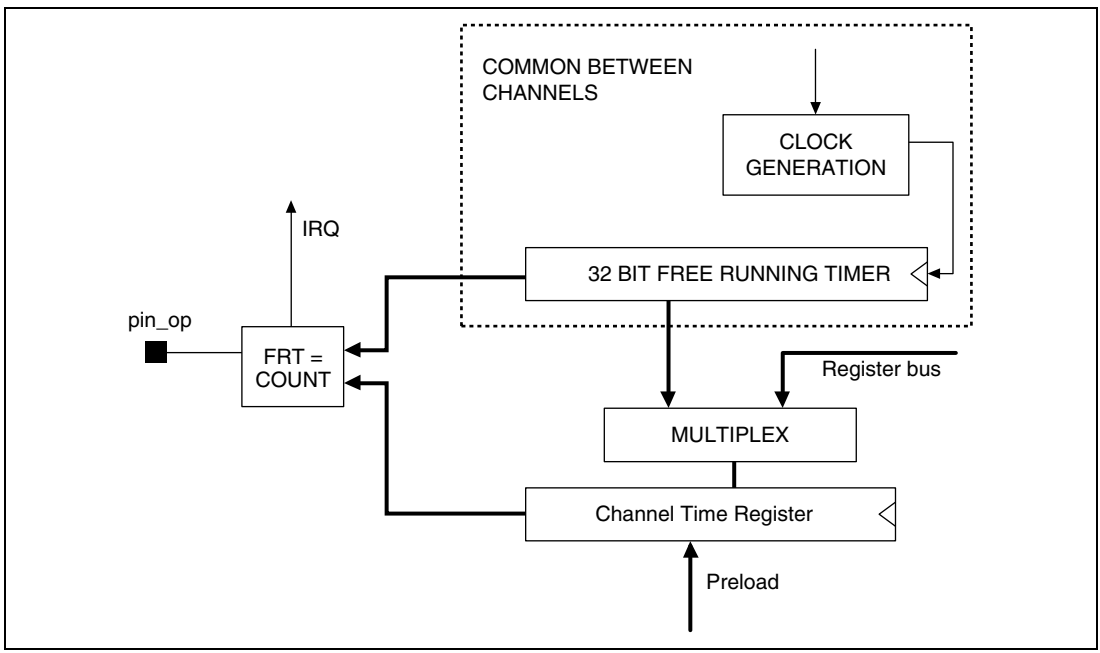


Figure 26.5 32-bit Timer Mode: Output Compare

26.7.5 Timer 16 bit: Input Capture

In this mode, the 16-bit counters will be free running using the clocks defined by Clock Control fields. When an active edge as defined by the Pin Active fields is received the Channel X Time will be set to the value of that channels 16 bit counter and the interrupt edge bit will be set. The counter will then reset to H'0000 and will begin counting again. The counters will remain or can be cleared to H'0000 by disabling the timer enable bits. At least 1 available channel, channel 0, for this mode. And another 3 channels will be available if it uses the same source clock as channel 0.

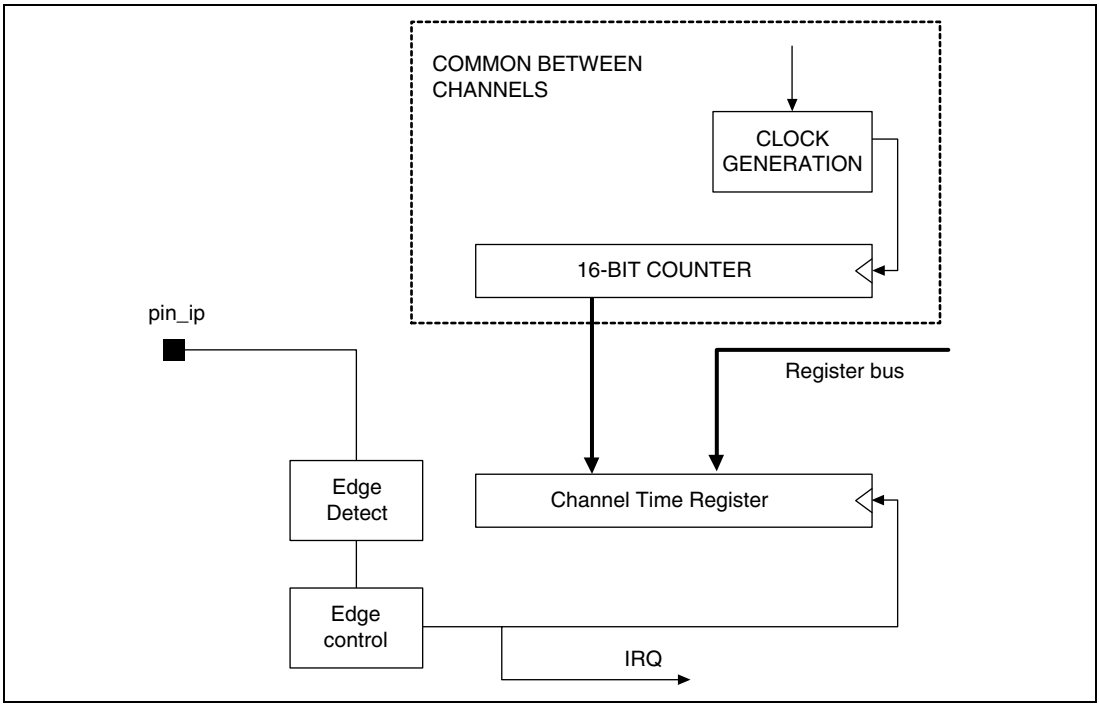


Figure 26.6 16 bit Timer Mode: Input captureS

26.7.6 Timer 16 bit: Output Compare

In this mode, the 16-bit counters will be free running using the clocks defined by Clock Control fields. Bits 15:0 of the Chanel X Time Register are compared with the 16 bit counter for that channel. When the values become equal, the output will invert from its current state (i.e. toggle) and the interrupt compare bit will be set. Please note that an interrupt for a compare match at 0x0000 is not generated. The counter will then reset to H'0000. The counter will begin counting again. Each time a match on the counter occurs the output will be toggled. The counters will remain or can be cleared to H'0000 by disabling the timer enable bits. When a channel is in output mode, the corresponding IEE has to be set to "0".

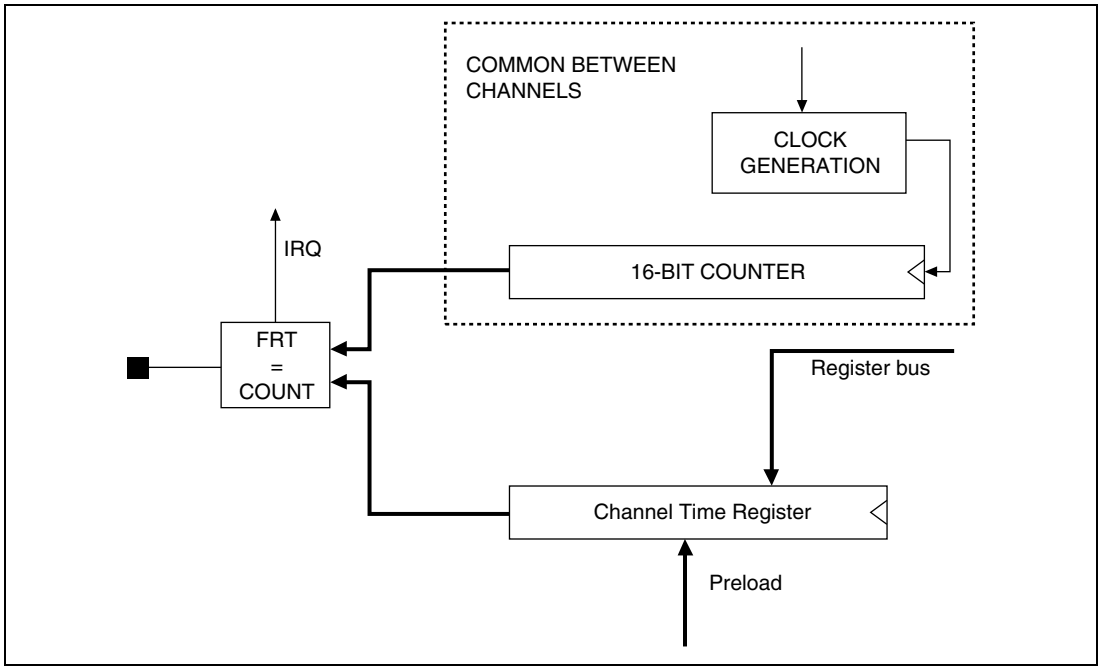


Figure 26.7 16 bit Timer Mode: Output Compare

26.7.7 Counter: Up/Updown Counter

Each of the pins can be connected to each of the four upcounters. These counters count up when an active edge is detected on the input pins. The counters can be written to by software to be preloaded and the current value can be read. Two of the upcounters can also be configured to count both up and down. For this two pins are required and so the second upcounter within the pair is not available. The pins are then referred to as up (pins 0 and 2) and down (pins 1 and 3). An active edge on these pins will cause the counter either to count up or down, or if both are active then the count will remain unchanged. In either upcounter or updown counter mode the counter will generate an interrupt if an edge is detected or if the count overflows or underflows.

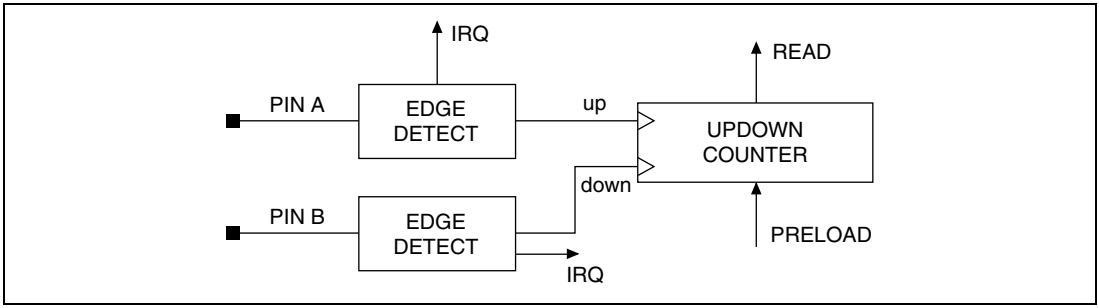


Figure 26.8 Updowncounter Mode

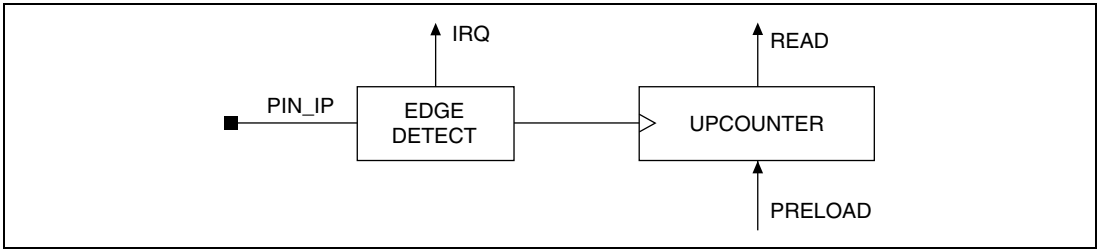


Figure 26.9 Upcounter Mode

26.7.8 Counter: Upcounter with Capture

In this mode, the 16-bit counter of channel 2 will operate either as free running upcounter or as upcounter with input capture, depending on the setting of FRCM bit. If FRCM is set to '0' the counter count up when an active edge is detected on the input pin of channel 3 and if FRCM is set '1' the counter is a free running counter. The counter will remain or can be cleared to H'0000 by disabling the timer enable bits.

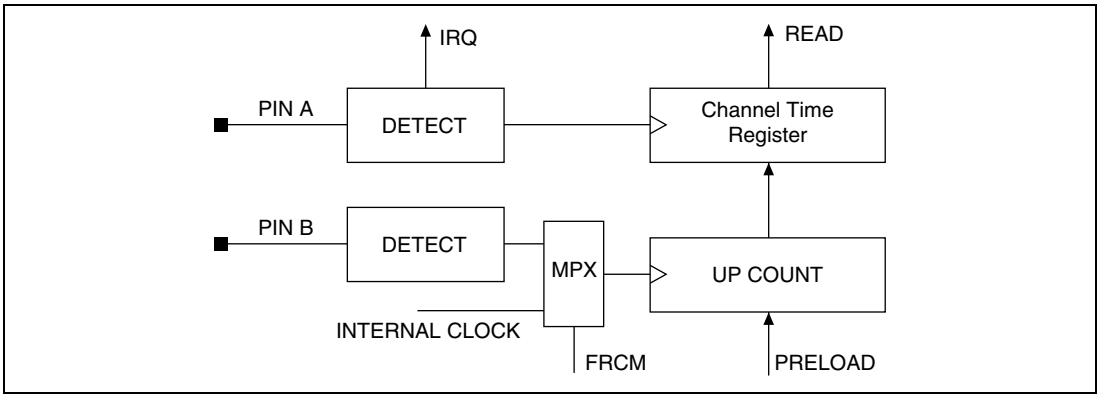


Figure 26.10 Upcounter with Capture Mode

26.7.9 Interrupts

The Status Register will have the interrupt status bits set for timer operation on either input capture or output compare regardless of the state of the interrupt enable bits. The counters will set an interrupt status bit if the count changes or the counter underflows or overflows. If the interrupt enable bit for a type of interrupt and the interrupt status bit of the same type for the same channel is set then an interrupt is generated.

i.e.

interrupt_channel <= (IE AND IEE) OR (IC AND ICE) OR (IO AND IOE)
interrupt <= OR (interrupt_channel(3:0))

26.7.10 Rotary mode

Each of the two updowncounters can operate in rotary mode. This treats the two input signals as encoded, as shown in figure 26.11. A rotary switch generates the following waveforms depending on direction. Thus the direction can be determined by the value of A when a falling edge is detected on the B input; if A is '1' the direction is left (down is '1') and if A is '0' the direction is right (up is '1'). A is pin 0 and pin 2. B is pin 1 and pin 3. The interrupt_edge status bit will be set whenever a change in the value of the counter occurs and if a counter overflow or underflow occurs the over/underflow interrupt status bit will be set as well.

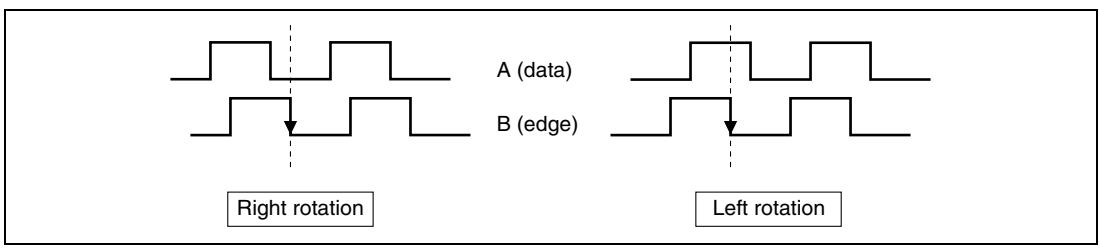


Figure 26.11 Rotary Mode

26.7.11 Timer Frequency

The frequency of the free running timer and the16-bit timers can be altered under software control to be 1 of 4 frequencies. Each 16-bit timer can have an independent clock.

26.7.12 Power Saving

To minimise power, each channel that is not used should be placed in input capture with the active edge bit set to "disabled".

26.7.13 Standby Mode

The Timer module allows clock gating to reduce power consumption. The module standby mode can be executed by controlling bit 20 in the Clock Control 1 (CC1) Register.

To wake up the module, bit 20 in the Clock Control 1 (CC1) Register must be enabled. After enabling this bit all access to the timer module can be possible.

To power down the module, the following procedure is required.

1. All channels needs to be in input capture mode (bit 3:0 to be set to 0).
2. The active edge for each channel needs to be disabled (ED0,1,2,3 to be set to 00).
3. Disable bit 20 in the Clock Control 1 (CC1) Register.

26.7.14 Register Bus

The unit will be programmable through the register bus and all accesses should be 32 bits.

Section 27 Pulse Width Modulation

27.1 General Description

This module provides four Pulse Width Modulation (PWM) channels.

27.2 Features

- Programmable high value and programmable cycle duration (8 bits).
- Programmable source clock frequency giving cycle time from 30ns with PCI bus or 20ns for MPX, to 2 minutes.
- Continuous or single shot

27.3 Interface

Table 27.1 PWM Interface

Signal	Function	Active	Direction
PWM(3:0)	Channel Outputs	N/A	OUT
Rba(1:0)	Register Address Bus	High	IN
Rbdi(31:0)	System programming bus	N/A	IN

27.4 Address Map

Table 27.2 PWM Register Map

Address (Bytes)	Register Name	Abbreviation	Access Size
H'66C0	PWM Control		32
H'66C4	PWM01 Counts		32
H'66C8	PWM23 Counts		32

27.5 Register Description

A set of registers are located in the address space of the PCI or MPX bus and are located in the PCI memory window.

Legends for register description:

Initial value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear , 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

27.5.1 PWM Control Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CC3				CC2				CC1				CC0			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									SS3	SS2	SS1	SS0	EN3	EN2	EN1	EN0
Initial:	-	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	CC3	0	R/W	**Clock control (CC0, CC1, CC2, CC3)**
27 to 24	CC2	0	R/W	The clock resolution for the PWM counters is from the register bus clock. It can then be divided to produce the following clocks. This is independent for each channel.
23 to 20	CC1	0	R/W	
19 to 16	CC0	0	R/W	
				Defining the register bus clock as AB MHz where AB can be up to 50 gives the following PWM frequencies.
				0000: Clock for timer is AB MHz 0001: Clock for timer is AB MHz/2^{2} 0010: Clock for timer is AB MHz/2^{4} 0011: Clock for timer is AB MHz/2^{6} 0100: Clock for timer is AB MHz/2^{8} 0101: Clock for timer is AB MHz/2^{10} 0110: Clock for timer is AB MHz/2^{12} 0111: Clock for timer is AB MHz/2^{14} 1000: Clock for timer is AB MHz/2^{16} 1001: Clock for timer is AB MHz/2^{18} 1010: Clock for timer is AB MHz/2^{20} 1011: Clock for timer is AB MHz/2^{22} 1100: Clock for timer is AB MHz/2^{24}
				If any of the values 1101, 1110, 1111 is entered, the resulting frequency will be AB MHz/2^{24}.
15 to 8	—	—	R	**Reserved**
7	SS3	0	R/W	**Channel 3 single shot (SS3)** 0: PWM3 channel operates in continuous mode 1: PWM3 channel will operate for one cycle and then stop
6	SS2	0	R/W	**Channel 2 single shot (SS2)** 0: PWM2 channel operates in continuous mode 1: PWM2 channel will operate for one cycle and then stop
5	SS1	0	R/W	**Channel 1 single shot (SS1)** 0: PWM1 channel operates in continuous mode 1: PWM1 channel will operate for one cycle and then stop
4	SS0	0	R/W	**Channel 0 single shot (SS0)** 0: PWM0 channel operates in continuous mode 1: PWM0 channel will operate for one cycle and then stop

Bit	Bit Name	Initial Value	R/W	Description
3	EN3	0	R/W	**Channel 3 enable (EN3)** 0: PWM3 remains in the idle state, output is high 1: PWM3 will oscillate between a high and a low state dependent on the count values This bit is cleared automatically in single shot mode.
2	EN2	0	R/W	**Channel 2 enable (EN2)** 0: PWM2 remains in the idle state, output is high 1: PWM2 will oscillate between a high and a low state dependent on the count values This bit is cleared automatically in single shot mode.
1	EN1	0	R/W	**Bit 1 Channel 1 enable (EN1)** 0: PWM1 remains in the idle state, output is high 1: PWM1 will oscillate between a high and a low state dependent on the count values This bit is cleared automatically in single shot mode.
0	EN0	0	R/W	**Channel 0 enable (EN0)** 0: PWM0 remains in the idle state, output is high 1: PWM0 will oscillate between a high and a low state dependent on the count values This bit is cleared automatically in single shot mode.

27.5.2 PWM01 Counts Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CYC1								PH1							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYC0								PH0							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	CYC1	0	R/W	**PWM1_cycle (CYC1)** Defines the cycle of the PWM1. This is the sum of the high and low times. Value H'00 is not valid.
23 to 16	PH1	0	R/W	**PWM1_high (PH1)** Defines the period for which the PWM1 signal will remain high. Value H'00 is not valid.
15 to 8	CYC0	0	R/W	**PWM0_cycle(CYC0)** Defines the cycle of the PWM0. This is the sum of the high and low times. Value H'00 is not valid.
7 to 0	PH0	0	R/W	**PWM0_high(PH0)** Defines the period for which the PWM0 signal will remain high. Value H'00 is not valid.

Note: When the values of the bytes CYC1, PH1, CYC0 and PH0 are set, they should be based on the number of divided clocks, which is specified by the Clock Control bits in the PWM Control Register.

27.5.3 PWM23 Counts Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CYC3								PH3							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CYC2								PH2							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	CYC3	0	R/W	**PWM3_cycle (CYC3)** Defines the cycle of the PWM3. This is the sum of the high and low times. Value H'00 is not valid.
23 to 16	PH3	0	R/W	**PWM3_high (PH3)** Defines the period for which the PWM3 signal will remain high. Value H'00 is not valid.
15 to 8	CYC2	0	R/W	**PWM2_cycle (CYC2)** Defines the cycle of the PWM2. This is the sum of the high and low times. Value H'00 is not valid.
7 to 0	PH2	0	R/W	**PWM2_high (PH2)** Defines the period for which the PWM2 signal will remain high. Value H'00 is not valid.

Note: When the values of the bytes CYC3, PH3, CYC2, and PH2 are set, they should be based on the number of divided clocks, which is specified by the Clock Control bits in the PWM Control Register.

27.6 Functional Description

27.6.1 General Functionality

There are four independent PWM channels that have a programmable mark/space ratio programmed through a high time and a cycle time. These are each based on 8 bit counters. When enabled the output will remain high until the count reaches the PWMx_high value. At this point it will be set low. It will remain low until the count reaches the PWMx_cycle value. At this point the signal will return high and the counter will be reset to 0.

If the channel is not enabled the output will remain in the high state and the counter will be reset.

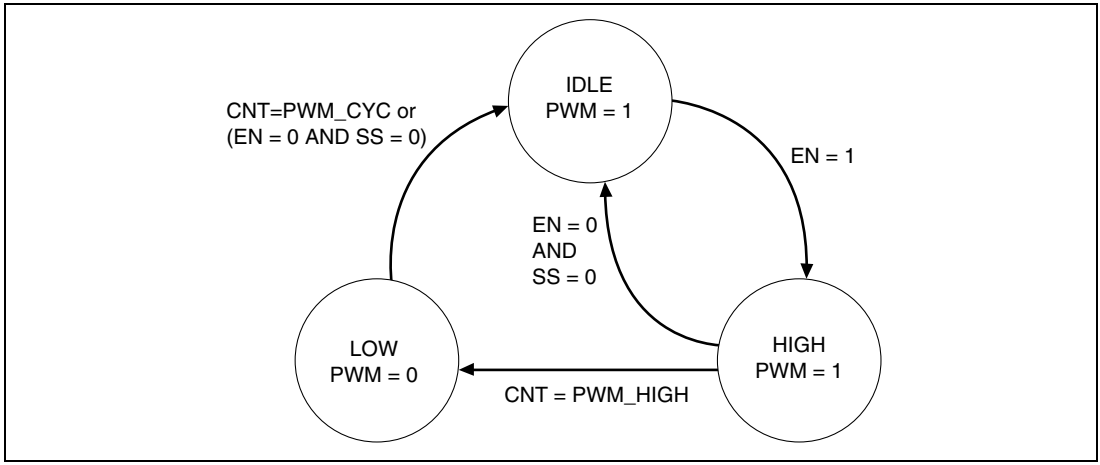


Figure 27.1 PWM State Transition

It should be noted that the number of clocks from the channel enabling write until the first falling edge of the channel pin will not accurately represent the configured values in pwm_control , pwm01_counts or pwm23_counts(maximum error is 100% when CYCx is set to H'01). However all subsequent edges are representative. This is illustrated in figure 27.2

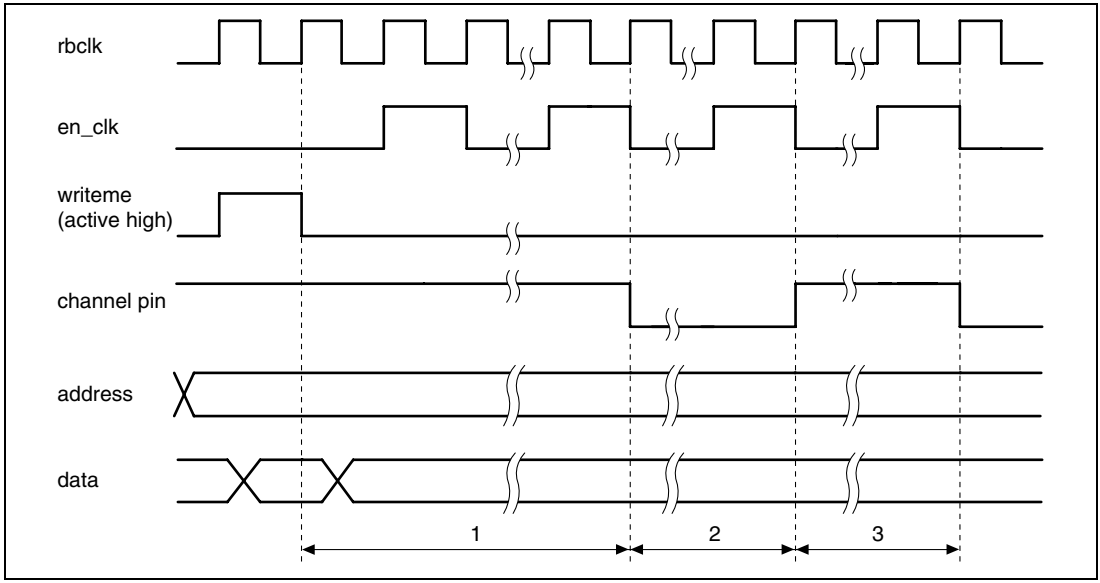


Figure 27.2 PWM Output Timing

1. Number of external clocks from time of enable until falling edge of channel pin will most likely be less than configured value.
2. Number of clocks between first falling edge of channel pin and rising edge will be as configured.
3. Number of clocks between rising and falling edge of channel pin will be as configured.

Each channel can work with a different input clock. This clock is derived from the register bus clock. The possible values are 1 to $1/2^{24}$ programmed through the PWM Control Register. This gives a minimum period of 30ns PCI bus (20 ns MPX bus) and a maximum period of approximately 2 minutes.

Each PWM channel can operate in either continuous or single shot mode. In continuous mode the PWM will continue to cycle as long as the enable is active. In single-shot mode the PWM channel will operate for one cycle. When in single shot mode, the enable will be cleared automatically after one count of CYC in either pwm01_counts or pwm23_counts.

In the case where either pwm01_counts or pwm23_counts are configured such that PWMx_high is greater than or equal to PWMx_cycle, the relevant channel pin will remain high. The channel pin will begin oscillating again when a channel enable is set with a new PWMx_high & PWMx_cycle configuration, where PWMx_high is less than PWMx_cycle.

27.6.2 Register Bus

The unit will be programmable through the register bus and all accesses should be 32 bits.

27.6.3 Standby Mode

This module allows clock gating to reduce power consumption. The module standby mode can be executed by controlling bit 21 in the Clock Control 1 (CC1) Register.

To wake up the module, bit 21 in the Clock Control 1 (CC1) Register must be enabled. After enabling this bit all access to the PWM module can be possible.

To power down the module, the following procedure is required.

1. Disable each PWM channel (EN in the PWM Control Register).
2. Disable bit 21 in the Clock Control 1 (CC1) Register.

Section 28 GPIO

28.1 General Description

This module is a 32-bit GPIO controller. This allows register control of the GPIO pins. The I/O direction of GPIO pins is controlled by Mode Register of Power Control & Configuration. If the direction is output then the value configured to another register is output on the pins. If the direction is input then the value on the pins is captured in another register. Individual GPIO pins can be made inactive to allow them to be used by an alternative function.

28.2 Features

- 32 independent GPIO
- Read from output register and pin
- Includes control for sharing the pin with alternative function

28.3 Interface

Table 28.1 GPIO Interface

Signal	Function	Direction
GP_OP(31:0)	General purpose output data	OUT
GP_IP(31:0)	General purpose input data	IN
GP_EN(31:0)	General purpose direction	OUT
SHARED_OUT(31:0)	Shared module data output	IN
SHARED_DIRN(31:0)	Shared module direction	IN
Register bus	System bus	—

28.4 Address Map

Table 28.2 Address Map

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6760	GPIO0 Direction		32
H'6764	GPIO0 Dataout		32
H'6768	GPIO0 Inactive		32
H'676C	GPIO0 Datain		32
H'6900	GPIO1 Direction		32
H'6904	GPIO1 Dataout		32
H'6908	GPIO1 Inactive		32
H'690C	GPIO1 Datain		32

28.5 Block Diagram

Figure 28.1 shows a block diagram of GPIO.

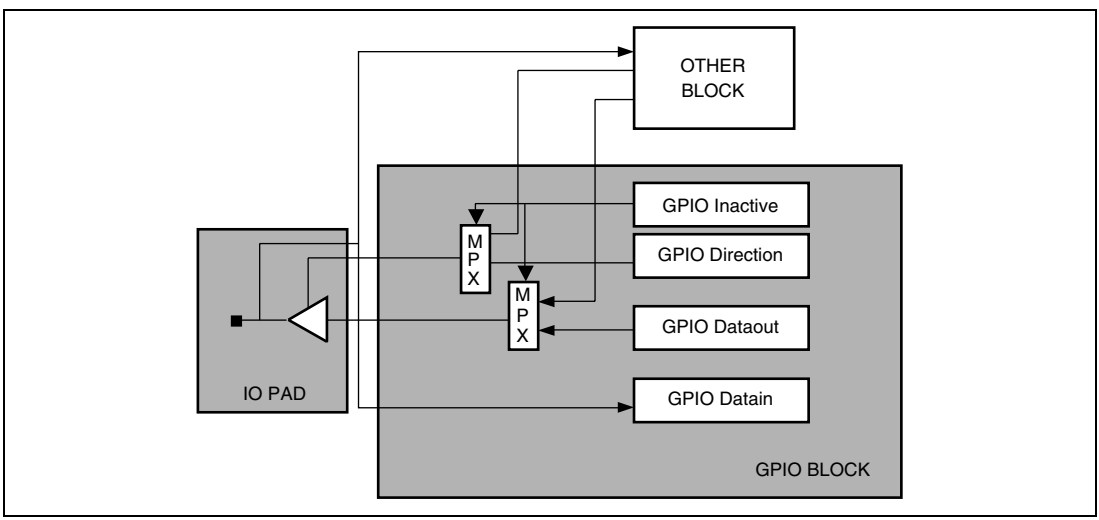


Figure 28.1 GPIO Block Diagram

28.6 Register Description

There are set of registers, which are located in the address space of the PCI or MPX bus and are located in the PCI memory window.

Each register has 1 bit corresponding to 1 channel of the GPIO.

Legends for register description:

Initial Value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

28.6.1 GPIO0 Inactive Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_inactive for GPIO(31) to GPIO(16)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_inactive for GPIO(15) to GPIO(0)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GPIO_inactive for GPIO(31) to GPIO(0)	0	R/W	**GPIO0 inactive** Indicates for each bit of the GPIO bus whether the GPIO controls the pin or the shared module. 0: GPIO controls the pin 1: Shared module controls the pin

28.6.2 GPIO1 Inactive Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_inactive for GPIO(63) to GPIO(56)												GPIO_inactive for GPIO(51) to GPIO(48)			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_inactive for GPIO(47) to GPIO(32)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GPIO_inactive for GPIO(63) to GPIO(56)	0	R/W	**GPIO1_inactive** Indicates for each bit of the GPIO bus whether the GPIO controls the pin or the shared module. 0: GPIO controls the pin 1: Shared module controls the pin
23 to 20	—	0	R	
19 to 0	GPIO_inactive for GPIO(51) to GPIO(32)	0	R/W	

Note: GPIO(52) to (55) are not defined.

28.6.3 GPIO0 Direction Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_ direction for GPIO(31) to GPIO(16)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_ direction for GPIO(15) to GPIO(0)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GPIO_ direction for GPIO(31) to GPIO(0)	0	R/W	**GPIO_direction** Indicates for each bit of the GPIO bus whether it is input or output. 0: Input 1: Output

28.6.4 GPIO1 Direction Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_direction for GPIO(63) to GPIO(56)												GPIO_direction for GPIO(51) to GPIO(48)			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_direction for GPIO(47) to GPIO(32)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GPIO_direction for GPIO(63) to GPIO(56)	0	R/W	**GPIO1_direction** Indicates for each bit of the GPIO bus whether it is input or output. 0: Input 1: Output
23 to 20	—	0	R	
19 to 0	GPIO_direction for GPIO(51) to GPIO(32)	0	R/W	

Note: GPIO (52) to (55) are not defined.

28.6.5 GPIO0 Dataout Register

Reset Value: H'00000000
Read/Write

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_ dataout for GPIO(31) to GPIO(16)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_ dataout for GPIO(15) to GPIO(0)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GPIO_dataout for GPIO(31) to GPIO(0)	0	R/W	**GPIO0_dataout** In output mode the pin reflects the value written to this register if the corresponding GPIO_inactive bit is set to 0.

28.6.6 GPIO1 Dataout Register

Reset Value: H'00000000
Read/Write

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_dataout for GPIO(63) to GPIO(56)												GPIO_dataout for GPIO(51) to GPIO(48)			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_dataout for GPIO(47) to GPIO(32)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GPIO_dataout for GPIO(63) to GPIO(56)	0	R/W	**GPIO1_dataout** In output mode the pin reflects the value written to this register if the corresponding GPIO_inactive bit is set to 0.
23 to 20	—	0	R	
19 to 0	GPIO_dataout for GPIO(51) to GPIO(32)	0	R/W	

Note: GPIO (52)-(55) are not defined.

28.6.7 GPIO0 Datain Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_datain for GPIO(31) to GPIO(16)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_datain for GPIO(15) to GPIO(0)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GPIO_datain for GPIO(31) to GPIO(0)	0	R	**GPIO0_datain** This register reflects the current value of the pin, regardless of the direction.

28.6.8 GPIO1 Datain Register

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GPIO_datain for GPIO(63) to GPIO(56)												GPIO_datain for GPIO(51) to GPIO(48)			
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GPIO_ datain for GPIO(47) to GPIO(32)															
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GPIO_datain for GPIO(63) to GPIO(56)	0	R	**GPIO1_datain** This register reflects the current value of the pin, regardless of the direction.
23 to 20	—	0	R	
19 to 0	GPIO_ datain for GPIO(51) to GPIO(32)	0	R	

Note: GPIO(52) to (55) are not defined.

28.7 Functional Description

28.7.1 General Functionality

The GPIO has two 32-bit ports, an input (gp_ip), an output (gp_op), and a direction 32-bit port (gp_en) to be used to enable or disable tristate buffers in external inout multiplex circuitry (which merges in and out ports into one single inout port). The direction can be specified on a bit level by GPIO Direction Register. Whatever this register's value, datain register always samples the signal fed to gp_ip.

GPIO may, in turn, be disabled so that an external source of signal takes over the control of the output pins. This behaviour can be controlled on a bit level. When the GPIO_inactive bit is 0 then the respective pin is used for GPIO. When the bit is set to 1 then the additional module may drive the pin. In all cases it is possible to read the value of the input pins through the GPIO Datain Register. GPIO Shared_dirn signal controls the direction of the bits whose control has been given up by the GPIO. Those Shared Dirn Register's bits which are equal to '1' (high level) are set up as input, and those which are equal to '0' (low level) to output (note it's an opposite conversion to that used in Data Direction Register).

The table below shows the possible configurations when an external tristate circuit is attached. When the output is defined as High-Z it is available as input.

Table 28.3 Configuration of each port

GPIO Inactive	GPIO Direction	GPIO Dataout	Shared Direction	Shared Output	Output (tri state)
0	0	X	X	X	High Z
0	1	1	X	X	1
0	1	0	X	X	0
1	X	X	Input	X	High Z
1	X	X	Output	1	1
1	X	X	Output	0	0

28.7.2 Register Bus

The unit is programmable through the register bus and all accesses are 32 bits.

28.7.3 Standby Mode

This module allows clock gating to reduce power consumption. The module standby mode can be executed by controlling bit 14 and 25 in the Clock Control 1 (CC1) Register.

To wake up the module, bit 14 and 25 in the Clock Control 1 (CC1) Register must be enabled. After enabling this bit all access to the GPIO module can be possible.

To power down the module bit 14 and 25 in the Clock Control 1 (CC1) Register (GPIO0 and GPIO1 respectively) need to be disabled.

28.8 References

For register base address information refer to DMAC block specification.

Section 29 Expansion Bus

29.1 General Description

The Expansion Bus Module controls the transfer of data to and from external peripherals or SRAM via the expansion port.

29.2 Features

- Standard 32-bit Register Bus DMA / processor interface to the host system
- Each peripheral can be accessed either directly by a processor using wait states, or by DMA
- Up to two external devices supported via two chip selects
- Up to 128 byte-wide locations (7-bit address) available for each chip selects
- Option to use just one chip select with a multiplexed 8-bit data/15-bit address bus

29.3 Architectural Overview

29.3.1 Block Diagram

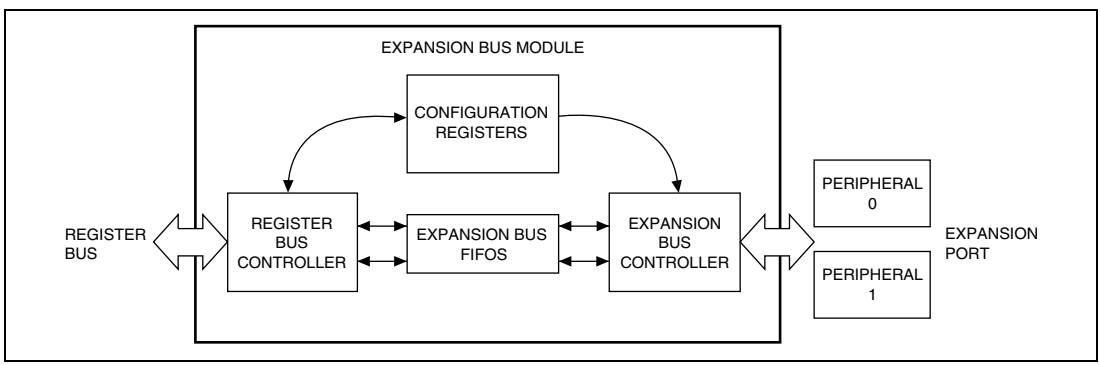


Figure 29.1 Block Diagram of Expansion Bus Module

29.3.2 Register Bus Interfacing

The expansion port module uses the standard register bus interface, and offers the option of either being controlled directly by the processor, or by the DMA Controller.

As the data in the expansion port is only 8 bits wide, the standard 32-bit register bus interface can be used, but bits 31 to 8 are ignored, and information is expected in bits 7 to 0.

29.4 Abbreviations

The following abbreviations are used in this document:

ALE: Address Latch Enable. An operating mode for the expansion bus module where just one peripheral is supported, and some of the address bits are multiplexed onto the data bus. Also referred to as "multiplexed mode" in this specification.

29.5 Pin Descriptions

Table 29.1 Expansion Bus Module Port Connections

Pin Name	Bits	I/O	Group	Function
EX_ADDR	7	O	Expansion Port	Address for both peripherals
EX_DATA	8	IO	Expansion Port	Data for both peripherals
$\overline{\text{EX\_RD}}$	1	O	Expansion Port	Active low read strobe for both peripherals
$\overline{\text{EX\_WR}}$	1	O	Expansion Port	Active low write strobe for both peripherals
$\overline{\text{EX\_CS0}}$	1	O	Expansion Port	Active low chip select for peripheral 0
$\overline{\text{EX\_CS1}}$	1	O	Expansion Port	Dual purpose: Active low 2nd chip select, or active high ALE signal

29.6 Register Descriptions

29.6.1 Memory Map

The memory map for the expansion bus module is shown below in Figure 29.2, and the memory maps for the expansion peripherals themselves is described in Figure 29.3, for the case where two peripherals are used, each with a 7-bit address range.

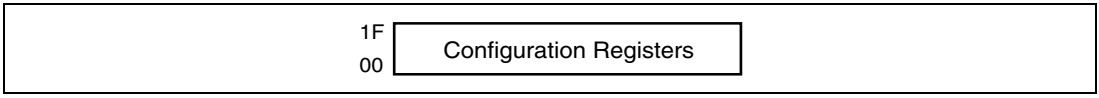


Figure 29.2 Byte Memory Map for the Expansion Bus Module

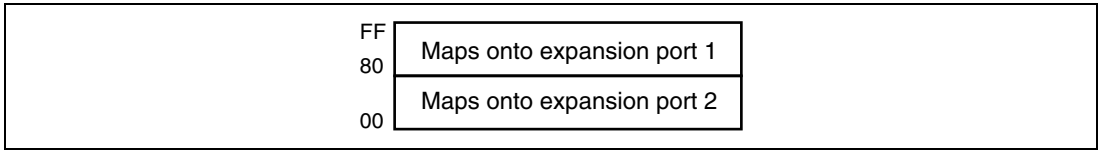


Figure 29.3 Byte Memory Map for two expansion ports

For the case where just one peripheral is used, with a multiplexed 15-bit address/8-bit data bus, the memory map is show in Figure 29.4 below. The value "xx" is a 7-bit page identifier, specified in the ex Page Number Register.

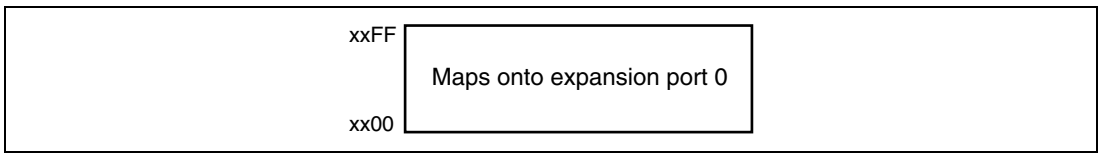


Figure 29.4 Byte Memory Map for single expansion port

29.6.2 Full Register List

The full list of registers available within the expansion bus module is summarised in Table 29.2 below, and each register is described in more detail in the subsequent pages.

Registers whose name ends in "0" are used to control peripheral 0, and those ending in "1" are used to control peripheral 1.

All expansion bus module registers must be accessed as 32-bit longwords. Any fields marked H'00 or similar are reserved. When writing to such fields, the bits must be set to 0. When reading, the values are not guaranteed.

Table 29.2 Expansion Bus Module Register Summary

Address (Bytes)	Register Name	Abbreviation	Access Size
H'6200	ex WaitStates0		32
H'6204	ex WaitStates1		32
H'6208	ex DMA Config0		32
H'620C	ex DMA Config1		32
H'6210	ex Config0		32
H'6214	ex Config1		32
H'6218	ex Page Number		32
H'621C	ex Mode Config		32

Legends for register description:

Initial value	: Register value after reset
—	: Undefined value
R/W	: Read and Write, write value can be read.
R	: Read only, for write always 0 write
R/WC0	: Read and Write, 0 write clear, 1 write is ignored
R/WC1	: Read and Write, 1 write clear, 0 write is ignored
W	: Write only, Read prohibited. If reserved, write always 0.
—/W	: Write only, Read value undefined.

ex WaitStates0, ex WaitStates1 Registers

These two registers, one for each peripheral, define the number of wait cycles to be inserted for read and write accesses from either peripheral, using the appropriate chip select.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										WTW				WTR		
Initial:	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6 to 4	WTW	1	R/W	**WTW** Number of wait cycles during write accesses, from H'0 to H'7
3	—	0	R	**Reserved**
2 to 0	WTR	1	R/W	**WTR** Number of wait cycles during read accesses, from H'0 to H'7

ex Config0, ex Config1 Registers

These two registers, one for each peripheral, configure how each of the expansion ports operate. They control the frequency of the internal reference clock, shown in the timing diagrams.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												CKD				
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R	**Reserved**
4 to 1	CKD	0	R/W	**CKD**
0	CKD	1	R/W	Clock Divider for deriving reference clock frequency by dividing the register bus clock frequency by CKD. Valid values are in the range of H'01 to H'1F. The following values offer the following characteristics: H'00 Reserved H'01 Each access phase lasts for 1 rbclk period H'02 Each access phase lasts for 2 rbclk periods : H'1E Each access phase lasts for 30 rbclk periods H'1F Each access phase lasts for 31 rbclk periods

ex DMA Config0, ex DMA Config1 Registers

These two registers, one per peripheral, initiate DMA operations for each of the two expansion ports. The DMA controller is programmed with information about the transfer, such as address ranges, but the expansion bus module must be programmed with the length of the transfer, so that it can be initiated.

Non-DMA PIO accesses to an expansion peripheral must not be used during DMA transfers to the same peripheral. Before PIO accesses can resume, it must be confirmed that the previous DMA transfer has completely finished by confirming that the LEN field of ex DMA Config0 or ex DMA Config1 Register, as appropriate, reports that H'00 bytes of the DMA transfer remain uncompleted. It is not sufficient to wait for the DMAC to report that the transfer has finished, because the DMAC is only able to report when the last byte has been transferred to the expansion bus module – it is not able to report when the expansion bus module has finished transferring data to the external peripheral.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
														KIL	INC	WR
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LEN								ADD							
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	0	R	**Reserved**
18	KIL	0	R/W	**KIL** Kills any outstanding DMA activity by writing '1' with other fields set to H'00000. Automatically clears itself back to '0'. Note that the DMA controller must be programmed beforehand to stop DMA transfers to the expansion bus module.
17	INC	0	R/W	**INC** Increment address. If set, the address increments after each access
16	WR	0	R/W	**WR** Write. If set, the DMA writes to the port, otherwise it reads.

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	LEN	0	R/W	**LEN** Writing initiates a DMA transfer of length LEN in bytes, from H'01 to H'80. Reading returns the number of bytes remaining.
7 to 0	ADD	0	R/W	**ADD** The address for DMA transfers.

ex Mode Config Register

This register determines the operating mode of the expansion bus module. It allows the user to enable or disable each chip select, or to choose the operating mode for a single peripheral where just one chip select, EX_CS0, is used with a multiplexed 15-bit address and 8-bit data bus and an ALE strobe.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												END	EME	EPM	EP1	EP0
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	0	R	**Reserved**
4	END	0	R/W	**END** If set, accesses to the expansion peripherals' memory space are assumed to be in little-endian format. If clear, accesses to the expansion peripherals' memory space are assumed to be in big-endian format.
3	EME	0	R/W	**EME** If set, enable Expansion Bus Module. If clear, disable Expansion Bus Module.

Bit	Bit Name	Initial Value	R/W	Description
2	EPM	0	R/W	**EPM** If set, enable ALE/multiplexed mode accesses using EX_CS0, and the EP1 and EP0 bits in this register are ignored. If clear, disable ALE/multiplexed Mode accesses using EX_CS0.
1	EP1	0	R/W	**EP1** If set, enable Peripheral 1 accesses via EX_CS1. If clear, disable Peripheral 1 accesses via EX_CS1.
0	EP0	0	R/W	**EP0** If set, enable Peripheral 0 accesses via EX_CS0. If clear, disable Peripheral 0 accesses via EX_CS0.

ex Page Number Register

For the operation mode where just one chip select is used with a multiplexed 15-bit address and 8-bit data bus, writes to this register determine the 7-bit page number in use. For example, if a value of H'07 is written to this register, then accesses to the expansion peripheral address the range from 0x0700 through to H'07FF. If a value of H'01 is written, then accesses to the expansion peripheral address the range from H'0100 through to H'01FF, and so on.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
										PAGE						
Initial:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 7	—	0	R	**Reserved**
6 to 0	PAGE	0	R/W	**PAGE** The Page Number being accessed, which is used to prefix all addresses.

29.7 Functional overview

The expansion bus module interfaces between the processor and up to two peripherals connected to the expansion port of the companion chip. It allows read and write access to each peripheral, and can be accessed either by means of DMA transfers, or directly by the processor.

The access cycles can be programmed independently, not just for each peripheral, but also for read and write cycles. This offers the best opportunity for a generic access cycle to interface with optimum efficiency to a wide range of peripherals.

To enhance efficiency, the module does not hold off the processor bus using the rbwaitn signal during write cycles unless it is absolutely necessary. This is useful for a peripheral which only has one byte written at a time, and allows data to be written at the peripheral's lower speed without stalling the whole companion chip register bus. However, if writes are performed in quick succession, then the expansion bus module automatically asserts wait during the second and subsequent writes, while the backlog is cleared.

During multiplexed (ALE) operation, when the EPM bit in ex Mode Config Register is set, address bits 7 to 0 are multiplexed with EX_DATA[7:0], and address bits 14 to 8 are sent out via EX_ADDR. Multiplexed operation can only be selected for peripheral 0, and in addition to the normal EX_CS0 chip select, an ALE signal is provided via the EX_CS1 output port. All parameters, such as wait states, are configured via the Port 0 configuration registers.

If incremental DMA accesses are used with multiplexed operation, then the page being addressed, in ex Page Number Register, automatically increments if enough DMA accesses occur to make the address roll over from H'FF to H'00. For example, after a DMA transfer to H'1FF, the next transfer is to H'200 if the INC bit is set in ex DMA Config01 Registers, or to H'1FF if the INC bit is clear. While a DMA transfer is taking place to a channel, that channel must not be accessed directly via PIO (direct register bus) accesses – each channel can only be used in either PIO or DMA mode at any given time, not both modes.

The arbitration method used by the expansion bus module is such that any pending write accesses will have priority over read accesses. If both peripherals are programmed for writes, or both peripherals are programmed for reads, then access to the expansion port will alternate between the two peripherals. This algorithm guarantees fairness for direct register bus access, ensuring that requests are serviced in the order that they are received.

A knock-on effect of this arbitration scheme, brought about by the non-blocking nature of DMA data transfers, is that if DMA mode is selected for both peripherals, but one peripheral is set up for write and the other is set up for read, then the write DMA accesses will tend to be favoured over the read DMA accesses, meaning that the time to complete the DMA read activity will be greater than that for the DMA write activity. In extreme cases, the DMA read transfer may not occur until the DMA write transfer has completed. Given that DMA transfers are limited to a maximum of 128 accesses, this delay should be imperceptible.

29.8 Expansion Bus Module Standby Mode

The expansion bus module allows clock gating to reduce power consumption.

To power down the module, the following procedure is required:-

1. If DMA is used, program the DMAC to cancel all DMA activity via the expansion bus
2. Ensure that all previous expansion bus accesses have finished by allowing a minimum of 20us delay after the last expansion bus read or write request
3. Write to ex DMA Config0 Register, setting the KIL bit
4. Write to ex DMA Config1 Register, setting the KIL bit
5. Write to ex Mode Config Register, clearing all bits
6. Write to the Clock Control 1 Register in the Power Control module, clearing the EXP bit

To wake up the module, the following procedure is required:-

1. Write to the Clock Control 1 Register in the Power Control module, setting the EXP bit
2. Configure the expansion bus module as required. The ex Page Number , ex Config0/1 and ex WaitStates0/1 Registers will have retained their pre-powerdown settings.

29.9 References

Hitachi Register Bus DMA Controller Specification

Section 30 JTAG

30.1 Overview

30.1.1 Features

The HD64404 JTAG is a serial input/output interface supporting JTAG, IEEE 1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture. The HD64404 JTAG uses five pins ($\overline{\text{TRSTN}}$, TCK, TMS, TDI, and TDO). The pin functions and serial transfer protocol support the JTAG specifications.

30.1.2 Block Diagram

Figure 30.1 shows a block diagram of the HD64404 JTAG. The TAP (test access port) controller and control registers are reset independently of the chip reset pin by driving the pin. The other circuits are reset and initialized in an ordinary reset. The JTAG circuit has two internal registers: SDBPR and SDIR. The SDBPR register supports the JTAG bypass mode, SDIR is the command register. SDIR can be accessed directly from the TDI.

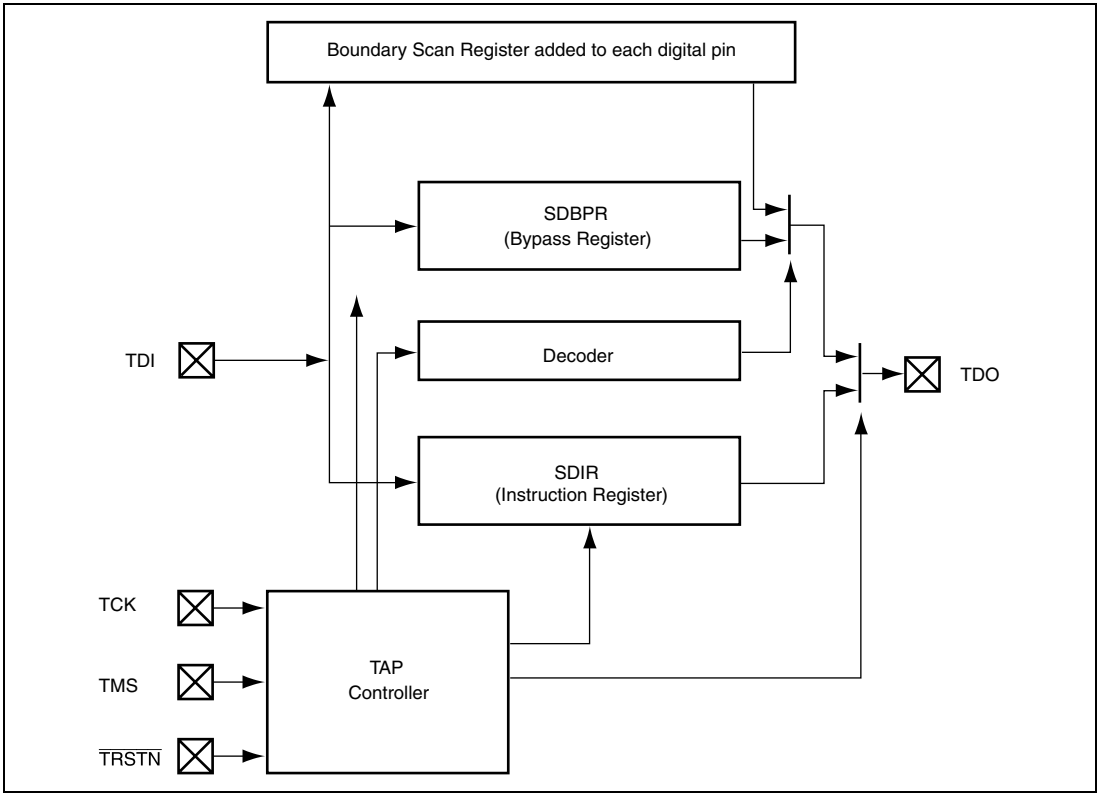


Figure 30.1 Block diagram of HD64404 JTAG Circuit

30.1.3 Pin Configuration

Table 30.1 shows the HD64404 JTAG pin configuration.

Table 30.1 HD64404 JTAG Pins

Pin Name	Abbreviation	I/O	Function	When Not Used
Clock pin	TCK	Input	Same as the JTAG serial clock input pin. Data is transferred from data input pin TDI to the JTAG circuit, and data is read from data output pin TDO, in synchronization with this signal.	Fixed 0 or 1
Mode pin	TMS	Input	The mode select input pin. Changing this signal in synchronization with TCK rising edge determines the meaning of the data input from TDI. The protocol supports the JTAG (IEEE Std 1149.1) specification.	Open (*[1])
Reset Pin	$\overline{\text{TRSTN}}$	Input	The input pin that resets JTAG circuit. This signal is received asynchronously with respect to TCK, and effects a reset of the JTAG interface circuit when low. TRSTN must be driven low for a certain period when powering on, regardless of whether or not JTAG is used.	(*[2])
Data input pin	TDI	Input	The data input pin. Data is sent to the JTAG circuit by changing this signal in synchronization with TCK	Open (*1)
Data Output pin	TDO	Output	The data output pin. Data is sent to the JTAG circuit by reading this signal in synchronization with TCK.	Open

Notes: 1. Pulled up inside the chip.
2. There are following considerations for TRSTN.
It must be asserted when power is on.
System reset and TRSTN must be separated on the board.

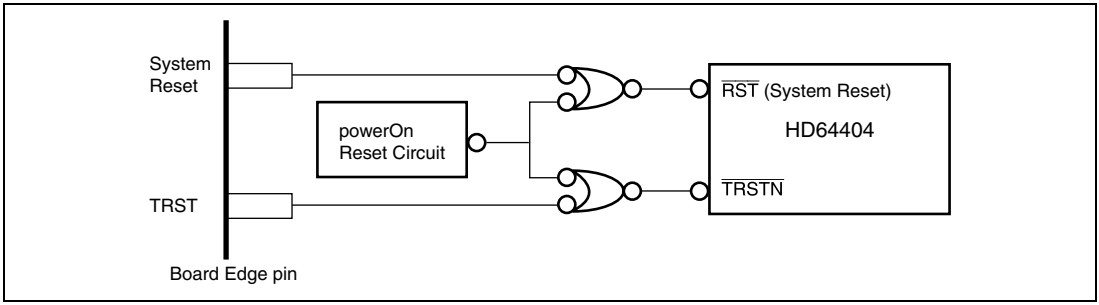


Figure 30.2 An Example of Reset signal Design on the Board

30.2 JTAG instruction

HD64404 supports three JTAG instructions.

No	Instruction	Opcode	Usage
1	BYPASS	111	HD64404 is bypassed through the boundary scan chain.
2	EXTEST	000	Connectivity check between LSIs for PCB.
3	SAMPLE/ PRELOAD	001	Sampling output data on LSI pins while LSI is in normal mode. Preloading the initial values on LSI pins for EXTEST.

30.3 Operation

30.3.1 TAP Control

Figure 30.2 shows the internal states of the TAP control circuit. These conform to the state transitions specified by JTAG.

The transition condition is the TMS value at the rising edge of TCK.

The TDI value is sampled at the rising edge of TCK, and shifted at the falling edge.

The TDO value changes at the falling edge of TCK. When not in the Shift-DR or Shift-IR state, TDO is in the high-impedance state.

In a transition to TRSTN = 0, a transition is made to the Test-Logic-Reset state asynchronously with respect to TCK.

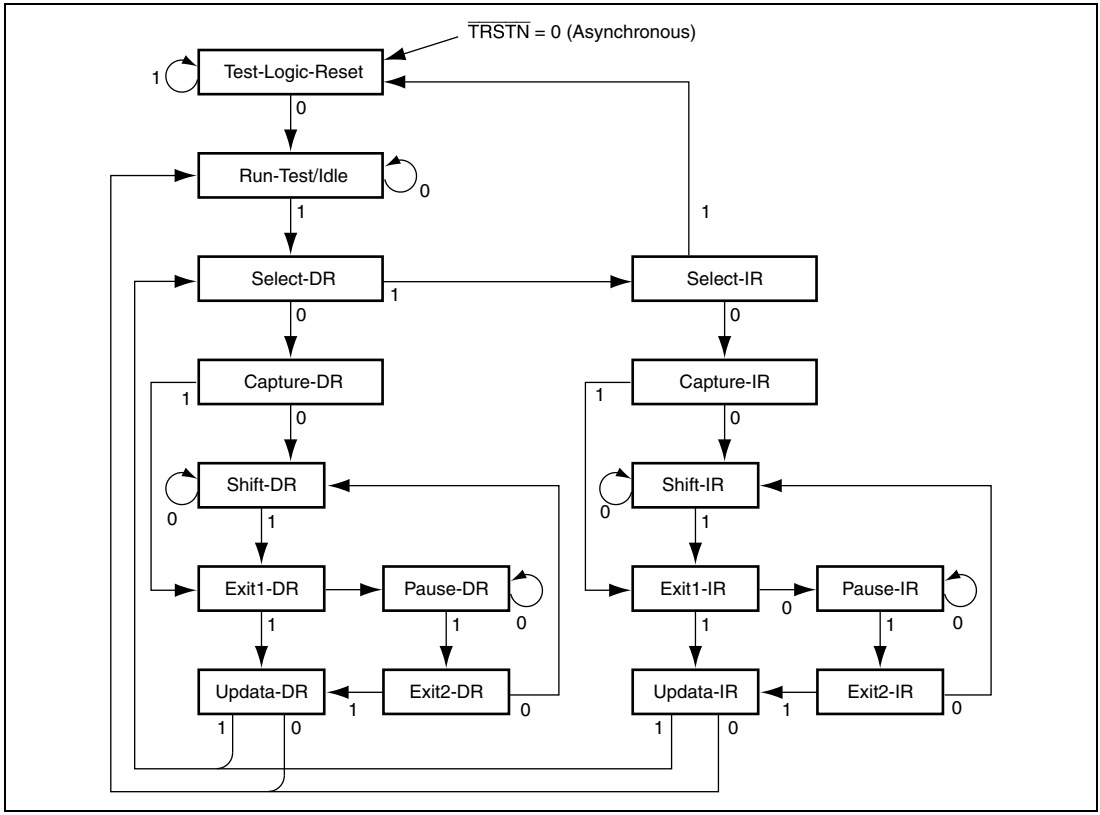


Figure 30.3 TAP Control State Transition Diagram

30.3.2 Boundary Scan Register

Boundary Scan Register is a shift register located near each LSI pin pad to control the direction of each I/O pin. Using EXTEST and SAMPLE/PRELOAD command, the boundary scan test can be executed. The relation between HD64404 Pin and boundary scan register is shown in the BSDL(Boundary Scan Description Language) file which Hitachi will provide.

Section 31 Electrical Specification

31.1 Absolute Maximum Ratings

Table 31.1 Absolute Maximum Ratings

Item	Spec.
Applied Voltage of VCC for I/O	–0.3 V to + 4.6 V
Applied Voltage of VCC for core	–0.3 V to + 2.1 V
Input Voltage	–0.3V to VCC for I/O + 0.3 V
Output Voltage	–0.3V to VCC for I/O + 0.3 V
Output Current for output pin	14 mA for Iol = 4 mA
	28 mA for Iol = 8 mA
Output Current for a pair of VCC and GND	42 mA
Storage Temperature	–55°C to + 125°C

31.2 VDD Voltage

Table 31.2 VDD Voltage

Item	Symbol		Unit
Core Voltage	Vdd	1.5 +/– 0.1	V
I/O Voltage	Vcc	3.3 +/– 0.3	V

31.2.1 Power On/Power Off Procedure

Power Consumption: T.B.D

Power On
GND → 3.3 V Power on → 1.5 V Power on → Signal Input

Power Off
Signal input off → 1.5 V Power off → 3.3 V Power off → GND
Or
Signal input off → 3.3 V power off → 1.5 V power off → GND

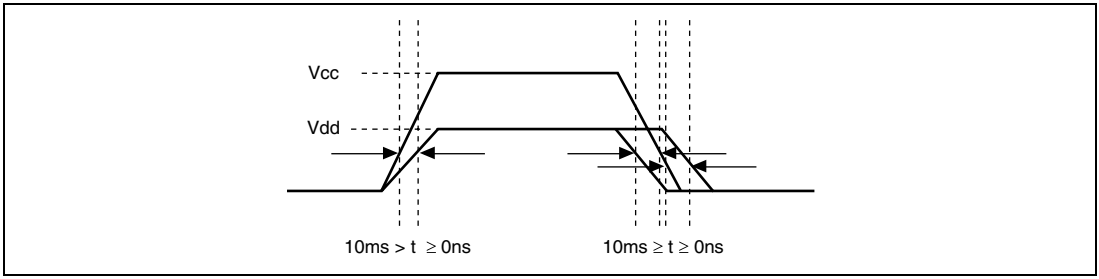


Figure 31.1 Power up/down sequence

31.3 All Digital I/O (76C Technology)

Table 31.3 All Digital I/O (76C Technology)

Parameter	Description	Min (V)	Max (V)	V_{cc} (V)	V_{dd} (V)	Note
V_{IL}	Low level input voltage	–0.3	0.8	3.0 to 3.6	1.4 to 1.6	Guaranteed Input Low voltage
V_{IH}	High level input voltage	2.0	V_{CC}+0.3	3.0 to 3.6	1.4 to 1.6	Guaranteed Input High voltage
V_{OL}	Low level output voltage	–0.3	0.4	3.0 to 3.6	1.4 to 1.6	I_{OL}, 2 mA (TTL)
V_{OH}	High level output voltage	2.4	V_{CC}+0.3	3.0 to 3.6	1.4 to 1.6	I_{OH}, 2 mA (TTL)
V_{T+}	High level input voltage for Schmitt	—	2.2	3.3	1.5	AT DMARQ0, AT DCHRDY0, AT DIRQ1
V_{T-}	Low level input voltage for Schmitt	0.8	—	3.3	1.5	AT DMARQ0, AT DCHRDY0, AT DIRQ1
CL	Pin Capacitance	—	10 pF	—	—	
Tr/Tf	Transition time	—	100 ns	3.0 to 3.6	1.4 to 1.6	TIMER/CTR[3:0]
	for Input pins	—	60 ns	3.0 to 3.6	1.4 to 1.6	SSIn_SCK
		—	10 ns	3.0 to 3.6	1.4 to 1.6	All other pins

31.4 USB I/O

Table 31.4 USB I/O

Parameter	Description	Min (V)	Max (V)	V_{cc} (V)	Note
V_{IL}	Low level input voltage	—	0.8	3.0 to 3.6	Guaranteed Input Low voltage
V_{IH}	High level input voltage	2.0	—	3.0 to 3.6	Guaranteed Input High voltage
V_{OL}	Low level output voltage	0.0	0.3	3.0 to 3.6	RL of 1.425 Kohm to VCC
V_{OH}	High level output voltage	2.8	VCC	3.0 to 3.6	RL of 14.25 Kohm to GND

31.5 Clock Reset Specification

Table 31.5 Clock Reset Specification

Pin	Item	Symbol	Min	Max	Unit	Figure
$\overline{RST}$	Power On PLL oscilating time	t_{osc1}	11	—	ms	31.2
PLL_ENABLEN	Clock Stop wait time	$t_{standby\_wait}$	11	—	ms	31.3
	Clock Recovery wait time	t_{osc2}	11	—	ms	31.4
AUDIO CLK	Audio clock oscilating time	t_{osc3}	200	—	ms	31.5
USB 1HP/HM USB 2HP/HM	USB clock oscilating time	t_{osc4}	200	—	ms	31.6

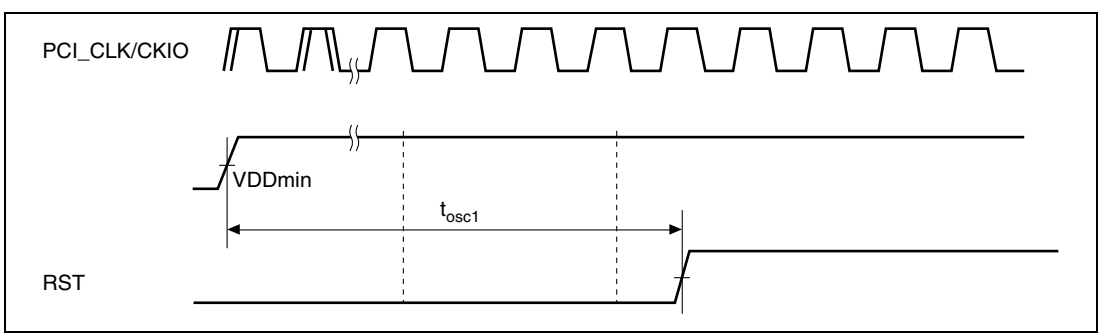


Figure 31.2 Power On Reset

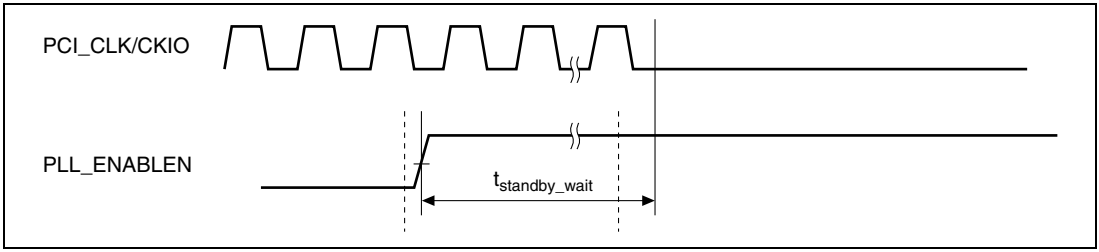


Figure 31.3 Standby

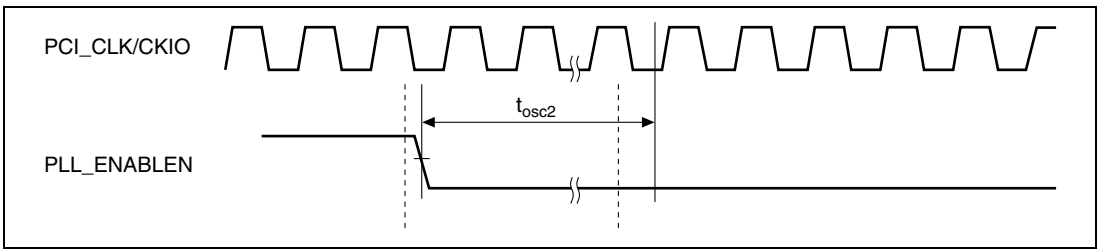


Figure 31.4 Clock Recovery from Standby

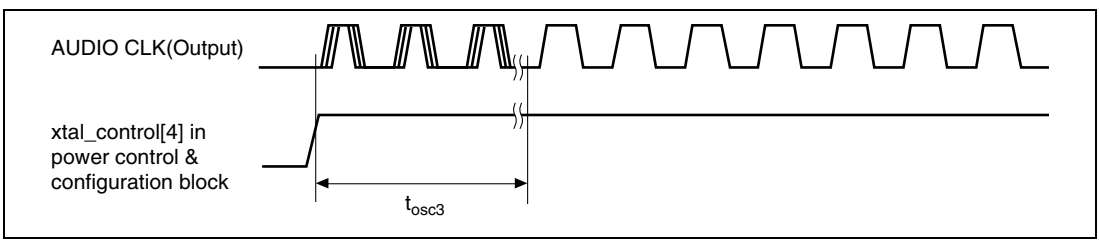


Figure 31.5 Audio crystal oscilating time

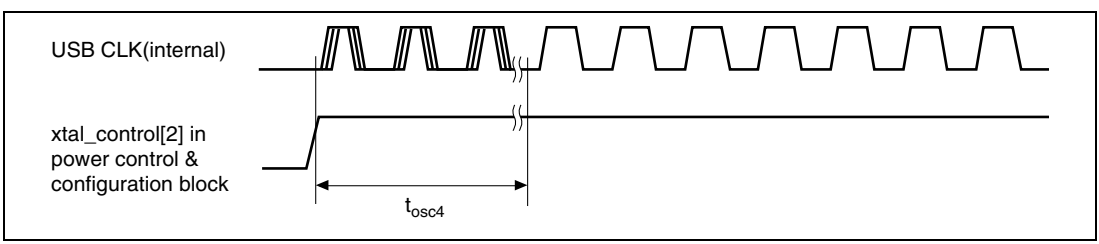


Figure 31.6 USB crystal oscilating time

31.6 PCI Signal Timing Specification

Table 31.6 PCI Signal Timing Specification

$V_{CC\ for I/O}$ = 3.0 to 3.6 V, C_L = 50 pF

			33 MHz			
Pin	**Item**	**Symbol**	**Min**	**Max**	**Unit**	**Figure**
PCICLK	Clock cycle	t_{PCICYC}	30	—	ns	31.7
	Clock pulse width (high)	$t_{PCIHIGH}$	11	—	ns	31.7
	Clock pulse width (low)	t_{PCILOW}	11	—	ns	31.7
	Clock rise time	t_{PCIr}	—	4	ns	31.7
	Clock fall time	t_{PCIf}	—	4	ns	31.7
PCIRST	Output data delay time	t_{PCIVAL}	—	9	ns	31.8
IDSEL	Input hold time	t_{PCIH}	3*	—	ns	31.9
	Input setup time	t_{PCISU}	6	—	ns	31.9
AD31 to AD0, $\overline{CBE3}$ to $\overline{CBE0}$, PAR, $\overline{FRAME}$, $\overline{IRDY}$, $\overline{TRDY}$, $\overline{STOP}$, $\overline{DEVSEL}$, $\overline{PERR}$	Output data delay time	t_{PCIVAL}	—	9	ns	31.8
	Tri-state drive delay time	t_{PCION}	—	9	ns	31.8
	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	ns	31.8
	Input hold time	t_{PCIH}	3*	—	ns	31.9
	Input setup time	t_{PCISU}	6	—	ns	31.9
$\overline{GNT}$, $\overline{REQ}$	Output data delay time	t_{PCIVAL}	—	9	ns	31.8
	Tri-state drive delay time	t_{PCION}	—	9	ns	31.8
	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	ns	31.8
	Input hold time	t_{PCIH}	3*	—	ns	31.9
	Input setup time	t_{PCISU}	6	—	ns	31.9
$\overline{SERR}$, $\overline{INTA}$	Tri-state drive delay time	t_{PCION}	—	9	ns	31.8
	Tri-state high-impedance delay time	t_{PCIOFF}	—	12	ns	31.8

Note: * Although those pins require more than 1ns hold time. HD6417751R′s PCI interface can be connected with 3ns hold time.

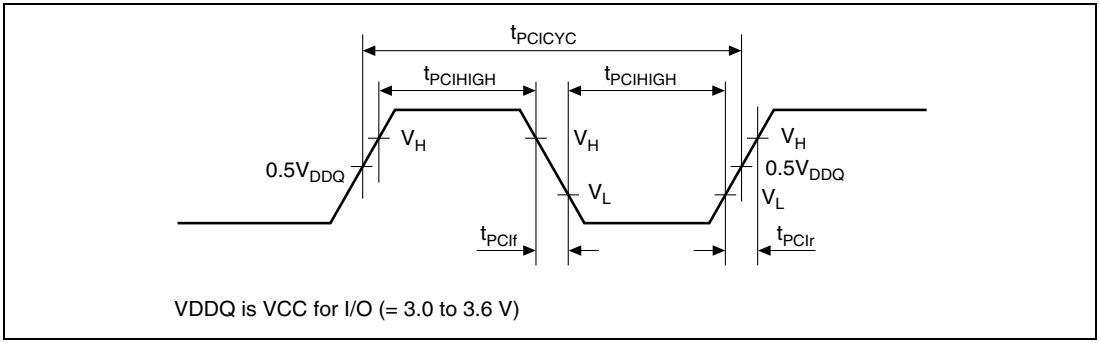


Figure 31.7 PCI Clock Input Timing

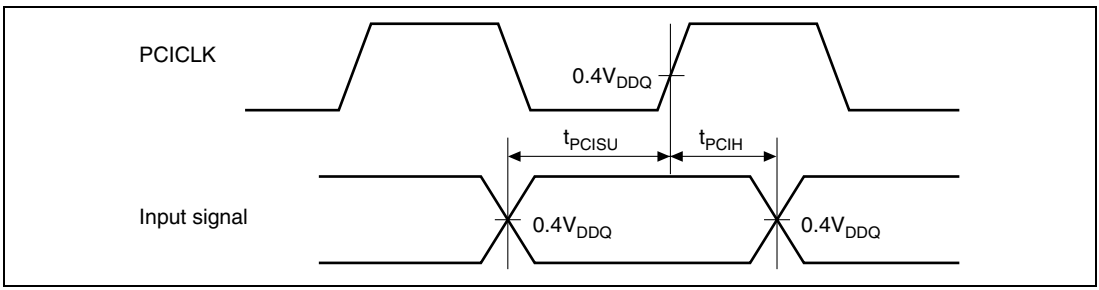


Figure 31.8 Input Signal Timing

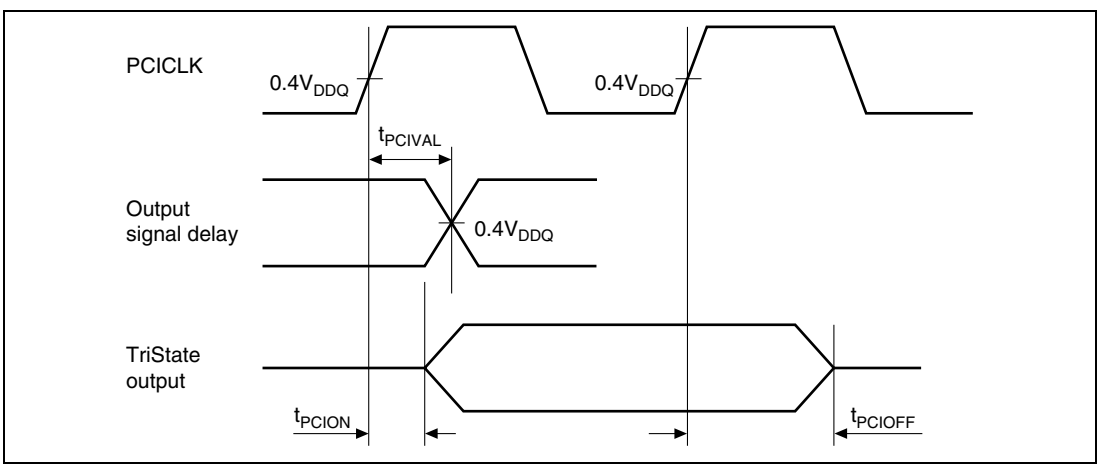


Figure 31.9 Output Signal Timing

31.7 MPX I/F

Table 31.7 MPX I/F

$V_{CC\ for I/O}$ = 3.0 to 3.6 V, C_L = 30 pF

		100 MHz		83 MHz			
Item	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Unit**	**Notes**
$\overline{FRAME}$, $\overline{BS}$, SH4_$\overline{CSA}$, SH4_$\overline{CSB}$, RD/$\overline{WR}$, D[63:61], DACK, DRAK setup time	t_{CS}	3.5	—	3.5	—	ns	
$\overline{FRAME}$, $\overline{BS}$, SH4_CSA, SH4_CSB, RD/$\overline{WR}$, D[63:61], DACK, DRAK hold time	t_{CH}	1.5	—	1.5	—	ns	
D[31:0] setup time	t_{DS}	3.5	—	3.5	—	ns	
D[31:0] hold time	t_{DH}	1.5	—	1.5	—	ns	
D[31:0] read data delay	t_{RDD}	—	6	—	7	ns	
$\overline{RDY}$, $\overline{DREQ}$ delay	t_{RDYD}	—	6	—	7	ns	
$\overline{IRL}$ delay	t_{IRL}	—	8	—	8	ns	

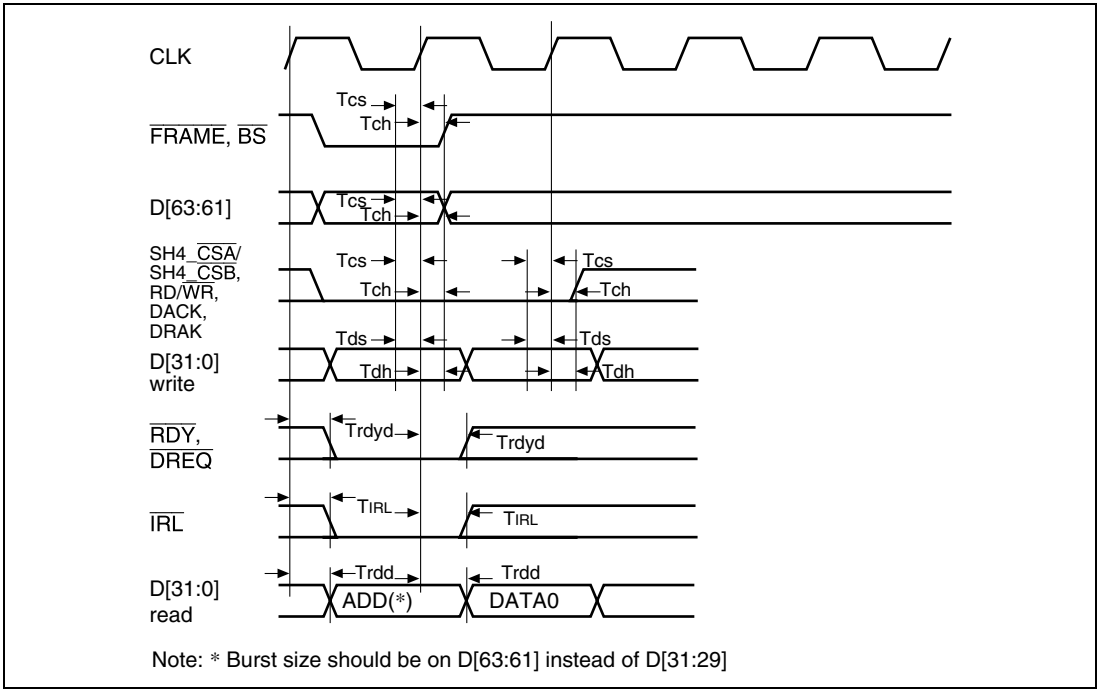


Figure 31.10 MPX Interface Timing

31.8 SDRAM I/F

Table 31.8 SDRAM I/F Timing

Item	Symbol	Min	Typ	Max	Unit	Figure
SD_CLK cycle period	t_{CK}	10.0	—	—	ns	31.11
SD_CLK High level pulse width	t_{CKH}	2.5	—	—	ns	31.12
SD_CLK Low level pulse width	t_{CKL}	2.5	—	—	ns	31.13
SD_CKE delay	t_{CE}	1.1*	—	6	ns	
$\overline{CS}$, $\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, $\overline{DQM[3:0]}$ delay	t_{CD}	1.1*	—	6	ns	
SD_AD[12:0], $\overline{BA0}$, $\overline{BA1}$ delay	t_{AD}	1.1*	—	6	ns	
SD_DATA[31:0] setup time	t_{DS}	3	—	—	ns	
SD_DATA[31:0] hold time	t_{DH}	1.5	—	—	ns	
SD_DATA[31:0] delay	t_{DT}	1.1*	—	6	ns	

Note: * In board design, SD_CLK must not be delayed more than those pins.

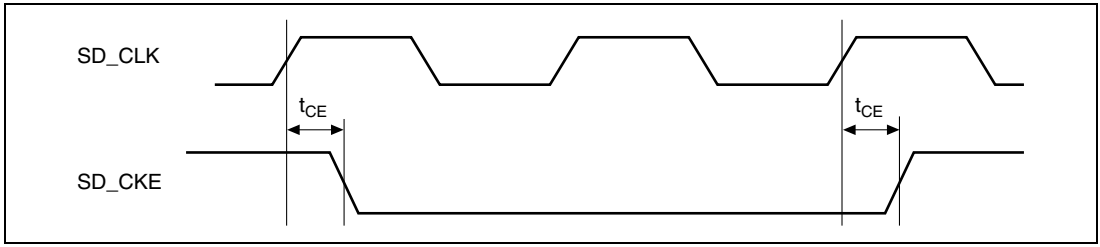


Figure 31.11 SDRAM Clock

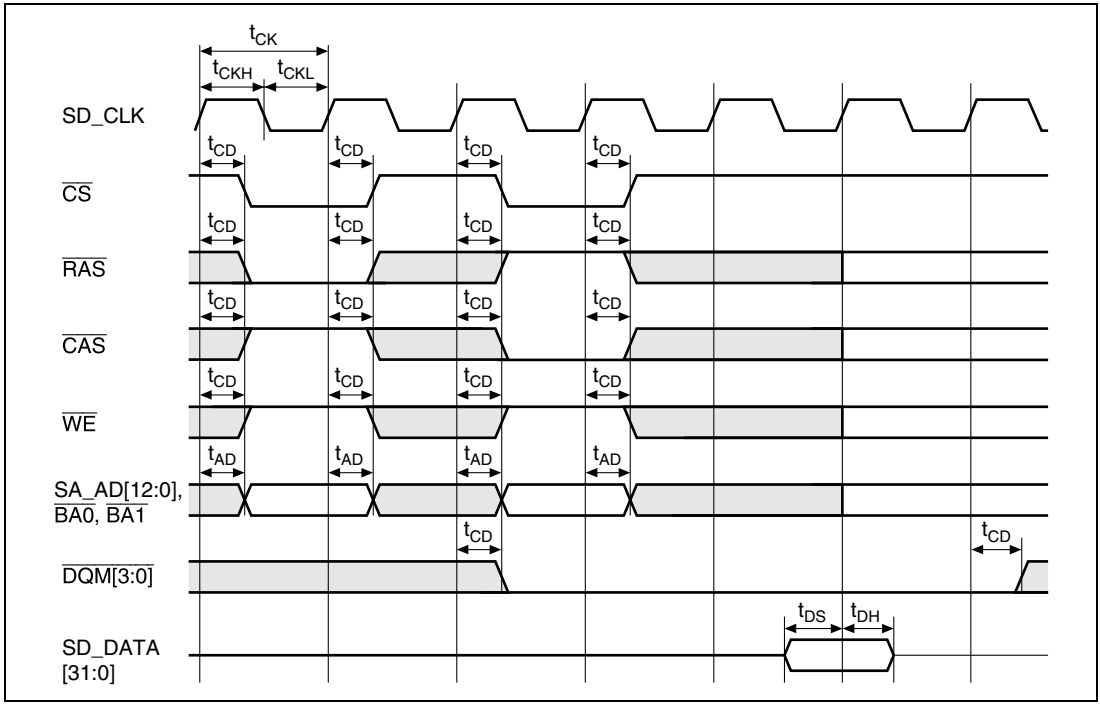


Figure 31.12 SDRAM Read Cycle

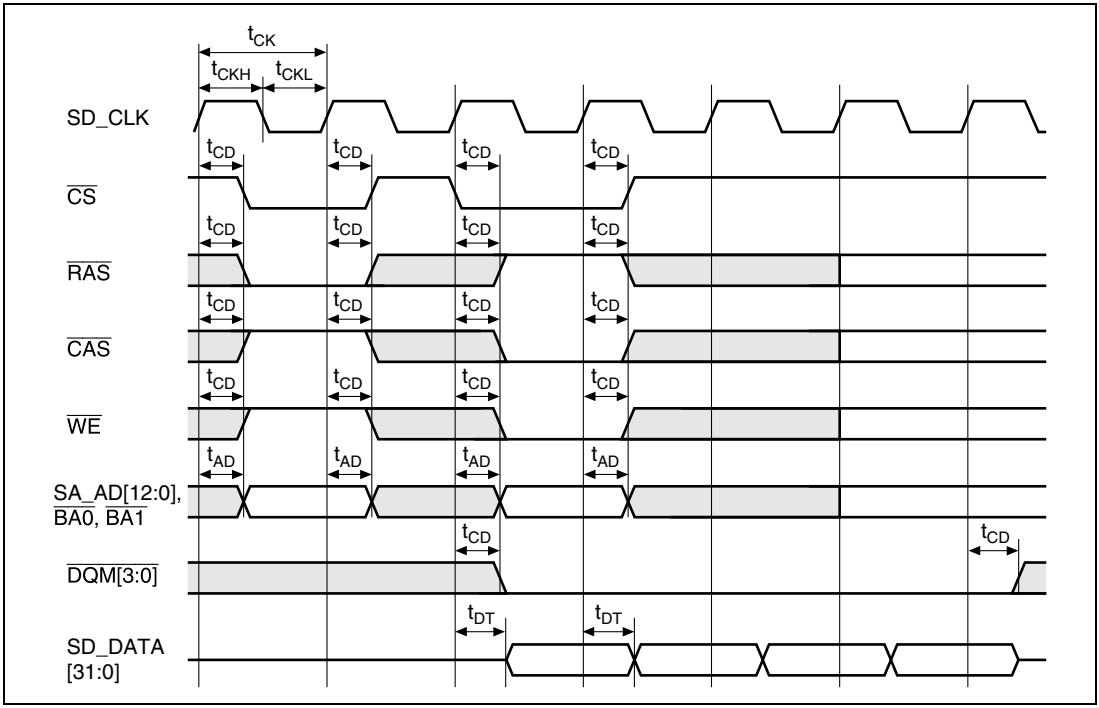


Figure 31.13 SDRAM Write Cycle

31.9 Display Out Interface

Table 31.9 Tab DisplayOut interface

Item	Symbol	Min	Typ	Max	Unit
DOT_CLK output period	TP	20	—	—	ns
DOT_CLK output high level period	TPW	4	—	—	ns
DO_VSYNC, DO_HSYNC input setup	TC	5	—	—	ns
DO_VSYNC, DO_HSYNC input hold time	TCW	3	—	—	ns
DOT_CLK → DO_DATA[17:0], DO_VSYNC, DO_HSYNC, DO_DEN output delay	TDD	—	—	13	ns

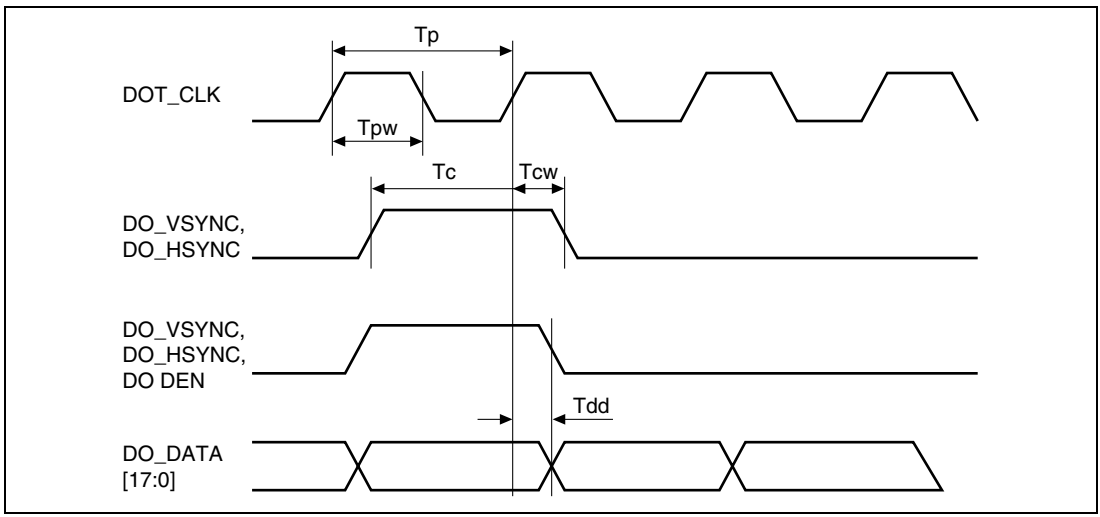


Figure 31.14 Display Out Interface

31.10 Video In

Table 31.10 VideoIn timing

Item	Symbol	Min	Typ	Max	Unit
VI_DATA input hold time	Tvidh	2	—	—	ns
VI_DATA input setup time	Tvids	10	—	—	ns
VI_CLK clock period	Tvicyc		37		ns
VI_CLK clock jitter		–3		3	ns
Duty factor	Dvicyc	40	50	60	%
VI_CLK rising time	Tvicr	—	—	8	ns
VI_CLK falling time	Tvicf	—	—	8	ns

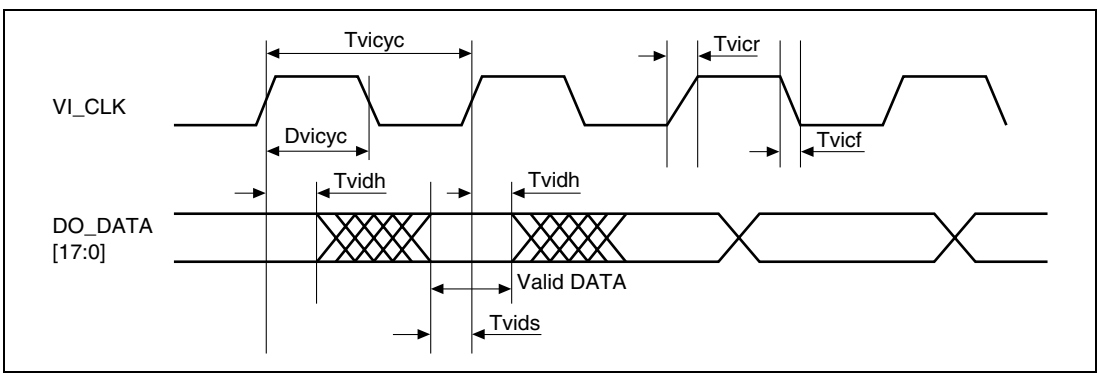


Figure 31.15 Video In Timing

31.11 GPIO, PWM, Interrupt INPUT, SPDIF, Timer, CAN Timing

Table 31.11 GPIO, INTERRUPT INPUT, SPDIF, Timer Timing

Item	Symbol	Min	Typ	Max	Unit	Figure
GPIO output delay	TIOPD	—	—	20	ns	
GPIO input setup time	TIOPS	20	—	—	ns	
GPIO input hold time	TIOPH	20	—	—	ns	
PWM output delay	TIOPD	—	—	20	ns	
INT[7:0] setup time	TINTS	20	—	—	ns	
INT[7:0] hold time	TINTH	20	—	—	ns	
TIMER/CTR output delay time	Ttmd	—	—	36	ns	
TIMER/CTR input setup time	Ttms	20	—	—	ns	
TIMER/CTR input hold time	Ttmh	20	—	—	ns	
TIMER clock level low width	Ttmlow	1.5	—	—	Tcyc	
TIMER clock level high width	Ttmhigh	1.5	—	—	Tcyc	
CAN_TX output delay time	Tcand	—	—	100	ns	
CAN_RX input setup time	Tcans	100	—	—	ns	
CAN_RX input hold time	Tcanh	100	—	—	ns	
AUDIO_OUT frequency	Faudio			512Fs		Fs = 33 kHz, 44.1 kHz, or 48 kHz
SPDIF_OUT delay time	Tspd	—	—	30	ns	
SPDIF_IN setup time	Tsps	10	—	—	ns	
SPDIF_IN hold time	Tsph	10	—	—	ns	

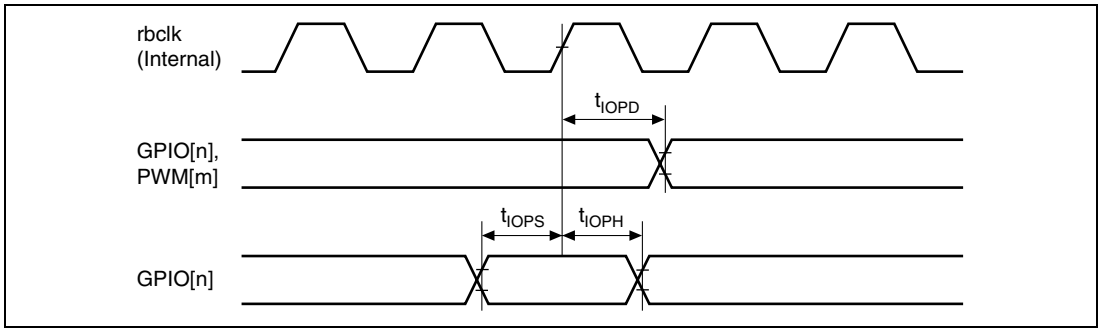


Figure 31.16 GPIO, PWM Timing

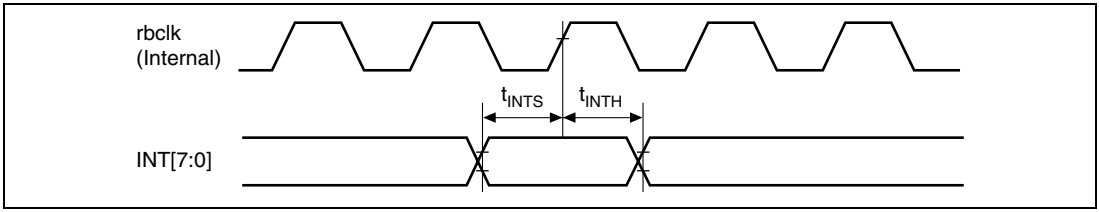


Figure 31.17 INTERRUPT INPUT Timing

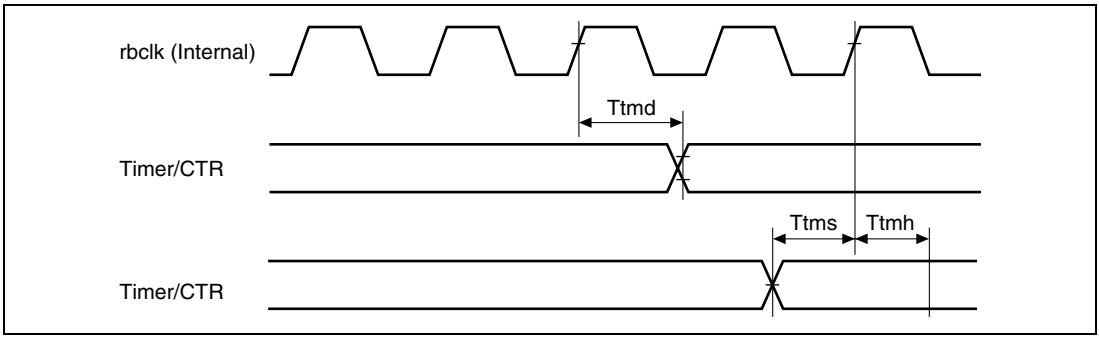


Figure 31.18 TIMER Timing (1)

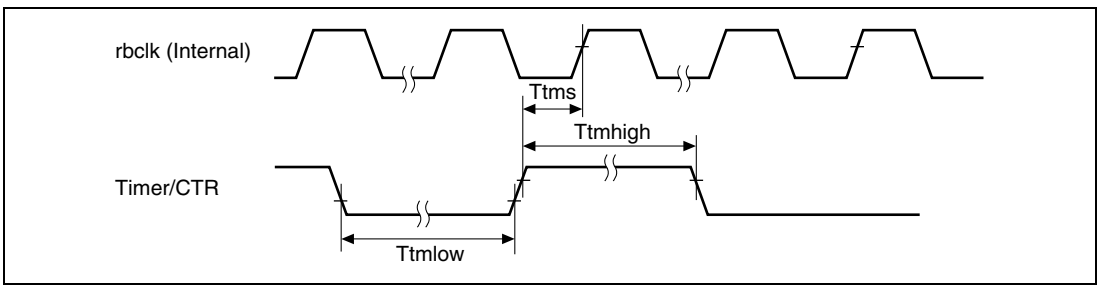


Figure 31.19 TIMER Timing (2)

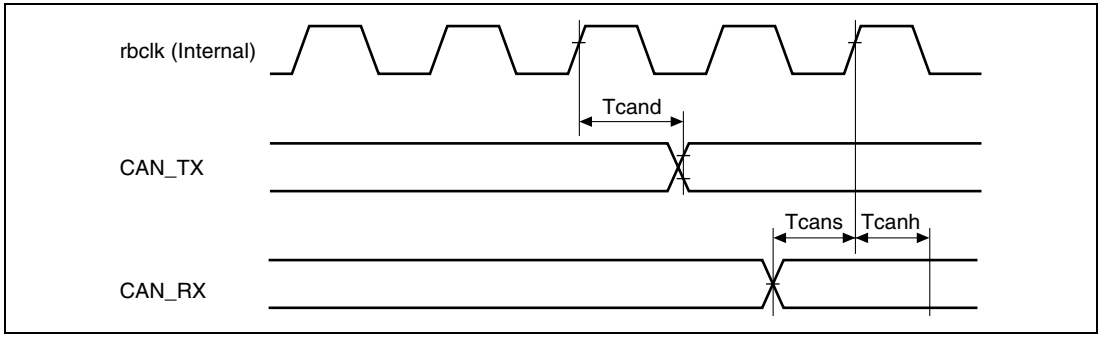


Figure 31.20 CAN Timing

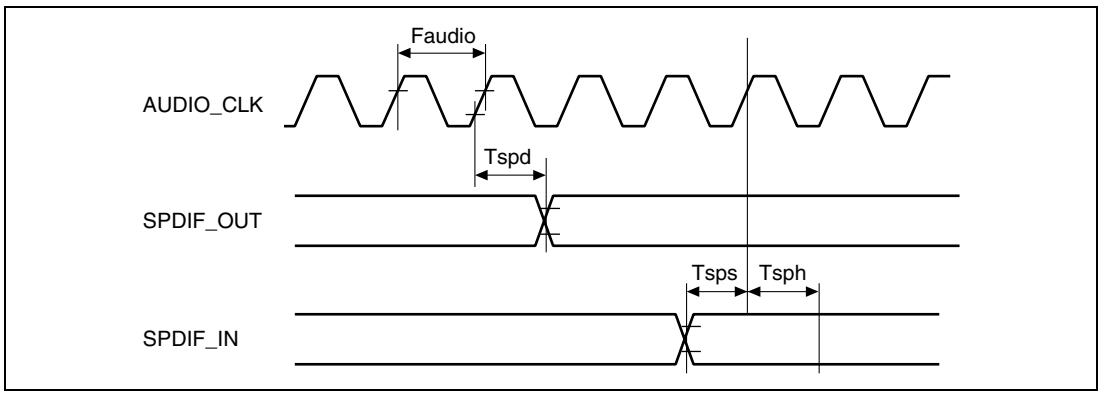


Figure 31.21 SPDIF Timing

31.12 I²C Interface

Table 31.12 I²C Timing

Item	Symbol	Min	Typ	Max	Unit
I2Cn_SCL frequency	Ticyc	84	—	—	Tcyc
I2Cn_SCL Low level pulse width	Ticwl	42	—	—	Tcyc
I2Cn_SCL High level pulse width	Ticwh	18	—	—	Tcyc
I2Cn_SCL/I2Cn_SDA rising time	Ticr	—	—	300	ns
I2Cn_SCL/I2Cn_SDA falling time	Ticf	—	—	300	ns
I2Cn_SDA Bus free time	Ticbf	18	—	—	Tcyc
I2Cn_SCL Start condition hold time	Tich	6	—	—	Tcyc
I2Cn_SCL Retransmission start condition set up time	Tics	18	—	—	Tcyc
I2Cn_SDA Stop Condition setup time	Ticst	18	—	—	Tcyc
I2Cn_SDA Set up time	Tdas	3	—	—	Tcyc
I2Cn_SDA Hold time	Ticdh	0	—	—	ns

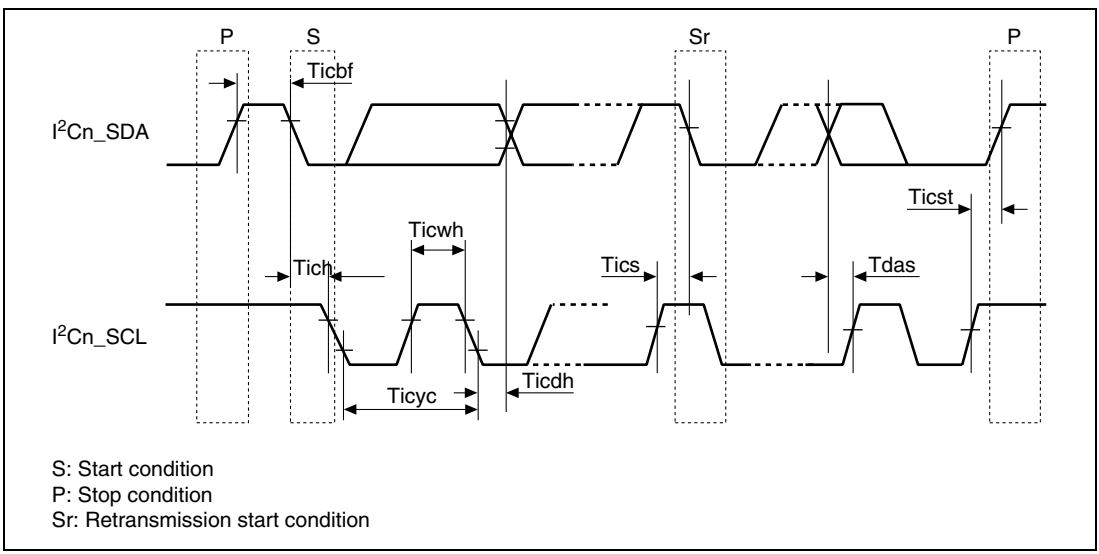


Figure 31.22 I²C Timing

31.13 ATAPI Interface

Table 31.13 ATAPI interface

Item	Symbol	Min	Typ	Max	Unit	Figure
pix_clk to AT_$\overline{\text{DCS}}$[1:0], AT_DSA[2:0] valid	t_1	—	—	8	ns	31.23
pix_clk to $\overline{\text{AT\_DIOR}}$/$\overline{\text{AT\_DIOW}}$ assert	t_2	—	—	8	ns	
pix_clk to AT_DSD valid (write)	t_3	—	—	8	ns	
$\overline{\text{AT\_DIOW}}$ data setup time(PIO)	t_4	20	—	—	ns	
$\overline{\text{AT\_DIOW}}$ data hold time(PIO)	t_5	10	—	—	ns	
$\overline{\text{AT\_DIOR}}$ data setup time(PIO)	t_6	30	—	—	ns	
$\overline{\text{AT\_DIOR}}$ data hold time(PIO)	t_7	5	—	—	ns	
$\overline{\text{AT\_DIOR}}$/$\overline{\text{AT\_DIOW}}$ to $\overline{\text{AT\_DCS}}$[1:0], AT_DSA[2:0] hold time	t_8	10	—	—	ns	
pix_clk to $\overline{\text{AT\_DMACK0}}$ assert	t_9			3	ns	31.24
$\overline{\text{AT\_DMACK0}}$ to $\overline{\text{AT\_DIOR}}$/$\overline{\text{AT\_DIOW}}$ setup time	t_{10}	0	—	—	ns	
$\overline{\text{AT\_DIOR}}$/$\overline{\text{AT\_DIOW}}$ to $\overline{\text{AT\_DMACK0}}$ hold time	t_{11}	0	—	—	ns	
$\overline{\text{AT\_DIOR}}$ to AT_DMARQ0 delay	t_{12r}	—	—	120	ns	
$\overline{\text{AT\_DIOW}}$ to AT_DMARQ0 delay	t_{12w}	—	—	40	ns	
$\overline{\text{AT\_DIOR}}$ data setup time	t_{13}	20	—	—	ns	
$\overline{\text{AT\_DIOR}}$ data hold time	t_{14}	5	—	—	ns	
$\overline{\text{AT\_DIOW}}$ data setup time	t_{15}	20	—	—	ns	
$\overline{\text{AT\_DIOW}}$ data hold time	t_{16}	10	—	—	ns	
$\overline{\text{AT\_DMACK0}}$ setup/hold time	t_{ack}	20	—	—	ns	31.25, 31.27 to 31.29, 31.32, 31.33
Envelope time	t_{env}	20	—	70	ns	31.25, 31.29
Data setup time at recipient	t_{ds}	7	—	—	ns	31.25, 31.26
Data valid hold time at recipient	t_{dh}	5	—	—	ns	
Strobe edge to edge time	t_{cyc}	54	—	—	ns	31.26
Strobe cycle time	t_{2cyc}	115	—	—	ns	
Limited interlock time	t_{li}	0	—	150	ns	31.29, 31.32, 31.33

Item	Symbol	Min	Typ	Max	Unit	Figure
Interlock time with minimum	t_{mli}	20	—	—	ns	31.27, 31.28, 31.32
Data valid setup time	t_{dvs}	30	—	—	ns	31.27 to 31.30, 31.32, 31.33
Data valid hold time	t_{dvh}	6	—	—	ns	
Unlimited interlock time	t_{ui}	0	—	—	ns	31.29
Ready to final strobe time	t_{rfs}	—	—	75	ns	31.31
Strobe to STOP delay time	t_{ss}	50	—	—	ns	31.32

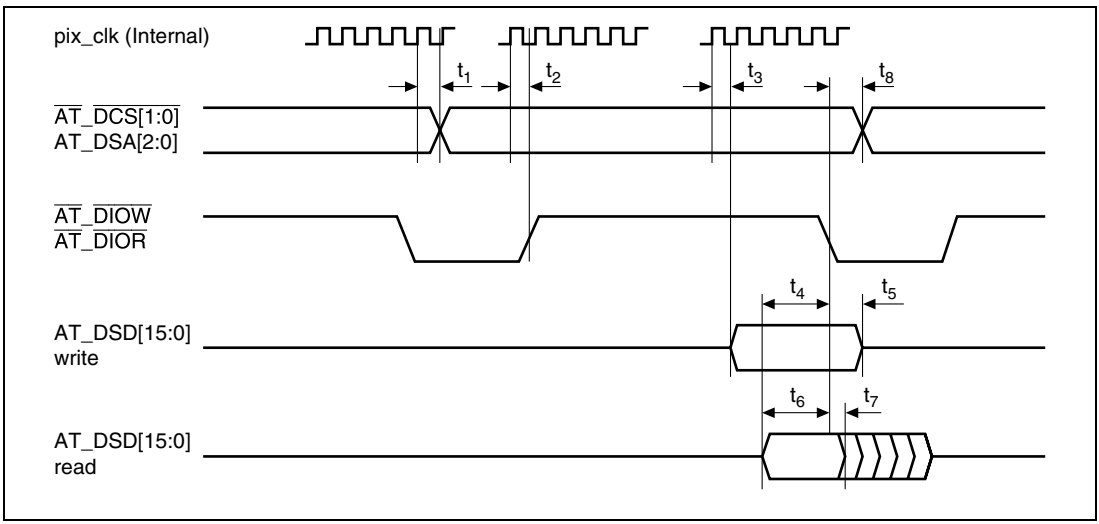


Figure 31.23 PIO Data Transfer to/from Device Register Transfer to/from Device

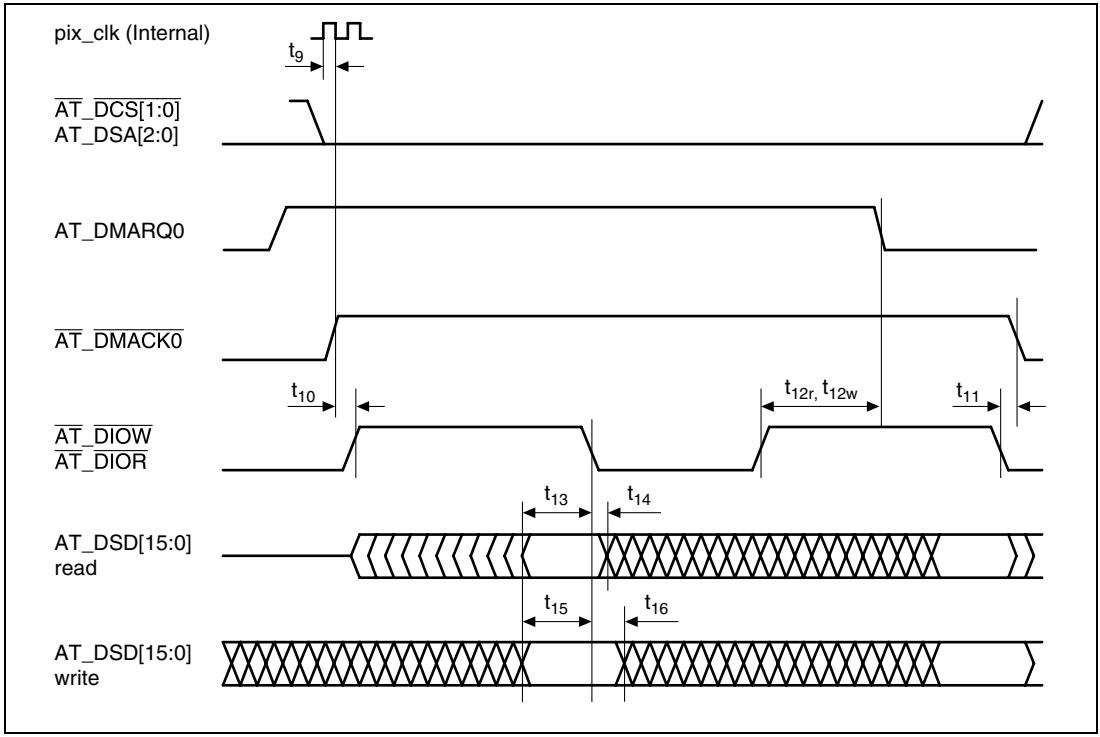


Figure 31.24 Multi Word DMA Data Transfer

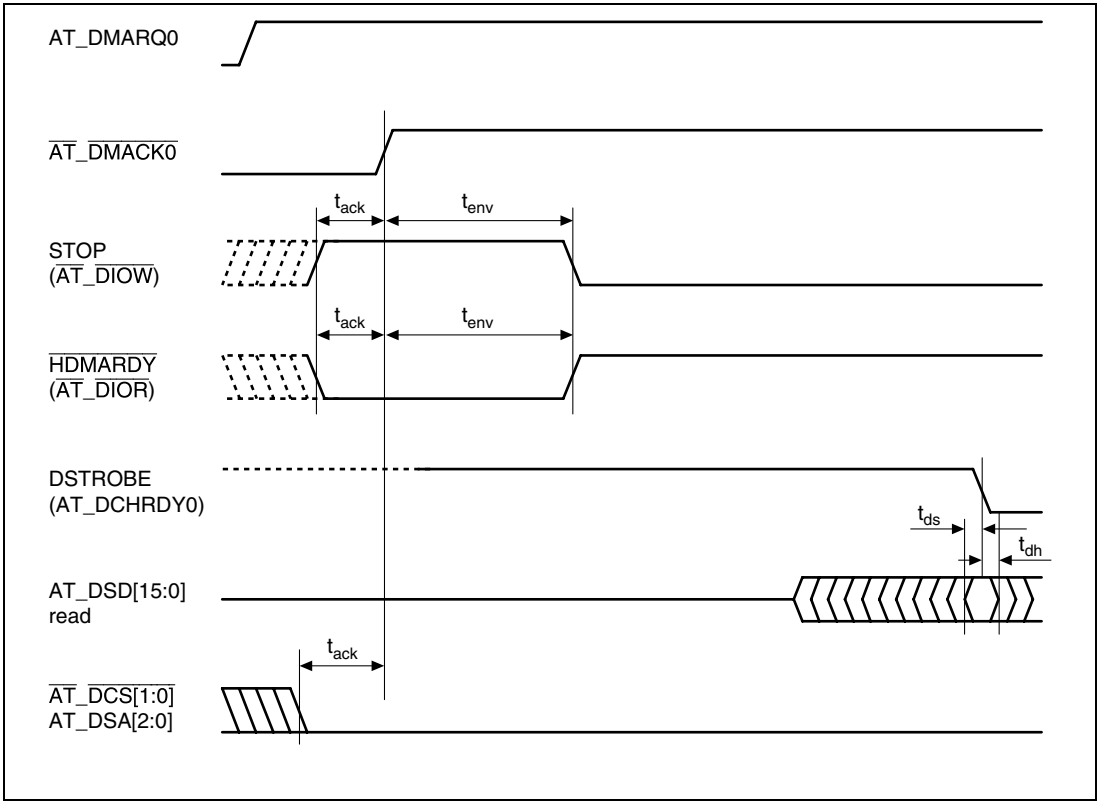


Figure 31.25 Initiating an Ultra DMA Data-in Burst

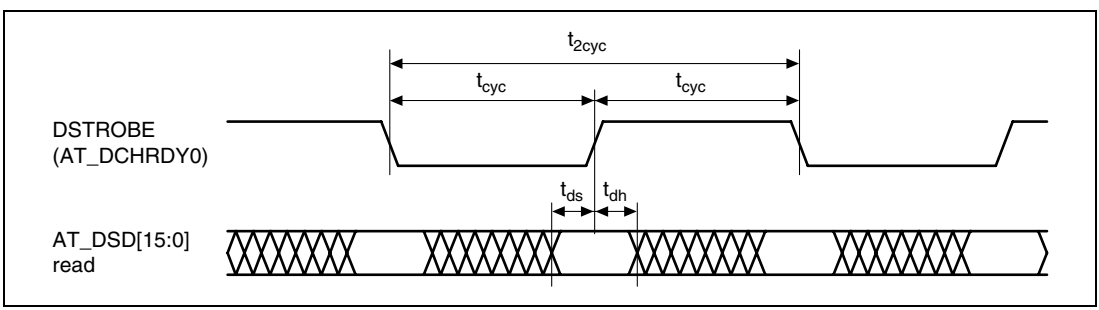


Figure 31.26 Sustained Ultra DMA Data-in Burst

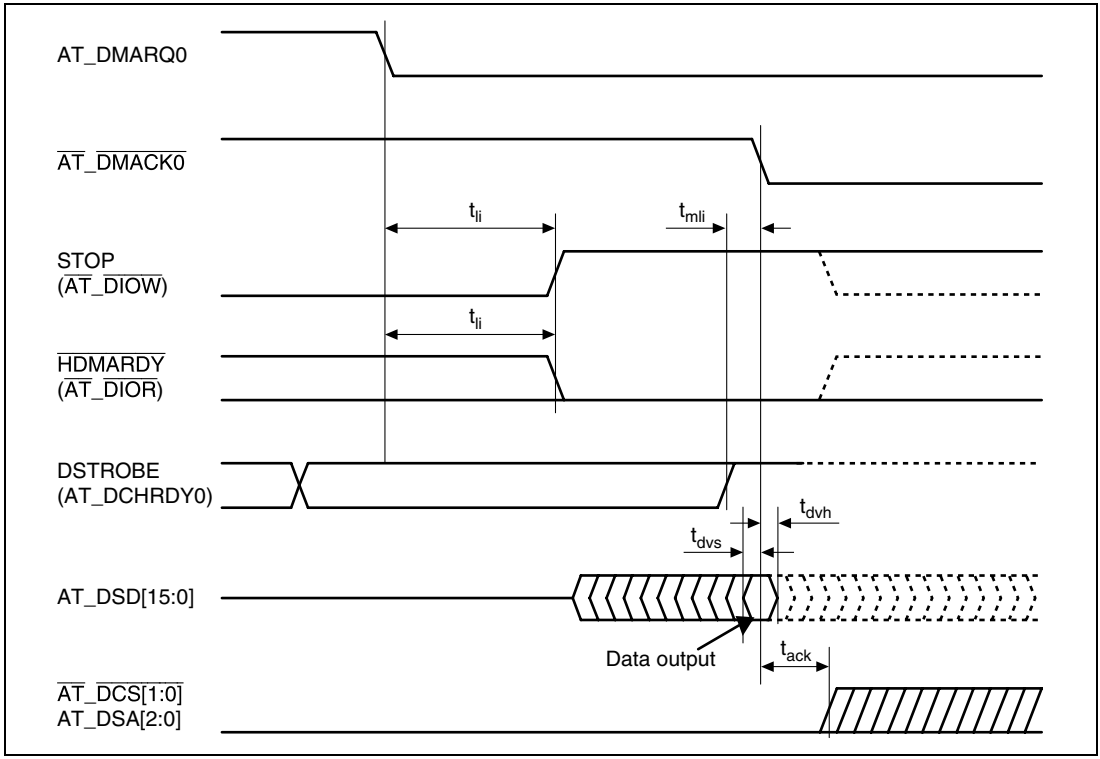


Figure 31.27 Device Terminating an Ultra DMA Data-in Burst

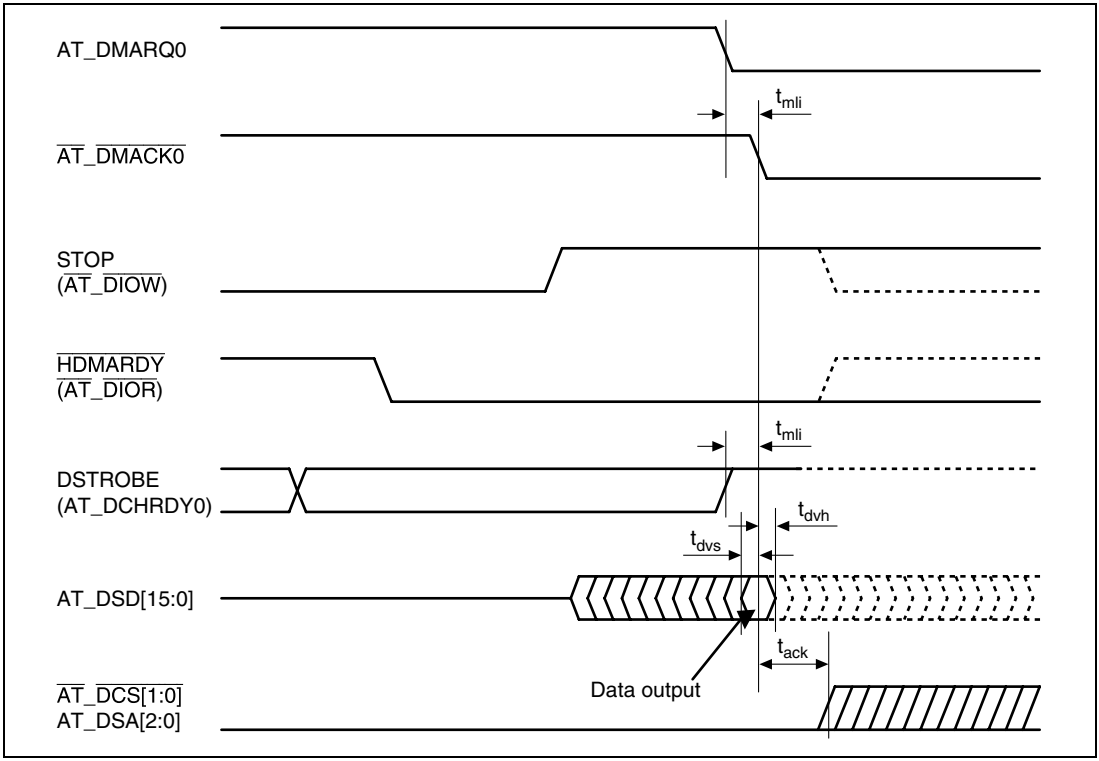


Figure 31.28 Host Terminating an Ultra DMA Data-in Burst

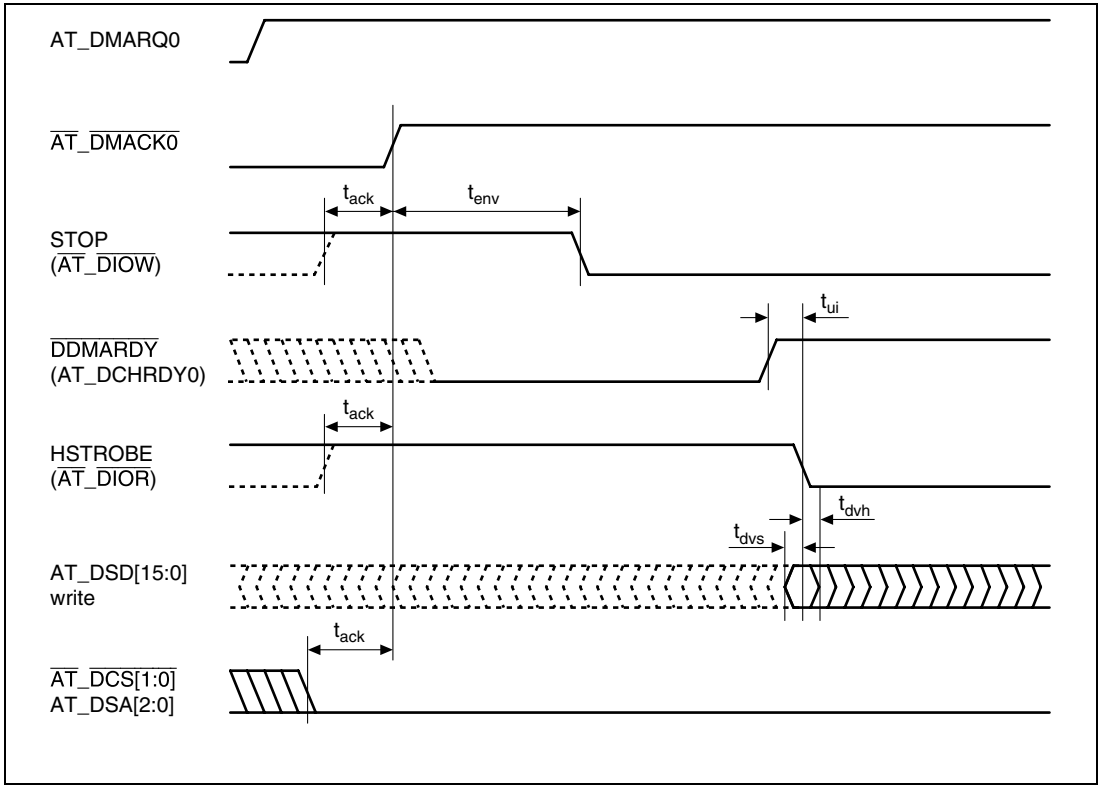


Figure 31.29 Initiating an Ultra DMA Data-out Burst

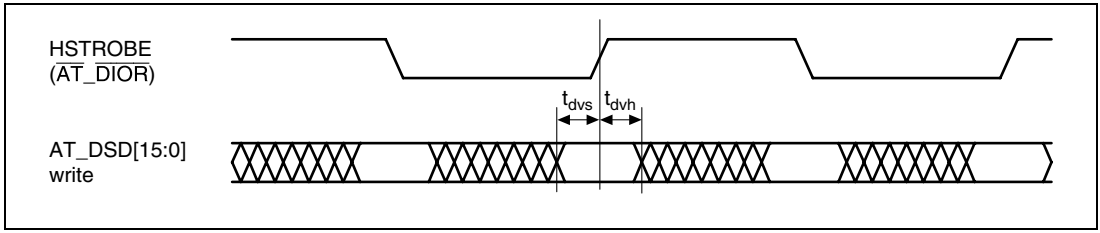


Figure 31.30 Sustained Ultra DMA Data-out Burst

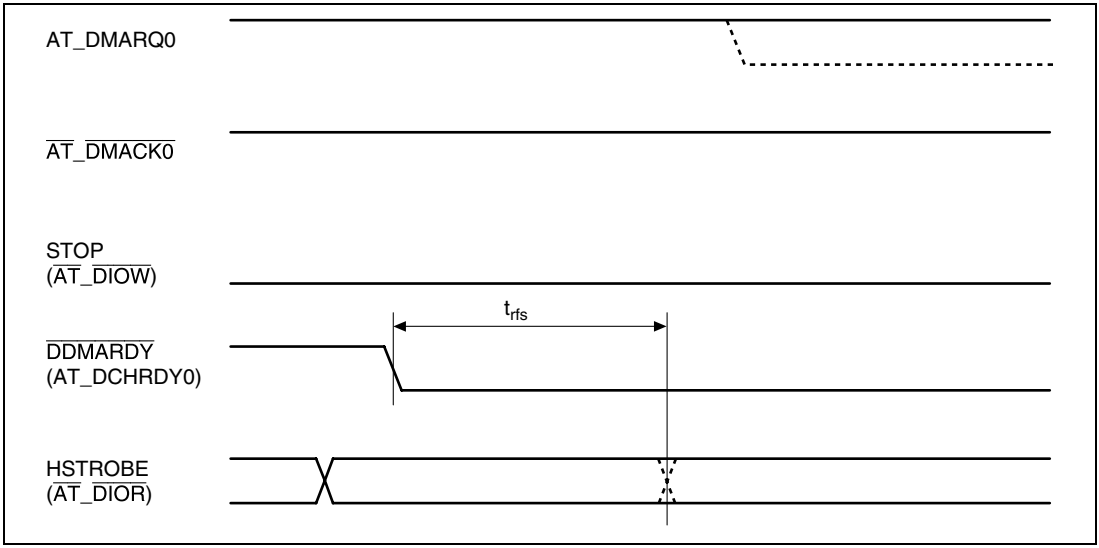


Figure 31.31 Device Pausing an Ultra DMA Data-out Burst

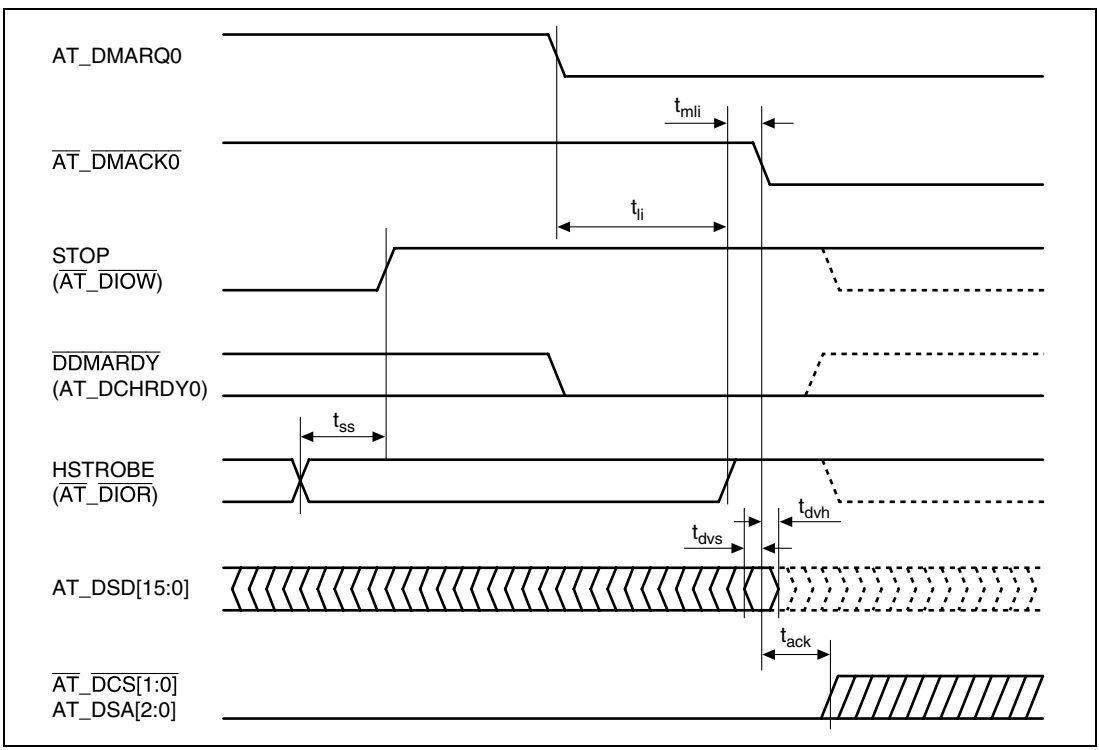


Figure 31.32 Host Terminating an Ultra DMA Data-out Burst

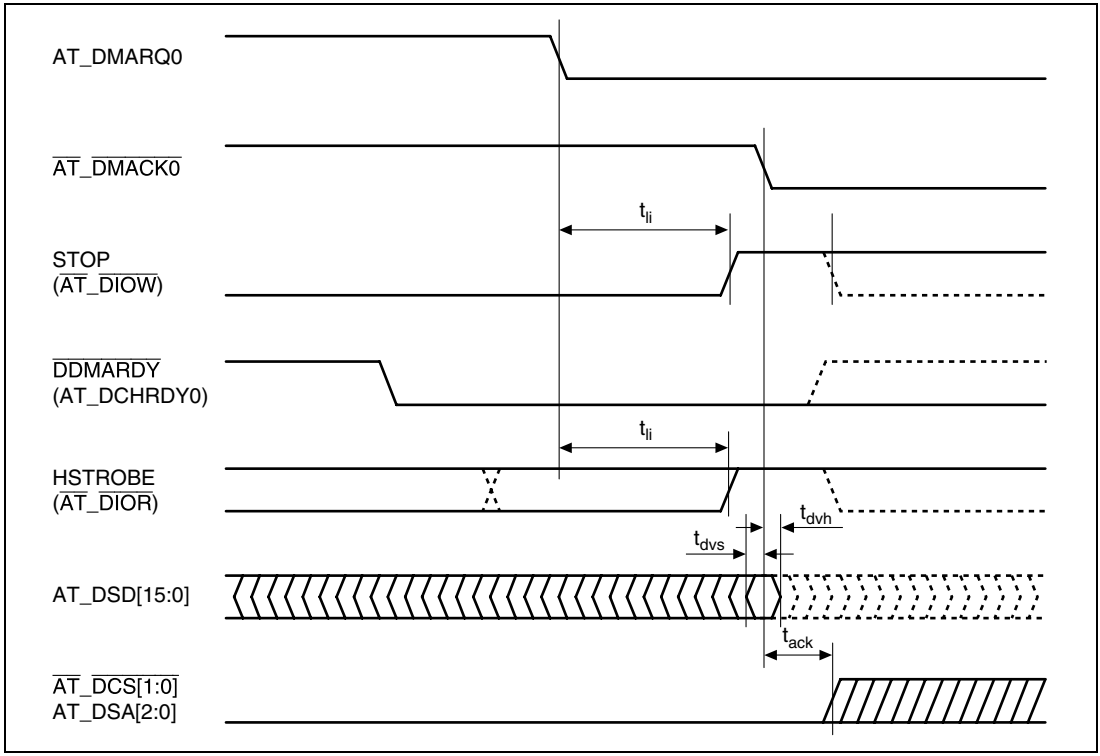


Figure 31.33 Device Terminating an Ultra DMA Data-out Burst

31.14 SSI Interface

Table 31.14 SSI Interface

Item	Symbol	Min	Typ	Max	Unit	Note
Output clock period	To	40.7	—	708.6	ns	Output
Input clock period	Ti	80	—	3300	ns	Input
Clock HIGH	t_{HC}	65	—	—	ns	
Clock LOW	t_{LC}	65	—	—	ns	Bidirection
Clock rise-time	t_{RC}	—	—	60	ns	Output
Delay	t_{dtr}	—	—	50	ns	Transmit
Set-up time	t_{sr}	10	—	—	ns	Receive (Including EF)
Hold time	t_{htr}	5	—	—	ns	Receive (Including EF)

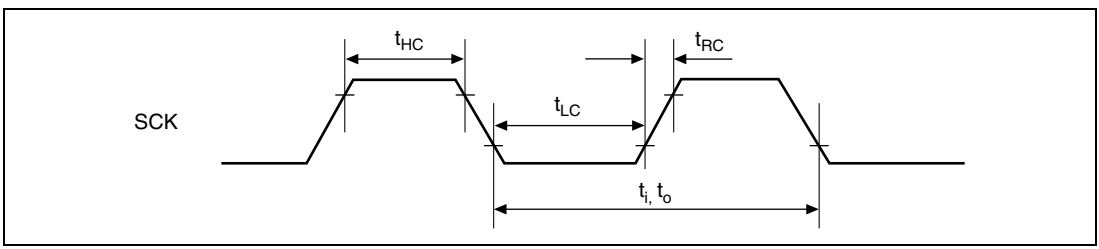


Figure 31.34 Clock Input, Output Timing

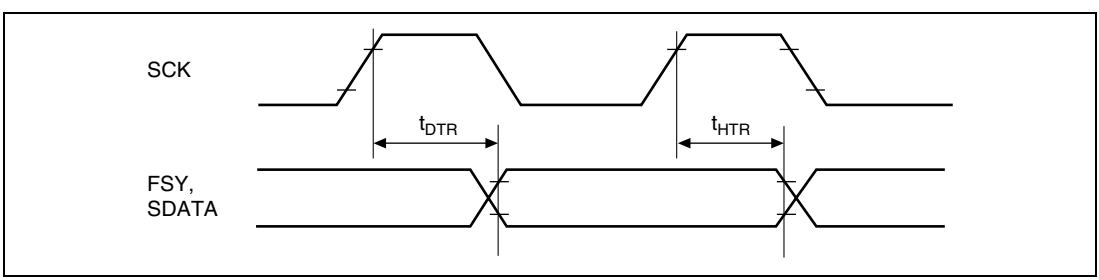


Figure 31.35 Timing for SSI Transmitter (1)

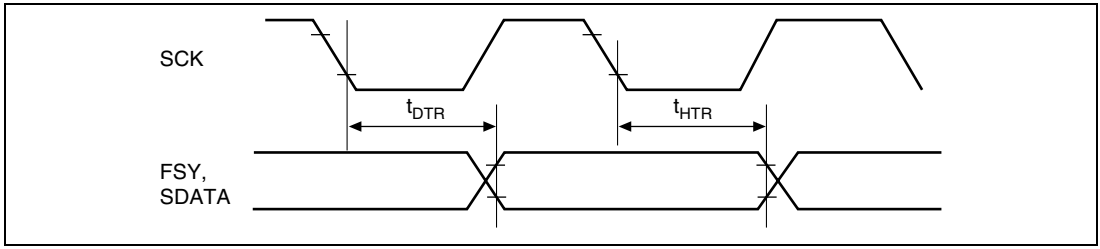


Figure 31.36 Timing for SSI Transmitter (2)

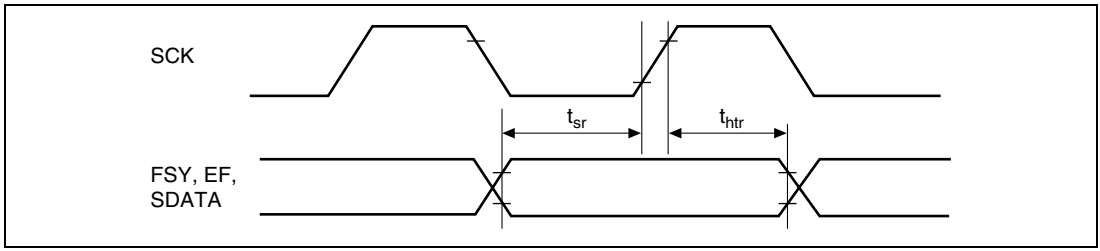


Figure 31.37 Timing for SSI Receiver (1)

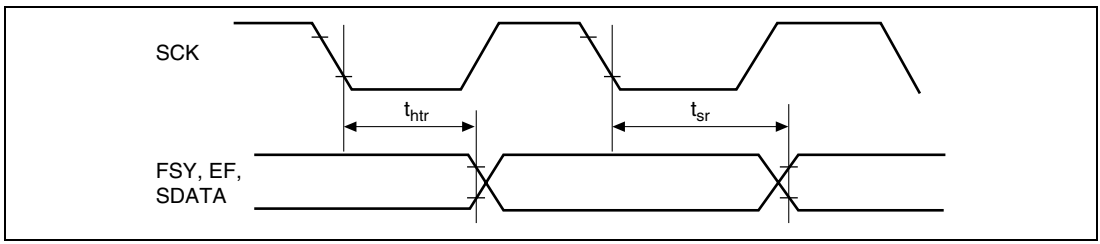


Figure 31.38 Timing for SSI Receiver (2)

31.15 Expansion Bus Interface

31.15.1 Timing Information

Table 31.15 lists the timing specifications for the expansion port signals. The "RW" column indicates whether a parameter applies to a write cycle (Figure), a read cycle (Figure), or both. All times are in ns. A reference to T1, T2, or T3 means "the start of the cycle T1/2/3".

Note that these timings are preliminary and extremely likely to change.

Table 31.15 Timing Characteristics for Expansion Port Accesses

Symbol	R/W	Min	Max	Description of Parameter
T1	—	20	100	Internal reference clock period around which timings are based
t_{AD}	RW	0	15	Address valid offset relative to T1
t_{CS}	RW	—	15	Chip select/ALE offset relative to T1 (assert) or T3 (negate)
t_{RD}	R	—	15	Read strobe offset relative to T2 (assert) or T3 (negate)
t_{WR}	W	—	15	Write strobe offset relative to T2 (assert) or T3 (negate)
t_{DON}	W	0	—	Delay before data bus is driven, after the end of cycle T1
t_{DV}	W	—	15	Delay before data bus is valid, from the start of the cycle
t_{DNV}	W	0	—	Delay before data bus is invalid relative to the end of T3
t_{DOF}	W	—	15	Delay before data bus is tristated relative to the end of T3
t_{WDH}	W	5	—	Data hold time relative to negation of ex_csn or ex_wrn
t_{DNZ}	R	0	—	Delay from ex_rdn assertion to peripheral driving data bus
t_{DZ}	R	—	15	Delay from ex_rdn negation to peripheral releasing data bus
t_{RDS}	R	—	15	Data setup time relative to start of cycle T3 (read)
t_{RDH}	R	5	—	Data hold time relative to earliest of ex_rdn or ex_csn negation

31.15.2 Notes on Timing Diagrams

The timing diagrams below indicate the characteristics of the expansion port. Standard clock intervals are labelled as T1, T2 and T3 in the timing diagrams, and wait intervals are W1, …, W7. The extra multiplexed-mode clock intervals are T1a, T1b, T1c.

For non-multiplexed operation, the minimum access time is 3 clock periods, when the number of wait states is set to zero. The minimum access time for multiplexed mode is 6 clock periods.

31.15.3 Write Cycle Timing Diagram—Non-multiplexed Address and Data Bus

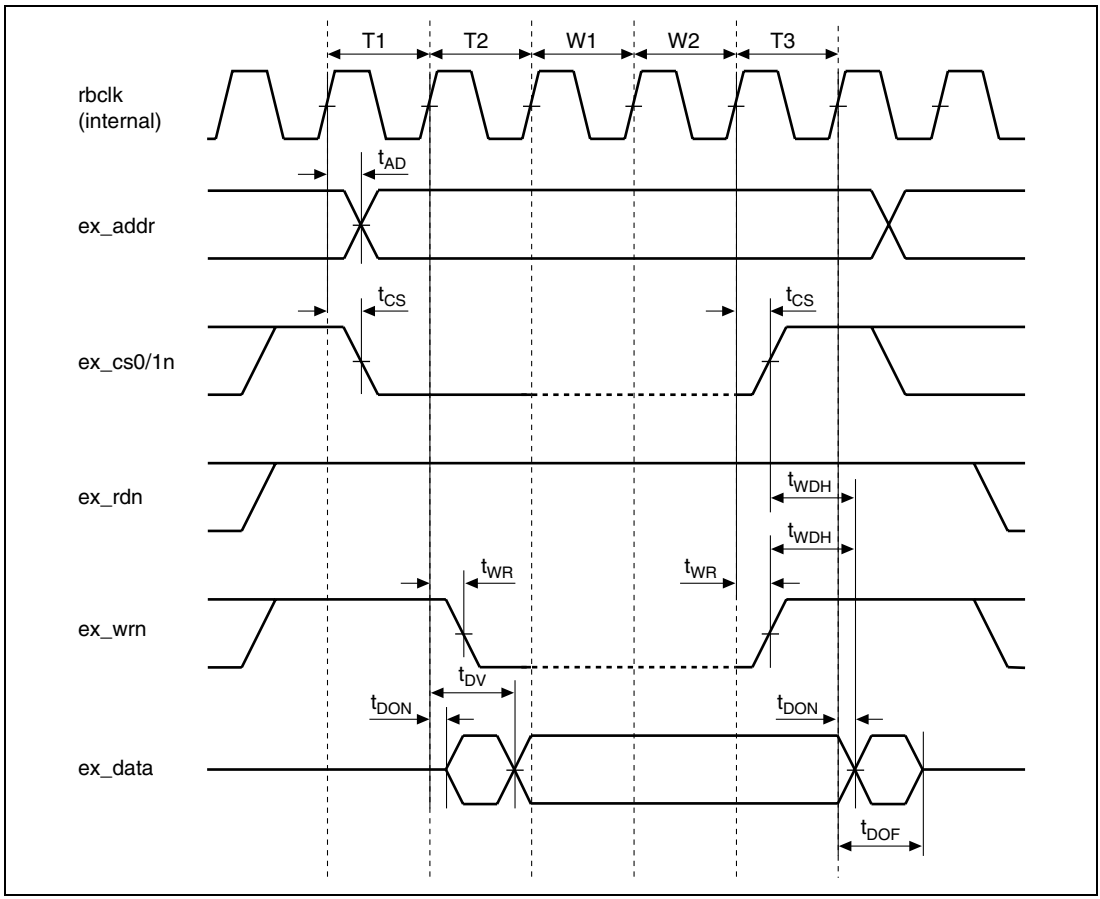


Figure 31.39 Timing Diagram for a Non-multiplexed Write Cycle with 2 Wait States

31.15.4 Read Cycle Timing Diagram—Non-multiplexed Address and Data Bus

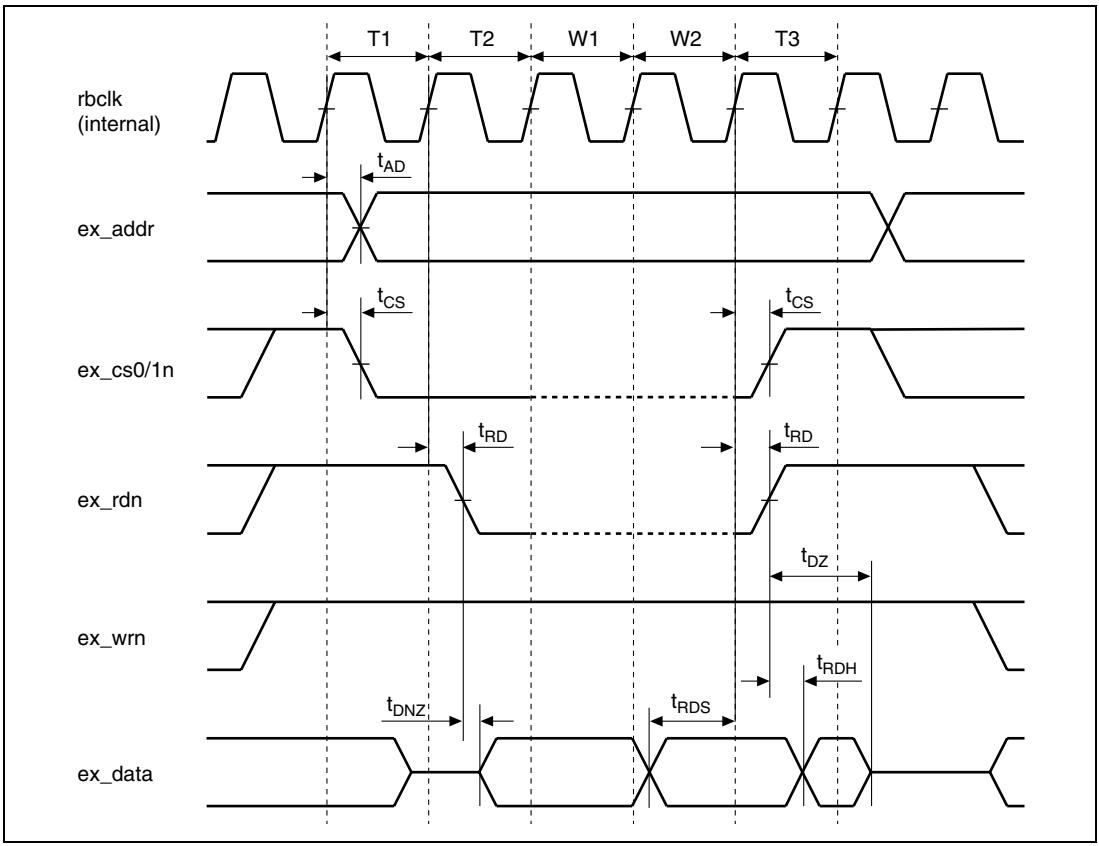


Figure 31.40 Timing Diagram for a Non-multiplexed Read Cycle with 2 Wait States

31.15.5 Write Cycle Timing Diagram—Multiplexed Address and Data Bus (ALE Mode)

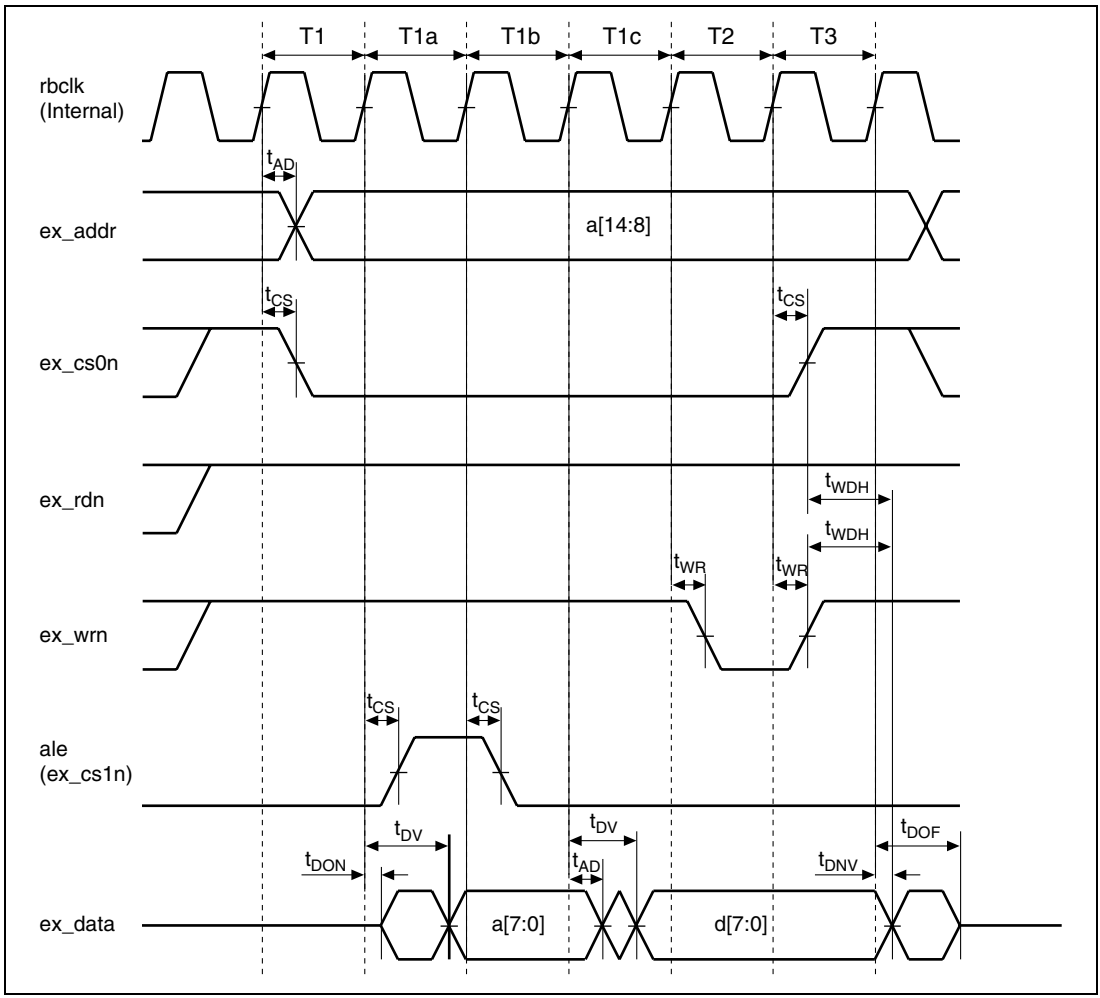


Figure 31.41 Timing Diagram for a Multiplexed Write Cycle Using No Wait States

31.15.6 Read Cycle Timing Diagram—Multiplexed Address and Data Bus (ALE Mode)

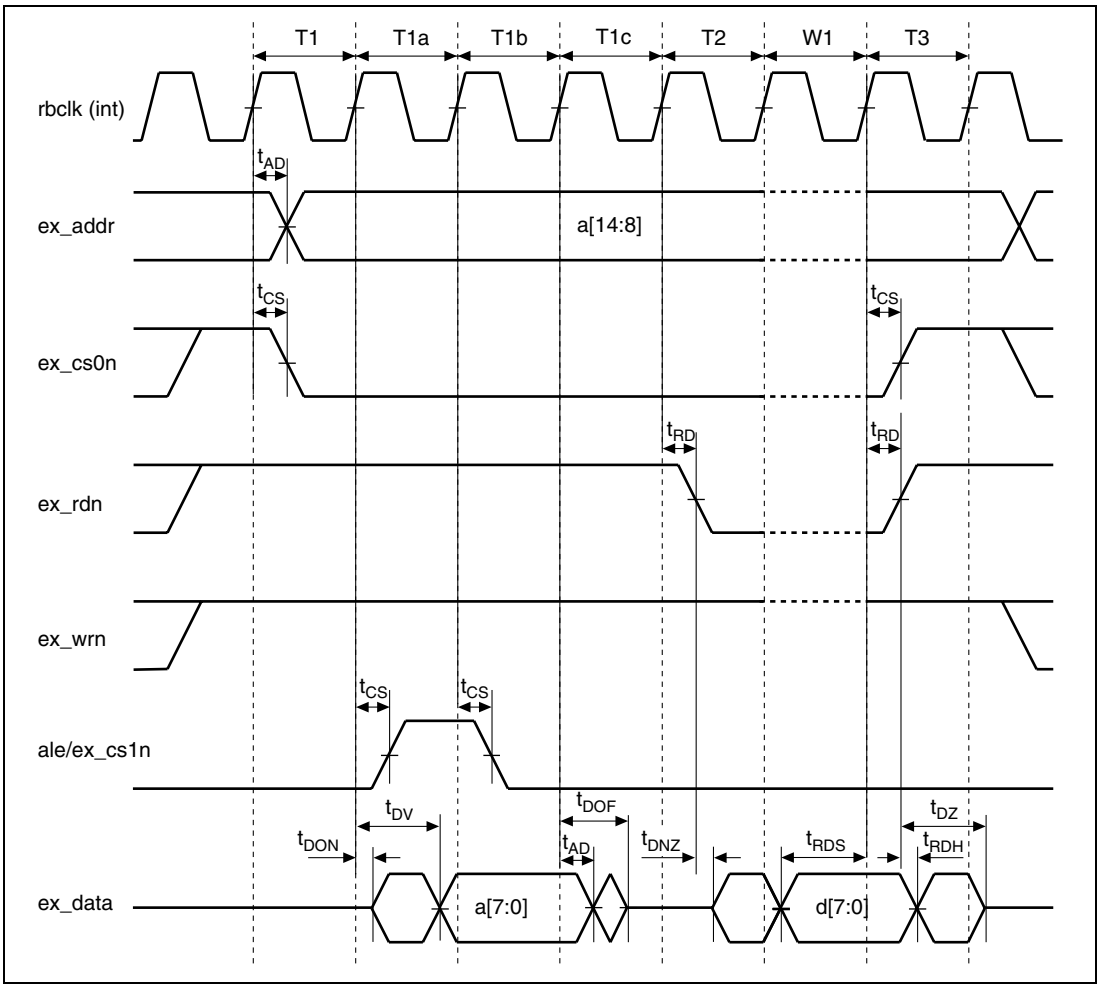


Figure 31.42 Timing Diagram for a Multiplexed Read Cycle with 1 Wait State

31.16 USB Timing

Table 31.16 USB Clock Timing

Item	Symbol	Min	Typ	Max	Unit	figure
Input clock frequency (48 MHz)	t_{FREQ}	47.9	—	48.1	MHz	
Input clock rising time	t_{R48}	—	—	2	ns	
Input clock falling time	t_{F48}	—	—	2	ns	
duty (t_{HIGH}/t_{LOW})	t_{DUTY}	90	—	110	%	

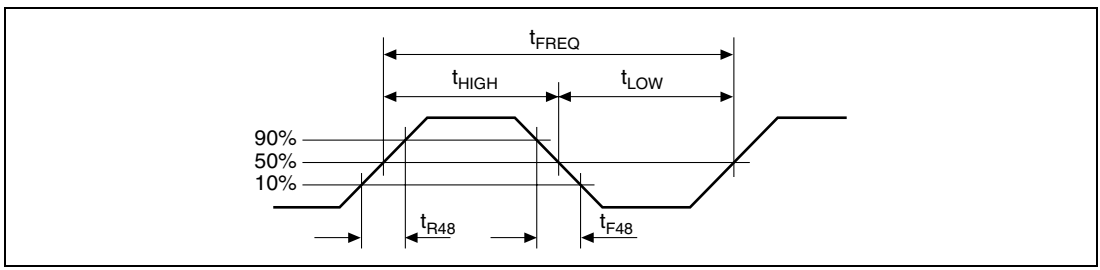


Figure 31.43 USB Clock Timing

Table 31.17 USB Transceiver Timing (Full Speed)

Item	Symbol	Min	Typ	Max	Unit	Note
Rise time	t_R	4	—	20	ns	VCC = 3.3 V typ
Fall time	t_F	4	—	20	ns	

Table 31.18 USB Transceiver Timing (Low Speed)

Item	Symbol	Min	Typ	Max	Unit	Note
Rise time	t_R	75	—	300	ns	VCC = 3.3 V typ
Fall time	t_F	75	—	300	ns	

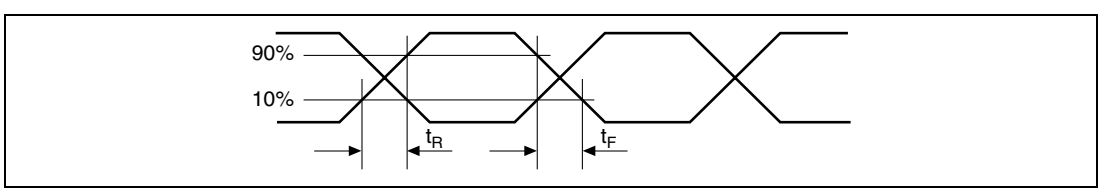


Figure 31.44 USB Transceiver Timing

31.17 SPI Timing

Table 31.19 SPI Timing

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
SPI Clock Cycle	t_{spicyc}	—	—	Register bus clock (MHz)/8, → 6.25 MHz	MHz	
SPI Clock High Width	t_{spihw}	60	—	—	ns	
SPI Clock Low Width	t_{spilw}	60	—	—	ns	
SPI TX Setup Time	$t_{suspitx}$	—	—	20	ns	
SPI TX Delay Time	t_{dspitx}	—	—	20	ns	
SPI RX Setup Time	$t_{suspirx}$	20	—	—	ns	
SPI RX Hold Time	$t_{hlspirx}$	20	—	—	ns	
SPI CS lead Time	t_{cslead}	100	—	—	ns	

All Signals Latched and Edged by rbclk

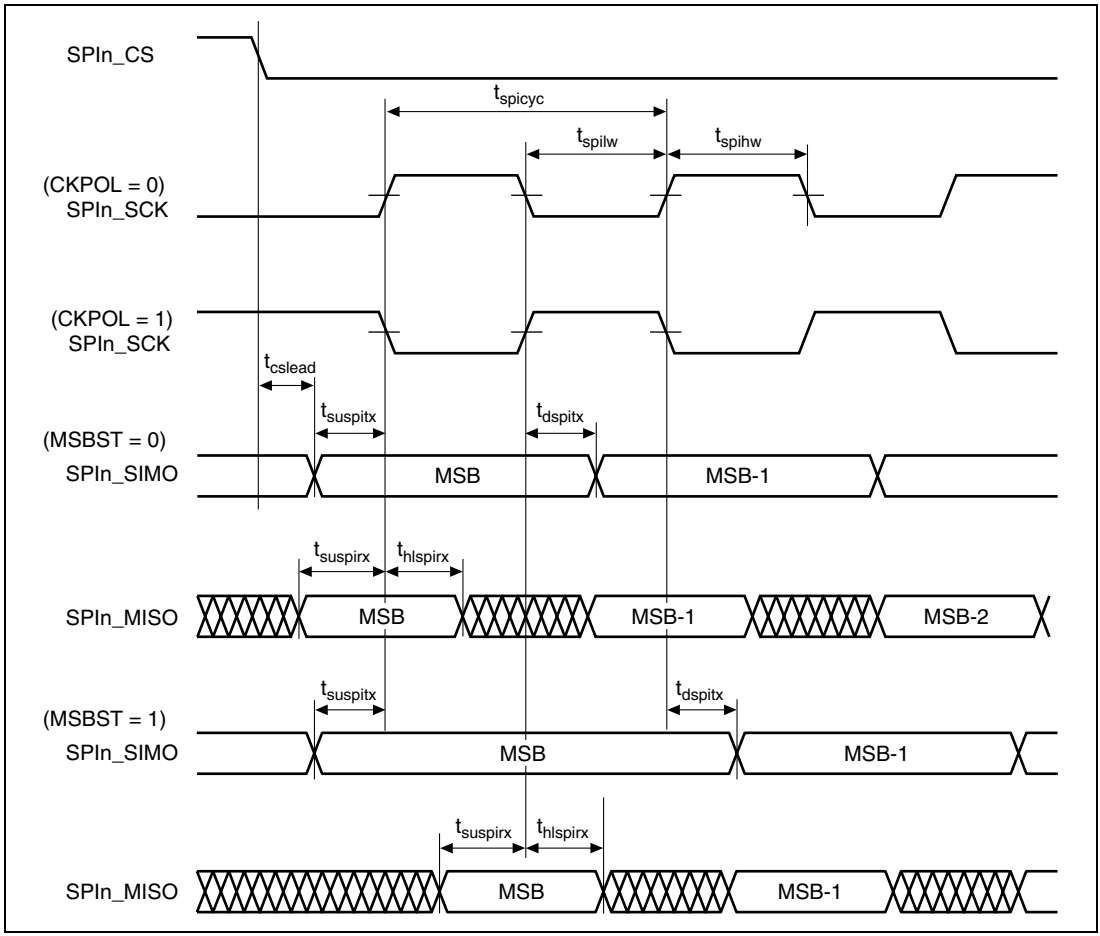


Figure 31.45 SPI Data Output/Input Timing

31.18 MOST Interface

Table 31.20 MOST Interface

Item	Symbol	Min	Typ	Max	Unit	Note
MPAD valid	Tpadvld	15	—	—	ns	Output
MRDN low	Trdlow	50	—	—	ns	Output
MPAD hold	Tpadhd	4	—	—	ns	Output
MRDN high	Trdhigh	25	—	—	ns	Output
MDATA read setup	Tdatrdsup	10	—	—	ns	Output
MDATA read hold	Tdatrdhd	5	—	—	ns	Output
MPAD delay to MCP_FLOW	Tcptopad	50	—	—	ns	Output
MWRN low	Twrlow	50	—	—	ns	Output
MWRN high	Twrhigh	25	—	—	ns	Output
MDATA write delay	Tdatwrdly	—	—	20	ns	Output
MDATA write hold	Tdatwrhd	4	—	—	ns	Output

All signals are latched and edged by rbclk.

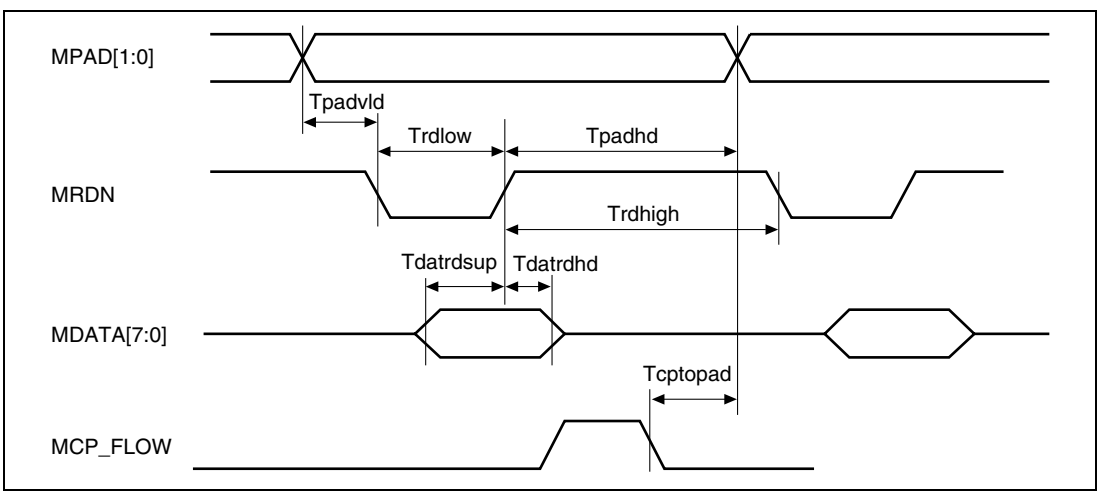


Figure 31.46 MOST Interface Timing (1)

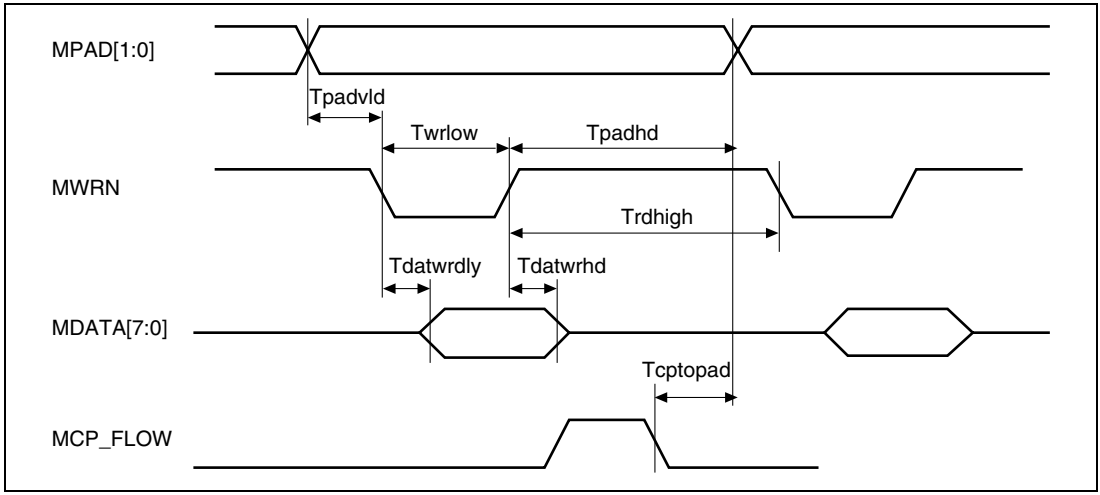


Figure 31.47 MOST Interface Timing (2)

31.19 Audio Codec Interface

Table 31.21 Audio Codec Timing Spec.

Parameter	Symbol	Min	Max	Unit	Note
AC_RES Active Low Pulse Width	t_{RST_LOW}	1000	—	ns	
AC_SYNC Active HIGH Pulse Width	t_{SYN_HIGH}	1000	—	ns	
AC_SYNC Delay Time 1	t_{SYNCD1}	—	15	ns	
AC_SYNC Delay Time 2	t_{SYNCD2}	—	15	ns	
AC_SDATA_IN Delay Time	t_{SDOUTD}	—	15	ns	
AC_SDATA_IN Setup Time	t_{SDINSU}	10	—	ns	
AC_SDATA_IN Hold Time	t_{SDINHD}	10	—	ns	

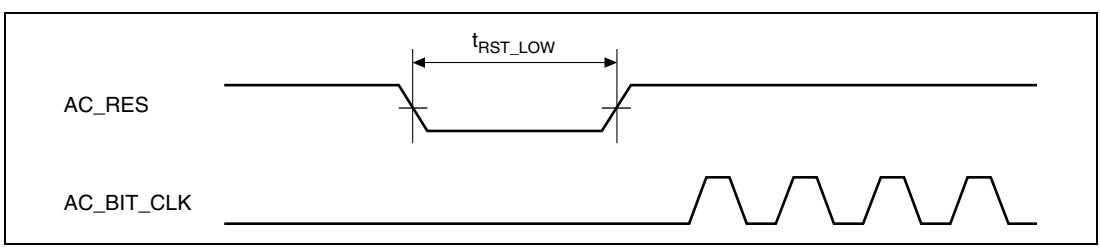


Figure 31.48 Cold Reset Timing

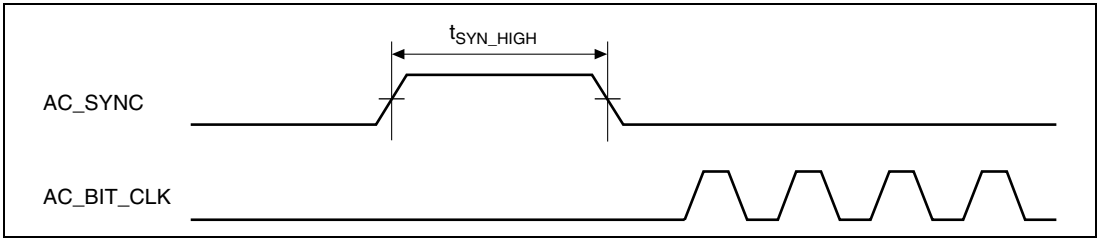


Figure 31.49 Warm Reset Timing

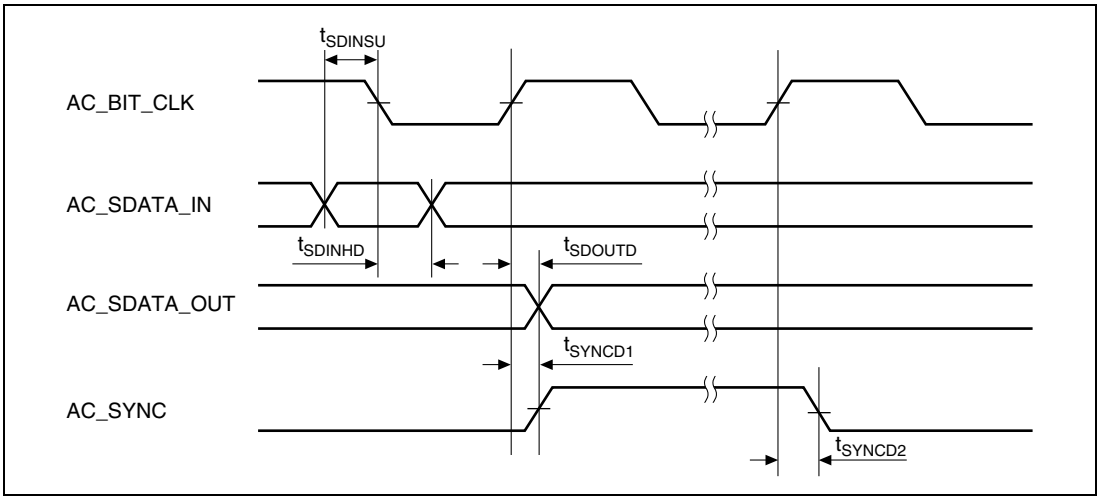


Figure 31.50 Audio Codec Interface Timing

31.20 JTAG Interface

Table 31.22 JTAG Interface

Module	Item	Symbol	Min	Max	Unit	Figure
JTAG	Input clock cycle	t_{TCKcyc}	1000	—	ns	
	Input clock pulse width (high)	t_{TCKH}	300	—	ns	
	Input clock pulse width (low)	t_{TCKL}	300	—	ns	
	Input clock rise time	t_{TCKr}	—	10	ns	
	Input clock fall time	t_{TCKf}	—	10	ns	
	TDI/TMS setup time	t_{TDIS}	300	—	ns	
	TDI/TMS hold time	t_{TDIH}	50	—	ns	
	TDO delay time	t_{TDO}	0	100	ns	

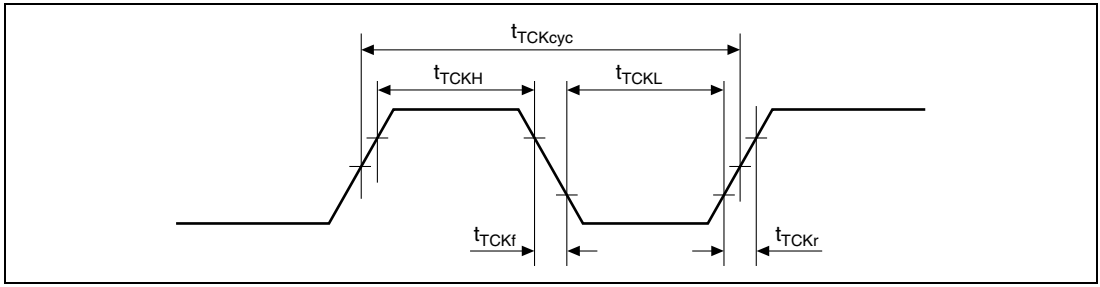


Figure 31.51 TCK Input Timing

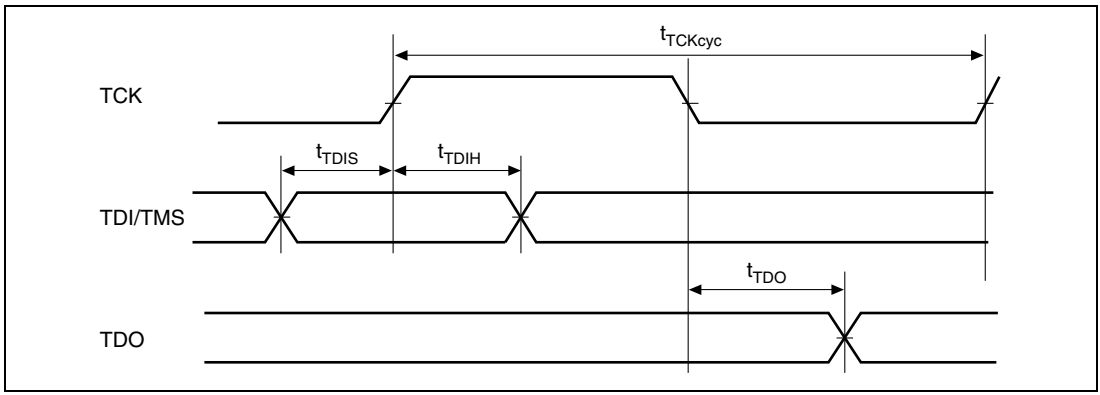


Figure 31.52 JTAG Data Transfer Timing

31.21 Electrical Characteristics Test Mode

31.21.1 Timing Testing

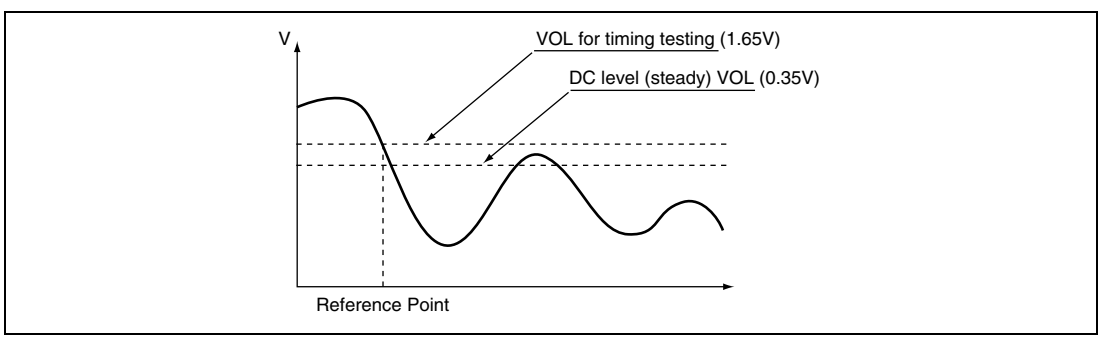


Figure 31.53 Timing Testing

AC Characteristics condition : Output signal reference is 1.65 V.

31.21.2 Test Load Circuit (All Output and Input/Output Pins)

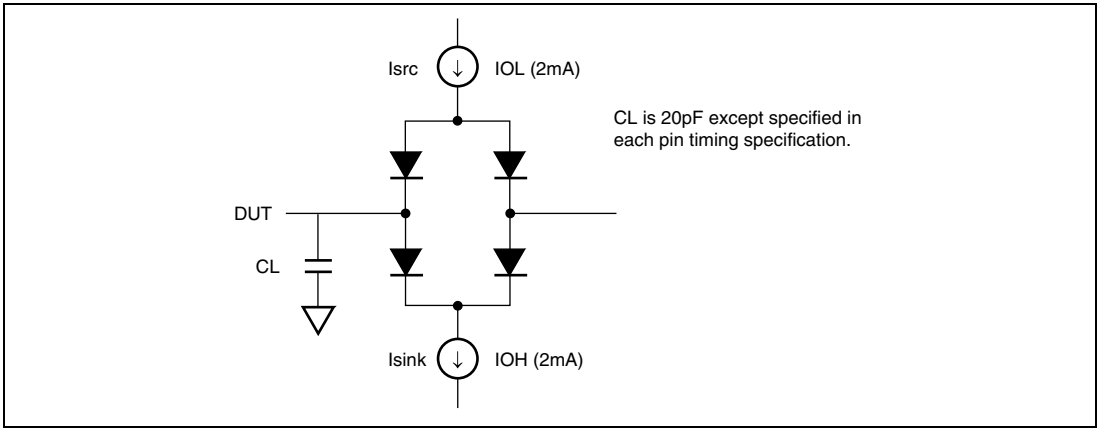


Figure 31.54 Test Load Circuit

31.22 Notes on Usage

31.22.1 Notes on Power Supply Pins

Be sure to insert bypass capacitors CB1 and CB2 (See Figure 31.55 and ′Notes on PLL Usage′)

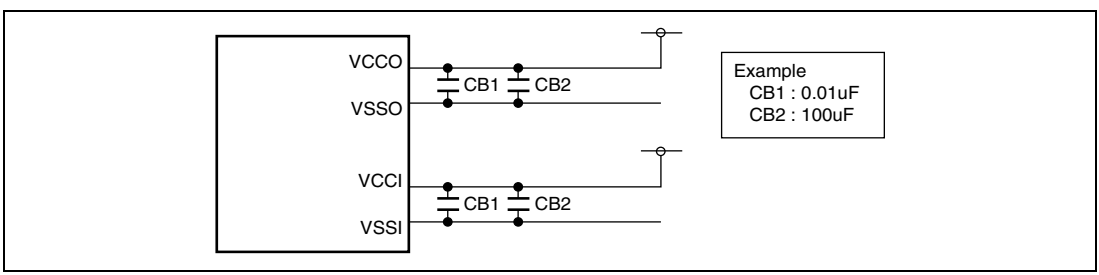


Figure 31.55 Notes on Power Supply

31.22.2 Notes on PLL Usage

Be sure to place capacitors CA1 and CA2 (See Figure 31.56) near the pins to stabilize oscillation without crossing other signal lines. Ensure those separate lines for VSSPLLA2, VCCPLLA1, VSSPLLA1, VSSPLLA2, VCCQ, VCCI, VSSQ, VSSI are provided from board′s power supply.

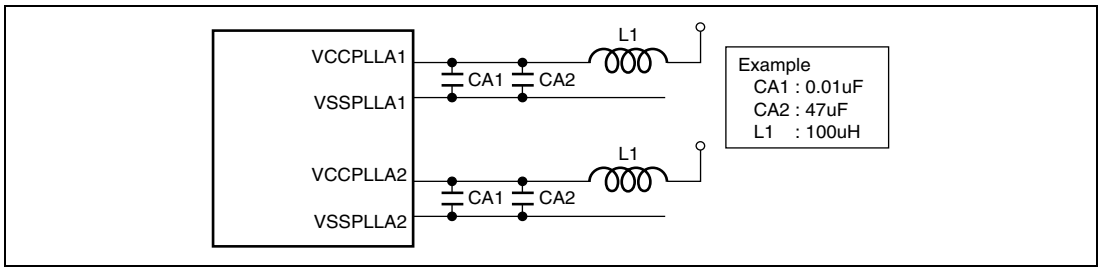
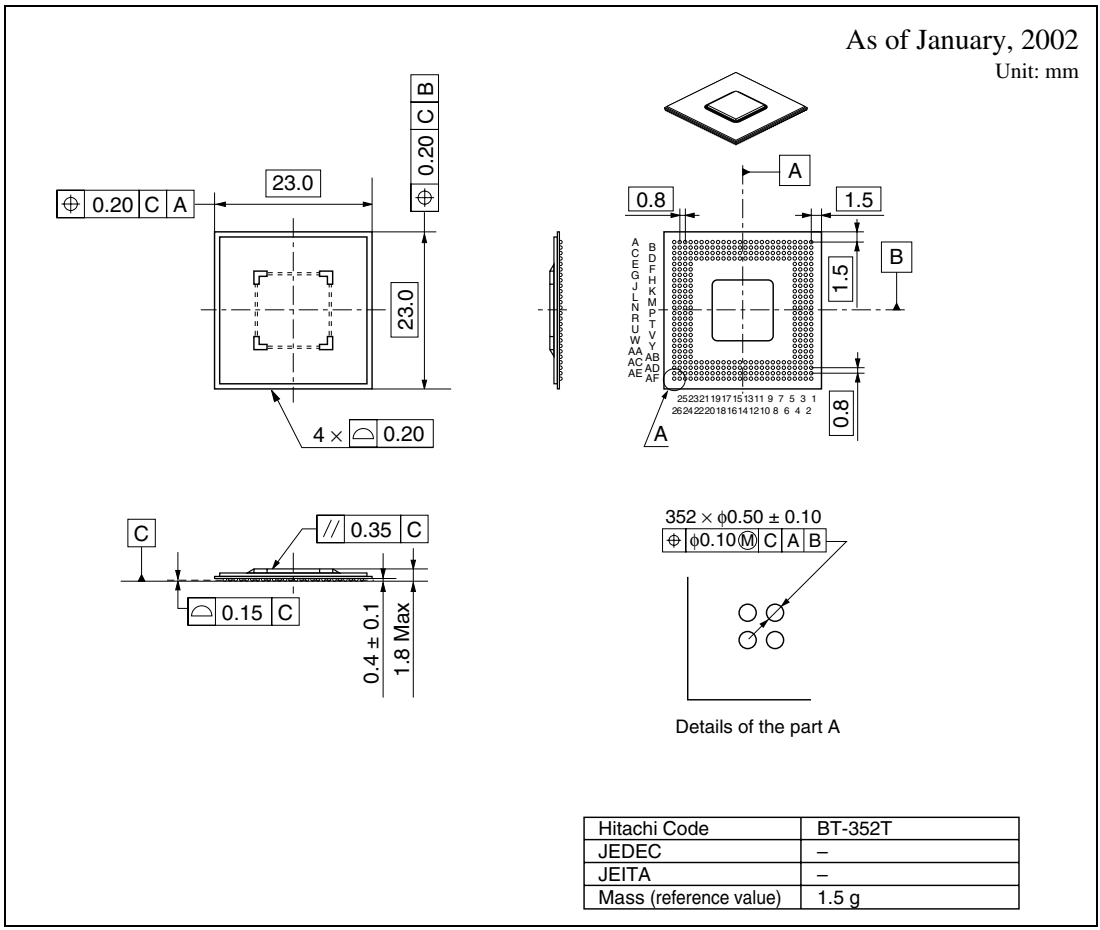


Figure 31.56 Notes on PLL Usage

Appendix A Package Dimensions

The HD64404 package dimensions are shown in figure A.1 (TBGA-352).



Hitachi Code	BT-352T
JEDEC	–
JEITA	–
Mass (reference value)	1.5 g

Figure A.1 Package Dimensions (TBGA-352)

HD64404 Hardware Manual

Publication Date: 1st Edition, September 2002
Published by: Business Operation Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.