Windows® CE Intelligent Peripheral Controller

HD64465

User's Manual

HITACHI

ADE-602-168B

Rev. 3.0
03/08/01
Hitachi Ltd.

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.

Revisions and Additions in this Edition

Page	Item	Description
-	Description of register address	<Former Edition> 0x……. / …….h <This Edition> H'…….
-	Description of HD64465BQ	Add the description of new product "HD64465BQ"
-	HD64465BP Specifications Changed	<Former Edition> PLL used in bypass mode and 48 MHz clock input into UCK terminal <This Edition> PLL used in bypass mode
16	Table 4.1 HD64465BP Signal Names (by pin numbers in alphabetical order)	Add the description of M12 to M15 pins
17		Add the description of N12 to N15, P12 to P15, R12 to R15 pins
		Signal name of R1 pin
26	Table 4.4 Pin Descriptions of CPU Interface	<Former Edition> RESET# <This Edition> RESETPI#
39	Table 4.24 Pin Descriptions of No Connected Pins	Correct the table name <Former Edition> Pin Descriptions of LCD Interface <This Edition> Pin Descriptions of No Connected Pins
	Table 4.25 Pin Descriptions of Power/Ground	Pin description of AVSS3 <Former Edition> Ground for analog circuit (can not connected) <This Edition> NC (No Connected Pin)
52	6.3 Register Dexcription Table 6.1 The Register List Of Power Management and System Configuration	<Former Edition> Register Size = 2 Access Size = 2 <This Edition> Register Size = 16 Access Size = 16

Page	Item	Description
54	6.3.2 System Configuration Register (SCONFR) Description of Bits 11 - 8	Add the following description Note that the relationship between HW[3:0] and CPU programmed inserted wait states (IWS) is 2 ≤ 1 WS ≤ 1+HW[3:0]. Hence, the CPU default inserted wait states should be 2.
58	6.3.4 System Peripheral Clock Control Register (SPCCR)	<Former Edition> Note: The parameter, Twkst, please refer the AC timing specification. <This Edition> Note: The parameter, Twkst = 15 ms
59	Figure 6.1 AFECK & LCK Related Clock Diagram	Add the description ofACCLK pin
71	7.3.1 Port Data Register	<Former Edition> GPCDR -- Address: 100004018h <This Edition> GPEDR -- Address: H'100004018
91	Table 9.1 The Register List of Timer Module	<Former Edition> Register Size = 2 Access Size = 2 <This Edition> Register Size = 16 Access Size = 16
93	9.2.3 TRVR1:Timer 1 Read Vlue Register	<Former Edition> R/W value = R/O <This Edition> R/W value = R
94	9.2.4 TRVR0:Timer 0 Read Value Register	
99	9.2.9 PWM1CS: PWM1 Clock Scale Register Bits 5 - 0	Delete the default value
102	9.2.12 PWM0CS: PWM0 Clock Scale Register Bits 5 - 0	
121	10.4.8 PCC1 General Control Register (PCC1GCR)	Description of bit 7 <Former Edition> PCC0 <This Edition> PCC1
		Description of bit 4 <Former Edition> VCC0SEL1 <This Edition> VCC1SEL0

Page	Item	Description
152	(26) Timing control Register (ITMCR) Bits 5 - 0	Delete the default value
176	(2) ECP Address FIFO Register (ECPAFifo)	Default value of bits 2 - 0 <Former Edition> 111b <This Edition> 111
	(4) Device Control Register (der) (Address H'1000A004, Mode All)	Default value of bits 2 - 0 <Former Edition> 11b <This Edition> 11
227	Table 14.3 AC97 Timing	<Former Edition> Unit : Us Ns <This Edition> Unit : µs ns
336	19.2.1 A/D Data Registers A to D (ADDRA to ADDRD, ADCAL)	<Former Edition> Bit 9 : AD8 Bit 8 : AD9 <This Edition> Bit 9 : AD9 Bit 8 : AD8
341	19.3.2 Acan Mode (SCAN = 1)	Add the ADCSR table
352, 353	Table 20.13 AFECK clock input AC Timing Spec. (PLL1:bypass) (unit : ns) Table 20.14 AFECK clock input AC Timing Spec. (PLL1:operatings) (unit : ns) Table 20.15 AFECK clock input AC Timing Spec. (PLL2:bypass) (unit : ns) Table 20.16 AFECK clock input AC Timing Spec. (PLL2:operating) (unit : ns)	New tables added
363, 364	Figure 20.26 AFECK Clock Input Timing Figure 20.27 UCK Clock Input Timing	New figures added

HD64465BP Specifications Changed

1. Change in Specifications

Item changed	Guaranteed value before change	Guaranteed value after change
PLL stabilization time	5ms	Not guaranteed

2. Major Influences Due to Above Change in Specifications

Preconditions	Problems in usage
PLL used in bypass mode	No problem (no standby time added)
Reset signal input after turning power on	No problem (no standby time added)
PLL standby not used although oscillation stop used	No problem (no standby time added)
Certain interval allowed between PLL standby and wakeup	Several seconds may be required for access by HD64465BP from CPU.

3. Countermeasures

- Use the system in a state without any problem by referring to "2." above.
- Create your program in a way to prohibit access other than to the HD64465BP system configuration register (offset address: H'00000000 to H'00000ff0) for a certain period of time after returning from PLL standby.

4. Debugged Version

   The cause of this problem has already been clarified, which can be solved by correcting the wiring layer. The new mask product (HD64465EBP) is available as the debugged version.

Contents

Tables

Figures

Section 1 Features

1.1 CPU Interface

- Supports Hitachi SH-4/SH7709/SH3-DSP family of CPUs with bus speeds from 15 MHz up to 66 MHz.
- Supports STANDBY mode when CKIO is stopped.
- Memory mapped on area 4 of SH4/SH7709/SH3DSP for internal registers
- 3.3 V-CMOS interface
- 32 - bit data interface

1.2 PCMCIA Controller

- PCMCIA PC card standard v2.1 compliant
- Supports dual PCMCIA memory or IO cards at SH4/SH7709/SH3-DSP area 5 and area 6
- 8- or 16-bit PCMCIA interface support
- Mixed voltage (3.3V or 5V) operation is fully supported for PCMCIA address, data and control signals.
- Supports TI TPS2206 serial interface
- Supports STANDBY mode

1.3 AFE Interface

- Supports SGS-THOMSON STLC7546 and STLC7550 interface
- Read buffer and write buffer are provided for performance enhancement
- Supports STANDBY mode

1.4 GPIO Function(Port Interrupt)

- GPIO pins can be programmed as input, output ports, or as interrupt inputs.
- Internal pull-up resistor ON/OFF control
- Interrupt events can be independently generated or masked on each I/O pin.
- Interrupt can be independently programmed to rising edge or falling edge trigger
- Power down control by software (input gated and output floating)
- Maximum 40 bits for I/O port functions

1.5 Interrupt Controller

- Provides an interrupt to SH-4/SH7709/SH3-DSP, which is generated by an internal module interrupt request
- Module interrupts can be masked on/off by setting the registers.

1.6 Power Management

- Supports STANDBY mode to stop clock for each module
- All clock inputs can be stopped
- All built-in PLLs can be set to STANDBY mode
- The CPU input signals can be gated

1.7 Timer

- 2-channel 16-bit auto-reloaded timer with pre-scale (1, 1/4, 1/8, 1/16) for dividing CKIO
- Supports generating DMA or Interrupt request whenever timer's count reaches zero
- Supports generating ADC external trigger whenever timer's count reaches zero
- Provides two-channel Pulse Width Modulation (PWM) for VR control of LCD.
- Supports STANDBY mode

1.8 Keyboard Controller Interface

- Supports ISA-bus-like interface to pair with the external keyboard controller
- Supports 2 channels of PS/2 interface to connect PS/2 device like keyboard and mouse.
- Supports STANDBY mode

1.9 UART

- Standard 16550 compatible full spec UART
- Supports one channel of serial port
- Supports STANDBY mode

1.10 Printer interface

- Supports three access modes, SPP, EPP and ECP(ECP mode only supports PIO mode)
- 5V interface to printer
- Supports STANDBY mode

1.11 Audio CODEC Interface

- Directly interfaced to CS4271/CS4218/AC97 Codec for controlling voice data to the speaker, or from the mic.
- Dual TX/RX FIFO (8×32 -bit) are supported for CS4271/CS4218 interface
- 12-channel TX FIFO (4×20-bit) and 9-channel RX FIFO (4×20 -bit) are supported for AC97 Codec interface
- Voice captures and playbacks can be supported by PIO or DMA mode access.
- The Codec Interface is able to provide SM3 Slave Mode for communication with CS4218 and CS4271, and SM3 Master Mode for CS4218.
- Supports AC97 version 1.03 and version 2.0 serial-link interface
- Supports STANDBY mode

1.12 IrDA

- Supports HP SIR or ASKIR infrared interface
- Supports FIR and MIR
- Provides DMA channel mode for FIR
- Supports STANDBY mode

1.13 Clock Generator and PLL

- Provides a $\times 4$ PLL from 12 MHz to 48 MHz for USB, IrDA, and Parallel Port
- Provides a $\times 3$ PLL from 12.288 MHz to 36.864 MHz for AFE, CS4218/CS4271/AC97 codec interface
- 12.288 MHz clock input for AFE interface, CODEC interface, USB, IrDA and Parallel Port
- Each clock generator and PLL supports STANDBY mode

1.14 USB Host Controller

- Supports direct interfaces of 2 USB ports
- Supports device bandwidth of 12Mbps or 1.5Mbps
- Supports power management mode to protect USB Bus power; and over-current detector to protect USB Bus from abnormal over-current load
- Fully compatible with the USB specification version 1.0 and register compatible with Open Host Controller Interface (OHCI) specification version v1.0 issued by Microsoft, Compaq and NS
- 4 K-byte SRAM provided for USB Open Host Controller driver to store frame lists, transaction descriptors for USB host controller's schedule control and this local memory is also used as data buffer for host controller to send/receive data to/from USB devices

1.15 10-bit ADC

- 10-bit resolution
- Provides four input channels
- High-speed conversion, conversion time is maximum 10 μs per channel
- Two conversion modes

  Single mode: one channel A/D conversions are supported.

  Scan mode: continuous conversions are operated in cycles from one to four channels.
- A/D conversion can be triggered by timer or software
- Supports STANDBY mode

1.16 Package

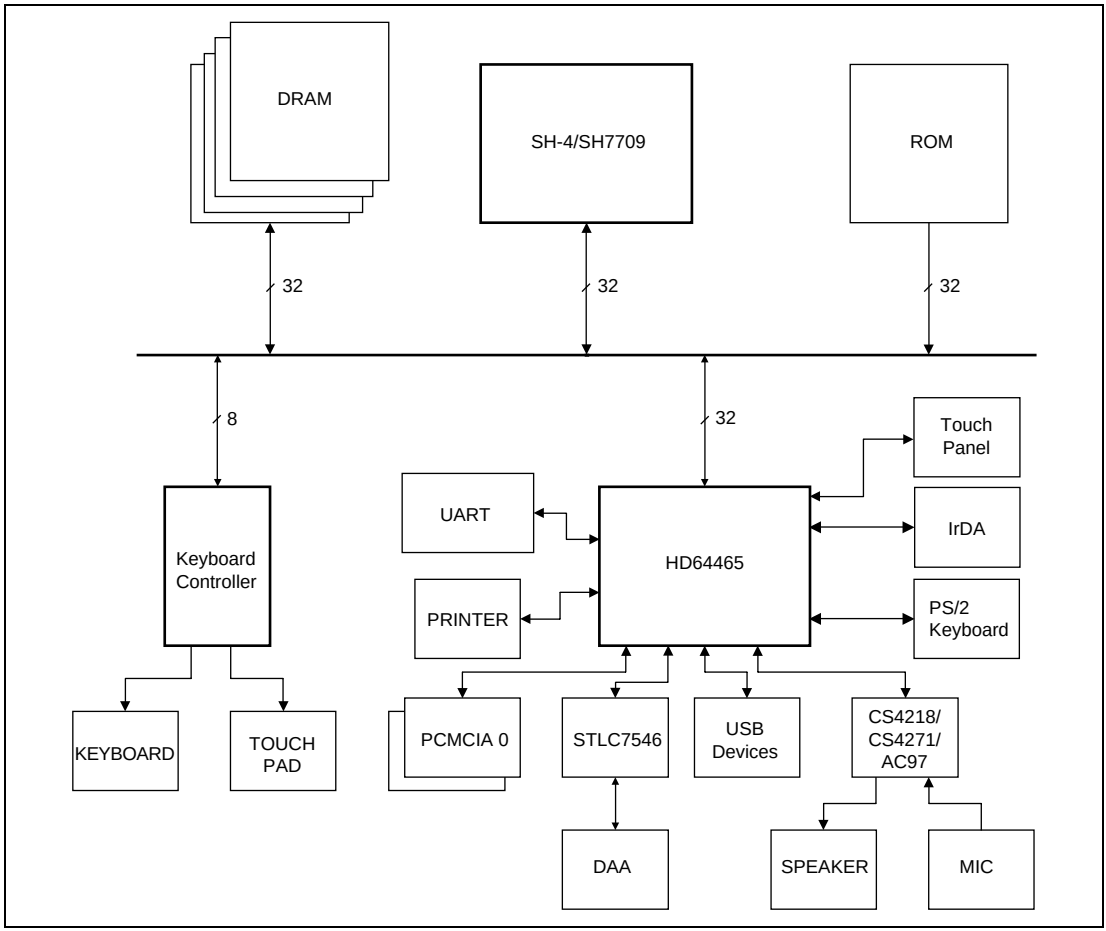
- 387-pin BGA (35 mm × 35 mm: HD64465BP)
- 387-pin BGA (27 mm × 27 mm: HD64465BQ)

Section 2 General Description

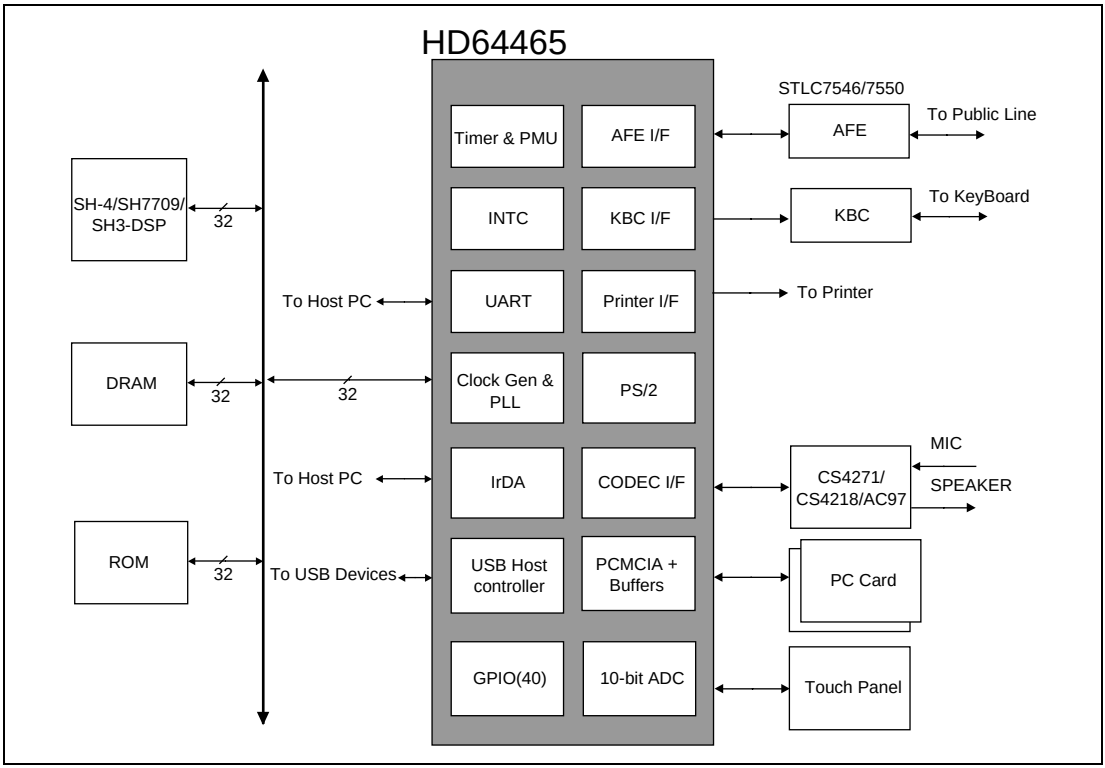
The HD64465 is directly connected to SH-4/SH7709/SH3-DSP, and consists of PCMCIA controller, analog front end (AFE) interface, I/O port controller, timer, UART, parallel port interface controller, keyboard interface, CS4218/CS4271/AC97 Codec interface, IrDA controller, USB Host controller, AC97 Codec, 10-bit ADC and power management unit. This chip pairs with SH-4/SH7709/SH3-DSP processors, and features all the key peripheral functions required by the sub sub-notebooks designed for Windows® CE v2.0 and above, providing a total solution for Windows® CE Mini NoteBook (SubsubNoteBook) PC system.

Section 3 System Block Diagram

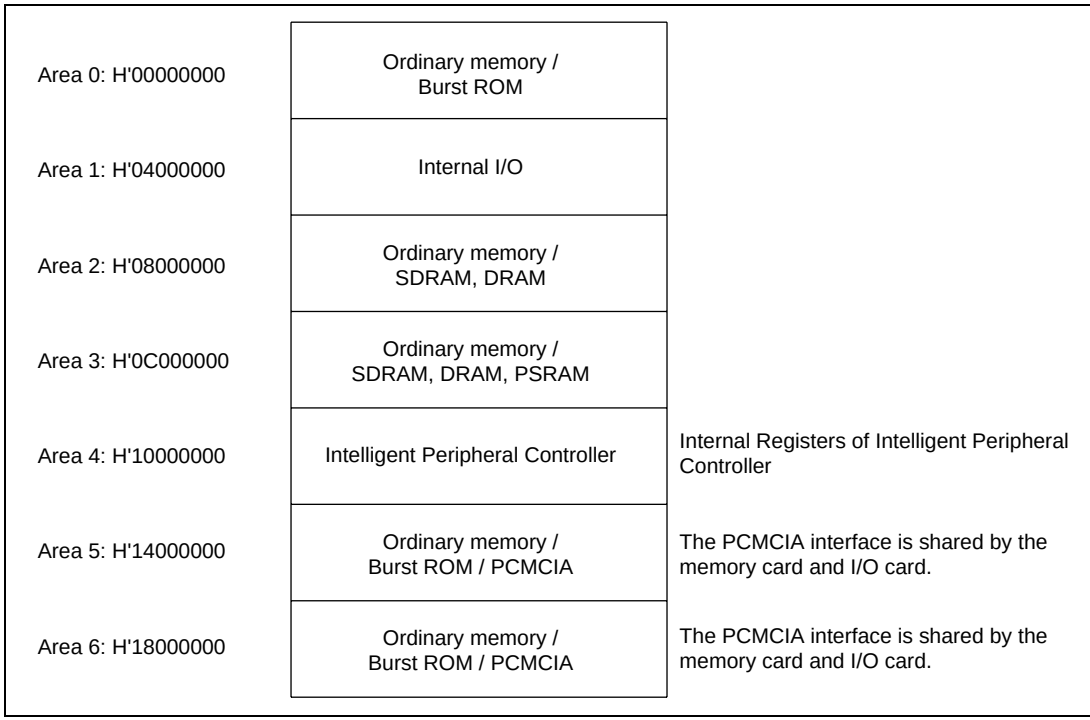
3.1 Application Circuit



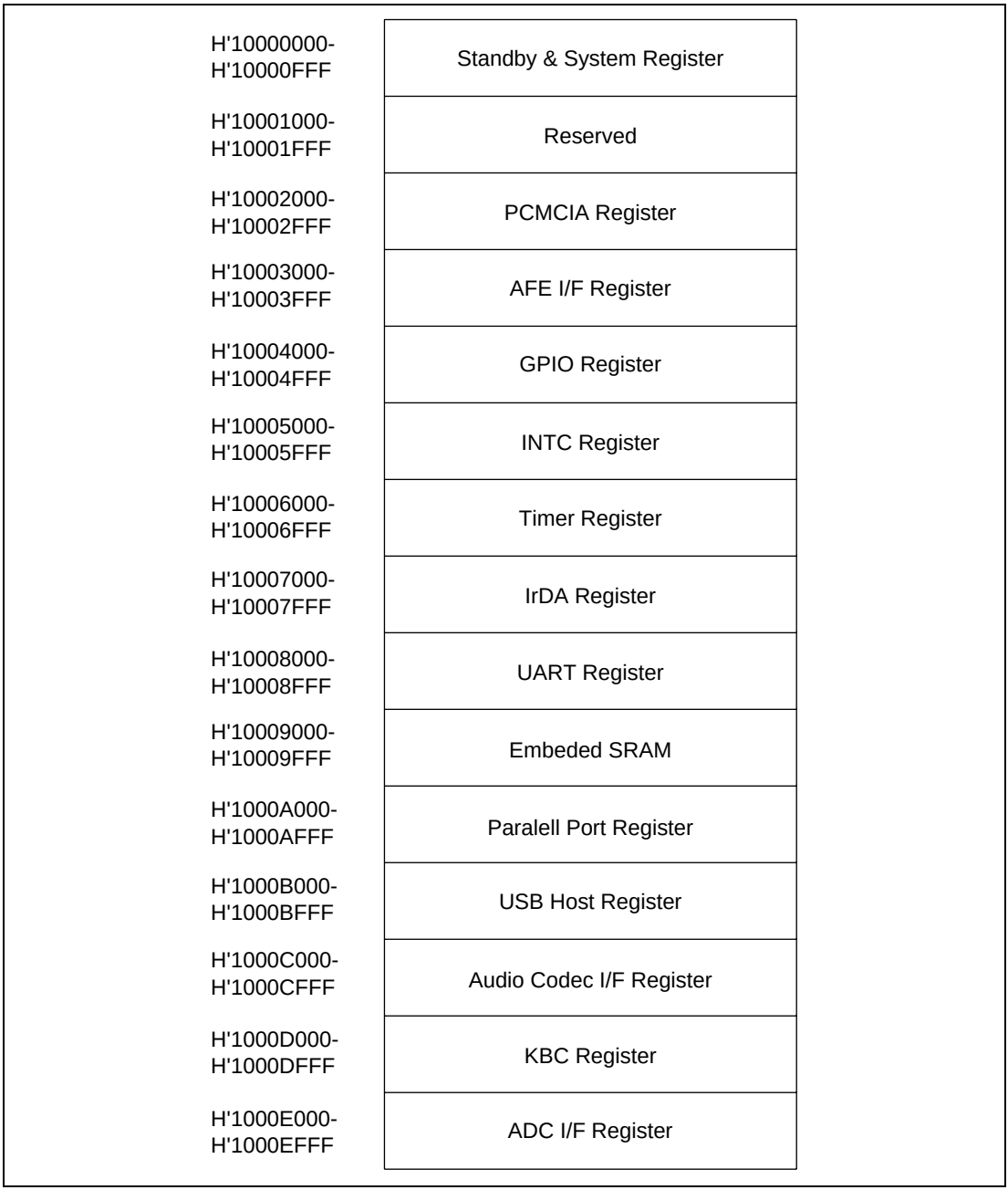
3.2 System Block Diagram



3.3 Physical Address Space

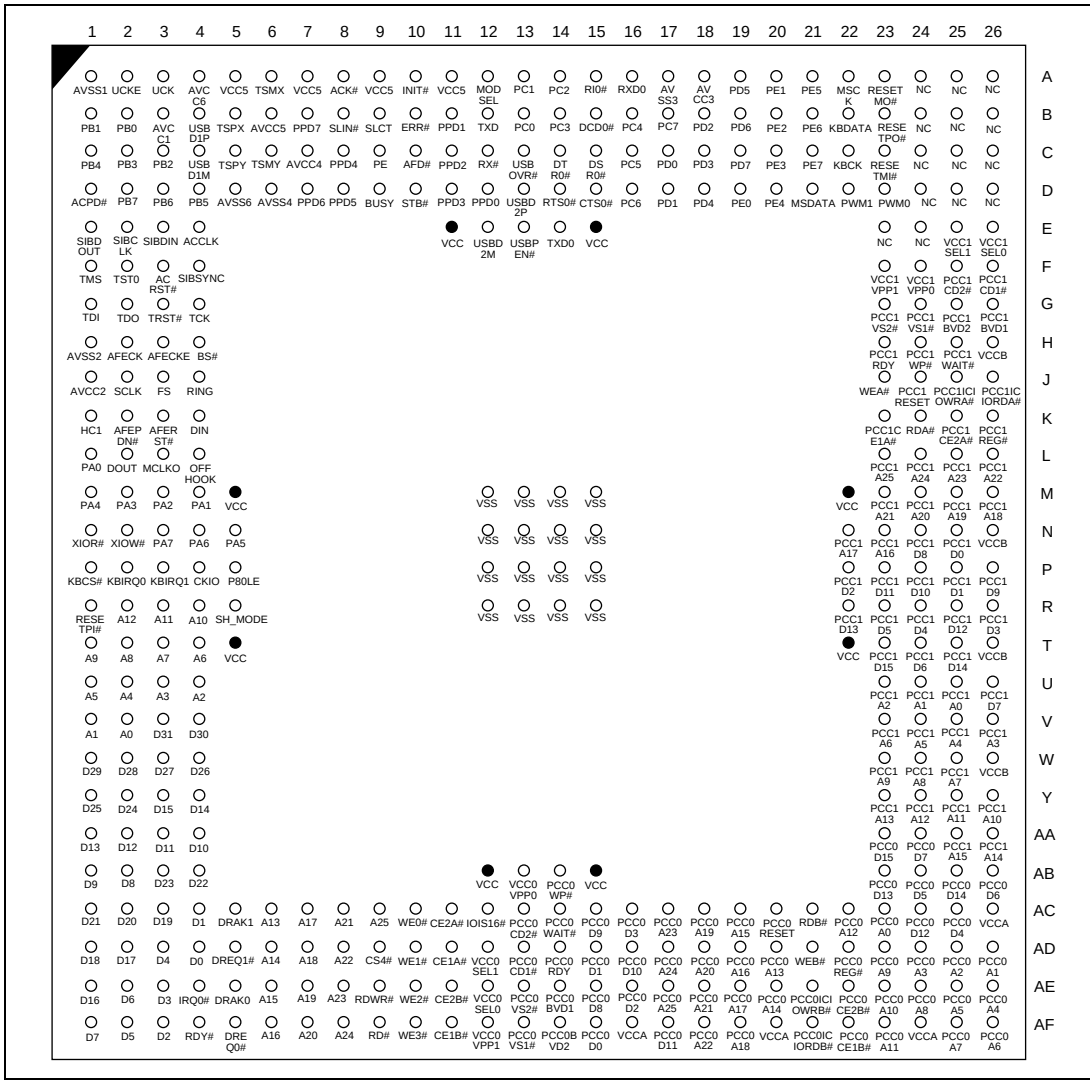


3.4 HD64465 Memory Address

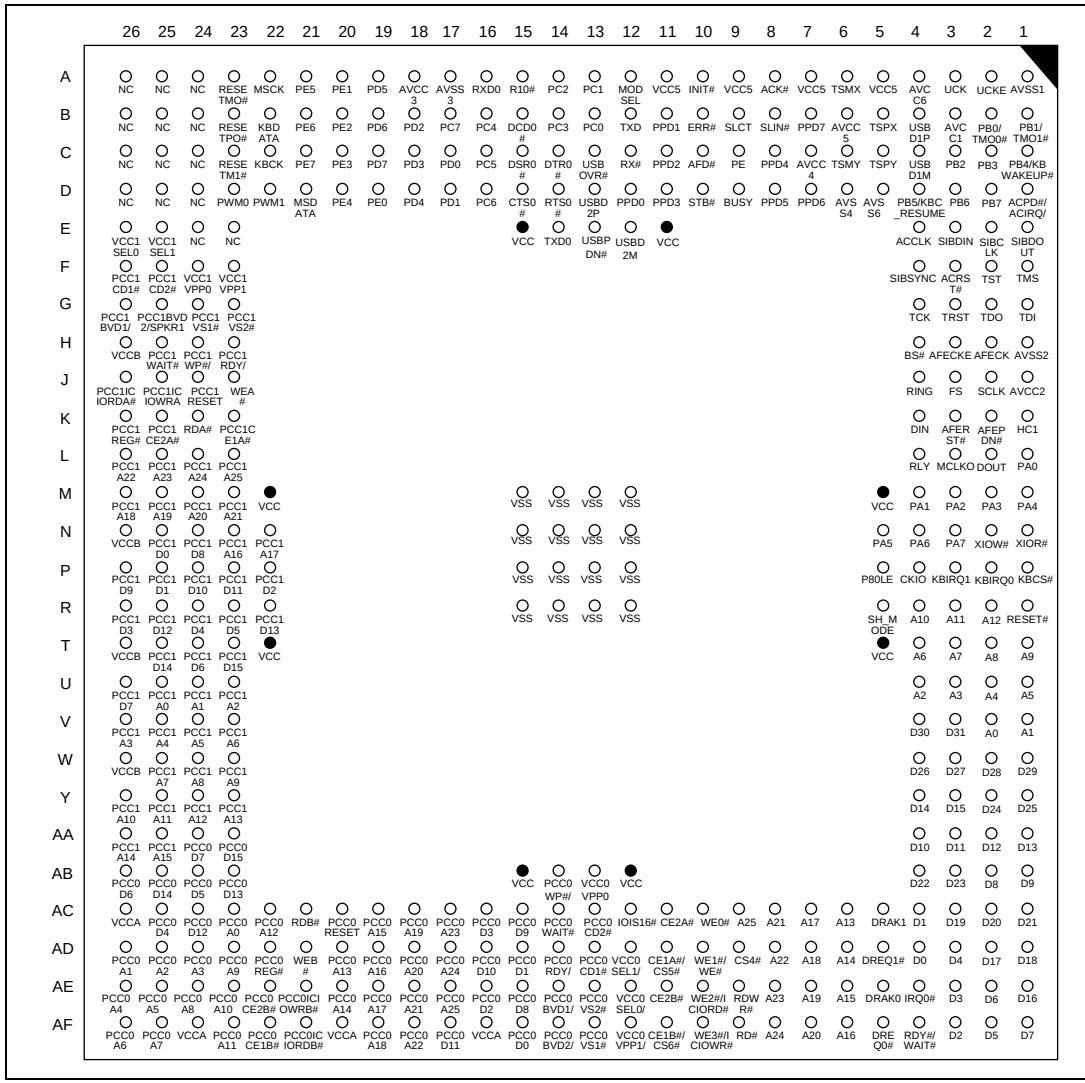


3.5 Pin Configuration

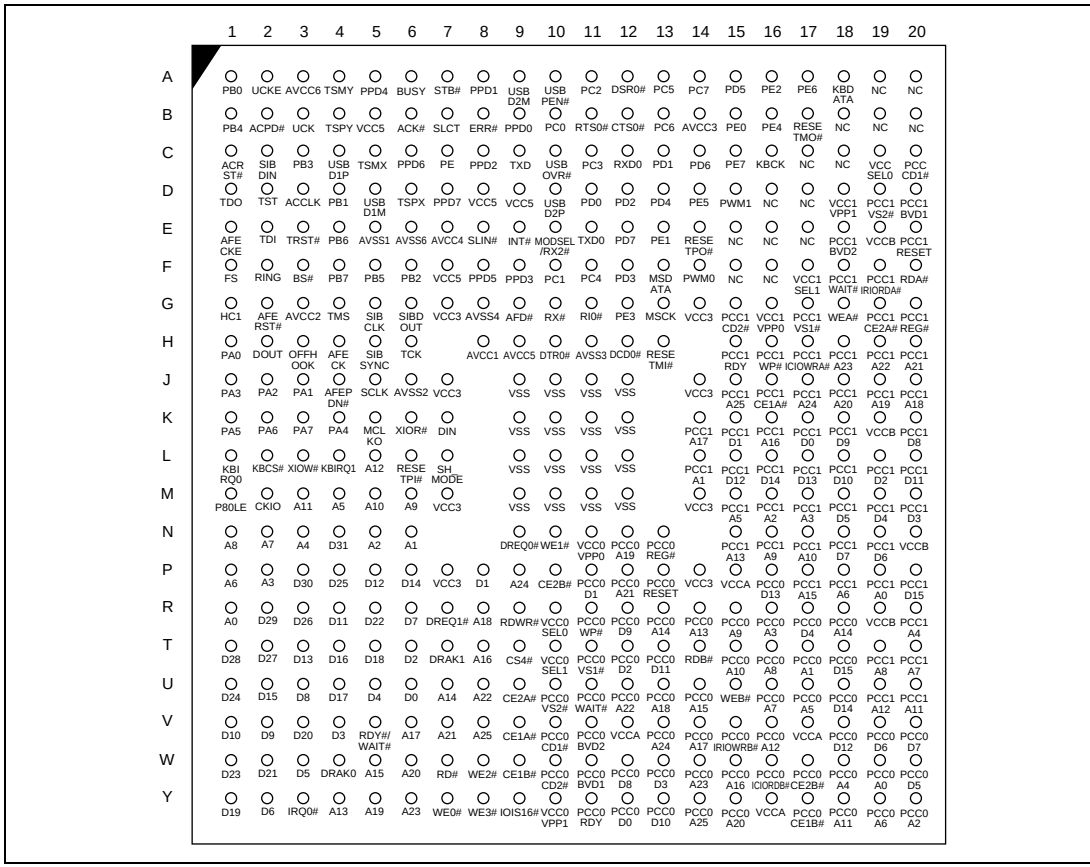
3.5.1 HD64465BP Top View



3.5.2 HD64465BP Bottom View



3.5.3 HD64465BQ Top View



3.5.4 HD64465BQ Bottom View

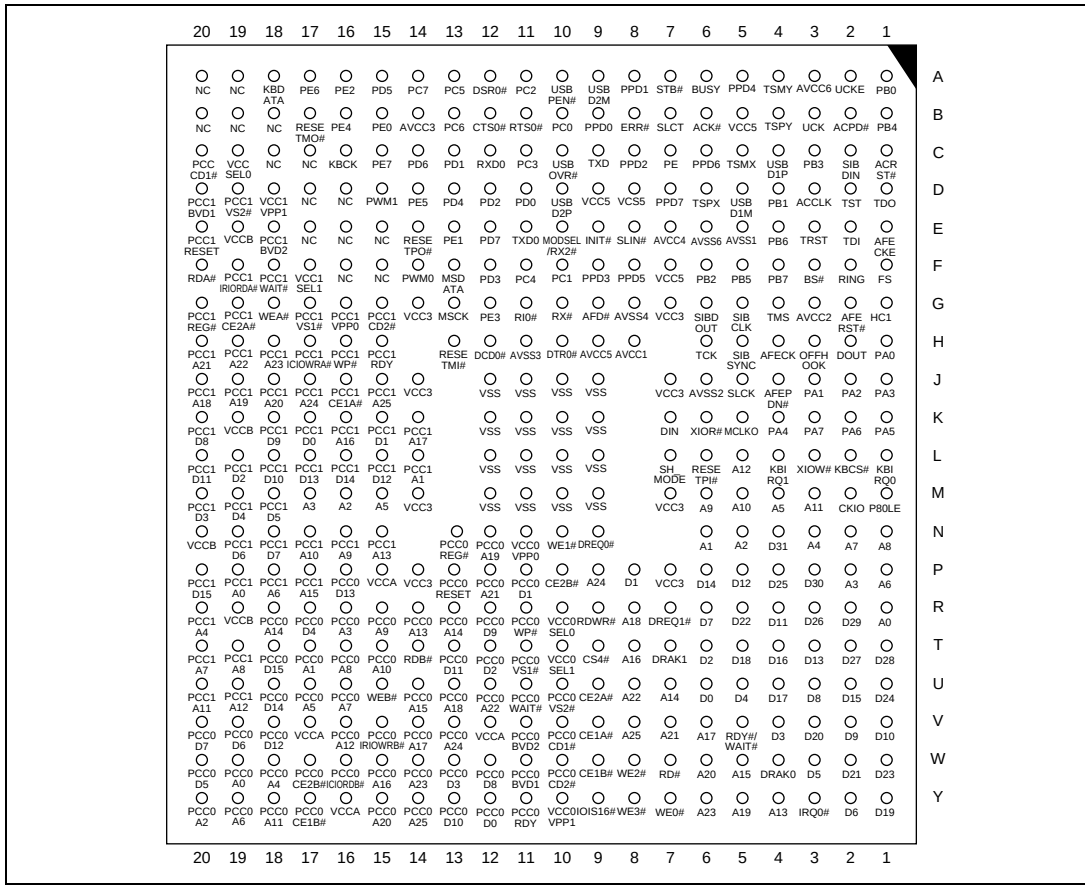


Table 4.1 HD64465BP Signal Names (by pin numbers in alphabetical order)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	AVSS1	B1	PB1/TMO1#	C1	PB4/KBWAKEUP#	D1	ACPD#/ACIRQ/ PWE#
A2	UCKE	B2	PB0/TMO0#	C2	PB3	D2	PB7
A3	UCK	B3	AVCC1	C3	PB2	D3	PB6
A4	AVCC6	B4	USBD1P	C4	USBD1M	D4	PB5/KBRESUME
A5	VCC5	B5	TSPX	C5	TSPY	D5	AVSS6
A6	TSMX	B6	AVCC5	C6	TSMY	D6	AVSS4
A7	VCC5	B7	PPD7	C7	AVCC4	D7	PPD6
A8	ACK#	B8	SLIN#	C8	PPD4	D8	PPD5
A9	VCC5	B9	SLCT	C9	PE	D9	BUSY
A10	INIT#	B10	ERR#	C10	AFD#	D10	STB#
A11	VCC5	B11	PPD1	C11	PPD2	D11	PPD3
A12	MODSEL/RX2#	B12	TXD	C12	RX#	D12	PPD0
A13	PC1	B13	PC0	C13	USBOVR#	D13	USBD2P
A14	PC2	B14	PC3	C14	DTR0#	D14	RTS0#
A15	RI0#	B15	DCD0#	C15	DSR0#	D15	CTS0#
A16	RXD0	B16	PC4	C16	PC5	D16	PC6
A17	AVSS3	B17	PC7	C17	PD0	D17	PD1
A18	AVCC3	B18	PD2	C18	PD3	D18	PD4
A19	PD5	B19	PD6	C19	PD7	D19	PE0
A20	PE1	B20	PE2	C20	PE3	D20	PE4
A21	PE5	B21	PE6	C21	PE7	D21	NC
A22	MSCK	B22	KBDATA	C22	KBCK	D22	PWM1
A23	RESETMO#	B23	RESETPO#	C23	RESETMI#	D23	PWM0
A24	NC	B24	NC	C24	NC	D24	NC
A25	NC	B25	NC	C25	NC	D25	NC
A26	NC	B26	NC	C26	NC	D26	NC

Table 4.1 HD64465BP Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
E1	SIBDOUT	G1	TDI	J1	AVCC2	L1	PA0
E2	SIBCLK	G2	TDO	J2	SCLK	L2	DOUT
E3	SIBDIN	G3	TRST	J3	FS	L3	MCLKO
E4	ACCLK	G4	TCK	J4	RING	L4	OFFHOOK/RLY
E11	VCC	G23	PCC1VS2#	J23	WEA#	L23	PCC1A25
E12	USBD2M	G24	PCC1VS1#	J24	PCC1RESET	L24	PCC1A24
E13	USBPEN#	G25	PCC1BVD2/SPKR1	J25	PCC1ICIOWRA#	L25	PCC1A23
E14	TXD0	G26	PCC1BVD1/STSCHG1#	J26	PCC1ICIORDA#	L26	PCC1A22
E15	VCC						
E23	NC	H1	AVSS2	K1	HC1	M1	PA4
E24	NC	H2	AFECK	K2	AFEPDN#	M2	PA3
E25	VCC1SEL1	H3	AFECKE	K3	AFERST#	M3	PA2
E26	VCC1SEL0	H4	BS#	K4	DIN	M4	PA1
		H23	PCC1RDY/IREQ1#	K23	PCC1CE1A#	M5	VCC
F1	TMS	H24	PCC1WP#/IOIS16#	K24	RDA#	M12	VSS
F2	TST	H25	PCC1WAIT#	K25	PCC1CE2A#	M13	VSS
F3	ACRST#	H26	VCCB	K26	PCC1REG#	M14	VSS
F4	SIBSYNC					M15	VSS
F23	VCC1VPP1					M22	VCC
F24	VCC1VPP0					M23	PCC1A21
F25	PCC1CD2#					M24	PCC1A20
F26	PCC1CD1#					M25	PCC1A19
						M26	PCC1A18

Table 4.1 HD64465BP Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
N1	XIOR#	R1	RESETPI#	U1	A5	W1	D29
N2	XIOW#	R2	A12	U2	A4	W2	D28
N3	PA7	R3	A11	U3	A3	W3	D27
N4	PA6	R4	A10	U4	A2	W4	D26
N5	PA5	R5	SH_MODE	U23	PCC1A2	W23	PCC1A9
N12	VSS	R12	VSS	U24	PCC1A1	W24	PCC1A8
N13	VSS	R13	VSS	U25	PCC1A0	W25	PCC1A7
N14	VSS	R14	VSS	U26	PCC1D7	W26	VCCB
N15	VSS	R15	VSS				
N22	PCC1A17	R22	PCC1D13	V1	A1	Y1	D25
N23	PCC1A16	R23	PCC1D5	V2	A0	Y2	D24
N24	PCC1D8	R24	PCC1D4	V3	D31	Y3	D15
N25	PCC1D0	R25	PCC1D12	V4	D30	Y4	D14
N26	VCCB	R26	PCC1D3	V23	PCC1A6	Y23	PCC1A13
				V24	PCC1A5	Y24	PCC1A12
P1	KBCS#	T1	A9	V25	PCC1A4	Y25	PCC1A11
P2	KBIRQ0	T2	A8	V26	PCC1A3	Y26	PCC1A10
P3	KBIRQ1	T3	A7				
P4	CKIO	T4	A6				
P5	P80LE	T5	VCC				
P12	VSS	T22	VCC				
P13	VSS	T23	PCC1D15				
P14	VSS	T24	PCC1D6				
P15	VSS	T25	PCC1D14				
P22	PCC1D2	T26	VCCB				
P23	PCC1D11						
P24	PCC1D10						
P25	PCC1D1						
P26	PCC1D9						

Table 4.1 HD64465BP Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
AA1	D13	AC1	D21	AD1	D18	AE1	D16
AA2	D12	AC2	D20	AD2	D17	AE2	D6
AA3	D11	AC3	D19	AD3	D4	AE3	D3
AA4	D10	AC4	D1	AD4	D0	AE4	IRQ0#
AA23	PCC0D15	AC5	DRAK1	AD5	DREQ1#	AE5	DRAK0
AA24	PCC0D7	AC6	A13	AD6	A14	AE6	A15
AA25	PCC1A15	AC7	A17	AD7	A18	AE7	A19
AA26	PCC1A14	AC8	A21	AD8	A22	AE8	A23
		AC9	A25	AD9	CS4#	AE9	RDWR#
AB1	D9	AC10	WE0#	AD10	WE1#/WE#	AE10	WE2#/ICIORD#
AB2	D8	AC11	CE2A#	AD11	CE1A#/CS5#	AE11	CE2B#
AB3	D23	AC12	IOIS16#	AD12	VCC0SEL1/ CLOCK	AE12	VCC0SEL0/ DATA
AB4	D22	AC13	PCC0CD2#	AD13	PCC0CD1#	AE13	PCC0VS2#
AB12	VCC	AC14	PCC0WAIT#	AD14	PCC0RDY/ IREQ0#	AE14	PCC0BVD1/ STSCHG0#
AB13	VCC0VPP0	AC15	PCC0D9	AD15	PCC0D1	AE15	PCC0D8
AB14	PCC0WP#/IOIS16B	AC16	PCC0D3	AD16	PCC0D10	AE16	PCC0D2
AB15	VCC	AC17	PCC0D23	AD17	PCC0A24	AE17	PCC0A25
AB23	PCC0D13	AC18	PCC0A19	AD18	PCC0A20	AE18	PCC0A21
AB24	PCC0D5	AC19	PCC0A15	AD19	PCC0A16	AE19	PCC0A17
AB25	PCC0D14	AC20	PCC0RESET	AD20	PCC0A13	AE20	PCC0A14
AB26	PCC0D6	AC21	RDB#	AD21	WEB#	AE21	PCC0ICIOWRB#
		AC22	PCC0A12	AD22	PCC0REG#	AE22	PCC0CE2B#
		AC23	PCC0A0	AD23	PCC0A9	AE23	PCC0A10
		AC24	PCC0D12	AD24	PCC0A3	AE24	PCC0A8
		AC25	PCC0D4	AD25	PCC0A2	AE25	PCC0A5
		AC26	VCCA	AD26	PCC0A1	AE26	PCC0A4

Table 4.1 HD64465BP Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal
AF1	D7	AF11	CE1B#/CS6#	AF21	PCC0ICIORDB#
AF2	D5	AF12	VCC0VPP1/LATCH	AF22	PCC0CE1B#
AF3	D2	AF13	PCC0VS1#	AF23	PCC0A11
AF4	RDY#/WAIT#	AF14	PCC0BVD2/SPKR0	AF24	VCCA
AF5	DREQ0#	AF15	PCC0D0	AF25	PCC0A7
AF6	A16	AF16	VCCA	AF26	PCC0A6
AF7	A20	AF17	PCC0D11		
AF8	A24	AF18	PCC0A22		
AF9	RD#	AF19	PCC0A18		
AF10	WE3#/ICIOWR#	AF20	VCCA		

Table 4.2 HD64465BQ Signal Names (by pin numbers in alphabetical order)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	PB0/TMO0	B1	PB4/KBWAEUP#	C1	ACRST#	D1	TDO
A2	UCKE	B2	ACPD#/ACIRQ	C2	SIBDIN	D2	TST
A3	AVCC6	B3	UCK	C3	PB3	D3	ACCLK
A4	TSMY	B4	TSPY	C4	USBD1P	D4	PB1/TMO1#
A5	PPD4	B5	VCC5	C5	TSMX	D5	USBD1M
A6	BUSY	B6	ACK#	C6	PPD6	D6	TSPX
A7	STB#	B7	SLCT	C7	PE	D7	PPD7
A8	PDD1	B8	ERR#	C8	PPD2	D8	VCC5
A9	USBD2M	B9	PPD0	C9	TXD	D9	VCC5
A10	USBPEN#	B10	PC0	C10	USBOVR#	D10	USBD2P
A11	PC2	B11	RTS0#	C11	PC3	D11	PD0
A12	DSR0#	B12	CTS0#	C12	RXD0	D12	PD2
A13	PC5	B13	PC6	C13	PD1	D13	PD4
A14	PC7	B14	AVCC3	C14	PD6	D14	PE5
A15	PD5	B15	PE0	C15	PE7	D15	PWM1
A16	PE2	B16	PE4	C16	KBCK	D16	NC
A17	PE6	B17	RESETMO#	C17	NC	D17	NC
A18	KBDATA	B18	NC	C18	NC	D18	VCC1VPP1
A19	NC	B19	NC	C19	VCC1SEL0	D19	PCC1VS2#
A20	NC	B20	NC	C20	PCC1CD1#	D20	PCC1BVD1/STSCHG1#

Table 4.2 HD64465BQ Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
E1	AFECKE	F1	FS	G1	HC1	H1	PA0
E2	TDI	F2	RING	G2	AFERST#	H2	DOUT
E3	TRST#	F3	BS#	G3	AVCC2	H3	OFFHOOK/RLY
E4	PB6	F4	PB7	G4	TMS	H4	AFECK
E5	AVSS1	F5	PB5/KBRESUME	G5	SIBCLK	H5	SIBSYNC
E6	AVSS6	F6	PB2	G6	SIBDOUT	H6	TCK
E7	AVCC4	F7	VCC5	G7	VCC3	H8	AVCC1
E8	SLIN#	F8	PPD5	G8	AVSS4	H9	AVCC5
E9	INIT#	F9	PPD3	G9	AFD#	H10	DTR0#
E10	MODSEL/RX2#	F10	PC1	G10	RX#	H11	AVSS3
E11	TXD0	F11	PC4	G11	RI0#	H12	DCD0#
E12	PD7	F12	PD3	G12	PE3	H13	RESETMI#
E13	PE1	F13	MSDATA	G13	MSCK	H15	PCC1RDY/IREQ1#
E14	RESETPO#	F14	PWM0	G14	VCC3	H16	PCC1WP#/IOIS16#
E15	NC	F15	NC	G15	PCC1CD2#	H17	PCC1ICIOWRA#
E16	NC	F16	NC	G16	VCC1VPP0	H18	PCC1A23
E17	NC	F17	VCC1SEL1	G17	PCC1VS1#	H19	PCC1A22
E18	PCC1BVD2/SPKR1	F18	PCC1WAIT#	G18	WEA#	H20	PCC1A21
E19	VCCB	F19	PCC1ICIORDA#	G19	PCC1CE2A#		
E20	PCC1RESET	F20	RDA#	G20	PCC1REG#		

Table 4.2 HD64465BQ Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
J1	PA3	K1	PA5	L1	KBIRQ0	M1	P80LE
J2	PA2	K2	PA6	L2	KBCS#	M2	CKIO
J3	PA1	K3	PA7	L3	XIOW#	M3	A11
J4	AFEPDN#	K4	PA4	L4	KBIRQ1	M4	A5
J5	SCLK	K5	MCLKO	L5	A12	M5	A10
J6	AVSS2	K6	XIOR#	L6	RESETPI#	M6	A9
J7	VCC3	K7	DIN	L7	SH_MODE	M7	VCC3
J9	VSS	K9	VSS	L9	VSS	M9	VSS
J10	VSS	K10	VSS	L10	VSS	M10	VSS
J11	VSS	K11	VSS	L11	VSS	M11	VSS
J12	VSS	K12	VSS	L12	VSS	M12	VSS
J14	VCC3	K14	PCC1A17	L14	PCC1A1	M14	VCC3
J15	PCC1A25	K15	PCC1D1	L15	PCC1D12	M15	PCC1A5
J16	PCC1CE1A#	K16	PCC1A16	L16	PCC1D14	M16	PCC1A2
J17	PCC1A24	K17	PCC1D0	L17	PCC1D13	M17	PCC1A3
J18	PCC1A20	K18	PCC1D9	L18	PCC1D10	M18	PCC1D5
J19	PCC1A19	K19	VCCB	L19	PCC1D2	M19	PCC1D4
J20	PCC1A18	K20	PCC1D8	L20	PCC1D11	M20	PCC1D3

Table 4.2 HD64465BQ Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
N1	A8	P1	A6	R1	A0	T1	D28
N2	A7	P2	A3	R2	D29	T2	D27
N3	A4	P3	D30	R3	D26	T3	D13
N4	D31	P4	D25	R4	D11	T4	D16
N5	A2	P5	D12	R5	D22	T5	D18
N6	A1	P6	D14	R6	D7	T6	D2
N9	DREQ0#	P7	VCC3	R7	DREQ1#	T7	DRAK1
N10	WE1#/WE#	P8	D1	R8	A18	T8	A16
N11	VCC0VPP0	P9	A24	R9	RDWR#	T9	CS4#
N12	PCC0A19	P10	CE2B#	R10	VCC0SEL0/ DATA	T10	VCC0SEL1/ CLOCK
N13	PCC0REG#	P11	PCC0D1	R11	PCC0WP#/ IOIS16	T11	PCC0VS1#
N15	PCC1A13	P12	PCC0A21	R12	PCC0D9	T12	PCC0D2
N16	PCC1A9	P13	PCC0RESET	R13	PCC0A14	T13	PCC0D11
N17	PCC1A10	P14	VCC3	R14	PCC0A13	T14	RDB#
N18	PCC1D7	P15	VCCA	R15	PCC0A9	T15	PCC0A10
N19	PCC1D6	P16	PCC0D13	R16	PCC0A3	T16	PCC0A8
N20	VCCB	P17	PCC1A15	R17	PCC0D4	T17	PCC0A1
		P18	PCC1A6	R18	PCC1A14	T18	PCC0D15
		P19	PCC1A0	R19	VCCB	T19	PCC1A8
		P20	PCC1D15	R20	PCC1A4	T20	PCC1A7

Table 4.2 HD64465BQ Signal Names (by pin numbers in alphabetical order) [cont'd]

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
U1	D24	V1	D10	W1	D23	Y1	D19
U2	D15	V2	D9	W2	D21	Y2	D6
U3	D8	V3	D20	W3	D5	Y3	IRQ0#
U4	D17	V4	D3	W4	DRAK0	Y4	A13
U5	D4	V5	RDY#/WAIT#	W5	A15	Y5	A19
U6	D0	V6	A17	W6	A20	Y6	A23
U7	A14	V7	A21	W7	RD#	Y7	WE0
U8	A22	V8	A25	W8	WE2#/ICIORD#	Y8	WE3#/ICIOWR#
U9	CE2A#	V9	CE1A#/CS5#	W9	CE1B#/CS6#	Y9	IOIS16#
U10	PCC0VS2#	V10	PCC0CD1#	W10	PCC0CD2#	Y10	VCC0VPP1/ LATCH
U11	PCC0WAIT#	V11	PCC0BVD2/SPKR0	W11	PCC0BVD1/ STSCHG0#	Y11	PCC0RDY/ IREQ0#
U12	PCC0A22	V12	VCCA	W12	PCC0D8	Y12	PCC0D0
U13	PCC0A18	V13	PCC0A24	W13	PCC0D3	Y13	PCC0D10
U14	PCC0A15	V14	PCC0A17	W14	PCC0A23	Y14	PCC0A25
U15	WEB#	V15	PCC0ICIOWRB	W15	PCC0A16	Y15	PCC0A20
U16	PCC0A7	V16	PCC0A12	W16	PCC0ICIORDB#	Y16	VCCA
U17	PCC0A5	V17	VCCA	W17	PCC0CE2B#	Y17	PCC0CE1B#
U18	PCC0D14	V18	PCC0D12	W18	PCC0A4	Y18	PCC0A11
U19	PCC1A12	V19	PCC0D6	W19	PCCA0	Y19	PCC0A6
U20	PCC1A11	V20	PCC0D7	W20	PCC0D5	Y20	PCC0A2

Table 4.3 Pin Descriptions of Test Mode Select

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
Test Mode				
F2	D2	TST	I	Test Mode Enable 0 = Disabled 1 = Enabled
G1	E2	TDI	I	Boundary Scan Data Input This pin can be floating when not using.
G2	D1	TDO	O	Boundary Scan Data Output
F1	G4	TMS	I	Boundary Scan Mode Select. This pin can be floating when not using.
G4	H6	TCK	I	Boundary Scan Clock. This pin can be floating when not using.
G3	E3	TRST	I	Boundary Scan Reset

Table 4.4 Pin Descriptions of CPU Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
CPU Interface				
P4	M2	CKIO	I	System clock
AC9, AF8, AE8, AD8, AC8, AF7, AE7, AD7, AC7,AF6, AE6, AD6, AC6, R2-4,T1-4, U1-4,V1-2	V8, P9, Y6, U8, V7, W6, Y5, R8, V6, T8, W5, U7, Y4, L5, M3, M5, M6, N1, N2, P1, M4, N3, P2, N5, N6, R1	A25-A0	I	Address bus of CPU
V3-4, W1-4, Y1-2, AB3-4, AC1-3, AD1-2, AE1, Y3-4, AA1-4, AB1-2, AF1, AE2, AF2, AD3, AE3, AF3, AC4, AD4	N4, P3, R2, T1, T2, R3, P4, U1, W1, R5, W2, V3, Y1, T5, U4, T4, U2, P6, T3, P5, R4, V1, V2, U3, R6, Y2, W3, U5, V4, T6, P8, U6	D31-D0	IO	Data bus of CPU

Table 4.4 Pin Descriptions of CPU Interface (cont'd)

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
AD9	T9	CS4#	I	Chip select 4 of CPU
AC10	Y7	WE0#	I	D7-D0 write strobe signal
AD10	N10	WE1#/WE#	I	D15-D8 write strobe signal, or PCMCIA write strobe signal
AE10	W8	WE2#/ICIORD#	I	D23-D16 write strobe signal, or PCMCIA I/O READ
AF10	Y8	WE3#/ICIOWR#	I	D31-D24 write strobe signal, or PCMCIA I/O WRITE
AE9	R9	RDWR#	I	Data bus direction indicator signal
AF9	W7	RD#	I	Strobe signal indicating the READ cycle
AF4	V5	RDY#/WAIT#	O	RDY# signal for SH4 / WAIT# signal for SH7709
H4	F3	BS#	I	Bus start of CPU
R1	L6	RESETPI#	I	RESET request
AF5	N9	DREQ0#	O	DMA request is generated by FIR
AD5	R7	DREQ1#	O	DMA request is generated by Timer or Codec interface module
AE5	W4	DRAK0	I	DMA request acknowledge for DREQ0#
AC5	T7	DRAK1	I	DMA request acknowledge for DREQ1#
AE4	Y3	IRQ0#	O	Interrupt request to CPU.
R5	L7	SH_MODE	I	SH7709/SH4 CPU interface selection. 0 = SH7709 1 = SH4
AF11	W9	CE1B#/CS6#	I	Chip enable 1 for PCMCIA card 0
AE11	P10	CE2B#	I	Chip enable 2 for PCMCIA card 0
AD11	V9	CE1A#/CS5#	I	Chip enable 1 for PCMCIA card 1
AC11	U9	CE2A#	I	Chip enable 2 for PCMCIA card 1
AC12	Y9	IOIS16#	O	Write protect I/O is 16 bits for PCMCIA card

Table 4.5 Pin Descriptions of PCMCIA 0 Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
PCMCIA 0 (memory and IO)				
AE17, AD17, AC17, AF18, AE18, AD18, AC18, AF19, AE19, AD19, AC19, AE20, AD20, AC22, AF23, AE23, AD23, AE24, AF25, AF26, AE25, AE26, AD24-26, AC23	Y14, V13, W14, U12, P12, Y15, N12, U13, V14, W15, U14, R13, R14, V16, Y13, T15, R15, T16, U16, Y19, U17, W18, R16, Y20, T17, W19	PCC0A25-A0	O	Address bus [25:0] of PCMCIA card 0
AA23, AB25, AB23, AC24, AF17, AD16, AC15, AE15, AA24, AB26, AB24, AC25, AC16, AE16, AD15, AF15	T18, U18, P16, V18, T13, Y13, R12, W12, V20, V19, W20, R17, W13, T12, P11, Y12	PCC0D15-D0	IO	Data bus [15:0] of PCMCIA card 0
AF22	Y17	PCC0CE1B#	O	PCMCIA card 0 low byte enable
AE22	W17	PCC0CE2B#	O	PCMCIA card 0 high byte enable
AC21	T14	RDB#	O	PCMCIA card 0 Read enable
AD21	U15	WEB#	O	PCMCIA card 0 Write enable
AF21	W16	PCC0ICIORDB#	O	PCMCIA card 0 I/O Read enable
AE21	V15	PCC0ICIOWRB#	O	PCMCIA card 0 I/O Write enable
AC20	P13	PCC0RESET	O	PCMCIA card 0 reset
AC14	U11	PCC0WAIT#	I	PCMCIA card 0 memory or I/O wait state

Table 4.5 Pin Descriptions of PCMCIA 0 Interface (cont'd)

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
AB14	R11	PCC0WP#/ IOIS16#	I	Reflects the states of the Write Protect switch on PCCMCIA memory cards. For I/O cards, PCC0WP# is used for the card, which is 16-bit Port (IOIS16#) function.
AD14	Y11	PCC0RDY/ IREQ0#	I	Driven low by memory PC cards to indicate that the memory card circuits are busy. For I/O card, PCC0RDY is used as an interrupt request.
AE14	W11	PCC0BVD1/ STSCHG0#	I	The signal is an indication of the battery condition on the PCC0 memory card. Both PCC0BVD1 and PCC0BVD2 are in high level when the battery is in good condition. When PCC0BVD1 is low, the PC card battery is no longer serviceable and data are lost. For I/O card, PCC0BVD1 is card status change (STSCHG) function.
AF14	V11	PCC0BVD2/ SPKR0	I	The signal is an indication of the battery condition on the PCC0 memory card. Both PCC0BVD1 and PCC0BVD2 are high level when the battery is in good condition. When PCC0BVD2 is low while PCC0BVD1 is in high level, the PC card battery is in a warning state. For I/O card, PCC0BVD2 acts as SPKR function.
AD13	V10	PCC0CD1#	I	Provides for PCMCIA card 0 insertion detection
AC13	W10	PCC0CD2#	I	Provides for PCMCIA card 0 insertion detection
AF13	T11	PCC0VS1#	I	PCMCIA card 0 Voltage sense
AE13	U10	PCC0VS2#	I	PCMCIA card 0 Voltage sense
AD22	N13	PCC0REG#	O	PCMCIA card 0 attribute memory select
AD12	T10	VCC0SEL1/ CLOCK	O	PCMCIA card 0 VCC power control / Clock for serial data word of TPS2206
AE12	R10	VCC0SEL0/ DATA	O	PCMCIA card 0 VCC power control / Serial data word of TPS2206
AF12	Y10	VCC0VPP1/ LATCH	O	PCMCIA card 0 VPP power control / Latch for serial data word of TPS2206
AB13	N11	VCC0VPP0	O	PCMCIA card 0 VPP power control

Table 4.6 Pin Descriptions of PCMCIA 1 Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
PCMCIA 1 (memory or IO)				
L23-26, M23-26, N22-23, AA25-26, Y23-26, W23-25, V23-26, U23-25	J15, J17, H18-20, J18-20, K14, K16, P17, R18, N15, U19, U20, N17, N16, T19, T20, P18, M15, R20, M17, M16, L14, P19	PCC1A25-A0	O	Address bus [25:0] of PCMCIA card 1
T23, T25, R22, R25, P23-24, P26, N24, U26, T24, R23-24, R26, P22, P25, N25	P20, L16, L17, L15, L20, L18, K18, K20, N18, N19, M18-20, L19, K15, K17	PCC1D15-D0	IO	Data bus [15:0] of PCMCIA card 1
K23	J16	PCC1CE1A#	O	PCMCIA card 1 low byte enable
K25	G19	PCC1CE2A#	O	PCMCIA card 1 high byte enable
K24	F20	RDA#	O	PCMCIA card 1 Read enable
J23	G18	WEA#	O	PCMCIA card 1 Write enable
J26	F19	PCC1ICIORDA#	O	PCMCIA card 1 I/O Read enable
J25	H17	PCC1ICIOWRA#	O	PCMCIA card 1 I/O Write enable
J24	E20	PCC1RESET	O	PCMCIA card 1 reset
H25	F18	PCC1WAIT#	I	PCMCIA card 1 memory or IO wait state
H24	H16	PCC1WP#/ IOIS16#	I	Reflects the states of the Write Protect switch on PCC1 memory cards. For I/O cards, PCC1WP# is used for the card, which is 16-bit Port (IOIS16#) function.
H23	H15	PCC1RDY/ IREQ1#	I	Driven low by memory PC cards to indicate that the memory card circuits are busy. For I/O card, PCC1RDY is used as an interrupt request.

Table 4.6 Pin Descriptions of PCMCIA 1 Interface (cont'd)

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
G26	D20	PCC1BVD1/ STSCHG1#	I	The signal indicates the battery condition on the PCC1 memory card. Both PCC1BVD1 and PCC1BVD2 are in high level when the battery is in good condition. When PCC1BVD1 is low, the PC card battery is no longer serviceable and data are lost. For I/O card, PCC1BVD1 is used as card status change (STSCHG) function.
G25	E18	PCC1BVD2/ SPKR1	I	The signal indicates the battery condition on the PCC1 memory card. Both PCC1BVD1 and PCC1BVD2 are in high level when the battery is in good condition. When PCC1BVD2 is low while PCC1BVD1 is in high level, the PC card battery is in a warning state. For I/O card, PCC1BVD2 is used SPKR function.
F26	C20	PCC1CD1#	I	Provided for PCMCIA card 1 insertion detection
F25	G15	PCC1CD2#	I	Provided for PCMCIA card 1 insertion detection
G24	G17	PCC1VS1#	I	PCMCIA card 1 Voltage sense
G23	D19	PCC1VS2#	I	PCMCIA card 1 Voltage sense
K26	G20	PCC1REG#	O	PCMCIA card 1 attribute memory select
E25	F17	VCC1SEL1	O	PCMCIA card 1 VCC power control
E26	C19	VCC1SEL0	O	PCMCIA card 1 VCC power control
F23	D18	VCC1VPP1	O	PCMCIA card 1 VPP power control
F24	G16	VCC1VPP0	O	PCMCIA card 1 VPP power control

Table 4.7 Pin Descriptions of UART 0

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
UART 0				
E14	E11	TXD0	O	Data output for UART0
A16	C12	RXD0	I	Data input for UART0
D14	B11	RTS0#	O	Request to Send Output for UART0
D15	B12	CTS0#	I	Clear to Send Input for UART0
C14	H10	DTR0#	O	Data Terminal Ready Output for UART0
C15	A12	DSR0#	I	Data Set Ready for UART0
B15	H12	DCD0#	I	Receive Line Signal Detect for UART0
A15	G11	RI0#	I	Ring Indicator for UART0

Table 4.8 Pin Description of IrDA

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
IrDA				
A12	E10	MODSEL/RX2#	O/I	Multifunction pin: For IrDA, Low or high frequency infrared select with IBM transceiver module / High frequency infrared data stream input with HP transceiver module. @
B12	C9	TXD	O	Infrared data stream output for IrDA.
C12	G10	RX#	I	For IrDA, infrared data stream input, is connected to infrared receive data stream output if IBM transceiver module is used, or is connected to low frequency infrared receive data stream output if HP transceiver module

Table 4.9 Pin Descriptions of Printer Port Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
Printer Interface				
D10	A7	STB#	O	Printer Strobe. Active low, this signal is the complement of bit 0 of the printer control register. It is used to strobe the printer data into the printer
C10	G9	AFD#	O	Printer Autofeed. Active low, this signal is the complement of bit 1 of the printer control register. This signal, when in low level, is used to feed one line after each line is printed when it is low.
B10	B8	ERR#	I	Printer Error. Active low, indicates printer has encountered an “ERROR” condition. It can be read at bit 3 of printer status register
A10	E9	INIT#	O	Printer Initialize. Active low, this signal is bit 2 of the printer control register, and is used to initiate printer when it is low.
B8	E8	SLIN#	O	Printer Select. Select the printer when it is low, this signal is the complement of bit 3 of the printer control register
A8	B6	ACK#	I	Printer Acknowledge. This signal goes low to indicate that the printer has already received a character and is ready to accept another
D9	A6	BUSY	I	Printer Busy. This signal goes high when the line printer has a local operation in process and cannot accept data
C9	C7	PE	I	Printer Paper End. This signal is set high by printer when it runs out of paper
B9	B7	SLCT	I	Printer Select. This signal goes high when the line printer has been selected
B7, D7-8, C8, D11, C11, B11, D12	D7, C6, F8, A5, F9, C8, A8, B9	PPD7-PPD0	IO	Parallel Port Data Bus. This bus provides a byte-wide input or output to the system. The eight lines are held in a high-impedance state when the port is not selected.

Table 4.10 Pin Descriptions of AFE Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
AFE Interface				
L2	H2	DOUT	O	Serial Transmit Data Output Pin (TxD). This signal is connected to AFE module DI.
K4	K7	DIN	I	Serial Receive Data Input Pin (RxD). This signal comes from AFE module DOUT.
J2	J5	SCLK	I	Shift Clock Input Pin. This signal comes from AFE module SCLK.
K1	G1	HC1	O	Hardware Control Signal 1 for STLC7546/STLC7550
J3	F1	FS	I	Frame Sync Signal Input Pin. This signal comes form AFE module FS.
K3	G2	AFERST#	O	This signal outputs to reset AFE module.
K2	J4	AFEPDN#	O	This signal outputs to power down AFE module.
L3	K5	MCLKO	O	Master Clock to AFE module
L4	H3	OFFHOOK/ RLY	O	This signal is used as the active control for the OFFHOOK relay / RLY control and dial pulse output
J4	F2	RING	I	Ringing Signal Input Pin

Table 4.11 Pin Descriptions of CODEC Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
CODEC Interface				
E4	D3	ACCLK	O	Audio Codec clock
F3	C1	ACRST#	O	Audio Codec reset. The pin can also be used to act as PLL3 test output pin at PLL test mode
D1	B2	ACPD#/ACIRQ /PWE#	IO	When the connected CODEC is CS4271, the pin is input and is used to interrupt the interface. But when the connected CODEC is CS4218, the pin is output and is used to power down CS4218.
E3	C2	SIBDIN	I	Serial Interface Input Data
E2	G5	SIBCLK	IO	Serial Interface Clock
E1	G6	SIBDOUT	O	Serial Interface Output Data
F4	H5	SIBSYNC	IO	Serial Interface Sync

Table 4.12 Pin Descriptions of USB Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
USB interface				
E13	A10	USBPEN#	O	USB Power Enable Control Signal
C13	C10	USBOVR#	I	USB Over-Current Detect
B4	C4	USBD1P	IO	USB Port 1 Data D+
C4	D5	USBD1M	IO	USB Port 1 Data D-
D13	D10	USBD2P	IO	USB Port 2 Data D+
E12	A9	USBD2M	IO	USB Port 2 Data D-

Table 4.13 Pin Descriptions of Keyboard Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
Keyboard Interface				
P1	L2	KBCS#	O	Keyboard Controller Chip Select
N2	L3	XIOW#	O	Keyboard Controller Write Enable
N1	K6	XIOR#	O	Keyboard Controller Read Enable
P2	L1	KBIRQ0	I	Keyboard Controller Interrupt 0
P3	L4	KBIRQ1	I	Keyboard Controller Interrupt 1

Table 4.14 Pin Descriptions of IO Port A

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
IO Port A (multifunction with AFE)				
N3	K3	PA7	IO	Port A bit 7 for GPIO
N4	K2	PA6	IO	Port A bit 6 for GPIO
N5	K1	PA5	IO	Port A bit 5 for GPIO
M1	K4	PA4	IO	Port A bit 4 for GPIO
M2	J1	PA3	IO	Port A bit 3 for GPIO
M3	J2	PA2	IO	Port A bit 2 for GPIO
M4	J3	PA1	IO	Port A bit 1 for GPIO
L1	H1	PA0	IO	Port A bit 0 for GPIO

Table 4.15 Pin Description of IO Port B

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
IO Port B (multifunction with TIMER or AFE)				
D2	F4	PB7	IO	Port B bit 7 for GPIO
D3	E4	PB6	IO	Port B bit 6 for GPIO
D4	F5	PB5/ KBRESUME	IO/I	Multifunction Pin: Port B bit 5 for GPIO/The signal is used to resume standby keyboard controller
C1	B1	PB4/ KBWAKEUP#	IO/O	Multifunction Pin: Port B bit 4 for GPIO/Keyboard controller wakes up the STANDBY system via signal KBWAKEUP#.
C2	C3	PB3	IO	Port B bit 3 for GPIO
C3	F6	PB2	IO	Port B bit 2 for GPIO
B1	D4	PB1/TMO1#	IO/O	Multifunction Pin: Port B bit 1 for GPIO / Timer 1 output signal is used to trigger external event
B2	A1	PB0/TMO0#	IO/O	Multifunction Pin: Port B bit 0 for GPIO / Timer 0 output signal is used to trigger external event

Table 4.16 Pin Descriptions of IO Port C

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
IO Port C				
B17	A14	PC7	IO	Port C bit 7 for GPIO
D16	B13	PC6	IO	Port C bit 6 for GPIO
C16	A13	PC5	IO	Port C bit 5 for GPIO
B16	F11	PC4	IO	Port C bit 4 for GPIO
B14	C11	PC3	IO	Port C bit 3 for GPIO
A14	A11	PC2	IO	Port C bit 2 for GPIO
A13	F10	PC1	IO	Port C bit 1 for GPIO
B13	B10	PC0	IO	Port C bit 0 for GPIO

Table 4.17 Pin Descriptions of IO Port D

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
IO Port D				
C19	E12	PD7	IO	Port D bit 7 for GPIO
B19	C14	PD6	IO	Port D bit 6 for GPIO
A19	A15	PD5	IO	Port D bit 5 for GPIO
D18	D13	PD4	IO	Port D bit 4 for GPIO
C18	F12	PD3	IO	Port D bit 3 for GPIO
B18	D12	PD2	IO	Port D bit 2 for GPIO
D17	C13	PD1	IO	Port D bit 1 for GPIO
C17	D11	PD0	IO	Port D bit 0 for GPIO

Table 4.18 Pin Descriptions of IO Port E

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
IO Port E				
C21	C15	PE7	IO	Port E bit 7 for GPIO
B21	A17	PE6	IO	Port E bit 6 for GPIO
A21	D14	PE5	IO	Port E bit 5 for GPIO
D20	B16	PE4	IO	Port E bit 4 for GPIO
C20	G12	PE3	IO	Port E bit 3 for GPIO
B20	A16	PE2	IO	Port E bit 2 for GPIO
A20	E13	PE1	IO	Port E bit 1 for GPIO
D19	B15	PE0	IO	Port E bit 0 for GPIO

Table 4.19 Pin Descriptions of 10-bit ADC Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
10-bit ADC Interface				
A6	C5	TSMX	I	Touch screen minus X - plate input
C6	A4	TSMY	I	Touch screen minus Y - plate input
B5	D6	TSPX	I	Touch screen plus X - plate input
C5	B4	TSPY	I	Touch screen plus Y - plate input

Table 4.20 Pin Descriptions of PS/2 Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
PS/2 Interface				
C22	C16	KBCK	IO	PS/2 keyboard clock input/output
B22	A18	KBDATA	IO	PS/2 keyboard data input/output
A22	G13	MSCK	IO	PS/2 mouse clock input/output
D21	F13	MSDATA	IO	PS/2 mouse data input/output

Table 4.21 Pin Descriptions of System Reset Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
System Reset Interface				
R1	L6	RESETPI#	I	System power-on reset input
C23	H13	RESETMI#	I	System manual reset input
B23	E14	RESETPO#	O	RESET# signal for SH-4/SH7709
A23	B17	RESETMO#	O	MRESET# signal for SH-4/RESETM signal for SH7709

Table 4.22 Pin Descriptions of Crystal Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
Crystal Interface				
H2	H4	AFECK	I	AFE Clock Input (12.288MHz)
H3	E1	AFECKE	IO	AFE Clock Output (12.288MHz)
A3	B3	UCK	I	USB Clock Input (12MHz)
A2	A2	UCKE	IO	USB Clock Output (12MHz)

Table 4.23 Pin Description of Miscellaneous Interface

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
Miscellaneous Interface				
D23	F14	PWM0	O	PWM channel 0 timer pulse output pins
D22	D15	PWM1	O	PWM channel 1 timer pulse output pins
P5	M1	P80LE	O	Port80 strobe for diagnostic

Table 4.24 Pin Descriptions of No Connected Pins

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
No Connection				
A24-A26, B24-B26, C24-C26, D24-D26, E23, E24	A20, A19, B20-18, C18, C17, D17, D16, E17-15, F16, F15	NC	I	No Connected pins (total 23 pins).

Table 4.25 Pin Descriptions of Power/Ground

Pin No. (HD64465BP)	Pin No. (HD64465BQ)	Symbol	I/O	Description
M12-15, N12-15, P12-15, R12-15	J9-12, K9-12, L9-12, M9-12	VSS	I	Ground
E11, E15, M5, M22, T5, T22, AB12, AB15	G7, G14, J7, J14, M7, M14, P7, P14	VCC (VCC3)*	I	3.3 Volt power
A5, A7, A9, A11	B5, D8, D9, F7	VCC5	I	5.0 Volt power for printer port/PS/2 port
AC26, AF16, AF20, AF24	P15, V12, V17, Y16	VCCA	I	0 / 3.3 / 5V power for address/data buffer of PCMCIA 0
H26, N26, T26, W26	E19, K19, N20, R19	VCCB	I	0 / 3.3 / 5V power for address/data buffer of PCMCIA 1
B3	H8	AVCC1	I	3.3 Volt power for analog PLL1
A1	E5	AVSS1	I	Ground for analog PLL1
J1	G3	AVCC2	I	3.3 Volt power for analog PLL2
H1	J6	AVSS2	I	Ground for analog PLL2
A18	B14	AVCC3	I	3.3 Volt power for analog circuit
A17	H11	AVSS3	I	NC (No Connected Pin)
C7	E7	AVCC4	I	3.3 Volt for analog portion of 10-bit ADC
D6	G8	AVSS4	I	Ground for analog portion of 10-bit ADC
B6	H9	AVCC5	I	3.3 Volt for analog portion of 10-bit ADC
A4	A3	AVCC6	I	3.3 Volt for analog USB Tranceiver
D5	E6	AVSS6	I	Ground for analog USB Tranceiver

Note: * HD64465BP = VCC
HD64465BQ = VCC3

Section 5 Internal CPU Interface

5.1 Introduction

CPU INTERFACE Module builds an internal peripheral bus interface on HD64465. This interface provides a bridge between Hitachi SH-3/SH-4 CPU and all peripheral modules in HD64465. This section will explain the functionality and timing of all signals defined in CPU interface module.

5.2 CPU Interface Signal Description

5.2.1 System Bus Interface Signals

In system bus interface, the configuration of **BSC** in SH-4/SH-3 needs to be programmed properly. The followings list the bus configuration requirements.

1. Area: Area 4
2. Bus width: Longword (32 bits) size and little endian access.
3. Idle state: No idle cycles
4. Wait state: 3 wait states

Signal Name	I/O Type	Description
CKIO	I	**CKIO:** SH3/SH4 system clock IO. This clock is used as the master clock for the internal logic of the CPU Interface
RESET#	I	**RESET#:** System reset signal.
CS4#	I	**CS4#:** System Chip select 4 signal.
RDY/WAIT#	O	**System Wait:** This signal is controlled by the CPU Interface to insert the wait state in CPU cycle. The inserted wait cycles are depended on peripheral *WAIT#.
ADDR[20:1]	I	**Address Bus [20:1]:** These are the address input signals to the CPU Interface. The CPU Interface address decoder uses these signals to decode the module select(*MS#) signal of all peripheral modules in IPC.
ADDR24	I	**Address bus 24:** This signal is address bus number 24 driven by the CPU.
ADDR25	I	**Address bus 25:** This signal is address bus number 25 driven by the CPU.
IDATA[31:0]	I	**Input Data Bus [31:0]:** These are the bit[31:0] of data bus driven by SH3/SH4.(write data).
ODATA[31:0]	O	**Output Data Bus [31:0]:** These are the bit[31:0] of data bus to be read by the CPU.(read data).
RDWR#	I	**Read/Write Command:** System read/write indicator driven by the CPU.
RD#	I	**Read Command:** When active along with CS4#, a valid data will be put onto the ODATA[15:0] for the Host CPU to read. This signal is driven by the CPU.
WE0#	I	**Write byte 0 Command:** When active along with CS4#, a valid data will be passed from the Host CPU to MIDATA[7:0]. This signal is driven by the CPU.
WE1#	I	**Write byte 1 Command:** When active along with CS4#, a valid data will be passed from the Host CPU to MIDATA[15:8]. This signal is driven by the CPU.
WE2#	I	**Write byte 2 Command:** When active along with CS4#, a valid data will be passed from the Host CPU to MIDATA[23:16]. This signal is driven by the CPU.
WE3#	I	**Write byte 3 Command:** When active along with CS4#, a valid data will be passed from the Host CPU to MIDATA[31:24]. This signal is driven by the CPU.

Notes: 1. * stands for peripheral module name.
2. # means that a signal is active low.

5.2.2 Internal Bus Interface Signals

Signal Name	I/O Type	Description
*RESET#	O	**Internal Module Reset:** This signal is module reset of *module. (For example, LCDC reset signal is LCDC_RESET#)
*STBY	O	**Module Standby:** This signal is used to control the standby mode for each peripheral module. (For LCDC as an example, when LCDC_STBY is asserted, LCDC module will go into the standby mode)
*WAIT#	I	**Module Wait:** This signal is controlled by a peripheral module to insert a wait state in CPU command cycle. (For LCDC as an example, when LCDC_WAIT# is active low, the CPU Interface will insert a wait signal to the CPU cycle until the LCDC_WAIT# is high)
*MS#	O	**Module Select:** This module select signal is decoded based on the IPC module address map. (For LCDC as an example, the LCDC control register address and LCDC Frame buffer address will activate LCDC_MS#)
IMADDR[20:1]	O	**Internal Module Address Bus [20:1]:** These are the address input signals to all the peripheral modules. The peripheral module address decoder can use these signals to decode the internal registers.
IMADDR24	O	**Internal Module Address bus 24:** This signal is address bus number 24 driven by CPU Interface. This signal is connected to all peripheral modules.
IMADDR25	O	**Internal Module Address bus 25:** This signal is address bus number 25 driven by CPU Interface. This signal is connected to all peripheral modules.
MIDATA[31:0]	O	**Module Input Data Bus [31:0]:** These are the bit[31:0] of data bus to be driven to all the peripheral modules. (Write data from the CPU)
MODATA[31:0]	I	**Module Output Data Bus [31:0]:** These are the bit[31:0] of data bus to be read by the CPU. (Read data from the peripheral module)
IMRDWR#	O	**Internal Module Read/Write Command:** Internal Module read/write indicator driven by CPU Interface.
IMRD#	O	**Internal Module Read Command:** When active along with *MS#, a valid data MODATA[31:0] will be mapped onto the ODATA[31:0] for the Host CPU to read. This signal is driven by CPU Interface.
IMWE0#	O	**Internal Module Write byte 0 Command:** When active along with *MS#, a valid data IDATA[7:0] will be passed from the Host CPU to MIDATA[7:0]. This signal is driven by CPU Interface.
IMWE1#	O	**Internal Module Write byte 1 Command:** When active along with *MS#, a valid data IDATA[15:8] will be passed from the Host CPU to MIDATA[15:8]. This signal is driven by CPU Interface.
IMWE2#	O	**Internal Module Write byte 2 Command:** When active along with *MS#, a valid data IDATA[23:16] will be passed from the Host CPU to MIDATA[23:16]. This signal is driven by CPU Interface.
IMWE3#	O	**Internal Module Write byte 3 Command:** When active along with *MS#, a valid data IDATA[31:24] will be passed from the Host CPU to MIDATA[31:24]. This signal is driven by CPU Interface.

Notes: 1. * stands for peripheral module name.
2. # means that a signal is active low.

5.3 Function Description

CPU Interface Module is a bridge between the CPU bus and all peripheral modules in HD64465. This interface module handles the command, address and data transaction between CPU and HD64465 modules. The diagram of Figure 5-1 illustrates how the interface signals are connected. This CPU interface can operate under the 66MHz CKIO at maximum. The timing relationship of peripheral signals will be discussed in details in Signal Timing Description.

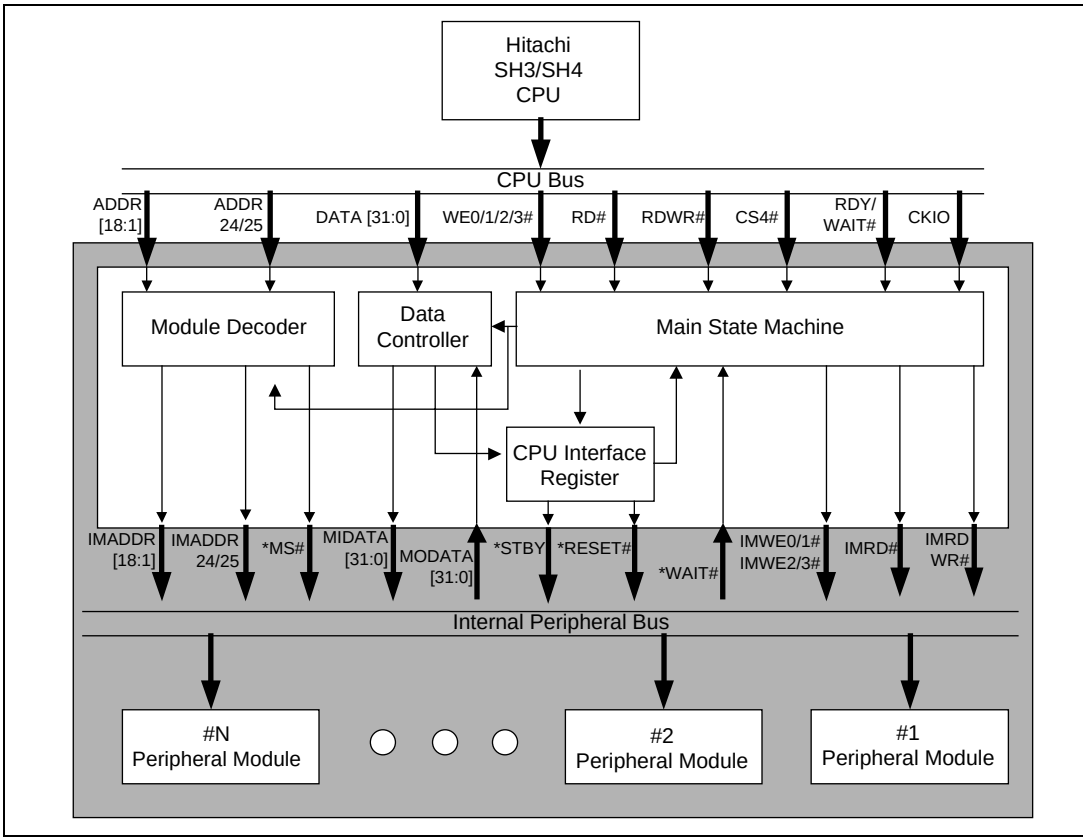


Figure 5.1 CPU Interface Module Interconnection Diagram

5.4 Signal Timing Description

5.4.1 Low Speed Timing

HD64465 provides a programmable bit (SLS) in System Configuration Register (SCONFR) to increase the system performance, depending on different bus clock (CKIO) rates. When SLS bit is programmed with 1, the internal bus timing is switched to the basic cycle composed of two wait states. When it is compared with High Speed Timing, a wait state will be found to save in command cycle. Thus, the performance in low bus clock rate is increased.

Low-Speed Basic Internal Peripheral Bus Access Timing is shown in the Figure 5-2. These basic cycles are T1, TWs1, TWs2, and T2 phases. Note that two wait states (TWs1, TWs2) are in command cycle. In this case, no external peripheral hardware wait is asserted. The *WAIT# signal is kept high before T2 stage. This means that peripheral module need no external cycles to accomplish the command. So, the command cycle enters the T2 phase after TWs2. At the end of T2 phase, the question that either T1 or T_idle phase is followed depends on the host CPU bus idle state configuration. If host CPU configures at least one idle state, the corresponding T_idle phase is followed. If host CPU configures no idle state, the T1 phase is followed after T2.

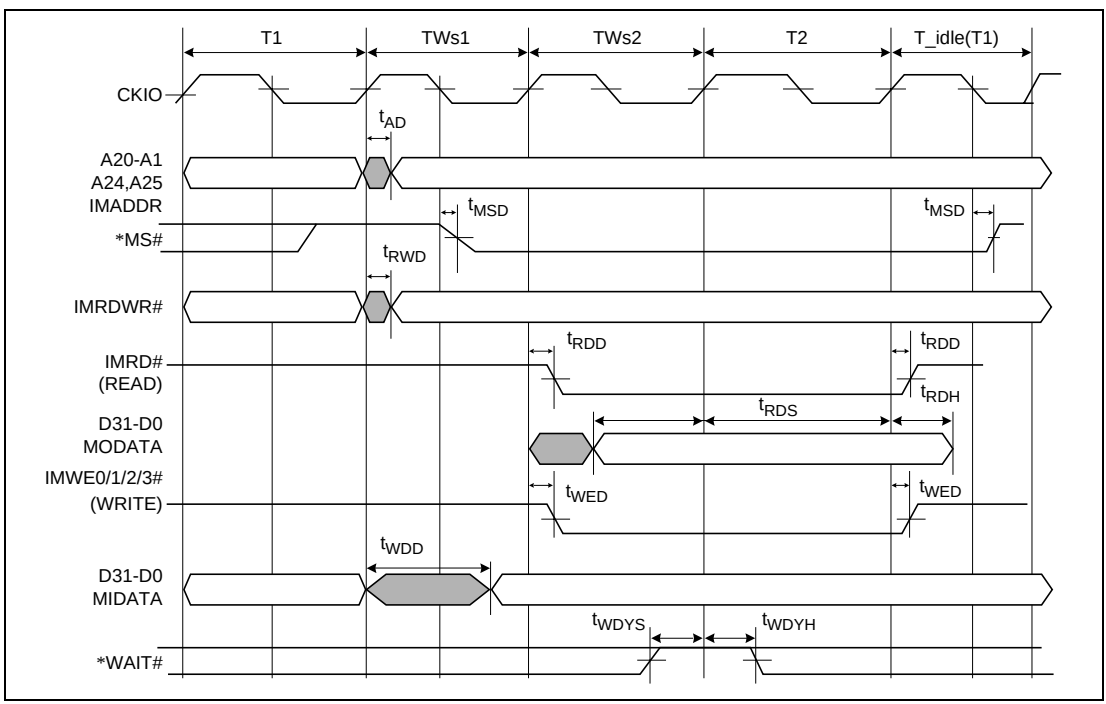


Figure 5.2 Low-Speed Basic Internal Peripheral Bus Access Timing

Low-Speed Internal Bus Access Timing with TWe phase is shown in the Figure 5-3. The diagram shows the cycles are T1, TWs1, TWs2, TWe and T2 phases. Note that there are two wait states and one external peripheral hardware wait state in command cycle. In this case, the basic command cycle is not enough for peripheral operation, it is responsible for the peripheral module to insert its own wait signal before T2 stage. And the *WAIT# signal controlled by the peripheral module must satisfy the setup/hold time defined in internal peripheral Bus AC timing specifications. In Figure 5-3, the TWe stands for the external peripheral hardware wait state. The command cycle can be inserted many TWe states, which depends on the signal *WAIT# of the peripheral module. At the end of T2 phase, the question that either T1 or T_idle phase is followed depends on the host CPU bus idle state configuration. If host CPU configures at least one idle state, the corresponding T_idle phase is followed. If host CPU configures no idle state, the T1 phase is followed after T2.

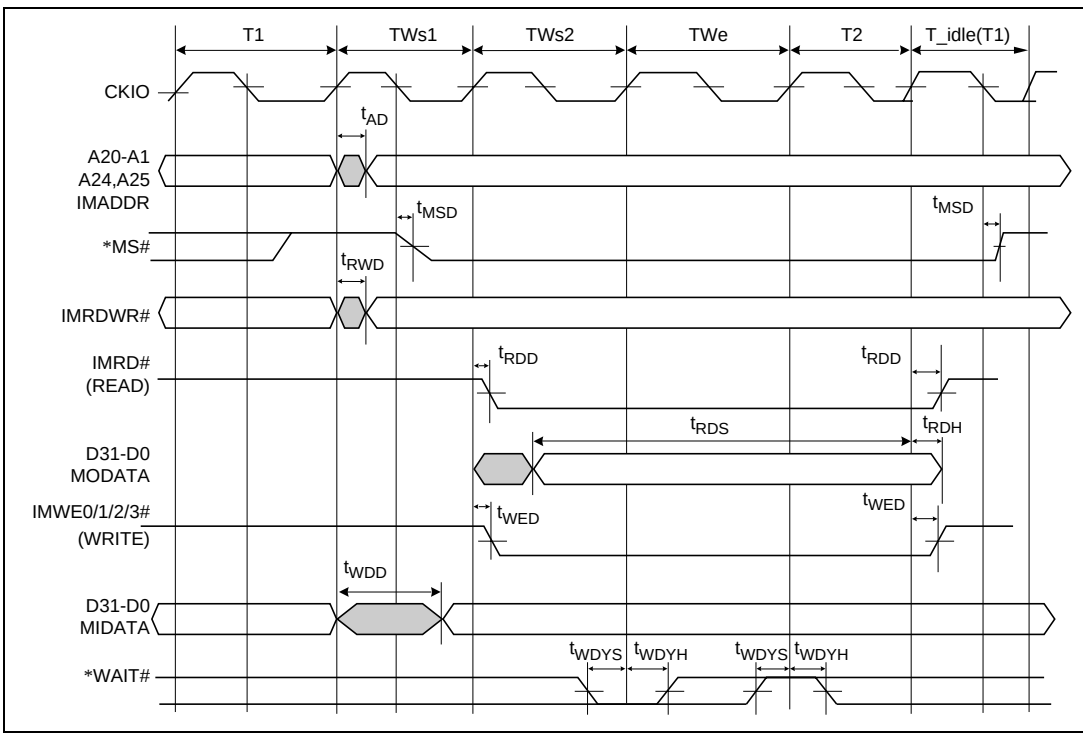


Figure 5.3 Low-Speed Internal Peripheral Bus Access Timing With TWe Phase

5.4.2 High Speed Timing

With the programmable bit (SLS) contained in System Configuration Register (SCONFR), the HD64465 is able to increase the system performance, depending on different bus clock (CKIO) rates. When SLS bit is programmed with 0, the internal bus timing is switched to the basic cycle composed of three wait states. This is because the internal H/W latency related to such a high speed clock rate. Though the High Speed Timing has to go through 3 wait states, which is one more wait state than the 2 wait states for Low Speed Timing, the performance of the High Speed Timing is still higher. This is because the High Speed Timing has higher CKIO rate. For example 25Mhz basic timing (4 clock cycles) needs 160 ns but 66Mhz basic timing (5 clock cycles) only requires 75 ns to complete.

High-Speed Basic Internal Peripheral Bus Access Timing is shown in the Figure 5-4. These basic cycles are T1, TWs1, TWs2, and TWs3 and T2 phases. Note that three wait states (TWs1, TWs2, and TWs3) are in command cycle. In this case, no external peripheral hardware wait is asserted. The *WAIT# signal is kept high before T2 stage. This means that peripheral module does not need external cycles to accomplish the command. Therefore, the command cycle enters the T2 phase after TWs2. At the end of T2 phase, the question that either T1 or T_idle phase is followed is determined by the configuration of the host CPU bus idle state. If host CPU configures at least one idle state, the corresponding T_idle phase is followed. If host CPU configures no idle state, the T1 phase is followed after T2.

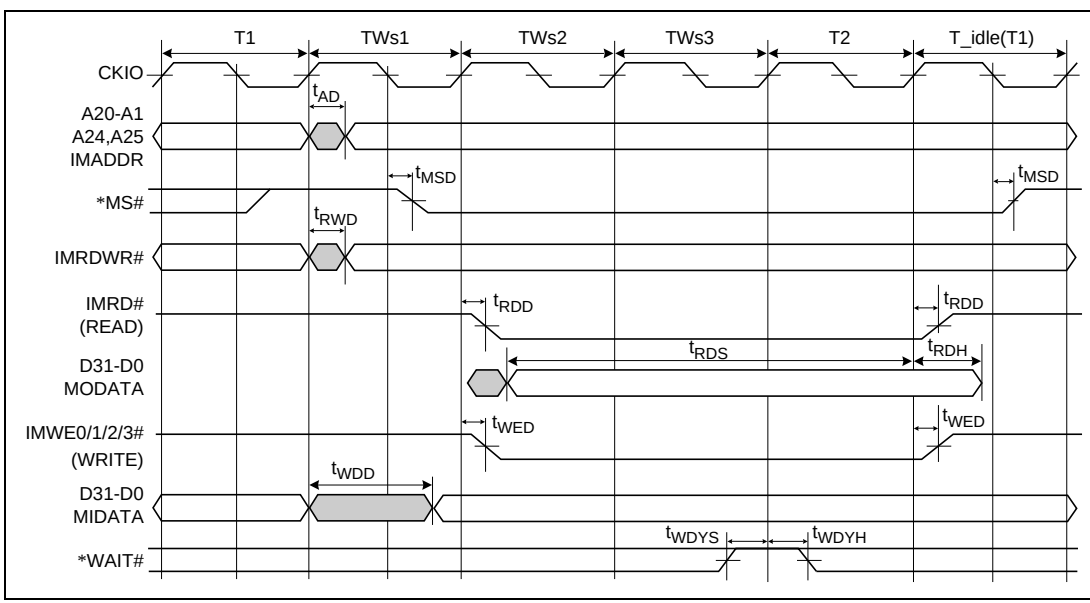


Figure 5.4 High-Speed Basic Internal Peripheral Bus Access Timing

High-Speed Internal Bus Access Timing with TWe phase is shown in the Figure 5-5. The diagram shows the cycles are T1, TWs1, TWs2, TWs3, TWe and T2 phases. Note that there are three wait states and one peripheral hardware wait state in command cycle. In this case, the basic command cycle is insufficient for peripheral operation. The basic command cycle is responsible for the peripheral module to insert its own wait signal before T2 stage. And the *WAIT# signal controlled by the peripheral module must satisfy the setup/hold time defined in internal peripheral Bus AC timing specifications. In Figure 5-5, the TWe stands for the peripheral hardware wait state. The command cycle can be inserted TWe states, and the number of Twe states insertion is depended on the signal *WAIT# of the peripheral module. At the end of T2 phase, the timing when either T1 or T_idle phase is followed is based on the same logic described in Figure 5-2 – the question that either T1 or T_idle phase is followed depends on the configuration of the host CPU bus idle state.

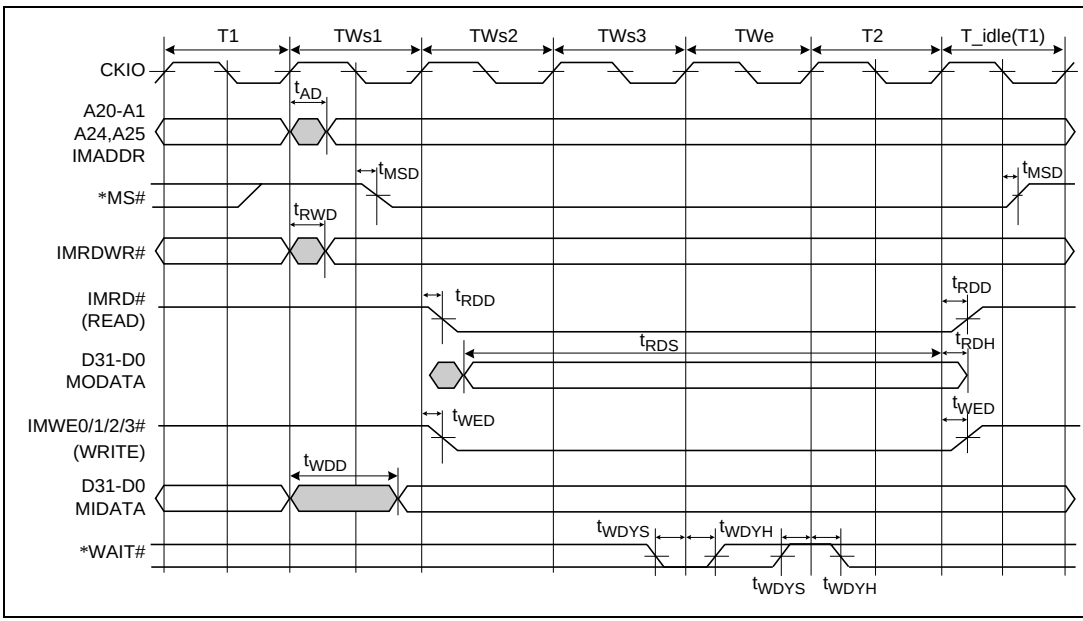


Figure 5.5 High-Speed Internal Peripheral Bus Access Timing With TWe Phase

5.5 Internal Bus Data Swap Rules

Internal Bus Data Swap Rules are defined to satisfy the legacy peripheral modules. This is because the data bus width has been changed from 16 bits to 32 bits. For the compliance with these legacy peripheral modules, it is required to establish a data swap mechanism described below:

Case 1: Word Access (16 bits)

IMADDR[1]=0	**CPU Bus**				**Internal Bus**			
Write Enable	WE3#	WE2#	WE1#	WE0#	IMWE3#	IMWE2#	IMWE1#	IMWE0#
Value	H	H	L	L	H	H	L	L
Data Position	X	x	Byte1	byte0	x	x	byte1	byte0

IMADDR[1]=1	**CPU Bus**				**Internal Bus**			
Write Enable	WE3#	WE2#	WE1#	WE0#	IMWE3#	IMWE2#	IMWE1#	IMWE0#
Value	L	L	H	H	H	H	L	L
Data Position	byte3	byte2	x	x	x	x	byte3	byte2

Case 2: Double Word Access (32 bits)

IMADDR[1]=0	**CPU Bus**				**Internal Bus**			
Write Enable	WE3#	WE2#	WE1#	WE0#	IMWE3#	IMWE2#	IMWE1#	IMWE0#
Value	L	L	L	L	L	L	L	L
Data Position	byte3	byte2	Byte1	byte0	byte3	byte2	byte1	byte0

5.6 Internal Peripheral Bus AC Timing Specification

		SH-3 (15MHz)		SH-3 (40MHz)		SH-4 (66MHz)		
Symbol	**Item**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
TAD	Address delay time	-	2	-	2	-	2	ns
TMSD	Module Select delay time	-	2	-	2	-	2	ns
TRWD	Read Write delay time	-	2	-	2	-	2	ns
TRDD	Read Strobe delay time	-	5	-	5	-	5	ns
TRDS	Read Data setup time	50	-	30	-	10	-	ns
TRDH	Read Data hold time	0	-	0	-	0	-	ns
TWED	Write Enable delay time	-	5	-	5	-	5	ns
TEDD	Write Data delay time	-	0	-	0	-	10	ns
TWDYS	Wait setup time	20	-	20	-	10	-	ns
TWDYH	Wait hold time	0	-	0	-	0	-	ns

Note: In Figure 5-2 and Figure 5-3, all AC Timings are related to CKIO. The CKIO signal is inside IPC, i.e. this CKIO signal is the signal comes out through IPC CKIO input pad and clock tree, and is used by all peripheral modules.

Section 6 Power Management and System Configuration

6.1 Overview

The System Power Management and Configuration registers control the functionality of Module Standby mode, Bus gating, Wait states, Peripheral Clock Control, Module Software Reset and Test Mode. Each module in the HD64465 is provided with the STANDBY mode. All peripheral module functions are halted in the STANDBY mode; thereby reducing the power consumption. The Bus gating control is used with STANDBY mode for further power saving capability The Hardware external wait cycle inserted by CPU interface module is an option to extend data read/write cycles. The system registers are described in details below:

6.2 Features

- Support STANDBY mode for each peripheral module
- All peripheral clocks can be halted
- Provide CPU interface input signals and GPIO pins gated function
- Flexible selection of peripheral functions
- Provide an Debug port for system bus test

6.3 Register Description

The following table lists all the registers. The unit of the register size and access size is byte. The register size is the actual size of registers. The access size defines the data bus width of host CPU, which is used to access each register. In other words, the access of each register is word (2 bytes) access type.

Table 6.1 The Register List of Power Management and System Configuration

Name	Address	Register Size	Access Size
System Module Standby Control Register	H'10000000	16	16
System Configuration Register	H'10000002	16	16
System Bus Control Register	H'10000004	16	16
System Peripheral Clock Control Register	H'10000006	16	16
System Peripheral S/W Reset Control Register	H'10000008	16	16
System PLL Control Register	H'1000000A	16	16
System Revision Register	H'1000000C	16	16
System Device ID Register	H'10000010	16	16
System Debug Port Register	H'10000FF0	16	16

6.3.1 System Module Standby Control Register (SMSCR)

This register provides the module standby control for each peripheral module. This standby control can make the peripheral module enter standby mode. The power consumption can be reduced considerably as a result. If the module needs to be activated again, the peripheral module must exit the standby mode by clearing the corresponding bit in the register SMSCR.

Address: 0x10000000

Bit	15	14	13	12	11	10	9	8
Bit Name	-	PS2ST	-	ADCST	UARTST	-	SCDIST	PPST
Initial Value	0	1	0	1	1	0	1	1
R/W	R	R/W	R	R/W	R/W	R	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Bit Name	-	PC0ST	PC1ST	AFEST	TM0ST	TM1ST	IRDAST	KBCST
Initial Value	0	1	1	1	1	1	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

System Module Standby Control Register (SMSCR) [cont'd]

Bit	Description	Default
15	Reserved.	0
14	**PS2ST:** PS2 Standby. When this bit is set, the PS2 will enter the standby mode until this bit is cleared. The PS2 will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
13	Reserved.	0
12	**ADCST:** A/D Controller Standby. When this bit is set, the A/D controller will enter the standby mode until this bit is cleared. The A/D controller will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
11	**UARTST:** UART Standby. When this bit is set, the UART will enter the standby mode until this bit is cleared. The UART will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
10	Reserved	0
9	**SCDIST:** Serial Codec Interface Standby. When this bit is set, the serial codec interface will enter the STANDBY mode until this bit is cleared. The serial codec interface will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
8	**PPST:** Parallel Port Standby. When this bit is set, the parallel port will enter the standby mode until this bit is cleared. The parallel port will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
7	Reserved	0
6	**PC0ST:** PCMCIA interface Channel 0 Standby. When this bit is set, the PCMCIA interface channel 0 will enter the standby mode until this bit is cleared. The PCMCIA interface channel 0 will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
5	**PC1ST:** PCMCIA interface Channel 1 Standby. When this bit is set, the PCMCIA interface channel 1 will enter the standby mode until this bit is cleared. The PCMCIA interface channel 1 will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
4	**AFEST:** AFE interface Standby. When this bit is set, the AFE interface will enter the standby mode until this bit is clear. The AFE interface will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
3	**TM0ST:** Timer channel 0 Standby. When this bit is set, the Timer channel 0 will enter the standby mode until this bit is cleared. The Timer channel 0 will be in normal operation mode after this bit is cleared. This bit is clear after reset.	1
2	**TM1ST:** Timer channel 1 Standby. When this bit is set, the Timer channel 1 will enter the standby mode until this bit is cleared. The Timer channel 1 will be in normal operation mode after this bit is cleared. This bit is clear after reset.	1
1	**IRDAST:** IRDA Controller Standby. When this bit is set, the IrDA controller will enter the standby mode until this bit is cleared. The IrDA controller will be in normal operation mode after this bit is cleared. This bit is set after reset.	1
0	**KBCST:** KBC Controller Standby. When this bit is set, the KBC controller will enter the standby mode until this bit is cleared. The KBC controller will be in normal operation mode after this bit is cleared. This bit is clear after reset.	1

6.3.2 System Configuration Register (SCONFR)

This register provides a flexible approach for system configuration. The hardware wait insertion control is flexible to control CPU interface command cycle. Parallel Port function select can also be programmed by this register. The detailed functionality, which can be configured, is described below:

Address: H'10000002

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	SLS	HWEN	HW3	HW2	HW1	HW0
Initial Value	0	0	0	1	0	0	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	USBCKS	SCDICKS	PPFMS1	PPFMS0	KBWUP	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R

Bit	Description	Default
15 - 14	Reserved.	0
13	**SLS:** System Low Speed Select. This bit is used to select the low speed or high-speed timing according to different bus clock (CKIO) rates. When this bit is set to 1, the low speed timing is selected in internal bus. When this bit is cleared to 0, the high-speed timing is selected in internal bus. For CKIO, - 25MHz, low speed is recommended. For CKIO, 25 – 66MHz, high speed is recommended.	
12	**HWEN:** CPU interface Hardware Wait Number Enable. This bit is used to enable the wait cycles of HW[3:0]. When this bit is set, the CPU interface will insert the hardware wait cycles as the HW[3:0] programmed. If this bit is cleared, the CPU interface will not insert any hardware wait cycles.	1
11 - 8	**HW[3:0]:** CPU interface Hardware Wait Number Control. The number of HW[3:0] stands for the cycles of hardware wait state inserted. The inserted cycles start at the second software wait state. This wait number is effective only after the HWEN has been set. The wait cycle can be any one number from 1 to 15. Note that the relationship between HW[3:0] and CPU programmed inserted wait states (IWS) is $2 \leq IWS \leq 1 + HW[3:0]$. Hence, the CPU default inserted wait states should be 2.	0001
7 - 6	Reserved.	0

Bit	Description	Default
5	**USBCKS:** USB Host interface Clock Switch. When this bit is cleared, USB Host interface clock will be the clock output from PLL1. When this bit is set, the USB Host interface clock will be the half frequency of CKIO.	0
4	**SCDICKS:** Serial Codec Interface Clock Switch. When this bit is set, SCDI_clk source will be 12MHz from UCK pad path. When this bit is cleared, the SCDI_clk source will be 12.288MHz from AFECK pad path.	0
3, 2	**PPFMS[1:0]:** Parallel Port Function Mode Select. When PPFMS[1:0] is 11, the ECP+EPP mode is selected. When PPFMS[1:0] is 10, the ECP mode is selected. When PPFMS[1:0] is 01, the EPP mode is selected. When PPFMS[1:0] is 00, the SPP mode is selected.	00
1	**KBWUP:** Key Board Wake Up. When this bit is set, the key board wake-up pulse will be generated to wake up key board controller. This bit is self-cleared after the wake-up pulse is done.	0
0	Reserved.	0

6.3.3 System Bus Control Register (SBCR)

This register controls the bus state of some input or output signals. This signal gating control can be used to save power consumption for various system conditions.

Address: H'10000004

Bit	15	14	13	12	11	10	9	8
Bit Name	PDOF	PDIG	PCOF	PCIG	PBOF	PBIG	PAOF	PAIG
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	-	CSPE	CMDPE	ADDRPE	DATAPE	CPUBIG	PEOF	PEIG
Initial Value	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

System Bus Control Register (SBCR) [cont'd]

Bit	Description	Default
15	**PDOF:** Port D Output Floating Control. When this bit is set, the output will be floating. When this bit is cleared, the output floating is disabled.	0
14	**PDIG:** Port D Input Gating Control. When this bit is set, the input to port D will be gated to fixed value. When this bit is cleared, the input remains unaffected.	0
13	**PCOF:** Port C Output Floating Control. When this bit is set, the output will be floating. When this bit is cleared, the output floating is disabled.	0
12	**PCIG:** Port C Input Gating Control. When this bit is set, the input to port C will be gated to fixed value. When this bit is cleared, the input remains unaffected.	0
11	**PBOF:** Port B Output Floating Control. When this bit is set, the output will be floating. When this bit is cleared, the output floating is disabled.	0
10	**PBIG:** Port B Input Gating Control. When this bit is set, the input to port B will be gated to fixed value. When this bit is cleared, the input remains unaffected.	0
9	**PAOF:** Port A Output Floating Control. When this bit is set, the output will be floating. When this bit is cleared, the output floating is disabled.	0
8	**PAIG:** Port A Input Gating Control. When this bit is set, the input to port A will be gated to fixed value. When this bit is cleared, the input remains unaffected.	0
7	Reserved	0
6	**CSPE:** CPU Chip Area Select Pull-up Enable. When this bit is cleared, the chip select CS4_ will be pull-up. When this bit is set, the CS4_ is not pull-up.	0
5	**CMDPE:** CPU Command/Status Pull-up Enable. When this bit is cleared, the signal RD_, RDWR_, WE0_ and WE1_ will be pull-up. When this bit is set, the signal RD_, RDWR_, WE0_ and WE1_ are not pull-up.	0
4	**ADDRPE:** CPU Address Bus Pull-up Enable. When this bit is cleared, the address bus will be pull-up. When this bit is set, the address bus is not pull-up.	0
3	**DATAPE:** CPU Data Bus Pull-up Enable. When this bit is set, the data bus will be pull-up. When this bit is cleared, the data bus is not pull-up.	0
2	**CPUBIG:** CPU Bus interface Input Gating Control. When this bit is set, the CPU bus interface will be automatic input gating by Intelligent Peripheral Controller Select(CS4_) for reducing power consumption. When this bit is cleared, the CPU bus interface is not gated.	0
0	**PEOF:** Port E Output Floating Control. When this bit is set, the output will be floating. When this bit is cleared, the output floating is disabled.	0
0	**PEIG:** Port E Input Gating Control. When this bit is set, the input to port E will be gated to fixed value. When this bit is cleared, the input remains unaffected.	0

6.3.4 System Peripheral Clock Control Register (SPCCR)

This register provides the function of peripheral clock control for each peripheral module. When the peripheral module is in standby mode, the peripheral clock can be turned off to reduce more power consumption, thanks to the free running of the peripheral clock. To stop the peripheral clock, the peripheral module standby mode must be asserted first.

Address: H'10000006

Bit	15	14	13	12	11	10	9	8
Bit Name	ADCCLK	-	UARTCLK	PPCLK	FIRCLK	SIRCLK	SCDICLK	KBCCLK
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Bit Name	USBCLK	AFECLK	-	-	-	-	UCKOSC	AFEOSC
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	Description	Default
15	**ADCCLK:** A/D Controller Clock Control. When this bit is set, the A/D controller clock will be halted. The A/D controller clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0
14	Reserved	0
13	**UARTCLK:** UART Controller Clock Control. When this bit is set, the UART clock will be halted. The UART channel 0 clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0
12	**PPCLK:** Parallel Port Controller Clock Control. When this bit is set, the PP clock will be halted. The PP clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0

System Peripheral Clock Control Register (SPCCR) [cont'd]

Bit	Description	Default
11	**FIRCLK:** FIR Controller Clock Control. When this bit is set, the FIR clock will be halted. The FIR clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0
10	**SIRCLK:** SIR Controller Clock Control. When this bit is set, the SIR clock will be halted. The SIR clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0
9	**SCDICLK:** Serial Codec Interface Clock Control. When this bit is set, the SCDI clock will be halted. The SCDI clock will run normally after this bit is cleared.. Note that this bit can be cleared only Twkst ms later after either AFEOSC bit or UCKOSC bit, which is determined by bit SCDICKS, is cleared.	0
8	**KBCCLK:** Key Board Controller Clock Control. When this bit is set, the KBC command clock will be halted. The KBC command clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0
7	**USBCLK:** USB Controller Clock Control. When this bit is set, the USB clock will be halted. The USB clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the UCKOSC bit has already been cleared.	0
6	**AFECLK:** AFE interface Controller Clock Control. When this bit is set, the AFE interface clock will be halted. The AFE interface clock will run normally after this bit is cleared. Note that this bit can be cleared only Twkst ms later, the AFEOSC bit has already been cleared.	0
5 - 2	Reserved.	0
1	**UCKOSC:** UCK Oscillator/Crystal Control. When this bit is set, the UCK oscillator/crystal will be halted. The UCK oscillator/crystal will start to run after this bit is cleared. This bit must be carefully used. This bit can only be set when all the clocks that use the UCK as a source clock have been halted.	0
0	**AFEOSC:** AFECK Oscillator/Crystal Control. When this bit is set, the AFECK oscillator/crystal will be halted. The AFECK oscillator/crystal will start to run after this bit is cleared. This bit must be carefully used. This bit can only be set when all the clocks that use the AFECK as a source clock have been halted.	0

Note: The parameter, Twkst = 15 ms.

Peripheral Clock Relationship Diagrams shows the working relationship among the clock source and generated peripheral clocks. It also indicates the sequence of turning off one peripheral clock without interfering the other peripheral clocks operation. For example, in Figure 6-1, If the bit **AFEOSC** is set, then peripheral clocks like AFE_clk and SCDI_clk will be halted because the source clock AFECK is halted. To stop the AFE_clk, users can set the bit **AFECLK** in the system peripheral clock control register (SPCCR). To turn on the AFE_clk again from the halted condition, just let the bit **AFECLK** be cleared, AFE_clk will then start to run.

Follow the steps below to save power consumption:

1. Set bits **AFECLK** and **SCDICLK** to turn off AFE_clk and SCDI_clk (under **SCDICKS**=0). This will reduce the power consumption of AFE and SCDI modules.
2. Even more power consumption can be saved on PLL1 if users set the bit **AFEOSC**.

Conversely, to make the peripheral clocks AFE_clk and SCDI_clk (under **SCDICKS**=0) to run from **AFEOSC** bit, which has been set, users are required to follow the steps described below:

1. Clear **AFEOSC**, then wait about $\mathbf{t_{PLL}}$ ms to allow PLL1 operate normally.
2. Clear **AFECLK** and **SCDICLK** to open the clock gating.

After these two steps, the peripheral clocks will start to run normally.

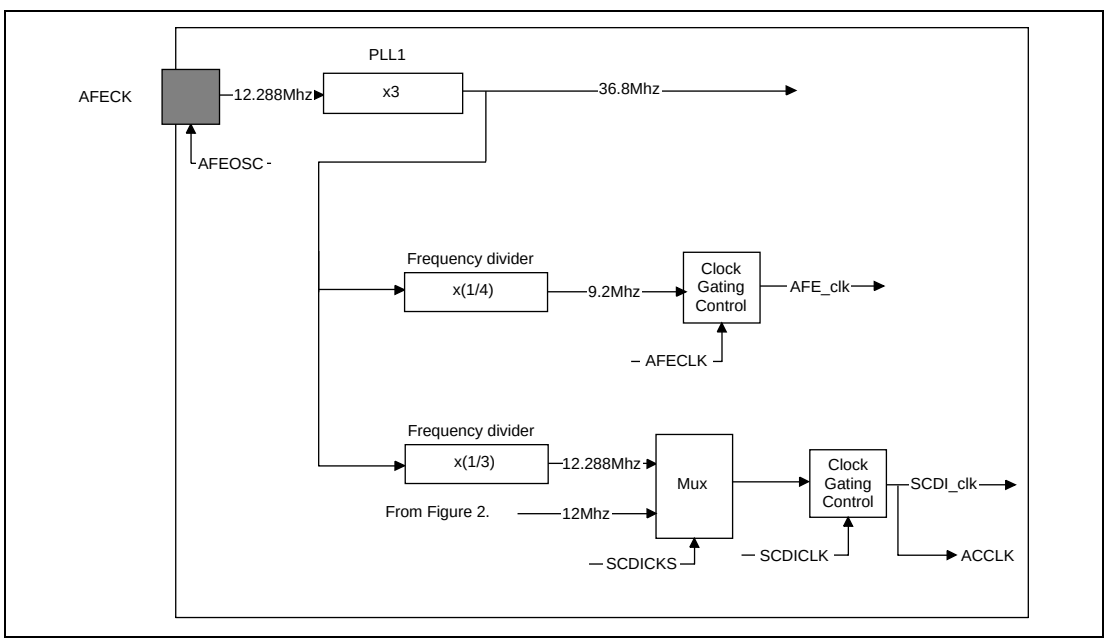


Figure 6.1 AFECK Related Clock Diagram

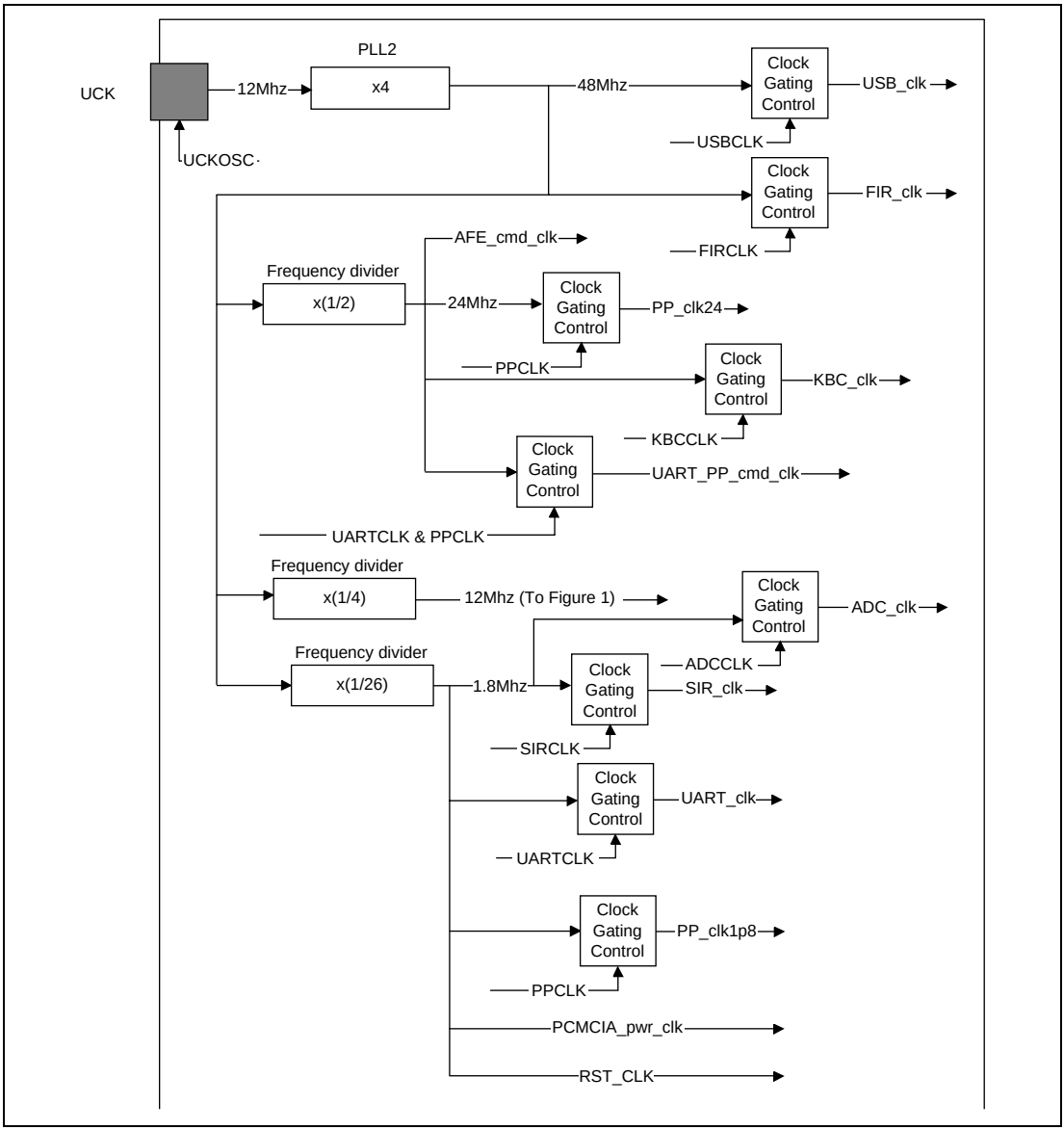


Figure 6.2 UCK Related Clock Diagram

6.3.5 System Peripheral S/W Reset Control Register (SPSRCR)

The software reset of each peripheral module is an option when the peripheral module encounters functional failures after clearing the STANDBY mode of the module. These software reset bits need the clock from UCK oscillator to count the reset period, which means that the **UCKOSC** bit needs to be cleared first before setting these reset bits. Note that multiple bits can be set at the same time. But during the reset period, no other bits can be set until all the bits, which have been set, are cleared.

Address: H'10000008

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	SPORST	PS2SRT	-	ADCSRT	UARTSRT	-	SCDISRT	PPSRT
Initial Value	1	0	0	0	0	0	0	0
R/W	R	R/W	R	R/W	R/W	R	R/W	R/W

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	USBSRT	PC0SRT	PC1SRT	AFESRT	TM0SRT	TM1SRT	IRDASRT	KBCSRT
Initial Value	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

Bit	**Description**	**Default**
15	**SPORST:** System Power-On Reset. When this bit is set, the system power-on reset is going. At this time, the whole chip is still in reset state after H/W reset. All R/W accesses must wait for this bit to clear to zero. Note this register is readable during system power-on reset state.	1
14	**PS2SRT:** PS2 Controller Software Reset. When this bit is set, the PS2 Controller will be reset. This reset is equivalent to hardware reset. All the PS2 Controller registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
13	Reserved.	0
12	**ADCSRT:** A/D Controller Software Reset. When this bit is set, the A/D controller will be reset. This reset is equivalent to hardware reset. All the A/D controller registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
11	**UARTSRT:** UART Controller Software Reset. When this bit is set, the UART will be reset. This reset is equivalent to hardware reset. All the UART registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
10	Reserved	0
9	**SCDISRT:** Serial Codec Controller Software Reset. When this bit is set, the SCDI will be reset. This reset is equivalent to hardware reset. All the SCDI registers are set to the reset default values. Note that the software reset bit is self-clearing.	0

System Peripheral S/W Reset Control Register (SPSRCR) [cont'd]

Bit	Description	Default
8	**PPSRT:** Parallel Port Controller Software Reset. When this bit is set, the parallel port controller will be reset. This reset is equivalent to hardware reset. All the parallel port registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
7	**USBSRT:** USB Controller Software Reset. When this bit is set, the USB will be reset. This reset is equivalent to hardware reset. All the USB registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
6	**PC0SRT:** PCMCIA Channel 0 Controller Software Reset. When this bit is set, the PCMCIA channel 0 will be reset. This reset is equivalent to hardware reset. All the PCMCIA channel 0 registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
5	**PC1SRT:** PCMCIA Channel 1 Controller Software Reset. When this bit is set, the PCMCIA channel 1 will be reset. This reset is equivalent to hardware reset. All the PCMCIA channel 1 registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
4	**AFESRT:** AFE interface Controller Software Reset. When this bit is set, the AFE interface controller will be reset. This reset is equivalent to hardware reset. All the AFE interface registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
3	**TM0SRT:** Timer Channel 0 Controller Software Reset. When this bit is set, the Timer channel 0 will be reset. This reset is equivalent to hardware reset. All the Timer channel 0 registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
2	**TM1SRT:** Timer Channel 1 Controller Software Reset. When this bit is set, the Timer channel 1 will be reset. This reset is equivalent to hardware reset. All the Timer channel 1 registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
1	**IRDASRT:** IrDA Controller Software Reset. When this bit is set, the IrDA controller will be reset. This reset is equivalent to hardware reset. All the IrDA registers are set to the reset default values. Note that the software reset bit is self-clearing.	0
0	**KBCSRT:** KBC Controller Software Reset. When this bit is set, the KBC controller will be reset. This reset is equivalent to hardware reset. All the KBC registers are set to the reset default values. Note that the software reset bit is self-clearing.	0

6.3.6 System PLL Control Register (SPLLCR)

This register provides the PLL control options.

Address: H'1000000A

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	PLL2SB	PLL1SB	-	-	PLL2BP	PLL1BP
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Description	Default
15 - 6	Reserved.	0
5	**PLL2SB:** PLL2 Standby control. When this bit is set, the PLL2 standby mode is enabled. When this bit is cleared, the PLL2 standby mode is disabled.	0
4	**PLL1SB:** PLL1 Standby control. When this bit is set, the PLL1 standby mode is enabled. When this bit is cleared, the PLL1 standby mode is disabled.	0
3, 2	Reserved.	0
1	**PLL2BP:** PLL2 Bypass control. When this bit is set, the PLL2 bypass mode is enabled. The clock input will directly connect to clock output, like the PLL multiplier is one. The multiplier is one. When this bit is cleared, the PLL2 bypass mode is disabled.	0
0	**PLL1BP:** PLL1 Bypass control. When this bit is set, the PLL1 bypass mode is enabled. The clock input will directly connect to clock output, like the PLL multiplier is one. When this bit is cleared, the PLL1 bypass mode is disabled.	0

6.3.7 System Revision Register (SRR)

This register records the revision number of the controller, which is read only. The revision number is presented with **mj.mi**. The **mj[7:0]** stands for the major change number and its value is the content of the register's high byte. The **mi[7:0]** stands for the minor change number, and its value is the content of the register's low byte. For this version, the major change number is **1**, and the minor change number is **0.**

Address: H'1000000C

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	mj7	mj6	mj5	mj4	mj3	mj2	mj1	mj0
Initial Value	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	mi7	mi6	mi5	mi4	mi3	mi2	mi1	mi0
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	**Description**	**Default**
15 - 8	**mj[7:0]:** This number is the major change number of revision	1
7 - 0	**mi[7:0]:** This number is the minor change number of revision.	0

6.3.8 System Device ID Register (SDID)

Address: H'10000010

This register stands for the chip ID. The value of this chip is 0x8122 and it is read only.

6.4 System Hardware Reset Timing

HD64465 provides the system hardware reset function to control SH-4/SH-3 CPU power-on reset or manual reset. In Figure 6-3, it shows that two input signals on HD64465, RESETMI# and RESETPI#, which control the manual reset and power-on reset respectively. The other signals, RESETPO# and RESETMO#, are the output signals connected to SH-4/SH-3 CPU for power-on reset and manual reset. AFECK must be input to enable this function.

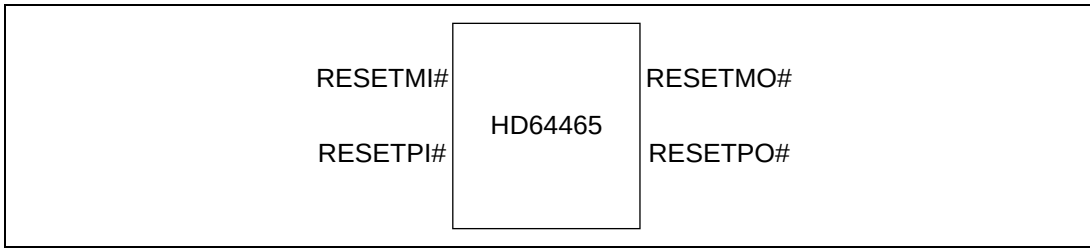


Figure 6.3 System Hardware Reset Related Pins

6.4.1 Power-On Reset Output

When RESETPI# is asserted, it means the power-on reset has occurred. The power-on reset signal output from HD64465, RESETPO#, is connected to CPU power-on reset input. The related timing is illustrated in Fig 6-4.

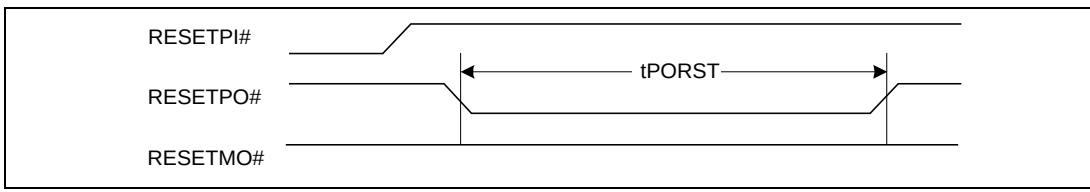


Figure 6.4 Power-On Reset Diagram, tPORST=10ms

6.4.2 Manual Reset Output

When RESETMI# is asserted, it means the manual reset is occurred. The manual reset signal from HD64465, RESETMO#, is connected to CPU manual reset input. For SH-4 and SH-3 CPU, the manual reset mechanism is different. Figure 6-5 shows the manual reset timing for SH-4 CPU and Figure 6-6 shows the manual reset timing for SH-3 CPU.

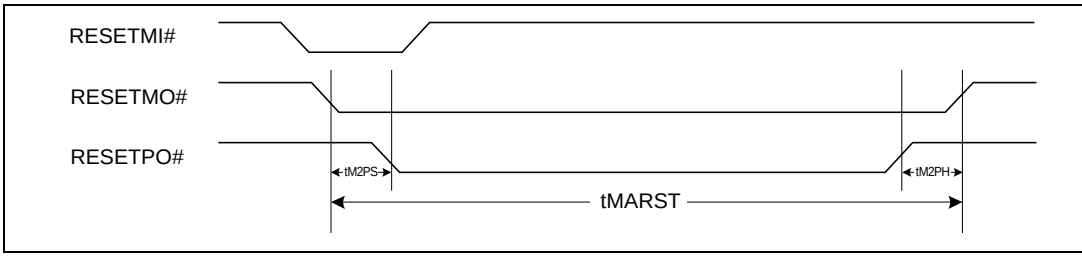


Figure 6.5 SH4 Manual Reset Diagram, tM2PS=tM2PH=80ns, tMARST=10ms

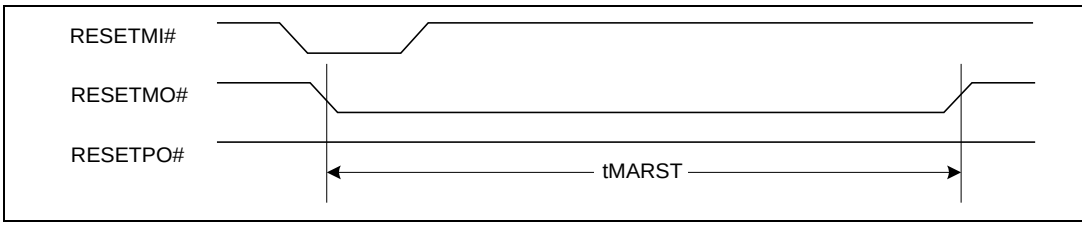


Figure 6.6 SH3 Manual Reset Diagram, tMARST=10ms

- CAUTION

  HD64465 dosen't have manual reset function for itself. Please be sure that manual reset will not initialize any register in HD64465.

Section 7 General Purpose I/O Port

7.1 Overview

The HD64465 incorporates five general purpose 8-bit I/O ports (Port A, Port B , Port C, Port D and Port E). As shown in the **Table 7-1** below, the port pins are multiplexed with other functions, which are controlled by Port Control Registers **(GPxCR x: A,B,C,D,E)**. Each port contains a 8-bit data register **(GPxDR x: A,B,C,D,E)** which reflects the data of the pins. Each I/O port pin has a pull-up MOS, which can be controlled by the Port Control Register to determine whether this pin will be pulled up or not. The Interrupt Control Registers **(GPxICR x: A,B,C,D,E)** are provided to individually control each pin which can be enabled or disabled to generate an Interrupt signal. The interrupt events type can also be selected through the Interrupt Control Registers **(GPxICR x: A,B,C,D,E)** to determine which edge (falling edge or rising edge) will generate an interrupt. As interrupt events occur at any GPIO pin, the Interrupt Status Registers **(GPxISR x: A,B,C,D,E)** can record the occurring interrupt events, which can be read by system. The interrupt is ended by writing '1' to the corresponding bit of the Interrupt Status Register, and the interrupt status is then cleared.

7.1.1 Features

- Input pull-up on/off control
- Interrupt events can be independently enabled or masked on each I/O pin
- Interrupt events can be independently selected as rising or falling edge trigger
- Function multiplex with AFE, KBC and TIMER control signals
- Support power down mode which is controlled by software

Table 7.1 The List of I/O Port Pin Function Configurations

Port	Function 1	Function 2
A	PA7 I/O (port)	Reserved
A	PA6 I/O (port)	Reserved
A	PA5 I/O (port)	Reserved
A	PA4 I/O (port)	Reserved
A	PA3 I/O (port)	Reserved
A	PA2 I/O (port)	Reserved
A	PA1 I/O (port)	Reserved
A	PA0 I/O (port)	Reserved

Table 7.1 The List of I/O Port Pin Function Configurations(cont'd)

Port	Function 1	Function 2
B	PB7 I/O (port)	Reserved
B	PB6 I/O (port)	Reserved
B	PB5 I/O (port)	KBC RESUME# (KBC)
B	PB4 I/O (port)	KBC WAKEUP# (KBC)
B	PB3 I/O (port)	Reserved
B	PB2 I/O (port)	Reserved
B	PB1 I/O (port)	TMO1# (Timer)
B	PB0 I/O (port)	TMO0# (Timer)
C	PC7 I/O (port)	Reserved
C	PC6 I/O (port)	Reserved
C	PC5 I/O (port)	Reserved
C	PC4 I/O (port)	Reserved
C	PC3 I/O (port)	Reserved
C	PC2 I/O (port)	Reserved
C	PC1 I/O (port)	Reserved
C	PC0 I/O (port)	Reserved
D	PD7 I/O (port)	Reserved
D	PD6 I/O (port)	Reserved
D	PD5 I/O (port)	Reserved
D	PD4 I/O (port)	Reserved
D	PD3 I/O (port)	Reserved
D	PD2 I/O (port)	Reserved
D	PD1 I/O (port)	Reserved
D	PD0 I/O (port)	Reserved
E	PE7 I/O (port)	Reserved
E	PE6 I/O (port)	Reserved
E	PE5 I/O (port)	Reserved
E	PE4 I/O (port)	Reserved
E	PE3 I/O (port)	Reserved
E	PE2 I/O (port)	Reserved
E	PE1 I/O (port)	Reserved
E	PE0 I/O (port)	Reserved

7.2 Register Configuration

Each I/O Port consists of four registers: Port Control Register, Port Data Register, Interrupt Control Register and Interrupt Status Register. Table 7-2 below summarizes the port address configuration of each register.

Table 7.2 The List of Register Configurations

Name	Abbr.	R/W	Initial Value	Address	Register Size	Access Size
Port A Control Register	GPACR	R/W	H'FFFF	H'10004000	16	16
Port B Control Register	GPBCR	R/W	H'FFFF	H'10004002	16	16
Port C Control Register	GPCCR	R/W	H'FFFF	H'10004004	16	16
Port D Control Register	GPDCR	R/W	H'FFFF	H'10004006	16	16
Port E Control Register	GPECR	R/W	H'FFFF	H'10004008	16	16
Port A Data Register	GPADR	R/W	-	H'10004010	8	16
Port B Data Register	GPBDR	R/W	-	H'10004012	8	16
Port C Data Register	GPCDR	R/W	-	H'10004014	8	16
Port D Data Register	GPDDR	R/W	-	H'10004016	8	16
Port E Data Register	GPEDR	R/W	-	H'10004018	8	16
Port A Interrupt Control Register	GPAICR	R/W	H'0000	H'10004020	16	16
Port B Interrupt Control Register	GPBICR	R/W	H'0000	H'10004022	16	16
Port C Interrupt Control Register	GPCICR	R/W	H'0000	H'10004024	16	16
Port D Interrupt Control Register	GPDICR	R/W	H'0000	H'10004026	16	16
Port E Interrupt Control Register	GPEICR	R/W	H'0000	H'10004028	16	16
Port A Interrupt Status Register	GPAISR	R	H'0000	H'10004040	8	16
Port B Interrupt Status Register	GPBISR	R	H'0000	H'10004042	8	16
Port C Interrupt Status Register	GPCISR	R	H'0000	H'10004044	8	16
Port D Interrupt Status Register	GPDISR	R	H'0000	H'10004046	8	16
Port E Interrupt Status Register	GPEISR	R	H'0000	H'10004048	8	16

7.3 Register Descriptions

All ports are 8-bit input/output ports with the pin configuration shown in **Figure 7-1** below. Each pin contains an input pull-up MOS, which is controlled by its I/O Control Register **(GPxCR x:A,B,C,D,E).**

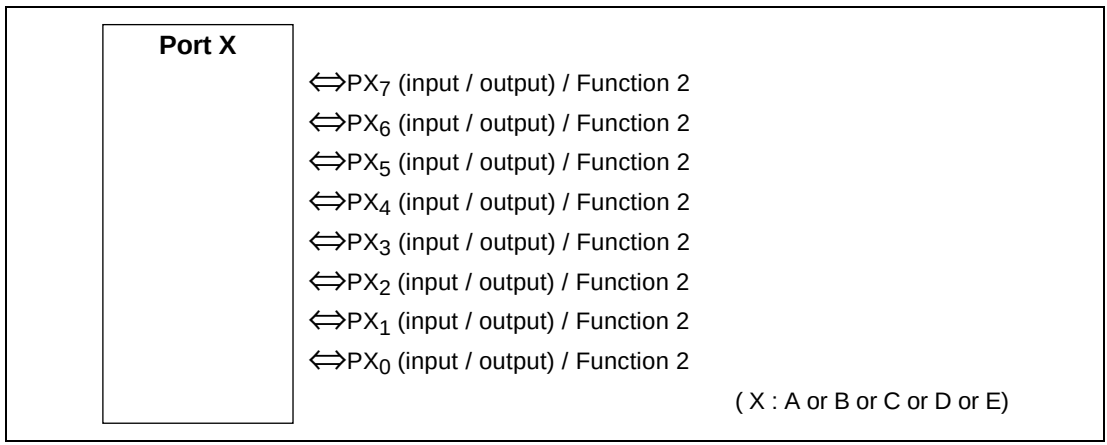


Figure 7.1 Pin Configuration of All Ports

7.3.1 Port Data Register

GPADR -- Address: H'10004010

Bit	7	6	5	4	3	2	1	0
Bit Name	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GPBDR -- Address: H'10004012

Bit	7	6	5	4	3	2	1	0
Bit Name	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GPCDR -- Address: H'10004014

Bit	7	6	5	4	3	2	1	0
Bit Name	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GPDDR -- Address: H'10004016

Bit	7	6	5	4	3	2	1	0
Bit Name	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GPEDR -- Address: H'10004018

Bit	7	6	5	4	3	2	1	0
Bit Name	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The Port Data register **(GPxDR x: A,B,C,D,E)** is an 8-bit register. When the pin function is set to be a general output port, the value of the **PxnDT(x: A,B,C,D,E n: 7~0)** bit is directly output to its corresponding pin. When the pin function is set to be a general input port, the pin level status can be detected by reading the corresponding register bit. Please refer to **Table 7-3** on page 65 for the read/write operation in regards to its setting functions**.**

7.3.2 Port Control Register

GPACR -- Address: H'10004000

Bit	15	14	13	12	11	10	9	8
Bit Name	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0
Initial Value	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

GPBCR -- Address: H'10004002

Bit	15	14	13	12	11	10	9	8
Bit Name	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0
Initial Value	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

GPCCR -- Address: H'10004004

Bit	15	14	13	12	11	10	9	8
Bit Name	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0
Initial Value	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

GPDCR -- Address: H'10004006

Bit	15	14	13	12	11	10	9	8
Bit Name	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PB0MD0
Initial Value	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

GPECR -- Address: H'10004008

Bit	15	14	13	12	11	10	9	8
Bit Name	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5MD1	PE5MD0	PE4MD1	PE4MD0
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0
Initial Value	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

The register is used to control the functions of each I/O port pin. Control bits of MD0 and MD1 are defined in **Table 7-3.**

Table 7.3 Control Bits Definition of the Port x Control Register and Its Relevant READ/WRITE Operation of Port Data Register

PxnMD1	PxnMD0	Pin Status	READ	WRITE
0	0	Function 2	Pin status	Can write to GPxDR, but has no effect on pin status.
	1	Output	Pin Status	Value written to GPxDR is output to pin.
1	0	Input (Pull-up MOS on)	Pin status	Can write to GPxDR, but has no effect on pin status.
	1	Input (Pull-up MOS off)	Pin status	Can write to GPxDR, but has no effect on pin status.

7.3.3 Port Interrupt Control Register

This register is used to enable or disable to generate the interrupt request when an interrupt event is triggered on each I/O port pin. An interrupt request is generated when an interrupt event is triggered and its corresponding register bit is set to “1”. But the Interrupt request will not be generated if its corresponding control register bit is “0,” despite that the interrupt event is triggered. This register can independently select the trigger edge of the interrupt events on each I/O port pin.

GPAICR -- Address: H'10004020

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	PA7TS	PA6TS	PA5TS	PA4TS	PA3TS	PA2TS	PA1TS	PA0TS
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	PA7IM	PA6IM	PA5IM	PA4IM	PA3IM	PA2IM	PA1IM	PA0IM
Initial Value	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

GPBICR -- Address: H'10004022

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	PB7TS	PB6TS	PB5TS	PB4TS	PB3TS	PB2TS	PB1TS	PB0TS
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	PB7IM	PB6IM	PB5IM	PB4IM	PB3IM	PB2IM	PB1IM	PB0IM
Initial Value	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

GPCICR -- Address: H'10004024

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	PC7TS	PC6TS	PC5TS	PC4TS	PC3TS	PC2TS	PC1TS	PC0TS
Initial Value	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

GPCICR -- Address: H'10004024 (cont'd)

Bit	7	6	5	4	3	2	1	0
Bit Name	PC7IM	PC6IM	PC5IM	PC4IM	PC3IM	PC2IM	PC1IM	PC0IM
Initial Value	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

GPDICR -- Address: H'10004026

Bit	15	14	13	12	11	10	9	8
Bit Name	PD7TS	PD6TS	PD5TS	PD4TS	PD3TS	PD2TS	PD1TS	PD0TS
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PD7IM	PD6IM	PD5IM	PD4IM	PD3IM	PD2IM	PD1IM	PD0IM
Initial Value	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

GPEICR: Address: H'10004028

Bit	15	14	13	12	11	10	9	8
Bit Name	PE7TS	PE6TS	PE5TS	PE4TS	PE3TS	PE2TS	PE1TS	PE0TS
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PE7IM	PE6IM	PE5IM	PE4IM	PE3IM	PE2IM	PE1IM	PE0IM
Initial Value	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

PxnTS: Port x Bit n interrupt trigger select
= 0: select falling edge trigger
= 1: select rising edge trigger

PxnIM: Port x Bit n interrupt mask
= 0: interrupt is enabled
= 1: interrupt is masked

7.3.4 Port Interrupt Status Register

GPAISR -- Address: H'10004040

Bit	7	6	5	4	3	2	1	0
Bit Name	PA7ISR	PA6ISR	PA5ISR	PA4ISR	PA3ISR	PA2ISR	PA1ISR	PA0ISR
Initial Value	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

GPBISR -- Address: H'10004042

Bit	7	6	5	4	3	2	1	0
Bit Name	PB7ISR	PB6ISR	PB5ISR	PB4ISR	PB3ISR	PB2ISR	PB1ISR	PB0ISR
Initial Value	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

GPCISR -- Address: H'10004044

Bit	7	6	5	4	3	2	1	0
Bit Name	PC7ISR	PC6ISR	PC5ISR	PC4ISR	PC3ISR	PC2ISR	PC1ISR	PC0ISR
Initial Value	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

GPDISR-- Address: H'10004046

Bit	7	6	5	4	3	2	1	0
Bit Name	PD7ISR	PD6ISR	PD5ISR	PD4ISR	PD3ISR	PD2ISR	PD1ISR	PD0ISR
Initial Value	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

GPEISR -- Address: H'10004048

Bit	7	6	5	4	3	2	1	0
Bit Name	PE7ISR	PE6ISR	PE5ISR	PE4ISR	PE3ISR	PE2ISR	PE1ISR	PE0ISR
Initial Value	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

When an interrupt event occurs on an I/O port pin and its corresponding interrupt control register (GPXICR) bit is set to “1” (enabled), the corresponding interrupt status bit is read as “1”. Note that interrupt output is kept active till writing ‘1’ to the corresponding status bit. The status bit and interrupt output will be cleared after “1” is written to the status register.

Section 8 Interrupt Controller (INTC)

8.1 Overview

The Intelligent Peripheral Controller interrupts are issued from the modules of PS/2, PCMCIA, AFE, GPIO port, Timer, Keyboard Controller, IrDA , UART, PP, SCDI, USB, and ADC. The controller contains a register for the interrupt request status issued from each module.

After SH-4/SH7709 detects the interrupt, it reads the interrupt request register to see which module generates the interrupt, and then reads the interrupt request register in each module. As the controller provides the feature of gathering interrupts from all modules into one register, it will help to simplify the CPU interrupt processing.

8.1.1 Features

- All interrupts issued from the internal modules are gathered into one register
- Only one external interrupt output pin IRQ0# is used to request the interrupt service
- Interrupt request lines from each module are high active and level trigger signals
- The priority order of interrupt request lines is determined by software
- Each module provides an interrupt mask bit. A mask register, which is able to perform masking for each module interrupt, is also included

8.1.2 Block Diagram

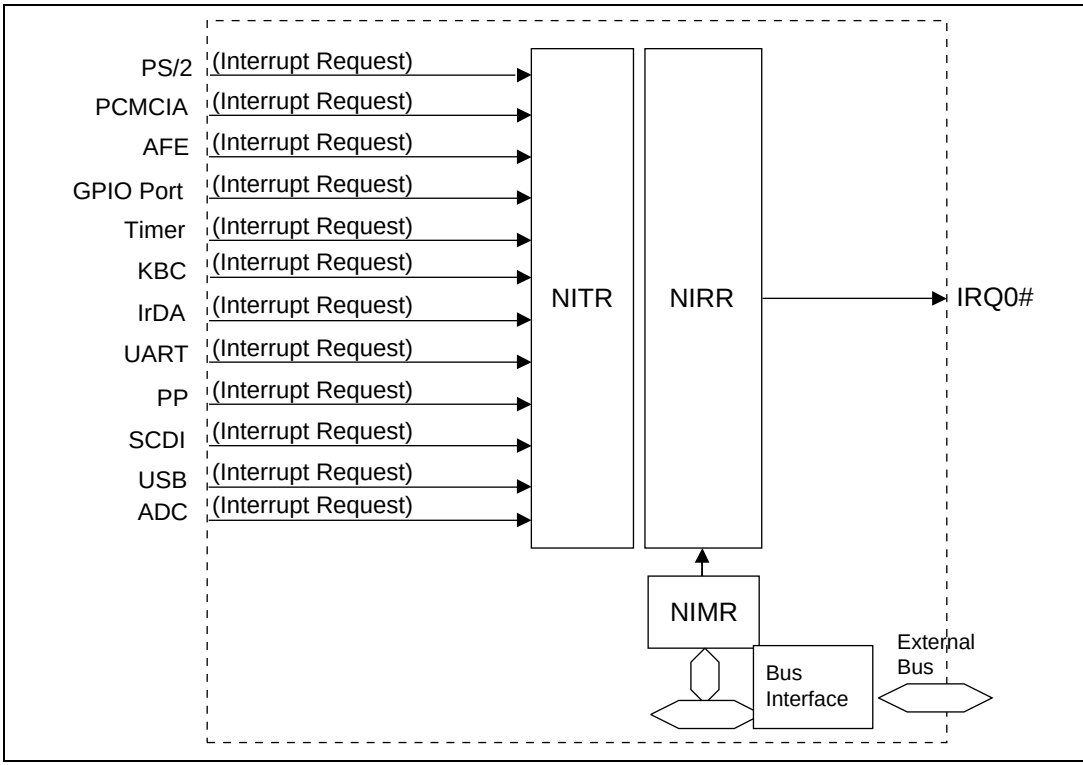


Figure 8.1 Block Diagram of the Interrupt Controller

8.1.3 Pin Configuration

Name	Abbr.	I/O	Description
Interrupt Request	IRQ0#	O	Interrupt output to SH-4/SH7709 from the Intelligent Peripheral Controller

8.1.4 Register Configuration

Name	Abbr.	R/W	Initial Value	Address	Access Size
Interrupt Request Register	NIRR	R/W	-	H'10005000	16
Interrupt Mask Register	NIMR	R/W	H'0000	H'10005002	16
Interrupt Trigger Mode Register	NITR	R/W	H'0000	H'10005004	16

8.2 Interrupt Sources

8.2.1 On-Chip Module Interrupt

Interrupt sources are derived from on-chip peripheral module Interrupts. Each interrupt provides a mask bit in each module listed below:

- PS/2 Keyboard
- PS/2 Mouse
- PCMCIA Controller (PCC)
- Analog Front End (AFE) Interface
- GPIO
- Timer
- Keyboard Controller (KBC)
- IrDA
- UART
- PP
- SCDI
- USB
- ADC

8.2.2 Interrupt Exception Processing and Priority

The priority order of the on-chip modules is determined by software. After detecting the interrupt request IRQ0# from the Intelligent Peripheral Controller, the CPU must read the NIRR (Interrupt Request Register) to check the interrupt sources. The interrupt sources will be recorded by the CPU. The CPU will then determine the priority order and execute the interrupt service based on the determined priority order. That is to say, the interrupt service will be executed for the interrupt requests in the order from the highest priority to the lowest.

After the highest priority interrupt service is decided by the CPU, the CPU will set mask bit of NIMR (Interrupt Mask Register) to those lower priority interrupts and set CPU priority level to the current serviced one.

One exception is that the CPU can still accept an incoming interrupt, which has higher priority than the one being serviced. After the CPU completes reading the interrupt request register in each module, the status of interrupt request for each module is cleared automatically, or by software.

8.3 NIRR: Interrupt Request Register

The NIRR, a 16-bit register, indicates interrupt requests from the internal modules of PS/2, PCMCIA, AFE, GPIO, Timer, KBC, IrDA, UART, PP, SCDI, USB, and ADC.

Address: H'10005000

Bit	15	14	13	12	11	10	9	8
Bit Name	PS2KBR	PCC0R	PCC1R	AFER	GPIOR	TMU0R	TMU1R	KBCR
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Bit Name	PS2MSR	IRDAR	UART0R	-	PPR	SCDIR	USBR	ADCR
Initial Value	-	-	-	-	-	-	-	-
R/W	R/W	R/W	R/W	-	R/O	R/O	R/O	R/O

Bit	Description	Default
15	**PS2 Keyboard interrupt request status (PS2KBR)** 1: The interrupt request is generated from PS2 Keyboard. 0: No interrupt requests are generated from PS2 Keyboard.	-
14	**PCMCIA0 interrupt request status (PCC0R)** This bit represents PCMCIA0 interrupt request status. 1: The interrupt request is generated from PCMCIA0. 0: No interrupt requests are generated from PCMCIA0.	-
13	**PCMCIA1 interrupt request status (PCC1R)** This bit represents PCMCIA1 interrupt request status. 1: The interrupt request is generated from PCMCIA1. 0: No interrupt requests are generated from PCMCIA1.	-
12	**AFE interrupt requests status (AFER)** This bit represents AFE interrupt requests status. 1: The interrupt request is generated from AFE, 0: No interrupt requests are generated from AFE.	-
11	**GPIO interrupt request status (GPIOR)** This bit represents GPIO interrupt request status. 1: The interrupt request is generated from GPIO. 0: No interrupt requests are generated from GPIO.	-
10	**TIMER 0 interrupt request status (TMU0R)** This bit represents TIMER 0 interrupt request status. 1: The interrupt request is generated from TIMER 0. 0: No interrupt requests are generated from TIMER 0.	-

NIRR: Interrupt Request Register [cont'd]

Bit	Description	Default
9	**TIMER 1 interrupt request status (TMU1R)** This bit represents TIMER 1 interrupt request status. 1: The interrupt request is generated from TIMER 1. 0: No interrupt requests are generated from TIMER 1.	-
8	**KBC interrupt request status (KBCR)** This bit represents KBC interrupt request status. 1: The interrupt request is generated from KBC. 0: No interrupt requests are generated from KBC.	-
7	**PS2 Mouse interrupt request status (PS2MSR)** 1: The interrupt request is generated from PS2 Mouse. 0: No interrupt requests are generated from PS2 Mouse.	-
6	**IrDA interrupt request status (IRDAR)** This bit represents IrDA interrupt request status. 1: The interrupt request is generated from IrDA. 0: No interrupt requests are generated from IrDA.	-
5	**UART interrupt request status (UARTR)** This bit represents UART0 interrupt request status. 1: The interrupt request is generated from UART0. 0: No interrupt requests are generated from UART0.	-
4	Reserved.	-
3	**Parallel port interrupt request status (PPR)** This bit represents Parallel port interrupt request status. 1: The interrupt request is generated from Parallel port. 0: No interrupt requests are generated from Parallel port.	-
2	**SCDI interrupt request status (SCDIR)** This bit represents SCDI interrupt request status. 1: The interrupt request is generated from SCDI. 0: No interrupt requests are generated from SCDI.	-
1	**USB interrupt request status (USBR)** This bit represents USB interrupt request status. 1: The interrupt request is generated from USB. 0: No interrupt requests are generated from USB.	-
0	**ADC interrupt request status (ADCR)** This bit represents ADC interrupt request status. 1: The interrupt request is generated from ADC. 0: No interrupt requests are generated from ADC.	-

8.4 NIMR: Interrupt Mask Register

The NIMR, a 16-bit register, serves to mask interrupt requests from the internal modules of PS/2, PCMCIA, AFE, GPIO, Timer, KBC, IrDA, UART, PP, SCDI, USB, and ADC. This register is initialized to H'0000 at RESET.

Address: H'10005002

Bit	15	14	13	12	11	10	9	8
Bit Name	PS2KBM	PCC0M	PCC1M	AFEM	GPIOM	TMU0M	TMU1M	KBCM
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PS2MS	IRDAM	UART0M	-	PPM	SCDIM	USBM	ADCM
Initial Value	0	0	0	-	0	0	0	0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

Bit	Description	Default
15	**PS2 Keyboard Interrupt Mask Control (PS2KBM)**	0
	This bit is used to control the mask option for PS2 Keyboard interrupt request.	
	1: The interrupt request from PS2 Keyboard is masked.	
	0: The interrupt request from PS2 Keyboard is not masked.	
14	**PCMCIA0 Interrupt Mask Control (PCC0M)**	0
	This bit is used to control the mask option for PCMCIA0 interrupt request.	
	1: The interrupt request from PCMCIA0 is masked.	
	0: The interrupt request from PCMCIA0 is not masked.	
13	**PCMCIA1 Interrupt Mask Control (PCC1M)**	0
	This bit is used to control the mask option for PCMCIA1 interrupt request.	
	1: The interrupt request from PCMCIA1 is masked.	
	0: The interrupt request from PCMCIA1 is not masked.	
12	**AFE Interrupt Mask Control (AFEM)**	0
	This bit is used to control the mask option for AFE interrupt request.	
	1: The interrupt request from AFE is masked.	
	0: The interrupt request from AFE is not masked.	
11	**GPIO Interrupt Mask Control (GPIOM)**	0
	This bit is used to control the mask option for GPIO interrupt request.	
	1: The interrupt request from GPIO is masked.	
	0: The interrupt request from GPIO is not masked.	

NIMR: Interrupt Mask Register [cont'd]

Bit	Description	Default
10	**TIMER 0 Interrupt Mask Control (TMU0M)** This bit is used to control the mask option for TIMER 0 interrupt request. 1: The interrupt request from TIMER 0 is masked. 0: The interrupt request from TIMER 0 is not masked.	0
9	**Timer 1 Interrupt Mask Control (TMU1M)** This bit is used to control the mask option for TIMER 1 interrupt request. 1: The interrupt request from TIMER 1 is masked. 0: The interrupt request from TIMER 1 is not masked.	0
8	**KBC Interrupt Mask Control (KBCM)** This bit is used to control the mask option for KBC interrupt request. 1: The interrupt request from KBC is masked. 0: The interrupt request from KBC is not masked.	0
7	**PS2 Mouse Interrupt Mask Control (PS2MSM)** This bit is used to control the mask option for PS2 Mouse interrupt request. 1: The interrupt request from PS2 Mouse is masked. 0: The interrupt request from PS2 Mouse is not masked.	0
6	**IrDA Interrupt Mask Control (IRDAM)** This bit is used to control the mask option for IrDA interrupt request. 1: The interrupt request from IrDA is masked. 0: The interrupt request from IrDA is not masked.	0
5	**UART Interrupt Mask Control (UARTM)** This bit is used to control the mask option for UART0 interrupt request. 1: The interrupt request from UART0 is masked. 0: The interrupt request from UART0 is not masked.	0
4	Reserved	0
3	**Parallel Port Interrupt Mask Control (PPM)** This bit is used to control the mask option for Parallel port interrupt request. 1: The interrupt request from Parallel port is masked. 0: The interrupt request from Parallel port is not masked.	0
2	**SCDI Interrupt Mask Control (SCDIM)** This bit is used to control the mask option for SCDI interrupt request. 1: The interrupt request from SCDI is masked. 0: The interrupt request from SCDI is not masked.	0
1	**USB Interrupt Mask Control (USBM)** This bit is used to control the mask option for USB interrupt request. 1: The interrupt request from USB is masked. 0: The interrupt request from USB is not masked.	0

NIMR: Interrupt Mask Register [cont'd]

Bit	Description	Default
0	**ADC Interrupt Mask Control (ADCM)**	0
	This bit is used to control the mask option for ADC interrupt request.	
	1: The interrupt request from ADC is masked.	
	0: The interrupt request from ADC is not masked.	

8.5 NITR: Interrupt Trigger Mode Register

The NITR, a 16-bit register, determines the trigger mode of the interrupt requests from the internal modules of PS/2, PCMCIA, AFE, GPIO, Timer, KBC, IrDA, UART, PP, SCDI, USB and ADC. This register is initialized to H'0000 at RESET.

Address: H'10005004

Bit	15	14	13	12	11	10	9	8
Bit Name	PS2KBT	PCC0T	PCC1T	AFET	GPIOT	TMU0T	TMU1T	KBCT
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	PS2MST	IRDAT	UARTT	-	PPT	SCDIT	USBT	ADCT
Initial Value	0	0	0	-	0	0	0	0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

Bit	Description	Default
15	**PS2 Keyboard Interrupt Trigger Mode (PS2KBT)**	0
	This bit is used to set the trigger mode for PS2 Keyboard interrupt request.	
	1: The interrupt request from PS2 Keyboard is in edge trigger mode.	
	0: The interrupt request from PS2 Keyboard is in level trigger mode.	
14	**PCMCIA0 Interrupt Trigger Mode (PCC0T)**	0
	This bit is used to set the trigger mode for PCMCIA0 interrupt request.	
	1: The interrupt request from PCMCIA0 is in edge trigger mode.	
	0: The interrupt request from PCMCIA0 is in level trigger mode.	
13	**PCMCIA1 Interrupt Trigger Mode (PCC1T)**	0
	This bit is used to set the trigger mode for PCMCIA1 interrupt request.	
	1: The interrupt request from PCMCIA1 is in edge trigger mode.	
	0: The interrupt request from PCMCIA1 is in level trigger mode.	

NITR: Interrupt Trigger Mode Register [cont'd]

Bit	Description	Default
12	**AFE Interrupt Trigger Mode (AFET)** This bit is used to set the trigger mode for AFE interrupt request. 1: The interrupt request from AFE is in edge trigger mode. 0: The interrupt request from AFE is in level trigger mode.	0
11	**GPIO Interrupt Trigger Mode (GPIOT)** This bit is used to set the trigger mode for GPIO interrupt request. 1: The interrupt request from GPIO is in edge trigger mode. 0: The interrupt request from GPIO is in level trigger mode.	0
10	**Timer 0 Interrupt Trigger Mode (TMU0T)** This bit is used to set the trigger mode for TIMER 0 interrupt request. 1: The interrupt request from TIMER 0 is in edge trigger mode. 0: The interrupt request from TIMER 0 is in level trigger mode.	0
9	**Timer 1 Interrupt Trigger Mode (TMU1T)** This bit is used to set the trigger mode for TIMER 1 interrupt request. 1: The interrupt request from TIMER 1 is in edge trigger mode. 0: The interrupt request from TIMER 1 is In level trigger mode.	0
8	**KBC Interrupt Trigger Mode (KBCT)** This bit is used to set the trigger mode for KBC interrupt request. 1: The interrupt request from KBC is in edge trigger mode. 0: The interrupt request from KBC is in level trigger mode.	0
7	**PS2 Mouse Interrupt Trigger Mode (PS2MST)** This bit is used to set the trigger mode for PS2 Mouse interrupt request. 1: The interrupt request from PS2 Mouse is in edge trigger mode. 0: The interrupt request from PS2 Mouse is in level trigger mode.	0
6	**IrDA Interrupt Trigger Mode (IRDAT)** This bit is used to set the trigger mode for IrDA interrupt request. 1: The interrupt request from IrDA is in edge trigger mode. 0: The interrupt request from IrDA is in level trigger mode.	0
5	**UART Interrupt Trigger Mode (UARTT)** This bit is used to set the trigger mode for UART interrupt request. 1: The interrupt request from UART is in edge trigger mode. 0: The interrupt request from UART is in level trigger mode.	0
4	Reserved	-
3	**Parallel Port Interrupt Trigger Mode (PPT)** This bit is used to set the trigger mode for Parallel port interrupt request. 1: The interrupt request from Parallel port is in edge trigger mode. 0: The interrupt request from Parallel port is in level trigger mode.	0

NITR: Interrupt Trigger Mode Register [cont'd]

Bit	Description	Default
2	**SCDI Interrupt Trigger Mode (SCDIT)** This bit is used to set the trigger mode for SCDI interrupt request. 1: The interrupt request from SCDI is in edge trigger mode. 0: The interrupt request from SCDI is in level trigger mode.	0
1	**USB Interrupt Trigger Mode (USBT)** This bit is used to set the trigger mode for USB interrupt request. 1: The interrupt request from USB is in edge trigger mode. 0: The interrupt request from USB is in level trigger mode.	0
0	**ADC Interrupt Trigger Mode (ADCT)** This bit is used to set the trigger mode for ADC interrupt request. 1: The interrupt request from ADC is in edge trigger mode. 0: The interrupt request from ADC is in level trigger mode.	0

Section 9 Timer

9.1 Overview

The timer in the HD64465 provides two channels of programmable counters. Interrupt requests are generated whenever the counter reaches zero. In order to reduce power consumption, clock pre-scaled circuit and STANDBY mode are added.

Two-channel pulse width modulation (PWM) functions are also provided for VR control of LCD. The PWM cycle is programmable by setting high pulse cycle and low pulse cycle.

9.1.1 Features

- Two channels programmable down-count timer
- 16-bit counter
- The counter is loaded with 16-bit constant registers and can be read or written to at any moment
- The counter constant is auto-reloaded when reaching zero
- Supports binary counting
- Supports START/STOP counting
- The input frequency of timer can be pre-scaled, the options are: 1, 1/4, 1/8, 1/16
- Generates interrupt output timer1r/timer0r to interrupt controller INTC when counter reaches zero
- Supports STANDBY mode, which is able to stop the clock operation of the timer
- Provides DMA request enable function, and DREQ1# is used (timer1)
- Provides A/D trigger signal ADTRIG# enable function (timer0)
- Provides timer output signal TMO1# and TMO0# options
- Two channels PWM
- 16-bit high pulse width counter and low pulse width counter for each channel
- Programmable PWM cycle
- Six clock scales (1, 1/2, 1/4, 1/8, 1/16, 1/32) with respect to CKIO are supported for pulse width counting

9.1.2 Block Diagram

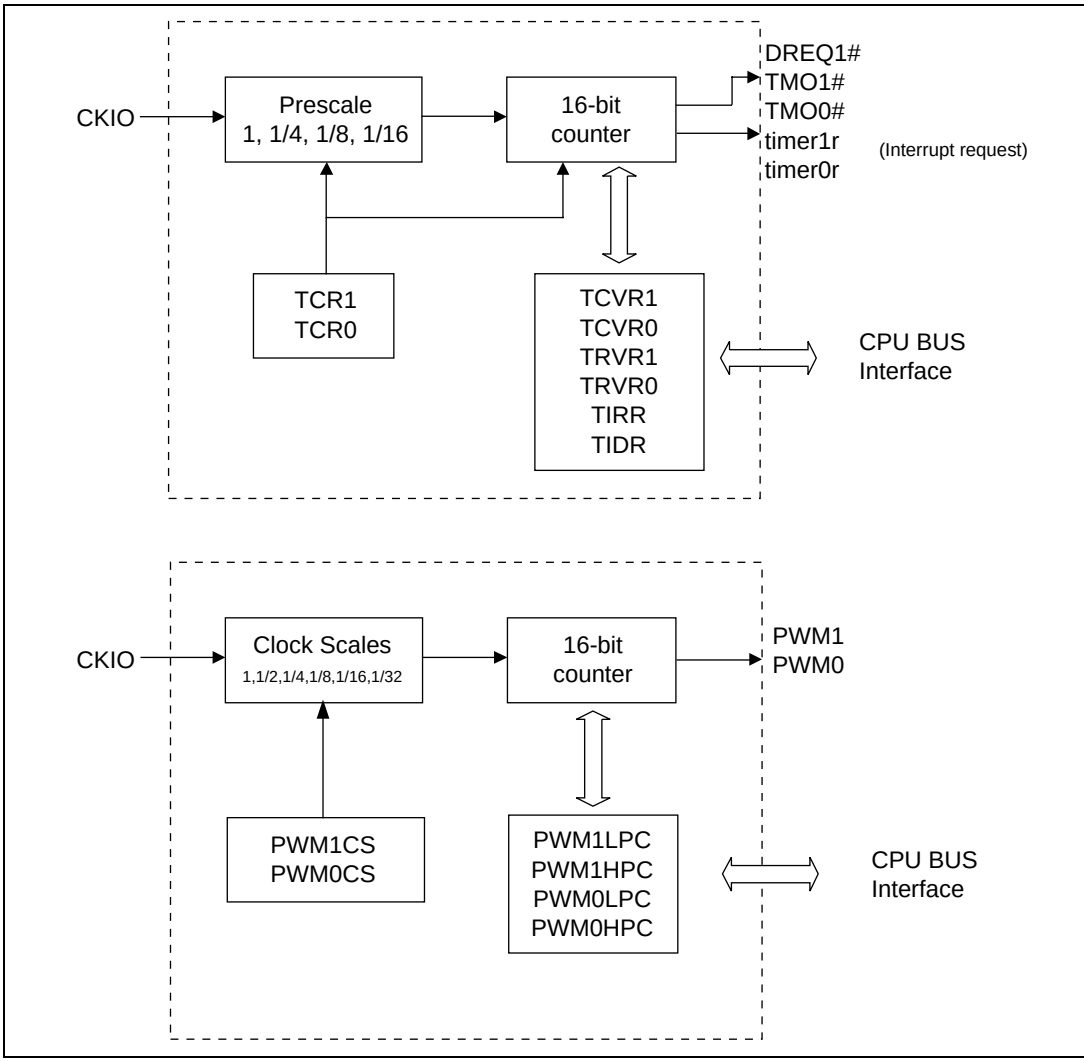


Figure 9.1 Block Diagram of Timer

9.1.3 Pin Configuration

Name	Abbr.	I/O	Description
Timer 1 output	TMO1#	O	Multiplexed with PB1/TMO1#. Timer output signal is enabled by bit 3 ETMO1 in Timer 1 Control register TCR1.
Timer 0 output	TMO0#	O	Multiplexed with PB0/TMO0#. Timer output signal is enabled by bit 3 ETMO0 in Timer 0 Control register TCR0.
DMA request	DREQ1#	O	DMA request.
PWM 1 output	PWM1	O	PWM channel 1 output
PWM 0 output	PWM0	O	PWM channel 0 output

9.1.4 Register Configuration

The timer contains seven registers listed in the table below:

Table 9.1 The Register List of Timer Module

Name	Address	Register Size	Access Size
Timer 1 constant value register (TCVR1)	H'10006000	16	16
Timer 0 constant value register (TCVR0)	H'10006002	16	16
Timer 1 read value register (TRVR1)	H'10006004	16	16
Timer 0 read value register (TRVR0)	H'10006006	16	16
Timer 1 control register (TCR1)	H'10006008	16	16
Timer 0 control register (TCR0)	H'1000600A	16	16
Timer interrupt request register (TIRR)	H'1000600C	16	16
Timer interrupt disable register (TIDR)	H'1000600E	16	16
PWM 1 clock scale register (PWM1CS)	H'10006010	16	16
PWM 1 low pulse width counter register (PWM1LPC)	H'10006012	16	16
PWM 1 high pulse width counter register (PWM1HPC)	H'10006014	16	16
PWM 0 clock scale register (PWM0CS)	H'10006018	16	16
PWM 0 low pulse width counter register (PWM0LPC)	H'1000601A	16	16
PWM 0 high pulse width counter register (PWM0HPC)	H'1000601C	16	16

9.2 Timer Register

9.2.1 TCVR1: Timer 1 Constant Value Register

The TCVR1, a 16-bit register, provides the counter constant for timer 1. The register is initialized to H’ FFFF at RESET.

Address: H'10006000

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	-
Initial Value	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

Bit	Description	Default
15 - 0	These bits are set as the timer 1 constant value.	H’FFFF

9.2.2 TCVR0: Timer 0 Constant Value Register

The TCVR0, a 16-bit register, provides the counter constant for the timer 0. The register is initialized to H’FFFF at RESET.

Address: H'10006002

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	-
Initial Value	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

Bit	Description	Default
15 - 0	These bits are set as the timer 0 constant value.	H’FFFF

9.2.3 TRVR1: Timer 1 Read Value Register

The TRVR1, a 16-bit register, serves to read the counting value in the timer 1. This register would be reset to H’0002 when the timer is stopped.

Address: H'10006004

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
15 - 0	The timer 1 counter value is read from these bits.	-

9.2.4 TRVR0: Timer 0 Read Value Register

The TRVR0, a 16-bit register, serves to read the counting value in timer 0. This register would be reset to H’0002 when the timer is stopped.

Address: H'10006006

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
15 - 0	The timer 0 counter value is read from these bits.	-

9.2.5 TCR1: Timer 1 Control Register

The TCR 1, a 16-bit register, is used to control the timer 1.

Address: H'10006008

Bit	15	14	13	12	11	10	9	8
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	reserved	EDMA	ETMO1	PST11	PST10	T1STP
Initial Value	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

Bit	Description	Default
15 - 5	Reserved.	0
4	This bit is used to enable the DMA request as timer 1 reaches zero. 1: Enable the DMA request 0: Disable the DMA request	0
3	This bit is used to enable the timer 1 output TMO1#. 1: Output DMA request to port TMO1# 0: Output DMA request to port DREQ1#	0
2, 1	These two bits are used for the pre-scale option for the timer 1. The input clock for timer 1 is selected by this option. 11: Timer 1 input clock is CKIO 10: Timer 1 input clock is CKIO/4 01: Timer 1 input clock is CKIO/8 00: Timer 1 input clock is CKIO/16	00
0	This bit is used to start or stop the counting of the timer 1. 1: Start timer 1 counting 0: Stop timer 1 counting	0

9.2.6 TCR0: Timer 0 Control Register

The TCR 0, a 16-bit register, is used to control the timer 0.

Address: H'1000600A

Bit	15	14	13	12	11	10	9	8
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	reserved	EADT	ETMO0	PST01	PST00	T0STP
Initial Value	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

Bit	Description	Default
15 - 5	Reserved.	0
4	This bit is used to enable the A/D converter trigger signal ADTRIG# as timer 0 reaches zero. 1: Enable the ADTRIG# signal 0: Disable the ADTRIG# signal	0
3	This bit is used to enable the timer 0 output TMO0#. 1: Output ADTRIG# to port TMO0# 0: Output ADTRIG# to internal ADC	0
2, 1	These two bits are used for the pre-scale option for the timer 0. The input clock for the timer 0 is selected by this option. 11: Timer 0 input clock is CKIO 10: Timer 0 input clock is CKIO/4 01: Timer 0 input clock is CKIO/8 00: Timer 0 input clock is CKIO/16	00
0	This bit is used to start or stop the counting of the timer 0. 1: Start timer 0 counting 0: Stop timer 0 counting	0

9.2.7 TIRR: Timer Interrupt Request Register

The TIRR, a 16-bit register, reflects the interrupt status from the timer 1 and the timer 0.

Address: H'1000600C

Bit	15	14	13	12	11	10	9	8
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	TMU1R	TMU0R
Initial Value	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

Bit	Description	Default
15 - 2	Reserved.	0
1	This bit reflects the interrupt service request from the timer 1. This bit is set when counting value of the timer 1 reaches zero, and it can be cleared by writing 0 to this bit. 1: There is an interrupt request from the timer 1. 0: There is no interrupt request from the timer 1.	0
0	This bit reflects the interrupt service request from timer 0. This bit is set when counting value of the timer 0 reaches zero, and it can be cleared by writing 0 to this bit. 1: There is an interrupt request from the timer 0. 0: There is no interrupt request from the timer 0.	0

9.2.8 TIDR*: Timer Interrupt Disable Register

The TIDR, a 16-bit register, is used to disable the interrupt request from the timer 1 and the timer 0.

Address: H'1000600E

Bit	15	14	13	12	11	10	9	8
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	TMU1D	TMU0D
Initial Value	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

Bit	Description	Default
15 - 2	Reserved.	0
1	This bit is used to disable the interrupt service request from the timer 1. 1: The interrupt request from the timer 1 is disabled. 0: The interrupt request from the timer 1 is enabled.	0
0	This bit is used to disable the interrupt service request from the timer 0. 1: The interrupt request from the timer 0 is disabled. 0: The interrupt request from the timer 0 is enabled.	0

* Note: TIDR is the same as TIMR in HD64461, the naming is changed to avoid confusion.

9.2.9 PWM1CS: PWM 1 Clock Scale Register

The PWM1CS, a 16-bit register, is used to control the clock scale for PWM 1 width counting. The register is initialized to H’ 0000 at RESET.

Address: H'10006010

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	P1CS5	P1CS4	P1CS3	P1CS2	P1CS1	P1CS0
Initial Value	-	-	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
15 - 6	Reserved.	-
5 - 0	**PWM 1 Counting Clock Scale (P1CS[5:0])** These six bits provide six clock scale modes for PWM 1 width counting. Only one bit will be active at the same time, and the lower bit has higher priority while several bits are on at the same time. B’000000 : PWM 1 function is disabled. B’000001 : PWM 1 is counting with clock scale CKIO. B’000010 : PWM 1 is counting with clock scale CKIO/2. B’000100 : PWM 1 is counting with clock scale CKIO/4. B’001000 : PWM 1 is counting with clock scale CKIO/8. B’010000 : PWM 1 is counting with clock scale CKIO/16. B’100000 : PWM 1 is counting with clock scale CKIO/32.	

9.2.10 PWM1LPC: PWM 1 Low Pulse Width Counter Register

The PWM1LPC, a 16-bit register, defines the low pulse width of PWM. The register is initialized to H' FFFF at RESET.

Address: H'10006012

Bit	15	14	13	12	11	10	9	8
Bit Name	P1LC15	P1LC14	P1LC13	P1LC12	P1LC11	P1LC10	P1LC9	P1LC8
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	P1LC7	P1LC6	P1LC5	P1LC4	P1LC3	P1LC2	P1LC1	P1LC0
Initial Value	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

Bit	Description	Default
15 - 0	**PWM 1 Low Pulse Counter Value (P1LC[15:0])** These bits define the low pulse width of PWM 1.	H'FFFF

9.2.11 PWM1HPC: PWM 1 High Pulse Width Counter Register

The PWM1HPC, a 16-bit register, defines the high pulse width of PWM 1. The register is initialized to H’ FFFF at RESET.

Address: H'10006014

Bit	15	14	13	12	11	10	9	8
Bit Name	P1HC15	P1HC14	P1HC13	P1HC12	P1HC11	P1HC10	P1HC9	P1HC8
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	P1HC7	P1HC6	P1HC5	P1HC4	P1HC3	P1HC2	P1HC1	P1HC0
Initial Value	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

Bit	Description	Default
15 - 0	**PWM 1 High Pulse Counter Value (P1HC[15:0])** These bits define the high pulse width of PWM 1.	H’FFFF

9.2.12 PWM0CS: PWM 0 Clock Scale Register

The PWM0CS, a 16-bit register, is used to control the clock scale for PWM 0 width counting. The register is initialized to H’ 0000 at RESET.

Address: H'10006018

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	P0CS5	P0CS4	P0CS3	P0CS2	P0CS1	P0CS0
Initial Value	-	-	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
15 - 6	Reserved.	-
5 - 0	**PWM 0 Counting Clock Scale (P0CS[5:0])** These six bits provide six clock scale modes for PWM 0 width counting. Only one bit will be active at the same time, and the lower bit has higher priority while several bits are on at the same time. B’000000 : PWM 0 function is disabled. B’000001 : PWM 0 is counting with clock scale CKIO. B’000010 : PWM 0 is counting with clock scale CKIO/2. B’000100 : PWM 0 is counting with clock scale CKIO/4. B’001000 : PWM 0 is counting with clock scale CKIO/8. B’010000 : PWM 0 is counting with clock scale CKIO/16. B’100000 : PWM 0 is counting with clock scale CKIO/32.	

9.2.13 PWM0LPC: PWM 0 Low Pulse Width Counter Register

The PWM0LPC, a 16-bit register, defines the low pulse width of PWM 0. The register is initialized to H’ FFFF at RESET.

Address: H'1000601A

Bit	15	14	13	12	11	10	9	8
Bit Name	P0LC15	P0LC14	P0LC13	P0LC12	P0LC11	P0LC10	P0LC9	P0LC8
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	P0LC7	P0LC6	P0LC5	P0LC4	P0LC3	P0LC2	P0LC1	P0LC0
Initial Value	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

Bit	Description	Default
15 - 0	**PWM 0 Low Pulse Counter Value (P0LC[15:0])** These bits define the low pulse width of PWM 0.	H’FFFF

9.2.14 PWM0HPC: PWM 0 High Pulse Width Counter Register

The PWM0HPC, a 16-bit register, defines the high pulse width of PWM 0. The register is initialized to H’ FFFF at RESET.

Address: H'1000601C

Bit	15	14	13	12	11	10	9	8
Bit Name	P0HC15	P0HC14	P0HC13	P0HC12	P0HC11	P0HC10	P0HC9	P0HC8
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	P0HC7	P0HC6	P0HC5	P0HC4	P0HC3	P0HC2	P0HC1	P0HC0
Initial Value	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

Bit	Description	Default
15 - 0	**PWM 0 High Pulse Counter Value (P0HC[15:0])** These bits define the high pulse width of PWM 0.	H’FFFF

9.3 Special Register Programming Sequence

In general, before setting the timer control registers TCR1/TCR0 to start counting, the timer constant registers TCVR1/TCVR0 should be set first. Otherwise, the selected timer will start counting from H’FFFF.

9.4 Interrupt Timing

The interrupt request is triggered when the counter reaches zero, and the request is cleared by writing 0 to the interrupt request bit TMU1R/TMU0R in TIRR. In case that the disable bit TMU1D/TMU0D in TIDR is set, the interrupt request will be disabled. The interrupt timing diagrams for each pre-scale clock are shown below:

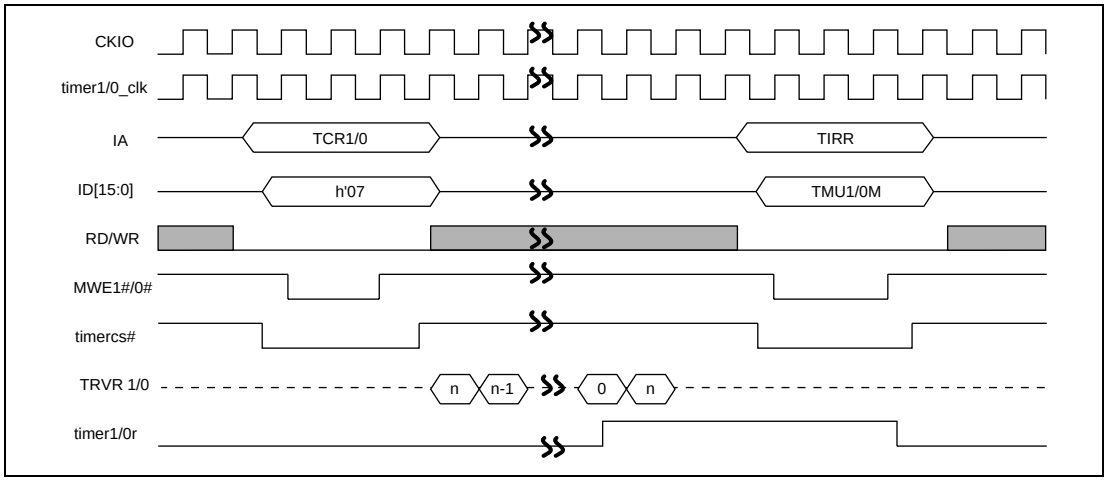


Figure 9.2 Interrupt Request Timer1/0r Timing Diagram in Case Prescale *1, Timer1/0_clk=CKIO

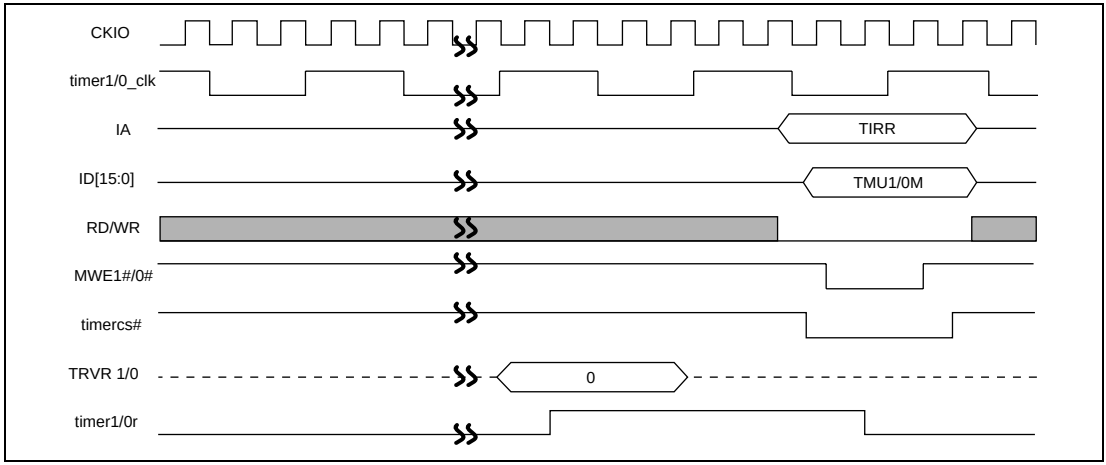


Figure 9.3 Interrupt Request Timer1/0r Timing Diagram in Case Prescale*1/4, Timer1/0_clk = CKIO/4

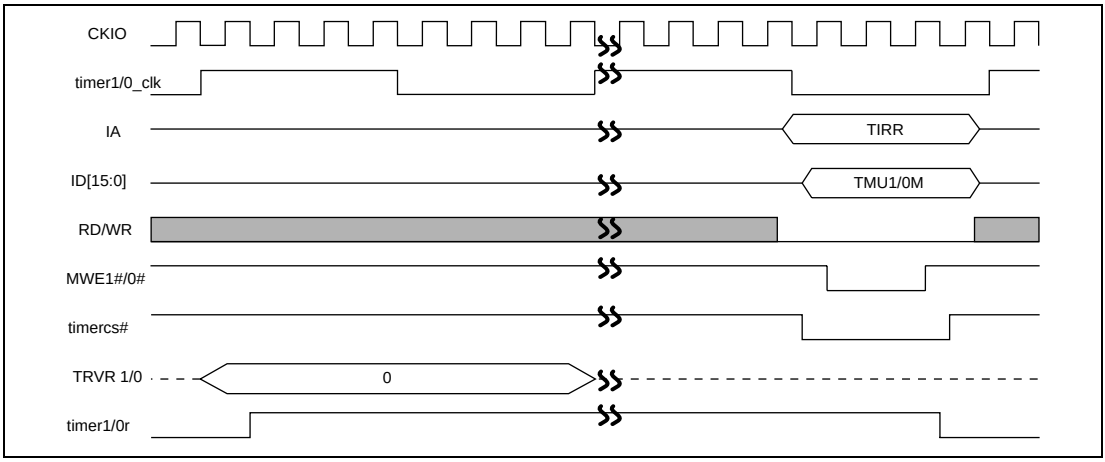


Figure 9.4 Interrupt Request Timer1/0r Timing Diagram in Case Prescale*1/8, Timer1/0_clk = CKIO/8

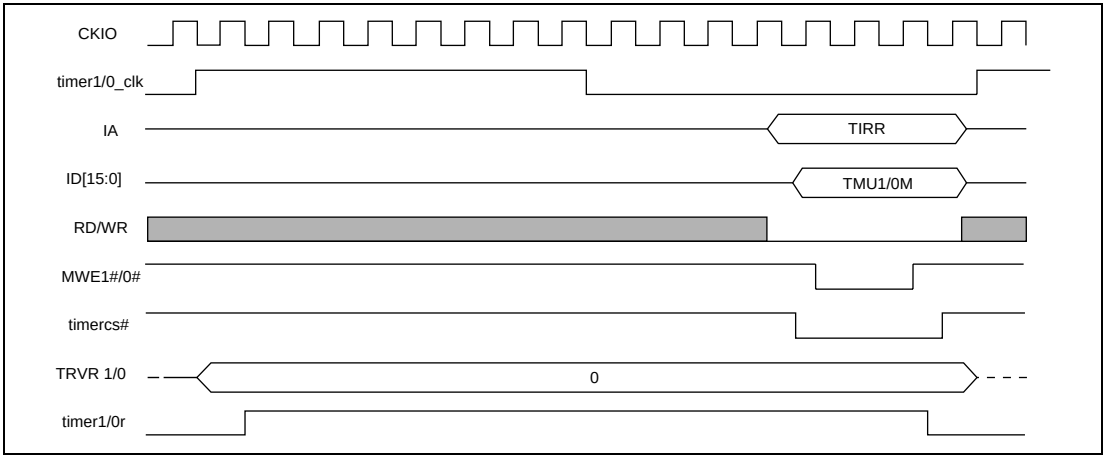


Figure 9.5 Interrupt Request Timer1/0r Timing Diagram in Case Prescale*1/16, Timer1/0_clk = CKIO/16

9.5 A/D Trigger Signal ADTRIG#

The A/D trigger signal ADTRIG# is enabled by the control bit EADT in TCR0. The pulse width of the A/D trigger signal is sixteen cycles of CKIO to trigger the A to D converter. Therefore, the timer constant register TCVR0 should be properly set to ensure the normal operations of the A/D trigger signal.

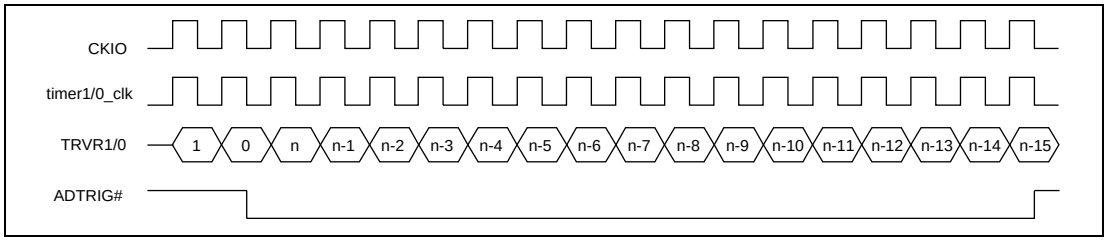


Figure 9.6 A/D Trigger Signal ADTRIG# Timing Diagram in Case Prescale 1, Timer0_clk=CKIO

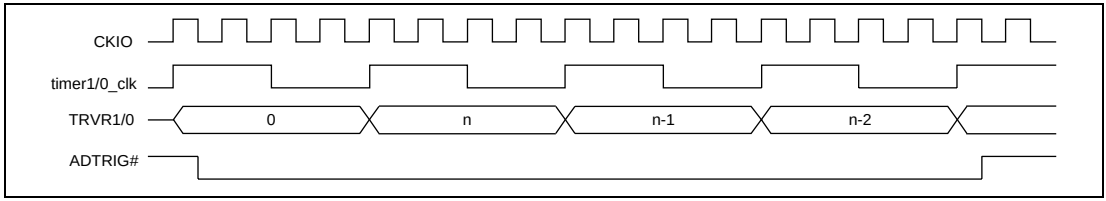


Figure 9.7 A/D Trigger Signal ADTRIG# Timing Diagram in Case Prescale 1/4, Timer0_clk=CKIO/4

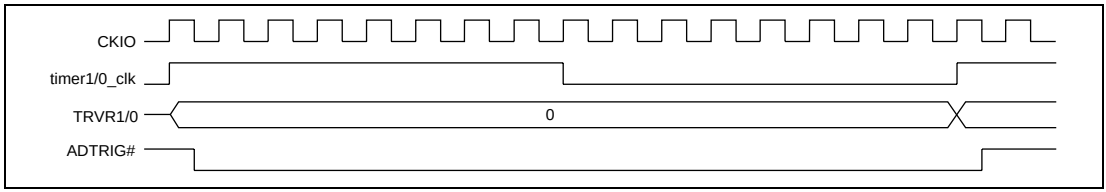


Figure 9.8 A/D Trigger Signal ADTRIG# Timing Diagram in Case Prescale 1/8, Timer0_clk=CKIO/8

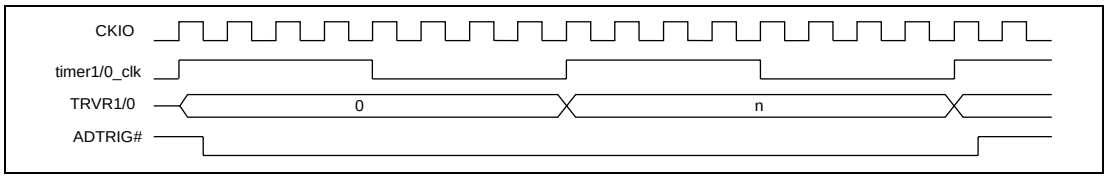


Figure 9.9 A/D Trigger Signal ADTRIG# Timing Diagram in Case Prescale 1/16, Timer0_clk=CKIO/16

* Note: The transient delays are ignored in the figures above.

9.6 DMA Request Enable Function

The Timer supports the DMA Request Enable Function using DREQ1# by setting the control bit EDMA in TCR1. When the DMA Request Function is enabled, the DMA cycle will operate in Burst Mode and perform the edge detection on DREQ1#; that is to say, the timer will issue a DMA request periodically by pulling the DREQ1# to low, and by releasing the DREQ1# when the first cycle DRAK1 output is detected. The DMA transfer is completed after it finishes transferring all the data transfers set in DMATCR.

9.7 PWM Operation

The PWM signal is generated while the proper value is written to the clock scale register (PWMCS) to activate the PWM generation function and decide the PWM width counting clock. Based on the decide PWM width counting clock, the PWM operating cycles and duty cycles can be determined by programming the low pulse width counting register (PWMLPC) and high pulse width counting register (PWMHPC).

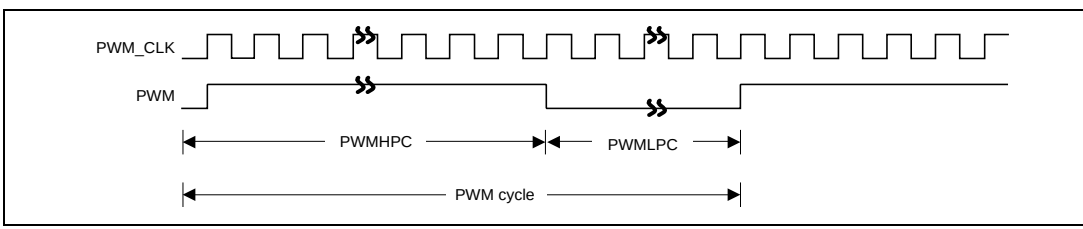


Figure 9.10 PWM Signals

Section 10 PC Card Controller (PCC)

10.1 Overview

The PC Card Controller (PCC) controls the internal buffers, interrupts, and PCMCIA-defined ports of the PC card interfaces, which should be connected to the Intelligent Peripheral Controller (IPC). The PCC enables two slots of the PC cards that are compliant with the specifications PCMCIA Rev. 2.1/JEIDA Version 4.2, thereby ensuring a smooth connection to the IPC.

10.2 Features

- Two slots of the PC card interfaces can be simultaneously controlled
- Supports IC memory card interface, and I/O and Memory card interface, which function as PC card interfaces and are connected to physical areas 5 and 6. Area 6, which is referred as channel 0 (PCC0), supports both of the IC memory card interface, and I/O & Memory card interface. Area 5, which is referred as channel1 (PCC1), supports both of the IC memory card interface, and I/O & Memory card interface.
- Able to switch to an Attribute memory or a Common memory, or an I/O space via the CPU addresses.
- Provides a segment bit (an address bit for the PC card) for the Common memory; therefore, a 64-MB space of full PCMCIA specifications can be accessed.
- Provides a flexible power switch interface to either TPS2206 or MIC2563.

10.3 Register Configuration

Table 10.1 PC Card Controller Registers

Register Name	Symbol	Read/ Write	Initial Value	Address	Register Size (Access Size)
PCC0 interface status register	PCC0ISR	R	-	H'10002000	8 bits (16 bits)
PCC0 general control register	PCC0GCR	R/W	H'00	H'10002002	8 bits (16 bits)
PCC0 card status change register	PCC0CSCR	R/W	H'00	H'10002004	8 bits (16 bits)
PCC0 card status change interrupt enable register	PCC0 CSCIER	R/W	H'00	H'10002006	8 bits (16 bits)
PCC0 software control register	PCC0SCR	R/W	H'00	H'10002008	8 bits (16 bits)
PCC serial power switch control register	PCCPSR	R/W	H'00	H'1000200A	8 bits (16 bits)

Table 10.1 PC Card Controller Registers [cont'd]

Register Name	Symbol	Read/ Write	Initial Value	Address	Register Size
PCC1 interface status register	PCC1ISR	R	-	H'10002010	8 bits (16 bits)
PCC1 general control register	PCC1GCR	R/W	H'00	H'10002012	8 bits (16 bits)
PCC1 card status change register	PCC1CSCR	R/W	H'00	H'10002014	8 bits (16 bits)
PCC1 card status change interrupt enable register	PCC1 CSCIER	R/W	H'00	H'10002016	8 bits (16 bits)
PCC1 software control register	PCC1SCR	R/W	H'00	H'10002018	8 bits (16 bits)

10.4 Register Description

10.4.1 PCC0 Interface Status Register (PCC0ISR)

Bit	7	6	5	4	3	2	1	0
Bit Name	P0READY /IREQ0	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2 /SPKR0	P0BVD1/ STSCHG0
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Note -: The initial value is determined by the status of the PC card.

The PCC0 interface status register (PCC0ISR) is an 8-bit READ-only register, which reads the status of the PC card connected to PCC0. PCC0ISR is subject to the PC card status.

Bit	Description	Default
7	**PCC0 Ready/IREQ0 Pin Status (P0READY/IREQ0)** If this bit is high: Indicates that the value of pin PCC0RDY is 1 when the PC card connected to PCC0 is the IC memory card interface. Indicates that the value of pin IREQ is 1 when the I/O and Memory card interface is the PC card connected to PCC0. If this bit is low: Indicates that the value of pin PCC0RDY is 0 when the PC card connected to PCC0 is the IC memory card interface. Indicates that the value of pin IREQ is 0 when the I/O and Memory card interface is the PC card connected to PCC0.	-
6	**PCC0 Memory Write Protect (P0MWP)** If this bit is high: Indicates that the value of pin PCC0WP is 1 when the Memory card interface is the PC card connected to PCC0. If this bit is low: Indicates that the value of pin PCC0WP is 0 when the PC card connected to PCC0 is the IC memory card interface. The value of bit 6 is always 0 when the I/O and Memory card interface is the PC card connected to PCC0.	-

PCC0 Interface Status Register (PCC0ISR) [cont'd]

Bit	Description	Default
5	**PCC0VS2# Pin Status (P0VS2)** If this bit is high: Indicates that the value of pin PCC0VS2# of the PC card connected to PCC0 is 1. If this bit is low: Indicates that the value of pin PCC0VS2# of the PC card connected to PCC0 is 0.	-
4	**PCC0VS1# Pin Status (P0VS1)** If this bit is high: Indicates that the value of pin PCC0VS1# of the PC card connected to PCC0 is 1. If this bit is low: Indicates that the value of pin PCC0VS1# in the PC card connected to PCC0 is 0.	-
3	**PCC0CD2# Pin Status (P0CD2)** If this bit is high: Indicates that the value of pin PCC0CD2# in the PC card connected to PCC0 is 1. If this bit is low: Indicates that the value of pin PCC0CD2# in the PC card connected to PCC0 is 0.	-
2	**PCC0CD1# Pin Status (P0CD1)** If this bit is high: Indicates that the value of pin PCC0CD1# in the PC card connected to PCC0 is 1. If this bit is low: Indicates that the value of pin PCC0CD1# in the PC card connected to PCC0 is 0.	-
IC Memory Interface		
1, 0	**PCC0BVD2/SPKR0 Pin Status (P0BVD2/SPKR0), PCC0BVD1/STSCHG0# Pin Status (P0BVD1/STSCHG0)** 11: Indicates that the battery voltage state of the PC card connected to PCC0 is normal. (Battery Good). 01: Indicates that the battery must be replaced although data integrity is guaranteed for the PC card connected to PCC0 (Battery Warning). 10: Indicates that the battery voltage state is abnormal and data integrity is not guaranteed for the PC card connected to PCC0 (Battery Dead). 00: Indicates that the battery voltage state is abnormal and data integrity is not guaranteed for the PC card connected to PCC0 (Battery Dead).	-
I/O and Memory Interface		
1	1: Indicates that the value of pin SPKR on the PC card connected to PCC0 is 1. 0: Indicates that the value of pin SPKR on the PC card connected to PCC0 is 0.	-
0	1: Indicates that the value of pin STSCHG in the PC card connected to PCC0 is 1. 0: Indicates that the value of pin STSCHG in the PC card connected to PCC0 is 0.	-

10.4.2 PCC0 General Control Register (PCC0GCR)

Bit	7	6	5	4	3	2	1	0
Bit Name	P0DRV	P0PCCR	P0PCCT	P0VCC0	P0MMOD	P0PA25	P0PA24	P0REG
Initial Value	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

The PCC0 general control register (PCC0GCR) is an 8-bit READ/WRITE register. It controls reset, address pins PCC0A25 and PCC0A24, and pin PCC0REG, and sets the PC card type for the PC card connected to PCC0. PCC0GCR is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	**PCC0 Output Drive (P0DRV)**	0
	If this bit is high: enables the output interfaces to PCC0.	
	If this bit is low: tri-states the output interfaces to PCC0.	
6	**PCC0 PC Card Reset (P0PCCR)**	0
	If this bit is high: Sets a high level to reset pin PCC0RESET for the PC card connected to PCC0.	
	If this bit is low: Sets a low level to reset pin PCC0RESET for the PC card connected to PCC0. (Initial value)	
5	**PCC0 PC Card Type (P0PCCT)**	0
	If this bit is high: Handles the PC card connected to PCC0 as the I/O and Memory card interface.	
	If this bit is low: Handles the PC card connected to PCC0 as the IC memory card interface. (Initial value)	
4	**PCC0 VCC Select 0 (P0VCC0)**	0
	If this bit is high: Sets a voltage control pin VCC0SEL0 to high level.	
	If this bit is low: Sets a voltage control pin VCC0SEL0 to low level.	

Bit	Description	Default
3	**PCC0 Memory Mode (P0MMOD)** If this bit is high: (continuous 16MB area mode)	0

Pin A25	Pin A24	Bit2	Bit1	Bit0	PCMCIA Access Range
0	0	don't care		0	16MB Attribute memory Area
0	1	Memory Bank Select		1	16MB of 64MB Common memory Area
1	0	don't care		don't care	16-MB I/O Space

If this bit is low: (continuous 32MB area mode)

Pin A25	Pin A24	Bit2	Bit1	Bit0	PCMCIA Access Range
0	x	don't care	don't care	0	32MB Attribute memory Area
0	x	Memory Bank Select	don't care	1	32MB of 64MB Common memory Area
1	x	don't care	don't care	don't care	32MB I/O Space

2	**PCC0A25 Pin Setting (P0PA25)** If this bit is high: When the Common memory space is accessed for the PC card connected to PCC0, 1 is output to pin PCC0A25. If this bit is low: When the Common memory space is accessed for the PC card connected to PCC0, 0 is output to pin PCC0A25. (Initial value)	0
1	**PCC0A24 Pin Setting(P0PA24)** If this bit is high: When bit P0MMOD is 1 and the Common memory space is accessed for the PC card connected to PCC0, 1 is output to pin PCC0A24. If this bit is low: When bit P0MMOD is 1 and the Common memory space is accessed for the PC card connected to PCC0, 0 is output to pin PCC0A24. (Initial value)	0
0	**PCC0REG# Pin Setting(P0REG)** If this bit is high: When bit 3 is 0 and the PC card connected to PCC0 is accessed, 1 is output to pin PCC0REG#. If this bit is low: When bit 3 is 0 and the PC card connected to PCC0 is accessed, 0 is output to pin PCC0REG#. (Initial value)	0

10.4.3 PCC0 Card Status Change Register (PCC0CSCR)

Bit	7	6	5	4	3	2	1	0
Bit Name	P0SCDI	PSW_SEL	P0IREQ	P0SC	P0CDC	P0RC	P0BW	P0BD
Initial Value	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/W

The PCC0 Card Status Change Register (PCC0CSCR) is an 8-bit READ/WRITE register. PCC0CSCR is set to 1 by each interrupt factor of the PC card connected to PCC0 (only bits 7 and 6 can be set to 1 as required). Bit 6 is for power switch selection. PCC0CSCR is initialized at power-up reset or in hardware-based STANDBY mode, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	**PCC0 Software Card Detect Interrupt (P0SCDI)** If this bit is high: Software card detection interrupt occurs for the PC card connected to PCC0. If this bit is low: No software card detection interrupt occurs for the PC card connected to PCC0. (Initial value) If bit 3 (card detection enable) in the PCC0 card status change interrupt enable register (PCC0CSCIER) is set to 1, a software card detect change interrupt can be generated by writing 1 to this bit. If bit 3 (card detection enable) in the PCC0CSCIER is reset to 0, interrupts will not occur even when writing “1” to this bit.	0
6	**Power Switch Select (PSWSEL)** Select power switch interface 0: MIC2563 1: TPS2206 serial power switch	0
5	**PCC0 Interrupt Request (P0IREQ)** If this bit is high: Indicates that an interrupt request for the IREQ pin in the PC card has occurred when the PC card is in the I/O and Memory card interface. If this bit is low: Indicates no interrupt request for the IREQ pin in the PC card when the PC card is in the I/O and Memory card interface. (Initial value) This bit can be reset to 0 only in the pulse mode. To reset, write “0” to this bit. This bit is not changed if 1 is written. In the level mode, bit 5 is a READ-only bit that reflects the state of IREQ pin (if the IREQ pin is in the low level, 1 is read). In this bit, 0 is always read in the IC memory card interface.	1

PCC0 Card Status Change Register (PCC0CSCR) [cont'd]

Bit	Description	Default
4	**PCC0 STSCHG Change (P0SC)**	0
	If this bit is high: Indicates that the STSCHG pin in the PC card is changed from 1 to 0 when the PC card is in the I/O and Memory card interface.	
	If this bit is low: Indicates that the value of pin STSCHG on the PC card remains unchanged when the PC card is in the I/O and Memory card interface. (Initial value)	
	When the STSCHG pin is changed from 1 to 0, this bit is set to 1. To reset, write "0" to this bit when it is set to "1". This bit is not changed if 1 is written. In this bit, 0 is always read in the IC memory card interface.	
3	**PCC0 Card Detect Change (P0CDC)**	0
	If this bit is high: Indicates that CD1 and CD2 in the PC card are changed.	
	If this bit is low: Indicates that CD1 and CD2 in the PC card are not changed. (Initial value)	
	Write 0 to bit 3 in order to reset this bit to 0. This bit is not changed if 1 is written.	
2	**PCC0 Ready Change (P0RC)**	0
	If this bit is high: Indicates that pin READY in the PC card is changed from 0 to 1 when the PC card is in the IC memory card interface.	
	If this bit is low: Indicates that pin READY in the PC card is not changed when the PC card is in the IC memory card interface. (Initial value)	
	WRITE 0 to bit 2 in order to reset this bit to 0. This bit is not changed if 1 is written. In this bit, 0 is always read in the I/O and Memory card interface.	
1	**PCC0 Battery Warning (P0BW)**	0
	If this bit is high: Indicates that pins BVD2 and BVD1 in the PC card are in the battery warning state that the battery must be replaced although the data integrity is guaranteed when the PC card is in the IC memory card interface.	
	If this bit is low: Indicates that pins BVD2 and BVD1 in the PC card are not in the battery warning state when the PC card is in the IC memory card interface. (Initial value)	
	This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 2 in order to reset this bit to 0. This bit is not changed if 1 is written. In this bit, 0 is always read in the I/O and Memory card interface.	
0	**PCC0 Battery Dead (P0BD)**	0
	If this bit is high: Indicates that pins BVD2 and BVD1 in the PC card are in the state that he battery must be replaced since the data integrity is not guaranteed when the PC card is in the IC memory card interface.	
	If this bit is low: Indicates that pins BVD2 and BVD1 in the PC card are not in the state that he battery must be replaced since the data integrity is not guaranteed when the PC card is in the IC memory card interface. (Initial value)	
	This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 2 in order to reset this bit to 0. This bit is not changed if 1 is written. In this bit, 0 is always read in the I/O and Memory card interface.	

10.4.4 PCC0 Card Status Change Interrupt Enable Register (PCC0CSCIER)

Bit	7	6	5	4	3	2	1	0
Bit Name	P0CRE	P0IREQE1	P0IREQE0	P0SCE	P0CDE	P0RE	P0BWE	P0BDE
Initial Value	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

The PCC0 card status change interrupt enable register (PCC0CSCIER) is an 8-bit READ and WRITE register. PCC0CSCIER is capable of setting a valid or invalid interrupt for each interrupt factor of the PC card connected to PCC0. When register PCC0CSCIER is set to 1, the interrupt is valid, and invalid when the register is set to 0. PCC0CSCR can be initialized by power-up reset. PCC0CSCIER is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	**PCC0 Change Reset Enable (P0CRE)**	0
	If this bit is high: The general control register (PCC0GCR) and software control register (PCC0SCR) in the PCC0 are initialized when a PC card connection change is detected in PCC0.	
	If this bit is low: The general control register (PCC0GCR) and software control register (PCC0SCR) in the PCC0 are not initialized even when a PC card change is detected in PCC0. (Initial value)	
6, 5	**PCC0 Interrupt Request Enable 1 (P0IREQE1)**	0
	PCC0 Interrupt Request Enable 0 (P0IREQE0)	
	00: Any kind of IREQ interrupt request signal is not accepted for the PC card connected to PCC0. Bit 5 in the status change register (PCC0CSCR) functions as a READ-only bit, and can indicate the status of the inversion signal of the IREQ pin. (Initial value)	
	01: The level-mode IREQ interrupt request signal is accepted for the PC card connected to PCC0. In the level mode, an interrupt occurs when level 0 of the signal input from the IREQ pin is detected.	
	10: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to PCC0. In the pulse mode, an interrupt occurs when a falling edge from 1 to 0 of the signal input from the IREQ pin is detected.	
	11: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to PCC0. In the pulse mode, an interrupt occurs when a rising edge from 0 to 1 of the signal input from the IREQ pin is detected.	
4	**PCC0 STSCHG Change Interrupt Enable (P0SCE)**	0
	If this bit is high: An interrupt occurs for the PC card connected to PCC0 when the value of the PCC0BVD1 pin (STSCHG) is changed from 1 to 0.	
	If this bit is low: No interrupt occurs for the PC card connected to PCC0 regardless of the value of the PCC0BVD1 pin. (STSCHG) (Initial value)	

PCC0 Card Status Change Interrupt Enable Register (PCC0CSCIER) [cont'd]

Bit	Description	Default
3	**PCC0 Card Detect Change Interrupt Enable (P0CDE)** If this bit is high: An interrupt occurs for the PC card connected to PCC0 when the values of the PCC0CD1# and PCC0CD2# pins are changed. If this bit is low: No interrupt occurs for the PC card connected to PCC0 regardless of the values of the PCC0CD1# and PCC0CD2# pins. (Initial value)	0
2	**PCC0 Ready Change Interrupt Enable (P0RE)** If this bit is high: An interrupt occurs for the PC card connected to PCC0 when the value of pin PCC0RDY is changed from 0 to 1. If this bit is low: No interrupt occurs for the PC card connected to PCC0 regardless of the value of pin PCC0RDY. (Initial value)	0
1	**PCC0 Battery Warning Interrupt Enable (P0BWE)** If this bit is high: An interrupt occurs when pins PCC0BVD2 and PCC0BVD1 are in the state that "the battery must be replaced although the data integrity is guaranteed". If this bit is low: No interrupt occurs when pins PCC0BVD2 and PCC0BVD1 are in the state that "the battery must be replaced although the data integrity is guaranteed". (Initial value)	0
0	**PCC0 Battery Dead Interrupt Enable (P0BDE)** If this bit is high: An interrupt occurs when pins PCC0BVD2 and PCC0BVD1 are in the state that "the battery must be replaced since the data integrity is not guaranteed". If this bit is low: No interrupt occurs when the value of pins PCC0BVD2 and PCC0BVD1 are in the state that "the battery must be replaced since the data integrity is not guaranteed". (Initial value)	0

10.4.5 PCC0 Software Control Register (PCC0SCR)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	$\overline{\text{SHDN}}$	P0VPP1	P0VPP0	P0VCC1	P0SWP
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

The PCC0 Software Control Register (PCC0CSCR) is an 8-bit READ/WRITE register. The way it controls pin VCC0SEL1 resembles to that of pin P0VCC0 in the PCC0GCR register does to pin VCC0SEL0. Bits 2 and 3 serve as the VPP power selection input to MIC2563, and bit 4 is the Shut-Down bit of the TPS2206. This register is also responsible for PC Card memory window write protect. PCC0SCR is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7 - 5	Reserved	0
4	D8 (SHDN) bit of the TPS2206	0
3	**PCC0 VPP Select 1 (P0VPP1)** If this bit is high: set the voltage control pin P0VPP1 as high level If this bit is low: set the voltage control pin P0VPP1 as low level	0
2	**PCC0 VPP Select 0 (P0VPP0)** If the bit is high: set the voltage control pin P0VPP0 as high level If this bit is low: set the voltage control pin P0VPP0 as low level	0
1	**PCC0 VCC Select 1 (P0VCC1)** If this bit is high: Sets the voltage control pin VCC0SEL1 as high level. If this bit is low: Sets the voltage control pin VCC0SEL1 as low level.	0
0	**PCC0 Software Write Protect (P0SWP)** If this bit is high: Enables write protect. Write operations to the PC Card yield no response. If this bit is low: Disables write protect. Write operations to the PC Card are allowed.	0

10.4.6 PCC Serial Power Switch Control Register (PCCPSR)

Bit	7	6	5	4	3	2	1	0
Bit Name	$\overline{\text{B\_VCC5}}$	$\overline{\text{B\_VCC3}}$	B_VPP_VCC	B_VPP_PGM	$\overline{\text{A\_VCC3}}$	$\overline{\text{A\_VCC5}}$	A_VPP_VCC	A_VPP_PGM
Initial Value	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

The PCC0PSR Serial Power Switch Control Register (P0PSR), an 8-bit READ/WRITE register, serves as the bit0-bit7 of the serial power switch TPS2206.

P0PSR is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	D7(B_VCC5) bit of the TPS2206	0
6	D6(B_VCC3) bit of the TPS2206	0
5	D5(B_VPP_VCC) bit of the TPS2206	0
4	D4(B_VPP_PGM) bit of the TPS2206	0
3	D3(A_VCC3) bit of the TPS2206	0
2	D2(A_VCC5) bit of the TPS2206	0
1	D1(A_VPP_VCC) bit of the TPS2206	0
0	D0(A_VPP_PGM) bit of the TPS2206	0

10.4.7 PCC1 Interface Status Register (PCC1ISR)

Bit	7	6	5	4	3	2	1	0
Bit Name	P1READY/IREQ1	P1MWP	P1VS2	P1VS1	P1CD2	P1CD1	P1BVD2/SPKR1	P1BVD1/STSCHG1
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Note -: The initial value is determined by the status of the PC card.

The PCC1 interface status register (PCC1ISR) is an 8-bit READ-only register, which reads the status of the PC card connected to PCC1. PCC1ISR is subject to the PC card status.

PCC1 Interface Status Register (PCC1ISR) [cont'd]

Bit	Description	Default
7	**PCC1 Ready/IREQ1 Pin Status (P1READY/IREQ1)** If this bit is high: Indicates that the value of pin PCC1RDY is 1 when the PC card connected to PCC1 is the IC memory card interface. Indicates that the value of pin IREQ is 1 when the I/O and Memory card interface is the PC card connected to PCC1. If this bit is low: Indicates that the value of pin PCC1RDY is 0 when the PC card connected to PCC1 is the IC memory card interface. Indicates that the value of pin IREQ is 0 when the I/O and Memory card interface is the PC card connected to PCC1.	-
6	**PCC1 Memory Write Protect (P1MWP)** If this bit is high: Indicates that the value of pin PCC1WP is 1 when the Memory card interface is the PC card connected to PCC1. If this bit is low: Indicates that the value of pin PCC1WP is 0 when the PC card connected to PCC1 is the IC memory card interface. The value of bit 6 is always 0 when the I/O and Memory card interface is the PC card connected to PCC1.	-
5	**PCC1VS2# Pin Status (P1VS2)** If this bit is high: Indicates that the value of pin PCC1VS2# of the PC card connected to PCC1 is 1. If this bit is low: Indicates that the value of pin PCC1VS2# of the PC card connected to PCC1 is 0.	-
4	**PCC1VS1# Pin Status (P1VS1)** If this bit is high: Indicates that the value of pin PCC1VS1# of the PC card connected to PCC1 is 1. If this bit is low: Indicates that the value of pin PCC1VS1# in the PC card connected to PCC1 is 0.	-
3	**PCC1CD2# Pin Status (P1CD2)** If this bit is high: Indicates that the value of pin PCC1CD2# in the PC card connected to PCC1 is 1. If this bit is low: Indicates that the value of pin PCC1CD2# in the PC card connected to PCC1 is 0.	-
2	**PCC1CD1# Pin Status (P1CD1)** If this bit is high: Indicates that the value of pin PCC1CD1# in the PC card connected to PCC1 is 1. If this bit is low: Indicates that the value of pin PCC1CD1# in the PC card connected to PCC1 is 0.	-
IC Memory Interface		
1 - 0	**PCC1BVD2/SPKR1 Pin Status (P1BVD2/SPKR1)** **PCC1BVD1/STSCHG1# Pin Status (P1BVD1/STSCHG1)** 11: Indicates that the battery voltage state of the PC card connected to PCC1 is normal. (Battery Good). 01: Indicates that the battery must be replaced although data integrity is guaranteed for the PC card connected to PCC1 (Battery Warning). 10: Indicates that the battery voltage state is abnormal and data integrity is not guaranteed for the PC card connected to PCC1 (Battery Dead). 00: Indicates that the battery voltage state is abnormal and data integrity is not guaranteed for the PC card connected to PCC1 (Battery Dead).	-

PCC1 Interface Status Register (PCC1ISR) [cont'd]

Bit	Description	Default
I/O and Card Interface		
1	1: Indicates that the value of pin SPKR on the PC card connected to PCC1 is 1.	-
	0: Indicates that the value of pin SPKR on the PC card connected to PCC1 is 0.	
0	1: Indicates that the value of pin STSCHG in the PC card connected to PCC1 is 1.	-
	0: Indicates that the value of pin STSCHG in the PC card connected to PCC1 is 0.	

10.4.8 PCC1 General Control Register (PCC1GCR)

Bit	7	6	5	4	3	2	1	0
Bit Name	P1DRV	P1PCCR	P1PCCT	P1VCC0	P1MMOD	P1PA25	P1PA24	P1REG
Initial Value	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

The PCC1 general control register (PCC1GCR) is an 8-bit READ/WRITE register. It controls reset, address pins PCC1A25 and PCC1A24, and pin PCC1REG, and sets the PC card type for the PC card connected to PCC1. PCC1GCR is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	**PCC1 Output Drive (P1DRV)**	0
	If this bit is high: enables the output interfaces to PCC1.	
	If this bit is low: tri-states the output interfaces to PCC1.	
6	**PCC1 PC Card Reset (P1PCCR)**	0
	If this bit is high: Sets a high level to reset pin PCC1RESET for the PC card connected to PCC1.	
	If this bit is low: Sets a low level to reset pin PCC1RESET for the PC card connected to PCC1. (Initial value)	
5	**PCC1 PC Card Type (P1PCCT)**	0
	If this bit is high: Handles the PC card connected to PCC1 as the I/O and Memory card interface.	
	If this bit is low: Handles the PC card connected to PCC1 as the IC memory card interface. (Initial value)	
4	**PCC1 VCC Select 0 (P1VCC0)**	0
	If this bit is high: Sets a voltage control pin VCC1SEL0 to high level.	
	If this bit is low: Sets a voltage control pin VCC1SEL0 to low level.	

Bit	Description	Default
3	**PCC1 Memory Mode (P1MMOD)** If this bit is high: (continuous 16MB area mode)	0

Pin A25	Pin A24	bit 2	bit 1	bit 0	PCMCIA Access Range
0	0	don't care		don't care	16MB Attribute memory Area
0	1	Memory Bank Select		don't care	16MB of 64MB Common memory Area
1	0	don't care		don't care	16-MB I/O Space

If this bit is low: (continuous 32MB area mode)

Pin A25	Pin A24	bit 2	bit 1	bit 0	PCMCIA Access Range
0	x	don't care	don't care	0	32MB Attribute memory Area
0	x	Memory Bank Select	don't care	1	32MB of 64MB Common memory Area
1	x	don't care	don't care	don't care	32MB I/O Space

2	**PCC1A25 Pin Setting (P1PA25)** If this bit is high: When the Common memory space is accessed for the PC card connected to PCC1, 1 is output to pin PCC1A25. If this bit is low: When the Common memory space is accessed for the PC card connected to PCC1, 0 is output to pin PCC1A25. (Initial value)	0
1	**PCC1A24 Pin Setting(P1PA24)** If this bit is high: When bit P1MMOD is 1 and the Common memory space is accessed for the PC card connected to PCC1, 1 is output to pin PCC1A24. If this bit is low: When bit P1MMOD is 1 and the Common memory space is accessed for the PC card connected to PCC1, 0 is output to pin PCC1A24. (Initial value)	0
0	**PCC1REG# Pin Setting(P1REG)** If this bit is high: When bit 3 is 0 and the PC card connected to PCC1 is accessed, 1 is output to pin PCC1REG#. If this bit is low: When bit 3 is 0 and the PC card connected to PCC1 is accessed, 0 is output to pin PCC1REG#. (Initial value)	0

10.4.9 PCC1 Card Status Change Register (PCC1CSCR)

Bit	7	6	5	4	3	2	1	0
Bit Name	P1SCDI	-	P1IREQ	P1SC	P1CDC	P1RC	P1BW	P1BD
Initial Value	0	-	0	0	0	0	0	0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W

The PCC1 Card Status Change Register (PCC1CSCR) is an 8-bit READ/WRITE register. PCC1CSCR is set to 1 by each interrupt factor of the PC card connected to PCC1 (only bit 7 and 6 can be set to 1 as required). Bit 6 is for power switch selection. PCC1CSCR is initialized at power-up reset or in hardware-based STANDBY mode, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	**PCC1 Software Card Detect Interrupt (P1SCDI)** If this bit is high: Software card detection interrupt occurs for the PC card connected to PCC1. If this bit is low: No software card detection interrupt occurs for the PC card connected to PCC1. (Initial value) If bit 3 (card detection enable) in the PCC1 card status change interrupt enable register (PCC1CSCIER) is set to 1, a software card detect change interrupt can be generated by writing 1 to this bit. If bit 3 (card detection enable) in the PCC1CSCIER is reset to 0, interrupts will not occur even when writing “1” to this bit.	0
6	Reserved	-
5	**PCC1 Interrupt Request (P1IREQ)** If this bit is high: Indicates that an interrupt request for the IREQ pin in the PC card has occurred when the PC card is in the I/O and Memory card interface. If this bit is low: Indicates no interrupt requests occur for the IREQ pin in the PC card when the PC card is in the I/O and Memory card interface. (Initial value) This bit can be reset to 0 only in the pulse mode. To reset, write “0” to this bit. This bit is not changed if 1 is written. In the level mode, bit 5 is a READ-only bit which reflects the state of IREQ pin(if the IREQ pin is in the low level, 1 is read). In this bit, 0 is always read in the IC memory card interface.	0
4	**PCC1 STSCHG Change (P1SC)** If this bit is high: Indicates that the STSCHG pin in the PC card is changed from 1 to 0 when the PC card is in the I/O and Memory card interface. If this bit is low: Indicates that the value of pin STSCHG on the PC card remains unchanged when the PC card is in the I/O and Memory card interface. (Initial value) When the STSCHG pin is changed from 1 to 0, this bit is set to 1. To reset, write “0” to this bit when it is set to “1”. This bit is not changed if 1 is written. In this bit, 0 is always read in the IC memory card interface.	0

PCC1 Card Status Change Register (PCC1CSCR) [cont'd]

Bit	Description	Default
3	**PCC1 Card Detect Change (P1CDC)**	0
	If this bit is high: Indicates that CD1 and CD2 in the PC card are changed.	
	If this bit is low: Indicates that CD1 and CD2 in the PC card are not changed. (Initial value)	
	Write 0 to bit 3 in order to reset this bit to 0. This bit is not changed if 1 is written.	
2	**PCC1 Ready Change (P1RC)**	0
	If this bit is high: Indicates that pin READY in the PC card is changed from 0 to 1 when the PC card is in the IC memory card interface.	
	If this bit is low: Indicates that pin READY in the PC card is not changed when the PC card is in the IC memory card interface. (Initial value)	
	Write 0 to bit 2 in order to reset this bit to 0. This bit is not changed if 1 is written. In this bit, 0 is always read in the I/O and Memory card interface.	
1	**PCC1 Battery Warning (P1BW)**	0
	If this bit is high: Indicates that pins BVD2 and BVD1 in the PC card are in the battery warning state that the battery must be replaced although the data integrity is guaranteed when the PC card is in the IC memory card interface.	
	If this bit is low: Indicates that pins BVD2 and BVD1 in the PC card are not in the battery warning state when the PC card is in the IC memory card interface. (Initial value)	
	This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 2 in order to reset this bit to 0. This bit is not changed if 1 is written. In this bit, 0 is always read in the I/O and Memory card interface.	
0	**PCC1 Battery Dead (P1BD)**	0
	If this bit is high: Indicates that pins BVD2 and BVD1 in the PC card are in the state that the battery must be replaced since the data integrity is not guaranteed when the PC card is in the IC memory card interface.	
	If this bit is low: Indicates that pins BVD2 and BVD1 in the PC card are not in the state that the battery must be replaced since the data integrity is not guaranteed when the PC card is in the IC memory card interface. (Initial value)	
	This bit is updated when the BVD2 and BVD1 pins are changed. Write 0 to bit 2 in order to reset this bit to 0. This bit is not changed if 1 is written. In this bit, 0 is always read in the I/O and Memory card interface.	

10.4.10 PCC1 Card Status Change Interrupt Enable Register (PCC1CSCIER)

Bit	7	6	5	4	3	2	1	0
Bit Name	P1CRE	P1IREQE1	P1IREQE0	P1SCE	P1CDE	P1RE	P1BWE	P1BDE
Initial Value	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

The PCC1 card status change interrupt enable register (PCC1CSCIER) is an 8-bit READ and WRITE register. PCC1CSCIER is capable of setting a valid or invalid interrupt for each interrupt factor of the PC card connected to PCC1. When register PCC1CSCIER is set to 1, the interrupt is valid, and invalid when the register is set to 0. PCC1CSCR can be initialized by power-up reset. PCC1CSCIER is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7	**PCC1 Change Reset Enable (P1CRE)** If this bit is high: The general control register (PCC1GCR) and software control register (PCC1SCR) in the PCC1 are initialized when a PC card connection change is detected in PCC1. If this bit is low: The general control register (PCC1GCR) and software control register (PCC1SCR) in the PCC1 are not initialized even when a PC card change is detected in PCC1. (Initial value)	0
6, 5	**PCC1 Interrupt Request Enable 1 (P1IREQE1)** **PCC1 Interrupt Request Enable 0 (P1IREQE0)** 00: Any kind of IREQ interrupt request signal is not accepted for the PC card connected to PCC1. Bit 5 in the status change register (PCC1CSCR) functions as a READ-only bit, and indicates the status of the inversion signal of the IREQ pin. (Initial value) 01: The level-mode IREQ interrupt request signal is accepted for the PC card connected to PCC1. In the level mode, an interrupt occurs when level 0 of the signal input from the IREQ pin is detected. 10: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to PCC1. In the pulse mode, an interrupt occurs when a falling edge from 1 to 0 of the signal input from the IREQ pin is detected. 11: The pulse-mode IREQ interrupt request signal is accepted for the PC card connected to PCC1. In the pulse mode, an interrupt occurs when a rising edge from 0 to 1 of the signal input from the IREQ pin is detected.	0
4	**PCC1 STSCHG Change Interrupt Enable (P1SCE)** If this bit is high: An interrupt occurs for the PC card connected to PCC1 when the value of the PCC1BVD1 pin (STSCHG) is changed from 1 to 0. If this bit is low: No interrupt occurs for the PC card connected to PCC1 regardless of the value of the PCC1BVD1 pin. (STSCHG) (Initial value)	0
3	**PCC1 Card Detect Change Interrupt Enable (P1CDE)** If this bit is high: An interrupt occurs for the PC card connected to PCC1 when the values of the PCC1CD1# and PCC1CD2# pins are changed. If this bit is low: No interrupt occurs for the PC card connected to PCC1 regardless of the values of the PCC1CD1# and PCC1CD2# pins. (Initial value)	0

PCC1 Card Status Change Interrupt Enable Register (PCC1CSCIER) [cont'd]

Bit	Description	Default
2	**PCC1 Ready Change Interrupt Enable (P1RE)** If this bit is high: An interrupt occurs for the PC card connected to PCC1 when the value of pin PCC1RDY is changed from 0 to 1. If this bit is low: No interrupt occurs for the PC card connected to PCC1 regardless of the value of pin PCC1RDY. (Initial value)	0
1	**PCC1 Battery Warning Interrupt Enable (P1BWE)** If this bit is high: An interrupt occurs when pins PCC1BVD2 and PCC1BVD1 are in the state that "the battery must be replaced although the data integrity is guaranteed". If this bit is low: No interrupt occurs when pins PCC1BVD2 and PCC1BVD1 are in the state that the battery must be replaced although the data integrity is guaranteed". (Initial value)	0
0	**PCC1 Battery Dead Interrupt Enable (P1BDE)** If this bit is high: An interrupt occurs when pins PCC1BVD2 and PCC1BVD1 are in the state that "the battery must be replaced since the data integrity is not guaranteed". If this bit is low: No interrupt occurs when the value of pins PCC1BVD2 and PCC1BVD1 are in the state that "the battery must be replaced since the data integrity is not guaranteed". (Initial value)	0

10.4.11 PCC1 Software Control Register (PCC1SCR)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	P1VPP1	P1VPP0	P1VCC1	P1SWP
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

The PCC1 Software Control Register (PCC1CSCR) is an 8-bit READ/WRITE register. The way it controls pin VCC1SEL1 resembles to that of pin P1VCC1 in the PCC1GCR register does to pin VCC1SEL0. Bits 2 and 3 serve as the VPP power selection input to MIC2563. This register is also responsible for PC Card memory window write protect. PCC1SCR is initialized at power-up reset, and holds its value at software reset or in software-based STANDBY mode.

Bit	Description	Default
7 - 4	Reserved	-
3	**PCC1 VPP Select 1 (P1VPP1)** If this bit is high: sets the voltage control pin P1VPP1 as high level If this bit is low: sets the voltage control pin P1VPP1 as low level	0
2	**PCC1 VPP Select 0 (P1VPP0)** If the bit is high: sets the voltage control pin P1VPP0 as high level If this bit is low: sets the voltage control pin P1VPP0 as low level	0
1	**PCC1 VCC Select 1 (P1VCC1)** If this bit is high: Sets the voltage control pin VCC1SEL1 as high level. If this bit is low: Sets the voltage control pin VCC1SEL1 as low level.	0
0	**PCC1 Software Write Protect (P1SWP)** If this bit is high: Enables write protect. Write operations to the PC Card yield no response. If this bit is low: Disables write protect. Write operations to the PC Card are allowed.	0

Section 11 FIR Module

11.1 Overview

The FIR module is fully IrDA1.1 compliant, and supports the serial infrared link at 288Kbit, 576Kbit, 1.152Mbit and 4Mbit. The three lower speeds are complied with Synchronous Data Link Control (SDLC) protocol, a packet with start/stop flags delimiting a data packet encoded by zero-insertion. The 4Mbit protocol uses an optical preamble and start/stop flags to delimit the 4 Pulse Position Modulated (4 PPM) packet data.

11.1.1 Features

- Supports HPSIR or ASKIR infrared interface
- Supports MIR and FIR
- Provides DMA channel mode for FIR, and DREQ0# is used

11.1.2 Functional Block Diagram of FIR

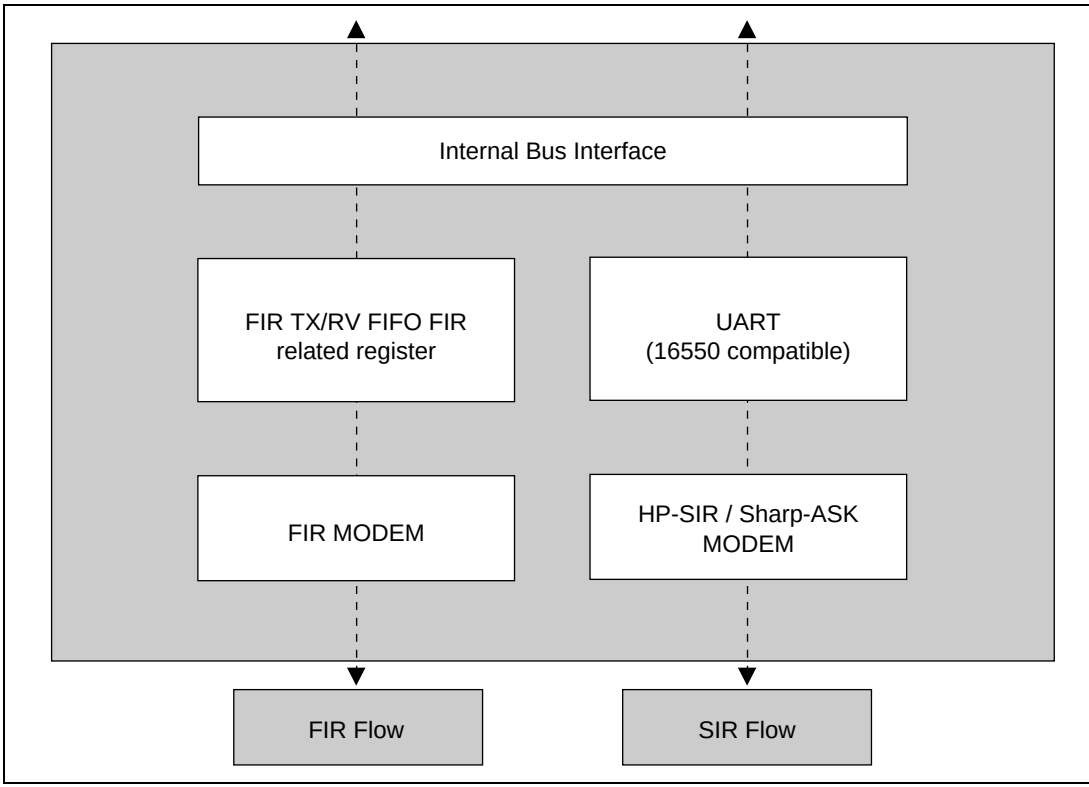


Figure 11.1 Functional Block Diagram of FIR

The FIR module contains five blocks. The Internal Bus Interface block implements the logic between FIR core and internal bus. The FIR TX/RV FIFO and FIR related register block implements data flow control from host to FIR MODEM. It contains transmitting FIFO, receiving FIFO and some registers to manage the FIFO operations. The related registers will be described in the later section. The FIR MODEM block contains the parallel data to/from serial data logic. This block also implements encoding the serial data to 4M mode bit stream. The UART block is designed to be compatible with 16550. This block is for parallel to/from serial data transfer. The function of baud rate generation is also featured in this block. HP-SIR and Sharp-ASK MODEM block is encoding and decoding from/to serial data to/from SIR data bit stream.

To use SIR or ASK transfer protocol, the data flow is from host to UART, and to HP-SIR MODEM:

1. Host sets **SIR Register (ISIRR)** to HP-SIR MODEM (see FIR Controller registers description in section 11.2.2 on the next page.)
2. Host writes the data to the UART for transmission. (host reads data from UART for data receiving)
3. UART converts parallel data into serial data for transmission. (UART converts serial data into parallel data for data receiving)
4. HP-SIR and SHARP-ASK MODEM implements encoding serial data into bit stream for transmission (decoding bit stream into serial data for data receiving)

For FIR transfer protocol, the data flow is from host to FIR TX/RV FIFO, and to FIR MODEM:

1. Host sets **SIR Register (ISIRR)** to FIR-MODEM (see FIR Controller registers description in section11.2.2 on the next page.)
2. Host writes data into TX FIFO for transmission. (host reads data from RX FIFO for data receiving)
3. FIR MODEM converts parallel data into serial data and implements encoding into bit stream for transmission. (decoding the bit stream from optical medium and translates from serial data to parallel data for data receiving)

11.2 FIR Controller Register Description

11.2.1 UART Register of FIR Portion

The register has the same definition as that of UART, except that the address spaces are different.

Register	DLAB*	Address	READ	WRITE
Data	0	H'10007000	IrRBR (Receiver Buffer Register)	IrTBR (Transmitter Buffer Register)
Control	0	H'10007002	IrIER (Interrupt Enable Register)	IrIER
	x	H'10007004	IrIIR (Interrupt Identification Register)	IrFCR (FIFO Control Register)
	x	H'10007006	IrLCR (Line Control Register)	IrLCR
	x	H'10007008	IrMCR (Modem Control Register)	IrMCR
	1	H'10007000	IrDLL (Divisor Latch LSB)	IrDLL
	1	H'10007002	IrDLM (Divisor Latch MSB)	IrDLM
Status	x	H'1000700A	IrLSR (Line Status Register)	IrLSR
	x	H'1000700C	IrMSR (Modem Status Register)	IrMSR
	x	H'1000700E	IrSCR (Scratch Pad Register)	IrSCR

Note: DLAB* is bit 7 of the Line Control register (IrLCR).

11.2.2 FIR Controller Register

Table 11.1 Summary of FIR Controller Registers

Bank	Address	READ	WRITE
0	H'10007100	IMSTCR (Master Control Register)	IMSTCR (Master Control Register)
0	H'10007102	IMSTSR (Master Status Register)	IMISCR (Miscellaneous Control Register)
0	H'10007104	IRFR (Rx FIFO Register)	ITFR (Tx FIFO Register)
0	H'10007106	ITC1R (Tx Control 1 Register)	ITC1R (Tx Control 1 Register)
0	H'10007108	ITC2R (Tx Control 2 Register)	ITC2R (Tx Control 2 Register)
0	H'1000710A	ITSR (Tx Status Register)	
0	H'1000710C	IRCR (Rx control Register)	IRCR (Rx Control Register)
0	H'1000710E	IRSR (Rx Status Register)	IRSTCR (Reset Command Register)
1	H'10007100	IMSTCR (Master Control Register)	IMSTCR (Master Control register)
1	H'10007102	IFAR (Address Register)	IFAR (Address Register)
1	H'10007104	IRBCLR (Rx Byte Count Low Register)	
1	H'10007106	IRBCHR (Rx Byte Count High Register)	
1	H'10007108	IRRFPLR (Rx Ring Frame Pointer Low Register)	
1	H'1000710A	IRRFPHR (Rx Ring Frame Pointer High Register)	
1	H'1000710C	ITBCLR (Tx Byte Count Low Register)	ITBCLR (Tx Byte Count Low Register)
1	H'1000710E	ITBCHR (Tx Byte Count High Register)	ITBCHR (Tx Byte Count High Register)
2	H'10007100	IMSTCR (Master Control Register)	IMSTCR (Master Control Register)
2	H'10007102	IIRC1R (Infrared Configuration 1 Register)	IIRC1R (Infrared configuration 1 Register)
2	H'10007104	IIRTCR (Infrared Transceiver Control Register)	IIRTCR (Infrared Transceiver Control Register)
2	H'10007106	IIRC2R (Infrared Configuration 2 Register)	IIRC2R (Infrared Configuration 2 Register)
2	H'10007108	ITMR (Timer Register)	ITMR (Timer Register)
2	H'1000710A	IIRC3R (Infrared Configuration 3 Register)	IIRC3R (Infrared Configuration 3 Register)
2	H'1000710C	Reserved	
2	H'1000710E	Reserved	
x	H'10007110	DMARP (DMA Data Read Port)	DMAWP (DMA Data Write Port)
x	H'10007112-H'1000711E	reserved	Reserved
x	H'10007120	ISIRR (SIR Register)	ISIRR (SIR Register)
x	H'10007122-H'10007DE	Reserved	Reserved
x	H'100071E0	IFIRCR (FIR Configuration Register)	IFIRCR (FIR Configuration Register)
x	H'100071E2-H'100071EE	Reserved	Reserved
x	H'100071F0	ITMCR (Timing Control Register)	ITMCR (Timing Control Register)
x	H'100071F2-H'10007FFE	Reserved	Reserved

Note: Bank is selected by BKSEL[4:0] of the Master Control Register (IMSTR)

11.2.3 Register Description

(1) Master Control Register (IMSTR)

Address: H'10007100 (Bank 0, 1, 2, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	IEN	TXEN	RXEN	BKSEL4	BKSEL3	BKSEL2	BKSEL1	BKSEL0
Initial Value	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

Bit	Description	Default
7	**FIR Interrupt Enable (IEN)** Setting this bit to 1 enables all FIR Controller interrupts.	0
6	**Transmitter Enable (TXEN)** Setting this bit to 1 enables the transmitter logic in the FIR Controller. No packets are transmitted until the transmitter has been enabled.	0
5	**Receiver Enable (RXEN)** Setting this bit to 1 enables the receiver logic in the FIR Controller. No packets are received until the receiver has been enabled.	0
4 - 0	**Bank Select (BKSEL[4:0])** Bank# b4 b3 b2 b1 b0 Bank0 0 0 0 0 0 Bank1 0 0 0 0 1 Bank2 0 0 0 1 0	00000

(2) Master Status Register (IMSTSR)

Address: H'10007102 (Bank 0, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	TMI	TXI	RXI	IID2	IID1	IID0	-
Initial Value	-	0	0	0	0	0	0	-
R/W	-	R	R	R	R	R	R	-

Bit	Description	Default
7	Reserved	-
6	**Timer Interrupt (TMI)** When set to 1, indicates a timer interrupt is pending.	-
5	**Transmitter Interrupt (TXI)** When set to 1, indicates a transmitter interrupt is pending.	-
4	**Receiver Interrupt (RXI)** When set to 1, indicates a receiver interrupt is pending. The following conditions clear the Rx interrupt condition. • Reading the Rx Ring Frame Counter Low Register • Issuing a RESET, Rx SPECIAL CONDITION INTERRUPT command • Clearing the Rx Enable bit • HARDWARE REST • SOFTWARE RESET	-
3 - 1	**Interrupt identification (IID[2:0])** These three bits correspond to interrupt identification ID2 - ID0 which provide an alternative method for identifying the interrupt source by indicating the interrupt type and priority level as shown below: **Interrupt Type / ID2 / ID1 / ID0 / Priority** Rx Special Condition / 1 / 0 / 0 / Highest 1. FIFO Overrun 2. Frame Error 3. EOF 4. Rx Abort 5. Sync/Hunt Rx Data Available / 1 / 0 / 1 / Second Tx Buffer Empty / 1 / 1 / 0 / Third Tx Special Condition / 1 / 1 / 1 / Fourth 1. FIFO Underrun 2. EOM 3. Early EOM	-
0	Reserved	-

(3) Miscellaneous Control Register (IMISCR)

Address: H'10007102 (Bank 0, Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	DCS1	DCS0	-	ILOOP	-	-	-	-
Initial Value	0	0	-	0	-	-	-	-
R/W	W	W	-	W	-	-	-	-

Bit	Description	Default
7 - 6	**DMA Channel Select (DCS[1:0])** DMA Channel Select. Specify DMA channel usage. b7 b6 DMA Channel Select 0 0 No DMA 0 1 DMA Channel for Receive 1 0 DMA Channel for Transmit 1 1 Reserved	00
5	Reserved	-
4	**Internal Loop-back (ILOOP)** When set to 1, the 4Mbit modem issues a transmit data output signal, which is then internally looped back to its receive data input. This allows for diagnostic testing of the modem transmit and receive data paths.	0
3 - 0	Reserved	-

(4) Rx FIFO Register (IRFR)

Address: H'10007104 (Bank 0, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
7 - 0	**Receive Data:** Used to read receive packet data from Rx FIFO.	-

(5) Tx FIFO Register (ITFR)

Address: H'10007104 (Bank 0, Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
Initial Value	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W

Bit	Description	Default
7 - 0	**Transmit Data:** Used to write transmit packet data to Tx FIFO.	-

(6) Tx Control 1 Register (ITC1R)

Address: H'10007106 (Bank 0, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	RTS	TFRIEN	TFUIEN	TFTL	ADRTS	ACEOM	TIDL	UA
Initial Value	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

Bit	Description	Default
7	**REQUEST TO SEND Control (RTS)** Setting this bit to '1' activates the REQUEST TO SEND signal to the modem. Note: The Tx Enable bit (bit 6 of Master Control Register) must be set, and Tx FIFO must contain transmit data prior to activating RTS. Setting this bit to '0' de-activates the REQUEST TO SEND signal to the modem.	0
6	**Tx FIFO Ready interrupt Enable (TFRIEN)** Setting this bit to '1' enables Tx FIFO Ready interrupts. Setting this bit to '0' disables Tx FIFO Ready interrupts.	0
5	**Tx FIFO Underrun/EOM Interrupt Enable (TFUIEN)** Enables Tx FIFO Underrun/EOM Interrupt: Setting this bit to '1' enables FIFO underrun and EOM interrupts.	0
4	**Tx FIFO Threshold Level Control (TFTL)** Setting this bit to '1' sets the Tx FIFO threshold to half-empty level. Setting this bit to '0' sets the Tx FIFO threshold to not full level. This setting affect timing of DMA request.	0
3	**Automatic Deactivation REQUEST TO SEND Control (ADRTS)** Setting this bit to '1' enables automatic deactivation of the modem REQUEST TO SEND line at the end of transmission. For back-to-back transmissions, it is desirable that REQUEST TO SEND signal remains active for the entire duration in which packets are transmitted. It is therefore recommended that AUTO RESET RTS should be disabled while running back-to-back transmissions.	0

Tx Control 1 Register (ITC1R) [cont'd]

Bit	Description	Default
2	**Automatic Clear EOM Control (ACEOM)** Setting this bit to '1' causes the EOM bit to be cleared automatically when the Tx Status Register is read. Setting this bit to '0' causes the EOM bit to remain set after the Tx Status Register has been read. Only a RESET FIFO UNDERRUN/EOM LATCH command or a hardware reset can clear it.	0
1	**Tx Idle Control (TIDL)** Setting this bit to '1' causes the TXD output line to remain in the inactive state when the transmitter is idle. Setting this bit to '0' causes the transmitter to transmit continuous flags (1Mbit mode) or continuous preambles (4Mbit mode) when the transmitter is idle.	0
0	**Underrun Abort (UA)** Underrun Abort: When a FIFO underrun occurs, the software has two options before transmission is terminated. One option is to send an abort sequence to the receiving end. The other option is to transmit a CRC and an ending/stop flag following the transmission of the last data byte in Tx FIFO. Setting this bit to '1' causes the transmitter to transmit an abort sequence when underrun occurs. Setting this bit to '0' causes the transmitter to transmit a CRC and an ending/stop flag immediately following the transmission of the last data byte in Tx FIFO before the underrun condition occurs.	0

(7) Tx Control 2 Register (ITC2R)

Address: H'10007108 (Bank 0, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	SB	ACRCG	SIP	NSFP	EEIL3	EEIL2	EEIL1	EEIL0
Initial Value	0	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
7	**Send Break (SB)** Send Break: Setting this bit to '1' causes the transmitter to transmit zeros.	0
6	**Automatic CRC Generation (ACRCG)** Setting this bit to '1' enables automatic CRC generation of all outgoing packets. The CRC is automatically generated by the transmitter logic and transmitted after the data field, but before the ending flag. Setting this bit to '0' disables CRC generation. This allows transmission of packets already containing a valid CRC.	1
5 - 4	**SIR Interaction Pulse Control (SIP)** Commands the 4Mbit modem to send a SIR Interaction Pulse based on the bit setting. A '01' bit setting instructs the 4Mbit modem to transmit a SIP at the end of current packet. A '10' bit setting instructs the 4Mbit modem to transmit a SIP immediately, regardless of the modem current activity. Note: SIP control bits are self-clearing.	00
3	**Number of Starting Flags or Preambles (NSFP)** Specifies the number of starting flags or preambles to transmit for a given packet. Setting this bit to '1' causes only two starting flags or a single preamble to be transmitted per packet. Setting this bit to '0' causes several starting flags or preambles to be transmitted.	0
2 - 0	**Early EOM Interrupt Level (EEIL[2:0])**	000

Early EOM Interrupt Level:

b2	b1	b0	Early EOM Interrupt Level
0	0	0	Interrupt by EOM Only
0	0	1	Tx byte Count = 16
0	1	0	Tx Byte Count = 32
0	1	1	Tx byte Count = 64
1	0	0	Tx Byte Count = 128
1	0	1	Tx Byte Count = 256
1	1	0	Tx Byte Count = 512
1	1	1	Tx byte Count = 1024

(8) Tx Status Register (ITSR)

Address: H'1000710A (Bank 0, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	TFUR	EOM	TFRDY	EEOM
Initial Value	-	-	-	-	0	0	1	0
R/W	-	-	-	-	R	R	R	R

Bit	Description	Default
7 - 4	Reserved	-
3	**Tx FIFO Underrun (TFUR)** When set to '1,' indicates Tx FIFO ran out of data before the transmitter could finish transmitting all the data (i.e. Tx FIFO is empty, and the Tx Byte Count value is greater than zero). This bit must be reset by an explicit FIFO UNDERRUN/EOM LATCH command.	0
2	**End of Message (EOM)** When set to '1,' indicates transmission completed successfully. The EOM interrupt occurs immediately after the CRC and ending flag have been transmitted. If bit 2 of Tx Control 1 Register (Auto Reset EOM) is enabled, the EOM bit will automatically clear when Tx Status is read. The EOM bit can also cleared by a RESET FIFO UNDERRUN LATCH command from the Reset Command Register.	0
1	**Tx FIFO Ready (TFRDY)** When set to '1,' indicates Tx FIFO is ready for more data transfers. When the bit 6 of Tx Control 1 Register is set, an interrupt is generated whenever this condition becomes true. Alternately, this bit may be polled when the interrupt is disabled. When Tx FIFO is full, this bit is cleared to '0'.	1
0	**Early EOM (EEOM)** When set to '1,' indicates the Tx Byte Count has reached the count level set by the Early EOM Interrupt Level bits. This bit is cleared by reading Tx Status.	0

(9) Rx Control Register (IRCR)

Address: H'1000710C (Bank 0, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	RFTL	ACRCC	RADM1	RADM0	SYNIEN	-	RFRIEN	SCIEN
Initial Value	0	1	0	0	0	-	0	0
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W

Bit	Description	Default
7	**Rx FIFO Threshold Level (RFTL)** Setting this bit to '1' sets the Rx FIFO threshold to half-full (more than 8 bytes of Receive data are still remaining in FIFO). Setting this bit to '0' sets the Rx FIFO threshold to not empty (more than 1 byte of Receive data remaining in FIFO). This setting affect DMA request timing.	0
6	**Automatic CRC check (ACRCC)** Setting this bit to '1' enables automatic CRC checking of all incoming packets. Setting this bit to '0' disables CRC checking. Disabling this bit results in no CRC errors being reported.	1
5 - 4	**RX Address Mode (RADM[1:0])** Specify the type of address filtering to apply for determining which receive frames to accept. b5 b4 Rx Address Mode 0 0 All packets accepted 0 1 Address must match Frame Address Register (FAR) 1 0 Address (high nibble) must match FAR 1 1 Reserved Note: Packets with a universal address 0x7F are always accepted.	00
3	**Sync/Hunt Change interrupt Enable (SYNIEN)** Enables Sync/Hunt Change interrupt: Setting this bit to 1 enables Sync/Hunt Change interrupts.	0
2	Reserved	-
1	**Rx FIFO Ready Interrupt Enable (RFRIEN)** Setting this bit to '1' enables Rx FIFO Ready interrupts. Setting this bit to '0' disables Rx FIFO Ready interrupts.	0
0	**Special Condition Interrupt Enable (SCIEN)** Setting this bit to '1' enables the following special condition interrupts: • Overrun • Frame Error • End of Frame (EOF) • Rx Abort Setting this bit to '0' disables the above special condition interrupts	0

(10) Rx Status Register (IRSR)

Address: H'1000710E (Bank 0, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	ABORT	CRCER	RFOVF	EOF	RFEM	SYNC	-	-
Initial Value	0	0	0	0	0	1	-	-
R/W	R	R	R	R	R	R	-	-

Bit	Description	Default
7	**Abort Detect (ABORT)** When set to '1', indicates abort sequence detected in the receive data stream of current packet. In 1Mbit mode, the abort sequence is characterized by seven or more consecutive 1 in the data stream. In 4Mbit mode, the abort sequence is represented by two (2) or more illegal symbols 0000 after a valid start flag but before a complete stop flag; or an illegal symbol, which is not part of a valid stop flag field, received any time after a valid flag.	-
6	**CRC or Alignment Error (CRCER)** When set to '1', indicates a CRC or alignment error was detected in the incoming data stream. This bit is automatically cleared upon detection of the beginning/start flag of the next incoming packet.	-
5	**RX FIFO Overflow (RFOVF)** When set to '1', indicates the host system was not fast enough removing the data out of Rx FIFO before it overflowed with receive data.	-
4	**End of Frame (EOF)** End of Frame (EOF): When set to '1', indicates an ending/stop flag or abort sequence was detected in the incoming data stream. This bit is automatically cleared upon detection of the beginning/start flag of the next incoming packet.	-
3	**RX FIFO Empty (RFEM)** When set to '1,' indicates Rx FIFO is not empty. When set to '0,' indicates Rx FIFO is empty. When this bit is set, it does not cause an interrupt; rather it is used to unload the FIFO by polling. Note: Rx FIFO Level (bit 7 of Rx Control Register) has no effect on the Rx Data Available bit.	-
2	**Sync/Hunt Status (SYNC)** When set to '1,' indicates a transition or status change occurred on the internal Sync/Hunt signal. The following conditions cause the Sync/Hunt signal to change states: • When Enter Hunt Mode command is issued • Valid SDLC start or stop flag is detected • Valid preamble or stop flag is detected (4Mbit mode) If bit 3 of Rx Control Register (Enables Sync/Hunt Change Interrupt bit) is enabled, the setting of Sync/Hunt Change bit causes an interrupt to the host system. Reading the Rx Status Register after the interrupt has occurred clears the Sync/Hunt Change bit. If the bit 3 of Rx Control Register is disabled, reading Rx Status register will directly provide the status of the Sync/Hunt signal and will not clear the Sync/Hunt Change bit.	-
1 - 0	Reserved	-

(11) Reset Command Register (IRSTCR)

Address: H'1000710E (Bank 0, Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	RSTC3	RSTC2	RSTC1	RSTC0	-	-	-	-
Initial Value	0	0	0	0	-	-	-	-
R/W	W	W	W	W	-	-	-	-

Bit	Description	Default
7 - 4	**RESET Command:** Used to send a RESET signal to the appropriate hardware in order to clear a particular status condition, a counter, or a general reset. b7 b6 b5 b4 RESET Command 0 0 0 1 Enter Hunt Mode 0 0 1 0 Reset Rx FIFO Pointer 0 0 1 1 Reset Rx Special Condition Interrupt 0 1 0 0 Reset Rx Ring Frame Pointer 0 1 0 1 Reset FIFO Underrun/EOM Latch 0 1 1 0 Reset Tx FIFO Pointer 0 1 1 1 Hardware Reset Note: These bits are self-clearing (i.e. a programmer does not need to reset the RESET Command bit value to 0000).	-
3 - 0	Reserved	-

(12) Frame Address Register (IFAR)

Address: H'10007102 (Bank 1, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	RFA7	RFA6	RFA5	RFA4	RFA3	RFA2	RFA1	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	-

Bit	Description	Default
7 - 1	**Rx Frame Address, A1-A7:** Specify the address value that must be contained in the address field of incoming frames. Bit 7 6 5 4 3 2 1 Bit Name A7 A6 A5 A4 A3 A2 A1	-
0	Bit 0 is always 0.	-

(13) Rx Byte Count Low Register (IRBCLR)

Address: H'10007104 (Bank 1, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
7 - 0	**Rx Byte Count, D0 - D7:** Provides a running count (Low-order value) of the number of bytes of data being received. It is useful when receiving back-to-back packets.	-

Bit	7	6	5	4	3	2	1	0
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0

(14) Rx Byte Count High Register (IRBCHR)

Address: H'10007106 (Bank 1, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	RBC12	RBC11	RBC10	RBC9	RBC8
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	R	R	R	R	R

Bit	Description	Default
7 - 5	Reserved	-
4 - 0	**Rx Byte Count, D8 -D12:** Provides a running count (high-order value) of the number of bytes of data being received. It is useful when receiving back-to-back packets.	-

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	D12	D11	D10	D9	D8

(15) Rx Ring Frame Pointer Low Register (IRRFPLR)

Address: H'10007108 (Bank 1, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
7 - 0	**Ring Frame Pointer (RFP), D0 - D7:** Used in back-to-back packet reception to provide the end-of-packet pointer value. Bit 7 6 5 4 3 2 1 0 Bit Name D7 D6 D5 D4 D3 D2 D1 D0 The RFP value is initially set to 0000h. Thus, before receiving the first packet, software should not use the RFP value for any computation.	-

(16) Rx Ring Frame Pointer High Register (IRRFPHR)

Address: H'1000710A (Bank 1, Read)

Bit	7	6	5	4	3	2	1	0
Bit Name	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
7 - 0	**Ring Frame Pointer (RFP), D8 - D15:** Used in back-to-back packet reception to provide the end-of-packet pointer value. Bit 7 6 5 4 3 2 1 0 Bit Name D15 D14 D13 D12 D11 D10 D9 D8	-

(17) Tx Byte Count Low Register (ITBCLR)

Address: H'1000710C (Bank 1, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	TBC7	TBC6	TBC5	TBC4	TBC3	TBC2	TBC1	TBC0
Initial Value	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

Bit	Description	Default
0 - 7	**Tx Byte Count, D0 – D7:** Provide a running count (low-order value) of the number of bytes remaining to be transmitted. Before enabling transmission, software loads this register with the low-order byte length of the data packet. When the counter reaches zero, the transmitter ceases to make DMA requests. Transmission continues until Tx FIFO is depleted.	

Bit	7	6	5	4	3	2	1	0
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0

(18) Tx Byte Count High Register (ITBCHR)

Address: H'1000710E (Bank 1, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	TBC12	TBC11	TBC10	TBC9	TBC8
Initial Value	-	-	-	1	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
7 - 5	Reserved	-
4 - 0	**Tx Byte Count, D8 - D12:** Specify the high-order byte length of the data packet to be transmitted. Refer to Tx Byte Count Low Register.	10000

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	D12	D11	D10	D9	D8

(19) Infrared Configuration 1 Register (IIRC1R)

Address: H'10007102 (Bank 2, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	IRSPD3	IRSPD2	IRSPD1	IRSPD0	IRMOD3	IRMOD2	IRMOD1	IRMOD0
Initial Value	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

Bit	Description	Default
7 - 4	**Infrared Speed (IRSPD[3:0])**	0000
	Specify the data rate under 1Mbit FIR modulation.	
	B7 b6 b5 b4 Infrared Speed	
	0 0 0 0 1.152 Mbps	
	0 0 0 1 0.756 Mbps	
	0 0 1 0 0.288 Mbps	
3 - 0	**Infrared Modulation (IRMOD[3:0])**	0000
	Specify the modulation mode of infrared communication.	
	b3 b2 b1 b0 Infrared Modulation	
	0 0 0 0 HP-SIR	
	0 0 0 1 Sharp ASK	
	0 0 1 0 1.152 Mbps IrDA	
	0 0 1 1 Reserved	
	0 1 0 0 4 Mbps IrDA	

(20) Infrared Transceiver Control Register (IIRTCR)

Address: H'10007104 (Bank 2, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	DFREQ	MODSEL	ECHO	-	TXDF	-
Initial Value	-	-	0	0	0	-	0	-
R/W	-	-	R/W	R/W	R/W	-	R/W	-

Bit	Description	Default
7 - 6	Reserved	-
5	**High/Low Data frequency (DFREQ)** When an HP-like transceiver is selected in the configuration register, high or low infrared data frequency is determined by this bit. Setting this bit to '1' causes the low frequency pin RX# to become the receiving source. Setting this bit to '0' causes the high frequency pin RX2# to become the receiving source. When an IBM-like transceiver is selected in the configuration register, this bit will be invalidated.	-
4	**Mode Select (MODSEL)** When an IBM-like transceiver is selected in the configuration register, mode select function will be present on the MODSEL pin. Setting this bit to '1' causes the external MODSEL pin to be high. Setting this bit to '0' causes the external MODSEL pin to be low. When an HP-like transceiver is selected in the configuration register, this bit will be invalidated.	-
3	**Echo On (ECHO)** Setting this bit to '1' sets the optical loopback feature. This bit is set to 0 on power-up.	-
2	Reserved	-
1	**TXD Force (TXDF)** This bit works together with Mode Select bit (bit 4) to switch the receiving mode between SIR and FIR for IBM-like transceiver, Setting this bit to '1' causes the external TXD pin to become asserted. Setting this bit to '0' causes the external TXD pin to become unasserted. Note: This bit must be set to '0' for normal operation.	-
0	Reserved	-

(21) Infrared Configuration 2 Register (IIRC2R)

Address: H'10007106 (Bank 2, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	ACEN	-	-	-	CCTRL1	CCTRL0	DSIRI	DFIRI
Initial Value	0	-	-	-	0	0	0	0
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W

Bit	Description	Default
7, 3, 2	**4M pulse auto-chopping mechanism (ACEN, CCTRL1, CCTRL0)** These bits control the 4M pulse auto-chopping mechanism. This feature handles transceivers which deliver single pulses that exceed the 165ns maximum supported by the 4M demodulator. When autochop is enabled, the circuit measures a typical pulse which during a frame preamble sequence and adjusts the chopping level accordingly. The error threshold can be adjusted by using the chop control bits (CCTRL0 and CCTRL1). The recommended setting for 4M mode is ACEN=1, CCTRL1=CCTRL0=0. These bits are reset to 0 on power-up. If this feature is used, the software must set these bits accordingly when entering 4M mode and reset them when leaving 4M mode. Chopping operation with the autochop enable bit reset is provided for diagnostic tests of the transceiver and is not recommended for normal operation where pulse widths can vary significantly. The setting and their effects are: **Autochop Enable / CCTRL1 / CCTRL0 — Effect** 000 — Chopping circuit is disabled 001 — Extend the single pulse width tolerance to 187ns. Back-to-back pulses must be greater than 209ns. 010 — Extend the single pulse width tolerance to 229ns. Back-to-back pulses must be greater than 249ns. 011 — Extend the single pulse width tolerance to 208ns. Back-to-back pulses must be greater than 229ns. Autochop enabled with maximum tolerance for error. Back-to-back pulses must be 62ns longer than a single pulse sample. 101 — Autochop enabled with less tolerance for error. Back-to-back pulses must be 42ns longer than a single pulse sample. 110 — Autochop enabled with zero tolerance for error. Back-to-back pulsed must be 42ns longer than a single pulse sample. 111 — Autochop enabled with zero tolerance for error. Back-to-back pulses must be longer than a single pulse sample. Digital transceiver with Rx signal in synchronous with 48MHz clock.	-
6 - 4	Reserved	-
1	**Disables SIR Interrupt (DSIRI)** Setting this bit to '1' causes SIR interrupt request to be masked.	-
0	**Disables FIR Interrupt (DFIRI)** Setting this bit to '1' causes FIR interrupt request to be masked.	-

(22) Timer Register (ITMR)

Address: H'10007108 (Bank 2, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
Initial Value	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

Bit	Description	Default
7 - 0	**Timer value (TMR[7:0])** Timer value, D0 - D7: Specify the initialization value for the down counter. The counter has a period of 128us. When the counter reaches zero, an interrupt is generated. Bit 7 6 5 4 3 2 1 0 Bit Name D7 D6 D5 D4 D3 D2 D1 D0	-

(23) Infrared Configuration 3 Register (IIRC3R)

Address: H'1000710A (Bank 2, Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	SCDIEN	SCD	-	-	-	-	TMIEN	TMI
Initial Value	0	0	-	-	-	-	0	0
R/W	R/W	R/W	-	-	-	-	R/W	R/W

Bit	Description	Default
7	**Enables Sharp CD Interrupt (SCDIEN)** Setting this bit to 1, enables Sharp Carrier Detect interrupts.	-
6	**Sharp Carrier Detect (SCD)** When set to 1, this READ-only status bit indicates a 500KHz Sharp ASK carrier has been detected. To clear the interrupt, software must write a 1 to this bit.	-
5 - 2	Reserved	-
1	**Enables Timer Interrupt (TMIEN)** Setting this bit to 1 enables Timer Interrupt.	-
0	**Timer Interrupt (TMI)** When set to 1, indicates a timer interrupt is pending. To clear the interrupt, software must write a 1 to this bit. This bit is self-clearing.	-

(24) SIR Register (ISIRR)

Address: H'10007120 (Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	SLOOP	SIRMOD
Initial Value	-	-	-	-	-	-	0	1
R/W	-	-	-	-	-	-	R/W	R/W

Bit	Description	Default
7 - 2	Reserved	-
1	**SIR Loop-back (SLOOP)** Sets this bit to 1 to enable SIR loop back mode. Sets this bit to 0 to enable SIR Normal mode.	0
0	**SIR Module Select (SIRMOD)** Sets this bit to 1, IrDA module only supports HP-SIR modulation and uses low frequency (SIR-CLK 1.8432MHz) clock. Sets this bit to 0, IrDA module supports 4M mode, and uses IRCLK (48MHz) clock.	1

(25) FIR Configuration Register (IFIRCR)

Address: H'100071E0 (Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	RX2_PP	RX_PP	TMODE
Initial Value	-	-	-	-	-	0	0	0
R/W	-	-	-	-	-	R/W	R/W	R/W

Bit	Description	Default
7 - 3	Reserved	0
2	**RX2# Pull_up (RX2_PP)** Setting this bit to 1 enables the pull-up function for MODSEL/RX2#. Setting this bit to 0 disables the pull-up function for MODSEL/RX2#.	0
1	**RX# Pull_up (RX_PP)** Setting this bit to 1 enables the pull-up function for RX#. Setting this bit to 0 disables the pull-up function for RX#.	0
0	**Transceiver Mode (TMODE)** When this bit is set to ‘1’, IBM-mode transceiver is supported. When this bit is reset to ‘0’, HP-mode transceiver is supported.	0

(26) Timing Control Register (ITMCR)

Address: H'100071F0 (Read/ Write)

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	TMCR2	TMCR1	TMCR0
Initial Value	-	-	-	-	-	1	1	0
R/W	-	-	-	-	-	R/W	R/W	R/W

Bit	Description	Default
7 - 3	Reserved	0
0 - 2	**Timing Control (TMCR[2:0])** These three bits TMCR[2:0] are used to adjust the timing for different CKIO frequency rates. 3'b000, 3'b001 : Set as CKIO frequency is 12.5MHz. 3'b010 : Set as CKIO frequency is 25MHz. 3'b011 : Set as CKIO frequency is 30MHz. 3'b100 : Set as CKIO frequency is 40MHz. 3'b101 : Set as CKIO frequency is 50MHz. 3'b110 : Set as CKIO frequency is 66MHz. 3'b111 : Reserved.	

11.3 FIR Transmit Operation

T1: Setup Phase:

(1) Set up Tx Control Registers for Transmitting options.
   a. Set DCS[1:0] in IMISCR to select DMA channel for Transmit. (Set 80h to IMISCR)
   b. Set SIP[1:0] and NSFP in ITC2R to send a SIR Interaction Pulse and decide the number of starting flags or preamble. (Set 58h to ITC2R)
   c. Set RSTC[3:0] in IRSTCR to reset Tx FIFO pointer. (Set 60h to IRSTCR)
   d. Set BKSEL[4:0] to select Bank 2 register. (Set 02h to IMSTCR)
   e. Set IRMOD[3:0] and IRSPD[3:0] in IIRC1R to specify the modulation rate. (Set 04h to IIRC1R)

(2) Load the byte count to the Tx Byte Count Register.
   a. Set BKSEL[4:0] to select Bank 1 register. (Set 01h to IMSTCR)
   b. Write the transmit byte counts into ITBCLR and ITBCHR.

(3) Set up the host DMA controller channel 0 and the Tx packet.

(4) Set RTS and Transmitter Enable bits.
   a. Set IEN, TXEN and BKSEL[4:0] in IMSTCR to enable FIR interrupt, Transmitter and select bank 0. (Set c0h to IMSTCR)
   b. Set RTS, TFRIEN, TFUIEN, TFTL, ADRTS, TIDL and UA in ITC1R to start to transmit data. RTS is used to activate the REQUEST TO SEND and start transmission. TFRIEN enables Tx FIFO Ready interrupt request, TFUIEN enables Tx FIFO underrun/EOM interrupt request and TFTL controls Tx FIFO Threshold level. ADRTS automatically deactives the REQUEST TO SEND, TIDL controls Tx Idle state and UA specifies the FIFO underrun sequence. (Set fbh to ITC1R)

T2: Startup Phase:

(1) RTS is active. If no carrier is detected (Rx doesn't receive data), then the transmitter begins to transmit.

(2) DMA request is activated if DMA is enabled. The Tx FIFO is being filled with transmitted data. If DMA is not enabled, Write Tx FIFO command can be used. (Write data to ITFR)

(3) If the NSFP bit in ITC2R is 0, the transmitter starts sending start flags (1M) or Preambles (4M) until the Tx FIFO is half filled (8 bytes). If the NSFP bit is 1, the transmitter waits until the Tx FIFO is half filled, then sends two (2) start flags (1M mode) or preambles and one (1) Start Flag (4M mode).

T3: Data Send Phase:

(1) The transmitter starts sending data stored in the FIFO.

(2) DMA request to the host is active when FIFO is not full.

T4: End of Transmission

(1) The byte Counter counts down to 0.
(2) DMA request stops. The transmitter sends out the remaining data in the FIFO.
(3) CRC generator inverts the CRC and sends it out.
(4) Closing Flag is sent. EOM latch is set, and Interrupt is activated.

T5: Idle Phase:

(1) The transmitter continues sending '1's or Flags (1M)/Preambles (4M) depending on TIDL bit in ITC1R.
(2) The host reads ITSR to check for transmission completion status.
(3) Reset EOM, Transmitter Enable and RTS bits.
   a. Set RSTC[3:0] in IRSTCR to reset FIFO EOM latch. (Set 50h to IRSTCR)
   b. Reset TXEN in IMSTCR to disable Transmitter. (Set 00h to IMSTCR)
   c. RTS is automatically deactivated depending on ADRTS in ITC1R.
(4) End of transmission.

11.4 FIR Receive Operation

Receiving logic facilities:

(1) Receive control circuitry.
(2) Rx Byte Count Register to keep track of received bytes.
(3) Rx FIFO, 16 x 11 bits 8-bit data + 3-bit status: Frame Error, Abort and End Of Frame.
(4) Rx Ring Frame Counter to keep track of the Rx byte number in the host Rx buffer.
(5) Rx Ring Frame Pointer which points to the last byte of the last received packet in the host Rx buffer.

T1: Setup Phase:

(1) Set Rx control Register for receiving options.
   a. Set DCS[1:0] in IMISCR to select DMA channel for Receive. (Set 40h to IMISCR)
   b. Set Rx interrupt enable bits in IRCR to enable interrupt requests. (Set 4bh to IRCR)
   c. Set RSTC[3:0] in IRSTCR to reset Rx FIFO pointer. (Set 20h to IRSTCR)
   d. Set BKSEL[4:0] to select Bank 2 register. (Set 02h to IMSTCR)
   e. Set IRMOD[3:0] and IRSPD[3:0] in IIRC1R to specify the modulation rate. (Set 04h to IIRC1R)
   f. Set ACEN, CCTRL[1:0] in IIRC2R to enable auto-chopping circuit. (Set f0h to IIRC2R)
(2) Set up the host DMA controller channel 0.

(3) Set Receiver Enable bit.

Set IEN and RXEN to enable FIR interrupt and Receiver. (Set a0h to IMSTCR)

(4) FIR mode logic detects Carrier, receive clock starts running. If consecutive '1's are received, the receive clock may not be synchronized with the incoming data.

T2: Flag(s)/Preambles Detection:

(1) When the Start flag is detected, all counters in the receiver are initialized. Characters can be recognized from this point on.

(2) '0' deletion starts for 1M-bit mode only.

T3: Address Matching:

(1) The first non-flag byte after the starting flag is the address. Depending on the address mode option (RADM[1:0] in IRCR), the frame can be rejected or started for receiving. If the frame is rejected, the receiver will look for the next Start flag and another address match.

T4: Data Receiving:

(1) When a data byte is received, the data and the three status bits are stored in the Rx FIFO.

(2) If DMA is enabled, DMA request is activated when the FIFO threshold level is reached. DMA request continues until all data stored in the FIFO have been transferred to the host receive buffer. However, the three status bits in the FIFO will not be transferred to the host.

(3) If DMA is not used, the READ Rx FIFO command can be used. (Read data from IRFR) Read the status bits first, then read the data byte. If EOF or Abort is set to 1, the data byte just read is the last byte of the packet. If the FIFO is still not empty, the next entry is the beginning of another packet.

(4) Rx Byte Counter and Rx Ring Frame counter increase accordingly.

T5: Closing Flag:

(1) End Of Frame bit will be set when the closing flag is detected.

(2) CRC pattern is checked. Frame error is set if CRC is wrong.

T6: Post Frame Phase:

(1) DMA request continues until all the received data in the FIFO have been transferred. Two more bytes in the following format will be stored in Rx FIFO and transmitted to the host receive buffer:

First byte: 7 - 0, Byte count 7 to 0

Second byte: 7 - Abort

6 - Frame Error

5 - Overrun

4 to 0 - Byte Count 1

(2) DMA de-activates.

(3) Rx Ring Frame Pointer is updated pointing to the second byte above that has been successfully stored in the host Rx buffer.

(4) If DMA is not used, the last two bytes will not be stored in the FIFO. The status bit EOF will be set at the last Frame Check byte received.

(5) Repeat steps T2 through T6 if receiving continuous frame is required.

(6) The host reads IRSR to check for receive completion status.

11.5 Example of Initialization and Programming Procedure for HP-SIR

1. **Enable HP-SIR:**

   For HP-SIR, SHARP-ASK and 4M mode FIR (use 48 MHz clock):

   (1) Set BKSEL[4:0] to select Bank 2 register. (Set 02h to IMSTCR)

   (2) Set IRMOD[3:0] in IIRC1R to select HP-SIR mode. (Set 00h to IIRC1R)

   (3) Set SIRMOD in ISIRR to select FIR MODEM. (Set 00h to ISIRR)

   For HP-SIR only (use 1.8432MHz clock):

   (1) Set SIRMOD in ISIRR to select HP-SIR MODEM. (Set 01h to ISIRR)

2. **Set UART portion of FIR module baud rate**

   (1) Set DLAB (Set 87h to IrLCR)

   (2) Set baud rate to 115.2K (Set 01h to IrDLL and 00h to IrDLM)

3. **Enable Interrupt**

   (1) Set DLAB = 0 (Set 07h to IrLCR)

   (2) Enable transmitting data interrupt (Set 02h to IrIER)

   (3) Enable modem interrupts (Set 08h to IrMCR)

4. **Fill data into UART**

   Write data to IrTBR. (Set aah to IrTBR)

5. **Observe the waveform on the pin (TXD) and IRQ0#**

Section 12 UART

12.1 Overview

The UART module is 16550 compatible. This module performs the serial to parallel conversion on received data, and parallel to serial conversion on transmitted data. Individually, they contain a programmable baud rate generator that is capable of dividing the input clock by a number from 1 to 65535; the data rate of each can also be programmed from 115.2K baud down to 50 baud. The c haracter options are programmable for 1 start bit; 1, 1.5, or 2 stop bits; even, odd, stick or no parity; and privileged interrupts. The HD64465 provides two UART ports.

12.2 Features

- Programmable FIFO or character mode
- In FIFO mode 16-byte FIFO buffer on the transmitter and receiver
- Add or delete standard asynchronous communication bits (start, stop, parity) to or from serial data.
- The programmable Baud rate generator allows the division of input clock by 1 to 2^{16}-1 and generates internal 16X clock
- MODEM control function (CTS#, RTS#, DTR#, DSR#, RI#, DCD#)
- Fully programmable serial-interface characteristics:
- 5, 6, 7 or 8 bit characters
- Even, odd, forced 0/1 or no parity bit generation and detection
- 1 ½, or 2 stop bit generation
- Baud rate generation (Dc to 115.2K baud)
- False start bit detection
- Two UART ports are provided (UART0 and UART1)

12.3 Serial Channel Register Description

Table 12.1 Serial Channel Registers

Register	DLAB*	Address	READ	WRITE
Data	0	Base+h0000	URBR (Receiver Buffer Register)	UTBR (Transmitter Buffer Register)
Control	0	Base+h0002	UIER (Interrupt Enable Register)	UIER
	X	Base+h0004	UIIR (Interrupt identification Register)	UFCR (FIFO Control Register)
	X	Base+h0006	ULCR (Line Control Register)	ULCR
	X	Base+h0008	UMCR (Modem Control Register)	UMCR
	1	Base+h0000	UDLL (Divisor Latch LSB)	UDLL
	1	Base+h0002	UDLM (Divisor Latch MSB)	UDLM
Status	X	Base+h000A	ULSR (Line Status Register)	ULSR
	X	Base+h000C	UMSR (Modem Status Register)	UMSR
	X	Base+h000E	USCR (Scratch Pad Register)	USCR

Notes: 1. DLAB* is bit 7 of the Line Control Register.
2. UART0 base address = H'10008000

12.3.1 Data Register

UTBR and URBR each hold from five to eight data bits. If the transmitted data is less than eight bits, it aligns to the LSB. Either received or transmitted data is buffered by a shift register and is latched first by a holding register. The bit 0 of any word is first received and transmitted.

(1) URBR (READ only)

This register receives and holds the entering data. It contains a non-accessible shift register which converts the incoming serial data stream to a parallel 8 bit word.

(2) UTBR (WRITE only)

This register holds and transmits the data via a non-accessible shift register. It converts the outgoing parallel data to a serial stream before transmission.

12.3.2 Control Registers: UIER, UIIR, UFCR, UDLL, UDLM, ULCR, UMCR

(1) UIER (READ/WRITE)

UIER is used to enable (or disable) four active high interrupts that activate the interrupt outputs, with its lower four bits: bit 0~bit 3.

Bit	Description	Default
7 - 4	These bits are always “0”.	0
3	Sets this bit high to enable the Modem Status Interrupt when one of the Modem Status Registers changes its bit state.	0
2	Sets this bit high to enable the Receiver Line Status Interrupt, which is caused when Overrun, Parity, Framing or Break occurs.	0
1	Sets this bit high to enable the Transmitter Holding Register Empty Interrupt.	0
0	Sets this bit high to enable the Received Data Available Interrupt (and Time-out Interrupt in the FIFO mode).	0

(2) UIIR (READ only)

This register facilitates the host CPU to determine interrupt priority and its source. The priority of four existing interrupt levels is as follows:

1. Received Line Status (highest priority)
2. Received Data Ready
3. Transmitter Holding Register Empty
4. Modem Status (lowest priority)

When a privileged interrupt is pending and the type of interrupt is stored in the UIIR which is accessed by the Host, the serial channel holds back all interrupts and indicates the highest priority pending interrupts to the Host. Any new interrupts will not be acknowledged until the Host access finishes. The contents of the UIIR are described in the table on the next page.

Table 12.2 Interrupt Identification Register

FIFO Mode	Interrupt Identification Register			Interrupt Set and Reset Functions			
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt RESET Control**
0	X	X	1	-	None	None	-
0	1	1	0	First	Receiver Line Status	OE, PE, FE, or BI	ULSR READ
0	1	0	0	Second	Received Data Available	Received Data Available	URBR Read or FIFO drops below the trigger level
1	1	0	0	Second	Character Timer-out Indication	No characters have been removed from or input to the RCVR FIFO during the last 4 character times, and there is at least 1 character in it at this time.	URBR READ
0	0	1	0	Third	Transmitter Holding Register Empty	Transmitter Holding Register Empty	UIIR READ if THRE is the Interrupt Source or THR write
0	0	0	0	Fourth	Modem Status	CTS#, DSR#, RI#, DCD#	UMSR READ

Note: X = Not Defined

Bit	Description	Default
7, 6	Are set when UFCR(0)=1	-
5, 4	Always logic 0	0
3	In non-FIFO mode, this bit is a logic 0. In the FIFO mode this bit is set along with bit 2 when a time-out Interrupt is pending.	-
2, 1	Are used to identify the highest priority pending interrupt.	-
0	Is used to indicate a pending interrupt in either a hard-wired prioritized or polled environment, with a logic 0 state. When this happens, UIIR contents may be used as a pointer to the appropriate interrupt service routine.	-

(3) UFCR (WRITE only)

This register is used to enable, clear the FIFO, and set the RCVR FIFO trigger levels.

Bit	Description			Default
7, 6	These bits set the trigger levels for the RCVR FIFO interrupt.			-
	UFCR(7)	**UFCR(6)**	**RCVR FIFO Trigger Level**	
	0	0	1 byte	
	0	1	4 bytes	
	1	0	8 bytes	
	1	1	14 bytes	
5, 4	Reserved			-
3	This bit doesn't affect the Serial Channel operation. RXRDY and TXRDY functions are not available on this controller.			-
2	This self-clearing bit clears all contents of the XMIT FIFO and resets its related counter to 0 by a logic "1".			-
1	Setting this self-clearing bit to logic 1 clears all contents of the RCVR FIFO and resets its related counter to 0 (except the shift register).			-
0	XMIT and RCVR FIFO are enabled when this bit is set high. XMIT and RCVR FIFO will be disabled and cleared when this bit is cleared to low. This bit has to be a logic 1 if the other bits of the UFCR are written to or they will not be properly programmed. When this register changes to non-FIFO mode, all contents will be cleared.			-

(4) Divisor Latches (READ/WRITE)

There are two 8-bit Divisor Latches (UDLL and UDLM) which store the divisor in a 16-bit binary format. They are loaded during initialization to generate a desired Baud Rate.

Baud Rate Generator (BRG)

Each serial channel contains a programmable BRG which can take any clock input (from DC to 8 MHz) to generate standard ANSI/CCITT bit rates for the channel clocking, with an external clock oscillator. The UDLL or UDLM is a number of 16-bit format, providing the divisor range from 1 to 2^{16} to obtain the desired baud rate. The output frequency is 16X data rate.

(5) Scratch Pad Register (READ/WRITE)

This 8-bit register does not control the operation of UART in any way. It is intended as a scratch pad register to be used by programmer to temporarily hold general-purpose data.

Table 12.3 Baud Rates Using (9.216MHz/5) Clock

Desired Baud Rate	Divisor Used
50	2304
75	1536
110	1047
134.5	857
150	768
300	384
600	192
1200	96
1800	64
2000	58
2400	48
3600	32
4800	24
7200	16
9600	12
19200	6
38400	3
57600	2
115200	1

(6) ULCR (READ/WRITE)

ULCR controls the format of the data character and gives the information of the serial line.

Bit	Description	Default
7	**Divisor Latch Access Bit:** Must be set high to access the Divisor Latches of the baud rate generator during read or write operations. It must be set low to access the Data Register (URBR and UTBR) or the Interrupt Enable Register.	-
6	**Break Control:** Forces the Serial Output (SOUT) to the spacing state (logic 0) by a logic 1, and this state will remain until a low level resetting ULCR(6), enabling the serial port to alert the terminal in a communication system.	-
5	**Stick Parity Bit:** When this bit and ULCR(3) are high at the same time, the parity bit is transmitted and then detected by receiver, in opposite state by ULCR(4) to force the parity to a known state and to check the parity bit in a known state.	-
4	**Even Parity Select (EPS):** When parity is enabled (ULCR(3)=1), ULCR(4)=0 selects odd parity, and ULCR(4)=1 selects even parity.	-

ULCR (READ/WRITE) [cont'd]

Bit	Description	Default
3	**Parity Enable (PEN):** A parity bit, between the last data word bit and stop bit, will be generated or checked (transmit or receive data) when ULCR(3) is high.	-
2	**Stop Bit Select (STB):** Specifies the number of stop bits in each serial character, summarized as the following: **ULCR(2) Word Length No. of Stop Bits** 0 - 1 1 5 bits 1.5 1 6 bits 2 1 7 bits 2 1 8 bits 2 Note: The receiver will ignore all stop bits beyond the first, regardless of the number used in transmission.	-
1 - 0	**Bit 0: Word Length Select Bit 0 (WLS 0):** Bit 1: Word Length Select Bit 1 (WLS 1). Specify the number of bits in each serial character, encoded as the following: **ULCR(1) ULCR(0) Word Length** 0 0 5 bits 0 1 6 bits 1 0 7 bits 1 1 8 bits	-

(7) UMCR (READ/WRITE)

Controls the interface with the modem or data set (or device emulating a modem).

Table 12.4 Modem Control Register Bits

MCR Bits	Logic 1	Logic 0
MCR(7) 0		
MCR(6) 0		
MCR(5) 0		
MCR(4) Loop	Loop Enabled	Loop Disabled
MCR(3) Interrupt (INT) Enable	INT Enabled	INT Disabled
MCR(2) 0		
MCR(1) Request to Send (RTS#)	RTS# Output Low	RTS# Output High
MCR(0) Data Terminal Ready (DTR#)	DTR# Output Low	DTR# Output High

(7) UMCR (READ/WRITE) [cont'd]

Bit	Description	Default
7 - 5	Are always low	0
4	Provides a loopback feature for diagnostic test of the serial channel when it is set high. Serial Output (SOUT) is set to the Marking State Shift Register output Loops back into the Receiver Shift Register. All Modem Control inputs (CTS#, DSR#, RI# and DCD#) are disconnected, the four Modem Control inputs and forced to inactive high. The transmitted data is immediately received, allowing the processor to verify the transmit and receive data path of the serial channel.	0
3	Is the Output 2 bit and enables the serial port interrupt output by a logic 1.	0
2	Controls the Output 1 bit, which does not have an output pin and can only be read or written by the CPU.	0
1	Controls the Request to Send (RTS#) which is in an inverse logic state with that of UMCR(1).	-
0	Controls the Data Terminal Ready (DTR#) which is in an inverse logic state with that of the UMCR(0).	-

12.3.3 Status Register ULSR and UMSR

(1) ULSR (READ/WRITE)

This register provides status indications and is usually the first register read by the CPU to determine the cause of an interrupt or to poll the status of each serial channel. The contents of the ULSR are described below:

Bit	Description	Default
7	In 16450 mode, this bit is always 0. In the FIFO mode, it is set high when there is at least one parity error, framing or break interrupt in the FIFO. This bit is cleared when the CPU reads ULSR, if there are no subsequent errors in the FIFO.	0
6	This READ-only bit indicates that the Transmitter Holding Register and Transmitter Shift Register are both empty; otherwise, this bit is "0". It has the same function in the FIFO mode.	-
5	Transmitter Holding Register Empty (THRE). This READ-only bit indicates that the UTBR is empty and ready to accept a new character for transmission. It is set high when a character is transferred from the THR into the Transmitter Shift Register, causing a priority 3 UIIR interrupt which is cleared by a read of UIIR. In the FIFO mode, it is set when the XMIT FIFO is empty and it is cleared when at least one (1) byte is written to the XMIT FIFO.	-
4	Break Interrupt (BI) status bit indicates that the last received character was a break character. The break interrupt status bit will be asserted only when the last received character, parity bits and stop bits are all break bits. When any of these error conditions is detected (ULSR(1) to ULSR(4)), a Receiver Line Status interrupt (priority 1) will be produced in the UIIR, with the UIER(2) previously enabled.	-
3	Framing Error (FE) bit, a logic 1, indicates that the stop bit in the received character was not valid. It is reset low when the CPU reads the contents of ULSR.	-

ULSR (READ/WRITE) [cont'd]

Bit	Description	Default
2	Indicates the parity error (PE) with a logic 1 indicating that the received data character does not have the correct even or odd parity, as selected by ULCR(4). It will be reset to "0" whenever the ULSR is read by the CPU.	-
1	Overrun Error (OE) bit which indicates by a logic 1 that the URBR has been overwritten by the next character before it had been read by the CPU. In the FIFO mode, the OE occurs when the FIFO is full and the next character has been completely received by the Shift Register. It will be reset when the SR is read by CPU.	-
0	Data Ready (DR) bit logic "1", which indicates a character has been received by URBR, and logic "0" indicating all of the data in URBR or RCV FIFO has been read.	-

Table 12.5 Line Status Register Bits

LSR Bits	Logic 1	Logic 0
LSR(7) PE/FE/BI (FIFO mode)	Error	No error
LSR(6) Transmitter Empty(TEMT)	Empty	Not empty
LSR(5) Transmitter Holding Register Empty(THRE)	Empty	Not empty
LSR(4) Break Interrupt(BI)	Break	No break
LSR(3) Framing Error(FE)	Error	No error
LSR(2) Parity Error(PE)	Error	No error
LSR(1) Overrun Error(OE)	Error	No error
LSR(0) Data Ready(DR)	Ready	Not ready

(2) UMSR (READ/WRITE)

This 8-bit register provides current state of the control lines from modems or peripheral devices. In addition to this current state information; bits 7~ 4 can provide change information when a modem control input changes state. It will be reset to low when the Host reads the UMSR.

Bit	Description	Default
7	**Data Carrier Detect (DCD#):** Indicates the complement status of Data Carrier Detect input. If bit 4 of UMCR is 1, this bit is equivalent to OUT2 of the UMCR.	-
6	**Ring Indicator (RI#):** Indicates the complement to the RI# input. If bit 4 of UMCR is 1, this bit is equivalent to OUT1 in the UMCR.	-
5	**Data Set Ready (DSR#):** Indicates the modem is ready to provide received data to the serial channel receiver circuitry. If the serial channel is in the loop mode (bit 5 of UMCR is 1), this bit is equivalent to DTR# in the UMCR.	-
4	**Clear to Send (CTS#):** Indicates the complement of CTS# input. If the serial channel is in the loop mode (bit 4 of UMCR is 1), this bit is equivalent to RTS# in the UMCR.	-
3	**Delta Data Carrier Detect (DDCD):** Indicates that the DCD# input state has been changed since the last time read by the Host.	-
2	**Trailing Edge of Ring Indicator (TERI):** Indicates that the RI input state to the serial channel has been changed from a low to high since the last time read by the Host. The change to logic 1 doesn't activate the TERI.	-
1	**Delta Data Set Ready (DDSR):** A logic "1" indicates that the DSR# input to the serial channel has changed state since the last time it was read by the Host.	-
0	**Delta Clear to Send (DCTS):** This bit indicates that the CTS# input state to the serial channel has been changed since the last time it was read by the Host	-

Table 12.6 Modem Status Register Bits

Bit	Mnemonic	Logic 0
7	DCD#	Data Carrier Detect
6	RI#	Ring Indicator
5	DSR#	Data Set Ready
4	CTS#	Clear To Send
3	DDCD	Delta Data Carrier Detect
2	TERI	Trailing Edge of Ring Indicator
1	DDSR	Delta Data Set Ready
0	DCTS	Delta Clear to Send

12.4 Reset

Reset of UART should be held to an idle mode reset high for 500ns until initialization, which causes the following:

1. Initialization of the transmitter and receiver internal clock counters.
2. Resetting all bits of ULSR, (except ULSR(5) and ULSR(6), THRE and TEMT (they are set only by a hardware reset), all bits of UMCR and all corresponding discrete lines, memory and logic elements. Before resetting, UART remains in the idle mode until programmed.

Table 12.7 Reset Control of Register and Pinout Signals

Register/Signal	Reset Control	Reset Status
Interrupt Enable Register	Reset	All bits Low
Interrupt Identification Register	Reset	Bit 0 is high and bits 1-7 are low
FIFO Control Register	Reset	All bits Low
Line Control Register	Reset	All bits Low
Modem Control Register	Reset	All bits Low
Line Status Register	Reset	Bits 5, 6 are high, others are low.
Modem Status Register	Reset	Bits 7-4 input signals, bits 0-3 low
SOUT	Reset	High
RTS#, DTR#	Reset	High

12.5 Programming

Serial channel is programmed by control registers whose contents define the character length, number of stop bits, parity, baud are modem interface. Even though the control register can be written in any order, the UIER should be the last because it controls the interrupt enables. After the port is programmed, these registers can still be updated whenever the port is not transferring data.

12.5.1 Programming Sequence

UART module in Intelligent Peripheral Controller is compatible with standard 16550. The following is standard 16550 compatible component register access sequence.

For access URBR/UTBR:

1. Set bit 7 of the ULCR register to ‘0”
2. Access URBR/UTBR

For Access UIER:

1. Set bit 7 of the ULCR register to “0”
2. Access UIER

For Access UDLL/UDLM:

1. Set bit 7 of the ULCR register to “1”
2. Access UDLL/UDLM

12.6 Software Reset

This method allows returning to a completely known state without a system reset. It consists of writing the required data to the ULCR, UDLL, UDLM and UMCR. The ULSR and URBR must be read before enabling interrupts in order to clear any residual data or status bits which may be invalid for subsequent operations.

12.7 Clock Input Operation

The input frequency of the Serial Channel is 48MHz/26, not exactly 1.8432MHz.

12.8 FIFO Interrupt Mode Operation

(1) RCVR Interrupt

When bit 0 of UFCR and bit 0 of UIER are set to 1, the RCVR FIFO and receiver interrupts are enabled. The RCVR interrupt occurs under the following conditions:

a. The receive data available interrupt and the UIIR, receive data available indication, will be issued only if the FIFO has reached its programmed trigger level. They will be cleared as soon as the FIFO drops below its trigger level.
b. The receiver line status interrupt has higher priority than the received data available interrupt.
c. The time-out timer will be reset after receiving a new character or after the Host reads the RCVR FIFO whenever a time-out interrupt occurs. The timer will be reset when the Host reads one character form the RCVR FIFO.

RCVR FIFO timer-out Interrupt: By enabling RCVR FIFO and receiver interrupts, the RCVR FIFO time-out interrupt will occur under the following conditions:

a. It will occur only if there is at least one character in the FIFO whenever the period between the most recent received serial character and the most recent Host read from the FIFO is longer than four (4) consecutive character times.
b. The time-out timer will be reset after receiving a new character or after the Host reads the RCVR FIFO whenever any time-out interrupt occurs. The timer will be reset when the Host reads one character from the RCVR FIFO.

(2) XMIT Interrupt

By setting the bit 0 of UFCR and the bit 1 of UIER to high, the XMIT FIFO and transmitter interrupts are enabled, and the XMIT interrupt will occur as follows:

a. The transmitter interrupt will occur when the XMIT FIFO is empty, and it will be reset if the THR is written or the UIIR is read.
b. The transmitter FIFO empty indications will be delayed one character time minus the last stop bit time whenever the following conditions occurs: THRE=1 and there have not been at least two bytes in the transmitter FIFO at the same time since the last THRE=1. The transmitter interrupt after changing the bit 0 of UFCR will be immediate. Once it is enabled, the THRE indication is delayed 1 character time minus the last stop bit time.

The character time-out and RCVR FIFO trigger level interrupts are in the same priority order as the received data available interrupt. The XMIT FIFO empty is in the same priority as the transmitter holding register empty interrupt.

FIFO Polled Mode Operation [bit 0 of UFCR is 1, and bits 0, 1, 2, 3 of UIER or all are zero]. Either one or both XMIT and RCVR can be in this operation mode which the user program will check RCVR and XMIT status via the ULSR as described below:

LSR(0): Will be high whenever the RCVR FIFO contains at least one byte.
LSR(1) - LSR(4): Specifies that errors have occurred, Character error status is handled the same way as that in the interrupt mode. The IIR is not affected since IER(2)=0.
LSR(5): The XMIT FIFO empty indication.
LSR(6): XMIT FIFO and Shift register empty.
LSR(7): RCVR FIFO error indication.

There is no trigger level reached or time-out condition indicated in the FIFO Polled Mode.

12.9 CAUTION

Set bit3 of UMCR to enable interrupt generation.

Section 13 Parallel Port

13.1 Overview

The Parallel Port module supports an IBM AT, PS/2 compatible bi-directional parallel port (SPP), the Enhanced Parallel Port (EPP) and the Extended Capabilities Port (ECP). But **NO** DMA transfers are supported in this module.

13.2 Features

- Support SPP, EPP and ECP modes
- Standard mode – bi-directional SPP
- Enhanced mode – EPP v1.7 and EPP v1.9 compliant
- High speed mode – ECP, IEEE 1284 compliant
- Backdrive current reduction
- Printer power-on damage reduction
- No DMA mode is supported in ECP mode

13.3 Parallel Port Register Description

Host Connector Pins	SPP	EPP	ECP
1	STB#	WRITE#	nStrobe
2 - 9	PD0 - PD7	PD0 - PD7	PD0 - PD7
10	ACK#	INTR	nAck
11	BUSY	WAIT#	Busy PeriphAck (2)
12	PE	(NU) (1)	Perro nAckReverse (2)
13	SLCT	(NU) (1)	Select
14	AFD#	DSTB#	nAutoFd HostAck(2)
15	ERR#	(NU) (1)	nFault nPeriphRequest (2)
16	INIT#	(NU) (1)	nInit nReverseRequest (2)
17	SLIN#	ASTB#	nSelectIn

Notes: 1. NU: Not used
2. Fast mode
3. For more information, please refer to the IEEE 1284 standard.

Table 13.1 The Register List of Parallel Port

SPP/EPP Mode

Name	Address	Register Size(Bit)	Access Size(Bit)
Data Port Register	H'1000A000	8	8
Status Port Register	H'1000A002	8	8
Control Port Register	H'1000A004	8	8
EPP Address Port Register(*)	H'1000A006	8	8
EPP Data Port0 Register(*)	H'1000A008	8	8
EPP Data Port1 Register(*)	H'1000A00A	8	8
EPP Data Port2 Register(*)	H'1000A00C	8	8
EPP Data Port3 Register(*)	H'1000A00E	8	8

(*)These registers are only used in EPP mode.

ECP Mode

Name	Address	Register Size(Bit)	Access Size(Bit)
Data Port Register(data)	H'1000A000	8	8
ECP Address FIFO Register(ecpAFifo)	H'1000A000	8	8
Status Register(dsr)	H'1000A002	8	8
Control Register(dcr)	H'1000A004	8	8
Parallel Port Data FIFO(cFifo)	H'1000A010	8	8
Test FIFO(tFifo)	H'1000A010	8	8
Configuration Register A(cnfgA)	H'1000A010	8	8
Configuration Register B(cnfgB)	H'1000A012	8	8
Extended Control Register(ecr)	H'1000A014	8	8

Table 13.2 Bit Map of the EPP Registers

Register	Address	I/O	D0	D1	D2	D3	D4	D5	D6	D7	Mode
Data Port	H'1000A000	R/W	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	SPP/EPP
Status Port	H'1000A002	R	TMOUT	1	1	ERR#	SLCT	PE	ACK#	BUSY#	SPP/EPP
Control Port	H'1000A004	R/W	STB	AFD	INIT	SLIN	IRQE	PDDIR	1	1	SPP/EPP
EPP Address Port	H'1000A006	R/W	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	EPP
EPP Data Port0	H'1000A008	R/W	P0D0	P0D1	P0D2	P0D3	P0D4	P0D5	P0D6	P0D7	EPP
EPP Data Port1	H'1000A00A	R/W	P1D0	P1D1	P1D2	P1D3	P1D4	P1D5	P1D6	P1D7	EPP
EPP Data Port2	H'1000A00C	R/W	P2D0	P2D1	P2D2	P2D3	P2D4	P2D5	P2D6	P2D7	EPP
EPP Data Port3	H'1000A00E	R/W	P3D0	P3D1	P3D2	P3D3	P3D4	P3D5	P3D6	P3D7	EPP

13.3.1 SPP and EPP Modes

(1) Data Port Register

This is a bi-directional 8-bit data register. The direction of data flow is determined by bit 5 of the control register. It forwards directions when the bit is low and reverses when the bit is high.

(2) Status Port Register

Table 13.3 Status Port Register Description

Bit	Description	Default
7	BUSY#: Inverse of printer BUSY signal. A logic “0” means that the printer is busy and cannot accept another character. A logic “1” means that the printer is ready to accept another character.	-
6	ACK#: Printer acknowledge. A logic “0” means that the printer has received a character and is ready to accept another. A logic “1” means that it is still processing the last character.	-
5	PE: Paper end, a logic “1” indicates a paper end.	-
4	SLCT: Printer selected, a logic “1” means that the printer is on line.	-
3	ERR#: Printer error, a logic “0” means that an error has been detected.	-
2, 1	Reserved: These bits are always “1” when read.	11
0	TMOUT: This bit is valid only in EPP mode and indicates that a 10 ms time out has occurred in EPP operation. If in other mode, this bit is always logic “1” when read.	-

(3) Control Port Register

This register provides all output signals to control the printer. The register can be read and written.

Table 13.4 Control Port Register Description

Bit	Description	Default
7, 6	Reserved: These bits are always “1” when read.	11
5	PDDIR: Data register direction control.	0
4	IRQE: Interrupt request enable	0
3	SLIN: Inverse of SLIN# pin. Set this bit to “1” to select the printer.	0
2	INIT: Initiate printer	0
1	AFD: Inverse of the AFD# pin.	0
0	STB: Inverse of the STB# pin. This pin controls the data strobe signal to printer.	0

(4) EPP Address Port Register

The EPP Address Port is only available in EPP mode. When the host writes to this port, the contents of D0 - D7 are buffered and output to PD0 - PD7 as an EPP Address Write cycle. When the host reads from this port, the contents of PD0 - PD7 are read as an EPP Address Read cycle.

(5) EPP Data Port 0 - 3

The EPP Data Port 0 - 3 are only available in EPP mode. When the host writes to these ports, the contents of D0 - D7 are buffered and output to PD0 - PD7 as an EPP DATA Write cycle. When the host reads from these ports, the contents of PD0 - PD7 are buffered and output to D0-D7, and causes an EPP DATA Read Cycle.

13.3.2 ECP Mode

This mode is both software and hardware compatible with that of the existing parallel ports, allowing ECP to be used as a standard LPT port when ECP is not required. It provides an automatic high-burst-bandwidth channel that supports ECP in both forward and reverse directions. A 16-byte FIFO is implemented in both forward and reverse directions to smooth data flow and improve the maximum bandwidth requirement. The port supports an automatic handshaking for the standard parallel port to improve compatibility and increase the speed of mode transfer. It also supports run-length encoded (RLE) decompression in hardware. Compression is accomplished by counting identical bytes and transmitting an RLE byte that indicates how many times a byte is repeated.

Table 13.5 Bit Map of the ECP Mode Register

Register	D7	D6	D5	D4	D3	D2	D1	D0
data	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
ecpAFifo	Addr/RLE	Address or RLE field						
dsr	nBusy	nAck	PError	Select	nFault	1	1	1
dcr	1	1	PDDIR	IRQE	SelectIn	nInit	AutoFd	Strobe
cFifo	Parallel Port Data FIFO							
ecpDFifo	ECP Data FIFO							
tFifo	Test FIFO							
cnfgA	0	0	0	1	0	0	0	0
cnfgB	0	intrValue	0	0	0	0	0	0
ecr	mode			nErrIntrEn		ServiceIntr	full	empty

Table 13.6 ECP Register Definition

Name	Address	I/O	ECP Mode	Function
data	H'1000A000	R/W	000-001	Data Register
ecpAFifo	H'1000A000	R/W	011	ECP FIFO (Address)
dsr	H'1000A002	R/W	All	Status Register
dcr	H'1000A004	R/W	All	Control Register
cFifo	H'1000A010	R/W	010	Parallel Port Data FIFO
ecpDFifo	H'1000A010	R/W	011	ECP FIFO (DATA)
tFifo	H'1000A010	R/W	110	Test FIFO
cnfgA	H'1000A010	R	111	Configuration Register A
cnfgB	H'1000A012	R/W	111	Configuration Register B
ecr	H'1000A014	R/W	All	Extended Control Register

Table 13.7 ECP Mode Description

Mode ECR[7:5]	Description
000	**Standard Parallel Port Mode:** The FIFO is reset and the direction bit dcr(5) is always zero (forward direction) in this mode.
001	**PS/2 Parallel Port Mode:** It is similar to the SPP mode, except that the dcr(5) is READ/WRITE. When dcr(5) is one, the PD bus is tristate. Reading the data port returns the value on the PD bus instead of the value of the data register.
010	**Parallel Port FIFO Mode:** This mode is similar to the 000 mode, except that the Host writes the data bytes to the FIFO. The FIFO data is then automatically sent to the peripheral using the standard parallel port protocol. This mode is only valid in the forward direction (dcr(5)=0).
011	**ECP Parallel Port Mode:** In the forward direction, bytes placed into the ecpDFifo and ecpAFifo are placed in a single FIFO and automatically sent to the peripheral under ECP protocol. In the reverse direction, bytes are sent tot the ecpDFifo from ECP port.
100, 101	Reserved, not defined.
110	**Test Mode:** In this mode, the FIFO may be read from or written to, but it cannot be sent to peripheral.
111	**Configuration Mode:** In this mode, the cnfgA and cnfgB registers are accessible at 0x400 and 0x401.

(1) Data Port Register (Address H'1000A000, Mode 000 and 001)

Its contents will be cleared by a RESET. In a WRITE operation, the contents are then sent without being inverted to PD0 - PD7. The contents of data ports are read and sent to the host in a READ operation.

(2) ECP Address FIFO Register (ecpAFifo)

Any data byte written to this port is placed in the FIFO and tagged as an ECP Address/RLE. The hardware then sends this data automatically to the peripheral. The operation of this port is only valid in forward direction (dcr(5)=0).

(3) Device Status Register (dsr) (Address H'1000A002, Mode All)

Address: H'1000A002, Mode All

Bit	7	6	5	4	3	2	1	0
Bit Name	nBusy	nAck	PError	Select	nFault	Reserved	Reserved	Reserved
Initial Value	0	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

Bit	Description	Default
7	**nBusy :** Inverse of the Busy input pin	0
6	**nAck :** The state of the nAck input pin.	1
5	**PError:** The state of the PError input pin.	1
4	**Select:** The state of the Select input pin.	1
3	**nFault :** The state of the nFault input pin.	1
2 - 0	**Reserved:** These bits are always “1”.	111

(4) Device Control Register (dcr) (Address H'1000A004, Mode All)

Address: H'1000A004, Mode All

Bit	7	6	5	4	3	2	1	0
Bit Name	Reserved	Reserved	PDDIR	IRQE	SelectIn	nInit	AutoFd	Strobe
Initial Value	1	1	0	0	-	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
7, 6	**Reserved:** These bits are always “1”.	11
5	**PDDIR:** Except in modes 000 and 010, setting this bit low means that the PD bus is in output operation; setting it high, in input operation. This bit will be forced low in mode 000.	0
4	**IRQE:** Setting this bit high enables interrupt request from peripheral to host due to a rising edge of the nAck input.	0
3	**SelectIn:** It is inverted and output to the pin nSelectIn.	-
2	**Ninit:** It is output to the pin nInit without inversion.	0
1	**AutoFd:** It is inverted and output to the pin nAutoFd.	0
0	**Strobe:** It is inverted and output to the pin nStrobe.	0

(5) Parallel Port Data FIFO (cFifo) (Address H'1000A010, Mode 010)

Bytes written to this FIFO are sent by a hardware handshake to the peripheral according to the standard parallel port protocol. This operation is only defined for the forward direction.

(6) ECP Data FIFO (ecpDFifo) (Address H'1000A010, Mode 011)

When the direction bit dcr(5) is “0”, bytes written to this FIFO are sent by a hardware handshaking to the peripheral according to the ECP parallel port protocol. When the dcr(5) is “1”, data bytes from the peripheral to this FIFO are read in an automatic hardware handshaking. The host can acquire these bytes by performing READ operations.

(7) Test FIFO mode (tFifo) (Address H'1000A010, Mode 110)

The host may operate READ/WRITE transfer to this FIFO in any direction. Data in this FIFO will be displayed on the PD bus without using hardware protocol handshaking. The tFifo will not accept new data after it is full. Performing a READ from an empty fFifo causes the last data byte to return.

(8) Configuration Register A (cnfgA) (Address H'1000A010, Mode 111)

This register is a read only register. When read, the returned data is valued at 10h, it indicates to the system that this is an 8-bit implementation. (Pword = 1byte)

(9) Configuration Register B (cnfgB) (Address H'1000A012, Mode 111)

Address: H'1000A012, Mode 111

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	Compress	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	**Description**	**Default**
7	**Compress:** This bit is read only. It is in a low level during a READ operation. This means that this controller does not support hardware RLE compression. It does support hardware de-compression!	0
6	Reserved.	1
5 - 0	**Reserved:** These bits are ‘0’s during a READ operation.	3Fh

(10) Extended Control Register (ECR) (Address H'1000A014, Mode All)

Address: H'1000A014, Mode All

Bit	7	6	5	4	3	2	1	0
Bit Name	MODEB2	MODEB1	MODEB0	nErrIntrEn	Reserved	ServiceIntr	Full	Empty
Initial Value	0	0	0	1	0	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
7 - 5	**Mode:** These bits are used for READ/WRITE and Mode selection.	0
4	**NerrIntrEn:**	1
	1: Disables the interrupt generated on the asserting edge of the nFault input.	
	0: Enables the interrupt pulse on the asserting edge of the nFault. An interrupt pulse will be generated if nFault is asserted, or if this bit is written from one to zero in the low level nFault	
3	Reserved.	0
2	**ServiceIntr:**	1
	1: Disables all service interrupts.	
	0: Enables the service interrupts. This bit will be set to “1” by hardware when one of the three service interrupts has occurred. Writing “1” to this bit will not generate an interrupt.	
1	Full	0
	1: The FIFO is full and cannot accept another byte.	
	0: The FIFO has one free data byte space at least.	
0	Empty	1
	1: The FIFO is empty.	
	0: The FIFO contains at least one data byte.	

(11) Mode Switching Operation

In programmed I/O control (mode 000 or 001), P1284 negotiation and all other tasks happening before data is transferred, and are controlled by software. Setting mode to 011 or 010 will cause the hardware to perform an automatic control-line handshaking, and transfer information between FIFO and the ECP port.

From mode 000 or 001, any other mode may be immediately switched to or from the other mode. To change direction, the mode must first be set to 001.

In extended forward mode, FIFO must be cleared and all signals deasserted before returning to mode 000 or 001. In ECP reverse mode, all data must be read from the FIFO before returning to mode 000 or 001. Discarded data is usually accumulated during ECP reverse handshaking, as when mode changed during a data transfer. If the above condition is satisfied, nAutoFd will be deasserted regardless of the transfer state. To avoid bugs during handshaking signals, the guidelines above must be followed.

(12) Software Operation (ECP)

Before ECP operation can begin, it is first necessary for the Host to switch the mode to 000 to negotiate with the parallel port. Host determines whether peripheral supports ECP protocol during the process.

After the negotiation is completed, the mode is set to 011 (ECP). To enable the drivers, direction must be set to “0”. Both strobe and autoFd are set to “0”, causing the nStrobe and nAutoFd signals to be deasserted.

All FIFO data transfers are Pword wide and Pword aligned. Permitted only in the forward direction, address/RLE transfers are byte-wide. ECP address/RLE bytes may be sent automatically by writing the ecpAFifo. Similauly, data Pwords may be sent automatically via ecpDFifo.

To change directions, Host switches mode to 001. It then negotiates either the forward or reverse channel, sets direction to one or zero, and finally switches mode to 001. If the direction is set to “1”, the hardware performs a handshaking form each ECP data byte READ, and tries to fill the FIFO. At this time, Pwords may be read from the expDFifo while it retains data. It is also possible for the hardware to performs ECP transfers by handshaking with individual bytes under program control in mode=001 or 000, even though this is a comparatively time-consuming approach.

(13) Interrupts

When any of the following states are reached, it is necessary to generate an interrupt.

1. serviceIntr = 0, direction = 0, and the number of Pwords in FIFO is greater than or equal to writeIntrThreshold.
2. serviceIntr = 0, direction = 1, and the number full Pwords in the FIFO is greater than or equal to READIntrThreshold.
3. nErrIntrEn = 0 and nFault goes from high to low or when nErrIntrEnor is set from “1” to “0” and nFault is asserted.
4. ackIntEn = 1. In current implementations using existing parallel ports, the generated interrupt may be either edge or level type, making it “ISA-friendly”.

Section 14 Serial CODEC Interface

14.1 Overview

This serial CODEC interface provides an interface between the system and the CODEC that is CS4218 or CS4271 or compatible with AC97. The major functions of the interface are transmitting D/A data from system to CODEC and receiving A/D data from CODEC to system.

14.1.1 Features

- Support for CS4218,CS4271 and AC97 Version1.03 and 2.0 CODECs
- Full-duplex data transfer between CODEC and this interface
- Both PIO and DMA modes are supported for communication with system
- Provides SM3 Slave mode for communication with CS4218 or CS4271, and SM3 Master mode for CS4218 only

14.1.2 Block Diagram

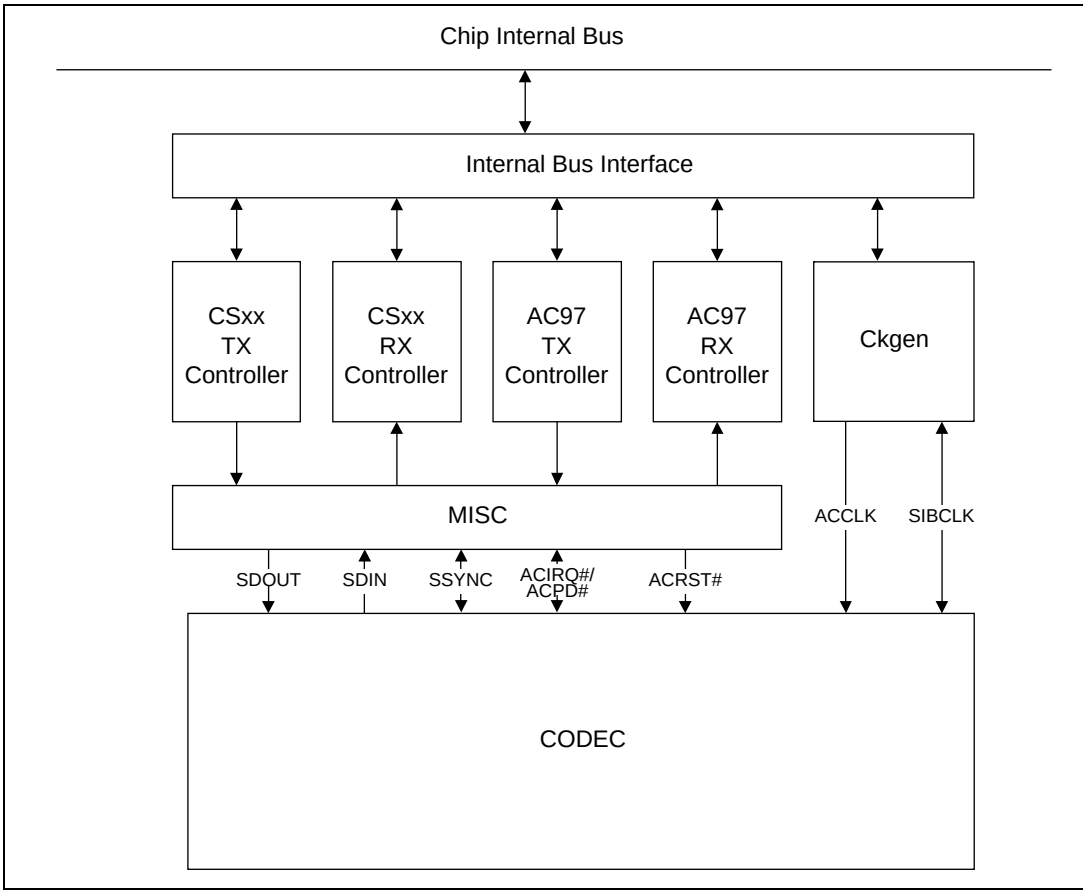


Figure 14.1 The Block Diagram of Serial CODEC Interface

Table 14.1 Pin Function of Serial CODEC Interface Module

Name	I/O	Pin Function
ACCLK	O	Master Clock
SIBCLK	I/O	Serial Port Bit Clock
SIBDIN	I	Serial interface data input: Data is transferred from CODEC to this module
SIBDOUT	O	Serial interface data output: Data is transferred from this module to CODEC
SIBSYNC	I/O	Serial interface sync: This high active pin indicates that a frame starts from the rising edge of the next SCLK.
ACIRQ#/ ACPD#/ PME#	I/O/O	When the connected CODEC is CS4271, the pin is an input and is used to request an interrupt via the interface. But when the connected CODEC is CS4218, the pin is an output used to power down CS4218. When CODEC is AC97 CODEC, this pin is an output used to indicate an Power Management Event has happened.
ACRST#	O	Reset: This pin, when low, resets the CS4218 to a known state.

14.2 Register Description

This Serial CODEC interface contains registers shown in the table below.

Table 14.2 Registers of SCDI

Register or Buffer	Function	Access Size	Address
TDR	Transmit Data Register for CS4218 or CS4271	32 bits	H'1000C000
RDR	Receive Data Register for CS4218 or CS4271	32 bits	H'1000C004
CR	Control Register	16 bits	H'1000C008
SR	Status Register for CS4218 or CS4271	16 bits	H'1000C00C
FSR	Frequency Select Register for CS4218	16 bits	H'1000C010
CAR	Command Address Register	32 bits	H'1000C020
CDR	Command Data Register	32 bits	H'1000C024
PCML	PCM Left channel data	32 bits	H'1000C028
PCMR	PCM Right channel data	32 bits	H'1000C02C
LINE1	Line 1 Data Register	32 bits	H'1000C030
PCMC	PCM Center channel	32 bits	H'1000C034
PCMLS	PCM Left Surround channel	32 bits	H'1000C038
PCMRS	PCM Right Surround channel	32 bits	H'1000C03C

Table 14.2 Registers of SCDI (cont'd)

Register or Buffer	Function	Access Size	Address
PCMLFE	PCM Low Frequency Enhance	32 bits	H'1000C040
LINE2	Line 2 data Register	32 bits	H'1000C044
HSET	HSET Data Register	32 bits	H'1000C048
IOCS	IO Control or Status register	32 bits	H'1000C04C
ATIER	AC97 TX Interrupt Enable Register	32 bits	H'1000C050
ATSR	AC97 TX Status Register	32 bits	H'1000C054
ARIER	AC97 RX Interrupt Enable Register	32 bits	H'1000C058
ARSR	AC97 RX Status Register	32 bits	H'1000C05C
ACR	AC97 Control Register	32 bits	H'1000C060
ATAGR	AC97 TAG Register	32 bits	H'1000C064
SRAR	Slot Request Active Register	16 bits	H'1000C068

14.2.1 Transmit Data Register (TDR)

TDR, a 32-bit Write only register, is used as a channel to write data to TX FIFO. TDR is not initialized.

Bit	31	30	29	28	27	10	9	8
Bit Name	TD31	TD30	TD29	TD28	TD27	TD26	TD25	TD24
Initial Value	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W

Bit	23	22	13	12	11	10	9	8
Bit Name	TD23	TD22	TD21	TD20	TD19	TD18	TD17	TD16
Initial Value	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8
Bit Name	TD15	TD14	TD13	TD12	TD11	TD10	TD9	TD8
Initial Value	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W

Bit	7	6	5	4	3	2	1	0
Bit Name	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0
Initial Value	-	-	-	-	-	-	-	-
R/W	W	W	W	W	W	W	W	W

Bit	Description	Default
31 - 0	TX Data (TD): MSB is the first bit to be transmitted to Serial Data Output Port.	-

14.2.2 Receive Data Register (RDR)

RDR, a 32-bit Read only register, is used as a channel to read data from RX FIFO. RDR is not initialized.

Bit	31	30	29	28	27	26	25	24
Bit Name	RD31	RD30	RD29	RD28	RD27	RD26	RD25	RD24
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	23	22	21	20	19	18	17	16
Bit Name	RD23	RD22	RD21	RD20	RD19	RD18	RD17	RD16
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8
Bit Name	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Receive Data Register (RDR) [con'd]

Bit	Description	Default
31 - 0	**Receive Data (RD):** MSB is the first bit to be received from Serial Data Input Port.	-

14.2.3 Control Register (CR)

CR is a 16-bit Read/Write register that is used to control Serial CODEC Interface. Bits 15 - 14 of this register are reserved. The other bits of this register are initialized to 0 at reset. CR is not initialized in standby mode.

Bit	15	14	13	12	11	10	9	8
Bit Name	reserved	reserved	DMAEN	SL18	CDRT	WMRT	AC97S	SWR
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Bit Name	PU	MS	ST	CRE	FTF	TXEN	FRF	RXEN
Initial Value	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

Bit	Description	Default
15 - 14	Reserved	-
13	**DMA mode Enable (DMAEN:** If this bit is set to 1, DMA mode for CS4218 or CS4271 is enabled.	0
12	**Select CS4218 (SL18):** If this bit is set to 1, CS4218 interface is selected. Otherwise, CS4271 interface is selected.	0
11	**Cold Reset for AC97 (CDRT):** Writes 1 to this bit will cause Cold AC97 Reset. This bit is always read 0.	0
10	**Warm Reset for AC97 (WMRT):** Writing 1 to this bit will cause Warm AC97 Reset. This bit is always read 0.	0
9	**AC97 Select (AC97S)** 1: Supports AC97 0: Supports CS4218 or CS4271	0
8	**Software Reset (SWR):** Setting 1 to this bit will reset the serial interface to an initialized state. This bit is always read 0.	0
7	**Power Up (PU):** (for CS4218 only): 1: Powers up CS4218 0: Pulls down PWDN input pin of CS4218 to cause it to enter power down mode.	0

Control Register (CR) [cont'd]

Bit	Description	Default
6	**Mode Select: (MS):** (for CS4218 or CS4271) 1:the serial CODEC interface is master. 0:the serial CODEC interface is slave. MS must be 1 for CS4271	0
5	**Start Transfer (ST):** (for SM3-slave mode or AC97): Writing 1 to this bit will start transmitting or receiving data. Writing 0 has no meaning. This bit is always read 0.	0
4	**Continue Run Enable (CRE):** (for SM3-slave mode or AC97) 1: SSYNC continues to be generated after transfer starts 0: SSYNC generation is not to be continued.	0
3	**Flush TX FIFO (FTF):** Setting this bit will flush TX FIFO, and resets the TX FIFO pointer. Always read 0.	0
2	**TX Enable (TXEN)** 1: Enables TX done interrupt and the control of transmit data path. 0: Disabled	0
1	**Flush RX FIFO (FRF):** Setting this bit will flush RX FIFO, and reset the RX FIFO pointer. Always read 0.	0
0	**RX Enable(RXEN)** 1: Enables RX done interrupt and the control of receive data path. 0: Disabled	0

14.2.4 Status Register (SR)

SR is a 16-bit Read only register that is used to reflect the status of this Serial CODEC Interface when the CODEC is CS4271 or CS4218. If the CODEC is AC97 CODEC, this register has no meaning. Bit 15 and bits 7 - 6 of this register are reserved. The other bits of this register are initialized to 0 at reset. SR is not initialized in STANDBY mode.

Bit	15	14	13	12	11	10	9	8
Bit Name	reserved	IR71	TNF	TFS1	TFS0	TFU	TFO	TDI
Initial Value	0	0	1	0	0	0	0	0
R/W	-	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	RNE	RFS1	RFS0	RFU	RFO	RDI
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	R	R	R	R	R	R

Status Register (SR) [cont'd]

Bit	Description	Default
15	Reserved	0
14	**IRQ from CS4271 (IR71):** This bit 1 indicates that CS4271 issues interrupt request.	0
13	**TX FIFO Not Full Flag (TNF)** 0: TX FIFO is full. 1: TX FIFO is not full.	1
12 - 11	**TX FIFO Status(TFS)** RFS[1:0] FIFO-1 FIFO-0 00 empty empty 01 empty not empty 10 not empty empty 11 not empty not empty	0
10	**TX FIFO under run (TFU):** This bit 1 indicates that TX FIFO is underrun. Writing 1 to this bit will clear this bit.	0
9	**TX FIFO over run (TFO):** This bit 1indicates that TX FIFO is overrun. Writing 1 to this bit will clear this bit.	0
8	**TX Done Interrupt (TDI):** 1 indicates that one block of Transmitting FIFO has been written to CODEC. Writing 1 to this bit will clear this bit.	0
7, 6	Reserved	-
5	**RX FIFO Not Empty Flag (RNE)** 0: RX FIFO is empty. 1: RX FIFO is not empty.	0
4, 3	**RX FIFO Status(RFS)** RFS[1:0] FIFO-1 FIFO-0 00 not full not full 01 not full full 10 full not full 11 full full	0
2	**RX FIFO under run (RFU):** This bit 1 indicates that RX FIFO is underrun Writing 1 to this bit will clear this bit.	0
1	**RX FIFO over run (RFO):** This bit 1 indicates that RX FIFO is overrun. Writing 1 to this bit will clear this bit.	0
0	**RX Done Interrupt (RDI):** This bit 1 indicates that one block of receiving FIFO has been filled by CODEC. Writing 1 to this bit will clear this bit.	0

14.2.5 Frequency Select Register

FSR, a 16-bit Read/Write register, is used to select frequency-sampling rate of CS4218 when SM3 slave mode is selected. Bits 15 - 3 in this register are reserved. The other bits in this register are initialized to 0 at reset. FSR is not initialized in STANDBY mode.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	reserved	reserved	reserved	reserved	reserved	FS2	FS1	FS0
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	R/W	R/W	R/W

Bit	**Description**	**Default**
15 - 3	Reserved	0
2 - 0	**Frequency Select (FS)** 000: 8kHz Frequency Sample Rate of CS4218 or CS4271 001: 9.6kHz 010:12kHz 011:16k 100:24k 101:48k 110,111:reserved	0

14.2.6 Command/Status Address Register (CSAR)

CSAR, a 32-bit Read/Write register, is a channel via which the system can write command address to CODEC or read status address from CODEC. Bits 31-20 and bits 11-0 are reserved. The other bits are initialized to 0 at reset. CAR is not initialized in STANDBY mode.

Bit	31	30	29	28	27	26	25	24
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	23	22	21	20	19	18	17	16
Bit Name	-	-	-	-	RW	CA6/SA6	CA5/SA5	CA4/SA4
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8
Bit Name	CA3/SA3	CA2/SA2	CA1/SA1	CA0/SA0	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	-	-	-	-

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	-

Bit	Description	Default
31 - 20	Reserved	-
19	**Read/Write Command (RW)** 1=read, 0=write	0
18 - 12	**Control Register Address 6-0 (CA6-CA0)/Status Address 6-0 (SA6-SA0):** When this register is written, these bits are control register address. Control register has 64 16-bit locations, addressed on even byte boundaries. When this register is read, these bits are the address of the register which data is being returned.	0
11 - 0	Reserved	0

14.2.7 Command/Status Data Register (CSDR)

CSDR, a 32-bit Read/Write register, is a channel via which the system can write command data to CODEC or read status data from CODEC. Bits 31-20 and bits 3-0 are reserved. The other bits are initialized to 0 at reset. CDR is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	CD15/ SD15	CD14/ SD14	CD13/ SD13	CD12/ SD12
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	CD11/ SD11	CD10/ SD10	CD9/ SD9	CD8/ SD8	CD7/ SD7	CD6/ SD6	CD5/ SD5	CD4/ SD4
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	CD3/ SD3	CD2/ SD2	CD1/ SD1	CD0/ SD0	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	-	-	-	-

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 4	**Command/Status Data 15-0 (CD15-0):** When this register is written, these bits will be written to the first empty entry of CDR TX FIFO and then transmitted to the connected CODEC in sequence. When this register is read, the first non-empty entry of CDR RX FIFO will be read. The data in this register reflects the status of the connected CODEC.	0
3 - 0	Reserved	0

14.2.8 PCM Playback/Record Left Channel (PCML)

PCML, a 32-bit Read/Write register, is a channel via which the system can write PCM Playback Left Channel data to CODEC or read PCM Record Left Channel data from CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. PCML is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, PCM Playback Left Channel Data will be written to the first empty entry of PCML TX FIFO and then transmitted to the connected CODEC in sequence. When this register is read, the first non-empty entry of PCML RX FIFO is read. PCML RX FIFO stores PCM Record Left Channel data received from the connected CODEC.	0

14.2.9 PCM Playback/Record Right Channel (PCMR)

PCMR, a 32-bit Read/Write register, is a channel via which system can write PCM Playback Right Channel data to CODEC or read PCM Playback Right Channel data from CODEC. Bits 31-20 are reserved. Bits 19-0 are initialized to 0 at reset. PCMR is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty 0 entry of PCMR TX FIFO and then transmitted to the connected CODEC. When this register is read, the first non-empty entry of PCMR RX FIFO will be read. PCMR RX FIFO stores the PCM Record Right Channel data from the connected CODEC.	

14.2.10 Line 1 Data Register (LINE1)

LINE1, a 32-bit Read/Write register, is a channel via which the system can write Line1 DAC data to CODEC or read Line1 ADC data from CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. LINE1 is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty entry of LINE1 TX FIFO and then transmitted to the connected CODEC. When this register is read, the data in the first non-empty entry of LINE1 RX FIFO will be read. LINE1 RX FIFO stores Line 1 ADC data from the connected CODEC.	0

14.2.11 PCM Center Playback/MIC ADC Channel (PCMC)

PCMC, a 32-bit Read/Write register, is a channel via which the system can write PCM Center Playback data to CODEC or read MIC ADC data from CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. PCMC is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty 0 entry of PCMC TX FIFO and then be transmitted to the connected CODEC. When this register is read, the data in the first non-empty entry of PCMC RX FIFO will be read. PCMC RX FIFO stores the MIC ADC data received from the connected CODEC.	

14.2.12 PCM Left Surround Channel Data Register (PCMLS)

PCMLS, a 32-bit Write Only register, is a channel via which the system can write PCM Left Surround Playback data to CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. PCMLS is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	W	W	W	W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty entry of PCMLS TX FIFO and then be transmitted to the connected CODEC.	0

14.2.13 PCM Right Surround Channel Data Register (PCMRS)

PCMRS, a 32-bit Write Only register, is a channel via which the system can write PCM Right Surround Playback data to CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. PCMRS is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	W	W	W	W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty 0 entry of PCMRS TX FIFO and then transmitted to the connected CODEC.	

14.2.14 PCMLFE Data Register (PCMLFE)

PCMLFE, a 32-bit Write Only register, is a channel via which the system can write PCMLFE data to CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. PCMLFE is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	W	W	W	W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty entry of PCMLFE TX FIFO and then be transmitted to the connected CODEC.	0

14.2.15 Line 2 Channel Data Register (LINE2)

LINE2, a 32-bit Read/Write register, is a channel via which the system can write Line 2 DAC data to CODEC or read Line2 ADC data from CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. LINE2 is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty 0 entry of LINE 2 TX FIFO and then be transmitted to the connected CODEC. When this register is read, the first non-empty entry of LINE2 RX FIFO will be read. LINE2 RX FIFO stores Line 2 ADC data from the connected CODEC.	

14.2.16 HSET Data Register (HSET)

HSET, a 32-bit Read/Write register, is a channel via which the system can write HSET DAC data to CODEC or read HSET ADC data from CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. HSET is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty entry of HSET TX FIFO and then be transmitted to the connected CODEC. When this register is read, the first non-empty entry of HSET RX FIFO will be read. HSET RX FIFO stores HSET ADC data from the connected CODEC.	0

14.2.17 IO Control/Status Data Register (IOCS)

IOCS, a 32-bit Read/Write register, is a channel via which the system can write IO Control data to CODEC or read IO Status data from CODEC. Bits 31-20 are reserved. The other bits are initialized to 0 at reset. IOCS is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	-	-	-	-	D19	D18	D17	D16
Initial Value	-	-	-	-	0	0	0	0
R/W	-	-	-	-	R/W	R/W	R/W	R/W

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Initial Value	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

Bit	**Description**	**Default**
31 - 20	Reserved	-
19 - 0	**Data 19-0 (D19-0):** When this register is written, these bits will be written to the first empty entry of IOCS TX FIFO and then be transmitted to the connected CODEC. When this register is read, the first non-empty entry of IOCS RX FIFO will be read. IOCS RX FIFO stores IO Status data received from the connected CODEC.	0

14.2.18 AC97 Transmit Interrupt Enable Register (ATIER)

ATIER, a 32-bit Read/Write register, is used to enable or disable AC97 TX Interrupt. Bits 31-30 are reserved. The other bits are initialized to 0 at reset. ATIER is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	PLTFRQIE	PRTFRQIE	L1TFRQIE	PCTFRQIE	PLSTFRQIE	PRSTFRQIE
Initial Value	-	-	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	PLFETFRQIE	L2TFRQIE	HTTFRQIE	IOCTFRQIE	PLTFOVIE	PRTFOVIE	L1TFOVIE	PCTFOVIE
Initial Value	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

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	PLSTFOVIE	PRSTFOVIE	PLFETFOVIE	L2TFOVIE	HTTFOVIE	IOCTFOVIE	PLTFUNIE	PRTFUNIE
Initial Value	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

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	L1TFUNIE	PCTFUNIE	PLSTFUNIE	PRSTFUNIE	PLFETFUNIE	L2TFUNIE	HTTFUNIE	IOCTFUNIE
Initial Value	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

Bit	**Description**	**Default**
31 - 30	Reserved	-
29	**PCML TX FIFO REQUEST Interrupt Enable (PLTFRQIE)** When this bit is 1, PCML TX FIFO Request is enabled. When this bit is 0, PCML TX FIFO Request is disabled.	0
28	**PCMR TX FIFO REQUEST Interrupt Enable (PRTFRQIE)** When this bit is 1, PCMR TX FIFO Request is enabled. When this bit is 0, PCMR TX FIFO Request is disabled.	0

AC97 Transmit Interrupt Enable Register (ATIER) [cont'd]

Bit	Description	Default
27	**Line 1 TX FIFO REQUEST Interrupt Enable (L1TFRQIE):** When this bit is 1, Line 1 TX FIFO Request is enabled. When this bit is 0, Line 1 TX FIFO Request is disabled.	0
26	**PCMC TX FIFO REQUEST Interrupt Enable (PCTFRQIE):** When this bit is 1, PCMC TX FIFO Request is enabled. When this bit is 0, PCMC TX FIFO Request is disabled.	0
25	**PCMLS TX FIFO REQUEST Interrupt Enable (PLSTFRQIE):** When this bit is 1, PCMLS TX FIFO Request is enabled. When this bit is 0, PCMLS TX FIFO Request is disabled.	0
24	**PCMRS TX FIFO REQUEST Interrupt Enable (PRSTFRQIE):** When this bit is 1, PCMRS TX FIFO Request is enabled. When this bit is 0, PCMRS TX FIFO Request is disabled.	0
23	**PCMLFE TX FIFO REQUEST Interrupt Enable (PLFETFRQIE):** When this bit is 1, PCMLFE TX FIFO Request is enabled. When this bit is 0, PCMLFE TX FIFO Request is disabled.	0
22	**Line2 TX FIFO REQUEST Interrupt Enable (L2TFRQIE):** When this bit is 1, Line 2 TX FIFO Request is enabled. When this bit is 0, Line 2 TX FIFO Request is disabled.	0
21	**HSET TX FIFO REQUEST Interrupt Enable (HTTFRQIE):** When this bit is 1, HSET TX FIFO Request is enabled. When this bit is 0, HSET TX FIFO Request is disabled.	0
20	**IO CTRL TX FIFO REQUEST Interrupt Enable (IOCTFRQIE):** When this bit is 1, IO CTRL TX FIFO Request is enabled. When this bit is 0, IO CTRL TX FIFO Request is disabled.	0
19	**PCML TX FIFO OVERRUN Interrupt Enable (PLTFOVIE):** When this bit is 1, PCML TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, PCML TX FIFO OVERRUN Interrupt is disabled.	0
18	**PCMR TX FIFO OVERRUN Interrupt Enable (PRTFOVIE):** When this bit is 1, PCMR TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, PCMR TX FIFO OVERRUN Interrupt is disabled.	0
17	**Line 1 TX FIFO OVERRUN Interrupt Enable (L1TFOVIE):** When this bit is 1, Line1 TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, Line1 TX FIFO OVERRUN Interrupt is disabled.	0
16	**PCMC TX FIFO OVERRUN Interrupt Enable (PCTFOVIE):** When this bit is 1, PCMC TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, PCMC TX FIFO OVERRUN Interrupt is disabled.	0

AC97 Transmit Interrupt Enable Register (ATIER) [cont'd]

Bit	Description	Default
15	**PCMLS TX FIFO OVERRUN Interrupt Enable (PLSTFOVIE):** When this bit is 1, PCMLS TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, PCMLS TX FIFO OVERRUN Interrupt is disabled.	0
14	**PCMRS TX FIFO OVERRUN Interrupt Enable (PRSTFOVIE):** When this bit is 1, PCMRS TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, PCMRS TX FIFO OVERRUN Interrupt is disabled.	0
13	**PCMLFE TX FIFO OVERRUN Interrupt Enable (PLFETFOVIE):** When this bit is 1, PCMLFE TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, PCMLFE TX FIFO OVERRUN Interrupt is disabled.	0
12	**Line2 TX FIFO OVERRUN Interrupt Enable (L2TFOVIE):** When this bit is 1, Line 2 TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, Line 2 TX FIFO OVERRUN Interrupt is disabled.	0
11	**HSET TX FIFO OVERRUN Interrupt Enable (HTTFOVIE):** When this bit is 1, HSET TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, HSET TX FIFO OVERRUN Interrupt is disabled.	0
10	**IO CTRL TX FIFO OVERRUN Interrupt Enable (IOCTFOVIE):** When this bit is 1, IO CTRL TX FIFO OVERRUN Interrupt is enabled. When this bit is 0, IO CTRL TX FIFO OVERRUN Interrupt is disabled.	0
9	**PCML TX FIFO UNDERRUN Interrupt Enable (PLTFUNIE):** When this bit is 1, PCML TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, PCML TX FIFO UNDERRUN Interrupt is disabled.	0
8	**PCMR TX FIFO UNDERRUN Interrupt Enable (PRTFUNIE):** When this bit is 1, PCMR TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, PCMR TX FIFO UNDERRUN Interrupt is disabled.	0
7	**Line 1 TX FIFO UNDERRUN Interrupt Enable (L1TFUNIE):** When this bit is 1, Line 1 TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, Line 1 TX FIFO UNDERRUN Interrupt is disabled.	0
6	**PCMC TX FIFO UNDERRUN Interrupt Enable (PCTFUNIE):** When this bit is 1, PCMC TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, PCMC TX FIFO UNDERRUN Interrupt is disabled.	0
5	**PCMLS TX FIFO UNDERRUN Interrupt Enable (PLSTFUNIE):** When this bit is 1, PCMLS TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, PCMLS TX FIFO UNDERRUN Interrupt is disabled.	0
4	**PCMRS TX FIFO UNDERRUN Interrupt Enable (PRSTFUNIE):** When this bit is 1, PCMRS TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, PCMRS TX FIFO UNDERRUN Interrupt is disabled.	0

AC97 Transmit Interrupt Enable Register (ATIER) [cont'd]

Bit	Description	Default
3	**PCMLFE TX FIFO UNDERRUN Interrupt Enable (PLFETFUNIE):** When this bit is 1, PCMLFE TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, PCMLFE TX FIFO UNDERRUN Interrupt is disabled.	0
2	**Line2 TX FIFO UNDERRUN Interrupt Enable (L2TFUNIE):** When this bit is 1, Line 2 TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, Line 2 TX FIFO UNDERRUN Interrupt is disabled.	0
1	**HSET TX FIFO UNDERRUN Interrupt Enable (HTTFUNIE):** When this bit is 1, HSET TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, HSET TX FIFO UNDERRUN Interrupt is disabled.	0
0	**IO CTRL TX FIFO UNDERRUN Interrupt Enable (IOCTFUNIE):** When this bit is 1, IO CTRL TX FIFO UNDERRUN Interrupt is enabled. When this bit is 0, IO CTRL TX FIFO UNDERRUN Interrupt is disabled.	0

14.2.19 AC97 TX FIFO Status Register

ATSR, a 32-bit Read Only register, is used to reflect the status of AC97 TX controller. Bits 31-30 are reserved. The other bits are initialized to 0 at reset. ATIER is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	-	-	PLTFRQ	PRTFRQ	L1TFRQ	PCTFRQ	PLSTFRQ	PRSTFRQ
Initial Value	-	-	0	0	0	0	0	0
R/W	-	-	R	R	R	R	R	R

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	PLFETFRQ	L2TFRQ	HTTFRQ	IOCTFRQ	PLTFOV	PRTFOV	L1TFOV	PCTFOV
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	PLSTFOV	PRSTFOV	PLFETFOV	L2TFOV	HTTFOV	IOCTFOV	PLTFUN	PRTFUN
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	L1TFUN	PCTFUN	PLSTFUN	PRSTFUN	PLFETFUN	L2TFUN	HTTFUN	IOCTFUN
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
31 - 30	Reserved	-
29	**PCML TX FIFO REQUEST (PLTFRQ):** 1 indicates that half of PCML TX FIFO is empty and must be filled by the system.	0
28	**PCMR TX FIFO REQUEST (PRTFRQ):** 1 indicates that half of PCMR TX FIFO is empty and must be filled by the system.	0
27	**Line 1 TX FIFO REQUEST (L1TFRQ):** 1 indicates that half of Line1 FIFO is empty and must be filled by the system.	0
26	**PCMC TX FIFO REQUEST(PCTFRQ):** 1 indicates that half of PCMC TX FIFO is empty and must be filled by the system.	0
25	**PCMLS TX FIFO REQUEST (PLSTFRQ):** 1 indicates that half of PCMLS TX FIFO is empty and must be filled by the system.	0
24	**PCMRS TX FIFO REQUEST (PRSTFRQ):** 1 indicates that half of PCMRS TX FIFO is empty and must be filled by the system.	0
23	**PCMLFE TX FIFO REQUEST (PLFETFRQ):** 1 indicates that half of PCMLFE FIFO is empty and must be filled by the system.	0
22	**Line2 TX FIFO REQUEST (L2TFRQ):** 1 indicates that half of Line2 FIFO is empty and must be filled by the system.	0
21	**HSET TX FIFO REQUEST (HTTFRQ):** 1 indicates that half of HSET TX FIFO is empty and must be filled by the system.	0
20	**IO CTRL TX FIFO REQUEST (IOCTFRQ):** 1 indicates that half of IO CTRL TX FIFO is empty and must be filled by the system.	0
19	**PCML TX FIFO OVERRUN (PLTFOV):** 1 indicates that PCML TX FIFO is overrun.	0
18	**PCMR TX FIFO OVERRUN (PRTFOV):** 1 indicates that PCMR TX FIFO is overrun.	0
17	**Line 1 TX FIFO OVERRUN (L1TFOV):** 1 indicates that Line 1 FIFO is overrun.	0
16	**PCMC TX FIFO OVERRUN(PCTFOV):** 1 indicates that PCMC TX FIFO is overrun.	0
15	**PCMLS TX FIFO OVERRUN (PLSTFOV):** 1 indicates that PCMLS FIFO is overrun.	0

AC97 TX FIFO Status Register [cont'd]

Bit	Description	Default
14	**PCMRS TX FIFO OVERRUN (PRSTFOV):** 1 indicates that PCMRS FIFO is overrun.	0
13	**PCMLFE TX FIFO OVERRUN (PLFETFOV):** 1 indicates that PCMLFE FIFO is overrun.	0
12	**Line2 TX FIFO OVERRUN (L2TFOV):** 1 indicates that Line2 FIFO is overrun.	0
11	**HSET TX FIFO OVERRUN (HTTFOV):** 1 indicates that HSET TX FIFO is overrun.	0
10	**IO CTRL TX FIFO OVERRUN (IOCTFOV):** 1 indicates that IO CTRL TX FIFO is overrun.	0
9	**PCML TX FIFO UNDERRUN (PLTFUN):** 1 indicates that PCML TX FIFO is underrun.	0
8	**PCMR TX FIFO UNDERRUN (PRTFUN):** 1 indicates that PCMR TX FIFO is underrun.	0
7	**Line 1 TX FIFO UNDERRUN (L1TFUN):** 1 indicates that Line1 FIFO is underrun.	0
6	**PCMC TX FIFO UNDERRUN(PCTFUN):** 1 indicates that PCMC TX FIFO is underrun.	0
5	**PCMLS TX FIFO UNDERRUN (PLSTFUN):** 1 indicates that PCMLS TX FIFO is underrun.	0
4	**PCMRS TX FIFO UNDERRUN (PRSTFUN):** 1 indicates that PCMRS TX FIFO is underrun.	0
3	**PCMLFE TX FIFO UNDERRUN (PLFETFUN):** 1 indicates that PCMLFE TX FIFO is underrun.	0
2	**Line2 TX FIFO UNDERRUN (L2TFUN):** 1 indicates that Line 2 TX FIFO is underrun.	0
1	**HSET TX FIFO UNDERRUN (HTTFUN):** 1 indicates that HSET TX FIFO is underrun.	0
0	**IO CTRL TX FIFO UNDERRUN (IOCTFUN):** 1 indicates that IO CTRL TX FIFO is underrun.	0

14.2.20 AC97 RX FIFO Interrupt Enable Register (ARIER)

ARIER, a 32-bit Write only register, is used to enable or disable AC97 RX Interrupt. Bits 31-23are reserved. The other bits are initialized to 0 at reset. ARIER is not initialized in standby mode.

Bit	31	30	29	28	27	26	25	24
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	23	22	21	20	19	18	17	16
Bit Name	-	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	L1RFRQIE	MICRFRQIE	L2RFRQIE
Initial Value	-	0	0	0	0	0	0	0
R/W	-	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8
Bit Name	HTRFRQIE	IOCSRFRQIE	PLRFOVIE	PRRFOVIE	L1RFOVIE	MICRFOVIE	L2RFOVIE	HTRFOVIE
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	7	6	5	4	3	2	1	0
Bit Name	IOCSRFOVIE	PLRFUNIE	PRRFUNIE	L1RFUNIE	MICRFUNIE	L2RFUNIE	HTRFUNIE	IOCSRFUNIE
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	Description	Default
31 - 23	Reserved	-
22	**Status Address Ready Interrupt Enable (STARYIE):** When this bit is set to 1, Status Address Ready Interrupt is enabled. Otherwise, Status Address Ready Interrupt is disabled.	0
21	**Status Data Ready Interrupt Enable (STDRYIE):** When this bit is set to 1, Status Data Ready Interrupt is enabled. Otherwise, Status Data Ready Interrupt is disabled.	0

AC97 RX FIFO Interrupt Enable Register (ARIER) [cont'd]

Bit	Description	Default
20	**PCML RX FIFO REQUEST Interrupt Enable (PLRFRQIE):** When this bit is set to 1, PCML RX FIFO Request Interrupt is enabled. When this bit is reset to 0, PCML RX FIFO Request Interrupt is disabled.	0
19	**PCMR RX FIFO REQUEST Interrupt Enable (PRRFRQIE):** When this bit is set to 1, PCMR RX FIFO Request Interrupt is enabled. When this bit is reset to 0, PCMR RX FIFO Request Interrupt is disabled.	0
18	**Line 1 RX FIFO REQUEST Interrupt Enable (L1RFRQIE):** When this bit is set to 1, Line 1 RX FIFO Request Interrupt is enabled. When this bit is reset to 0, Line 1 RX FIFO Request Interrupt is disabled.	0
17	**MIC RX FIFO REQUEST Interrupt Enable (MICRFRQIE):** When this bit is set to 1, MIC RX FIFO Request Interrupt is enabled. When this bit is reset to 0, MIC RX FIFO Request Interrupt is disabled.	0
16	**Line2 RX FIFO REQUEST Interrupt Enable (L2RFRQIE):** When this bit is set to 1, Line 2 RX FIFO Request Interrupt is enabled. When this bit is reset to 0, Line 2 RX FIFO Request Interrupt is disabled.	0
15	**HSET RX FIFO REQUEST Interrupt Enable (HTRFRQIE):** When this bit is set to 1, HSET RX FIFO Request Interrupt is enabled. When this bit is reset to 0, HSET RX FIFO Request Interrupt is disabled.	0
14	**IO CTRL/STA RX FIFO REQUEST Interrupt Enable (IOCSRFRQIE):** When this bit is set to 1, IO CTRL/STA RX FIFO Request Interrupt is enabled. When this bit is reset to 0, IO CTRL/STA RX FIFO Request Interrupt is disabled.	0
13	**PCML RX FIFO OVERRUN Interrupt Enable (PLRFOVIE):** When this bit is set to 1, PCML RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, PCML RX FIFO OVERRUN Interrupt is disabled.	0
12	**PCMR RX FIFO OVERRUN Interrupt Enable (PRRFOVIE):** When this bit is set to 1, PCMR RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, PCMR RX FIFO OVERRUN Interrupt is disabled.	0
11	**Line 1 RX FIFO OVERRUN Interrupt Enable (L1RFOVIE):** When this bit is set to 1, Line 1 RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, Line 1 RX FIFO OVERRUN Interrupt is disabled.	0
10	**MIC RX FIFO OVERRUN Interrupt Enable (MICRFOVIE):** When this bit is set to 1, MIC RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, MIC RX FIFO OVERRUN Interrupt is disabled.	0
9	**Line2 RX FIFO OVERRUN Interrupt Enable (L2RFOVIE):** When this bit is set to 1, Line 2 RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, Line 2 RX FIFO OVERRUN Interrupt is disabled.	0

AC97 RX FIFO Interrupt Enable Register (ARIER) [cont'd]

Bit	Description	Default
8	**HSET RX FIFO OVERRUN Interrupt Enable (HTRFOVIE):** When this bit is set to 1, HSET RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, HSET RX FIFO OVERRUN Interrupt is disabled.	0
7	**IO CTRL/STA RX FIFO OVERRUN Interrupt Enable (IOCSRFOVIE):** When this bit is set to 1, IO CTRL/STA RX FIFO OVERRUN Interrupt is enabled. When this bit is reset to 0, IO CTRL/STA RX FIFO OVERRUN Interrupt is disabled.	0
6	**PCML RX FIFO UNDERRUN Interrupt Enable (PLRFUNIE):** When this bit is set to 1, PCML RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, PCML RX FIFO UNDERRUN Interrupt is disabled.	0
5	**PCMR RX FIFO UNDERRUN Interrupt Enable (PRRFUNIE):** When this bit is set to 1, PCMR RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, PCMR RX FIFO UNDERRUN Interrupt is disabled.	0
4	**Line 1 RX FIFO UNDERRUN Interrupt Enable (L1RFUNIE):** When this bit is set to 1, Line 1 RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, Line 1 RX FIFO UNDERRUN Interrupt is disabled.	0
3	**MIC RX FIFO UNDERRUN Interrupt Enable (MICRFUNIE):** When this bit is set to 1, MIC RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, MIC RX FIFO UNDERRUN Interrupt is disabled.	0
2	**Line2 RX FIFO UNDERRUN Interrupt Enable (L2RFUNIE):** When this bit is set to 1, Line 2 RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, Line 2 RX FIFO UNDERRUN Interrupt is disabled.	0
1	**HSET RX FIFO UNDERRUN Interrupt Enable (HTRFUNIE):** When this bit is set to 1, HSET RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, HSET RX FIFO UNDERRUN Interrupt is disabled.	0
0	**IO CTRL/STA RX FIFO UNDERRUN Interrupt Enable (IOCSRFUNIE):** When this bit is set to 1, IO CTRL/STA RX FIFO UNDERRUN Interrupt is enabled. When this bit is reset to 0, IO CTRL/STA RX FIFO UNDERRUN Interrupt is disabled.	0

14.2.21 AC97 RX Status Register (ARSR)

ARSR, a 32-bit Read Only register, is used to reflect the status of AC97 RX controller. Bits 31-23 are reserved. The other bits are initialized to 0 at reset. ARSR is not initialized in STANDBY mode.

Bit	31	30	29	28	27	26	25	24
Bit Name	-	-	-	-	-	-	-	-
Initial Value	-	-	-	-	-	-	-	-
R/W	-	-	-	-	-	-	-	-

Bit	23	22	21	20	19	18	17	16
Bit Name	-	STARY	STDRY	PLRFRQ	PRRFRQ	L1RFRQ	MICRFRQ	L2RFRQ
Initial Value	-	0	0	0	0	0	0	0
R/W	-	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8
Bit Name	HTRFRQ	IOCSRFRQ	PLRFOV	PRRFOV	L1RFOV	MICRFOV	L2RFOV	HTRFOV
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	IOCSRFOV	PLRFUN	PRRFUN	L1RFUN	MICRFUN	L2RFUN	HTRFUN	IOCSRFUN
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
31 - 21	Reserved	-
22	**Status Address Ready (STARY):** When this bit is set to 1, it indicates that status address is ready.	0
21	**Status Data Ready (STDRY):** When this bit is set to 1, it indicates that status data is ready.	0
20	**PCML RX FIFO REQUEST (PLRFRQ):** When this bit is set to 1, it indicates that half of PCML RX FIFO is full and must be cleared by the system.	0

Bit	Description	Default
19	**PCMR RX FIFO REQUEST (PRRFRQ):** When this bit is set to 1, it indicates that half of PCMR RX FIFO is full and must be cleared by the system.	0
18	**Line 1 RX FIFO REQUEST (L1RFRQ):** When this bit is set to 1, it indicates that half of Line 1 RX FIFO is full and must cleared by the system.	0
17	**MIC RX FIFO REQUEST (MICRFRQ):** When this bit is set to 1, it indicates that half of MIC RX FIFO is full and must be cleared by the system.	0
16	**Line2 RX FIFO REQUEST (L2RFRQ):** When this bit is set to 1, it indicates that half of Line 2 RX FIFO is full and must be cleared by the system.	0
15	**HSET RX FIFO REQUEST (HTRFRQ):** When this bit is set to 1, it indicates that half of HSET RX FIFO is full and must be cleared by the system.	0
14	**IO CTRL/Status RX FIFO REQUEST (IOCSRFRQ):** When this bit is set to 1, it indicates that half of IO CTRL/Status RX FIFO is full and must be cleared by the system.	0
13	**PCML RX FIFO OVERRUN (PLRFOV):** When this bit is set to 1, it indicates that PCML RX FIFO is overrun.	0
12	**PCMR RX FIFO OVERRUN (PRRFOV):** When this bit is set to 1, it indicates that PCMR RX FIFO is overrun.	0
11	**Line 1 RX FIFO OVERRUN (L1RFOV):** When this bit is se to 1, it indicates that Line 1 RX FIFO is overrun.	0
10	**MIC RX FIFO OVERRUN (MICRFOV):** When this bit is set to 1, it indicates that MIC RX FIFO is overrun.	0
9	**Line2 RX FIFO OVERRUN (L2RFOV):** When this bit is set to 1, it indicates that Line 2 RX FIFO is overrun.	0
8	**HSET RX FIFO OVERRUN (HTRFOV):** When this bit is set to 1, it indicates that HSET RX FIFO is overrun.	0
7	**IO CTRL/Status RX FIFO OVERRUN (IOCSRFOV):** When this bit is set to 1, it indicates that IO CTRL/Status RX FIFO is overrun.	0
6	**PCML RX FIFO UNDERRUN (PLRFUN):** When this bit is set to 1, it indicates that PCML RX FIFO is underrun.	0
5	**PCMR RX FIFO UNDERRUN (PRRFUN):** When this bit is set to 1, it indicates that PCMR RX FIFO is underrun	0
4	**Line1 RX FIFO UNDERRUN (L1RFUN):** When this bit is set to 1, it indicates that Line 1 RX FIFO is underrun	0
3	**MIC RX FIFO UNDERRUN (MICRFUN):** When this bit is set to 1, it indicates that MIC RX FIFO is underrun	0

Bit	Description	Default
2	**Line2 RX FIFO UNDERRUN (L2RFUN):** When this bit is set to 1, it indicates that Line 2 RX FIFO is underrun	0
1	**HSET RX FIFO UNDERRUN (HTRFUN):** When this bit is set to 1, it indicates that HSET RX FIFO is underrun	0
0	**IO CTRL/STA RX FIFO UNDERRUN (IOCSRFUN):** When this bit is set to 1, it indicates that IO CTRL/STA RX FIFO is underrun	0

14.2.22 AC97 Control Register (ACR)

ACR, a 32-bit Write only register, is used to control AC97 controller. Bits 30-23 are reserved. The other bits are initialized to 0 at reset. ACR is not initialized at STANDBY mode.

Bit	31	30	29	28	27	26	25	24
Bit Name	VS	-	-	-	-	-	-	-
Initial Value	0	-	-	-	-	-	-	-
R/W	W	-	-	-	-	-	-	-

Bit	23	22	21	20	19	18	17	16
Bit Name	-	RXDMA_EN	TXDMA_EN	FCAF	FCDF	FSTAF	FSTDF	FPLTF
Initial Value	-	0	0	0	0	0	0	0
R/W	-	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8
Bit Name	FPRTF	FL1TF	FPCTF	FPLSTF	FPRSTF	FPLETF	FL2TF	FHTF
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

Bit	7	6	5	4	3	2	1	0
Bit Name	FIOCTF	FPLRF	FPRRF	FL1RF	FMRF	FL2RF	FHRF	FIOSRF
Initial Value	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W

AC97 Control Register (ACR) [cont'd]

Bit	Description	Default
31	**Version Select (VS):** If this bit is set to 1, this module supports AC97 Ver2.0. Otherwise, this module supports AC97 Ver1.03	0
30 - 21	Reserved	-
22	**RX DMA Enable (RXDMA_EN):** If this bit is set to 1, RX DMA function is enabled. Otherwise, RX DMA function is disabled.	0
21	**TX DMA Enable (TXDMA_EN):** If this bit is set to 1, TX DMA function is enabled. Otherwise, TX DMA function is disabled.	0
20	**Flush CAR FIFO (FCAF):** When this bit is written by 1, command address FIFO is flushed.	0
19	**Flush CDR FIFO (FCDF):** When this bit is written by 1, command data FIFO is flushed.	0
18	**Flush STA FIFO (FSTAF):** When this bit is written by 1, status address FIFO is flushed.	0
17	**Flush STD FIFO (FSTDF):** When this bit is written by 1, status data FIFO is flushed.	0
16	**Flush PCML TX FIFO (FPLTF):** When this bit is written by 1, PCML TX FIFO is flushed.	0
15	**Flush PCMR TX FIFO (FPRTF):** When this bit is written by 1, PCMR TX FIFO is flushed.	0
14	**Flush Line1 TX FIFO (FL1TF):** When this bit is written by 1, Line1 TX FIFO is flushed.	0
13	**Flush PCMC TX FIFO (FPCTF):** When this bit is written by 1, PCMC TX FIFO is flushed.	0
12	**Flush PCML Surround TX FIFO (FPLSTF):** When this bit is written by 1, PCMLS TX FIFO is flushed.	0
11	**Flush PCMR Surround TX FIFO (FPRSTF):** When this bit is written by 1, PCMRS TX FIFO is flushed.	0
10	**Flush PCM LFE TX FIFO (FPLETF):** When this bit is written by 1, PCMLFE TX FIFO is flushed.	0
9	**Flush Line2 TX FIFO (FL2TF):** When this bit is written by 1, Line 2 TX FIFO is flushed.	0
8	**Flush HSET TX FIFO (FHTF):** When this bit is written by 1, HSET TX FIFO is flushed.	0
7	**Flush IO CTRL TX FIFO (FIOCTF):** When this bit is written by 1, IO CTRL TX FIFO is flushed.	0
6	**Flush PCML RX FIFO (FPLRF):** When this bit is written by 1, PCML RX FIFO is flushed.	0
5	**Flush PCMR RX FIFO (FPRRF):** When this bit is written by 1, PCMR RX FIFO is flushed.	0
4	**Flush Line1 RX FIFO (FL1RF):** When this bit is written by 1, Line 1 RX FIFO is flushed.	0
3	**Flush MIC RX FIFO (FMRF):** When this bit is written by 1, MIC RX FIFO is flushed.	0
2	**Flush Line2 RX FIFO (FL2RF):** When this bit is written by 1, Line 2 RX FIFO is flushed.	0
1	**Flush HSET RX FIFO (FHRF):** When this bit is written by 1, HSET RX FIFO is flushed.	0
0	**Flush IO STATUS RX FIFO (FIOSRF):** When this bit is written by 1, IO Status RX FIFO is flushed.	0

Note: Once bit 21 or 22 is set, PCML and PCMR FIFO will be accessed through PCML register with interleaving method.

14.2.23 AC97 TAG Register (ATAGR)

ATAGR, a 32-bit Read/Write register, is used to write TX tag or read RX tag. Bits 18-13 are reserved. The other bits are initialized to 0 at reset. ATAGR is not initialized in STANDBY mode.

Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**
Bit Name	CR	RXVS1	RXVS2	RXVS3	RXVS4	RXVS5	RXVS6	RXVS7
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Bit Name	RXVS8	RXVS9	RXVS10	RXVS11	RXVS12	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	-	-	-

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Bit Name	-	-	-	TXVS3	TXVS4	TXVS5	TXVS6	TXVS7
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Name	TXVS8	TXVS9	TXVS10	TXVS11	TXVS12	ZERO	ZERO	ZERO
Initial Value	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

Bit	**Description**	**Default**
31	**CODEC Ready (CR):** When this bit is set to 1, it indicates that the connected CODEC is ready.	0
30 - 19	**RX Valid Slot 1-12 (RXVS1-12):** These bits indicate which slots read from CODEC are valid.	0
18 - 13	Reserved	0
12 - 3	**TX Valid Slot 3-12 (TXVS3-12):** These bits are valid only in AC97 Version 1.03. System can write these bits to indicate which slots of the connected CODEC are valid.	0
2 - 0	**ZERO:** Don't write 1 to these bits.	0

14.2.24 Slot Request Active Register (SRAR)

SRAR, a 16-bit Read/Write register, is used to set the active level of the solt request of slot3-12. Bits 15-13 and 2-0 are reserved. The other bits are initialized to 0 at reset. SRAR is not initialized in STANDBY mode.

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	SL12RA	SL11RA	SL10RA	SL9RA	SL8RA
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
Bit Name	SL7RA	SL6RA	SL5RA	SL4RA	SL3RA	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	-	-	-

Bit	Description	Default
16 - 10	Reserved	0
12	**Slot Request 12 Active (SL12RA):** When this bit is set to 1, slot 12 request is high active. Otherwise, slot 12 request is low active.	0
11	**Slot Request 11 Active (SL11RA):** When this bit is set to 1, slot 11 request is high active. Otherwise, slot 11 request is low active.	0
10	**Slot Request 10 Active (SL10RA):** When this bit is set to 1, slot 10 request is high active. Otherwise, slot 10 request is low active.	0
9	**Slot Request 9 Active (SL9RA):** When this bit is set to 1, slot 9 request is high active. Otherwise, slot 9 request is low active.	0
8	**Slot Request 8 Active (SL8RA):** When this bit is set to 1, slot 8 request is high active. Otherwise, slot 8 request is low active.	0
7	**Slot Request 7 Active (SL7RA):** When this bit is set to 1, slot 7 request is high active. Otherwise, slot 7 request is low active.	0
6	**Slot Request 6 Active (SL6RA):** When this bit is set to 1, slot 6 request is high active. Otherwise, slot 6 request is low active.	0

Slot Request Active Register (SRAR) [cont'd]

Bit	Description	Default
5	**Slot Request 5 Active (SL5RA):** When this bit is set to 1, slot 5 request is high active. Otherwise, slot 5 request is low active.	0
4	**Slot Request 4 Active (SL4RA):** When this bit is set to 1, slot 4 request is high active. Otherwise, slot 4 request is low active.	0
3	**Slot Request 3 Active (SL3RA):** When this bit is set to 1, slot 3 request is high active. Otherwise, slot 3 request is low active.	0
2 - 0	Reserved	-

14.3 Function Description

This Serial CODEC Interface incorporates Internal Bus Interface, Clock generator, CS42xx Transmit/Receive data controller, AC97 Transmit/Receive data controller, and miscellaneous function block. Each function block is described in the following sections.

14.3.1 Internal Bus Interface

This function block provides the registers that can read/write by system via an internal bus. These registers include control registers, status registers and TX/RX data registers. These registers are described in Section 14-2.

14.3.2 Clock Generator

This function block provides a variable frequency master clock generator and a gating CKIO circuit. The master clock generator supports a master clock for CS4218 CODEC to allow the CODEC have a sample rate from 8KHz to 48KHz. The frequency can be controlled by FSR register. And the serial clock, which has half frequency of master clock, is generated in this generator, too. The gating CKIO circuit gates CKIO in STANDBY mode to reduce power consumption.

14.3.3 CS4218 or CS4271 TX Controller

This module, as shown in Fig14-2, contains a TX FIFO, a 32-bit buffer (BUF), a parallel-to-serial shift register (PSR) and an inferred control circuit. This TX FIFO has two blocks and each block has 4 32-bit entries. When TX transaction begins, the data stored in TX FIFO will be written to BUF whenever BUF is empty. The data stored in BUF is written to PSR at specific timing points of each frame and then the data is serially transmitted to the connected CODEC via serial data output port. In PIO mode, this module signals an interrupt request to the system under one of the following conditions: TX FIFO transmit done, underrun and overrun. TX FIFO transmit done means that any of the two blocks is empty. The system can read SR to realize the source of the interrupt request and perform related interrupt service routine. In DMA mode, this module signals a DMA request (DREQ#) whenever TX FIFO is nearly empty, and releases DREQ# whenever TX FIFO is close to full. DMA mode is more preferred because it will not the increase the CPU workload.

14.3.4 CS4218 or CS4271 RX Controller

This module, as shown in Fig14-3, contains a RX FIFO, a 32-bit buffer (BUF), a serial-to-parallel shift register(SPR) and an inferred control circuit. This RX FIFO has two blocks and each block has 4 32-bit entries. When RX transaction works, serial data from serial data input port is serially shifted into SPR one bit per serial clock cycle and the data stored in SPR is written to BUF at specific timing points of each frame. The data stored in BUF is written to RX FIFO whenever BUF is full. In PIO mode, this module will signal an interrupt request to the system under one of the following conditions: RX FIFO transmit done, underrun and overrun. RX FIFO transmit done means that any of the two blocks is full. In DMA mode, this module signals DREQ# whenever RX FIFO is close to full and releases DREQ# whenever RX FIFO is nearly empty.

14.3.5 AC97 TX Controller

This module, as shown in Fig14-4 on page 169, contains several TX FIFOs, a TX TAG register, a 20-bit buffer, a parallel-to-serial shift register and an inferred control circuit. Each TX FIFO has 4 20-bit entries and is divided into two blocks. The operation mechanism is analogous to CSxx TX controller. The output of TX TAG register and that of the SLOTREQ register are multiplexed and selected, depending on which version is selected. If version 2.0 is selected, AC97 RX controller gets slot request from the receive frame and sends it to AC97 TX controller as the reference of the valid slot flags of the next transmit frame.

If version 1.03 is selected, AC97 TX controller gets the data in ATAGR register as valid flags of slots 3-12 of each transmit frame. The valid frame bit of a transmit frame is 0 if all of the valid flags of slots 1-12 are 0. Otherwise, it is 1. The valid flag of slot1 or 2 of a transmit frame is 1 when CAR or CDR TX FIFO is not empty.

DMA transfer only supports PCML (slot3) and PCMR (slot4) at the same time. Sound data must be prepared as stereo sound data. Only PCML (slot3) register is used for stereo data transfer with automatic register switching.

14.3.6 AC97 RX Controller

This module, as shown in Fig14-5, contains several RX FIFOs, a SLOTREQ register, a 20-bit buffer, a serial-to-parallel shift register and an inferred control circuit. Each RX FIFO has 4 20-bit entries and is divided into two blocks. The operation mechanism is analogous to CSxx RX controller.

DMA transfer only supports PCML (slot3) and PCMR (slot4) at the same time. Sound data will be saved as stereo sound data. Only PCML (slot3) register is used for stereo data transfer with automatic register switching.

14.3.7 Miscellaneous Function Block

This module controls many miscellaneous signals, including output pad enable, input pad enable, serial SYNC, power down signal and reset signal.

14.3.8 Data Structure of Memory in DMA Mode

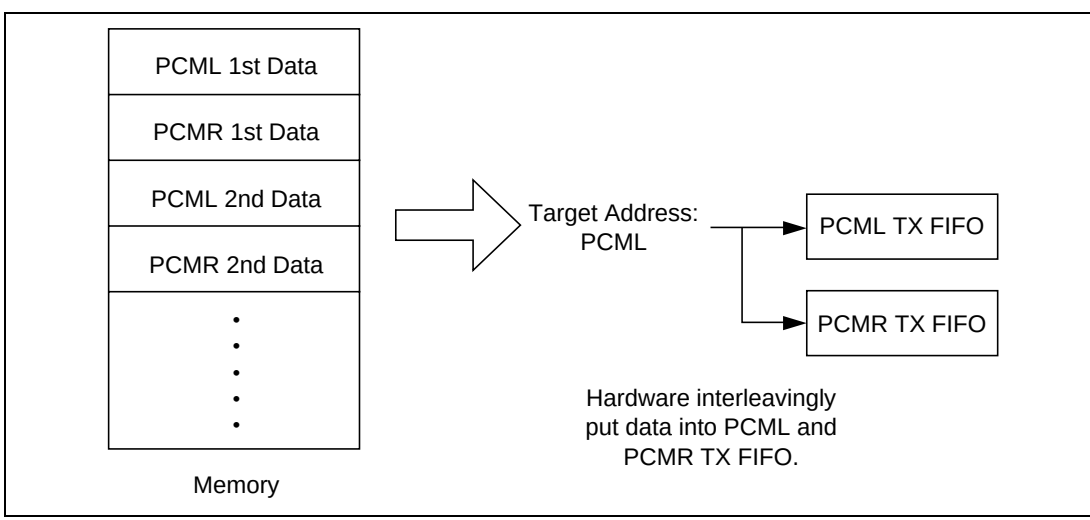


Figure 14.2 Data Transfer Scheme in DMA TX Mode

14.4 Program Flow

Figure14-7 on page 171 illustrates the program flow of CS4218 or CS4271 in PIO mode. Figure14-8 on page 172 illustrates the program flow of AC97 in DMA mode. The flows under the other conditions including that of CS42xx in DMA mode and that of AC97 in PIO mode are analogous to these flows.

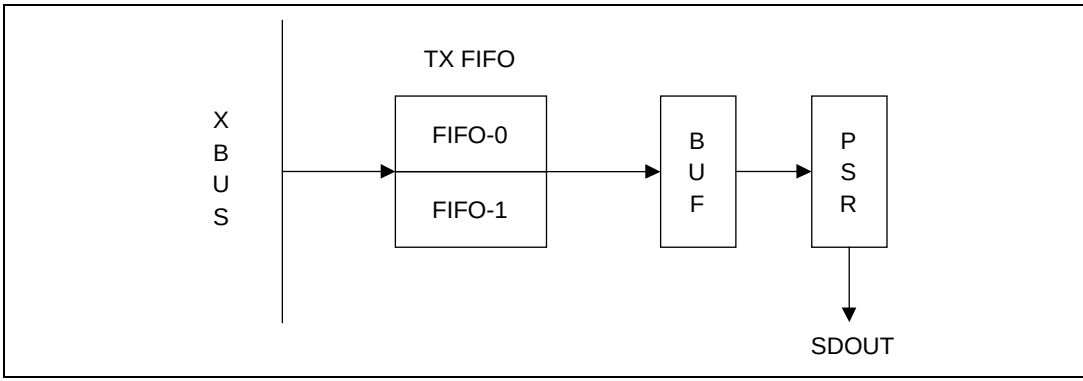


Figure 14.3 CS4218 or CS4271 TX Controller

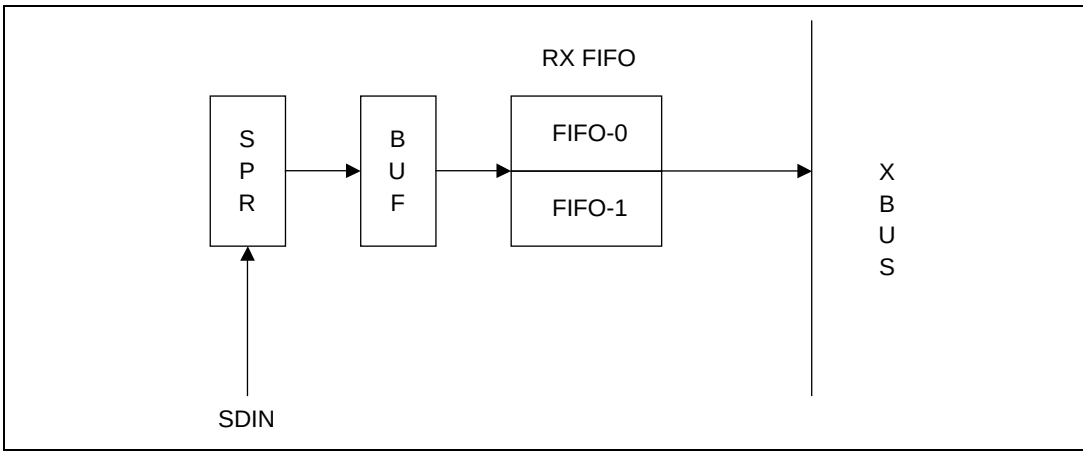


Figure 14.4 CS4218 or CS4271 RX Controller

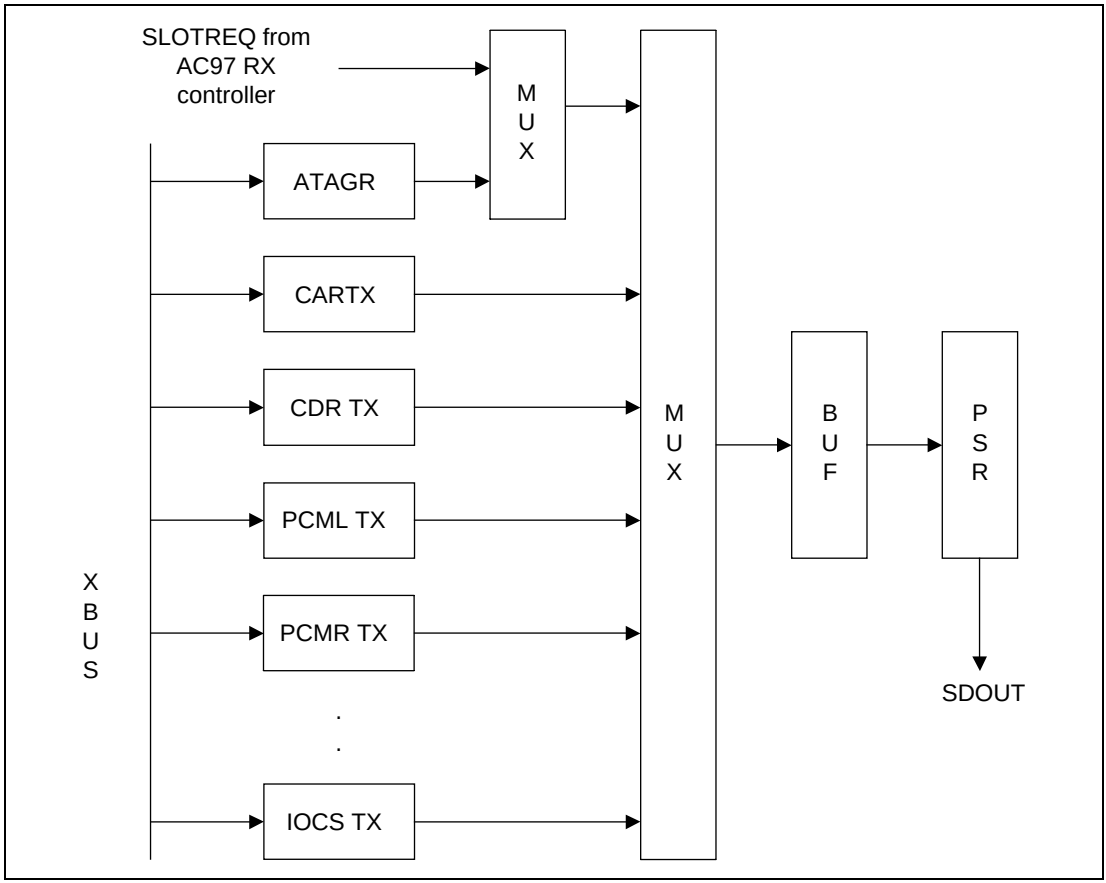


Figure 14.5 AC97 TX Controller

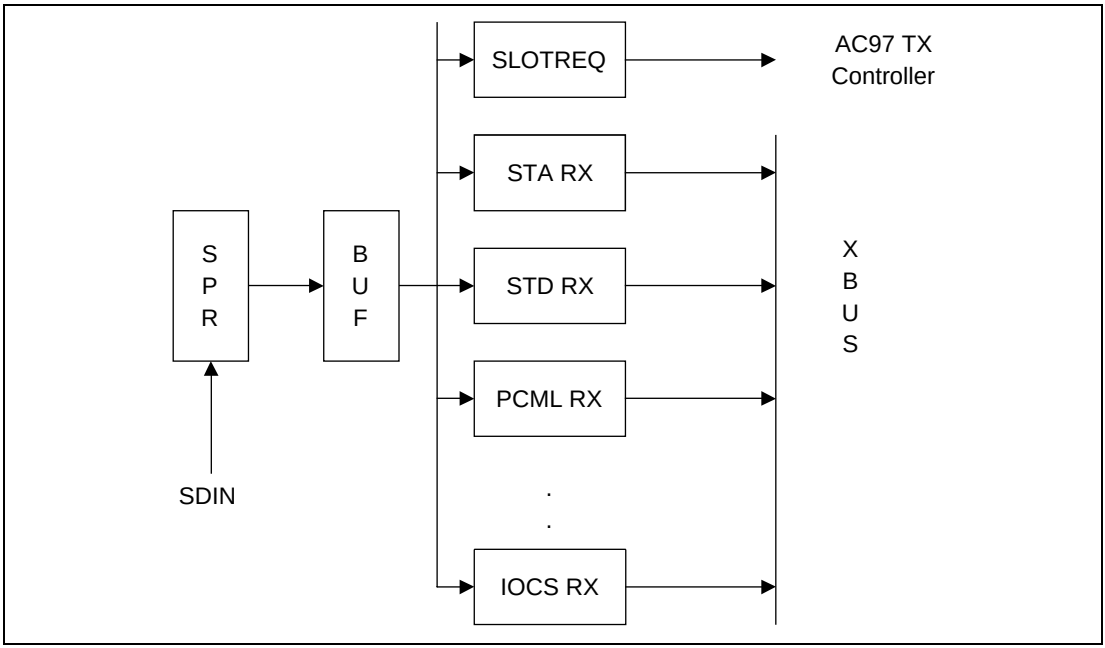


Figure 14.6 AC97 RX Controller

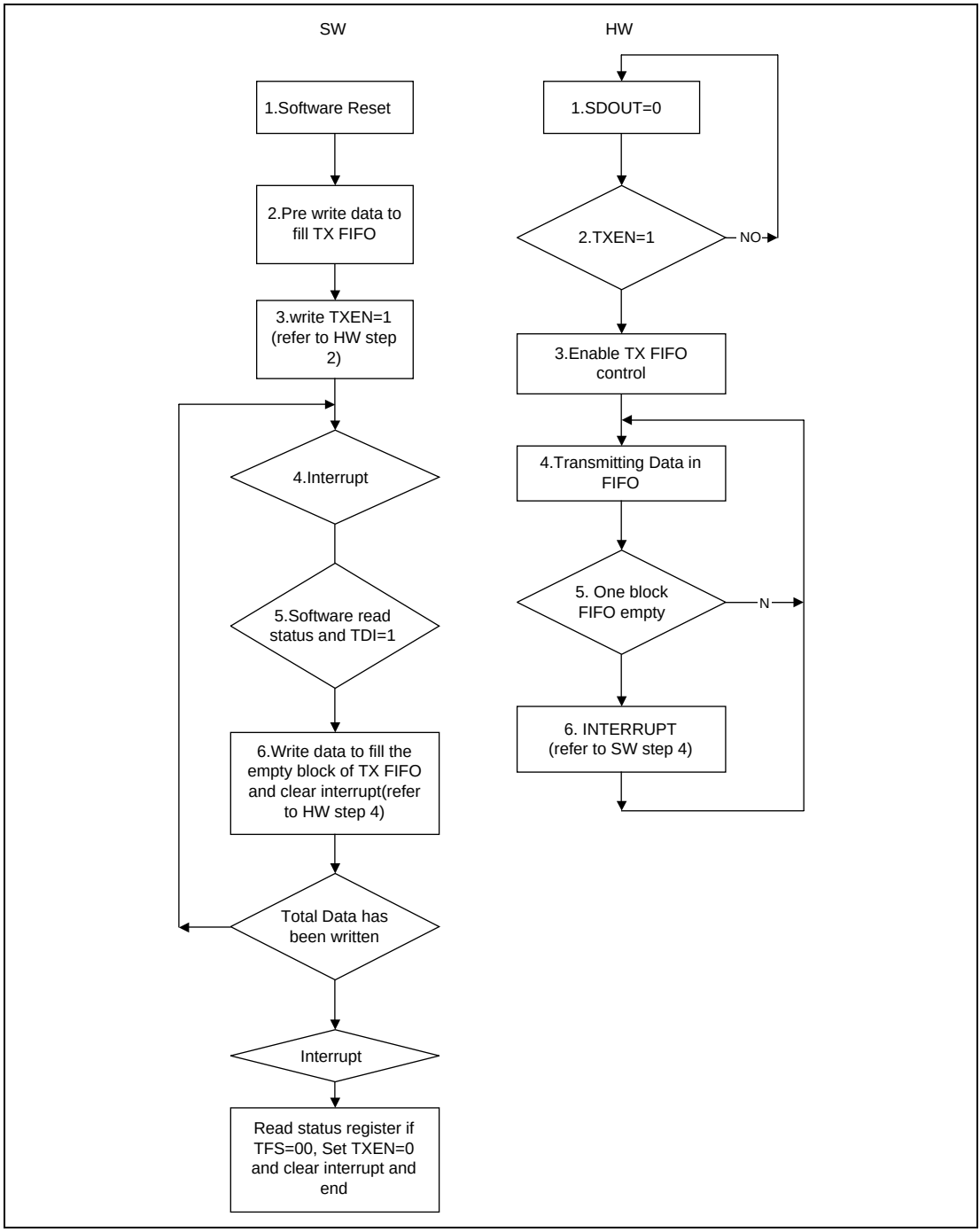


Figure 14.7 TX Flow in PIO Mode for CS4218 or CS4271

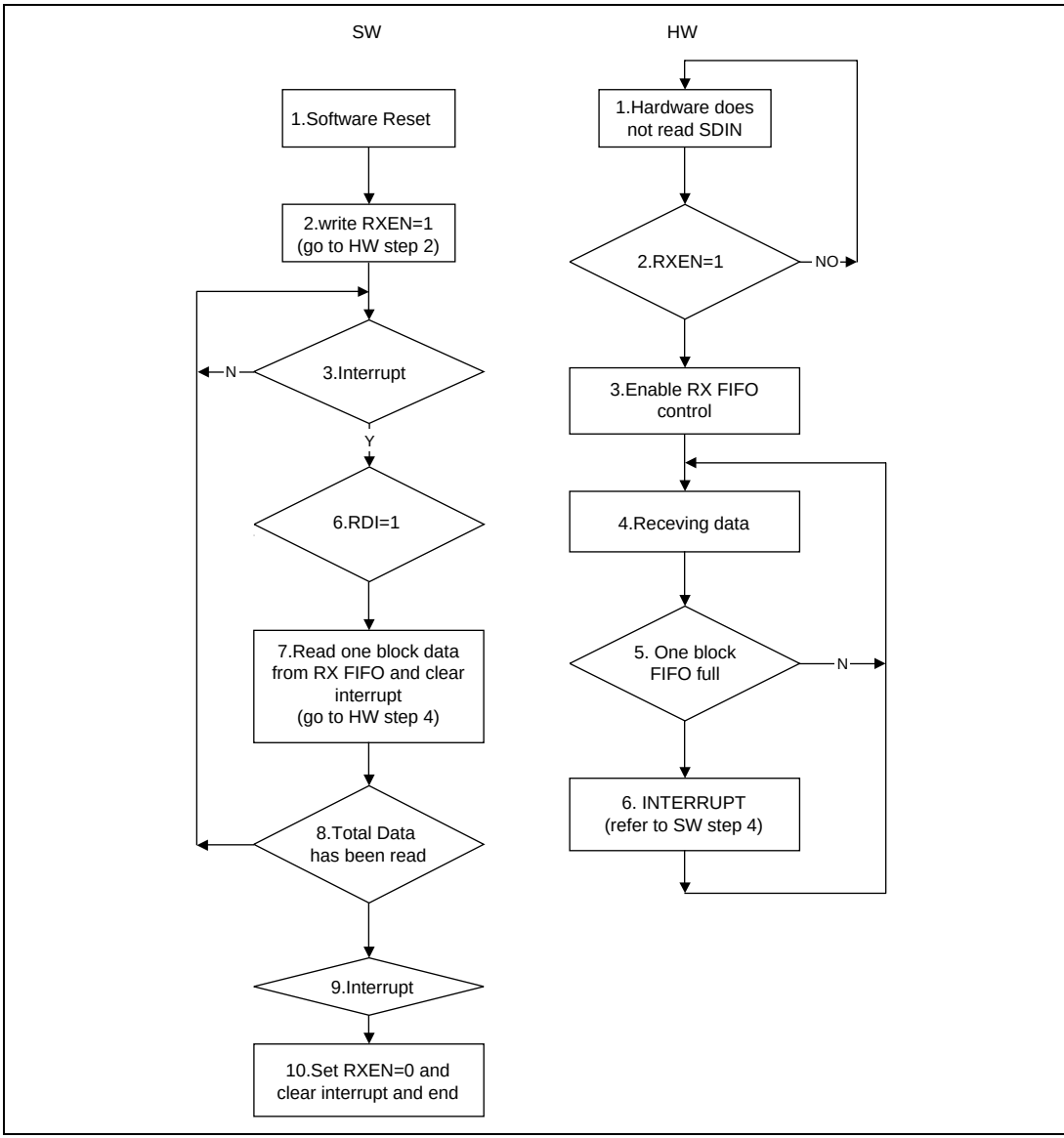


Figure 14.8 RX Flow in PIO Mode for CS4218 or CS4271

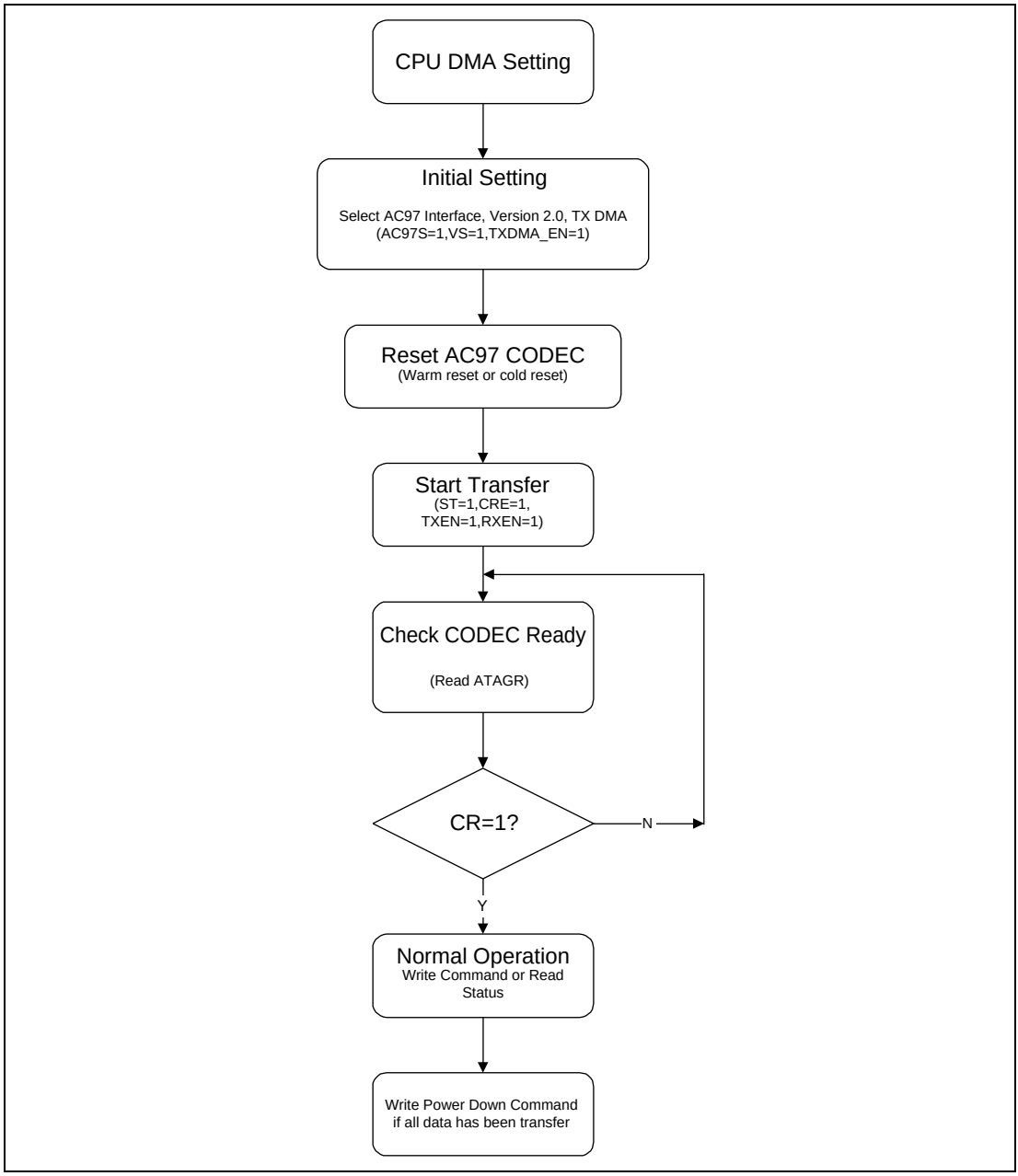


Figure 14.9 AC97 DMA Program Flow

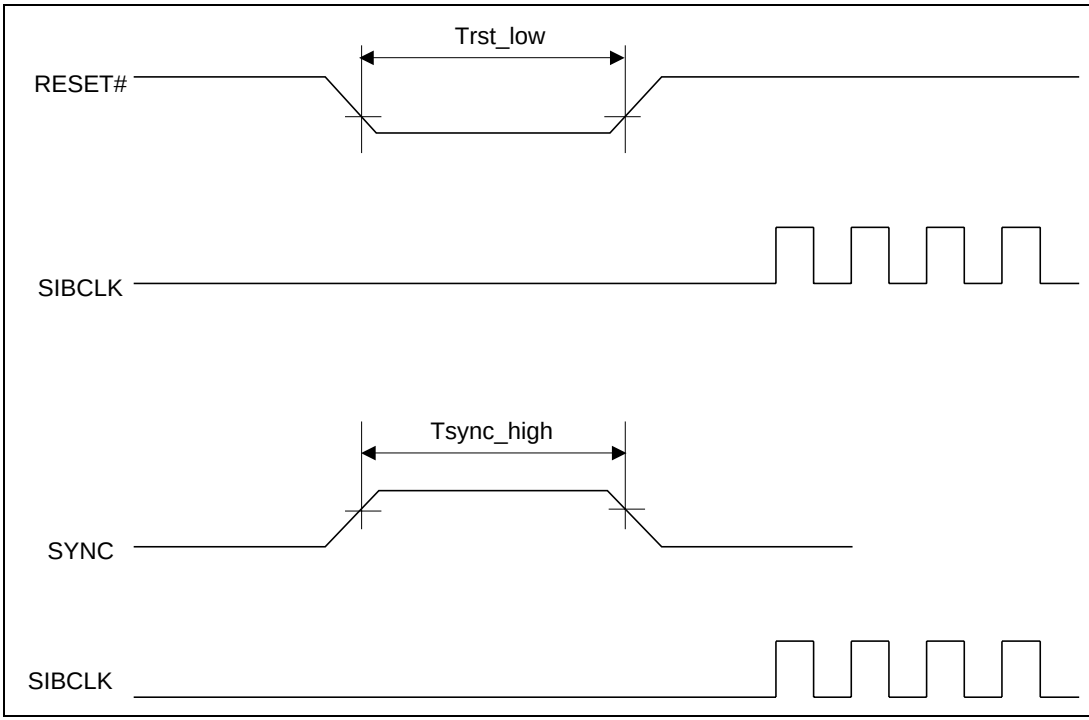


Figure 14.10 Warm/Cold Reset Timing

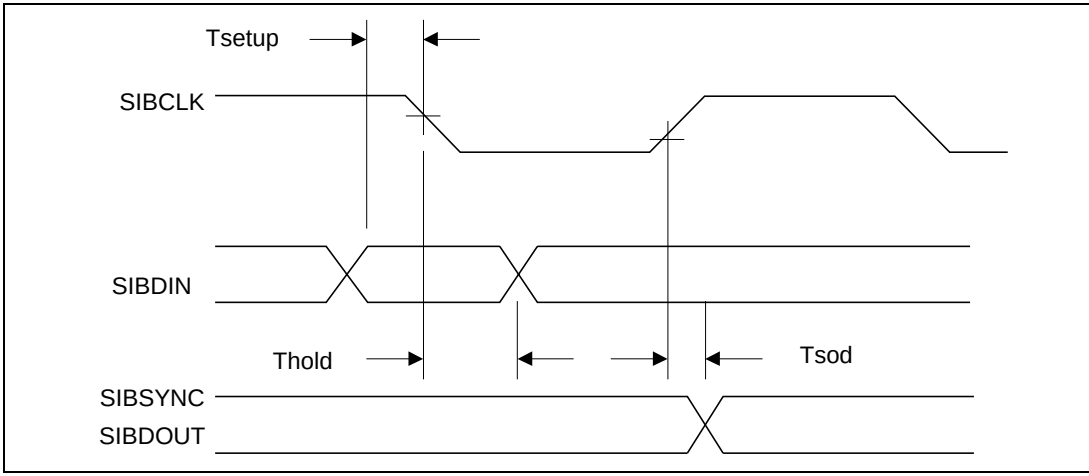


Figure 14.11 Serial Data Setup, Hold and Output Delay Timing

Table 14.3 AC97 Timing

Item	Symbol	Min	Typ	Max	Unit
RESET# Active Low Pulse Width	Trst_low		1.3		µs
SYNC Active High Pulse Width	Tsync_high		1.3		µs
Setup to Falling Edge of SIBCLK	Tsetup	10			ns
Hold to Falling Edge of SIBCLK	Thold	10			ns
Output Delay to Rising Edge of SIBCLK	Tsod			15	ns

Section 15 AFE Interface

15.1 Overview

An AFE interface is a circuit to interface with Modem AFE (SGS-Thomson STLC7546/7550); and is capable of performing a serial data transfer function. A divider is also incorporated to transmit a master clock to Modem AFE.

15.1.1 Features

- Supports full-duplex serial-data transfer can be achieved with the presence of buffers. The buffers can be independently used for data transmit and receive.
- Data can be transmitted to and received from the CPU in each 1-word unit to 48-word unit. An AFE interface contains a 1-word data register and two 48-word data buffers, which can be selected for data READ or WRITE by users based on the applications.
- Users can perform READ or WRITE operations to only one buffer. Although two buffers are generally involved for each data transmit or receive, users can use just one buffer, which is not currently used for the external transfer.
- All the transmit/receive registers and accessible buffers are memory-mapped.
- Data transmit/receive is performed via an interrupt signal. An AFE interface generates a transmit data empty interrupt and a receive data full interrupt to request data transmit or receive.
- Data transfer is performed in the MSB-First way.
- Supports three division ratio options: 1/8, 1/7, or 1/6.

15.1.2 Block Diagram

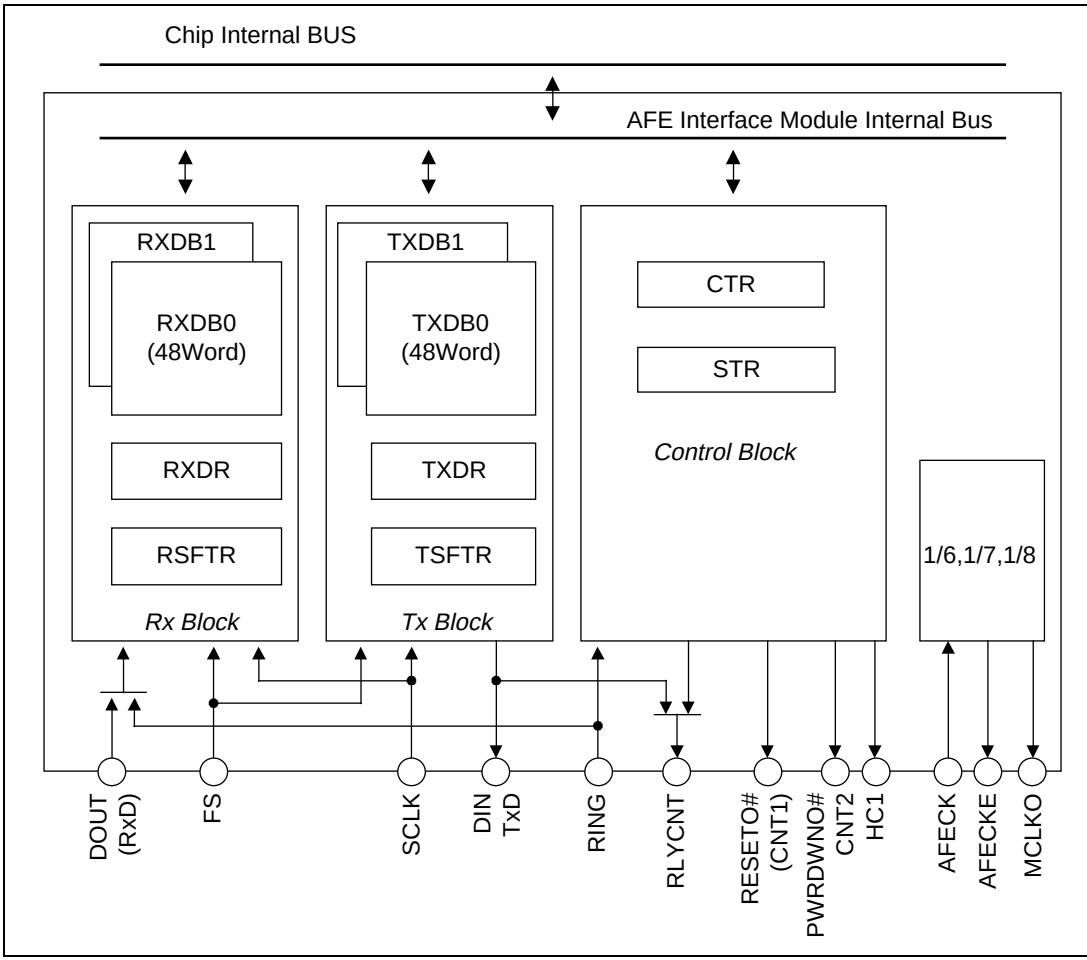


Figure 15.1 AFE Interface Block Diagram

Table 15.1 Pin Function of AFE Interface Module

Name	I/O	Pin Function
DOUT(RxD)	I	Serial Receive Data Input Pin, from AFE module
FS	I	Frame Sync Signal Input Pin
SCLK	I	Shift Clock Input Pin
DIN(TxD)	O	Serial Transmit Data Output Pin, to AFE module
RESETO# (CNT1)	O	External Chip Control Signal 1 Output Pin (RESETO#)
PWRDWNO#(CNT2)	O	External Chip Control Signal 2 Output Pin (PWRDWNO#)
RLYCNT	O	RLYCNT Control and Dial Pulse Output Pin
HC1	O	Hardware Control Signal 1 for STLC7546/7550
RING	I	Ringing Signal Input Pin
AFECK	I	Crystal Oscillator clock input Pin
AFECKE	I/O	Crystal Oscillator Output Pin
MCKO	O	Master Clock for Modem Pin

15.2 Register Description

An AFE interface contains registers and buffers described in the table below.

Table 15.2 Registers of AFE Interface

Register or Buffer	Function	Access Size	Address
CTR	AFE Interface Control Register	16 bits	H'10003200
STR	AFE Interface Status Register	16 bits	H'10003202
TXDR	Transmit Data Register	16 bits	H'10003206
RXDR	Receive Data Register	16 bits	H'10003204
TXDB0,1	Transmit Data Buffers 0,1	16 bits	H'10003100 - 5F
TSFTR	Transmit Shift Register	Can not be accessed	*
RXDB0,1	Receive Data Buffers Register 0,1	16 bits	H'10003000 - 5F
RSFTR	Receive Shift register	Can not be accessed	*

15.2.1 Control Register (CTR)

CTR, a 16-bit READ/WRITE register, is used to control an AFE interface. All the bits on this register are initialized to 0 at RESET. CTR is not initialized in the STANDBY mode.

Bit	15	14	13	12	11	10	9	8
Bit Name	HC	Div2	Div1	RLYCNT	CNT2	CNT1	TSW	RSW
Initial Value	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

Bit	7	6	5	4	3	2	1	0
Bit Name	RDETM	TEIE	REIE	TXIE	RXIE	BUFD	TE	RE
Initial Value	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

Bit	Description	Default
15	When this bit is set to 1, 1 is output from the HC1 pin when the next transmit ends, and then the value of TXDR is output when the second FS is received. After that, this bit is cleared to 0.	0
14, 13	Variable M setting for division ratio 00: Division ratio is 1/8 01: Division ratio is 1/7 10: Division ratio is 1/6 11: Reserved	0
12	RLYCNT pin output setting. Output from the RLYCNT pin.	0
11	External control signal 2 (CNT2) setting. Output from the PWRDWNO# pin.	0
10	External control signal 1 (CNT1) setting. Output from the RESETO# Pin.	0
9	RLYCNT pin output switch 1: transmitted data is output from the RLYCNT pin. 0: the value is output, and is set via bit 12 of the RLYCNT pin.	0
8	Receive data input pin switch 1: RING pin input indicates the received data. 0: DOUT (RxD) pin input indicates the received data.	0
7	RDET Interrupt MASK 1: disable 0: enable	0
6	Transmit Error Interrupt (TERI) enable 1: enable 0: disable	0

Control Register (CTR) [cont'd]

Bit	Description	Default
5	Receive Error Interrupt (RERI) enable 1: enable 0: disable	0
4	Transmit Data Empty Interrupt (TDEI) enable 1: enable 0: disable	0
3	Receive Data Full Interrupt (RDFI) enable 1: enable 0: disable	0
2	Buffer Disable 1: data is transferred with register (TXDB and RXDB). 0: data is transferred with buffers (TXDB and RXDB).	0
1	Transmit Enable 1: enable 0: disable	0
0	Receive enable. 1: enable 0: disable	0

15.2.2 Status Register (STR)

STR, a 6-bit READ only register (0s can only be written to lower four bits for clearing after 1s are read), indicates the status of an AFE interface. STR is not initialized in the STANDBY mode. STR must be read in word. The valid values cannot be guaranteed after a byte read is performed.

Bit	15	14	13	12	11	10	9	8
Bit Name	TAB	RAB	reserved	reserved	reserved	reserved	reserved	reserved
Initial Value	0	0	0	0	0	0	0	0
R/W	R/(W)	R/(W)	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	reserved	reserved	TERR	RERR	TDE	RDF
Initial Value	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R/(W)	R/(W)	R/(W)	R/(W)

Status Register [cont'd]

Bit	Description	Default
15	Indicates the transmit buffer is accessible 1: READ/WRITE can be performed to TXDB1. 0: READ/WRITE can be performed to TXDB0. This bit will be set when the following two conditions are met. 1) When only transmit data buffer 1 is empty 2) When the TE bit is set to 1 The bit will be cleared when either one of the following conditions is met. 1) When transmit data buffer 0 becomes empty 2) When the TE bit in CTR is cleared to 0 3) At RESET This bit can be written to when the TE bit is 0.	0
14	Indicates the receive-buffer is accessible 1: READ/WRITE can be performed to RXDB1. 0: READ/WRITE can be performed to RXDB0. This bit will be set when the receive data buffer 1 is full and receive data buffer 0 is not full. This bit will be cleared when either one of the following conditions is met. 1) When receive data buffer 0 becomes full 2) At RESET This bit can be written to when the RE bit is 0.	0
13 - 4	Reserved	0
3	Indicates a transmit error 1: indicates that a transmission error has occurred. 0: indicates that a transmission error does not occur. This bit will be set when the next FS is received while both transmit data buffers are empty. This bit will be cleared when either one of the following conditions is met. 1. At RESET 2. When STR is read and 0 is written to this bit, after the bit is set to 1 3. When the TE bit in CTR is cleared to 0	0

Status Register [cont'd]

Bit	Description	Default
2	Receive error indication 1: indicates that a receive error has occurred. 0: indicates that a receive error does not occur. This bit will be set when the next data receive is completed while both receive data buffers are full. This bit will be cleared when either one of the following conditions is met. 1. At RESET 2. When the RE bit in CTR is cleared to 0 3. When STR is read and 0 is written to this bit, after the bit is set to 1	0
1	Indicates that no transmitted data are in either one of the two transmit data buffers (TXDR). 1: indicates that at least either one of the two transmit data buffers (TXDR) does not have the transmitted data. 0: indicates that both transmit data buffers contain the transmitted data. This bit will be set when the following conditions are met. 1. At RESET 2. When the data in either one of the two transmit data buffers have been completely transmitted. 3. When the TE bit in CTR is cleared to 0. This bit will be cleared when STR is read and 0 is written to this bit, after the bit is set to 1. Note: When both of the transmit data buffers are empty (as in the cases when the TE bit is cleared to 0), a clear operation must be performed twice since the empty status of only one buffer has been canceled, as indicated by TAB bit.	0
0	Indicates that one or two receive data buffers (RXDR) are full of the received data. 1: indicates that at least one receive data buffer is full of the received data. 0: indicates that two receive data buffers are not completely full. This bit will be set when data are stored in RXDR, or in all the words in RXDB0 or RXDB1. This bit will be cleared when either one of the following conditions is met. 1) At RESET 2) When STR is READ and 0 is written to this bit, after the bit is set to 1. Note: A clear operation must be performed twice when both of the receive data buffers are full since the full status of only one receive data buffer has been canceled as indicated by RAB.	0

15.2.3 Transmit Data Register (TXDR)

TXDR, a 16-bit READ/WRITE register, is used to transmit the stored data. All the bits in this register are initialized to 0 at RESET. TXDR is not initialized in the STANDBY mode.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	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

TXDR functions as a transmit data register when the BUFD bit (bit 15) in CTR is 1, and an AFE control data transmit register when the BUFD bit is 0. (see Figure 15.1 on page 175)

15.2.4 Receive Data Register (RXDR)

RXDR, a 16-bit READ only register, is used to receive the stored data. All the bits in this register are initialized to 0 at RESET. RXDR is not initialized in the STANDBY mode.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	0	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

RXDR functions as a receive data register when the BUFD bit (bit 15) in CTR is 1, and an AFC control data receive register when the BUFD bit is 0. AFE control data are stored and transmitted to TXDR at the same time.

15.2.5 Transmit Data Buffers (TXDB0,1)

TXDB0 and TXDB1 act as transmit data storage buffers, and are able to store 48-word data. The hardware configuration determines the buffer, from which the data are transmitted. A hardware configuration determines which buffer data will be transmitted. Users can access only one buffer, which is not currently used for data transmit. TDB0 and TDB1 are not initialized in the STANDBY mode.

15.2.6 Transmit Shift Register (TSFTR)

TSFTR, a 16-bit register, is used to convert a parallel transmit data into a serial one. Note that READ/WRITE operations cannot be performed to this register. The initial value of this register is undefined at RESET or in the STANDBY mode.

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

15.2.7 Receive Data Buffers (RXDB0,1)

RXDB0 and RXDB1 are receive data storage buffers, and are able to store 48-word data. The hardware configuration determines the buffer, where the data are received. Users can access only one buffer that is not used to receive data. RDB0 and RDB1 are not initialized in the STANDBY mode.

15.2.8 Receive Shift Register (RSFTR)

RSFTR, a 16-bit register, is used to convert a serial receive data into a parallel one. Note that READ/WRITE operations cannot be performed to this register. The initial value of this register is undefined at RESET or in the STANDBY mode.

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

15.3 Data Transfer

15.3.1 Data Transmit

Data are transmitted by setting the TE bit in CTR to 1. The transmitted data are stored in TXDR (transmit data register), or TXSB0/TXDB1 (transmit data buffers 0 or 1), which can be selected by the BUFD bit in CTR.

Data Transmit with Buffer

When data are transmitted via TXDB0/TXDB1, buffer 0 or 1, which is used for data transmission, is selected by hardware. This will enable the CPU (SH3) to access only one buffer, which is not used for data transmission.

When the data in the buffer 0 are completely transmitted, a transmit data empty interrupt (TDEI) is generated to the CPU interface, and an interrupt request is then output as a chip interrupt from the CPU interface to the CPU (SH3). A transmit data error interrupt (TERI) is then output if FS (frame synchronous signal) are received when the data in the buffer 1 are completely transmitted and the next data to be transmitted is not written. When a buffer (TXDB0/TXDB1) is used for data transmission, a transmit data register (TXDR) is used as a transmit register to control the AFE data.

Data Transmit with Register

When data are transmitted via TXDR, data stored in TXDR are used as the transmitted data. When data transmission starts in TXDR, TDEI is output. TERI is output when the next FX is received before the transmitted data are written to TXDR.

Before 1st Data Transmit in Buffer Use Mode

In an AFE-interface module, it is defined that both transmit data buffers are empty when the TE bit is 0. Consequently, data must be written to both of the transmit data buffers, and the TDE bit in STR must be cleared before initiating the data transmit (before the TE bit is set to 1). Since the TAB bit is 0 when they are empty, the TAB bit should be first set to 1 by the first clear operation, and then cleared to 0 by the second clear operation.

The following details how the whole procedure works:

1) 48 words of data are written to one transmit data buffer.
2) STR is read and 0 is written to bit 1.
3) Further 48 words of data are written to the other transmit data buffer.
4) STR is read and 0 is written to bit 1.

If the TDE bit is cleared without writing any data to transmit data buffers (only steps 2 and 4 in the above procedure are performed), the output value cannot be guaranteed for the first 96 words of data.

15.3.2 Data Receive

Data are received by setting the RE bit in CTR to 1.Received data can be stored in RXDR (receive data register) or RXDB0/RXDB1 (receive data buffer 0 or 1), which is determined by the BUFD bit in CTR.

Data Receive with Buffer

When data are to be received via RXDB0/RXDB1, the hardware will determine it will receive and select the data in buffer 0 or buffer 1. At this moment, the CPU (SH3) can only access one buffer, which is not used to receive data.

When the received data are completely stored in one buffer, a receive data full interrupt (RDFI) is output. A receive data error interrupt (RERI) is output when a received data is pending while the received data have been fully written in the two buffers.

When a buffer is used to receive data, RXDR is used as an AFE control data receive register, in which data are stored and transmitted synchronously via the third FS (frame synchronous single) after the HC bit is set to 1.

Data Receive with Register

When data are received in RXDR, the received data are stored in RXDR.

When data are stored in RXDR, RDFI is output. A receive data error interrupt is output when the next data are received in the receive data full status.

When Both Receive Data Buffers Are Full

If both receive data buffers become full, the RDF bit must be cleared (clearing to 0 after reading 1) twice in the same way as described in section 15.1.3.1.3 on the last page. (However, since both receive data buffers are initially empty, it can be considered erroneous occurrences if they are simultaneously full.)

15.4 Divider

In the divider, division ratio can be selected among 1/8, 1/7, and 1/6 via the Div1 and the Div2 bits in CTR (see figure 15-2 below).

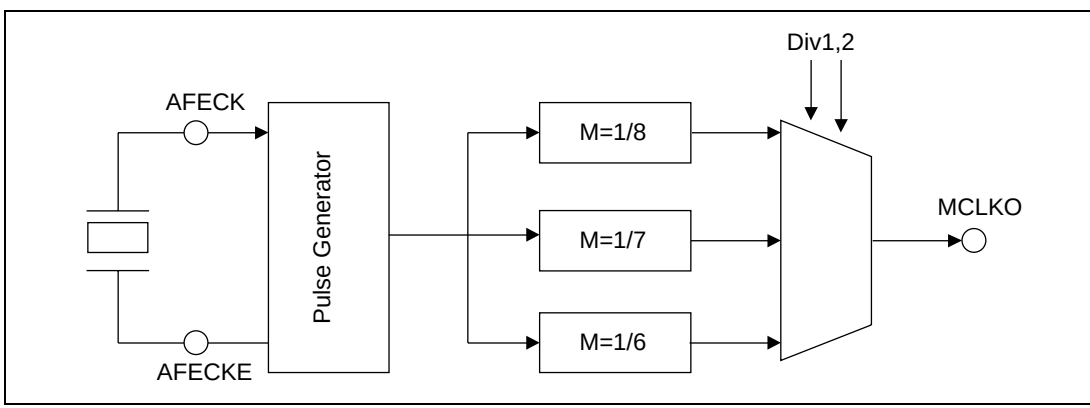


Figure 15.2 Divider Configuration

15.5 External Chip Control Signal

An AFE interface is capable of transmitting the three external chip control signals of RESETO#, PWRDWNO# and HC1. The output for each of the signals is individually controlled by the corresponding bits in CRT. For the two signals of CNT1 (RESETO#) and CNT2 (PWRDWNO#), the values set in CTR are output without delay. HC1, on the other hand, is output when the data transfer finishes (point A of figure 15-3) immediately after the HC1 bit is set to 1. After a data is successfully transferred into a control data (point B of figure 15-3), the output value of HC1 output is 0 and the bit in CTR is cleared to 0 (see figure 15-3).

Note that the HC1 bit can be used for the HC1 pin in STL7546 of SGS Thomson.

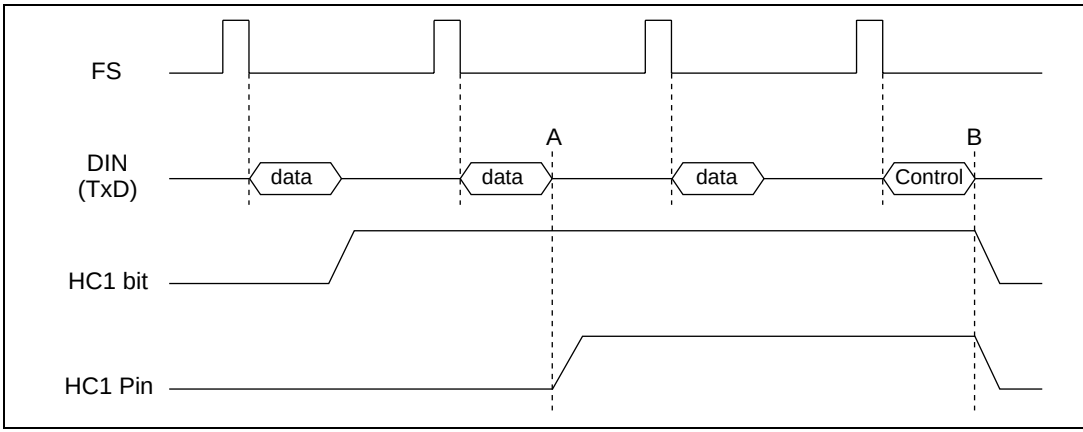


Figure 15.3 HC1 Pin and Control Data Outputs

15.6 Interrupt

An AFE interface is capable of generating five interrupts. Among these interrupts, four interrupts of TDEI, RDFI, TERI, and RERI are issued by setting the enable bits in CTR to 1. On the other hand, an interrupt of RDWT is masked by setting the RDETM bit to 1.

Interrupt Source	Issuance Condition	How to clear the interrupt
TDEI	Transmit register or buffer includes no transmit data.	Clear the TDE bit in STR
RDFI	Receive register or buffer is full of received data.	Clear the RDF bit in STR
TERI	Next data transmit starts while no transmit data is included.	Clear the TERR bit in STR
RERI	Receive register or buffer receives the next data in spite of being in the full status.	Clear the RERR bit in STR
RDET	When a low level is input to the RING pin (this is a level interrupt).	Write 1 to the RSW bit in CTR

Figures 15-4 and 15-5 shown below display the output timings of TDEI and RDFI.

For timings of TERI and RERI, TERI is output when the last word data transmission starts while RERI is output when the last word data transmission is completed. When the BUFD bit is set to 1, transmission is performed in 1 word, but output timings do not differ from those of TDEI and RDFI.

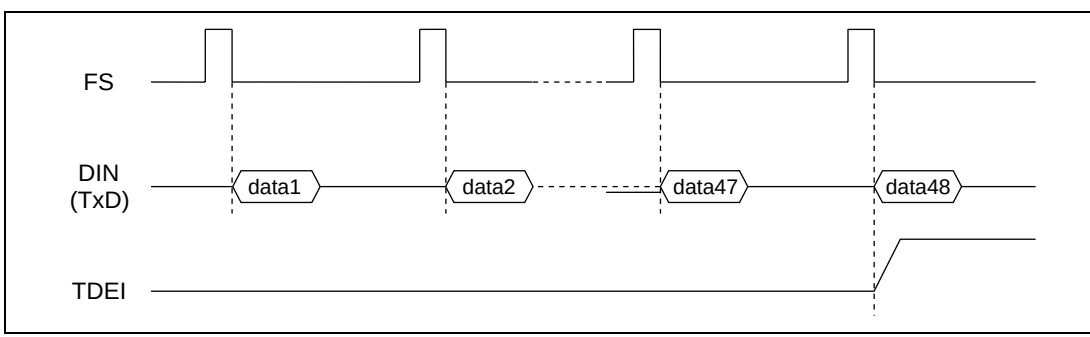


Figure 15.4 TDEI Output Timing

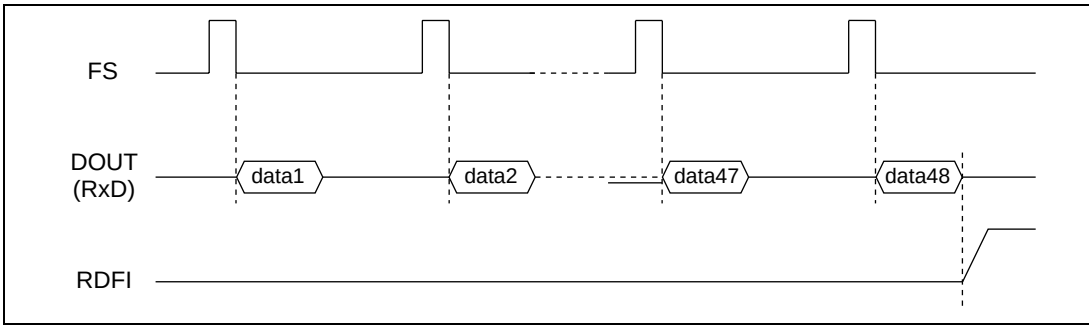


Figure 15.5 RDFI Output Timing

15.7 How to Use the Special Pin (RLYCNT, RING)

An AFE interface contains two special pins: RLYCNT and RING.

15.7.1 How to use the RLYCNT pin

The RLYCNT pin is used for relay control and dial pulse generation. Relay control is achieved by writing a value to the RLYCNT bit in CTR while the TSW bit in CTR is 0. A RLYCNT pin output is the value set in the RLYCNT bit.

Dial pulses can be generated by setting the TSW bit in CTR to 1 and writing data to buffers of data transmission. At this moment, an Modem interface must be activated and the FS must be accepted after one word of data is transmitted. The value of the last bit is maintained until the next data transmission starts.

15.7.2 How to Use the RING pin

The RING pin is used to detect a ringing signal. For this purpose, an AFE interface issues an interrupt when the RING pin goes low (the RDETM bit must remain at 0). By setting the RSW and the RDETM bits to 1, reading the value of the RING pin at a receive data buffer and calculating the cycles of 0 and 1 can be performed while the ringing signal is detected.

Note that a RING interrupt is canceled when the RDETM bit is set to 1.

Section 16 Keyboard Controller Interface

16.1 Overview

The keyboard controller interface provides an ISA-like interface to connect CPU and keyboard controller.

16.1.1 Features

- Power management function of keyboard controller is supported

16.1.2 Block Diagram

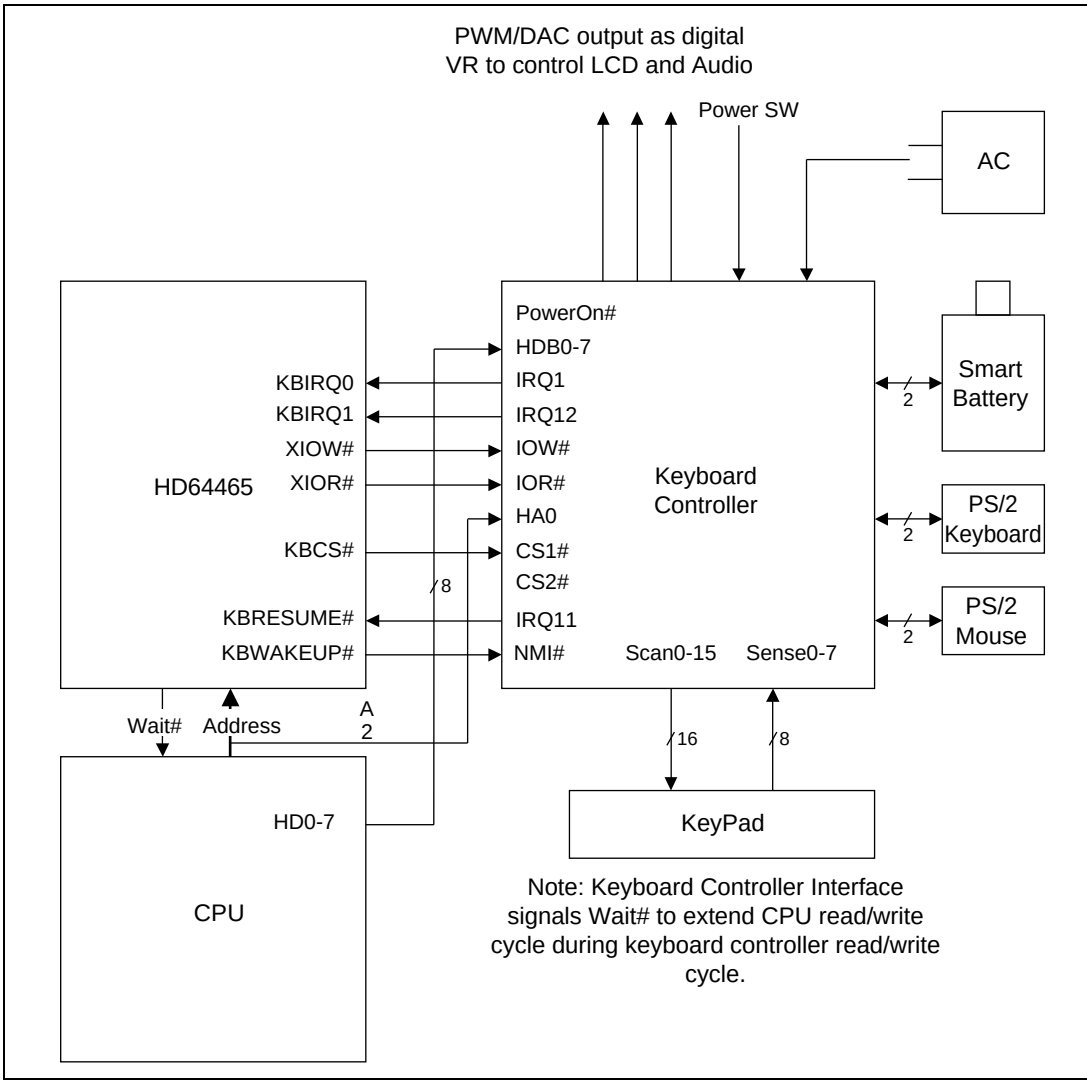


Figure 16.1 H8 Keyboard Controller Interface Block Diagram

Table 16.1 Pin Function of Keyboard Controller Interface Module

Name	I/O	Pin Function
KBCS#	O	Keyboard Controller Chip Select
XIOW#	O	Keyboard Controller Write Enable
XIOR#	O	Keyboard Controller Read Enable
KBIRQ0	I	Keyboard Controller interrupt 0
KBIRQ1	I	Keyboard Controller interrupt 1
KBRESUME#	I	Keyboard Controller Resume
KBWAKEUP#	O	Keyboard Controller wakes up the STANDBY system via signal KBWAKEUP#.

16.2 Register Description

Register or Buffer	Function	Access Size	Address
CR	Control Register	8 bits	H'1000D800
SR	Status Register	8 bits	H'1000D802
H8C1R	H8 Control 1 Register	8 bits	H'1000D000
H8C2R	H8 Control 2 Register	8 bits	H'1000D004

16.2.1 Control Register (CR)

CR, a 16-bit read/write register, is used to control keyboard controller interface. All bits in this register are initialized to 0 at reset. CR is not initialized in STANDBY mode.

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	KCRTTS1	KCRTTS0	IRQ1TTS1	IRQ1TTS0	IRQ0TTS1	IRQ0TTS0
Initial Value	-	-	0	0	0	0	0	0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Description	Default
7 - 6	Reserved	-
5	**KCRESUME Trigger Type Select 1 (KCRTTS1):** 1: If edge trigger is selected, falling edge trigger is selected. If level trigger is selected, low level trigger is selected. 0: If edge trigger is selected, rising edge trigger is selected. If level trigger is selected, high level trigger is selected.	0

Control Register (CR) [cont'd]

Bit	Description	Default
4	**KCRESUME Trigger Type Select 0 (KCRTTS0):** 1: edge trigger is selected. 0: level trigger is selected.	0
3	**IRQ1 Trigger Type Select 1 (IRQ1TTS1):** 1: If edge trigger is selected, falling edge trigger is selected. If level trigger is selected, low level trigger is selected. 0: If edge trigger is selected, rising edge trigger is selected. If level trigger is selected, high level trigger is selected.	0
2	**IRQ1 Trigger Type Select 0 (IRQ1TTS0):** 1: edge trigger is selected 0: level trigger is selected	0
1	**IRQ0 Trigger Type Select 1(IRQ0TTS1):** 1: If edge trigger is selected, falling edge trigger is selected. If level trigger is selected, low level trigger is selected. 0: If edge trigger is selected, rising edge trigger is selected. If level trigger is selected, high level trigger is selected.	0
0	**IRQ0 Trigger Type Select 0(IRQ0TTS0):** 1: edge trigger is selected 0: level trigger is selected	0

16.2.2 Status Register (SR)

SR, an 8-bit Read Only register, is used to reflect the status of keyboard controller interface. All bits in this register are initialized to 0 at reset. SR is not initialized in STANDBY mode.

Bit	7	6	5	4	3	2	1	0
Bit Name	reserved	reserved	reserved	reserved	reserved	KCR	KCIRQ1	KCIRQ0
Initial Value	-	-	-	-	-	0	0	0
R/W	-	-	-	-	-	R	R	R

Bit	Description	Default
7 - 3	Reserved	-
2	**Keyboard Controller Resume (KCR):** 1 indicates that the Keyboard Controller signals resume. Write 1 to clear this bit.	0
1	**Keyboard Controller Interrupt Request 1 (KCIRQ1):** 1 indicates that the Keyboard Controller signals IRQ1. Write 1 to clear this bit.	0
0	**Keyboard Controller Interrupt Request 0 (KCIRQ0):** 1 indicates that the Keyboard Controller signals IRQ0. Write 1 to clear this bit.	0

16.2.3 H8 Control 1 Register (H8C1R)

H8C1R is an 8-bit read/write register. System can write data to IDR1 of H8 via this register and read the data of ODR1 of H8 via this register.

16.2.4 H8 Control 2 Register (H8C2R)

H8C2R is an 8-bit read/write register. System can write command to IDR1 of H8 via this register and read the status of STR1 of H8 via this register.

16.3 Function Description

PB5 can be programmed to be Resume#, and PB4 can be programmed to be WakeUp#, in order to support power management function of H8 keyboard controller.

16.4 Timing Diagram

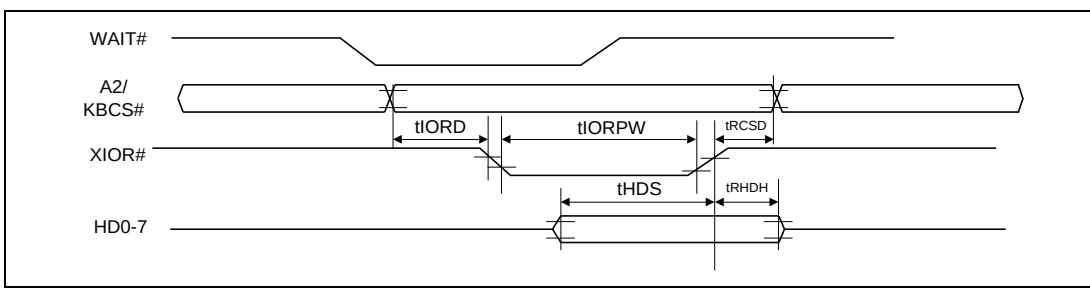


Figure 16.2 Keyboard Controller Interface Read Timing

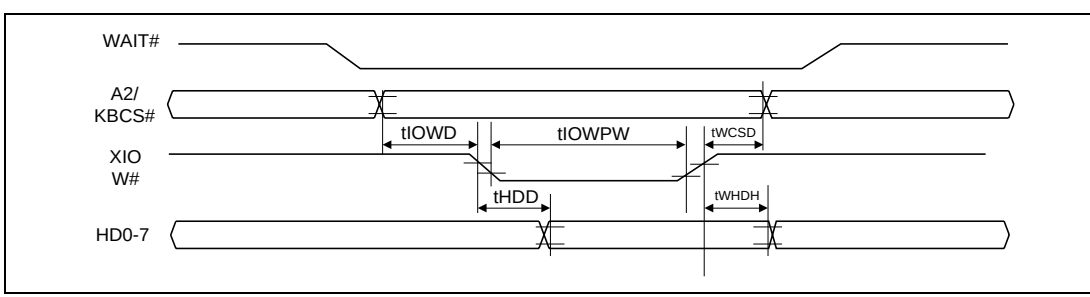


Figure 16.3 Keyboard Controller Interface Write Timing

Table 16.2 Keyboard Controller Interface Read Cycle AC Timing

Item	Symbol	Min	Max	Unit
XIOR# delay time	tIORD	42	-	ns
KBCS#/A2 delay time	tRCSD	45	-	ns
XIOR# pulse width	tIORPW	250	-	ns
HD0-7 setup time	tHDS	-	-	ns
HD0-7 hold time	tRHDH	-	-	ns

Table 16.3 Keyboard Controller Interface Write Cycle AC Timing

Item	Symbol	Min	Max	Unit
XIOW# delay time	tIOWD	41	-	ns
KBCS#/A2 delay time	tWCSD	43	-	ns
XIOW# pulse width	tIOWPW	126	-	ns
HD0-7 delay time	tHDD	-	-	ns
HD0-7 hold time	tWHDH	90	-	ns

Section 17 PS/2 Interface

17.1 Overview

The PS/2 interface is implemented as a Register Interface peripheral. All protocols and timing conform to IBM-PC PS/2 specifications. The interface supports two emulation channels to be connected PS/2 keyboard and PS/2 mouse individually.

17.2 Pin Configuration

Pin Number	Pin Name	I/O	Description
C22	KBCK	IO	Keyboard clock input/output
B22	KBDATA	IO	Keyboard data input/output
A22	MSCK	IO	Mouse clock input/output
D21	MSDATA	IO	Mouse data input/output

17.3 Registers Description

Table 17-1 lists the control registers used by the PS/2 interface. All control registers are 16 bits wide, are aligned to 4-byte boundaries, and control various aspects of the system. These may only be read and written as 16 bit words. Control registers are located in the 0x1000DC00-0x1000DC18 block of the physical address space.

Table 17.1 PS/2 Interface Control Registers

Name	Address	Register Size	Access Size
Keyboard Control/status register(KBCSR)	H'1000DC00	16 bits	16 bits
Keyboard interrupt status register(KBISR)	H'1000DC04	16 bits	16 bits
Mouse Control/status register(MSCSR)	H'1000DC10	16 bits	16 bits
Mouse interrupt status register(MSISR)	H'1000DC14	16 bits	16 bits

17.3.1 Keyboard Control/Status Register (KBCSR)

This keyboard control register contains receive data shift register and keyboard interface control/status bits.

Address: H'1000DC00

Bit	15	14	13	12	11	10	9	8
Bit Name	KBCIE	KBCOE	KBDOE	KBCD	KBDD	KBCS	KBDS	KBDP
Initial Value	0	0	0	0	0	0	0	-
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	KBD7	KBD6	KBD5	KBD4	KBD3	KBD2	KBD1	KBD0
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
15	**KBCK Input Enable Bit (KBCIE)** This bit is used to be KBCK pin input enable control signal. 1: KBCK pin input enable. 0: KBCK pin input is disabled. At transferring data (system->keyboard), this bit must be cleared to 0. This bit is also cleared to 0 when 1-byte of data is received from keyboard.	0
14	**KBCK Output Enable Bit (KBCOE)** This bit is used to be KBCK pin output enable control signal. 1: KBCK pin output enable. 0: KBCK pin output is disabled (signal will go to Hi-Z).	0
13	**KBDATA Output Enable Bit (KBDOE)** This bit is used to be KBDATA pin output enable control signal. 1: KBDATA pin output enable. 0: KBDATA pin output is disabled (signal will go to Hi-Z).	0
12	**KBCK Driven Bit (KBCD)** 1: KBCK signal is driven to high level. 0: KBCK signal is driven to low level.	0
11	**KBDATA Driven Bit (KBDD)** 1: KBDATA signal is driven to high level. 0: KBDATA signal is driven to low level.	0

Keyboard Control/Status Register (KBCSR) [cont'd]

Bit	Description	Default
10	**KBCK Pin Status (KBCS)** 1: KBCK pin is high level. 0: KBCK pin is low level.	0
9	**KBDATA Pin Status (KBDS)** 1: KBDATA pin is high level. 0: KBDATA pin is low level.	0
8	**Parity Bit (KBDP)** This bit indicates the parity bit of the received data. This bit is read-only; writing this bit is not effective.	-
7 - 0	**Received Data (KBD7-KBD0)** Receiving data is stored. This bit is read-only; writing this bit is not effective.	-

17.3.2 Keyboard Interrupt Status Register (KBISR)

This keyboard control register indicates that receive data shift-register is full or not. When this register is set, an interrupt will occur. If the software writes 1 to this register, it can clear the interrupt request.

Address: H'1000DC04

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	KBRDF
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit	Description	Default
15 - 1	**Reserved**	-
0	**Received Data Full (KBRDF)** When the keyboard receive data register is full (one data is received), this bit will be set to "1" and an interrupt request will occur. At the same time, KBCK signal is driven to low automatically. Otherwise, the software can clear the interrupt request by writing "1" to this bit.	0

17.3.3 Mouse Control/Status Register (MSCSR)

This mouse control register contains receive data shift register and mouse interface control/status bits.

Address: H'1000DC10

Bit	15	14	13	12	11	10	9	8
Bit Name	MSCIE	MSCOE	MSDOE	MSCD	MSDD	MSCS	MSDS	MSDP
Initial Value	0	0	0	0	0	0	0	-
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	MSD7	MSD6	MSD5	MSD4	MSD3	MSD2	MSD1	MSD0
Initial Value	-	-	-	-	-	-	-	-
R/W	R	R	R	R	R	R	R	R

Bit	Description	Default
15	**MSCK Input Enable Bit (MSCIE)** This bit is used to be MSCK pin input enable control signal. 1: MSCK pin input enable. 0: MSCK pin input is disabled. At transferring data (system->mouse), this bit must be cleared to 0. This bit is also cleared to 0 when 1-byte of data is received from mouse.	0
14	**MSCK Output Enable Bit (MSCOE)** This bit is used to be MSCK pin output enable control signal. 1: MSCK pin output enable. 0: MSCK pin output is disabled (signal will go to Hi-Z).	0
13	**MSDATA Output Enable Bit (MSDOE)** This bit is used to be MSDATA pin output enable control signal. 1: MSDATA pin output enable. 0: MSDATA pin output is disabled (signal will go to Hi-Z).	0
12	**MSCK Driven Bit (MSCD)** 1: MSCK signal is driven to high level. 0: MSCK signal is driven to low level.	0
11	**MSDATA Driven Bit (MSDD)** 1: MSDATA signal is driven to high level. 0: MSDATA signal is driven to low level.	0

Mouse Control/Status Register (MSCSR) [cont'd]

Bit	Description	Default
10	**MSCK Pin Status (MSCS)** 1: MSCK pin is high level. 0: MSCK pin is low level.	0
9	**MSDATA Pin Status (MSDS)** 1: MSDATA pin is high level. 0: MSDATA pin is low level.	0
8	**Parity Bit (MSDP)** This bit indicates the parity bit of the received data. This bit is read-only; writing this bit is not effective.	-
7 - 0	**Received Data (MSD7-MSD0)** Receiving data is stored. This bit is read-only; writing this bit is not effective.	-

17.3.4 Mouse Interrupt Status Register (MSISR)

This mouse control register indicates that receive data shift-register is full or not. When this register is set, an interrupt will occur. If the software writes 1 to this register, it can clear the interrupt request.

Address: H'1000DC14

Bit	15	14	13	12	11	10	9	8
Bit Name	-	-	-	-	-	-	-	-
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	MSRDF
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Bit	Description	Default
15 - 1	**Reserved**	-
0	**Received Data Full (MSRDF)** When the mouse receive data register is full (one data is received), this bit will be set to "1" and an interrupt request will occur. At the same time, MSCK signal is driven to low automatically. Otherwise, the software can clear the interrupt request by writing "1" to this bit.	0

17.4 Block Diagram

Figure 17-1 shows a block diagram of the PS/2 keyboard or mouse interface. When one byte of data is received from the external keyboard or mouse, the interface generates an interrupt to the CPU.

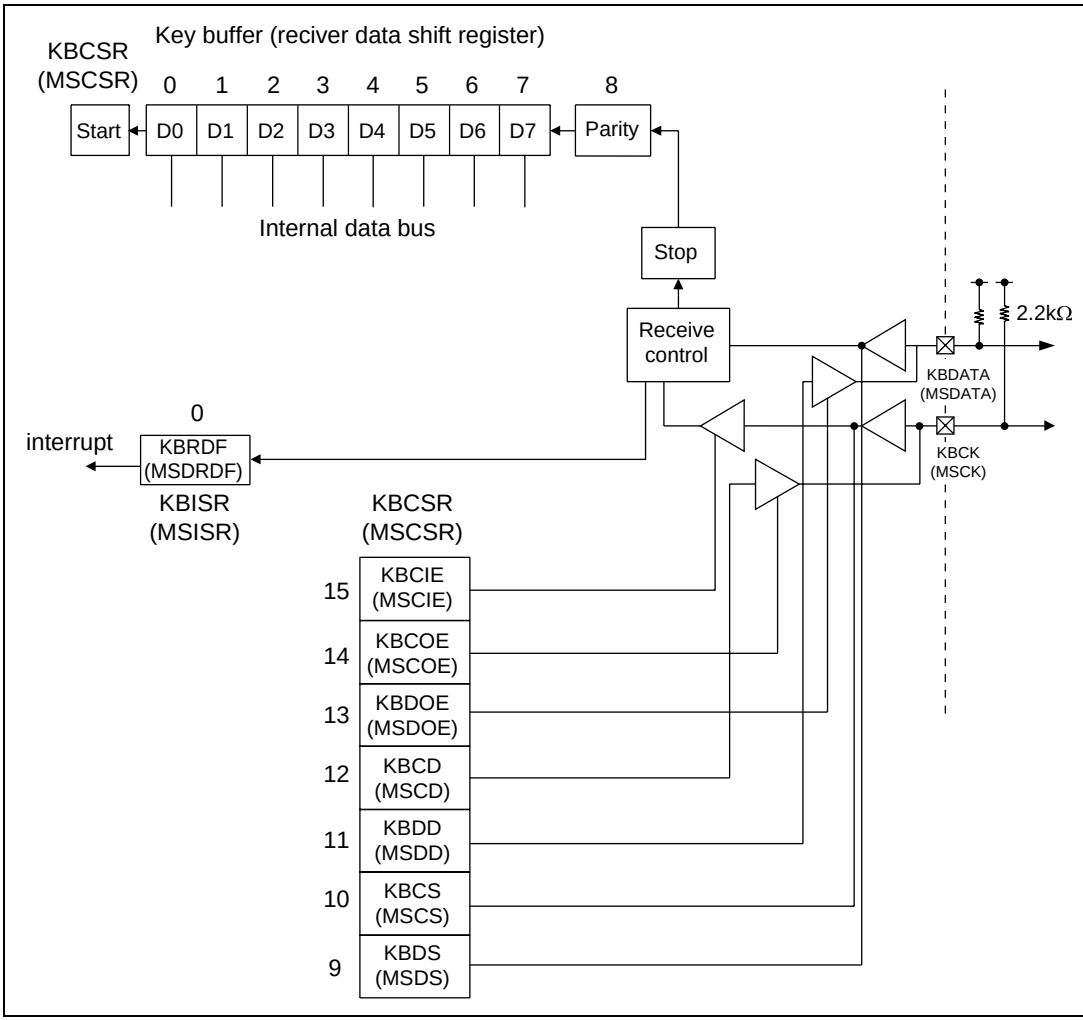


Figure 17.1 PS/2 Keyboard/Mouse Interface Block Diagram

17.5 Operation

The PS/2 interface module supports some registers and hardware circuit to implement standard PS/2 interface. At the following sections, they describe the serial data format, software operational sequence and communication protocol. These descriptions take the keyboard operation as principal thing.

17.5.1 Serial Data Format

The keyboard's serial data format is shown in Figure 17-2. Serial data is asynchronous and consists of 1 start bit, 8 data bits, 1 parity bit, and 1 stop bit. The data-stream order is LSB first. The parity bit is added after the data stream, so that eight data bits and the parity bit always have an odd number of 1s.

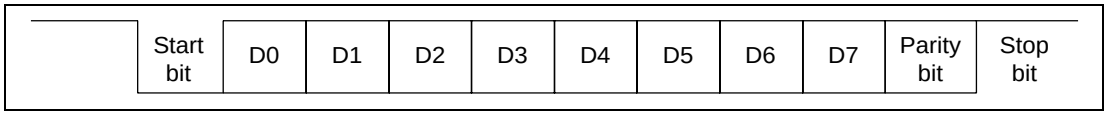


Figure 17.2 Keyboard Serial Data Format

17.5.2 Software Operational Sequence

Receiving Data from the Keyboard

1. Set the KBCIE bit of KBCSR to 1, and clear the KBCOE/KBDOE bit to 0. This enables the system to receive data from the keyboard (when KBCK input is enabled, the signal output buffer is disabled.)

   Example: Set H'8000 to KBCSR.
2. When data is received, an interrupt is set and the scan code of the key is placed in the lower nine bits of KBCSR, including the parity bit. Check the received data full flag (KBRDF, bit 0 of KBISR); it indicates the cause of an interrupt (the flag checks the bit count).
3. Check the parity bit before data processing. The parity bit is indicated by bit 8 of KBCSR. The parity bit indicates an odd parity. If parity is abnormal, software must send the re-send command (H'FE) to the keyboard.
4. After data reception, hardware stops incoming data by driving KBCK signal to low. The KBCIE bit is cleared to 0 automatically.
5. To enter the data reception waiting state, the KBCIE bit must be set again. To clear the cause of the interrupt, software must clear the KBRDF bit.

Transferring Data to the Keyboard:

1. Clear the KBCIE bit to 0, set the KBCOE bit to 1, and clear the KBCD bit to 0. This drives KBCK signal to low level. The keyboard will be inhibited by detecting a low level of KBCK pin. KBCOE works as an enable bit for the output buffer.
2. Set the KBDOE bit to 1. To place the start bit at the KBDATA signal, the KBDD bit must be cleared to 0.
3. After 30 µs, clear the KBCOE bit to 0. This causes the KBCK signal to be pulled up. The keyboard will know that the system has data to send by detecting the KBCK is high and KBDATA is low. The keyboard will sample data while the KBCK signal is high-level.
4. Next, the KBCK signal is driven by the keyboard. During data transfer, software must do the polling of the KBCS bit, which indicates the KBCK signal level. Each time the falling edge of the KBCK signal is detected, the system can send the next bit (set data to KBDD bit).
5. After transferring the stop bit, the keyboard will drive KBDATA signal to low, and the KBCK signal will be counted once. By detecting this status, the system knows that the keyboard has received the data.
6. After data transfer, software must set the KBCIE and KBDOE bits to re-enter the data reception waiting state or data transfer state.

17.5.3 Communication Protocol

Receiving Data from the Keyboard

Figure 17-3 and Table 17-2 show the timing of receiving data from the keyboard. The receiving sequence is as follows:

1. The keyboard must check the CLK signal. If CLK is inactive (low level), the keyboard can not output data.
2. The keyboard must check the DATA signal. If DATA is inactive (low level), the system has data to send. The keyboard receives the data from the system.
3. The keyboard must check the CLK signal during the transmission at intervals within 60•sec. If the keyboard detect the system holding the CLK signal inactive (low), the keyboard stops transmission. The system can stop transmission of the keyboard by holding CLK low before the falling edge of the 10th CLK.
4. The keyboard must check the CLK signal within 5sec after the 10th CLK.
5. The system can inhibit keyboard transmission by setting CLK inactive.
6. If the system has data to transmit, it sets DATA inactive.
7. The system sets the CLK active to allow the next transmission.

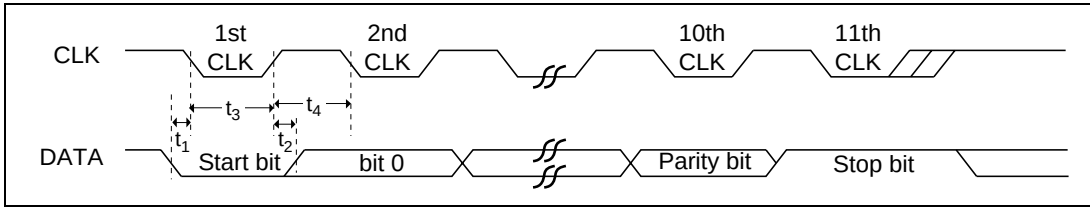


Figure 17.3 Data Receive Timing

Table 17.2 Data Receive Timing Parameters

Timing Parameter	Explanation	Min (us)	Max (us)
t_1	Time from DATA transition to falling edge of CLK	5	25
t_2	Time from rising edge of CLK to DATA transition	5	t4-5
t_3	Duration CLK inactive	30	50
t_4	Duration of CLK active	30	50
t_5	Time to keyboard inhibit after clock 11 to ensure the keyboard does not start another transmission	>0	50

Sending Data from the System

Figure 17-4 and Table 17-3 show the timing for sending data from the system. The sequence is as follows:

1. If the keyboard transmission is in process and beyond the 10th CLK, the system must wait until transmission ends.
2. If the CLK signal is inactive, output from the keyboard is inhibited.
3. The keyboard checks DATA. If data is inactive, the keyboard set CLK inactive. The system places the first bit on DATA.
4. Each time the keyboard sets CLK inactive, the system places the next bit on DATA.
5. While the CLK is active, the keyboard samples the data on DATA. DATA is stable within 1 µs after the rising edge of CLK.
6. The keyboard checks the DATA signal after the 10th CLK. If DATA is inactive, the keyboard sets DATA inactive for one clock.
7. If DATA is inactive after the 10th CLK, the keyboard waits until DATA becomes active and sends a re-send command to the system.

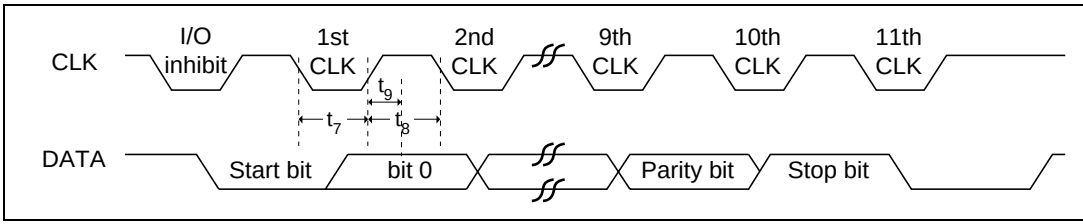


Figure 17.4 Data Send Timing

Table 17.3 Data Send Timing Parameters

Timing Parameter	Explanation	Min (us)	Max (us)
t_7	Duration of CLK inactive	30	50
t_8	Duration of CLK active	30	50
t_9	Time from inactive to active CLK transmission, used to time when the keyboard samples DATA	5	25

17.6 CAUTION

System cannot recognize mouse without confirmation using data-packet between system and mouse.

[Example]

1. System resets PS/2 mouse sending 0xff.
2. System wait to receive ACK.
3. Wait
4. System send command
5. System wait to receive ACK.

:
:
:

Section 18 USB Host Controller

18.1 Introduction

18.1.1 Device Description / Purpose

This USB Host Controller is a PCI-based implementation of the Universal Serial Bus (USB) 1.0 specification utilizing the OpenHCI standard developed by Compaq, Microsoft, and National Semiconductor. It contains an integrated Root Hub with two USB ports, PCI interface, and USB Host Controller. Keyboard and Mouse legacy support are also included for DOS compatibility with USB devices. This USB Host Controller supports limited number of endpoints (not full support) with limited buffer size (4 kbyte).

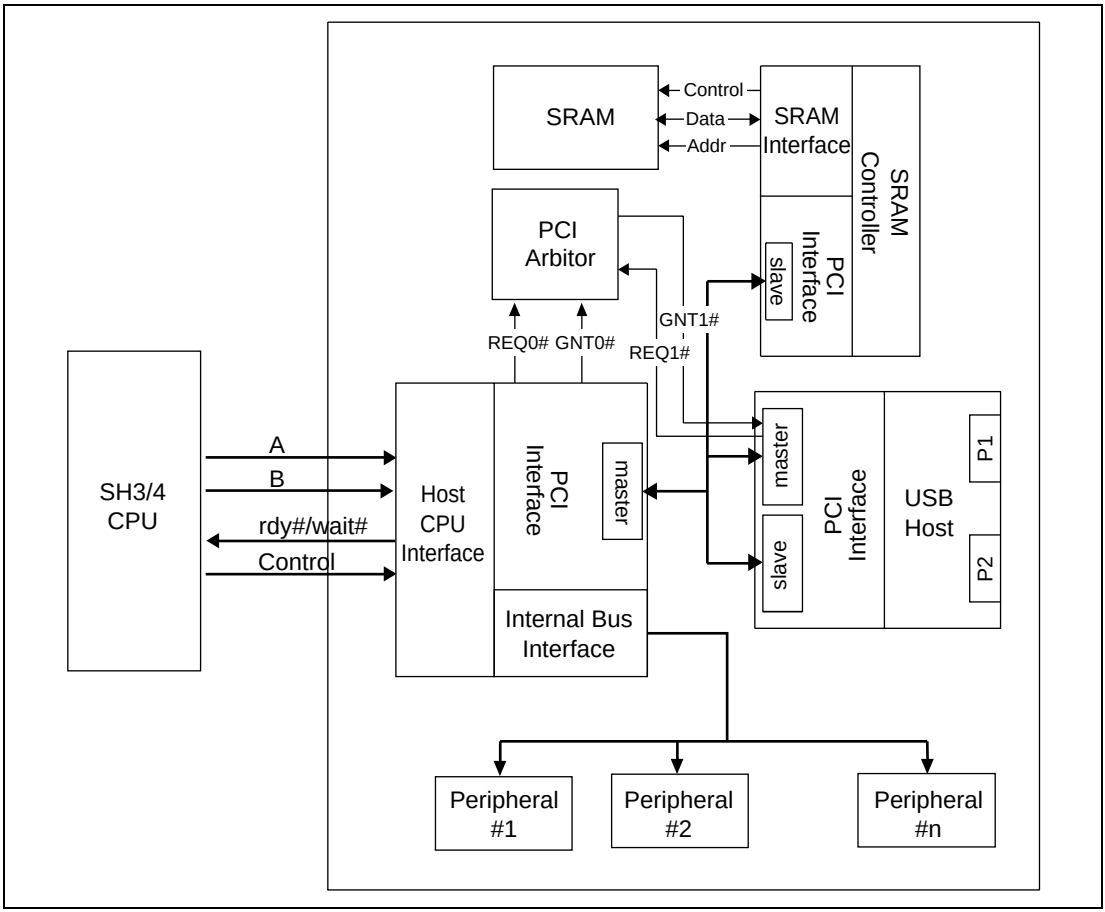
18.1.2 Reference Information

- “USB Specification”, Version 1.0
- “OpenHCI Specification”, Version 1.0
- “PCI Specification”, Version 2.1

18.2 Function Description

18.2.1 System Architecture

The following diagram shows the whole system architecture, which relates the Host CPU(Hitachi SH4 CPU) and HD64465 USB Host Controller by built-in Host/PCI bridge and SRAM Controller. It means that the internal SRAM memory space is the communication channel between Host CPU and USB Host Controller. The Host CPU may write the ED and TD data into SRAM memory then the USB Host read the data structure from SRAM memory and write back the operation result. Again, Host CPU read the result data from SRAM memory. Hence, the communication channel is setup through this methodology.



18.2.2 USB Host Controller

USB States

The Host Controller has four USB states visible to the Host Controller Driver via the Operational Registers: USBOPERATIONAL, USBRESET, USBSUSPEND, and USBRESUME. These states define the Host Controller responsibilities relating to USB signaling and bus states.

The USB states are reflected in the **HostControllerFunctionalState** field of the *HcControl* register. The Host Controller Driver is permitted to perform only the USB state transitions shown in Figure 18-1. The Host Controller may only perform a single state transition. During a remote wakeup event, the Host Controller may transition from USBSUSPEND to USBRESUME.

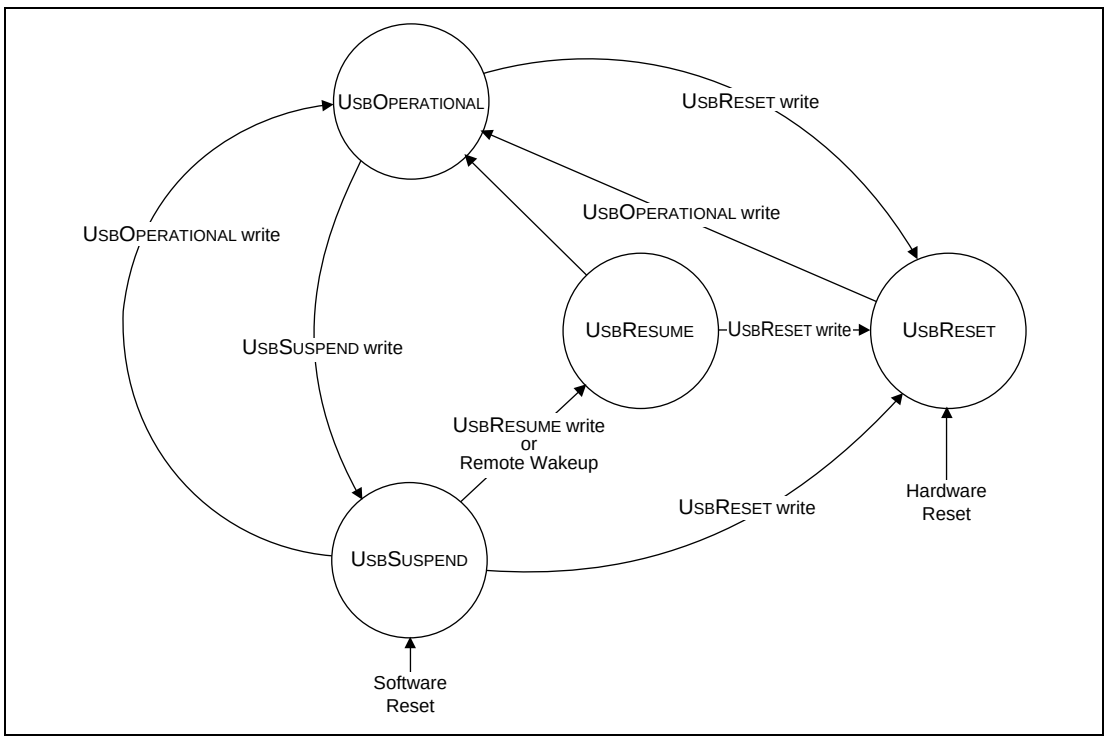


Figure 18.1 USB States

USB OPERATIONAL

When in the USBOPERATIONAL state, the Host Controller may process lists and will generate SOF Tokens. The USBOPERATIONAL state may be entered from the USBRESUME or USBRESET states. It may be exited to the USBRESET or USBSUSPEND states.

When transitioning from USBRESET or USBRESUME to USBOPERATIONAL, the Host Controller is responsible for terminating the USB reset or resume signaling as defined in the USB Specification prior to sending a token.

A transition to the USBOPERATIONAL state affects the frame management registers of the Host Controller. Simultaneously with the Host Controller state transition to USBOPERATIONAL, the **FrameRemaining** field of *HcFmRemaining* is loaded with the value of the **FrameInterval** field in *HcFmInterval*. There is no SOF Token sent at this initial load of the **FrameRemaining** field. The first SOF Token sent after entering the USBOPERATIONAL state is sent following next frame boundary when **FrameRemaining** transitions from 0 to **FrameInterval**. The **FrameNumber** field of *HcFmNumber* is incremented on a state transition to USBOPERATIONAL.

USB RESET

When in the USBRESET state, the Host Controller forces reset signaling on the bus. The Host Controller list processing and SOF Token generation are disabled while in USBRESET. In addition, the **FrameNumber** field of *HcFmNumber* does not increment while the Host Controller is in the USBRESET state. The USBRESET state can be entered from any state at any time. The Host Controller defaults to the USBRESET state following a hardware reset. The Host Controller Driver is responsible for satisfying USB Reset signaling timing defined by the USB Specification.

USB SUSPEND

The USBSUSPEND state defines the USB Suspend state. The Host Controller list processing and SOF Token generation are disabled. However, the Host Controller remote wakeup logic must monitor USB wakeup activity. The **FrameNumber** field of *HcFmNumber* does not increment while the Host Controller is in the USBSUSPEND state.

USBSUSPEND is entered following a software reset or from the USBOPERATIONAL state on command from the Host Controller Driver. While in USBSUSPEND, the Host Controller may force a transition to the USBRESUME state due to a remote wakeup condition. This transition may conflict with the Host Controller Driver initiating a transition to the USBRESET state. If this situation occurs, the HCD-initiated transition to USBRESET has priority. The Host Controller Driver must wait 5 ms after transitioning to USBSUSPEND before transitioning to the USBRESUME state. Likewise, the Root Hub must wait 5 ms after the Host Controller enters USBSUSPEND before generating a local wakeup event and forcing a transition to USBRESUME. Following a software reset, the Host Controller Driver may cause a transition to USBOPERATIONAL if the transition occurs no more than 1 ms from the transition into USBSUSPEND. If the 1-ms period is violated, it is possible that devices on the bus will go into Suspend.

USB RESUME

When in the USBRESUME state, the Host Controller forces resume signaling on the bus. While in USBRESUME, the Root Hub is responsible for propagating the USB Resume signal to downstream ports as specified in the USB Specification. The Host Controller's list processing and SOF Token generation are disabled while in USBRESUME. In addition, the **FrameNumber** field of *HcFmNumber* does not increment while the Host Controller is in the USBRESUME state.

USBRESUME is only entered from USBSUSPEND. The transition to USBRESUME can be initiated by the Host Controller Driver or by a USB remote wakeup signaled by the Root Hub. The Host Controller is responsible for resolving state transition conflicts between the hardware wakeup and Host Controller Driver initiated state transitions. Legal state transitions from USBRESUME are to USBRESET and to USBOPERATIONAL.

The Host Controller Driver is responsible for USB Resume signal timing as defined by the USB Specification.

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.

List Control Block

1) Priority

The list priorities are modified as endpoints are serviced and at periodic intervals. In each frame, an interval of time is reserved for processing of items in the Control and Bulk lists. This interval is at the beginning of each frame. The Host Controller Driver limits this time by writing *HcPeriodicStart* with a bit time after which periodic transfers (Interrupt and Isochronous) have priority for use of the bus. During periodic list processing, the Interrupt list specific to the current frame is serviced before the Isochronous list. When processing of the periodic lists ends, processing of the Control and Bulk lists resumes. Figure 18-2 shows the priority among periodic lists and non-periodic lists within a single frame.

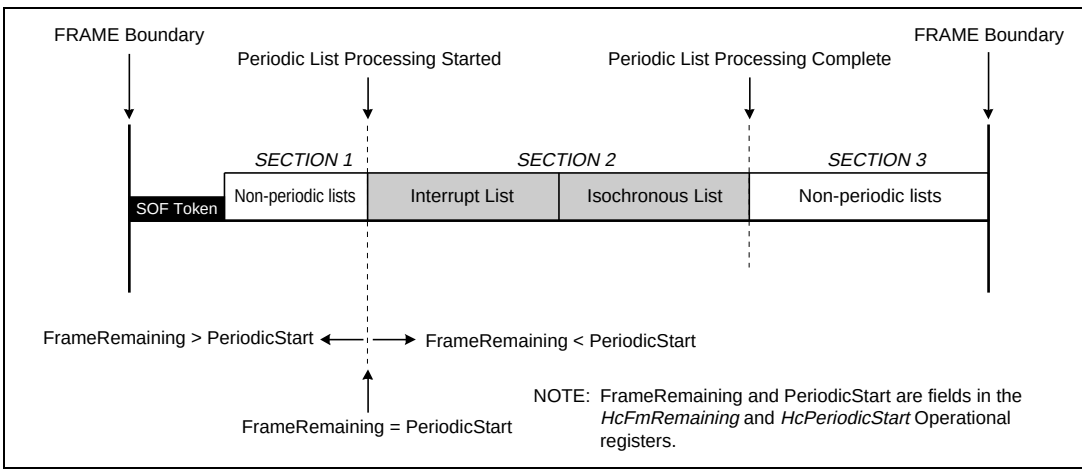


Figure 18.2 List Priority within a USB Frame

a) List Priority

The lists built up by the Host Controller Driver are classified as either periodic or non-periodic. The Interrupt list and the Isochronous list are periodic because the endpoints on those lists require service at specific times in a deterministic manner. The Control list and the Bulk list are non-periodic because endpoints on those lists can tolerate latency and expect service only on a time-available basis.

The Host Controller breaks the USB frame up into three distinct sections with regard to list service as shown in Figure 18-2. Section 1 is devoted to the non-periodic lists. This is followed by Section 2, which is a section reserved for the periodic lists in which both the Interrupt list and the Isochronous list are serviced to completion. Section 3 of the frame is again devoted to the non-periodic lists.

- Periodic Lists

  The list priority between the periodic lists is fixed with the Interrupt list having priority over the Isochronous list. When servicing the periodic lists, the Host Controller is actually servicing a single list, called the Periodic list, which contains both Interrupt Endpoint Descriptors and Isochronous Endpoint Descriptors. The Host Controller Driver ensures that all Interrupt Endpoint Descriptors are placed on the list in front of any Isochronous Endpoint Descriptors.

- Non-periodic Lists

  The priority algorithm between the non-periodic lists is more complicated than that of the periodic lists. Control endpoints are given equal or more access to the bus in comparison with Bulk Endpoints. More specifically, N Control Endpoints are given access to the bus for every 1 Bulk Endpoint. This ratio of N:1 is termed the Control Bulk Service Ratio. The Host Controller Driver has control over the Control Bulk Service Ratio via the **ControlBulkServiceRatio** field of the *HcControl* Register. The range of possible Control Bulk Service Ratios is from 1:1 to 4:1. An example of a 4:1 Control/Bulk Service Ratio is shown in Figure 18-3.

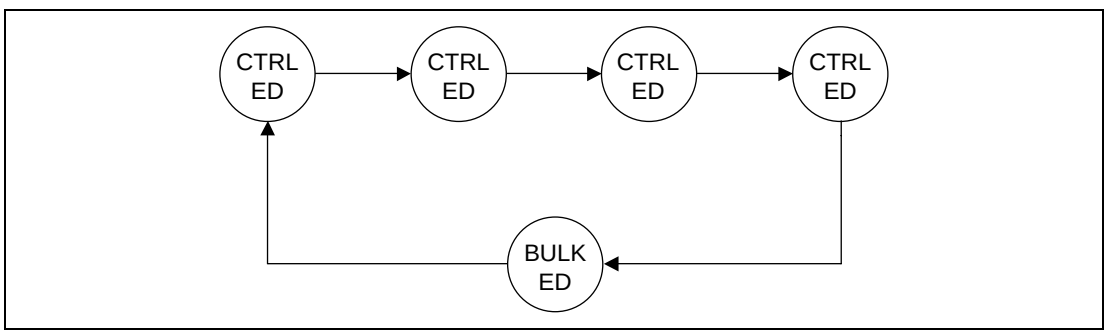


Figure 18.3 Example of Control/Bulk Service Ratio of 4:1

The Host Controller enforces the Control Bulk Service Ratio regardless of the number of Control and Bulk Endpoint Descriptors present on their respective lists. If there is only 1 CTRL ED (Control Endpoint Descriptor) on the Control list and the Control Bulk Ratio is 4:1, that CTRL ED is serviced 4 times before a BULK ED (Bulk Endpoint Descriptor) is serviced. If there are no Endpoint Descriptors on the Control list or the Bulk list when the Host Controller attempts to service that list, the Host Controller will "skip" that list and immediately begin servicing the other non-periodic list and complete the required number of EDs. The Host Controller will continue to check the empty list when ever the Control Bulk Service Ratio dictates, servicing any new Endpoint Descriptors according to the Control Bulk Service Ratio.

b) Interface to ED Block

The List Control block initiates all list processing. There is a series of requests and acknowledge signals passed from the List Control Block to the ED Block to the TD Block to enable correct timing for processing and data requests between the three blocks.

After determining a list is "active" the List Control Block signals the ED Block that an ED needs to be fetched by providing a "list Active" signal for 1 of 3 lists, the Control list, the Bulk list, or the Periodic list. The ED Block will begin fetching Endpoint Descriptors for the "active" list.

The ED Block assists the List Control block by providing three flags. The first flag is a "Isochronous ED" flag for management of the Periodic list. The second flag is an "ED Service Complete" flag, which is used as a handshake so the List Control Block knows when to switch between the non-periodic lists. The third flag is a "list Empty" flag.

ED Block

The ED Block is responsible for processing Endpoint Descriptors. This includes reading EDs from memory, servicing an active TD if necessary, and writing back information if necessary.

1) List Service Flow

This section describes the actions required of the Host Controller during list processing. These actions are taken after the Host Controller has determined which particular list is to be serviced. The general list service flow is depicted in Figure 18-4.

— List Enabled Check

The first action the Host Controller takes when processing a list is to check that the list is enabled. Periodically, lists are disabled by the Host Controller Driver for the purpose of altering an Endpoint Descriptor (or other reasons). If the list is enabled, the Host Controller may service the list. If the list is disabled, the Host Controller skips that list and moves on to the next list. Lists are enabled/disabled with the list enable bits of the *HcControl* register. When a list is disabled during a frame, the Host Controller must not process the list beyond the next frame boundary. When a list is enabled, it is not available for processing until the next frame. In addition, when a list is enabled after being previously disabled, the only piece of information the Host Controller may assume is valid is the list **HeadED** pointer and, if a non-periodic list, the list **CurrentED** pointer.

The **IsochronousListEnable** bit is used to disable processing of the Isochronous list which is always at the tail of the periodic list. If the Host Controller finds an Isochronous Endpoint Descriptor while servicing the Periodic list and the **IsochronousListEnable** bit is '0', the Host Controller stops processing the list.

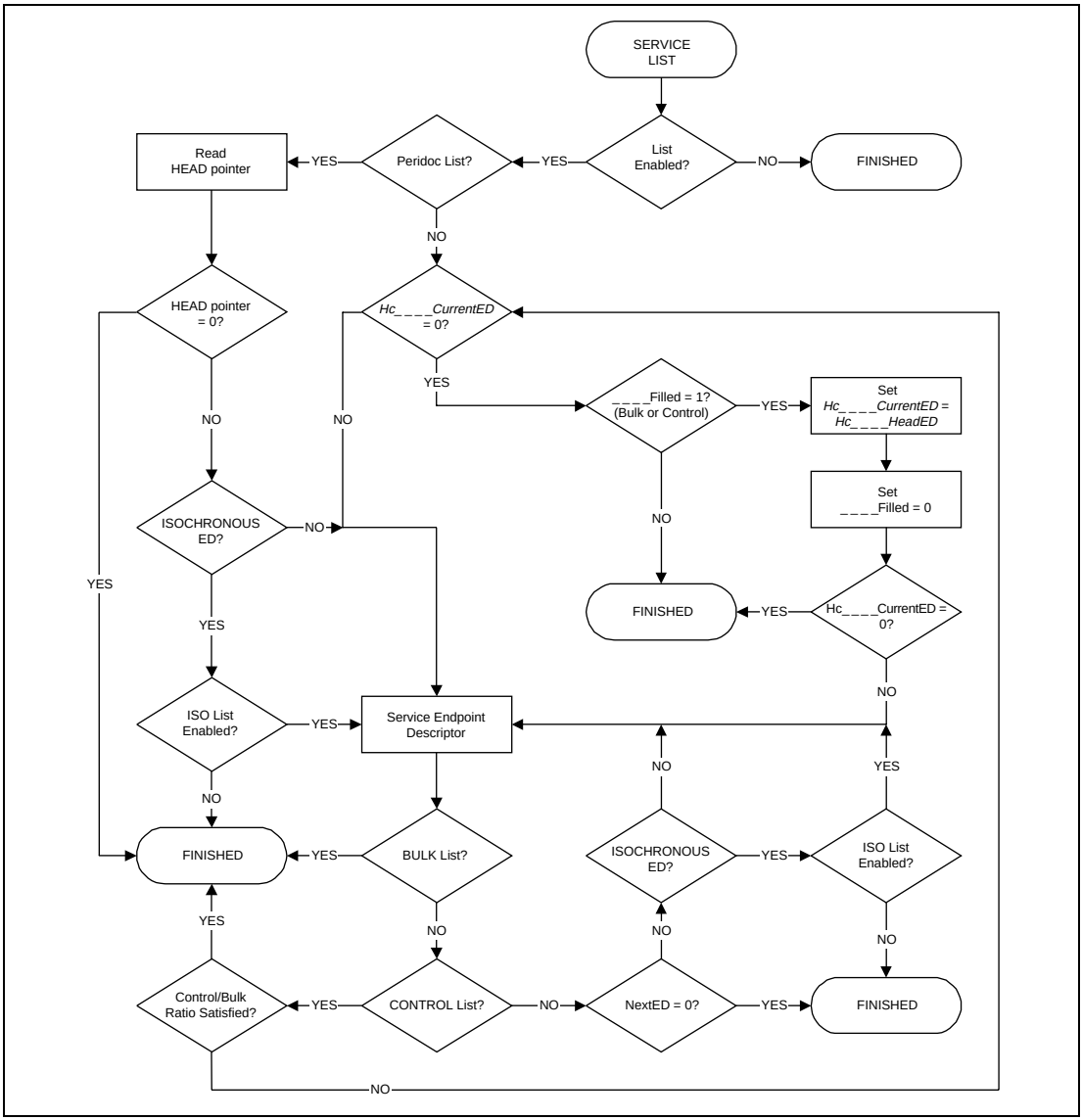


Figure 18.4 List Service Flow

— Locating Endpoint Descriptors

After determining a list is enabled, the Host Controller locates the first Endpoint Descriptor requiring service. The first time the Host Controller services a list after entering the USBOPERATIONAL state, it uses the lists Head Pointer to locate the first Endpoint Descriptor on the list. If the Head Pointer is '0', there are no Endpoint Descriptors on the list and the Host Controller proceeds to the next list.

The Host Controller always uses the Head Pointer to find the first Endpoint Descriptor when servicing the Interrupt (periodic) list. All of the Interrupt Head Pointers are located in the ***HccaInterruptTable***. The Host Controller makes a determination of which Head Pointer to use by using the low order 5 bits of the **FrameNumber** field of *HcFmNumber* as a Dword index into the table. An index value of '00000' binary corresponds to the Head Pointer (Dword) at offset '0x00.' An index value of '11111' binary corresponds to the Head Pointer (Dword) at offset '0x7C.'

In the case of the non-periodic lists, the operation is slightly different. Since the non-periodic lists are serviced on a time-available basis, the Host Controller may not be able to service an entire list within a single frame. In order to satisfy the requirement of servicing Endpoint Descriptors in a round-robin priority, the Host Controller maintains "CurrentED" pointers for each list (the *HcBulkCurrentED* register and the *HcControlCurrentED* register). These pointers always point to the next Endpoint Descriptor requiring service on their respective list. When servicing the non-periodic lists, the Host Controller checks the *HcBulkCurrentED* or *HcControlCurrentED* register to see if there is a non-zero value. If the value of the "CurrentED" register contains a non-zero pointer to an Endpoint Descriptor, the Host Controller attempts to process that Endpoint Descriptor. If the "CurrentED" register contains a value of '0,' the Host Controller has reached the end of the list. At this point, the Host Controller checks the **BulkListFilled** bit or **ControlListFilled** bit of the *HcCommandStatus* register. If the respective filled" bit is set to '1,' there is at least one Endpoint Descriptor on the list which needs service. In this case, the Host Controller will copy the value of *HcControlHeadED* or *HcBulkHeadED* into *HcControlCurrentED* or *HcBulkCurrentED* respectively, clear the "filled" bit to '0,' and attempt to process the Endpoint Descriptor now present in the "CurrentED" register. If the "filled" bit is a '0' when checked, there are no Endpoint Descriptors on the list needing service and the Host Controller moves on to the next list.

After servicing an Endpoint Descriptor, the Host Controller proceeds differently, depending on the list type.

1. If the current list is the periodic list, the Host Controller checks the **NextEndpointDescriptor** pointer of the just completed Endpoint Descriptor. If non-zero, the Host Controller continues processing with the next Endpoint Descriptor. If zero, the Host Controller moves on to the non-periodic lists.
2. If the current list is the Bulk list, after servicing a single Endpoint Descriptor the Host Controller moves on to the Control list.

3. If the current list is the Control List, the Host Controller next action is dependent on whether or not the number of Control Endpoint Descriptors dictated by the Control/Bulk Service Ratio have been serviced. If the Control/Bulk Service Ratio has been satisfied, the Host Controller moves on to the Bulk list; otherwise, service of another Control Endpoint Descriptor is attempted.

— Endpoint Description Priority

Within a list, Endpoint Descriptors are serviced with a round robin priority scheme. The Host Controller must initially begin service at the head of the list and service each Endpoint Descriptor on the list sequentially. For non-periodic lists, when the Host Controller reaches the end of the list, it reads the list? Head Pointer and starts again with the first Endpoint Descriptor on the list. Servicing an Endpoint Descriptor is defined as making a single transaction attempt from the first Transfer Descriptor in the queue. Once a transaction attempt is made, whether successful or not, and the appropriate actions are taken to complete that transaction, the Host Controller will service the next Endpoint Descriptor rather than make a second transaction attempt on the current Endpoint Descriptor. The only exception to this is for an Isochronous TD that is late. The late Isochronous TD will be retired and the next Isochronous TD connected to the Endpoint Descriptor will be processed immediately.

2) Endpoint Description Processing

During the processing of a list, the Host Controller is required to interpret and service the Endpoint Descriptors present on that list. The flow for service of an Endpoint Descriptor is shown in Figure 18-5.

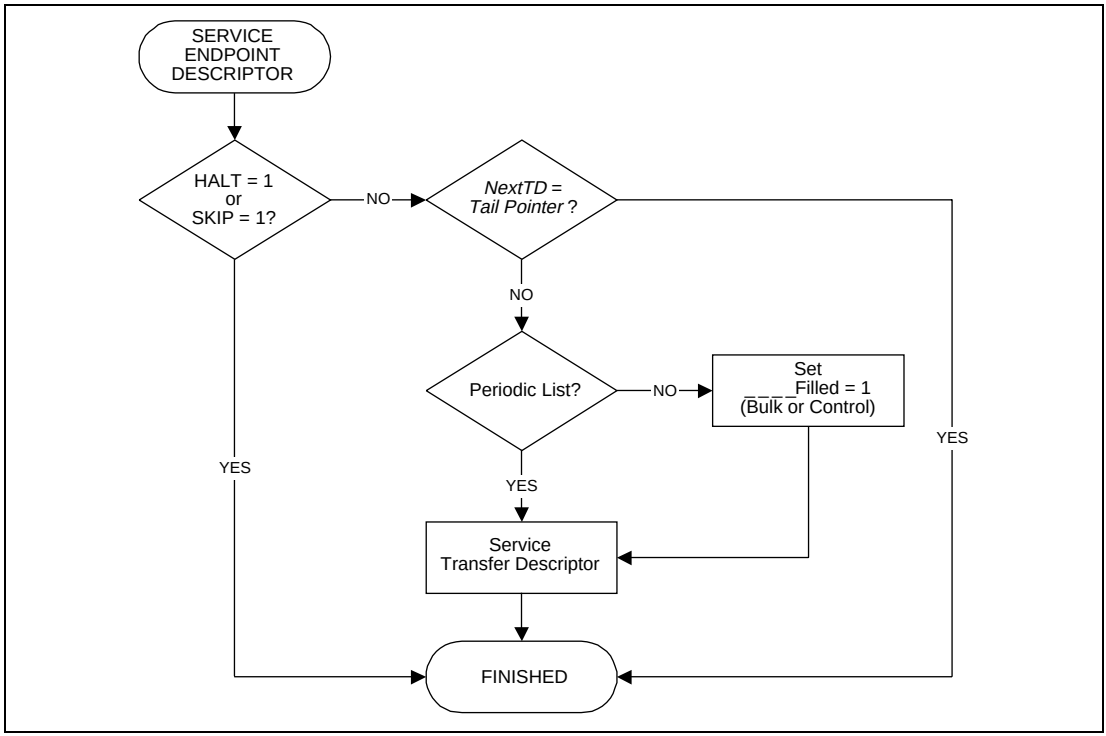


Figure 18.5 Endpoint Descriptor Service Flow

When the Host Controller reads an Endpoint Descriptor, it first determines if the descriptor should be skipped. If either the **sKip** bit or the **Halt** bit in the Endpoint Descriptor is a '1,' the Endpoint Descriptor is skipped and the Host Controller proceeds normally with the next Endpoint Descriptor or the next list. If the Endpoint Descriptor is not skipped, the Host Controller performs a check to determine if there is a Transfer Descriptor on the queue. If not, the Host Controller proceeds to the next Endpoint Descriptor or the next list.

To determine if there is a Transfer Descriptor on the queue that can be processed, the Host Controller compares the Endpoint Descriptor? **TailPointer** and **NextTransferDescriptor** fields. If the fields are different, there is a Transfer Descriptor available for processing. If they are equal, there is not a valid Transfer Descriptor on the list. If a valid Transfer Descriptor is present on the queue, the Host Controller attempts to service that Transfer Descriptor. Service of the Transfer Descriptor involves making only a single transaction attempt.

— Transfer Description Priority

The priority of Transfer Descriptors on a queue is first-come-first-serve. The Transfer Descriptors the Host Controller services are always part of a queue attached to an Endpoint Descriptor. The Host Controller services the first Transfer Descriptor on the queue which is pointed to by the **NextTransferDescriptor** field of the Endpoint Descriptor. When that Transfer Descriptor is retired, it is removed from the queue and the Transfer Descriptor linked with the **NextTransferDescriptor** field of that Transfer Descriptor is moved to the front of the queue.

3) Interface to the TD Block

The ED Block is responsible for providing the TD block with a “o” signal. When the ED Block determines there is a Transfer Descriptor needing service it signals the TD Block to fetch that TD and begin processing. The ED Block provides the address of the Transfer Descriptor and all other required information from the Endpoint Descriptor to the TD Block along with the “o” flag. When processing of the Transfer Descriptor is complete the TD Block responds with a “ED complete” flag.

4) Operational Registers

— HcHCCA

This register is a pointer to the base address of the OpenHCI defined HCCA and is defined by software. When the Host Controller indexes into the Interrupt Table to fetch the Periodic List head pointer, the index offset is added to this value to determine the physical address of the list head pointer.

— HcBulkHeadED

This register is a pointer to the head ED of the Bulk list and is maintained by software.

— HcControlHeadED

This register is a pointer to the head ED of the Control list and is maintained by software.

— HcPeriodCurrentED

When the Host Controller processes the Periodic List, the address of the ED being serviced is stored here. The Host Controller maintains this register without interference from software.

— HcBulkCurrentED

When the Host Controller processes the Bulk List, the address of the ED being serviced is stored here. When the Host Controller moves to the next ED, the **NextED** field of the previous ED is copied into this register. This allows the Host Controller to keep its place when processing of the Bulk List is interrupted.

— HcControlCurrentED

When the Host Controller processes the Control List, the address of the ED being serviced is stored here. When the Host Controller moves to the next ED, the **NextED** field of the previous ED is copied into this register. This allows the Host Controller to keep its place when processing of the Control List is interrupted.

5) Description Registers

The ED Block maintains 4 32-bit registers to hold requested Endpoint Descriptors. These registers are used as temporary storage for Endpoint Descriptors being serviced and are not addressable by software. Figure 18-6 shows the fields present in each of the 4 Endpoint Descriptor registers.

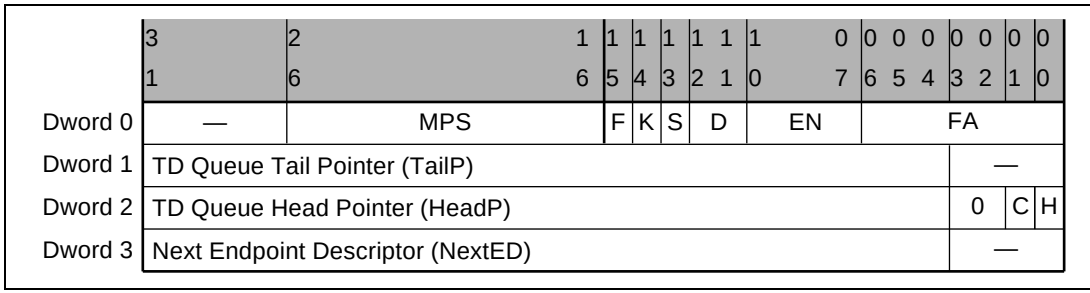


Figure 18.6 Endpoint Descriptor

Please refer to the OpenHCI specification for details on particular field names and functions.

TD Block

The TD Block is responsible for processing Transfer Descriptors. This is the bulk of the work performed by the Host Controller. This block is responsible for reading Transfer Descriptors from memory, requesting transactions when appropriate, writing back status, and retiring TDs when necessary (including the management of the Done Queue).

1) Transfer Description Processing

Transfer Descriptor processing is the fundamental operation performed by a Host Controller. The service flow for servicing a Transfer Descriptor is shown in Figure 18-7 below. The rest of this section describes the steps necessary for completing service of a Transfer Descriptor.

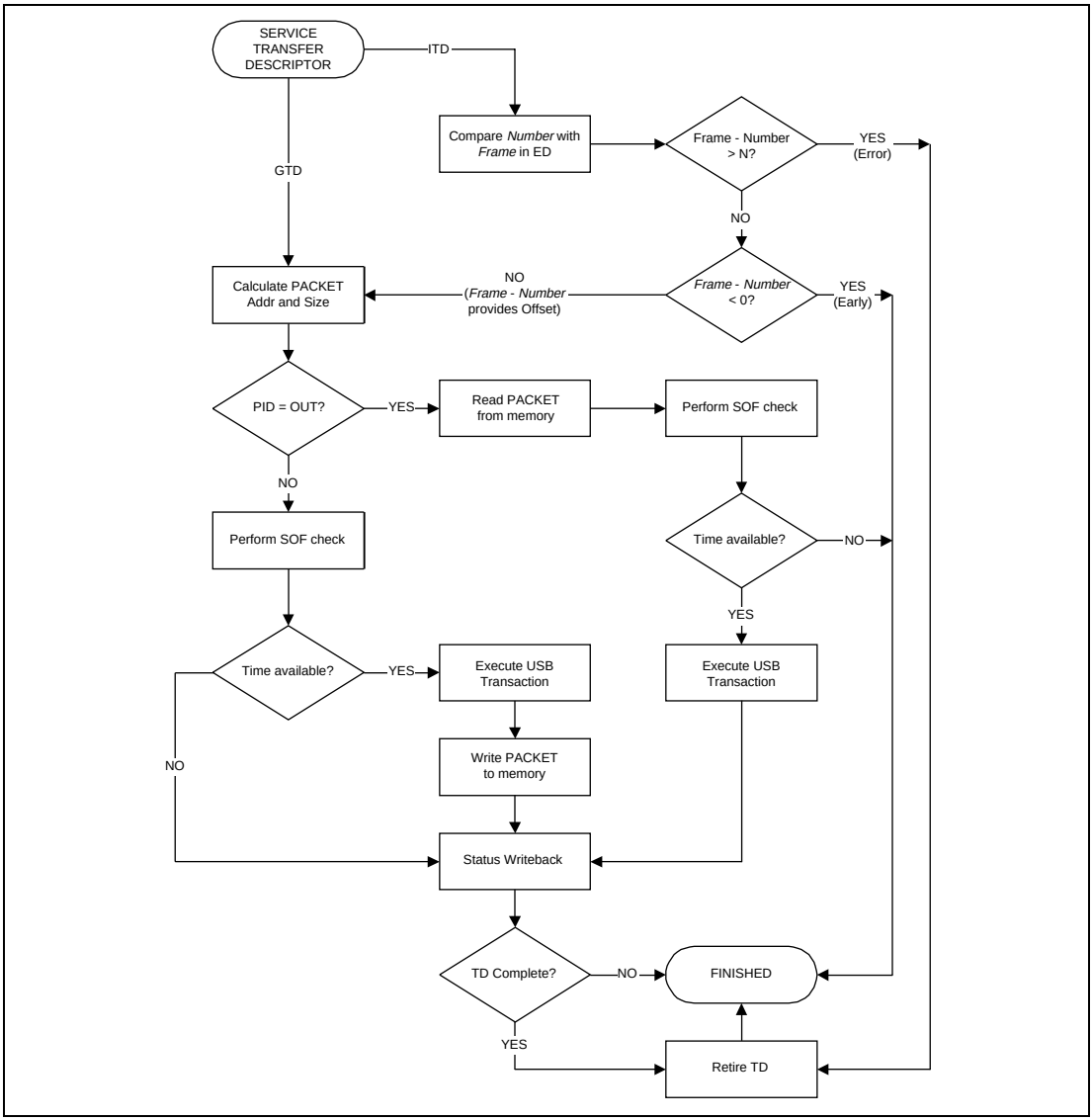


Figure 18.7 Transfer Description Service Flow

a) Isochronous Relative Frame Number Calculation

When processing an Isochronous Transfer Descriptor, the Host Controller must calculate the relative frame number. This calculation determines which, if any, packet will be sent during the current frame.

An Isochronous TD may contain buffers for 1 to 8 consecutive frames of data (**FrameCount**+1) with the first (0th) data packet of an Isochronous TD sent in the frame for which **FrameNumber** match the **StartingFrame** field of the Isochronous TD.

The Host Controller does an unsigned subtraction of **StartingFrame** from **FrameNumber** to arrive at a signed value for a relative frame number, R.

1. If the relative frame number is negative, then the current frame is earlier than the 0th frame of the Isochronous TD and the Host Controller advances to the next ED.
2. If the relative frame number is greater than **FrameCount,** then the Isochronous TD is late and a error condition exists. The current ITD is retired, and the next ITD on the list for the current ED is processed immediately.
3. If the relative frame number is between 0 and **FrameCount,** then the Host Controller issues a token to the endpoint and attempts a data transfer. See next section for the packet buffer address calculation.

   When the last data packet of an Isochronous TD is transferred, the Isochronous TD is retired to the Done Queue.

Table 18.1 Example Calculation of R and Host Controller Action

HcFmNumber	ITD.Frame	R	ITD.FC	HC Action
H'FFFC	H'FFFE	H'FFFE (-2)	3	Do nothing
H'FFFD	H'FFFE	H'FFFF (-1)	3	Do nothing
H'FFFE	H'FFFE	H'0000	3	Send data packet 0
H'FFFF	H'FFFE	H'0001	3	Send data packet 1
H'0000	H'FFFE	H'0002	3	Send data packet 2
H'0001	H'FFFE	H'0003	3	Send data packet 3 and retire Isochronous TD

b) Isochronous Packet Begin and End Address Calculation

When processing an Isochronous Transfer Descriptor, the RelativeFrame (R) is used to select two offset values, BeginAddrOffset and EndAddrOffset. EndAddrOffset is the lower 12 bits of the last byte of the packet +1. The source of these offsets is shown in Table 18-2. Refer to next section for a description of the Descriptor Registers used.

Table 18.2 ITD Packet Offset Location

RelativeFrame Is Even	RelativeFrame Equals FrameCount	BeginAddrOffset	EndAddrOffset
Yes	No	Offset0 Low	Offset0 High
No	No	Offset0 High	Offset1
Yes	Yes	Offset0 Low	BEOffset
No	Yes	Offset0 High	BEOffset

The initial address of the transfer is determined from Offset[R]. If bit 12 (the 13th LSb) of Offset[R] is 0, then the initial buffer address resides in the physical memory page specified in **BufferPage0** of the Isochronous Transfer Descriptor. If bit 12 is 1, then the initial buffer address will reside in the physical page indicated by **BufferEndPage**. When the upper 20 bits are selected, the address is completed by using BeginAddrOffset as the low 12 bits of the address.

c) General Last Packet Calculation

The **GTDLastPacket** signal is used for two purposes.

- Setting the **CBP** value to 0h upon a successful transfer.
- Limiting the transfer size of the last packet to be the remaining space in the buffer instead of **MaxPacketSize.**

GTDLastPacket is asserted when the size of the packet is zero-length (indicated by **CBP** = 0h) or when the last byte of the buffer lies within the address range **CBPOffset** plus **MaxPacketSize.**

This latter condition occurs when the 13 bit value of **CBPOffset** plus **MaxPacketSize** is greater than or equal to **BEOffset_12** concatenated with **BEOffset,** with one exception, as stated below. **BEOffset_12** is the buffer end relative page. It is set when **BEOffset** does not reside in the same page as **CBPOffset**. There are two conditions for setting **BEOffset_12**: **CBPPage** = **BEPage,** and the increment of the end address to form **BEOffset** crosses a page boundary.

The exception for setting **GTDLastPacket** is when the two conditions for setting **BEOffset_12** are both valid. This indicates that the buffer end is at the end of the second page, and there is at least one page worth of data remaining in the buffer.

d) General Packet Begin and End Address Calculation

When the Host Controller fetches a General Transfer Descriptor, it gets the address of the next memory location to be accessed from **CurrentBufferPage** and **CurrentBufferOffset**. If these values are both 0, then the packet size will be zero, regardless of the setting of **MaximumPacketSize** in the Endpoint Descriptor. As the data is transferred it might cross a page boundary. If it does, **BufferEndPage** is substituted for the current **CurrentBufferPage**. This page boundary crossing may occur during a packet transfer (i.e., a single packet may cross a page boundary.)

The BeginAddrOffset for a GTD is always **CBPOffset**. The value of EndAddrOffset is **CBPOffset** plus **MaxPacketSize** unless it is the last packet defined by the buffer, in which case it will be **BEOffset.**

e) Transfer Description Size Calculation

The maximum data packet size is that of an ITD, 1023 bytes. Therefore the **DataPacketSize** field need only be 10 bits. The **DataPacketSize** is calculated by {1,EndAddrOffset[9:0]} - {0,BeginAddrOffset[9:0]}. The concatenation of the 1 and the 0 to the operation accounts for the case of a page crossing where BeginAddrOffset[9:0] would be larger than EndAddrOffset[9:0].

f) Status Writeback

At the completion of a transaction attempt, the Host Controller performs a status writeback to the Transfer Descriptor. The information written back differs depending on what type of Transfer Descriptor is being serviced.

- General Transfer Description Status Writeback

  General Transfer Descriptors are updated after every attempted transaction that is serviced. There are four fields that require updating after a transaction attempt. They are the **CompletionCode** field, the **DataToggleControl** field, the **CurrentBufferPointer** field, and occasionally the **ErrorCount** field.

  1. DataToggelControl

     The **DataToggleControl** field consists of two bits. When MSB is set it indicates that the DataToggle value is obtained from the ED. Otherwise it is obtained from bit 0.

     The **DataToggleControl** field must be updated to reflect the data toggle for the next transfer. If the packet just transmitted completed successfully, the Host Controller sets the MSb to 1 and toggles the LSb of **DataToggleControl** field to reflect a new value for the next packet. If the current packet did not complete with a proper ACK or NAK, the field should not be changed.

  2. CurrentBufferPointer

     The **CurrentBufferPointer** must be updated if the **CompletionCode** is NoError, DataUnderrun, or DataOverrun. For all other cases the **CurrentBufferPointer** should not be updated because the Host Controller may retry the current packet.

If the **CurrentBufferPointer** requires an update and is not the last packet, the number of bytes transmitted in the packet should be added to the current value of the **CurrentBufferPointer** field. If the packet crossed a page boundary, the upper 20 bits of the **CurrentBufferPointer** should be updated with the **BufferEndPage** field to reflect the change in page base address. The lower 12 bits of the **CurrentBufferPointer** will roll over correctly with a normal addition to reflect the new packet address.

If the **CompletionCode** is NoError and it is the last packet, the **CurrentBufferPointer** field is set to 0h.

3. ErrorCount

   The **ErrorCount** field is incremented each time a transmission error occurs. A transmission error is CRC, BitStuffing, DataToggleMismatch, DeviceNotResponding, and PIDCheckFailure.

   Upon a completion code of NoError the **ErrorCount** field is reset to 0.

   All other responses, including a NAK response from the device, will maintain the current **ErrorCount** value.

   If the **ErrorCount** is 2 and another transmission error occurs, the Transfer Descriptor is retired with the error code reflected in the **CompletionCode** field.

4. CompletionCode

   The **CompletionCode** field of a General Transfer Descriptor is updated after every attempted transaction whether successful or not. The new **CompletionCode** value will be the value provided by the SIE, unless the SIE reports DataUnderrun and the BufferRounding bit is set, in which case the new **CompletionCode** will be NoError.

   When an endpoint returns a NAK handshake for a transmission, all General Transfer Descriptor fields will be the same after the transaction as they were when the transaction began. The Host Controller does not make any changes and no writeback occurs.

- Isochronous Transfer Description Status Writeback

  The Host Controller updates the Offset[R] field after packet transmission using the Packet Status Word. For an OUT packet, the **Size** field is set to 0 if there is no error. For an IN, the **Size** field will reflect the actual number of bytes written to the memory buffer. Regardless of transfer direction, the **CompletionCode** field is updated to reflect the outcome of the transmission.

  The **CompletionCode** field of the ITD is set to NoError unless the ITD is late, in which case it is set to DataOverrun.

- Completion Codes

  Table 18-3 provides the encoding for the Completion Codes used in Transfer Descriptor writebacks and Packet Status Word writebacks.

Table 18.3 Completion Codes

Code	Meaning	Description
0000	NoError	General TD or isochronous data packet processing completed with no detected errors
0001	CRC	Last data packet from endpoint contained a CRC error.
0010	BitStuffing	Last data packet from endpoint contained a bit stuffing violation
0011	DataToggleMismatch	Last packet from endpoint had data toggle PID that did not match the expected value.
0100	Stall	TD was moved to the Done Queue because the endpoint returned a STALL PID
0101	DeviceNotResponding	Device did not respond to token (IN) or did not provide a handshake (OUT)
0110	PIDCheckFailure	Check bits on PID from endpoint failed on data PID (IN) or handshake (OUT)
0111	UnexpectedPID	Receive PID was not valid when encountered or PID value is not defined.
1000	DataOverrun	The amount of data returned by the endpoint exceeded either the size of the maximum data packet allowed from the endpoint (found in **MaximumPacketSize** field of ED) or the remaining buffer size.
1001	DataUnderrun	The endpoint returned less than **MaximumPacketSize** and that amount was not sufficient to fill the specified buffer
1010	reserved	
1011	reserved	
1100	BufferOverrun	During an IN, HC received data from endpoint faster than it could be written to system memory
1101	BufferUnderrun	During an OUT, HC could not retrieve data from system memory fast enough to keep up with data USB data rate.
111x	NotAccessed	This code is set by software before the TD is placed on a list to be processed by the HC.

g) Transfer Description Retirement

When a transfer descriptor is complete (all data sent/received), the **ErrorCount** reaches 3 for a GTD, a fatal error condition occurs for a GTD, or an ITD is late, the Transfer Descriptor must be retired. A fatal error is classified as the new **CompletionCodes** of Stall, DataOverrun, DataUnderrun, and UnexpectedPID.

Several actions are required to retire a Transfer Descriptor.

1. Dequeue the Transfer Descriptor
2. Enqueue the Transfer Descriptor to the Done Queue
3. Update the Endpoint Descriptor

- Dequeue the Transfer Descriptor

  To dequeue the Transfer Descriptor, the Host Controller copies the current Transfer Descriptor **NextTransferDescriptor** field to the **NextTransferDescriptor** of the Endpoint Descriptor.

- Enqueue the Transfer Descriptor

  Following the dequeuing of the Transfer Descriptor from the Endpoint Descriptor Queue, the Transfer Descriptor is enqueued to the Done Queue. To accomplish this, the Host Controller first writes the value of the *HcDoneHead* to the **NextTransferDescriptor** field of the Transfer Descriptor being enqueued. Second, the *HcDoneHead* is written with the address of the Transfer Descriptor being enqueued.

  The next step is to potentially update the Done Queue Interrupt Counter. The **InterruptDelay** field of the Transfer Descriptor specifies the maximum number of SOFs that may occur before the Host Controller writes the HcDoneHead to the HCCA and generates an interrupt. If the value of the **InterruptDelay** field is 111b, the Host Controller Driver does not require an interrupt for the Transfer Descriptor and the Done Queue Interrupt Counter is left unchanged. If the value of the **InterruptDelay** field is not 111b, but is greater than or equal to the current value of the Done Queue Interrupt Counter, the counter is also left unchanged. In this case, another Transfer Descriptor already on the Done Queue requires an interrupt earlier than the Transfer Descriptor being retired. If the value of the **InterruptDelay** field is not 111b, but is less than the current value of the Done Queue Interrupt Counter, the counter is loaded with the value of the **InterruptDelay** field. In this case, the Transfer Descriptor being retired requires an interrupt earlier than all of the Transfer Descriptors currently on the Done Queue.

  If the Transfer Descriptor is being retired with an error, then the Done Queue Interrupt Counter is cleared as if the **InterruptDelay** field were zero.

- Update the Endpoint Descriptor

  In addition, the Host Controller must update the Endpoint Descriptor to reflect the changes to the **NextTransferDescriptor** pointer, the **DataToggleCarry** field, and potentially the **Halt** field. The **Halt** field is only set for a GTD with an error count of 3 or a fatal error condition.

h) Done Queue

Occasionally (as determined by the Done Queue Interrupt Counter = 0), when the Done Queue contains one or more Transfer Descriptors, the Host Controller writes the current value of *HcDoneHead* into the *HccaDoneHead* immediately following a frame boundary and generates an interrupt. These actions are taken so that the Host Controller Driver can complete the processing of retired Transfer Descriptors. After the *HcDoneHead* value is written to the HCCA, the Host Controller resets the value of *HcDoneHead* to '0' and sets the **WritebackDoneHead** bit located in the *HcInterruptStatus* register to '1.' While the **WritebackDoneHead** bit is set, the Host Controller may not write *HcDoneHead* to the HCCA. The **WritebackDoneHead** bit is cleared by the Host Controller Driver when it is ready to receive another Done Queue from the Host Controller.

- Done Queue Interrupt Counter

  The Host Controller maintains a 3-bit counter which is used to determine how often the *HcDoneHead* register value must be written to *HccaDoneHead*. The counter is initialized with a value of 111b at software reset, hardware reset, and when the Host Controller transitions to the USBOPERATIONAL state. The counter functions when the Host Controller is in the USBOPERATIONAL state by decrementing at every frame boundary simultaneous with the incrementing of the **FrameNumber** field in *HcFmNumber* if the current value of the counter is other than 111b or 0. If the current value of the counter is 111b or 0, the counter is effectively disabled and does not decrement.

  The Host Controller checks the value of the counter during the last bit time of every frame when in the USBOPERATIONAL state. If the value of the counter is 0 at that time, the Host Controller checks the current value of the **WritebackDoneHead** bit in *HcInterruptStatus*. If **WritebackDoneHead** is '0,' immediately following the frame boundary, the Host Controller writes the *HcDoneHead* register value to *HccaDoneHead*, sets **WritebackDoneHead** to '1,' and resets the counter to 111b. If **WritebackDoneHead** is '1,' the Host Controller takes no further action until the end of the next frame when it performs the same checks again.

2) Operational Registers

— HcDoneHead

This register contains the address of the first TD on the Done Queue. When the Host Controller retires a TD, it will write the current value of *HcDoneHead* into that TD **NextTD** field. The address of the TD being retired (obtained from a "CurrentED" register) is written to *HcDoneHead*.

3) Description Registers

The TD Block maintains 5 32-bit registers to hold requested Transfer Descriptors. These registers are used as temporary storage for Transfer Descriptors being serviced and are not addressable by software. The same registers are used to hold both Isochronous and General Transfer Descriptors. A brief description of these registers is given below.

a) Dword0

Dword0 contains the status information for the Transfer Descriptor.

- Field

Table 18.4 Dword0 GTD Fields

Bit Range	Field
31 - 28	CompletionCode
27 - 26	ErrorCount
25 - 24	DataToggle
23 - 21	DelayInterrupt
20 - 19	Direction
18	BufferRounding

Table 18.5 Dword0 ITD Fields

Bit Range	Field
31 - 28	CompletionCode
26 - 24	FrameCount
23 - 21	DelayInterrupt
15 - 0	StartingFrame

- Load Condition

  Dword0 is loaded with **pci_Data[31:0]** when **LoadDescriptorDword[0]** is asserted.
- Update Condition

  Dword0 is updated when the TD has been serviced (**fm_TransactionServiced)** and is being written back to memory (**UpdateTD).**

  The updated fields for GTD are CompletionCode, ErrorCount, and DataToggle.

  The updated field for ITD is CompletionCode.

b) Dword1

Dword1 contains the current buffer pointer (CBP) for a GTD, and the buffer page 0 for an ITD.

- Fields

Table 18.6 Dword1 GTD Fields

Bit Range	Field
31 - 12	CBPage
11 - 0	CBPOffset

Table 18.7 Dword1 ITD Fields

Bit Range	Field
31 - 12	BufferPage0 (BP0)

- Load Condition

  Dword1 is loaded with **pci_Data[31:0]** when **LoadDescriptorDword[1]** is asserted.
- Update Condition

  Dword1 is updated with the new CBP only for a GTD when the **fm_TransactionServiced** and **UpdateTD** are asserted, and the completion code is DataOverrun, DataUnderrrun, or NoError.

c) Dword2

Dword2 contains the address of the next TD in the linked list.

- Fields

Table 18.8 Dword2 Fields

Bit Range	Field
31 - 4	NextTD

- Load Condition

  Dword2 is loaded with **pci_Data[31:4]** when **LoadDescriptorDword[2]** is asserted.
- Update Condition

  Dword2 is updated with the current value of the DoneHead pointer when the TD is retired.

d) Dword3

Dword3 contains the value of the last byte of the data packet.

- Fields

Table 18.9 Dword3 GTD Fields

Bit Range	Field
31 - 12	BufferEndPage (BEPage)
11 - 0	BEOffset

Table 18.10 Dword3 ITD Fields

Bit Range	Field
31 - 12	BufferEndPage (BEPage)
11 - 0	BEOffset / Offset1

BEOffset is the buffer end address +1. **Offset1** is used for ITDs under certain circumstances to store begin address of the next packet.

- Load Condition

  There are two load conditions for Dword3. The first condition occurs for every TD that is loaded. Dword3 is loaded when **LoadDescriptorDword[3]** is asserted. **BEPage** is loaded with **pci_Data[31:12]. BEOffset** is loaded with **pci_Data[11:0]** +1.

  The second condition applies only to an ITD. The end address offset +1 for an ITD comes from **Offset**[**RelativeFrame**+1] when **RelativeFrame** is not equal to **FrameCount**. When **RelativeFrame** is odd and not equal to **FrameCount, Offset**[**RelativeFrame**+1] is stored in **Offset1**. The load condition is

  LoadDescriptorDword[5] && RelativeFrame == 1 ||

  LoadDescriptorDword[6] && RelativeFrame == 3 ||

  LoadDescriptorDword[7] && RelativeFrame == 5 ||

- Update Condition

  There is no update condition for Dword3.

e) Offset0

Offset0 is used exclusively for ITDs. It contains the Offset/PSW Dword of the current relative frame.

- Fields

Table 18.11 Offset0 Field Description

Bit Range	Field
31 - 16	Offset0 High
15 - 0	Offset0 Low

If the **RelativeFrame** is even, **Offset[RelativeFrame]** is stored in Offset0 Low and **Offset[RelativeFrame+1]** is stored in Offset0 High.

If the RelativeFrame is odd, Offset[RelativeFrame] is stored in Offset0 High and Offset[RelativeFrame+1] is stored in Offset1.

- Load Condition

  Offset0 is loaded with the Offset/PSW value of the current **RelativeFrame**. The condition is when **LoadDescriptorDword[*n*]** is asserted, where *n* is 4 + (**RelativeFrame** / 2).

- Update Condition

  Offset0 is updated when the TD has been serviced (**fm_TransactionServiced)** and is being written back to memory (**UpdateTD**).

  The update value is the completion code for the packet and the number of bytes received on an IN transer or zero for an OUT transfer.

  Only the Word containing **Offset**[**RelativeFrame**] is updated. The other Word remains unchanged.

Request Block

The system request portion of the List Processor is the top level coordinator of all activity. It arbitrates the master requests, multiplexes the data and control signals to the Host Controller Bus Master.

1) Master Arbitration

   There are three sources of master requests from the List Processor: HCCA Writeback request (HCCAWBReq), ED request (EDReq), and TD request (TDReq).

   The HCCA Writeback has the highest priority. It is granted access to the Bus Master when its request is acknowledged (HCCAWBAck). The request is acknowledged when neither the ED or TD has access to the Bus Master (EDAck or TDAck) and the Bus Master is idle (!bm_active).

   The TD request and ED request are at the lowest priority. The state machine interaction will prevent both requests from being active simultaneously. The individual request (TDReq or EDReq) is granted access to the Bus Master when it is acknowledged (TDAck or EDAck). The request is acknowledged when the HCCA Writeback request is not active and the Bus Master is idle (!bm_active).

2) Data and Control Muxing

   — Master

   The Request Block multiplexes the control and data signals to the Bus Master for the submaster that has the acknowledge.

   There are five control signals provided:

Table 18.12 List Processor Control Signals

Signal	Description
lp_SystemReq	List Processor system request. Asserted for one clock when the submaster is granted its acknowledge.
lp_SystemReqIsData	Transfer is data.
lp_SystemReqIsRead	Transfer is read.
lp_ReqSize	Transfer size in bytes.
lp_BeginAddr	Transfer begin address.

   The data from the submaster that has the acknowledge is multplexed onto the WriteDataOut bus.

   — Slave

   The Request Block multiplexes the slave read data from all of the operational registers in the List Proccessor onto the ReadDataOut bus that goes to the IO module. The mux select comes from the decode signals provided by the IO module.

Frame Management

The frame management block is responsible for managing the frame specific tasks required by the USB specification and the OpenHCI specification. These tasks are:

1. Management of the OpenHCI frame specific Operational Registers
2. Operation of the Largest Data Packet Counter
3. Performing frame qualifications on USB Transaction requests to the SIE
4. Generate SOF token requests to the SIE

Operational Registers

The Host Controller uses four registers to perform the frame timing and status tasks of frame management. These registers are *HcFmInterval*, *HcFmRemaining*, *HcFmNumber*, *HcLSThreshold*, and *HcPeriodicStart*. The function and operation of each register is detailed below.

1) HcFmlnterval Register

   The *HcFmInterval* register contains three fields. These fields are **FrameInterval**, **FrameIntervalToggle**, and **FSLargestDataPacket**.

   The **FrameInterval** field specifies the length of a frame in USB bit times and is programmed by the HCD. **FrameInterval** and **FrameIntervalToggle** are loaded into the **FrameRemaining** counter and **FrameRemainingToggle**, respectively, at the beginning of each frame. When programming **FrameInterval**, the **FrameIntervalToggle** field is toggled by the HCD each time a new interval is specific. The HCD uses the **FrameIntervalToggle** and **FrameRemainingToggle** fields to see when the Host Controller has started using a newly programmed **FrameInterval** value (see discussion below).

   The **FSLargestDataPacket** field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. (See next section for a discussion of this function) This field is programmed by the HCD.

2) HcFmRemaining Register

   The *HcFmRemaining* register contains two fields, **FrameRemaining** and **FrameRemainingToggle**.

   The **FrameRemaining** field is a 14-bit counter. When operating, the counter value decrements once per USB bit time. The **FrameRemaining** counter is valid only when the Host Controller is in the USBOPERATIONAL state. When **FrameRemaining** reaches a value of 0, it is loaded with the value of **FrameInterval** at the next rising edge of the 12 MHz reference clock. In addition, the counter will load with **FrameInterval** when the Host Controller transitions into the USBOPERATIONAL state to re-initialize the frame period.

The **FrameRemainingToggle** field (bit) is loaded with **FrameIntervalToggle** coinciding with **FrameRemaining** being loaded with **FrameInterval**. **FrameRemainingToggle** and **FrameIntervalToggle** are used by the HCD to determine which value of FrameInterval is being used in the current frame. If the toggle fields match when read, the Host Controller loaded the current value of **FrameInterval** into **FrameRemaining** at the beginning of the current frame (current frame is using a newly programmed value). If the toggle fields do not match, the Host Controller loaded the previous value of **FrameInterval** into **FrameRemaining** at the beginning of the current frame (current frame is not using a newly programmed value). This information is provided so that the HCD may determine exactly which frame is using a newly programmed value of **FrameInterval.**

3) HcFmNumber Register

   The *HcFmNumber* register contains only the **FrameNumber** field.

   FrameNumber is a 16 bit counter. **FrameNumber** is incremented by the Host Controller at each frame boundary. Specifically, **FrameNumber** is incremented when **FrameRemaining** is loaded with **FrameInterval**. The lower 11 bits of the frame number are used in the SOF token data field. Each time the frame number is incremented the new value is written back to the HCCA.

4) HcPeriodicStart Register

   The *HcPeriodicStart* register contains only the **PeriodicStart** field.

   PeriodicStart is a 14 bit programmable value which determines when the Host Controller must start processing the periodic list. When processing lists while in the USBOPERATIONAL state, the Host Controller will start processing the periodic list in a given frame when **FrameRemaining** is less than **PeriodicStart** (or if no other list is enabled).

5) HcLSThreshold Register

   The HcLSThreshold register contains only the **LSThreshold** field.

   LSThreshold is a 12 bit programmable value which provides a bit time threshold in a frame beyond which low speed transactions are prohibited. When the Host Controller has a low speed transaction to send, **FrameRemaining** is compared to **LSThreshold**. The low speed transaction may be started if and only if **FrameRemaining** is greater than or equal to **LSThreshold.**

Packet Size Check

When a TD is loaded, the packet size must be guaranteed to complete within the remaining frame time. The sections below discuss FS and LS transaction frame requirements.

1) Full speed Check

   The full speed packet size check is accomplished by comparing the value of a data transactions data packet size (in bits) with the current value of the Largest Data Packet Counter. The data packet size is provided by the List Processor and in the form of the **MaximumPacketSize** field of the Endpoint Descriptor. If the data packet size is greater than the current value of the Largest Data Packet Counter the transaction may not be started. If the data packet size is less than or equal to the current value of the Largest Data Packet Counter the transaction may be started.

   — Largest Data Packet Counter Operation

     The Largest Data Packet Counter is used to determine whether or not a full speed transaction with a data packet of a given size may be completed in the current frame. The value accounts for bit stuffing requirements and overhead of the token, data, and handshake packets.

     At each frame boundary, coincident with the loading of **FrameRemaining**, the Largest Data Packet Counter is loaded with the value of the **FSLargestDataPacket** field of the *HcFmInterval* register. The counter value is decremented 6 of every 7 bit times (too simulate the bit stuffing)

2) Low Speed Check

   The low speed Packet Size Check is accomplished by comparing the current value of **FrameRemaining** with the value of **LSThreshold**. If **FrameRemaining** is less than **LSThreshold**, the low speed transaction cannot be started.

Transaction Requests

Transaction requests to the SIE are initiated by either the List Processor or the Frame Management block. All of the requests are delivered to the SIE via the Frame Management block. The List Processor requests must be qualified with the Packet Size Check. This check happens continually from the time the Frame Management block receives the request from the List Processor until a request acknowledge is received from the SIE indicating the start of the transaction. If the Packet Check fails prior to the SIE acknowledge, a Packet Check failure signal is sent to the List Processor by the Frame Management block.

In addition to qualifying List Processor transaction requests, the Frame Management block will issue the SOF Token request. This request is issued when the value of **FrameRemaining** is 0001h. The early request is to account for the SIE output delay between the request and a change on the bus such that the SOP of the SOF token coincides with the reloading of **FrameInterval** in **FrameRemaining**

Interrupt Processing

Interrupts are the communication method for HC-initiated communication with the Host Controller Driver. There are several events which may trigger an interrupt from the Host Controller. Each specific event sets a specific bit in the *HcInterruptStatus* register. The Host Controller requests an interrupt when all three of the following conditions are met:

- The **MasterInterruptEnable** bit in *HcControl* is set to '1'
- A bit in *HcInterruptStatus* is set to '1'
- The corresponding enable bit in *HcInterruptEnable* for the *HcInterruptStatus* bit is set to '1'.

If the Host Controller supports an SMI pin, the interrupts caused by most events are routable, based on the value of the **InterruptRouting** bit of the *HcControl* register, to either the INT pin or the SMI pin. Enabled interrupt events causes an interrupt to be signaled on the INT pin when the **InterruptRouting** bit is a '0' and signaled on the SMI pin if the **InterruptRouting** bit is a '1.' However, OpenHCI Host Controllers are not required to implement an SMI pin. If a Host Controller does not implement an SMI pin and the **InterruptRouting** bit is a '1,' interrupts are not generated. The notable exception for interrupt routing is the **OwnershipChange** event which is always routed to the SMI pin.

Each of the following subsections describes a specific event, and therefore a specific bit, represented in the *HcInterruptStatus* registe.

SchedulingOverrun Event

When a scheduling overrun occurs, the Host Controller sets the **SchedulingOverrun** bit following the completion of the next **HccaFrameNumber** update. A scheduling overrun occurs when the Host Controller determines that the Periodic list for the current frame cannot be completed before the end of the frame.

WritebackDoneHead Event

Periodically, the Host Controller is required to update **HccaDoneHead** with the value of the *HcDoneHead* register (see Section 17.2.2.2.3.1.8). When the write of *HcDoneHead* to **HccaDoneHead** completes, the Host Controller sets the **WritebackDoneHead** bit. The corresponding interrupt (if enabled) will inform the Host Controller Driver that it must service the Done Queue.

StartOfFrame Event

When **FrameRemaining** is loaded with **FrameInterval, t**he Host Controller sets the **StartOfFrame** bit following completion of the next **HccaFrameNumber** update. This corresponds to a frame boundary. The Host Controller Driver will normally disable this event, enabling the event when it requires a deterministic interrupt at a frame boundary.

ResumeDetected Event

A resume detected event occurs when the Root Hub detects resume signaling on the USB bus. The Host Controller will set the **ResumeDetected** bit when resume signaling is detected.

A **ResumeDetected** interrupt is only possible in the USBSUSPEND state. A resume event can be either an upstream resume signal or a connect/disconnect detection at a port. The connect/ disconnect resume event is enabled by the **RemoteWakeupEnable** in the *HcRhStatus* register. If a port is either in the progress of selectively resuming or has completed the selective resume and set **PortSuspendStatusChange** when the Root Hub enters the USBSUSPEND state, the port resume is cleared and the hub resume, **ResumeDetected**, is generated.

The **ResumeDetected** event may occur while the system clock is disabled. The interrupt processor will generate an interrupt for an enabled **ResumeDetected** event. The interrupt will remain active until the clock is enabled and the **ResumeDetected** bit in the interrupt status register is cleared.

UnrecoverableError Event

This implementation does not support generation of an Unrecoverable Error event.

FrameNumberOverflow Event

When the MSB (bit 15) of the **FrameNumber** field of *HcFmNumber* changes value, the **FrameNumberOverflow** bit is set by the Host Controller following the next **HccaFrameNumber** update. The event occurs on both the ‘1’ to ‘0’ or the ‘0’ to ‘1’ transition. This event allows the Host Controller Driver to perform any necessary manipulation of its software based frame number to ensure that number is correct.

RootHubStatusChange Event

The Host Controller sets the **RootHubStatusChange** bit whenever there is a change to any bit in *HcRhStatus* or *HcRhPortStatus*. Any changes in these registers define a change in status that must be communicated to the Host Controller Driver. Since OpenHCI provides a register-level interface to the Root Hub, the need for Root Hub Transfer Descriptors is eliminated. This provides for a more efficient Root Hub interface, but does not provide the Host Controller Driver a good mechanism for polling the Root Hub on a periodic basis. To compensate for the lack of a good polling mechanism, the Host Controller delivers an interrupt on every Root Hub status change.

OwnershipChange Event

The **OwnershipChange** bit is set by the Host Controller when the Host Controller Driver sets the **OwnershipChangeRequest** bit in the *HcCommandStatus* register. This ensures that an interrupt is generated (unless it is masked) whenever ownership of the Host Controller is passed to and from the operating system Host Controller Driver and any SMM-based Host Controller Driver in the system. All interrupts resulting from an OwnershipChange event are not routable with the **InterruptRouting** bit of the *HcControl* register and are delivered on the SMI pin only. If the Host Controller does not implement an SMI pin, interrupts will not be generated at all on an OwnershipChange event.

Host Controller Bus Master

The Host Controller Bus Master is the central block in the data path. It coordinates all access to the PCI Interface.

There are two sources of bus mastering within USB Host Controller: the List Processor and the Data Buffer Engine. The List Processor has control when it transfers Endpoint Descriptors, Transfer Descriptors, and HCCA information. The List Processor gives control to the Data Buffer Engine when the data pointed to by a Transfer Descriptor is to be transferred.

The Host Controller Bus Master consists of three main modules: The Bus Master Controller, the Data Buffer Engine, and the Page Crossing Controller.

Bus Master Controller

The Bus Master Controller provides the PCI Interface (PCI IF) with the signals necessary to perform PCI cycles on behalf of the Data Buffer Engine or List Processor. These signals are the address of the first byte in the transfer, the size of the transfer in bytes, the direction of the transfer, and the data for write cycles. The PCI IF will carry out the appropriate PCI cycle based on the information provided.

In addition there are three control signals across the interface: a request signal from the Bus Master Controller, and a cycle active and a data ready signal from the PCI Controller. The Bus Master Controller requests access when it needs to transfer a cycle on the PCI Bus. The PCI IF arbitrates internally to the ASIC for access to the PCI bus. (For USB Host Controller the arbitration is immediate since it has no other PCI masters). When access is granted, the PCI IF asserts the cycle active signal which remains asserted until the cycle completes. The PCI IF asserts the data ready signal for one PCI period for each Dword transfer across the interface. On a write, the data ready signal indicates that the PCI IF has consumed the Dword presented by the Bus Master Controller. On a read, the data ready signal indicates that a valid Dword is present on the PCI read data bus.

The Bus Master Controller maintains the address of the next PCI cycle to be requested. The address is initialized with a value from the List Processor when it initiates a request. The address is updated after each burst transfer completes. This is needed for data transfers that require more than a single burst cycle.

Data Buffer Engine

The Data Buffer Engine controls the bus master requests for transferring data to and from the Data Buffer. The Data Buffer Engine monitors the status of the Data Buffer by using the SIE pointer and the PCI pointer. These pointers indicate the next location to be read from or written to by the SIE data and the PCI data respectively. When the Data Buffer Engine determines that there is space available in the Data Buffer, it makes a request to the Page Crossing Controller that a bus master cycle be initiated.

The Data Buffer is divided into either two 32-byte or four 16-byte regions, depending on the value of the **DataBufferRegion16** bit in the PCI configuration register *OperationalMode*. Whenever the PCI pointer is not in the same region as the SIE pointer, there is sufficient room in the Data Buffer to transfer at least one region worth of data. Once a transfer request has been issued, the size of the transfer will not change, even if more data becomes available in the buffer before the PCI Controller services the request.

If the region size is 32, then there are 32 bytes available in the buffer while a PCI Master cycle is in progress. The maximum PCI latency that can be tolerated without losing Isochronous data is:

32 bytes avail * 8 clocks/byte * 83.33 ns/clock = 21,333 ns

If the region size is 16, then there can be between 16 to 48 bytes available in the buffer for SIE data while a PCI Master cycle is in progress. Initially there are three regions available for SIE data while one region is being transferred on PCI. If the SIE transfers data in the three regions before the PCI transfer completes, then only one region is available for SIE data during the next PCI transfer. The maximum PCI latency that can be tolerated without losing Isochronous data ranges from:

16 bytes avail * 8 clocks/byte * 83.33 ns/clock = 10,666 ns
32 bytes avail * 8 clocks/byte * 83.33 ns/clock = 21,333 ns
48 bytes avail * 8 clocks/byte * 83.33 ns/clock = 32,000 ns

The region size of 16 is advantageous for a Most Recently Used type PCI arbitration scheme where the initial grant may require up to 32,000 ns but the subsequent grant occurs within 10,666 ns

Page Crossing Controller

The Page Crossing Controller is responsible for controlling the page address of data transfers. If it receives a request from the Data Buffer Engine that crosses a page boundary, it will break up the request into two separate requests to the Bus Master Controller. The first request will contain the data from the first page, the second request will contain the data from the second page.

The Page Crossing Controller is also responsible for indicating to the Bus Master Controller to load the page address of the second page into the upper 20 bits of the address register. This occurs whenever the address increments to the next page, either due to a page crossing or the current request ends at the end of a page.

Data Buffer (DB)

The Data Buffer serves as the data interface between the PCI Controller and the SIE. It is a combination of a 64-byte latched based bi-directional asynchronous FIFO and a single Dword PCI Holding Register.

The PCI Holding Register is a 32-bit edge-triggered register that is used for two purposes: First, to serve as a data pipeline stage so that the PCI control signal TRDY# can be synchronized before going into the write enable logic of the latches in the Data Buffer. Second, to serve as a holding register for the last Dword of an OUT transfer that won? fit into the FIFO. This is required for General Transfer Descriptors since the entire data packet, up to 64 bytes, must be read entirely into the buffer before the data request begins. Depending on the begin address of the packet, there could be up to 17 Dwords to be stored.

The Data Buffer maintains the number of bytes transferred for the current data packet. This value is used by the List Processor to update the status of the Transfer Descriptor.

The number of bytes transferred is compared with the expected number of bytes to produce a last byte flag when the values are equal. The SIE uses the last byte flag for two purposes: First, to determine if the current IN packet is a data overrun or data underrun. Second, to indicate that there are no more bytes for the current OUT packet. The Data Buffer also uses the last byte flag to inhibit any data beyond the expected size for an IN packet from being stored in the buffer

18.2.3 USB Interface

The USB interface includes the integrated Root Hub with two external ports, Port 1 and Port 2 as well as 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.

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.

SIE Control

Table 18-13 lists all other control information which is required to generate and manage the requested transaction. Most of the information is extracted from the ED and TD in the List Processor.

Table 18.13 Transaction Control Information

Information	Description
Direction[1:0]	The data packet direction and PID must be defined as IN, OUT, SETUP.
Format	The transaction must be identified as isochronous to remove the handshake phase from the bus protocol.
Data Toggle State	The data toggle state is required for selecting PID DATA0 or DATA1 and data toggle comparison. **(ToggleCarry/DataToggle)**
Speed	The endpoint speed is required to select the bus data rate.
EndpointNumber[3:0]	The token endpoint data.
FunctionAddress[7:0]	The token address data.
HcFmNumber[10:0]	The SOF token data field for device frame synchronization.

Packet Control

The sections below describe the packet types and formats generated by the SIE.

1) Sync Pattern

   The first byte of every packet is a synchronization pattern that allows the data receiver to synchronize to the transmitters data rate prior to any meaningful data. The pattern 80h translates into an NRZI encoding of 54h on the bus. The byte? ending sequence of KK signifies the start of the PID byte.

2) End of Packet (EOP)

Every packet is terminated by an end of packet (EOP) consisting of 2 transmitted single-ended zeros (SE0) followed by a driven idle state for another bit time.

3) Packet Identifier (PID)

Each packet contains a packet identifier (PID) declaring the packet function. The PID is transmitted as a 4-bit encoding followed by the bitwise inversion of the PID as an error check. If the PID check fails, the packet is ignored since without the a valid PID the packet cannot be identified. Table 18-14 shows the PID encoding and definitions for each packet type.

Table 18.14 PID Encoding

Packet Type	Function	PID[3:0]	Description
Token	OUT	0001	Host to Device transaction request
	IN	1001	Device to Host transaction request
	SOF	0101	Start of Frame timing mark
	SETUP	1101	Host to Device Control transaction setup
Data	DATA0	0011	Even data packet
	DATA1	1011	Odd data packet
Handshake	ACK	0010	Receiver accepts the data packet
	NAK	1010	Receiver rejects the data packet or transmitter cannot send data packet
	STALL	1110	Endpoint is stalled.
Special	PRE	1100	Preamble for low speed driver enable in hubs

4) Token Packet

There are four types of token packets described below: IN, OUT, SETUP, Start of Frame (SOF). IN and OUT specify data packet direction in the data phase of the transaction. SETUP has the same meaning as an OUT token to the SIE but has special meaning to the device decoding it. A SOF is used to synchronize all USB devices to a nominal 1 ms bus interval.

An IN, OUT, or SETUP token packet contains an 11-bit data field containing the device and endpoint address. For an SOF token, this same 11-bit field contains the lower 11 bits of **FrameNumber** in the *HcFmNumber* register. The data field is followed by a 5-bit CRC. The root hub is the only device which can send a token packet so it is not necessary to decode a token packet. All fields are serially transmitted LSb first (except the CRC).

Packet Byte	Bit 7							Bit 0
0	Sync Pattern[7:0]							
1	~PID[3:0]				PID[3:0]			
2	Endpoint [0]	Address[7:0]						
3	~CRC[0:4]					Endpoint[3:1]		
4	High-Z					Idle	SE0	

Figure 18.8 Standard Token Packet Format

Packet Byte	Bit 7							Bit 0
0	Sync Pattern[7:0]							
1	~PID[3:0]				PID[3:0]			
2	**FrameNumber**[7:0]							
3	~CRC[0:4]					**FrameNumber**[10:8]		
4	High-Z					Idle	SE0	

Figure 18.9 SOF Token Packet Format

5) Data Packet

The data packet contains the actual data transfer between the host and endpoint specified in the token packet. DATA0 and DATA1 identify the packet as a data packet. The 0/1 tag on the PID provides a mechanism for data synchronization from one data packet to the next to the same endpoint.

The data packet includes the 4 bit PID, 4 bit inverted PID, 0-1023 data bytes, and a 16 bit CRC. All fields are serially transmitted LSB and LSb first (except CRC). The number of data bytes for an OUT is determined by the HC. The IN data byte count is tracked by the Data Buffer Engine. The number of data bytes is irrelevant to the operation of the SIE.

Packet Byte	Bit 7							Bit 0
0	Sync Pattern[7:0]							
1	~PID[3:0]				PID[3:0]			
2	Data0[7:0]							
...	...							
N+2	DataN[7:0]							
N+3	~CRC[8:15]							
N+4	~CRC[0:7]							
N+5	High-Z					Idle	SE0	

Figure 18.10 Data Packet Format

6) Handshake Packet

The handshake packet is used to close the bus transaction and report completion status. An ACK handshake concludes a successful transaction. The STALL handshake is returned when the device is unable to handle data due to an internal error. NAK is used when the endpoint has no data to send or does want any data. The host can only send an ACK handshake to the device, but the device can return any handshake. The handshake packet includes only the 4 bit PID and 4 bit inverted PID.

Packet Byte	Bit 7							Bit 0
0	Sync Pattern[7:0]							
1	~PID[3:0]				PID[3:0]			
2	High-Z					Idle	SE0	

Figure 18.11 Handshake Packet Format

7) Preamble Packet

The preamble packet is used to inform downstream hubs that a low speed packet is coming. This allows the low speed drivers to be enabled by all ports connected to low speed devices. This is repeated for all packets within the transaction and not just preceding the token packet. This procedure applies only to packets from the host and not packets received from the device. The preamble packet, like the handshake packet, only includes the 4 bit PID and 4 bit inverted PID, except that it is not terminated by an EOP. Instead, the bus is driven idle for 4 FS clock, then the data rate changes to 1.5 MHz and the sync pattern of the LS packet is begun immediately. The LS packets are otherwise as described in the previous sections.

Packet Byte	Bit 7							Bit 0
0	Sync Pattern[7:0]							
1	~PID[3:0]				PID[3:0]			
2	Change to LS data Rate and begin LS token, data, or handshake packet as defined in the above sections.				Idle			

Figure 18.12 Preamble Packet Format

Serializer

The Serializer performs the parallel/serial conversion for both transmitting and receiving. The logic consists of a packet control state machine, 8-bit parallel-load/serial-shift register, 8-bit data latch, PID encoder/decoder, CRC generator/checker, bit stuff counter, NRZI encoder, and data receiver circuit. Data paths of transmitted and received paths share logic since the bus only operates one direction at a time and the path's logical organizations are simply reversed.

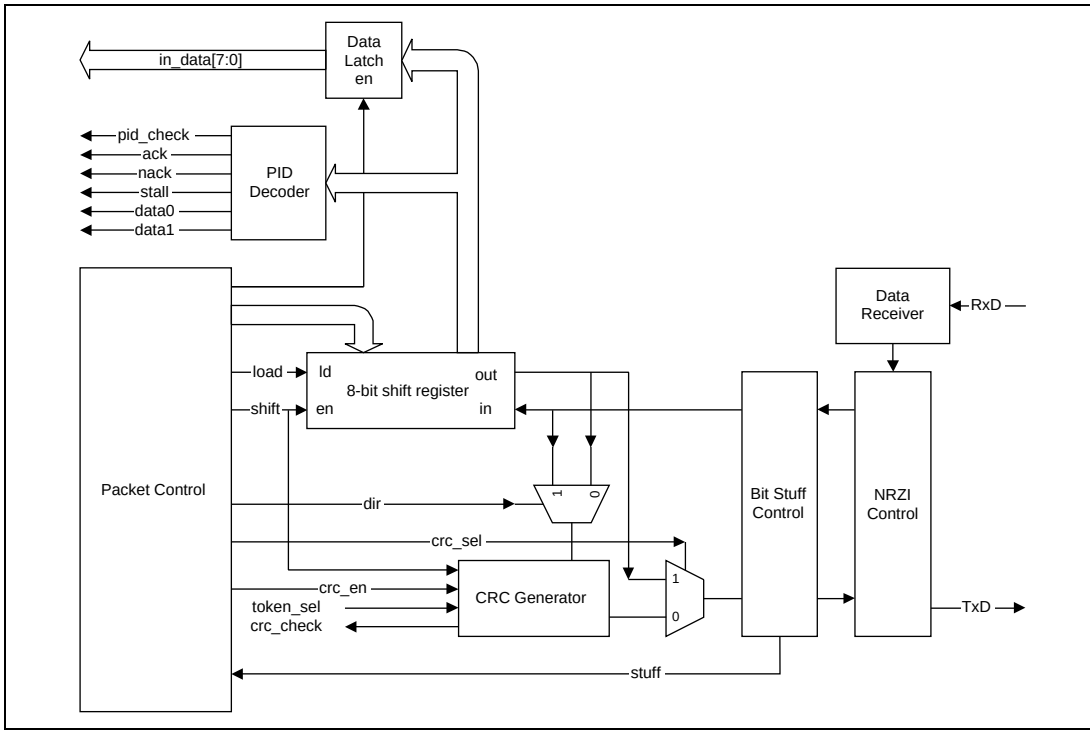


Figure 18.13 Serializer

1) Data

Data fields of the token packets are provided by the List Processor and Frame Manager. For a transmitted data packet, the Data Buffer Engine provides the data. Each time a byte is loaded into the shifter the buffer pointer is incremented. The data buffer marks the data byte enabling the CRC remainder into the data stream.

If the data buffer experiences an underrun condition such that the data packet cannot be completed, the Serializer will immediately force a bit stuff error followed by the EOP.

For a received data packet, the data is provided to the data buffer after each byte. A latch holds each byte while a new byte is collected. The latch is necessary to allow synchronization be the PCI clock domain of the data buffer and the USB clock domain of the receiver. When the EOP is detected the last two bytes sent to the data buffer are invalidated as the two CRC bytes. These bytes can only be identified after the fact.

2) CRC Generator/Checker

The CRC is calculated using a generator polynomial applied to a data pattern. The data pattern includes all data fields excluding sync patterns, PIDs, and data bytes sans the NRZI encoding. Below is a description of the CRC procedure. See also Figure 18-14.

— At the start of the packet, preload 1 into the CRC remainder registers (5 bits for token, 16 bits for data).

— The next data bit is XORed with the MSb of the remainder and left shifted (to MSb) filling with 0.

— If the XOR result was 1 the generator polynomial is XORed with the remainder during the shift. (The LSb of the remainder is the fill 0).

At the end of the data pattern the remainder is inverted and transmitted at the tail of the packet, MSb first.

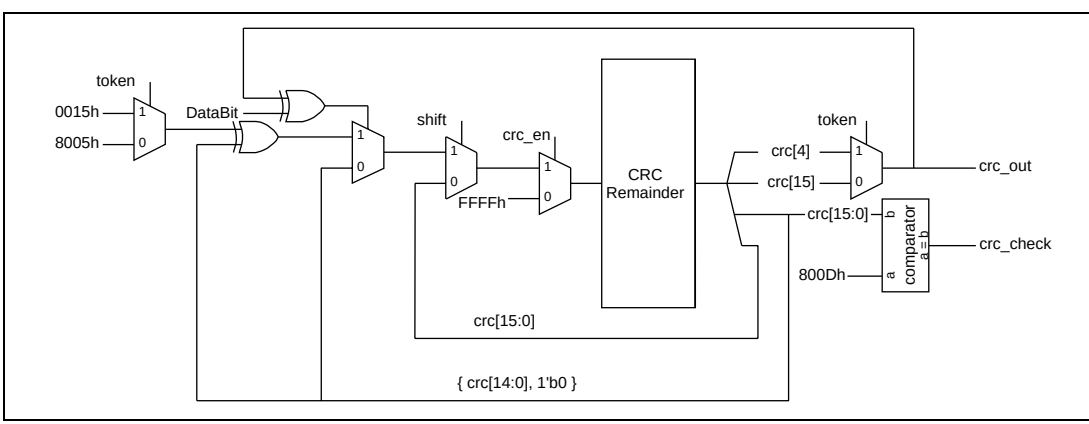


Figure 18.14 CRC Logic

1. Token - All data following the PIDs are included in the CRC generation. Equation 18-1 is the polynomial used to generate the token packet CRC:

$$G(X) = X^5 + X^2 + 1 = 10101b$$

Equation 18.1 Token Packet CRC Generator Polynomial

After the token data is finished the source of the transmitted bit stream is switched to the CRC (MSb first). The byte counter completes the last full byte of the token packet with the inverted 5-bit remainder of the CRC generator.

The checker polynomial (Equation 18-2) is provided only as a reference since the root device never receives a token packet and never checks against it.

$$G(X) = X^3 + X^2 = 01100b$$

Equation 18.2 Token Packet CRC Checker Polynomial

2. Data - For transmitting, all data following the PIDs are included in the CRC generation. Equation 18-3 is the polynomial used to generate the data packet CRC.

$$G(X) = X^{16} + X^{15} + X^2 + 1 = 10005h$$

Equation 18.3 Data Packet CRC Generator Polynomial

The inverted 16-bit remainder follows the data bytes of a data packet (MSb first), even for a 0 byte data packet.

For receiving, all bits after the PID until the EOP must be processed through the CRC checker (including the CRC bytes since they cannot be identified until after EOP). The checker is only valid for data packets, i.e. the packet PID is DATA0 or DATA1. After processing the data and the CRC from the packet, the remainder is compared with the checker polynomial (see below). If equal the data was correctly received. Otherwise, a CRC error is present and the packet is discarded.

Until the EOP is detected it is impossible to separate the functional data bytes from the two CRC bytes. This means that all bits until the EOP must be processed through the checker. The minimum number of bytes through the checker is two bytes (16 bit CRC code for a 0 data byte packet). After the second full byte and for each following full byte, the CRC remainder is checked against the checker polynomial. If equal and the EOP is received before the next full byte of bus data, the packet is successfully received. If the EOP is detected and the last CRC check failed, the packet is unsuccessful and the EOP is treated as a false EOP. At the EOP the last two bytes are identified as the CRC code and must be invalidated in the data buffer. Equation 18-4 shows the data packet checker polynomial.

$$G(X) = X^{15} + X^3 + X^2 + 1 = 800Dh$$

Equation 18.4 Data Packet CRC Checker Polynomial

3) Bit Stuffing

A data rate phase lock requires frequent bus transitions to maintain the lock before the drift potential threaten to corrupt the data stream. To guarantee transitions occur before the data rate drifts to far to maintain the phase lock, bit stuffing is employed to force transitions after 6 bit times without a transition. For every 6 consecutive 1's in the data pattern (non-NRZI encoded) a 0 is inserted to force a bus transition. This 0 must be removed from the pattern by the receiving device. The bit stuffing policy is enforced between the SOP and EOP. If an incoming data stream violates the bit stuff policy by containing more that 6 consecutive 1's, the packet is invalidated.

To guarantee that a bit stuff detected during the EOP does not invalidate the packet, the bit stuff error is updated after each valid byte like the CRC check.

A forced bit stuff error for a buffer underrun or EOF violation will enable the stuffing mechanism for 16 bit times to guarantee the an error and no response.

4) Data Encoding / Decoding (NRZI)

All data on the USB must be NRZI encoded/decoded. A logical 0 is represented by a change in the differential level while a logical 1 is represented by no change in the signal level. This is done after the bit stuff is added for transmitting and before the bit stuff is removed for receiving.

5) Receiver

The receiver is responsible for identifying and managing an incoming data stream. When a port identifies an SOP, the receiver establishes connectivity to that port until an EOP is detected or EOF. It maintains the data rate phase lock and extracts the each data bit.

— SOP

The SOP is detected by a J to K bus state transition at any enabled port. When the SOP is detected, the port is connected to the phase lock and data bit extraction circuit. The receiver only detects an SOP when the SIE is expecting a response packet from a device. Any other activity is ignored.

— Phase Lock

The receiver must establish a data rate synchronization with the data source in order to extract the packet data. This is accomplished by locking the internal data rate to each K to J bus state transition in the data stream. By using a 8x oversampling clock (48 MHz) the data period is divided into eight phases utilizing both edges of the oversample clock. Each transition detected synchronizes the SIE's data rate clock to the external data rate. The USB jitter allowance permits tracking a single edge transition while reliably extracting all data bits in between. LS phase locking requires some debouncing of the data transitions due to the slow transition times. For the phase lock jitter timing analysis.

When the first K to K NRZI-encoded bit pattern is detected, it identifies the last bit of the sync pattern and the packet PID begins at the next bit.

When a valid EOP is detected receiver lock is released and the data rate clock resynchronizes to the static 12 MHz SOF clock.

The phase lock function is required regardless of the data rate, FS or LS, though timing requirements differ.

— EOP

A valid EOP is identified by the receiver by sampling a SE0 for three consecutive 48 MHz clock periods following by a J-state. This allows for an approximate minimum SE0 pulse width of 40 ns. A valid EOP is detected independent of the data rate phase lock due to its switching characteristics. The trailing J-state is also detected independent of the phase lock. A valid EOP should occur within the byte boundary following the last byte of the packet. In other words the data stream is captured and validated on byte boundaries until the EOP detected. If the CRC check failed at the last byte boundary, the EOP is considered false and the receiver connection is maintained until a valid EOP or 16 bit times of J-state is detected.

1) Non-Isochronous Transactions

Figure 18-15 shows the valid packet sequencing of a non-isochronous transfer.

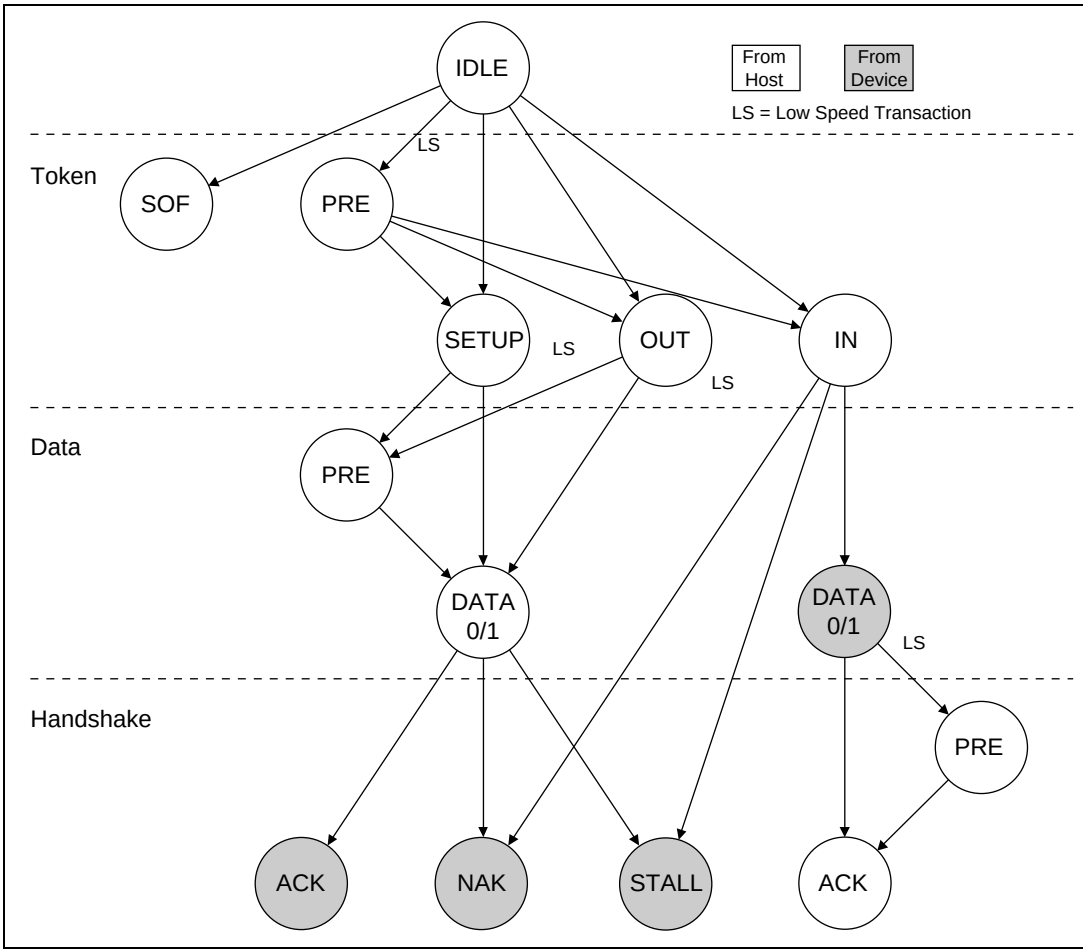


Figure 18.15 Non-Isochronous Bus Transaction

2) Isochronous Transfers

Figure 18-16 shows the valid packet sequence of an isochronous transfer. An isochronous transfer is characterized by the lack of a handshake phase after a data phase. Note that a handshake is still permitted following the token phase, if a handshake condition is present which supersedes the data phase.

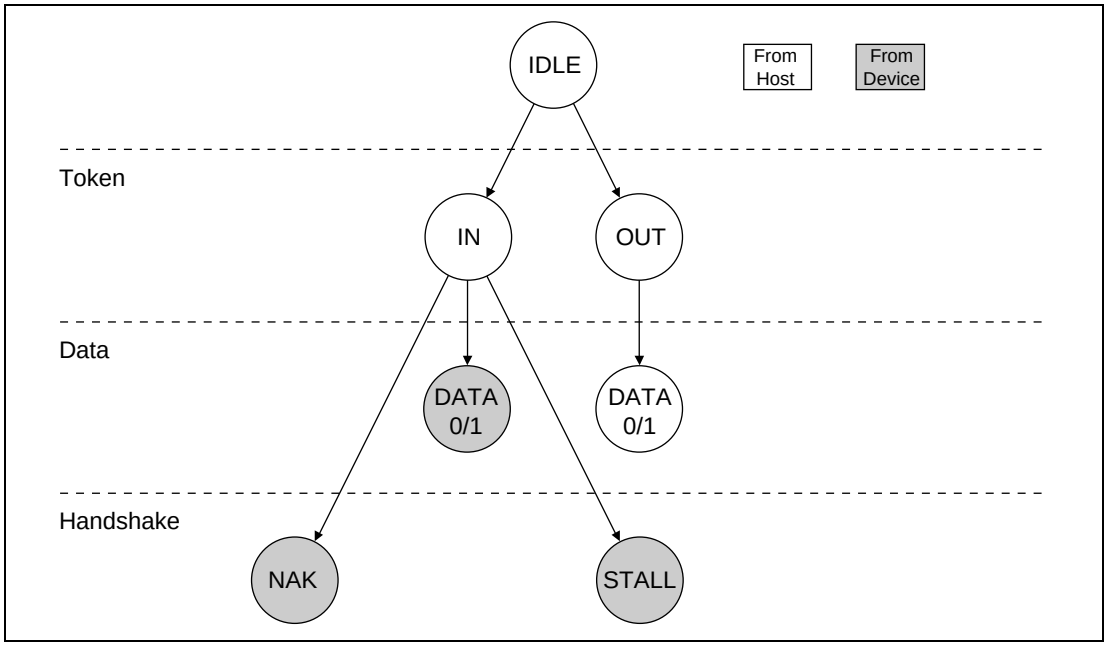


Figure 18.16 Isochronous Bus Transaction

3) Packet to Packet Timing

Table 18-15 summarizes the packet to packet timing parameters. These times do not account for receiver synchronization delays that may postpone the detection of bus events.

Table 18.15 Bus Time-out Periods

Parameter	Bit Times
Packet to Packet Delay	2
Response Delay	7.5
Response Time-out	16
Packet Error Time-out	16

Note: All bit times are relative to the device speed.

a) Back to Back Host Packets

During back to back host packets of the same transaction (e.g. token to data packet), a minimum of 2 idle bit times are required from the end of the EOP's SE0 to the SOP of the next packet. The bus is driven during this period,

b) Response Turnaround

After an IN data packet, the handshake packet is initiated within 1 bit time of the EOP detection. The EOP detection delay is TBD. The only valid handshake from the host is an ACK. If an error is detected, the host does not respond with any handshake.

c) Host Response Time-out

When waiting for a device to respond following a token or data packet, the transaction times out after 16 bit times plus the delay of the SOP detection delay. The SOP detection delay is TBD. A new transaction may begin after a total of 18 bit times. This is not expected to be achievable due to latencies of the transaction's retirement and retrieval of the next transaction.

d) Packet Error Time-out

When an incoming packet is rejected due to errors and the device expects a response, the next transaction cannot begin until the previous transaction has timed-out at the device. The minimum host delay from the end of the EOP's SE0 to the next SOP is a total of 18 bit times.

4) Frame Timing (SOF, EOF)

The frame is controlled by the Frame Manager which issues SOF requests each frame period. The SIE is further guaranteed to be idle when this request is generated by EOF timing marks which allow any bus activity that threatens to interfere with SOF request to be terminated. Table 18-16 summarizes the minimum USB EOF requirements and the value of *HcFmRemaining* at which they occur.

Table 18.16 SIE EOF Timing Requirements

Parameter	*HcFmRemaining*	Description
EOF FS Start	47	Latest start time of a FS packet
EOF LS Start	184	Latest start time of a LS packet
EOF Rx Disable	41	Invalidate Rx packet beyond this time.

If software improperly sets the **FSLargestDataPacket** field of the *HcFmInterval* register or *HcLSThreshold*, the transmitter may overrun the frame and miss the SOF event. As a hardware fail-safe, the SIE terminates any packet such packet with a 16 bit time J state to generate a bit stuff error and prevent a response.

5) Data Toggle Synchronization

The data toggle PIDs, DATA0 and DATA1, are the final check to ensure the device received a valid ACK handshake. The data packet receiver is responsible for maintaining data toggle bit synchronization. Therefore, the HC is required to correct the IN data toggle synchronization only, while the device corrects OUT data synchronization.

The data toggle state of any data packet is provided by the List Processor from the Endpoint and Transfer Descriptors. On an outgoing data packet, when an ACK is received the List Processor updates the data toggle state. For an incoming data packet, the SIE compares the PID with the data toggle state from the List Processor. If they match, the data is accepted, an ACK is returned, and the List Processor updates the data toggle state. If they do not match, the data is rejected and an ACK is returned (assuming no other errors), but the List Processor does not update the data toggle.

Isochronous transfers do not perform a data toggle state check and accept any Data0/1 PID.

6) False EOP

If an EOP is detected and the data stream resulted in a CRC error at the last full byte of data, the EOP is labeled a false EOP. An EOP is also considered false if it failed to reach the minimum packet size (through 2 CRC bytes for data packets or the PID for handshake packets):

When a false EOP is detected, the packet connectivity remains until another EOP is detected. If after a false EOP no transitions occur for 16 bit times and the bus is in the idle state, the transaction is terminated and the next transaction may begin after another 2 bit times. Note that this is similar to a packet time-out after a packet error.

7) Packet Error

If an error is detected while receiving a packet, the transaction is retired with errors. When a bit stuff violation, PID check failure or undefined PID, or a CRC error is detected, the SIE waits 18 bit times after the EOP to allow the transaction to time-out at the device before another transaction is initiated.

An invalid PID is a PID, which is undefined by USB or is inappropriate within the predefined protocol of transaction PID packet sequences. A packet with an invalid PID is ignored.

8) Internal Buffer Errors

For general TDs the maximum packet size cannot exceed the internal 64 byte buffer. This guarantees an uninterrupted data transfer. For isochronous transfers the maximum packet size can be up to 1023. This results in the possibility of having an internal over/underrun condition in the buffer which interrupts the data transfer. On an OUT data packet, if an underrun occurs the packet is terminated with a 16 bit time J state resulting in a bit stuff violation. For an IN data packet, an overrun results in discarded data, but does not affect the processing of the bit stream. The packet is not acknowledge and will be retried, hopefully with better results.

9) Logical Buffer Errors

A logical buffer error occurs on an IN data packet when the expected number of bytes received does not match the expected number of bytes. If no other error is detected, the packet is acknowledged and the data accepted. The over or underrun condition is reported to the HCD.

10) Transaction Completion Status

When a transaction completes, the completion status is reported to the List Processor. The SIE can compile all error information to create the **CompletionCode** field for the TD writeback, but may be overridden by the List Processor. Table 18-17 shows the transaction status (nearly identical to the **CompletionCode**) definitions generated by the SIE to the List Processor. Table 18-18 and Table 18-19 show the prioritization of transaction status used by the SIE to create the encoding.

Table 18.17 SIE Completion Status

Completion Code	Definition	Description
0000	NoError	ACK handshake for general TD, Successful data packet transfer for isochronous TD.
0001	CRC	CRC Error detected in the received data packet.
0010	BitStuffing	Bit Stuff violation occurred in received packet.
0011	DataToggleMismatch	Data toggle state of the PID did not match in received data packet.
0100	STALL	STALL handshake received. Additionally, a NAK handshake following an isochronous IN token.
0101	DeviceNotResponding	Endpoint did not respond the token request and timed-out.
0110	PIDCheckFailure	The PID check bits did not match in received packet.
0111	UnexpectedPID	Received a PID which did not match the packet size or protocol or is not defined.
1000	DataOverrun	Number of data bytes received exceeded the expected.
1001	DataUnderrun	Number of data bytes received is less than expected
1010	Reserved	
1011	Reserved	
1100	BufferOverrun	Internal data buffer is full resulting in lost IN data.
1101	BufferUnderrun	Internal data buffer is empty resulting in bad OUT data.
1110	NotAccessed	Transaction aborted due to EOF.
1111	NotAccessed	NAK handshake received.

Table 18.18 IN Transaction Error Response

IN Transaction / Fdata Phase

EOF Violation	Packet Time-out	Packet Error	Buffer Overrun	Data Toggle Mismatch	NAK/ STALL	Data Overrun/ Underrun	Status Code	Action
yes	-	-	-	-	-	-	1110	Retire transaction.
no	yes	-	-	-	-	-	0101	Retire transaction.
no	-	yes	-	-	-	-	0001 0010 0110 0111	Retire transaction.
no	-	no	yes	-	-	-	1100	Retire transaction.
No	-	no	no	yes	-	-	0011	Retire transaction. Return ACK handshake.
no	-	no	-	-	yes	-	0100 1111	Retire transaction.
no	-	no	no	no	-	yes	1000 1001	Retire transaction. Return ACK handshake..
no	no	no	no	no	no	no	0000	Retire transaction. Return ACK handshake.

Table 18.19 OUT Transaction Error Response

OUT Transaction

Transaction Phase	EOF Violation	Packet Time-out	Packet Error	STALL/ NAK	Buffer Underrun	Status Code	Action
Data Phase	yes	-	-	-	-	1110	Retire transaction.
	no	-	-	-	yes	1101	Force Bit Stuff Error. Wait for Handshake response time-out.
Handshake Phase	yes	-	-	-	-	111x	Retire transaction.
	no	yes	-	-	-	0101	Retire transaction.
	no	no	yes	-	-	0001 0010 0110 0111	Retire transaction.
	no	no	no	yes	-	0100 1111	Retire transaction.
	no	no	no	no	-	0000	Retire transaction.

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

Table 18.20 Hub / Port Commands

Command	Register	Description
SetGlobalPower	*HcRhStatus*	Enable Power to all Ports connected to the global switch.
ClearGlobalPower	*HcRhStatus*	Disable Power to all Ports connected to the global switch
SetPortPower	*HcRhPortStatus*	Enable Power to Port
ClearPortPower	*HcRhPortStatus*	Disable Power to Port
SetPortEnable	*HcRhPortStatus*	Enable Port
ClearPortEnable	*HcRhPortStatus*	Disable Port
SetPortSuspend	*HcRhPortStatus*	Selectively Suspend Port
ClearPortSuspend	*HcRhPortStatus*	Selectively Resume Port
SetPortReset	*HcRhPortStatus*	Enable Port Reset Signaling

Hub Control

The HC states defined in HC Register Summary section also reflect the hub state. For example, when the HC is suspended, USBSUSPEND, the Root Hub is suspended. When the HC is in USBRESUME, the hub generates the appropriate bus signaling. USBRESET resets the Root Hub. The following sections describe hub and bus related controls and status.

1) Power Switching

   The hub controls power to the downstream ports. USB Host Controller implement global power control, which means that all port power status, **PortPowerStatus**, is controlled by **Set**/**ClearGlobalPower** commands. Individual port power switching as well as no power switching modes are also implemented. Table 18-21 shows the power switching configurations in the descriptor registers. The global power switch is connected to PWREN.

Table 18.21 Power Switching Configurations

Mode	NoPowerSwitching	PowerSwitchingMode	PortPowerControlMask
No Switching	1	-	-
Global Switching	0	0	-
Individual Port Switching	0	1	1
Global Switch in Individual Switching Mode	0	1	0

2) Over-Current Protection

   Over-Current is reported on a global basis. Input pin OVRCUR is read directly through **OverCurrentIndicator** in *HcRhStatus*. Over-current mode is configured in *HcRhDescriptorA* by **NoOverCurrentProtection** and **OverCurrentProtectionMode**. When in individual over-current mode, OVRCUR status is reported through the **PortOverCurrentIndicator** in *HcRhPortStatus* for test purposes.

   When an over-current condition occurs **OverCurrentIndicatorChange** is set and power is disabled. If port power is not switched (**NoPowerSwitch**), the port power status is not affected by an over-current condition.

3) Reset

   When the HC is in the USBRESET state, the entire USB interface is reset. This includes the Root Hub and SIE logic. The only logic not affected by this reset is the Root Hub Descriptor registers which are configured by the system and not the HCD and, therefore, are reset only by the hardware reset (PCIRST#).

4) Suspend

   When the HC is in the USBSUSPEND state, all control and status is unaffected. The HC will cease packet requests which effectively removes all bus activity through the hub and ports. USBSUSPEND also disables the USB 12 MHz clock.

5) Resume

When the HC is in USBRESUME, the resume signal, K-state, is broadcast to all enabled ports, but not selectively suspended ports. Signaling is maintained until the HC leaves USBRESUME. A trailing LS EOP is used to terminate the resume. After the Resume is removed, bus activity returns to normal.

6) Wakeup Events

If the HC is in USBSUSPEND, status changes in the hub or ports create wakeup conditions for the Root Hub and HC. A remote wakeup generates a **ResumeDetected** interrupt and a transition to the USBRESUME state. A remote wakeup is defined as follows.

— A K-state(resume) at an enabled port.

— A port connect/disconnect detection and **DeviceRemoteWakeupEnable** is set.

— A port resume is in progress at a selectively suspended port.

When the Root Hub forces a transition to USBRESUME state, the resulting bus activity is the same as that described in HC Register Summary section.

7) Low Speed EOP

The Root Hub must generate LS EOPs during normal operation with each SOF and after exiting USBRESUME to USBOPERATIONAL. The EOP is 16 FS bit times of SE0 and 8 FS bit times of the J-state.

The LS keep-alive EOP is initiated in parallel to the SOF transaction when the Frame Manager generates an SOF request. The pulse is guaranteed to complete prior to any attempted access to the LS device.

The resume LS EOP prevents any other bus activity from the SIE.

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.

1) Connect/Disconnect

Ports are responsible for detecting when devices are attached and disconnected. A port must have power enabled before any bus state evaluation may occur. Thus, a port is always disconnected following a reset (not a port reset). An unconnected port is characterized by a SE0 bus state. A FS connection is characterized by a differential '1'. A LS connection is characterized by a differential '0'.

CurrentConnectStatus reports the connection state of the port. When the port is not driving the bus, a connect event is detected by non-SE0 state on the bus for 30 bit times (2.5 us). A disconnect event is a SE0 on the bus for 30 bit times. Each time an event is detected and **CurrentConnectStatus** changes, **ConnectStatusChange** is set. **LowSpeedDeviceAttached** is set if the connect state is LS when a connect event is detected. When a LS device is attached, the transmitted and received data bit stream is inverted (J=0,K=1). This allows all data to be processed through the SIE at the FS state definition.

If **DeviceRemoveable** is set for the port, which means not removable (yes, it is backwards), **CurrentConnectStatus** is always read a '1'. After power is enabled to the port or a hardware reset when no power switching is implemented, a connect event will be detected through the previously defined method resulting in **ConnectStatusChange** being set and **LowSpeedDeviceAttached** being configured. A disconnect should never be detected by nature and does not need to be specifically prevented.

If the port is not connected when one of the following commands is received, the command is ignored and **ConnectStatusChange** is set. This causes software to re-evaluate **CurrentConnectStatus**. These commands are not defined for a port which is not connected.

— SetPortEnable

— ClearPortEnable

— SetPortSuspend

— ClearPortSuspend

— SetPortReset

When a port is disconnected **PortEnabledStatus** and **PortSuspendStatus** are be cleared.

2) Reset

Individual ports may be reset by a **SetPortReset** command. The port sets **PortResetStatus** and generates a 10 ms reset (SE0). At the conclusion of the reset pulse, the port is enabled and **PortResetStatusChange** is set. **PortSuspendStatus** or **PortSuspendStatusChange** are also cleared by a **SetPortReset** command.

3) Enabled/Disabled

Once a device is connected, port behavior is defined by **PortEnableStatus**. The ports are individually enabled and disabled by the host. The port can never enable itself, but can be disabled when a port is active at EOF. While disabled the port does not propagate any upstream or downstream traffic. All upstream activity is ignored, except for a disconnect. While enabled the port propagates all Resume signaling, upstream packets, and downstream packets. If a LS device is connected downstream traffic is only enabled following a PRE (preamble) token.

4) Suspend/Resume

The port can be selectively suspended by issuing a **SetPortSuspend** command which sets **PortSuspendStatus**. While suspended the port does not propagate any downstream traffic. The port can be awakened by any of the following methods.

— A ClearPortSuspend command

— An upstream J to K transition

— A J to SE0 transition which results in a disconnect event (no resume)

When the port is resumed, the port drives the resume signal downstream for 20 ms followed by a LS EOP. The port waits an additional 3 ms during which it propagates all traffic and then sets **PortSuspendStatusChange** and clears **PortSuspendStatus**.

A disconnect event at a selectively suspended port does not generate a resume pulse. It sets **ConnectStatusChange** and clears **PortSuspendStatus**.

If the hub is suspended, i.e. USBSUSPEND, while the port is also suspended, the port does not respond to a USBRESUME transition. However, an upstream resume at the port forces a USBRESUME transition.

Interrupts

A Root Hub interrupt, **RootHubStatusChange**, is generated by setting any of the **XXXXChange** fields in the *HcRhStatus* and *HcRhPortStatus* registers. Writing the change bit to '1' clears the source of the interrupt.

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 USBSUSPEND 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. For more information on the data receiver see HC Register Summary section.

18.2.4 Power Management

At this time, USB Host Controller supports minimal system level power management features. Each system has its own requirements which makes it impossible to satisfy. The only power management feature implemented is the disabling of the USB clock generator in USBSUSPEND state. Additional power management features will require slight modifications.

The design supports PCICLK frequencies from 0 to 33 MHz. Synchronization between the PCI and USB clock domains is frequency independent.

Remote wakeup of USB is asynchronously implemented from the Ports to the PCI INTA#.

The design currently requires CLK48 to be operational at all times. If it is necessary to stop the 48 MHz clock the system design will require that the signal used to enable and disable the USB clock generators also be used to wake the 48 MHz clock source. Currently, the **RemoteWakeupConnected** and **RemoteWakeupEnable** bits in the *HcControl* register are not implemented. These bits may be implemented in conjunction with such an implementation to wake the clocking. Also, the OpenHCI 1.0 spec. description is incorrect in stating that this signal is associated with the **ResumeDetected** interrupt. It can be but is not limited to **ResumeDetected**, and is intended to be defined by the system wakeup requirements above OpenHCI (i.e. system event trigger, if necessary).

18.2.5 Register/Address Summary

OpenHCI Registers

Although USB Host Controller has a PCI interface, however the configuration cycle is not needed to perform on it. From the point of view of CPU, the access of these registers is transparent just like the memory access. The base address of HC registers is **0x1000B000.**

HC Register Summary

Table 18.22 HC Operational Register Summary

Offset	Register
00-03	*HcRevision*
04-07	*HcControl*
08-0B	*HcCommandStatus*
0C-0F	*HcInterruptStatus*
10-13	*HcInterruptEnable*
14-17	*HcInterruptDisable*
18-1B	*HcHCCA*
1C-1F	*HcPeriodCurrentED*
20-23	*HcControlHeadED*
24-27	*HcControlCurrentED*
28-2B	*HcBulkHeadED*
2C-2F	*HcBulkCurrentED*
30-33	*HcDoneHead*
34-37	*HcFmInterval*
38-3B	*HcFrameRemaining*
3C-3F	***HcFmNumber***
40-43	***HcPeriodicStart***
44-47	***HcLSThreshold***
48-4B	***HcRhDescriptorA***
4C-4F	***HcRhDescriptorB***
50-53	***HcRhStatus***
54-57	*HcRhPortStatus [1]*
58-5C	*HcRhPortStatus [2]*

Table 18.23 HcRevision Register

Register: *HcRevision*			**Offset: 00-03**
Bits	**Reset**	**R/W**	**Description**
31 - 8	0h	-	Reserved. Read/Write 0's
7 - 0	10h	R	**Revision** Indicates the OpenHCI Specification revision number implemented by the Hardware. (X.Y = XYh) USB Host Controller supports the 1.0 specification.

Table 18.24 HcControl Register

Register: *HcControl*			Offset: 04-07
Bits	**Reset**	**R/W**	**Description**
31 - 11	0h	-	Reserved. Read/Write 0's
10	0b	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	0b	R	**RemoteWakeupConnected** This bit indicated whether the HC supports a remote wakeup signal. This implementation does not support any such signal. The bit is hard-coded to '0.'
8	0b	R/W	**InterruptRouting** This bit is used for interrupt routing: 0: Interrupts routed to normal interrupt mechanism (INT). 1: Interrupts routed to SMI.
7 - 6	00b	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 USBSUSPEND to USBRESUME after detecting resume signaling from a downstream port.
5	0b	R/W	**BulkListEnable** When set this bit enables processing of the Bulk list.
4	0b	R/W	**ControlListEnable** When set this bit enables processing of the Control list.
3	0b	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.

Table 18.24 HcControl Register (cont'd)

Register: *HcControl*			Offset: 04-07
Bits	**Reset**	**R/W**	**Description**
2	0b	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	00b	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' = 3 Control Endpoints)

HccommandStatus

Table 18.25 HcCommandStatus Register

Register: HcCommandStatus			Offset: 08-0B
Bits	**Reset**	**R/W**	**Description**
31 - 18	0h	-	Reserved. Read/Write 0's
17 - 16	00b		**ScheduleOverrunCount** This field increments every time the **SchedulingOverrun** bit in HcInterruptStatus is set. The count wraps from '11' to '00.'
15 - 4	0h	-	Reserved. Read/Write 0's
3	0b	R/W	**OwnershipChangeRequest** When set by software, this bit sets the **OwnershipChange** field in HcInterruptStatus. The bit is cleared by software.
2	0b	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	0b	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	0b	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

All bits are set by hardware and cleared by software.

Table 18.26 HcInterruptStatus Register

Register: HcInterruptStatus			**Offset: 0C-0F**
Bits	**Reset**	**R/W**	**Description**
31	0h	-	Reserved. Read/Write 0's
30	0b	R/W	**OwnershipChange** This bit is set when the **OwnershipChangeRequest** bit of HcCommandStatus is set.
29 - 7	0h	-	Reserved. Read/Write 0's
6	0b	R/W	**RootHubStatusChange** This bit is set when the content of HcRhStatus or the content of any HcRhPortStatus register has changed.
5	0b	R/W	**FrameNumberOverflow** This bit is set when bit 15 of **FrameNumber** changes value from ‘0’ to ‘1’ or from ‘1’ to ‘0.’
4	0b	R	**UnrecoverableError** This event is not implemented and is hard-coded to ‘0.’ All writes are ignored.
3	0b	R/W	**ResumeDetected** This bit is set when the Host Controller detects resume signaling on a downstream port.
2	0b	R/W	**StartOfFrame** This bit is set when the Frame Management block signals a ?tart of Frame’ event.
1	0b	R/W	**WritebackDoneHead** This bit is set after the Host Controller has written HcDoneHead to HccaDoneHead.
0	0b	R/W	**SchedulingOverrun** This bit is set when the List Processor determines a Schedule Overrun has occurred.

HcInterruptEnable

Writing a ‘1’ to a bit in this register sets the corresponding bit, while writing a ‘0’ to a bit leaves the bit unchanged.

Table 18.27 HcInterrutpEnable Register

Register: HcInterruptEnable			**Offset: 10-13**
Bits	**Reset**	**R/W**	**Description**
31	0b	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 above.
30	0b	R/W	**OwnershipChangeEnable** 0: Ignore 1: Enable interrupt generation due to Ownership Change.
29 - 7	0h	-	Reserved. Read/Write 0's
6	0b	R/W	**RootHubStatusChangeEnable** 0: Ignore 1: Enable interrupt generation due to Root Hub Status Change.
5	0b	R/W	**FrameNumberOverflowEnable** 0: Ignore 1: Enable interrupt generation due to Frame Number Overflow.
4	0b	R/W	**UnrecoverableErrorEnable** This event is not implemented. All writes to this bit will be ignored.
3	0b	R/W	**ResumeDetectedEnable** 0: Ignore 1: Enable interrupt generation due to Resume Detected.
2	0b	R/W	**StartOfFrameEnable** 0: Ignore 1: Enable interrupt generation due to Start of Frame.
1	0b	R/W	**WritebackDoneHeadEnable** 0: Ignore 1: Enable interrupt generation due to Writeback Done Head.
0	0b	R/W	**SchedulingOverrunEnable** 0: Ignore 1: Enable interrupt generation due to Scheduling Overrun.

HcInteruptDisable

Writing a ‘1’ to a bit in this register clears the corresponding bit, while writing a ‘0’ to a bit leaves the bit unchanged.

Table 18.28 HcInterruptDisable Register

Register: HcInterruptDisable			Offset: 14-17
Bits	**Reset**	**R/W**	**Description**
31	0b	R/W	**MasterInterruptEnable** This bit is a global interrupt disable. A write of ‘1’ disables all interrupts.
30	0b	R/W	**OwnershipChangeEnable** 0: Ignore 1: Disable interrupt generation due to Ownership Change.
29 - 7	0h	-	Reserved. Read/Write 0's
6	0b	R/W	**RootHubStatusChangeEnable** 0: Ignore 1: Disable interrupt generation due to Root Hub Status Change.
5	0b	R/W	**FrameNumberOverflowEnable** 0: Ignore 1: Disable interrupt generation due to Frame Number Overflow.
4	0b	R/W	**UnrecoverableErrorEnable** This event is not implemented. All writes to this bit will be ignored.
3	0b	R/W	**ResumeDetectedEnable** 0: Ignore 1: Disable interrupt generation due to Resume Detected.
2	0b	R/W	**StartOfFrameEnable** 0: Ignore 1: Disable interrupt generation due to Start of Frame.
1	0b	R/W	**WritebackDoneHeadEnable** 0: Ignore 1: Disable interrupt generation due to Writeback Done Head.
0	0b	R/W	**SchedulingOverrunEnable** 0: Ignore 1: Disable interrupt generation due to Scheduling Overrun.

HcHCCA

Table 18.29 HcHCCA Register

Register: *HcHCCA*			Offset: 18-1B
Bits	**Reset**	**R/W**	**Description**
31 - 8	0h	R/W	HCCA Pointer to HCCA base address. (Within SRAM memory space)
7 - 0	0h	-	Reserved. Read/Write 0's

HcPeriodCurrntED

Table 18.30 HcPeriodCurrentED Register

Register: HcPeriodCurrentED			Offset: 1C-1F
Bits	**Reset**	**R/W**	**Description**
31 - 4	0h	R/W	**PeriodCurrentED** Pointer to the current Periodic List ED. (Within SRAM memory space)
3 - 0	0h	-	Reserved. Read/Write 0's

HcControlHeadED

Table 18.31 HcControlHeadED

Register: HcControlHeadED			Offset: 20-23
Bits	**Reset**	**R/W**	**Description**
31 - 4	0h	R/W	**ControlHeadED** Pointer to the Control List Head ED. (Within SRAM memory space)
3 - 0	0h	-	Reserved. Read/Write 0's

HcControlCurrentED

Table 18.32 HcControlCurrentED Register

Register: HcControlCurrentED			Offset: 24-27
Bits	**Reset**	**R/W**	**Description**
31 - 4	0h	R/W	**ControlCurrentED** Pointer to the current Control List ED. (Within SRAM memory space)
3 - 0	0h	-	Reserved. Read/Write 0's

HcBulkHeadED

Table 18.33 HcBulkHeadED Register

Register: HcBulkHeadED			Offset: 28-2B
Bits	**Reset**	**R/W**	**Description**
31 - 4	0h	R/W	**BulkHeadED** Pointer to the Bulk List Head ED. (Within SRAM memory space)
3 - 0	0h	-	Reserved. Read/Write 0's

HcBulkCurrentED

Table 18.34 HcBulkCurrentED Register

Register: HcBulkCurrentED			Offset: 2C-2F
Bits	**Reset**	**R/W**	**Description**
31 - 4	0h	R/W	**BulkCurrentED** Pointer to the current Bulk List ED. (Within SRAM memory space)
3 - 0	0h	-	Reserved. Read/Write 0's

HcDoneHead

Table 18.35 HcDoneHead Register

Register: *HcDoneHead*			**Offset: 30-33**
Bits	**Reset**	**R/W**	**Description**
31 - 4	0h	R/W	**DoneHead** Pointer to the current Done List Head ED. (Within SRAM memory space)
3 - 0	0h	-	Reserved. Read/Write 0's

HcFmInterval

Table 18.36 HcFmInterval Register

Register: HcFmInterval			**Offset: 34-37**
Bits	**Reset**	**R/W**	**Description**
31			**FrameIntervalToggle** This bit is toggled by HCD whenever it loads a new value into **FrameInterval**.
30 - 16			**FSLargestDataPacket** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame.
15 - 14	0h	-	Reserved. Read/Write 0's
13 - 0	2EDFh	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

Table 18.37 HcFrameRemaining Register

Register: HcFrameRemaining			**Offset:** 38-3B
Bits	**Reset**	**R/W**	**Description**
31	0b	R	**FrameRemainingToggle** This bit is loaded with **FrameIntervalToggle** when **FrameRemaining** is loaded.
30 - 14	0h	-	Reserved. Read/Write 0's
13 - 0	0b	R	**FrameRemaining** This field is a 14 bit decrementing counter used to time a frame. When the Host Controller is in the USBOPERATIONAL 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 USBOPERATIONAL.

HcFmNumber

Table 18.38 HcFmNumberb Register

Register: ***HcFmNumber***			**Offset: 3C-3F**
Bits	**Reset**	**R/W**	**Description**
31 - 16	0h	-	Reserved. Read/Write 0's
15 - 0	0b	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 ?FFFh' to '0h.'

HcPeriodicStart

Table 18.39 HcPeriodicStart Register

Register: HcPeriodicStart			**Offset: 40-43**
Bits	**Reset**	**R/W**	**Description**
31 - 14	0h	-	Reserved. Read/Write 0's
13 - 0	0b	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.

Table 18.40 HcLSThreshold Register

Register: HcLSThreshold			Offset: 44-47
Bits	**Reset**	**R/W**	**Description**
31 - 12	0h	-	Reserved. Read/Write 0's
11 - 0	0b	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

This register is only reset by a power-on reset (PCIRST#). It is written during system initialization to configure the Root Hub. These bit should not be written during normal operation.

Table 18.41 HcRhDescriptorA Register

Register: HcRhDescriptorA			Offset: 48-4B
Bits	**Reset**	**R/W**	**Description**
31 - 24	01h	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.
23 - 13	0h	-	Reserved. Read/Write 0's
12	0	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	0	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 - 0	02h	R	**NumberDownstreamPorts** USB Host Controller supports two downstream ports.

HcRhDescriptorB

This register is only rest by a power-on reset (PCIRST#). It is written during system initialization to configure the Root Hub. These bit should not be written during normal operation.

Table 18.42 HcRhDescriptorB Register

Register: HcRhDescriptorB			**Offset: 4C-4F**
Bits	**Reset**	**R/W**	**Description**
31 - 16	0000h	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 = Global power switching 1 = Individual port switching Port Bit relationship 16 : Reserved 17 : Port 1 18 : Port 2 ... 31 : Port 15 Unimplemented ports are reserved, read/write '0'.
15 - 0	0000h	R/W	**DeviceRemoveable** 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

This register is reset by the USBRESET state.

Table 18.43 HcRhStatus Register

Register: *HcRhStatus*			**Offset: 50-53**
Bits	**Reset**	**R/W**	**Description**
31	0	W	(write) **ClearRemoteWakeupEnable** Writing a '1' to this bit clears **DeviceRemoteWakeupEnable**. Writing a '1' has no effect.
30 - 18	0h	-	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 - 2	0h	-	Reserved. Read/Write 0's
1	-	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 '0'. (write) **ClearGlobalPower** Writing a '1' issues a **ClearGlobalPower** command to the ports. Writing a '0' has no effect.

HcRhPortStatus[1:2]

This register is reset by the USBRESET state.

Table 18.44 HcRhPortStatus Register

Register: HcRhPortStatus[1:2]			Offset: 54-57,58-5B
Bits	**Reset**	**R/W**	**Description**
31 - 21	0h	-	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.
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 Device Removeable is set, this bit is set only after a Root Hub reset to inform the system that the device is attached.
15 - 10	0h	-	Reserved. Read/Write 0's
9	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

Register: HcRhPortStatus[1:2] **Offset: 54-57,58-5B**

Bits	Reset	R/W	Description
8	0	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 - 5	0h	-	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.
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) **ClearPortSuspend** 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.

Section 19 A/D Converter

19.1 Overview

The controller includes a 10-bit successive-approximations A/D converter with a wide selection of up to four analog input channels.

19.1.1 Features

- A/D converter features are listed below:
- 10-bit resolution
- Four input channels
- High-speed conversion
- Conversion time: maximum 10μs per channel (with 1MHz peripheral clock)
- Two conversion modes are supported.
- Single mode: one channel A/D conversion
- Scan mode: continuos conversions are operated in cycles from one to four channels
- Four 16-bit data registers
- A/D conversion results are transferred for storage into data registers in correspondences to the selected channels.
- Sample-and- hold function
- A/D conversion can be triggered by internal timer 0
- A/D interrupt requested at the end of the conversion
  At the end of A/D conversion, an A/D end interrupt (ADI) can be requested
- Provides STANDBY mode to reduce Power Consumption

19.1.2 Block Diagram

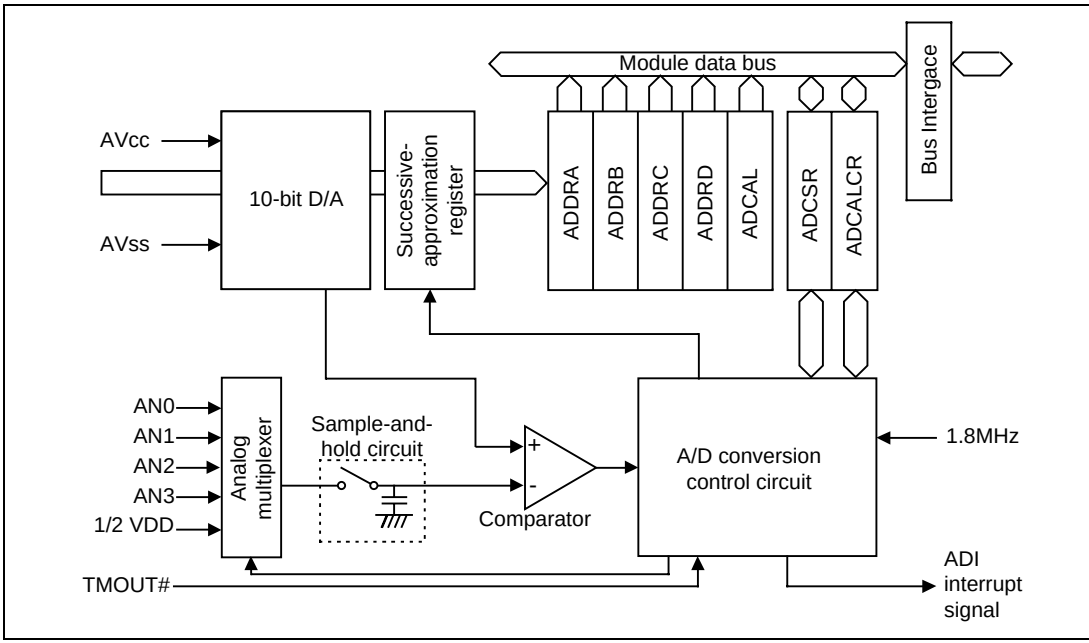


Figure 19.1 A/D Converter Block Diagram

19.1.3 Input Pins

Four analog input pins from AN0 to AN3 are provided. AVcc and AVss are the power supply pins and ground pin respectively for the analog circuits in the A/D converter.

Table 19.1 A/D Converter Pins

Pin Name	Abbreviation	I/O	Function
Analog power supply pin	AVcc	I	Analog power supply
Analog ground pin	AVss	I	Analog ground and reference voltage
Analog input pin 0	AN0 (TSMX)	I	Analog inputs
Analog input pin 1	AN1 (TSMY)	I	
Analog input pin 2	AN2 (TSPX)	I	
Analog input pin 3	AN3 (TSPY)	I	

19.1.4 Register Configuration

Table 19.2 A/D Converter Registers

Address	Name	Abbreviation	R/W	Initial Value
H'1000E000	A/D data register A	ADDRA	R	H’0000
H'1000E002	A/D data register B	ADDRB	R	H’0000
H'1000E004	A/D data register C	ADDRC	R	H’0000
H'1000E006	A/D data register D	ADDRD	R	H’0000
H'1000E008	A/D control/status register	ADCSR	R/W	H’0000
H'1000E00A	A/D calibration sample control	ADCALCR	W	H’0000
H'1000E00C	A/D calibration data register	ADCAL	R	H’0000

19.2 Register Descriptions

19.2.1 A/D Data Registers A to D (ADDRA to ADDRD, ADCAL)

The five A/D data registers (ADDRA to ADDRD, ADCAL) are 16-bit read-only registers that store the results of A/D conversion.

An A/D conversion produces 10-bit data, which is transferred for storage into the A/D data register in correspondence to the selected channel. The lower 8 bits of the result are stored in the lower byte of the A/D data register. The upper 2 bits are stored in the upper byte. Bits 15 to 10 of an A/D data register are reserved bits that are always read 0. Table 19-3 below indicates the pairings of the analog input channels and A/D data registers.

The CPU can always read the A/D data registers.

The A/D data registers are initialized to H'0000 by a reset.

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ADDRn (upper byte)	0	0	0	0	0	0	AD9	AD8
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	R	R

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADDRn (lower byte)	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Initial Value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

(n = A to D)

Table 19.3 Analog Input Channels and A/D Data Registers

Analog Input Channel (Value)	**A/D Data Register**
AN0	ADDRAX
AN1	ADDRBX
AN2	ADDRCX
AN3	ADDRDX
1/2VDD (1/2 x 3.3 Volt)	ADCAL

19.2.2 A/D Control/Status Register (ADCSR)

ADCSR is a 16-bit readable/writable register that selects the mode and controls the A/D converter. ADCSR is initialized to H'0000 by a reset and is in STANDBY mode.

Bit	7	6	5	4	3	2	1	0
Bit Name	ADF*	ADST	ADIS*	TRGE	ADIE	SCAN	CH1	CH0
Initial Value	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

*NOTE: Only 0 can be written to clear the flag bits.

Bit	Description	Default
7	**A/D End Flag (ADF):** Indicates the end of A/D conversion. When set to 1: In Single mode, A/D conversion ends. In Scan mode, A/D conversion ends in all selected channels. When set to 0: Cleared by reading ADF while ADF = 1, then writing 0 in ADF.	0
6	**A/D Start (ADST):** Starts or stops A/D conversion. The ADST bit value remains at 1 during A/D conversion. It can also be set to 1 by external trigger input at the ADTRG pin. When set to 1: In Single mode, A/D conversion starts; ADST is automatically cleared to 0 when conversion ends. In Scan mode, A/D conversion starts and continues, cycling among the selected channels, until ADST is cleared to 0 by software, or by a reset, or by switching to STANDBY mode.	0
5	**Interrupt Status (ADIS):** Indicates the status of A/D interrupt request. When the bit is 1, interrupt request is active. When the bit is 0, interrupt request is inactive. If software writes 0 to this bit, it can disable the interrupt request.	0
4	**Trigger Enable (TRGE):** Enables or disable external triggering of A/D conversion. When set to 1, A/D conversion starts at the falling edge of the external trigger signal (ADTRG). When set to 0, A/D conversion cannot be externally triggered.	0
3	**A/D Interrupt Enable (ADIE):** Enables or disables the interrupt (ADI) requested at the end of an A/D conversion. When this bit is set to 1, A/D end interrupt request (ADI) is enabled. When this bit is set to 0, A/D end interrupt request (ADI) is disabled.	0
2	**Scan Mode (SCAN):** Selects single mode or scan mode. For further information on the operation of these modes, see section **19.3 Operation** on page 331. Clear the ADST bit to 0 before switching the conversion mode. When this bit is set to 1 is Scan mode. Set this bit to 0 to enter Single mode.	0
1 - 0	**Channel Select 1 and 0 (CH1 and CH0):** These bits and the SCAN bit select the analog input channels. Clear the ADST bit to 0 before changing the selected channel.	0

19.2.3 A/D Calibration Sample Control Register (ADCALCR)

ADCALCR is a 16-bit write-only register that enables A/D calibration sample conversion. When the calibration sample is enabled, the A/D conversion produces 10-bit data which corresponds to the calibration analog value (½ VDD = ½ x 3.3 Volt). The 10-bit data will be stored into ADCAL register.

The A/D conversion software can take the advantage of the disparity between the data and the golden value to calibrate the influence of the semiconductor process. The procedure will enhance the accuracy of the sample conversion described below:

Bit	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	-	CAL
Initial Value	0	0	0	0	0	0	0	0
R/W	-	-	-	-	-	-	-	W

Bit	Description	Default
0	Calibration enable (CAL): Enables or disables calibration sample conversion. When set to 1, A/D conversion starts sampling the internal calibration analog value (½ VDD). The converted 10-bit data will be stored into ADCAL register. When the conversion ends, CAL is automatically cleared to 0.	0

19.3 Operation

The A/D converter operates by successive approximations with 10-bit resolution. It has two operating modes: single mode and scan mode.

19.3.1 Single Mode (SCAN = 0)

Single mode should be selected when only one A/D conversion in one channel is required. The A/D conversion starts when the ADST bit is set to 1 by software, or by external trigger input. The ADST bit value remains at 1 during the A/D conversion and is automatically cleared to 0 when the conversion ends.

When the conversion ends, the ADF bit is set to 1. If the ADIE bit is also set to 1, an ADI interrupt is requested at this time. To clear the ADF flag to 0, first read ADCSR, then write 0 in ADF.

When the mode or analog input channel must be switched during the analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the required changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time when the mode or channel is being changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next.

Figure 19-2 shows a timing diagram for this example.

1. Single mode is selected (SCAN = 0), input channel AN1 is selected (CH1= 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).
2. When A/D conversion is completed, the result is transferred into ADDRB. At the same time when the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.
3. Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.
4. The A/D interrupt handling routine starts.
5. The routine reads ADCSR, then writes 0 in the ADF flag, and writes 0 in ADIS flag to reset interrupt.
6. The routine reads and processes the conversion result (ADDRB).

The execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps 2 to 6 are repeated.

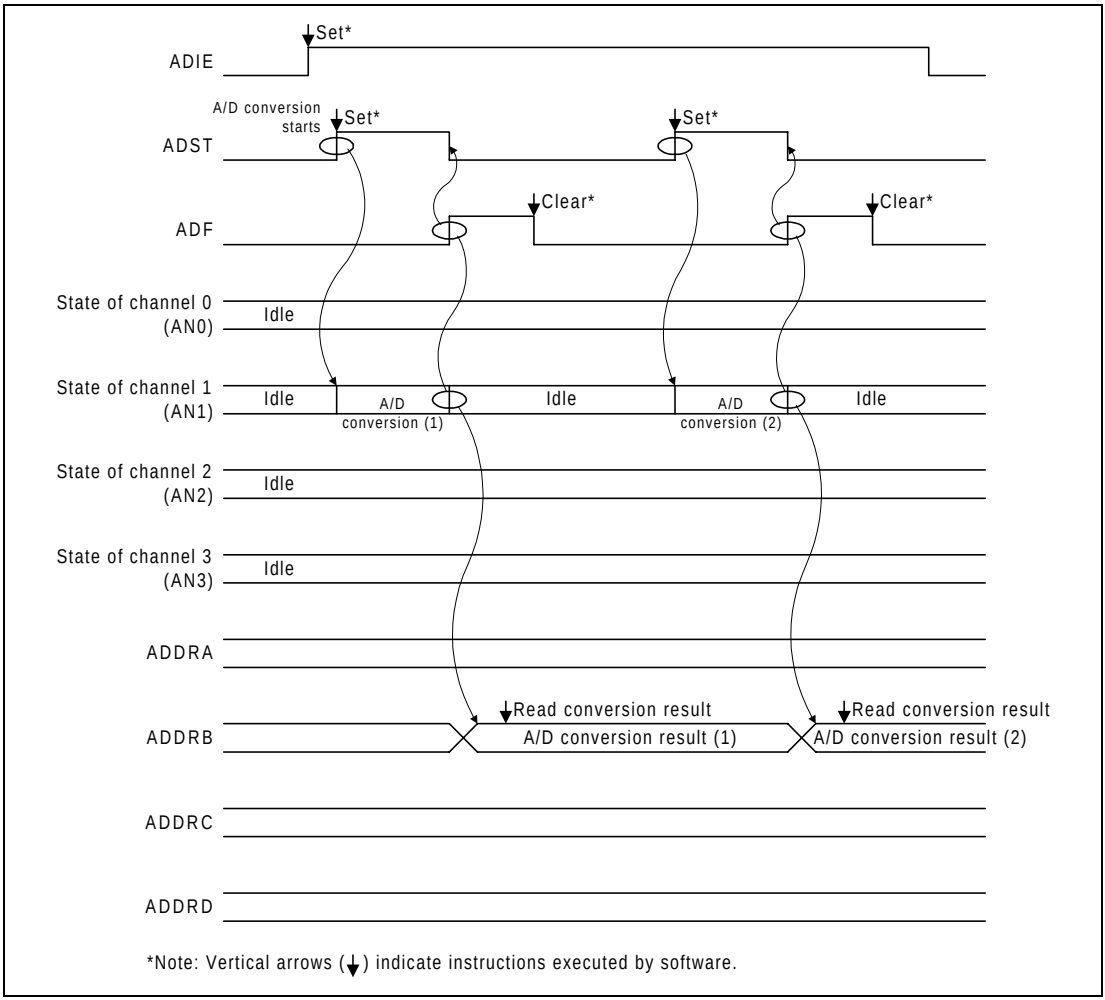


Figure 19.2 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

19.3.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs of one or more channels. When the ADST bit is set to 1 by software or external trigger input, A/D conversion starts on the first channel (AN0). When two or more channels are selected, the conversion of the second channel (AN1) starts immediately after the first channel finishes the conversion. The A/D conversion continues in cycles on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the A/D data registers in correspondences with the selected channels.

When the mode or analog input channel selection must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1. The A/D conversion will then start again from the first channel. The ADST bit can be set at the same time when the mode or channel selection is changed.

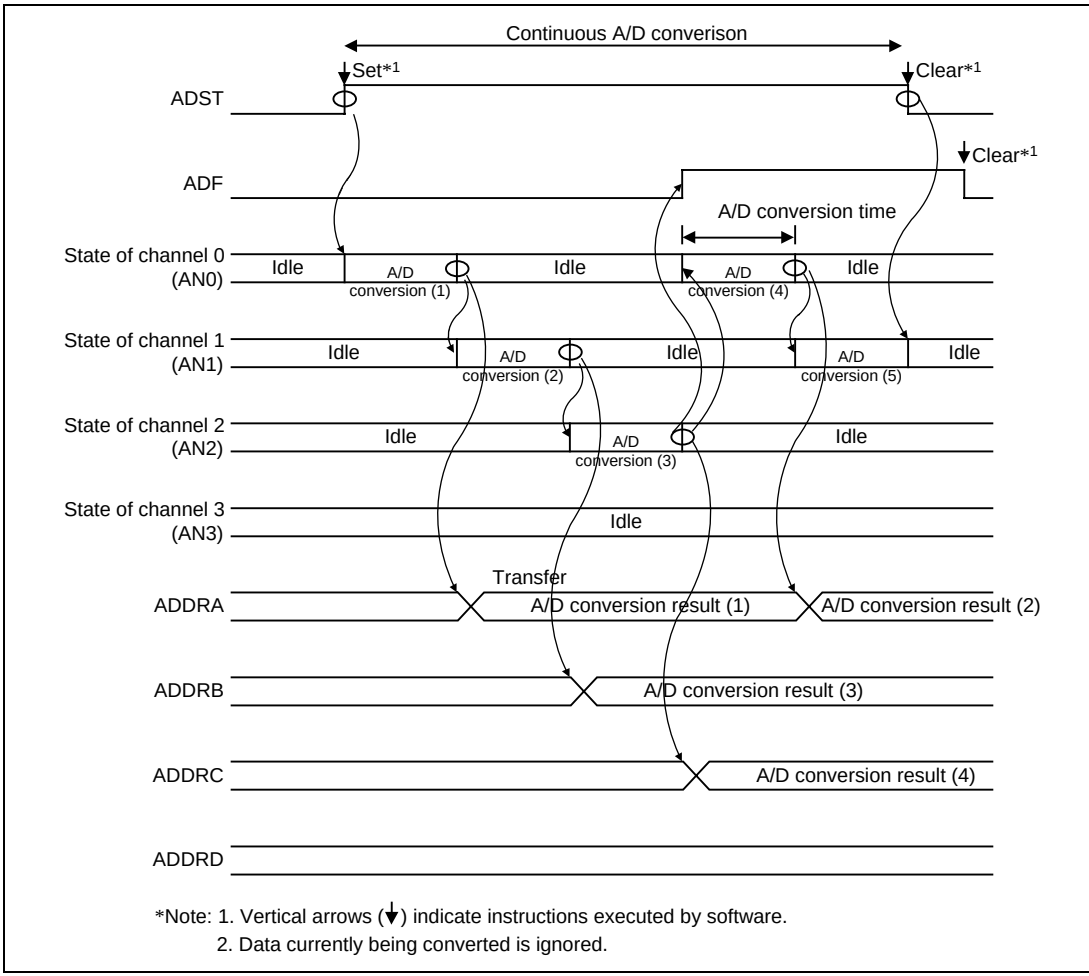
Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 19-3 shows a timing diagram for this example.

1. Scan mode is selected (SCAN = 1), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1).
2. When A/D conversion of the first channel (AN0) is completed, the conversion result is transferred into ADDRA. Next, conversion of the second channel (AN1) starts automatically.
3. Conversion proceeds in the same way through the third channel (AN2).
4. When conversion of all selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1, an ADI interrupt is requested at this time.

Steps 2 to 4 are repeated as long as the ADST bit value remains at 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).

ADCSR (SCAN = 1)

CH1	CH0	ANn
0	0	AN0
0	1	AN0 to AN1
1	0	AN0 to AN2
1	1	AN0 to AN3



*Note: 1. Vertical arrows (↓) indicate instructions executed by software.
2. Data currently being converted is ignored.

Figure 19.3 Example of A/D Converter Operation (Scan Mode, Channels AN0 to AN2 Selected)

19.3.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts the conversion. Figure 19-4 shows the A/D conversion timing. Table 19-4 on the next page indicates the A/D conversion time.

As indicated in figure 19-4 below, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 19-4.

In scan mode, the values given in table 19-4 are applied to the first conversion. In the second and subsequent conversions, the conversion time is the same as the first conversion time.

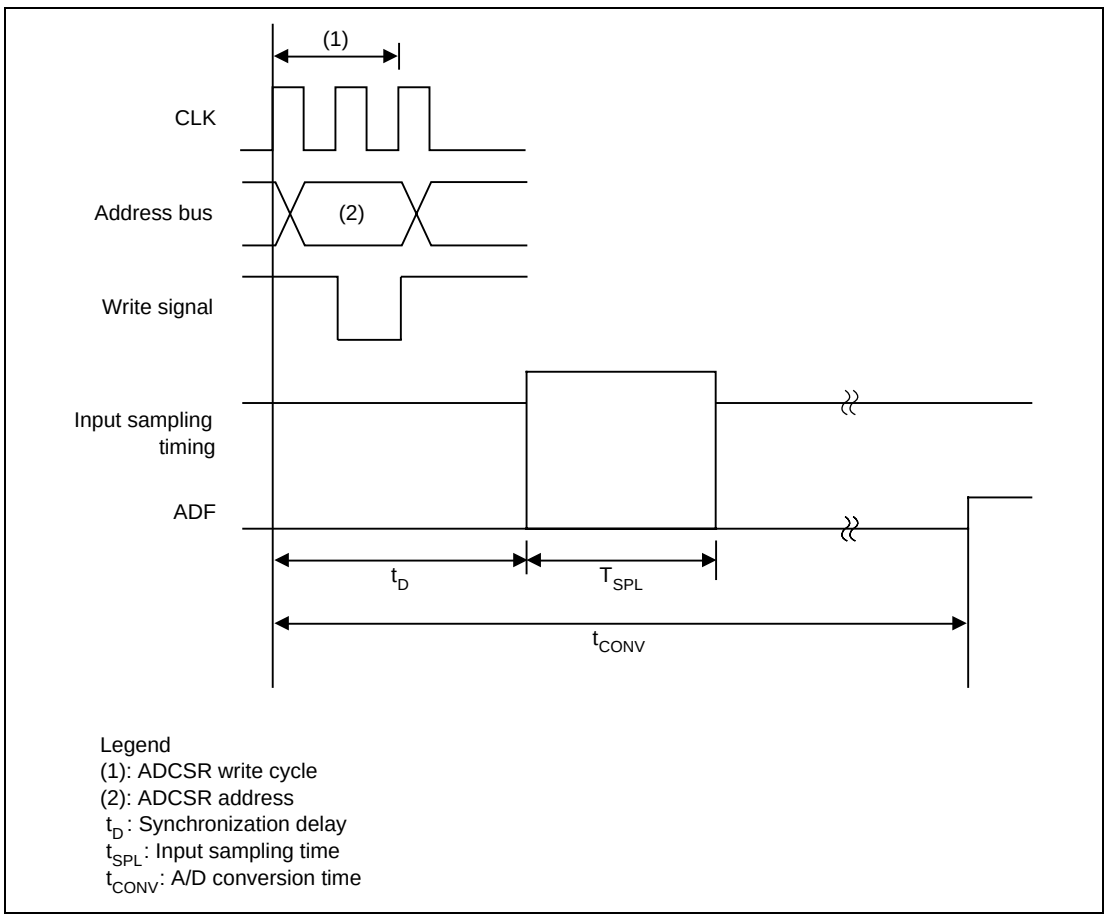


Figure 19.4 A/D Conversion Timing

Table 19.4 A/D Conversion Time (Single Mode) unit: µs

	Symbol	Min	Typ	Max
Synchronization delay	t_D	0.5	-	1.2
Input sampling time	T_{SPL}	-	2.8	-
A/D conversion time	T_{CONV}	-	-	10

19.3.4 A/D External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGE bit is set to 1 in ADCR, the external trigger input is enabled by TMOUT#. A high-to-low transition at the TMOUT# sets the ADST bit of ADCSR to 1, and starts A/D conversion. Other operation, in both single and scan modes, are the same as if the ADST bit had been set to 1 by software. Figure 19-5 shows the timing.

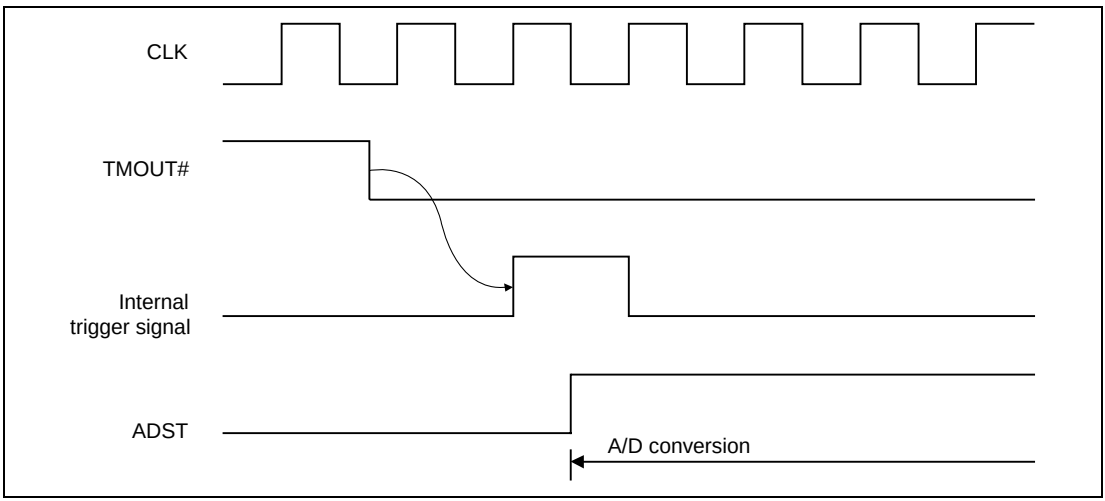


Figure 19.5 External Trigger Input Timing

19.4 Interrupts

The A/D converter generates an interrupt (ADI) at the end of A/D conversion. The ADI interrupt request can be enabled or disabled by the ADIE bit in ADCSR.

19.5 Usage Notes

When using the A/D converter, note the following points:

Analog Input Voltage Range: During A/D conversion, the voltages input to the analog input pins ANn should be in the range $0 \leq ANn \leq 3.3V$. (n = 0 to 3)

AVcc and AVss Input Voltages: AVss should have the following value: AVss = Vss.

If the A/D converter is not used, the values should be AVcc = Vss and AVss = Vss.

19.6 A/D Conversion Characteristics

Table 19-5 lists the A/D conversion characteristics.

Table 19.5 A/D Conversion Characteristics

(Vcc = 5 ± 0.5V, AVcc = 5 ± 0.5V, Ta = -20 to 75 °C)

Item	Min	Typ	Max	Unit
Resolution	10	10	10	bits
Conversion time	-	-	10	µs
Analog input capacitance	-	-	20	pF
Permissible signal-source impedance	-	-	5	KΩ
Non-linearity error	-	-	±3.0	LSB
Offset error	-	-	±2.0	LSB
Full-scale error	-	-	±2.0	LSB
Quantization error	-	-	±0.5	LSB
Absolute accuracy	-	-	±4.0	LSB

19.7 Analog Input Pin Characteristics

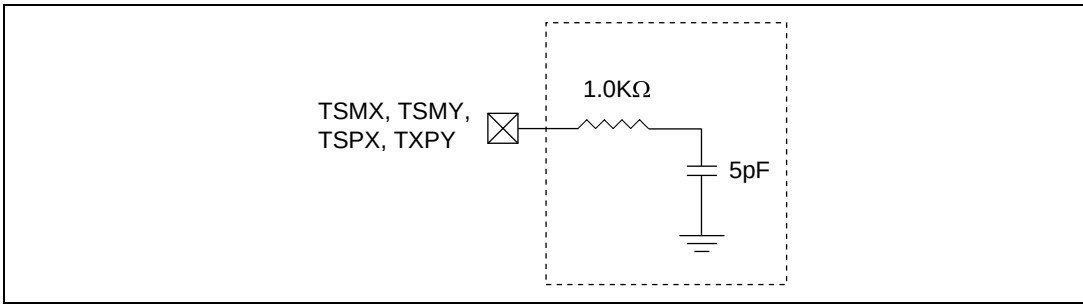


Figure 19.6 Analog Input Pin RC Equivalent Circuit

Table 19.6 Analog Input Pin Characteristics

Item	**Min.**	Max.	Unit
Analog Input Capacitance	-	5	pF
Analog Input Impedance	-	1	KΩ

Section 20 Electrical Characteristics

(VCC =3.3V±0.3V, VCCA, VCCB, VCC5=5.0V±0.5V, Ta=0°C to 70°C, unit : ns)

20.1 DC Electrical Characteristics

Absolute Maximum Ratings*

- Applied Voltage of VCC .....-0.3V to +4.6V
- Applied Voltage of VCC5 .....-0.3V to +7.0V
- Applied Voltage of VCCA and VCCB .....-0.3V to +7.0V
- Input Voltage (Except Printer Port Interface) .....-0.3V to VCC +0.3V
- Input Voltage of Printer Port Interface .....-0.3V to VCC5 +0.3V
- Tcase ..... 0°C to +70°C
- Storage Temperature .....-55°C to +125°C

***Comments:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to this device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied, and exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 20.1 DC Electrical Characteristics (Ta=0°C to 70°C)

Items		Symbol	Min.	Typ.	Max.	Unit	Measurement Condition
Power supply voltage		VCC	3.0	3.3	3.6	V	
		VCCA,B	3.0	3.3	3.6	V	For 3.3V operation
		VCCA,B	4.5	5.0	5.5	V	For 5V operation
		AVCC	3.0	3.3	3.6	V	
Current	Normal operation	ICC	-	50	100	mA	
	Standby mode	ICC	-	-	70	µA	At Ta=25°C
		ICCA,B	-	-	6		

Table 20.1 DC Electrical Characteristics (Ta=0°C to 70°C) [cont'd]

Symbol	Parameter	Min.	Typ.	Max.	Conditions
3.3V Interface					
V_{IL}	Input Low Voltage	- 0.3V		VCC x 0.2	VCC=3.0 ~ 3.6V
V_{IH}	Input High Voltage	VCC x 0.7		VCC+ 0.3V	VCC=3.0 ~ 3.6V
V_{OL}	Output Low Voltage			0.5	I_{OL}= -2mA
V_{OH}	Output High Voltage	2.4			I_{OH}= 2mA
I_{IL}	Input Low Current	-10µA	1µA		V_{IL}= V_{SS} no pull-up or pull-down
I_{IH}	Input High Current		1µA	10µA	V_{IH}= VCC no pull-up or pull-down
I_{OZ}	Tri-state Leakage Current		1µA	10µA	
Cin	Input Capacitance		5pF		
Cout	Output Capacitance		5 pF		
Cbld	Bi-directional Buffer		5 pF		
5V Interface					
V_{IL}	Input Low Voltage	- 0.3V		VCC5 x 0.2	VCC5=4.5V ~ 5.5V
V_{IH}	Input High Voltage	VCC5 x 0.7		VCC5+ 0.3V	VCC5=4.5V ~ 5.5V
V_{OL}	Output Low Voltage			0.5	I_{OL}= -2mA
V_{OH}	Output High Voltage	2.4			I_{OH}= 2mA
I_{IL}	Input Low Current	-10µA	1µA		V_{IL}= V_{SS} no pull-up or pull-down
I_{IH}	Input High Current		1µA	10µA	V_{IH}= VCC5 no pull-up or pull-down
I_{OZ}	Tri-state Leakage Current		1µA	10µA	
Cin	Input Capacitance		5pF		
Cout	Output Capacitance		5 pF		
Cbld	Bi-directional Buffer		5 pF		

20.2 AC Characteristics

Table 20.2 CPU Interface AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{BSD}	BS# delay time	3	-	Figure 20.1
t_{BSS}	BS# setup time	7	-	Figure 20.1
t_{BSH}	BS# hold time	0	-	Figure 20.1
t_{CS4S}	CS4# setup time	7	-	Figure 20.1
t_{AS}	Address setup time	5	-	Figure 20.1
t_{RWS}	RDWR# setup time	5	-	Figure 20.1
t_{WES}	Write enable setup time	5	-	Figure 20.1
t_{WTD1HS}	RDY#/WAIT# delay time 1 (High Speed)	6	10	Figure 20.1
t_{WTD1LS}	RDY#/WAIT# delay time 1 (Low Speed)	6	20	Figure 20.1
t_{WTD2HS}	RDY#/WAIT# delay time 2 (High Speed)	8	13	Figure 20.1
t_{WTD2LS}	RDY#/WAIT# delay time 2 (Low Speed)	8	30	Figure 20.1
t_{WDS}	Write data setup time	5	-	Figure 20.1
t_{WDH}	Write data hold time	0	-	Figure 20.1
t_{RSS}	RD# setup time	5	-	Figure 20.2
t_{RDD1}	Read data delay time 1	-	30	Figure 20.2
t_{RDD2}	Read data delay time 2	-	25	Figure 20.2

Table 20.3 Crystal/Oscillator and PLL Settle AC Timing Spec. (unit : ms)

Symbol	Parameter	Min.	Max	Comment
t_{OSC}	Oscillator stablization time	-	15	Figure 20.3
t_{PLL}	PLL stablization time	-	5	Figure 20.3
t_{PORST}	Power On Reset Output time	-	10	Figure 20.3

Table 20.4 GPIO AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{PxQDF}	Pxn* to IRQ0# delay time (falling trigger)	-	16	Figure 20.4
t_{PxQDR}	Pxn* to IRQ0# delay time (rising trigger)	-	16	Figure 20.5

*: Pxn: PA7 – PA0, PB7 – PB0, PC7– PC0, PD7 – PD0, PE7 – PE0.

Table 20.5 I/O Port Interrupt AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{TM0D1}	PB0/TMO0# delay time 1	-	20	Figure 20.6
t_{TM0D2}	PB0/TMO0# delay time 2	-	20	Figure 20.6
t_{TM1D1}	PB1/TMO1# delay time 1	-	20	Figure 20.7
t_{TM1D2}	PB1/TMO1# delay time 2	-	20	Figure 20.7
t_{TMQD1}	IRQ0# delay time 1	-	28	Figure 20.6, Figure 20.7
t_{TMQD2}	IRQ0# delay time 2	-	28	Figure 20.6, Figure 20.7
t_{DREQD}	DREQ0#, DREQ1# delay time	-	17	Figure 20.8

Table 20.6 PCMCIA AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{AD}	Address Delay to CPU Address	-	12	Figure 20.9, Figure 20.10
t_{CSD}	Card Enable Delay to CPU Card Enable	-	11	Figure 20.9, Figure 20.10
t_{RSD}	Read Strobe Delay to CPU Read Strobe	-	11	Figure 20.10
t_{WED}	Write Enable Delay to CPU Write Enable	-	12	Figure 20.10
t_{WDD}	Write Data Delay to CPU Write Data Delay	-	14	Figure 20.9, Figure 20.10
t_{ICRSD}	IO Read Strobe Delay to CPU IO Read Strobe Delay	-	11	Figure 20.9
t_{ICWSD}	IO Write Strobe Delay to CPU IO Write Enable Delay	-	11	Figure 20.8

Table 20.7 UART AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
$t_{URDTRD1}$	UART DTR0# delay time 1	-	45	Figure 20.11
$t_{URDTRD1}$	UART RTS0# delay time 1	-	45	Figure 20.11
$t_{URDTRD2}$	UART DTR0# delay time 2	-	45	Figure 20.11
$t_{URDTRD2}$	UART RTS0# delay time 2	-	45	Figure 20.11
t_{URIRQD}	Delay from stop bit (RXD0) to set IRQ0#	-	26*	Figure 20.12

*unit cycle time: (UCK cycle time 26) / 4. For example, if UCK clock frequency is 12 MHz, the unit cycle time is 556 ns.

Table 20.8 Parallel Port AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
$t_{PPIRQD1}$	Delay from ACK# asserted to IRQ0# asserted	-	100	Figure 20.13
$t_{PPIRQD2}$	Delay from ACK# deasserted to IRQ0# deasserted	-	100	Figure 20.13
$t_{PPSLIND1}$	SLIN# deassert to BUSY asserted (EPP write address cycle)	65	460	Figure 20.14
$t_{PPSLIND1}$	AFD# deassert to BUSY asserted (EPP write data cycle)	65	460	Figure 20.14
$t_{PPSLIND2}$	SLIN # deassert to BUSY asserted (EPP read data cycle)	65	460	Figure 20.15
$t_{PPSLIND2}$	AFD# deassert to BUSY asserted (EPP read data cycle)	65	460	Figure 20.15
$t_{PPSTBD1}$	STB# asserted to BUSY deasserted (forward)	-	750	Figure 20.16
$t_{PPSTBD2}$	STB# deasserted to BUSY asserted (forward)	-	900	Figure 20.16
$t_{PPAFDD1}$	AFD# asserted to BUSY asserted (backward)	-	1100	Figure 20.17
$t_{PPAFDD2}$	AFD# deasserted to ACK# deasserted (backward)	-	800	Figure 20.17

Table 20.9 SCDI AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{RST_LOW}	ACRST# Active Low Pulse Width	1000	-	Figure 20.19
t_{SYN_HIGH}	SIBSYNC Active HIGH Pulse Width	1000	-	Figure 20.20
t_{SYNCD1}	SIBSYNC Delay Time 1	-	15	Figure 20.21
t_{SYNCD2}	SIBSYNC Delay Time 2	-	15	Figure 20.21
t_{SDOUTD}	SIBDOUT Delay Time	-	15	Figure 20.21
$t_{DREQ1D1}$	DREQ1# Delay Time 1	-	15	Figure 20.18
$t_{DREQ1D2}$	DREQ1# Delay Time 2	-	15	Figure 20.18
t_{SDINSU}	SIBDIN Setup Time	10	-	Figure 20.21
t_{SDINHD}	SIBDIN Hold Time	10	-	Figure 20.21

Table 20.10 AFE Interface AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{MCKCYC}	MCLKO Period	300	-	Figure 20.22
t_{MCKL}	MCLKO Width Low	130	-	Figure 20.22
t_{MCKH}	MCLKO Width High	130	-	Figure 20.22
t_{MCKF}	MCLKO Falling Time	-	20	Figure 20.22
t_{MCKR}	MCLKO Rising Time	-	20	Figure 20.22
t_{FSS}	FS setup Time	100	-	Figure 20.22
t_{FSH}	FS hold Time	100	-	Figure 20.22
t_{RXS}	DIN setup Time	50	-	Figure 20.22
t_{RXH}	DIN hold Time	50	-	Figure 20.22
t_{TXD}	DOUT Delay Time	-	50	Figure 20.22

Table 20.11 KBC AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{IORD}	XIOR# Delay to KBCS#	40	-	Figure 20.23
t_{RCSD}	KBCS# Delay to XIOR#	40	-	Figure 20.23
t_{IORPW}	XIOR# pulse width	240	-	Figure 20.23
t_{IOWD}	XIOW# Delay to KBCS#	40	-	Figure 20.23
t_{WCSD}	KBCS# Delay to XIOW#	40	-	Figure 20.23
t_{IOWPW}	XIOW# pulse width	120	-	Figure 20.24

Table 20.12 USB Host AC Timing Spec. (unit : ns)

Symbol	Parameter	Min.	Max	Comment
t_{OTP}	USB over-current to power enable	-	250	Figure 20.25

Table 20.13 AFECK clock input AC Timing Spec. (PLL1 : bypass) (unit : ns)

Symbol	Parameter	Min	Typ	Max	Comment
t_{ACKcyc}	AFECK clock input cycle time	26	27.174	28	Figure 20.26
t_{ACKH}	AFECK clock input high level pulse width	8		-	Figure 20.26
t_{ACKL}	AFECK clock input low level pulse width	8		-	Figure 20.26
t_{ACKr}	AFECK clock input rise time	-		4	Figure 20.26
t_{ACKf}	AFECK clock input fall time	-		4	Figure 20.26

Table 20.14 AFECK clock input AC Timing Spec. (PLL1 : operating) (unit : ns)

Symbol	Parameter	Min	Typ	Max	Comment
t_{ACKcyc}	AFECK clock input cycle time	80	81.38	83	Figure 20.26
t_{ACKH}	AFECK clock input high level pulse width	34		-	Figure 20.26
t_{ACKL}	AFECK clock input low level pulse width	34		-	Figure 20.26
t_{ACKr}	AFECK clock input rise time	-		4	Figure 20.26
t_{ACKf}	AFECK clock input fall time	-		4	Figure 20.26

Table 20.15 UCK clock input AC Timing Spec. (PLL2 : bypass) (unit : ns)

Symbol	Parameter	Min	Typ	Max	Comment
t_{UCKcyc}	UCK clock input cycle time	19	20.83	22	Figure 20.27
t_{UCKH}	UCK clock input high level pulse width	5		-	Figure 20.27
t_{UCKL}	UCK clock input low level pulse width	5		-	Figure 20.27
t_{UCKr}	UCK clock input rise time	-		4	Figure 20.27
t_{UCKf}	UCK clock input fall time	-		4	Figure 20.27

Table 20.16 UCK clock input AC Timing Spec. (PLL2 : operating) (unit : ns)

Symbol	Parameter	Min	Typ	Max	Comment
t_{UCKcyc}	UCK clock input cycle time	81	83.33	85	Figure 20.27
t_{UCKH}	UCK clock input high level pulse width	35		-	Figure 20.27
t_{UCKL}	UCK clock input low level pulse width	35		-	Figure 20.27
t_{UCKr}	UCK clock input rise time	-		4	Figure 20.27
t_{UCKf}	UCK clock input fall time	-		4	Figure 20.27

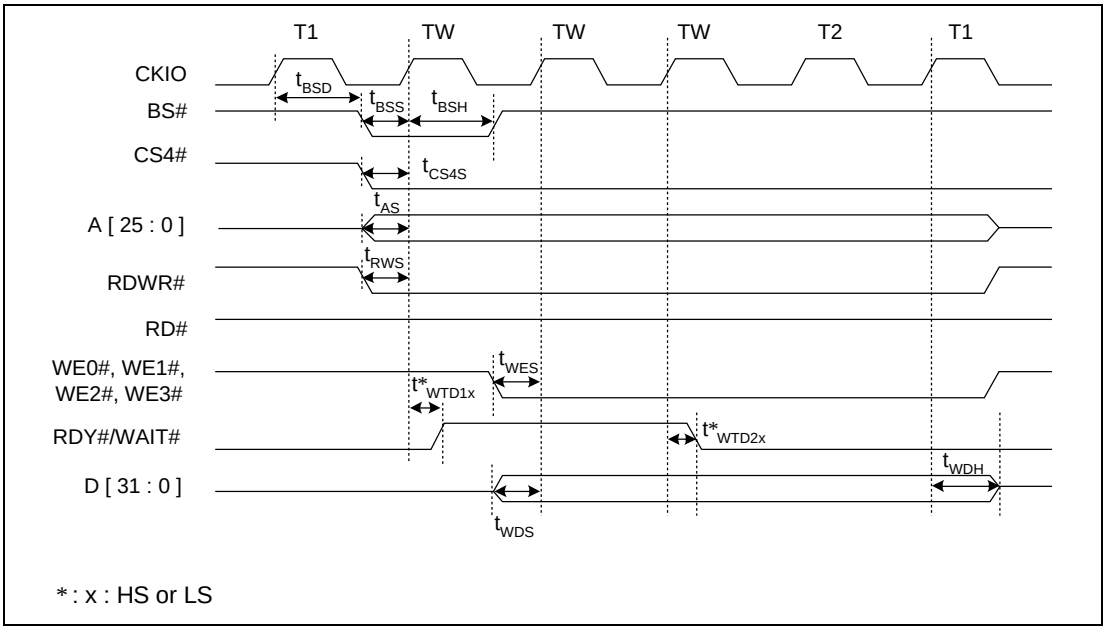


Figure 20.1 CPU Write Cycle Timing Diagram

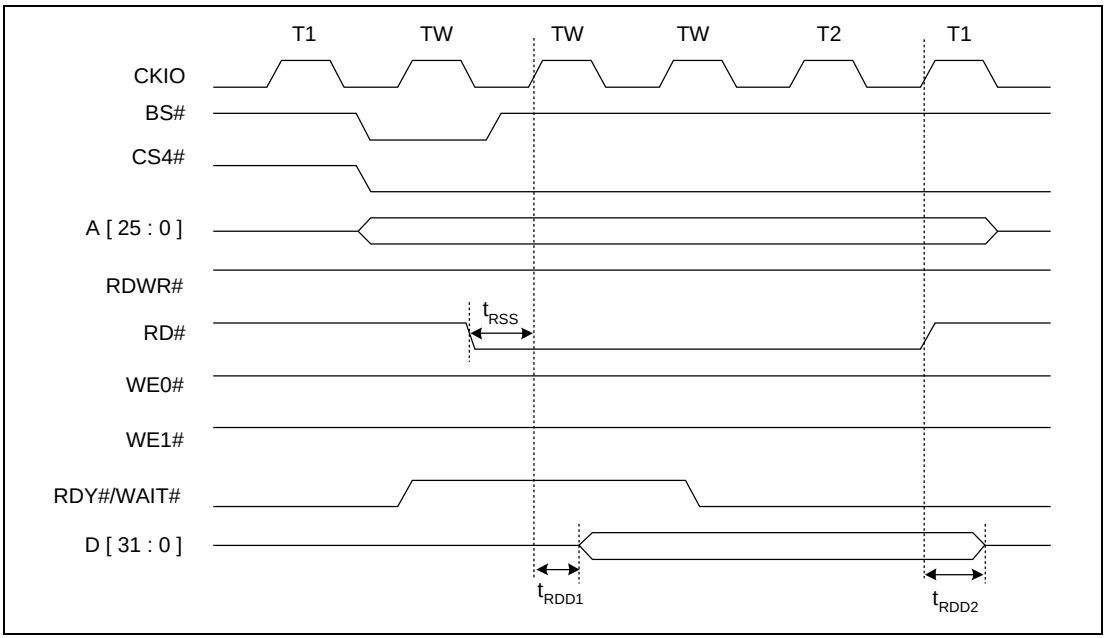


Figure 20.2 CPU Read Cycle Timing Diagram

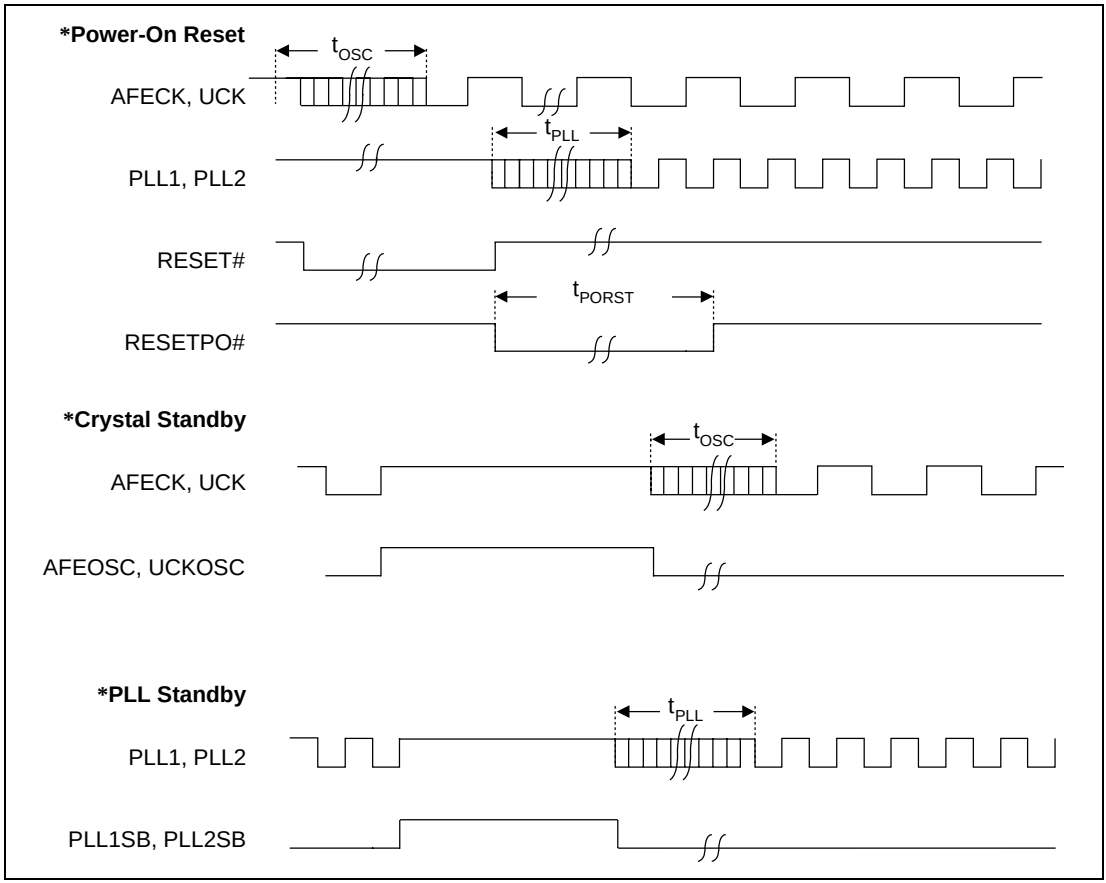


Figure 20.3 Crystal/Oscillator and PLL Settle Timing Diagrams

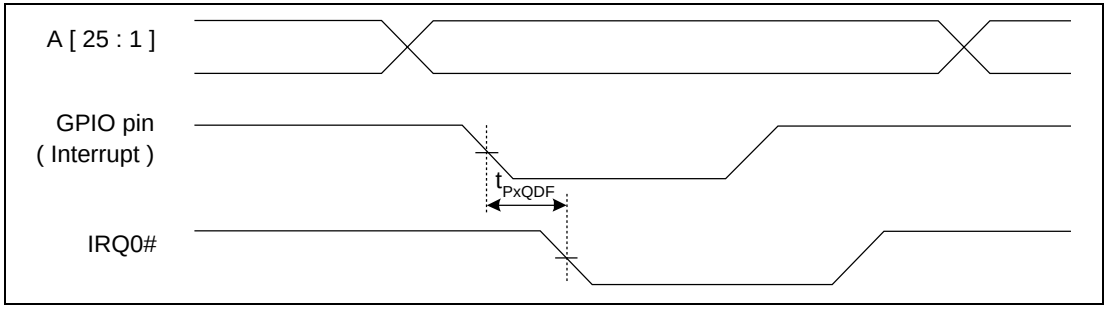


Figure 20.4 I/O Port Interrupt Timing (Falling Edge Trigger)

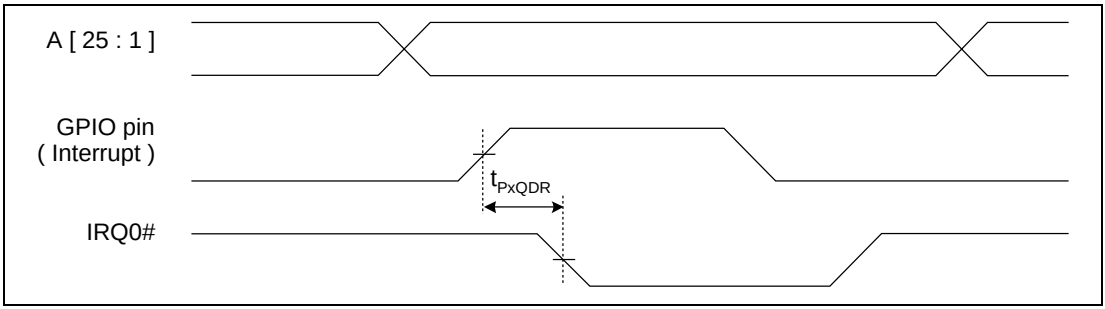


Figure 20.5 I/O Port Interrupt Timing (Rising Edge Trigger)

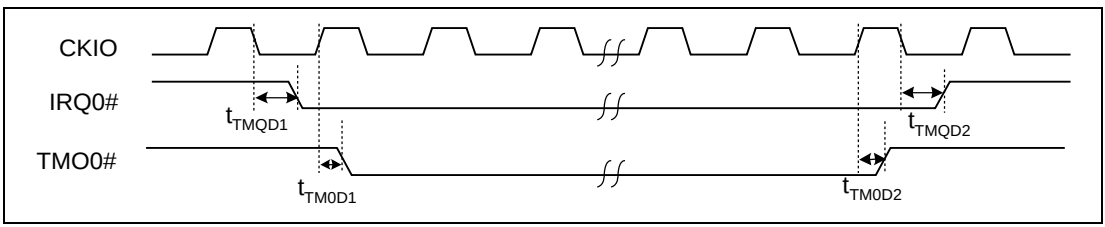


Figure 20.6 IRQ0#/TMO0# Timing For Timer

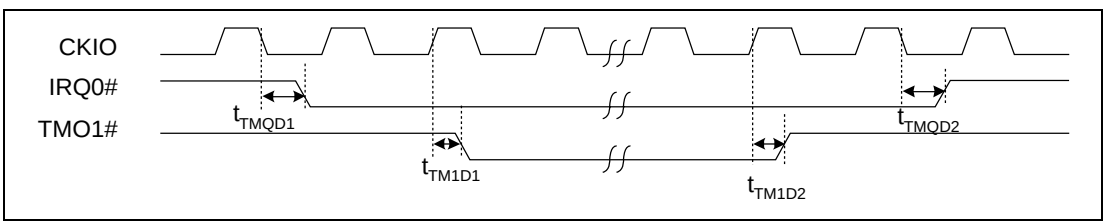


Figure 20.7 IRQ0#/TMO1# Timing For Timer

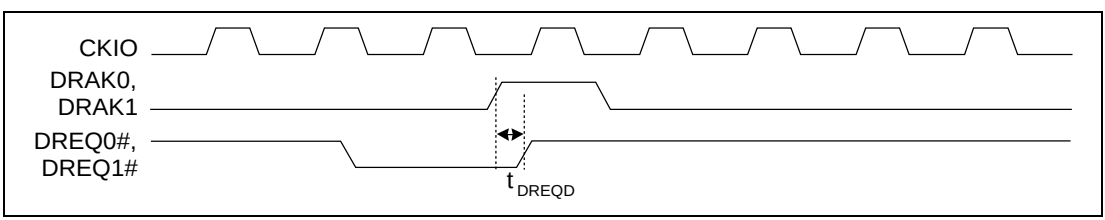


Figure 20.8 DREQ0# / DREQ1# Timing

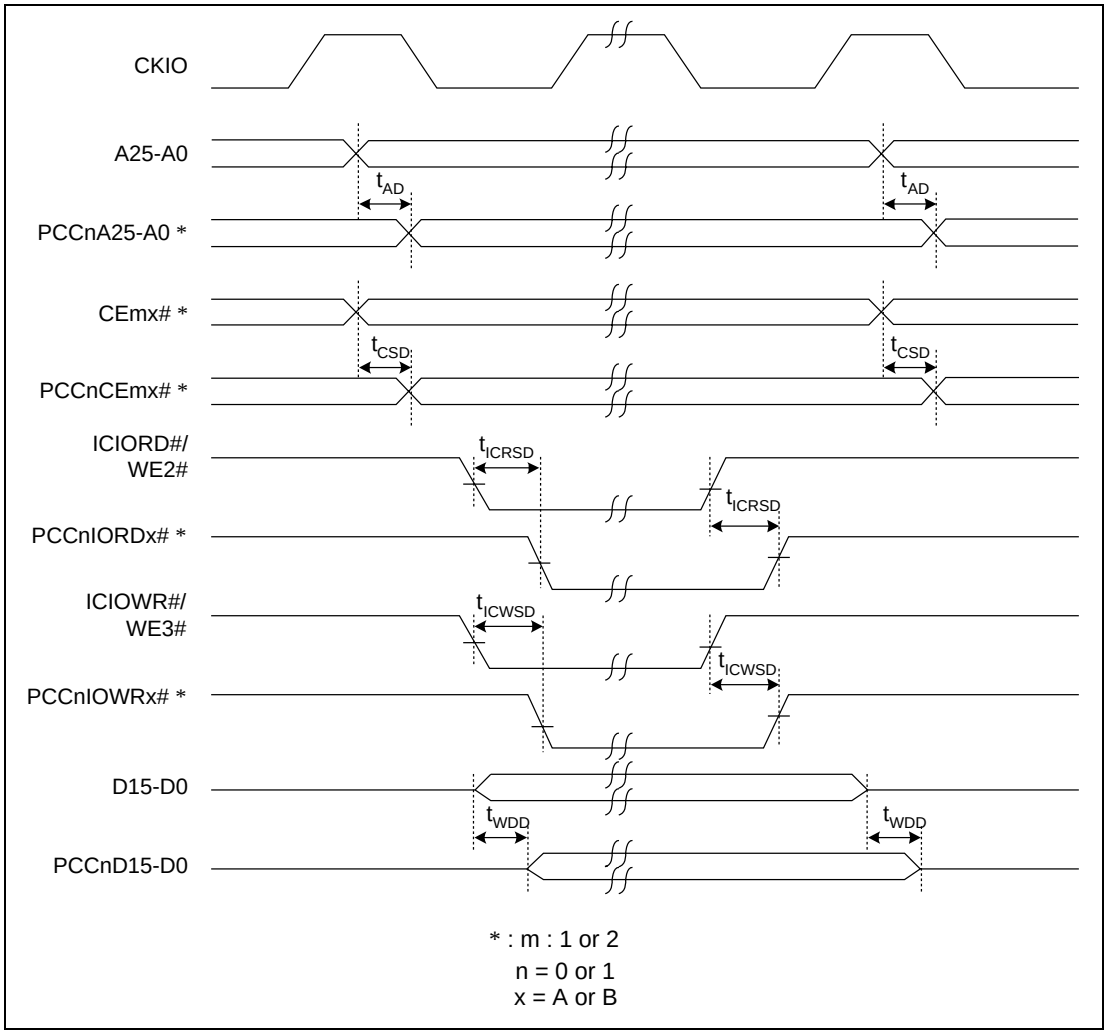


Figure 20.9 PCMCIA I/O Bus Cycle (NO Wait)

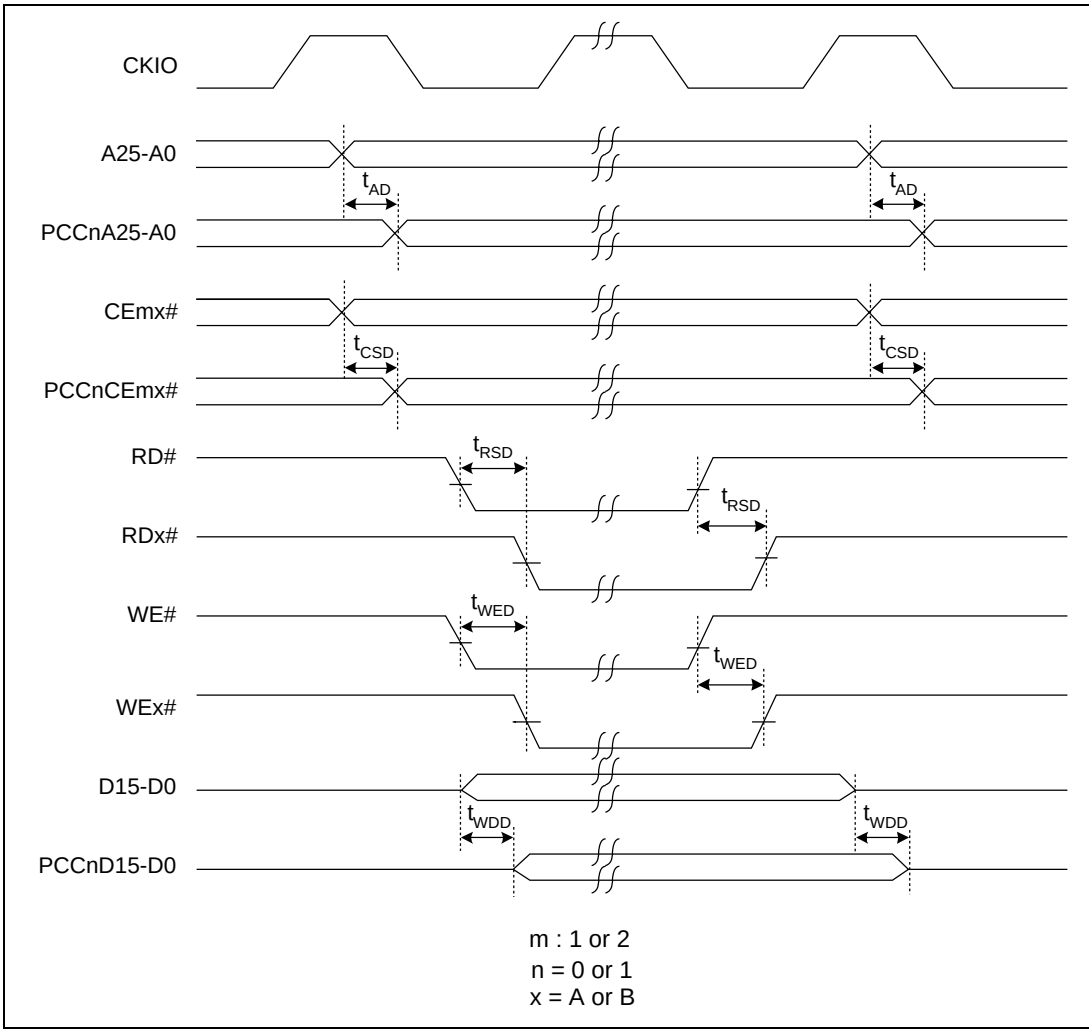


Figure 20.10 PCMCIA Memory Bus Cycle (No Wait)

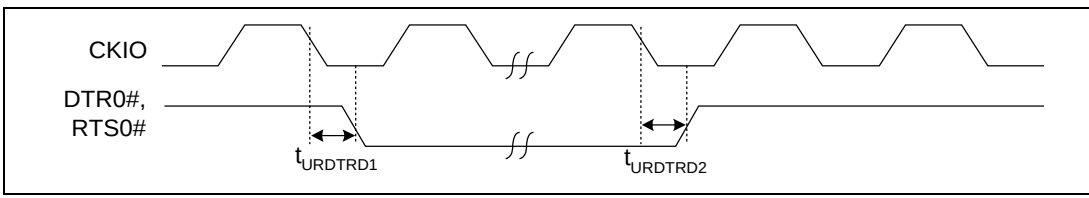


Figure 20.11 UART DTR, RTS Timing

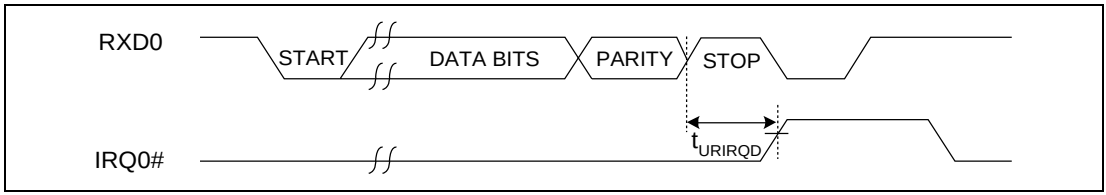


Figure 20.12 UART Rx Timing

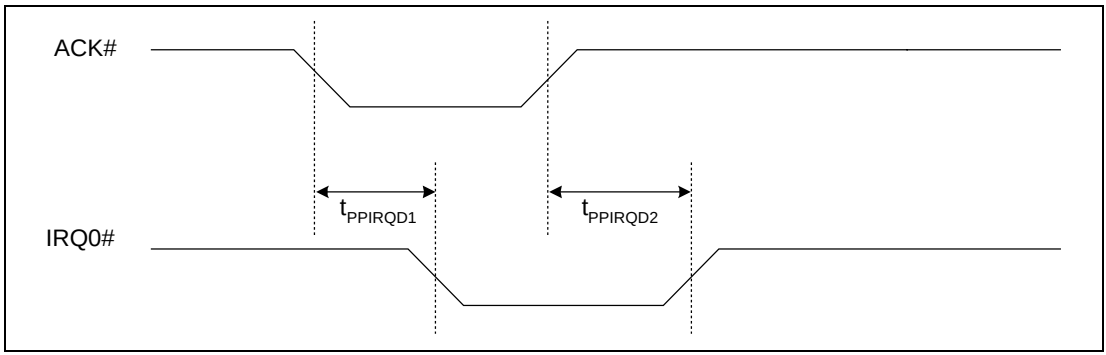


Figure 20.13 Control Signal Delay Time of Parallel Port Timing

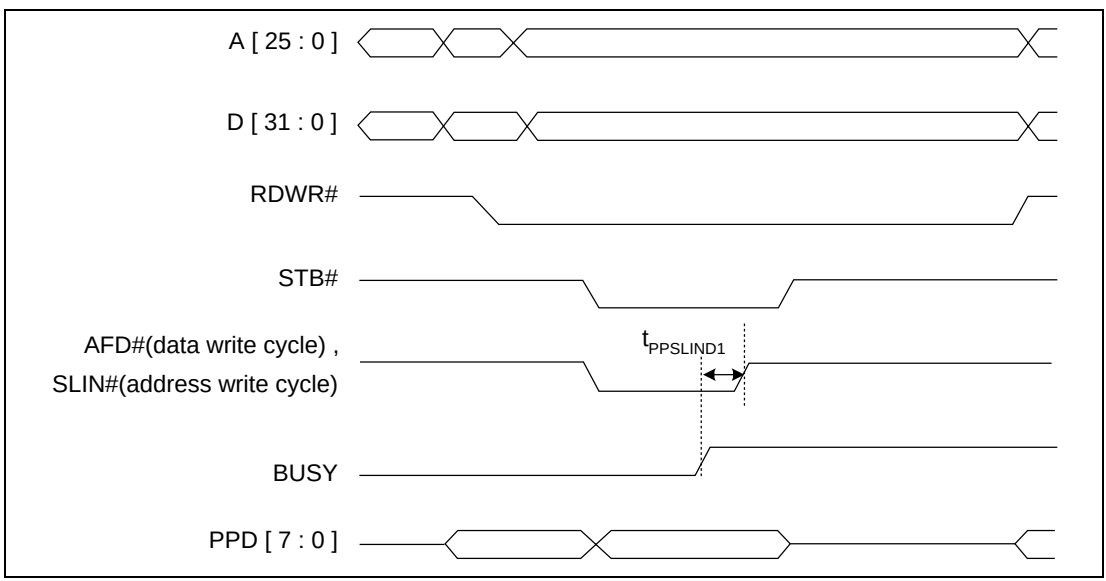


Figure 20.14 EPP Address or Data Write Timing

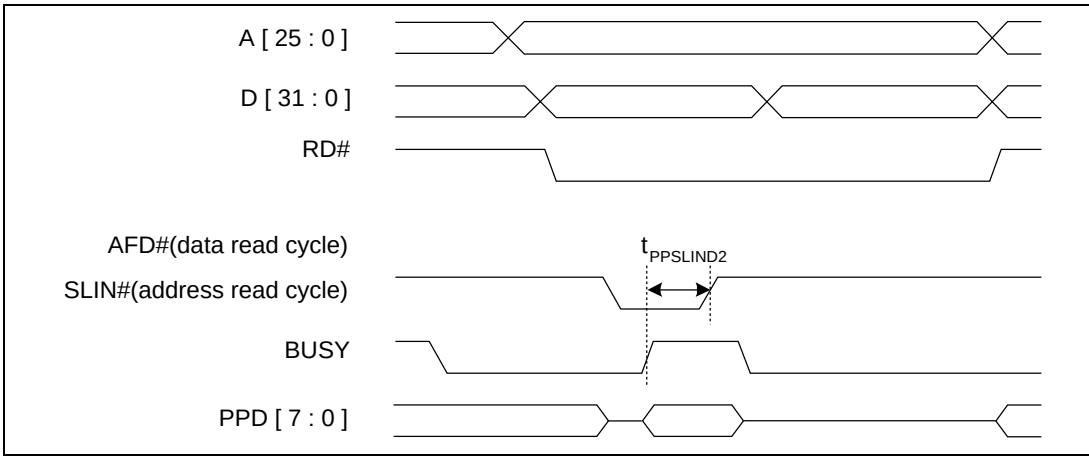


Figure 20.15 EPP Address or Data Read Timing

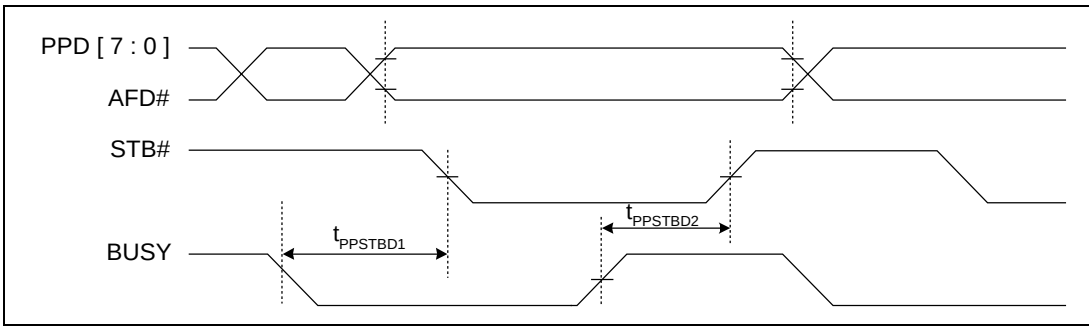


Figure 20.16 ECP Parallel Port Forward Timing

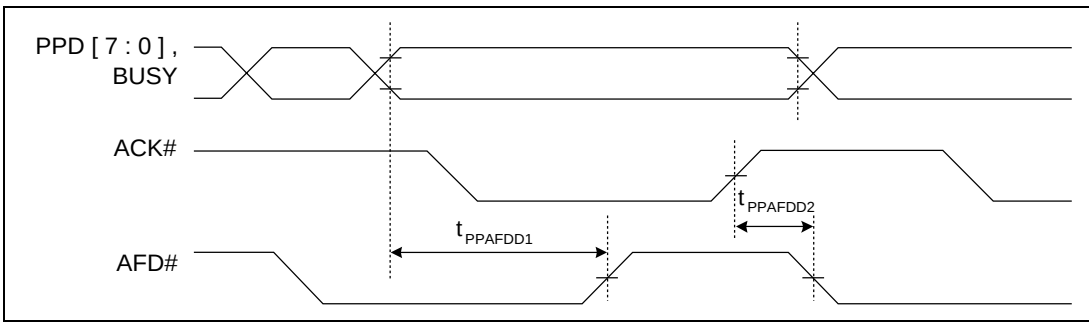


Figure 20.17 ECP Parallel Port Backward Timing

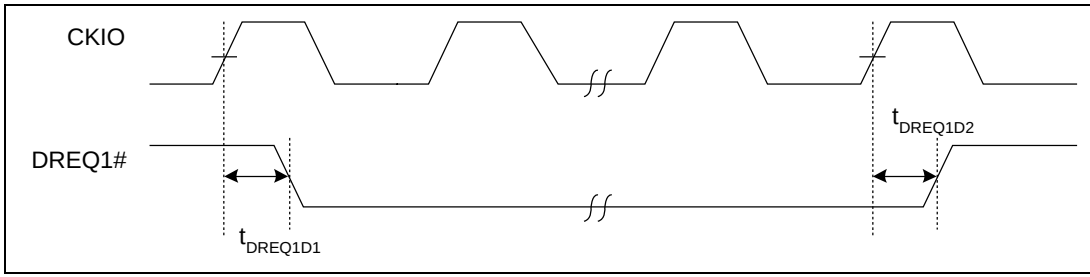


Figure 20.18 SCDI DMA Request Timing

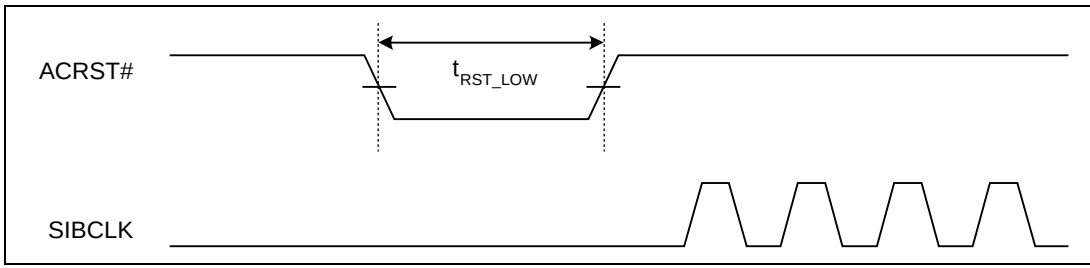


Figure 20.19 Cold Reset Timing

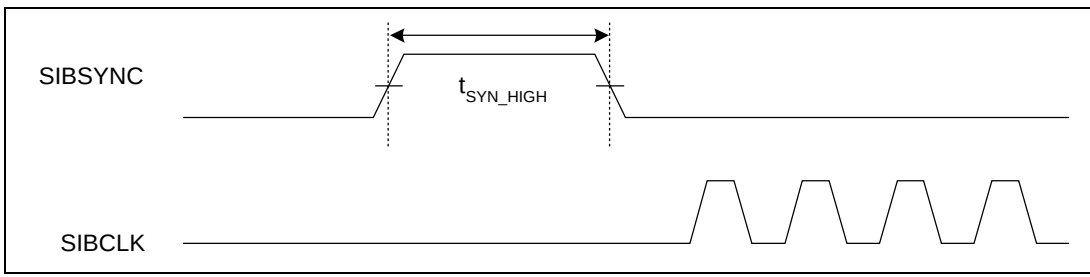


Figure 20.20 Warm Reset Timing

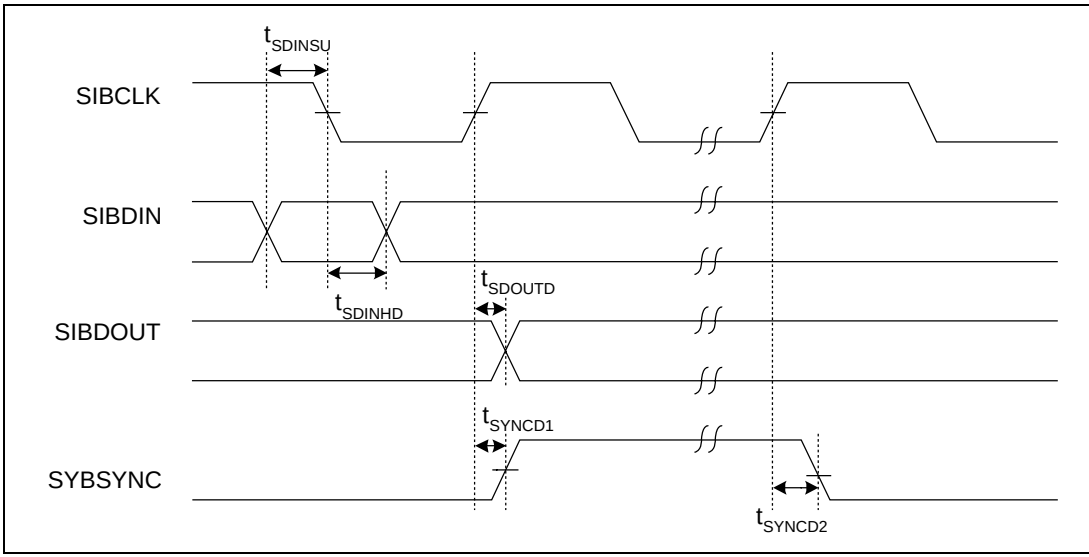


Figure 20.21 SCDI Sync and Data Timing

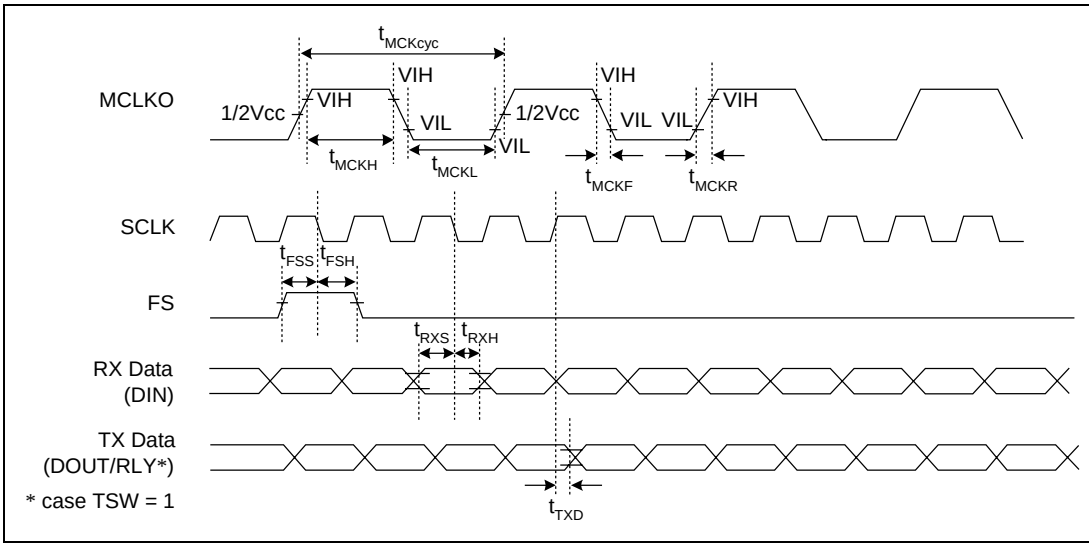


Figure 20.22 AFE Interface Access Timing

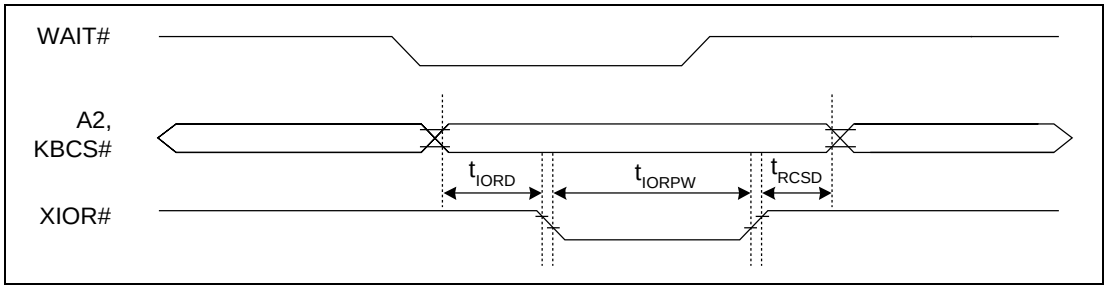


Figure 20.23 Keyboard Controller Interface Read Timing

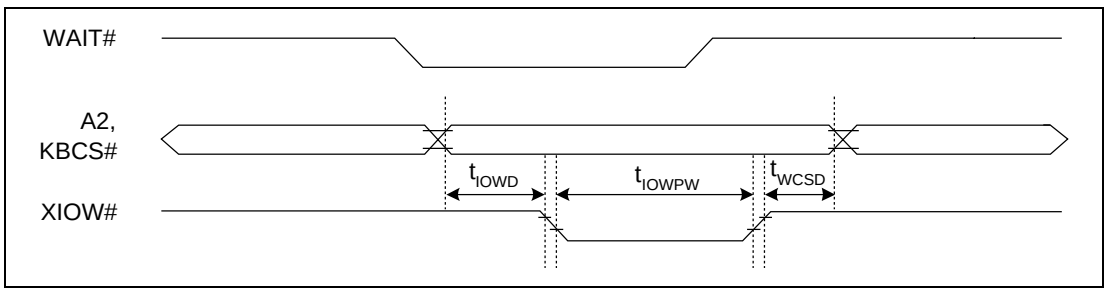


Figure 20.24 Keyboard Controller Interface Write Timing

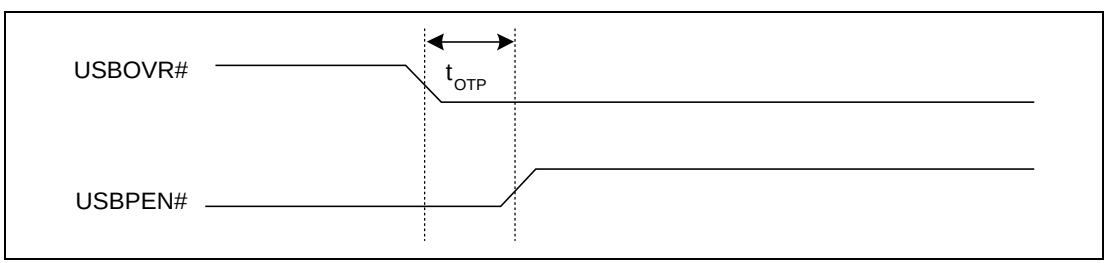


Figure 20.25 USB Over-Current Detect to Power Down Timing

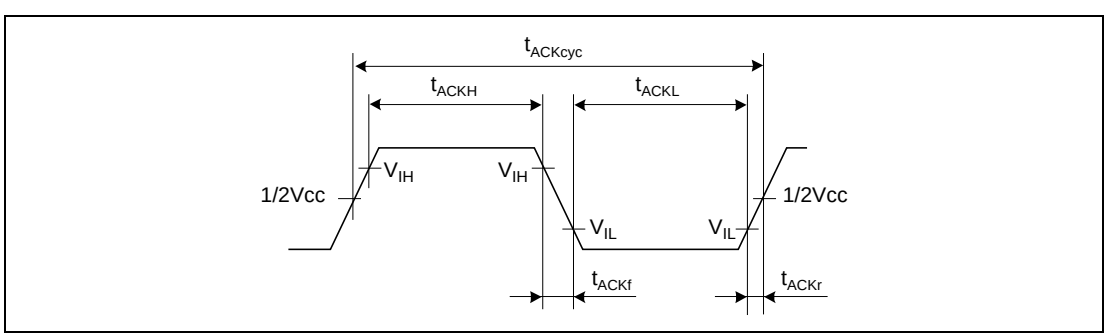


Figure 20.26 AFECK Clock Input Timing

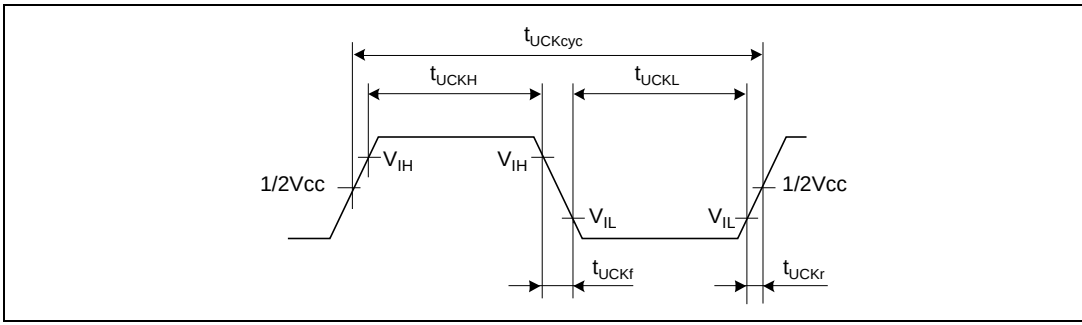
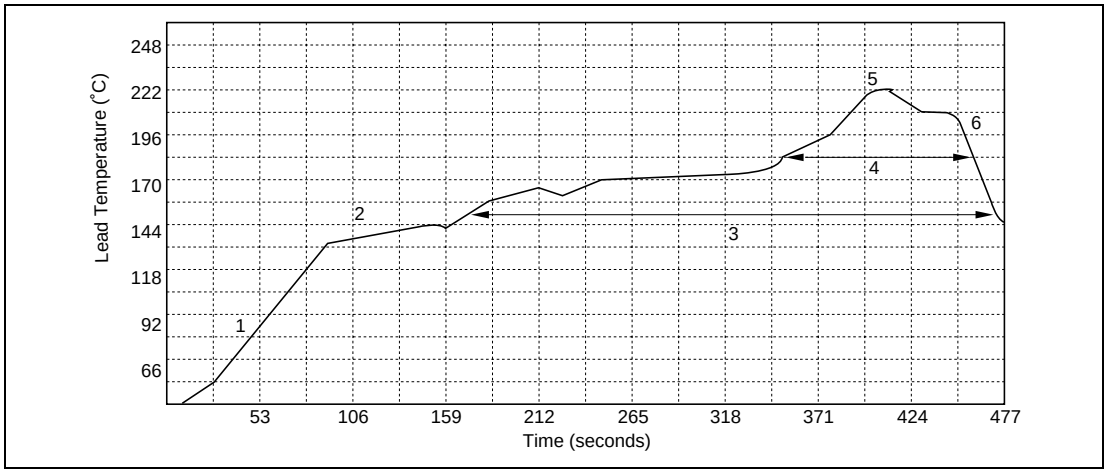


Figure 20.27 UCK Clock Input Timing

Section 21 Recommended Reflow Condition



Lead Soldering Profile in the IR Oven

Limits for IR Reflow Profile Characteristics of Package Leads

Characteristics #	Characteristic Description	Limits
1	Initial Heating Rate of Leads	0.8-1.2°C / Sec.
2	Peak Lead Temperature in Preheat Zone	125°C±20°C
3	Time above 150°C	400 Secs Max
4	Time above 183°C (63 / 37 Tin / Lead Paste Melting Point)	Max-150 sec.
5	Peak Reflow Lead Temperature*	215 ±5°C
6	Cooling rate of lead	Max 4°C / Sec.

Note: Devices classified as moisture sensitive should limit maximum body temperature to 220°C.

unit: mm

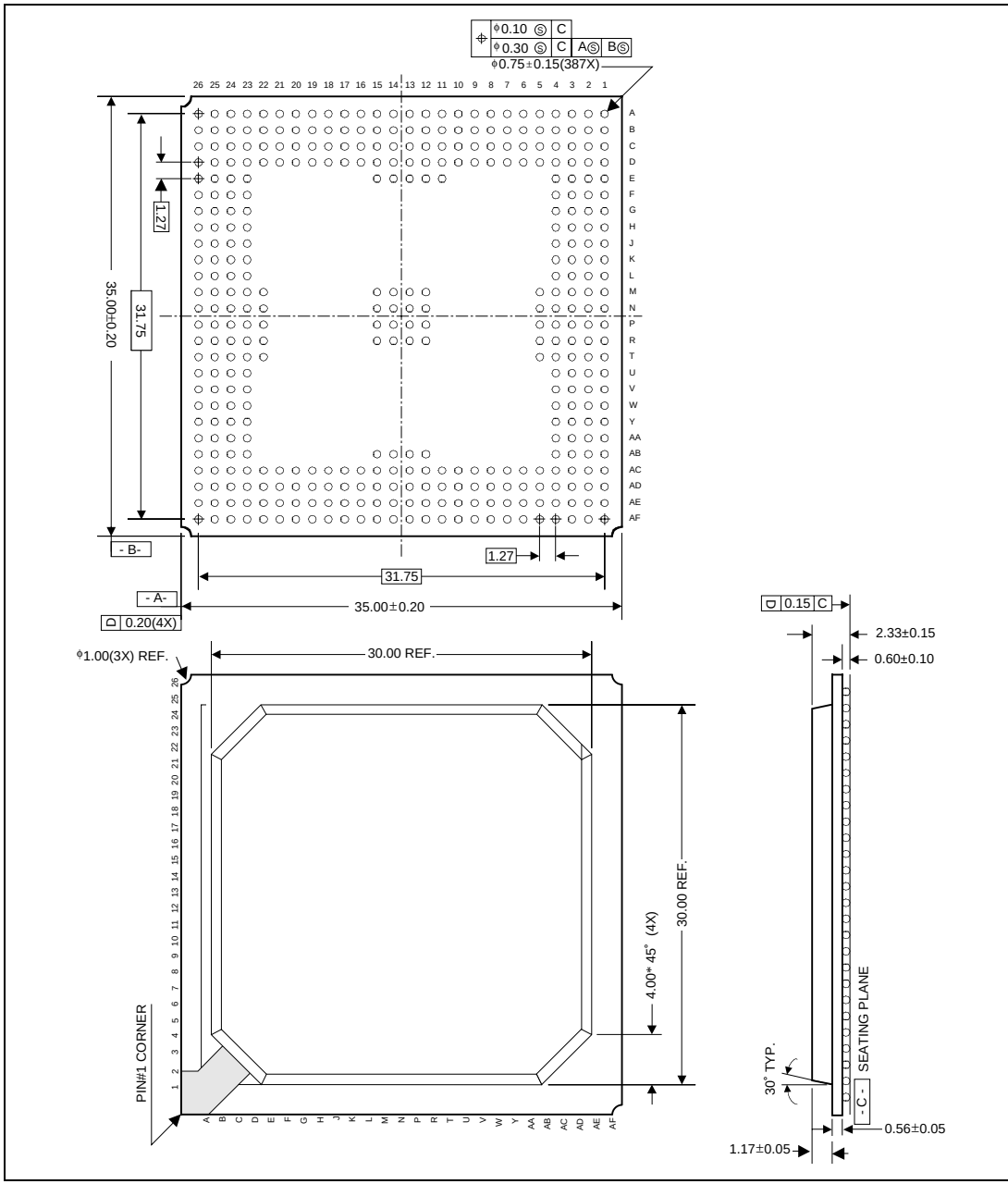


Figure 22.1 HD64465BP Package Dimensions

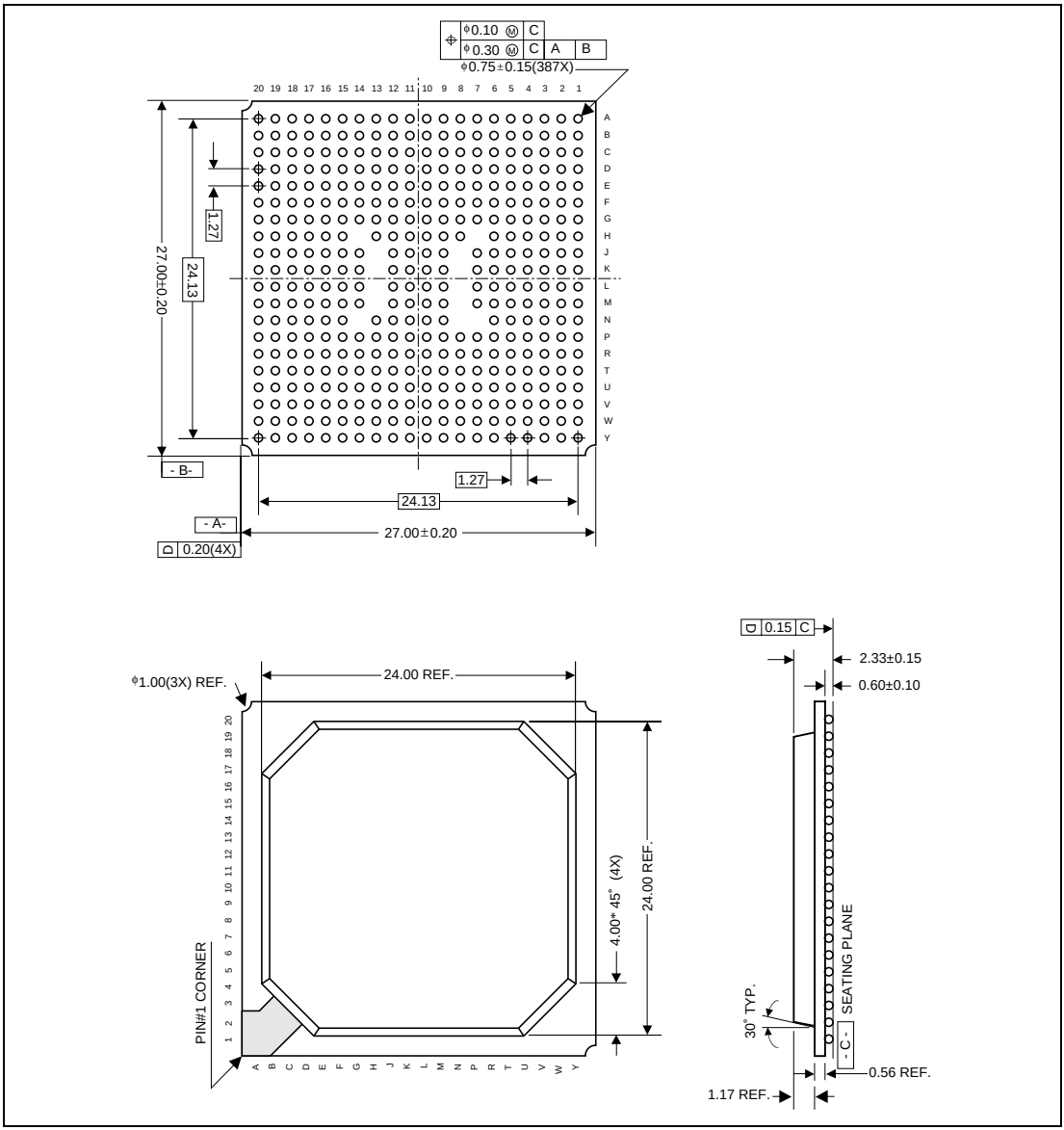


Figure 22.2 HD64465BQ Package Dimensions

Section 23 Ordering Information

Product Type	Mark Code	Package
HD64465	HD64465BP	387-pin BGA (35 mm × 35 mm)
	HD64465BQ	387-pin BGA (27 mm × 27 mm)

Appendix

(1) HD64465 Registers Index

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
Power Management and System Configuration Registers																		
H'10000000	SMSCR	-	PS2ST	-	ADCST	UARTST	-	SCDIST	PPST	-	PC0ST	PC1ST	AFEST	TM0ST	TM1ST	IRDAST	KBCST	H'5BFF
H'10000002	SCONFR	-	-	SLS	HWEN	HW3	HW2	HW1	HW0	-	-	USBCKS	SCDICKS	PPFMS1	PPFMS0	KBWUP	-	H'1100
H'10000004	SBCR	PDOF	PDIG	PCOF	PCIG	PBOF	PBIG	PAOF	PAIG	-	CSPE	CMDPE	ADDRPE	DATAPE	CPUBIG	PEOF	PEIG	H'0000
H'10000006	SPCCR	ADCCLK	-	UARTCLK	PPCLK	FIRCLK	SIRCLK	SCDICLK	KBCCLK	USBCLK	AFECLK	-	-	-	-	UCKOSC	AFEOSC	H'0000
H'10000008	SPSRCR	SPORST	PS2SRT	-	ADCSRT	UARTSRT	-	SCDISRT	PPSRT	USBSRT	PC0SRT	PC1SRT	AFESRT	TM0SRT	TM1SRT	IRDASRT	KBCSRT	H'8000
H'1000000A	SPLLCR	-	-	-	-	-	-	-	-	-	-	PLL2SB	PLL1SB	-	-	PLL2BP	PLL1BP	H'0000
H'1000000C	SRR	MJ7	MJ6	MJ5	MJ4	MJ3	MJ2	MJ1	MJ0	MI7	MI6	MI5	MI4	MI3	MI2	MI1	MI0	H'0101
H'1000000E	STMCR	-	-	-	-	-	-	-	AFETST	PCITST	SDBTST	USBTST	PLL2TST	PLL1TST	URTTST	ACTST	DCTST	H'0000
H'10000010	SDIDR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H'8122
H'10000FF0	SDPCR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	H'0000
PCMCIA Registers																		
H'10002000	PCC0ISR	-	-	-	-	-	-	-	-	P0READY /IREQ0	P0MWP	P0VS2	P0VS1	P0CD2	P0CD1	P0BVD2/ SPKR0	P0BVD1/ STSCHG0	H'00XX
H'10002002	PCC0GCR	-	-	-	-	-	-	-	-	P0DRV	P0PCCR	P0PCCT	P0VCC0	P0MMOD	P0PA25	P0PA24	P0REG	H'0000
H'10002004	PCC0CSCR	-	-	-	-	-	-	-	-	P0SCDI	PSWSEL	P0IREQ	P0SC	P0CDC	P0RC	P0BW	P0BD	H'0020
H'10002006	PCC0CSCIER	-	-	-	-	-	-	-	-	P0CRE	P0IREQE1	P0IREQE0	P0SCE	P0CDE	P0RE	P0BWE	P0BDE	H'0000
H'10002008	PCC0SCR	-	-	-	-	-	-	-	-	-	-	-	SHDN	P0VPP1	P0VPP0	P0VCC1	P0SWP	H'0000
H'1000200A	PCC0PSR	-	-	-	-	-	-	-	-	B_VCC5	B_VCC3	B_VPP_ VCC	B_VPP_ PGM	A_VCC3	A_VCC5	A_VPP_ VCC	A_VPP_ PGM	H'0000
H'10002010	PCC1ISR	-	-	-	-	-	-	-	-	P1READY /IREQ1	P1MWP	P1VS2	P1VS1	P1CD2	P1CD1	P1BVD2/ SPKR1	P1BVD1/ STSCHG1	H'00XX
H'10002012	PCC1GCR	-	-	-	-	-	-	-	-	P1DRV	P1PCCR	P1PCCT	P1VCC0	P1MMOD	P1PA25	P1PA24	P1REG	H'0000
H'10002014	PCC1CSCR	-	-	-	-	-	-	-	-	P1SCDI	-	P1IREQ	P1SC	P1CDC	P1RC	P1BW	P1BD	H'0020
H'10002016	PCC1CSCIER	-	-	-	-	-	-	-	-	P1CRE	P1IREQE1	P1IREQE0	P1SCE	P1CDE	P1RE	P1BWE	P1BDE	H'0000
H'10002018	PCC1SCR	-	-	-	-	-	-	-	-	-	-	-	-	P1VPP1	P1VPP0	P1VCC1	P1SWP	H'0000

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
AFE Interface Registers																		
H'10003200	CTR	HC	DIV2	DIV1	RLY	CNT2	CNT1	TSW	RSW	RDETM	TEIE	REIE	TXIE	RXIE	BUFD	TE	RE	H'0000
H'10003202	STR	TAB	RAB	-	-	-	-	-	-	-	-	-	-	TERR	RERR	TDE	RDF	H'0002
H'10003204	RXDR	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	H'0000
H'10003206	TXDR	TD15	TD14	TD13	TD12	TD11	TD10	TD9	TD8	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	H'0000
H'10003000-H'1000305F	RXDB0,1	RXD15	RXD14	RXD13	RXD12	RXD11	RXD10	RXD9	RXD8	RXD7	RXD6	RXD5	RXD4	RXD3	RXD2	RXD1	RXD0	-
H'10003100-H'1000315F	TXDB0,1	TXD15	TXD14	TXD13	TXD12	TXD11	TXD10	TXD9	TXD8	TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0	-
GPIO Registers																		
H'10004000	GPACR	PA7MD1	PA7MD0	PA6MD1	PA6MD0	PA5MD1	PA5MD0	PA4MD1	PA4MD0	PA3MD1	PA3MD0	PA2MD1	PA2MD0	PA1MD1	PA1MD0	PA0MD1	PA0MD0	H'FFFF
H'10004002	GPBCR	PB7MD1	PB7MD0	PB6MD1	PB6MD0	PB5MD1	PB5MD0	PB4MD1	PB4MD0	PB3MD1	PB3MD0	PB2MD1	PB2MD0	PB1MD1	PB1MD0	PB0MD1	PB0MD0	H'FFFF
H'10004004	GPCCR	PC7MD1	PC7MD0	PC6MD1	PC6MD0	PC5MD1	PC5MD0	PC4MD1	PC4MD0	PC3MD1	PC3MD0	PC2MD1	PC2MD0	PC1MD1	PC1MD0	PC0MD1	PC0MD0	H'FFFF
H'10004006	GPDCR	PD7MD1	PD7MD0	PD6MD1	PD6MD0	PD5MD1	PD5MD0	PD4MD1	PD4MD0	PD3MD1	PD3MD0	PD2MD1	PD2MD0	PD1MD1	PD1MD0	PD0MD1	PD0MD0	H'FFFF
H'10004008	GPECR	PE7MD1	PE7MD0	PE6MD1	PE6MD0	PE5ME1	PE5MD0	PE4MD1	PE4MD0	PE3MD1	PE3MD0	PE2MD1	PE2MD0	PE1MD1	PE1MD0	PE0MD1	PE0MD0	H'FFFF
H'10004010	GPADR	-	-	-	-	-	-	-	-	PA7DT	PA6DT	PA5DT	PA4DT	PA3DT	PA2DT	PA1DT	PA0DT	-
H'10004012	GPBDR	-	-	-	-	-	-	-	-	PB7DT	PB6DT	PB5DT	PB4DT	PB3DT	PB2DT	PB1DT	PB0DT	-
H'10004014	GPCDR	-	-	-	-	-	-	-	-	PC7DT	PC6DT	PC5DT	PC4DT	PC3DT	PC2DT	PC1DT	PC0DT	-
H'10004016	GPDDR	-	-	-	-	-	-	-	-	PD7DT	PD6DT	PD5DT	PD4DT	PD3DT	PD2DT	PD1DT	PD0DT	-
H'10004018	GPEDR	-	-	-	-	-	-	-	-	PE7DT	PE6DT	PE5DT	PE4DT	PE3DT	PE2DT	PE1DT	PE0DT	-
H'10004020	GPAICR	PA7TS	PA6TS	PA5TS	PA4TS	PA3TS	PA2TS	PA1TS	PA0TS	PA7IM	PA6IM	PA5IM	PA4IM	PA3IM	PA2IM	PA1IM	PA0IM	H'00FF
H'10004022	GPBICR	PB7TS	PB6TS	PB5TS	PB4TS	PB3TS	PB2TS	PB1TS	PB0TS	PB7IM	PB6IM	PB5IM	PB4IM	PB3IM	PB2IM	PB1IM	PB0IM	H'00FF
H'10004024	GPCICR	PC7TS	PC6TS	PC5TS	PC4TS	PC3TS	PC2TS	PC1TS	PC0TS	PC7IM	PC6IM	PC5IM	PC4IM	PC3IM	PC2IM	PC1IM	PC0IM	H'00FF
H'10004026	GPDICR	PD7TS	PD6TS	PD5TS	PD4TS	PD3TS	PD2TS	PD1TS	PD0TS	PD7IM	PD6IM	PD5IM	PD4IM	PD3IM	PD2IM	PD1IM	PD0IM	H'00FF
H'10004028	GPEICR	PE7TS	PE6TS	PE5TS	PE4TS	PE3TS	PE2TS	PE1TS	PE0TS	PE7IM	PE6IM	PE5IM	PE4IM	PE3IM	PE2IM	PE1IM	PE0IM	H'00FF
H'10004040	GPAISR	-	-	-	-	-	-	-	-	PA7ISR	PA6ISR	PA5ISR	PA4ISR	PA3ISR	PA2ISR	PA1ISR	PA0ISR	H'0000
H'10004042	GPBISR	-	-	-	-	-	-	-	-	PB7ISR	PB6ISR	PB5ISR	PB4ISR	PB3ISR	PB2ISR	PB1ISR	PB0ISR	H'0000
H'10004044	GPCISR	-	-	-	-	-	-	-	-	PC7ISR	PC6ISR	PC5ISR	PC4ISR	PC3ISR	PC2ISR	PC1ISR	PC0ISR	H'0000
H'10004046	GPDISR	-	-	-	-	-	-	-	-	PD7ISR	PD6ISR	PD5ISR	PD4ISR	PD3ISR	PD2ISR	PD1ISR	PD0ISR	H'0000
H'10004048	GPEISR	-	-	-	-	-	-	-	-	PE7ISR	PE6ISR	PE5ISR	PE4ISR	PE3ISR	PE2ISR	PE1ISR	PE0ISR	H'0000
Interrupt Controller Registers																		
H'10005000	NIRR	PS2KBR	PCC0R	PCC1R	AFER	GPIOR	TMU0R	TMU1R	KBCR	PS2MSR	IRDAR	UARTR	-	PPR	SCDIR	USBR	ADCR	-
H'10005002	NIMR	PS2KBM	PCC0M	PCC1M	AFEM	GPIOM	TMU0M	TMU1M	KBCM	PS2MSM	IRDAM	UARTM	-	PPM	SCDIM	USBM	ADCM	H'0000
H'10005004	NITR	PS2KBT	PCC0T	PCC1T	AFET	GPIOT	TMU0T	TMU1T	KBCT	PS2MST	IRDAT	UARTT	-	PPT	SCDIT	USBT	ADCT	H'0000

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
Timer Registers																		
H'10006000	TCVR1	TCV115	TCV114	TCV113	TCV112	TCV111	TCV110	TCV19	TCV18	TCV17	TCV16	TCV15	TCV14	TCV13	TCV12	TCV11	TCV10	H'FFFF
H'10006002	TCVR0	TCV015	TCV014	TCV013	TCV012	TCV011	TCV010	TCV09	TCV08	TCV07	TCV06	TCV05	TCV04	TCV03	TCV02	TCV01	TCV00	H'FFFF
H'10006004	TRVR1	TRV115	TRV114	TRV113	TRV112	TRV111	TRV110	TRV19	TRV18	TRV17	TRV16	TRV15	TRV14	TRV13	TRV12	TRV11	TRV10	H'0002
H'10006006	TRVR0	TRV015	TRV014	TRV013	TRV012	TRV011	TRV010	TRV09	TRV08	TRV07	TRV06	TRV05	TRV04	TRV03	TRV02	TRV01	TRV00	H'0002
H'10006008	TCR1	-	-	-	-	-	-	-	-	-	-	-	EDMA	ETMO1	PST11	PST10	T1STP	H'0000
H'1000600A	TCR0	-	-	-	-	-	-	-	-	-	-	-	EADT	ETMO0	PST01	PST00	T0STP	H'0000
H'1000600C	TIRR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TMU1R	TMU0R	H'0000
H'1000600E	TIDR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	TMU1D	TMU0D	H'0000
H'10006010	PWM1CS	-	-	-	-	-	-	-	-	-	-	P1CS5	P1CS4	P1CS3	P1CS2	P1CS1	P1CS0	H'0000
H'10006012	PWM1LPC	P1LC15	P1LC14	P1LC13	P1LC12	P1LC11	P1LC10	P1LC9	P1LC8	P1LC7	P1LC6	P1LC5	P1LC4	P1LC3	P1LC2	P1LC1	P1LC0	H'FFFF
H'10006014	PWM1HPC	P1HC15	P1HC14	P1HC13	P1HC12	P1HC11	P1HC10	P1HC9	P1HC8	P1HC7	P1HC6	P1HC5	P1HC4	P1HC3	P1HC2	P1HC1	P1HC0	H'FFFF
H'10006018	PWM0CS	-	-	-	-	-	-	-	-	-	-	P0CS5	P0CS4	P0CS3	P0CS2	P0CS1	P0CS0	H'0000
H'1000601A	PWM0LPC	P0LC15	P0LC14	P0LC13	P0LC12	P0LC11	P0LC10	P0LC9	P0LC8	P0LC7	P0LC6	P0LC5	P0LC4	P0LC3	P0LC2	P0LC1	P0LC0	H'FFFF
H'1000601C	PWM0HPC	P0HC15	P0HC14	P0HC13	P0HC12	P0HC11	P0HC10	P0HC9	P0HC8	P0HC7	P0HC6	P0HC5	P0HC4	P0HC3	P0HC2	P0HC1	P0HC0	H'FFFF
IrDA Registers																		
H'10007000	IrRBR	-	-	-	-	-	-	-	-	RBR7	RBR6	RBR5	RBR4	RBR3	RBR2	RBR1	RBR0	-
H'10007000	IrTBR	-	-	-	-	-	-	-	-	TBR7	TBR6	TBR5	TBR4	TBR3	TBR2	TBR1	TBR0	-
H'10007002	IrIER	-	-	-	-	-	-	-	-	-	-	-	-	EMSI	ERLSI	ETHREI	ERDVI	-
H'10007004	IrIIR	-	-	-	-	-	-	-	-	IIR7	IIR6	-	-	IIR3	IIR2	IIR1	IIR0	H'0001
H'10007004	IrFCR	-	-	-	-	-	-	-	-	FCR7	FCR6	-	-	-	XFRST	RFRST	FEN	H'0000
H'10007006	IrLCR	-	-	-	-	-	-	-	-	DLAB	BREAK	SP	EPS	PEN	STB	WLS1	WLS0	H'0000
H'10007008	IrMCR	-	-	-	-	-	-	-	-	-	-	-	LOOP	OUT2	OUT1	RTS	DTR	H'0000
H'10007000	IrDLL	-	-	-	-	-	-	-	-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	H'0000
H'10007002	IrDLM	-	-	-	-	-	-	-	-	Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	H'0000
H'1000700A	IrLSR	-	-	-	-	-	-	-	-	ERF	TEMT	THRE	BI	FE	PE	OE	DR	H'0060
H'1000700C	IrMSR	-	-	-	-	-	-	-	-	DCD#	RI#	DSR#	CTS#	DDCD	TERI	DDSR	DCTS	H'0000
H'1000700E	IrSCR	-	-	-	-	-	-	-	-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	-
H'10007100	IMSTCR	-	-	-	-	-	-	-	-	IEN	TXEN	RXEN	BKSEL4	BKSEL3	BKSEL2	BKSEL1	BKSEL0	H'0000
H'10007102	IMSTSR	-	-	-	-	-	-	-	-	-	TMI	TXI	RXI	IID2	IID1	IID0	-	H'0000
H'10007102	IMISCR	-	-	-	-	-	-	-	-	DCS1	DCS0	-	ILOOP	-	-	-	-	H'0000
H'10007104	IRFR	-	-	-	-	-	-	-	-	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	-
H'10007104	ITFR	-	-	-	-	-	-	-	-	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	-
H'10007106	ITC1R	-	-	-	-	-	-	-	-	RTS	TFRIEN	TFUIEN	TFTL	ADRTS	ACEOM	TIDL	UA	H'0000
H'10007108	ITC2R	-	-	-	-	-	-	-	-	SB	ACRCG	SIP1	SIP0	NSFP	EEIL2	EEIL1	EEIL0	H'0040

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
H'1000710A	ITSR	-	-	-	-	-	-	-	-	-	-	-	-	TFUR	EOM	TFRDY	EEOM	H'0002
H'1000710C	IRCR	-	-	-	-	-	-	-	-	RFTL	ACRCC	RADM1	RADM0	SYNIEN	-	RFRIEN	SCIEN	H'0040
H'1000710E	IRSR	-	-	-	-	-	-	-	-	ABORT	CRCER	RFOVF	EOF	RFEM	SYNC	-	-	H'0004
H'1000710E	IRSTCR	-	-	-	-	-	-	-	-	RSTC3	RSTC2	RSTC1	RSTC0	-	-	-	-	H'0000
H'10007102	IFAR	-	-	-	-	-	-	-	-	RFA7	RFA6	RFA5	RFA4	RFA3	RFA2	RFA1	-	H'0000
H'10007104	IRBCLR	-	-	-	-	-	-	-	-	RBC7	RBC6	RBC5	RBC4	RBC3	RBC2	RBC1	RBC0	H'0000
H'10007106	IRBCHR	-	-	-	-	-	-	-	-	-	-	-	RBC12	RBC11	RBC10	RBC9	RBC8	H'0000
H'10007108	IRRFPLR	-	-	-	-	-	-	-	-	RFP7	RFP6	RFP5	RFP4	RFP3	RFP2	RFP1	RFP0	H'0000
H'1000710A	IRRFPHR	-	-	-	-	-	-	-	-	RFP15	RFP14	RFP13	RFP12	RFP11	RFP10	RFP9	RFP8	H'0000
H'1000710C	ITBCLR	-	-	-	-	-	-	-	-	TBC7	TBC6	TBC5	TBC4	TBC3	TBC2	TBC1	TBC0	H'0000
H'1000710E	ITBCHR	-	-	-	-	-	-	-	-	-	-	-	TBC12	TBC11	TBC10	TBC9	TBC8	H'00F0
H'10007102	IIRC1R	-	-	-	-	-	-	-	-	IRSPD3	IRSPD2	IRSPD1	IRSPD0	IRMOD3	IRMOD2	IRMOD1	IRMOD0	H'0000
H'10007104	IIRTCR	-	-	-	-	-	-	-	-	-	-	DFREQ	MODSEL	ECHO	-	TXDF	-	H'0001
H'10007106	IIRC2R	-	-	-	-	-	-	-	-	ACEN	-	-	-	CCTRL1	CCTRL0	DSIRI	DFIRI	H'0000
H'10007108	ITMR	-	-	-	-	-	-	-	-	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0	H'0000
H'1000710A	IIRC3R	-	-	-	-	-	-	-	-	SCDIEN	SCD	-	-	-	-	TMIEN	TMI	H'0000
H'10007110	DMARP	-	-	-	-	-	-	-	-	DMAR7	DMAR6	DMAR5	DMAR4	DMAR3	DMAR2	DMAR1	DMAR0	-
H'10007110	DMAWP	-	-	-	-	-	-	-	-	DMAW7	DMAW6	DMAW5	DMAW4	DMAW3	DMAW2	DMAW1	DMAW0	-
H'10007120	ISIRR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SLOOP	SIRMOD	H'0001
H'100071E0	IFIRCR	-	-	-	-	-	-	-	-	-	-	-	-	-	RX2_PP	RX_PP	TMODE	H'0000
H'100071F0	ITMCR	-	-	-	-	-	-	-	-	-	-	-	-	-	TMCR2	TMCR1	TMCR0	H'0006
UART Registers																		
H'10008000	URBR	-	-	-	-	-	-	-	-	URBR7	URBR6	URBR5	URBR4	URBR3	URBR2	URBR1	URBR0	-
H'10008000	UTBR	-	-	-	-	-	-	-	-	UTBR7	UTBR6	UTBR5	UTBR4	UTBR3	UTBR2	UTBR1	UTBR0	-
H'10008002	UIER	-	-	-	-	-	-	-	-	-	-	-	-	EMSI	ERLSI	ETHREI	ERDVI	-
H'10008004	UIIR	-	-	-	-	-	-	-	-	UIIR7	UIIR6	-	-	UIIR3	UIIR2	UIIR1	UIIR0	H'0001
H'10008004	UFCR	-	-	-	-	-	-	-	-	UFCR7	UFCR6	-	-	-	XFRST	RFRST	FEN	H'0000
H'10008006	ULCR	-	-	-	-	-	-	-	-	DLAB	BREAK	SP	EPS	PEN	STB	WLS1	WLS0	H'0000
H'10008008	UMCR	-	-	-	-	-	-	-	-	-	-	-	LOOP	OUT2	-	RTS	DTR	H'0000
H'10008000	UDLL	-	-	-	-	-	-	-	-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	H'0000
H'10008002	UDLM	-	-	-	-	-	-	-	-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	H'0000
H'1000800A	ULSR	-	-	-	-	-	-	-	-	ERF	TEMT	THRE	BI	FE	PE	OE	DR	H'0060
H'1000800C	UMSR	-	-	-	-	-	-	-	-	DCD#	RI#	DSR#	CTS#	DDCD	TERI	DDSR	DCTS	H'0000
H'1000800E	USCR	-	-	-	-	-	-	-	-	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	-

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
Parallel Port Registers/SPP & EPP MODES																		
H'1000A000	Data Port	-	-	-	-	-	-	-	-	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	-
H'1000A002	Status Port	-	-	-	-	-	-	-	-	BUSY#	ACK#	PE	SLCT	ERR#	1	1	TMOUT	-
H'1000A004	Control Port	-	-	-	-	-	-	-	-	1	1	PPDIR	IRQE	SLIN	INIT	AFD	STB	H'00CX
H'1000A006	EPP Addr. Port	-	-	-	-	-	-	-	-	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	-
H'1000A008	EPP Data Port0	-	-	-	-	-	-	-	-	P0D7	P0D6	P0D5	P0D4	P0D3	P0D2	P0D1	P0D0	-
H'1000A00A	EPP Data Port1	-	-	-	-	-	-	-	-	P1D7	P1D6	P1D5	P1D4	P1D3	P1D2	P1D1	P1D0	-
H'1000A00C	EPP Data Port2	-	-	-	-	-	-	-	-	P2D7	P2D6	P2D5	P2D4	P2D3	P2D2	P2D1	P2D0	-
H'1000A00E	EPP Data Port3	-	-	-	-	-	-	-	-	P3D7	P3D6	P3D5	P3D4	P3D3	P3D2	P3D1	P3D0	-
Parallel Port Registers/ECP MODE																		
H'1000A000	data	-	-	-	-	-	-	-	-	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	-
H'1000A000	ecpAFifo	-	-	-	-	-	-	-	-	Addr/RLE	Address or RLE field							H'0000
H'1000A002	dsr	-	-	-	-	-	-	-	-	nBusy	nAck	PError	Select	nFault	1	1	1	-
H'1000A004	dcr	-	-	-	-	-	-	-	-	1	1	PPDIR	IRQE	SelectIn	nInit	AutoFd	Strobe	H'00CX
H'1000A010	cFifo	-	-	-	-	-	-	-	-	Parallel Port Data FIFO								-
H'1000A010	ecpDFifo	-	-	-	-	-	-	-	-	ECP Data FIFO								-
H'1000A010	tFifo	-	-	-	-	-	-	-	-	Test FIFO								-
H'1000A010	cnfgA	-	-	-	-	-	-	-	-	0	0	0	1	0	0	0	0	H'0010
H'1000A012	cnfgB	-	-	-	-	-	-	-	-	Compress	-	-	-	-	-	-	-	-
H'1000A014	ecr	-	-	-	-	-	-	-	-	mode			nErrIntEn	-	Service Intr	Full	Empty	H'0015
USB Host Controller Registers																		
H'1000B000	HR	Revision																H'0110
H'1000B002	HR	-																H'0000
H'1000B004	HC	-					RWCE	RWC	-	HCFS		BLE	CLE	IE	PLE	CBSR		H'0000
H'1000B006	HC	-																H'0000
H'1000B008	HCS	-													BLF	CLF	HCR	H'0000
H'1000B00A	HCS	-														SOC		H'0000
H'1000B00C	HIS	-									RHSC	FNO	UE	RD	SOF	WDH	SO	H'0000
H'1000B00E	HIS	-		-														H'000D
H'1000B010	HIE	-									RHSCE	FNOE	UEE	RDE	SOFE	WDHE	SOE	H'0000
H'1000B012	HIE	MIE	-	-														H'000D
H'1000B014	HID	-									RHSCE	FNOE	UEE	RDE	SOFE	WDHE	SOE	H'0000
H'1000B016	HID	MIE	-	-														H'000D
H'1000B018	HHCCA	HCCA[15:8]								-								H'0000

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
H'1000B01A	HHCCA	HCCA[31:16]																H'0000
H'1000B01C	HPCED	PCED[15:4]												-				H'0000
H'1000B01E	HPCED	PCED[31:16]																H'0000
H'1000B020	HCHED	CHED[15:4]												-				H'0000
H'1000B022	HCHED	CHED[31:16]																H'0000
H'1000B024	HCCED	CCED[15:4]												-				H'0000
H'1000B026	HCCED	CCED[31:16]																H'0000
H'1000B028	HBHED	BHED[15:4]												-				H'0000
H'1000B030	HBHED	BHED[31:16]																H'0000
H'1000B018	HHCCA	HCCA[15:8]								-								H'0000
H'1000B01A	HHCCA	HCCA[31:16]																H'0000
H'1000B01C	HPCED	PCED[15:4]												-				H'0000
H'1000B01E	HPCED	PCED[31:16]																H'0000
H'1000B020	HCHED	CHED[15:4]												-				H'0000
H'1000B022	HCHED	CHED[31:16]																H'0000
H'1000B024	HCCED	CCED[15:4]												-				H'0000
H'1000B026	HCCED	CCED[31:16]																H'0000
H'1000B028	HBHED	BHED[15:4]												-				H'0000
H'1000B02A	HBHED	BHED[31:16]																H'0000
H'1000B02C	HBCED	BCED[15:4]												-				H'0000
H'1000B02E	HBCED	BCED[31:16]																H'0000
H'1000B030	HDH	DH[15:4]												-				H'0000
H'1000B032	HDH	DH[31:16]																H'0000
H'1000B034	HFI	-		FI														H'2EDF
H'1000B036	HFI	FIT	FSLDP															H'0000
H'1000B038	HFR	-		FR														-
H'1000B03A	HFR	FRT	-															H'0000
H'1000B03C	HFN	FN																H'0000
H'1000B03E	HFN	-																H'0000
H'1000B040	HPS	-		PS														H'000D
H'1000B042	HPS	-																H'0000
H'1000B044	HLST	-				LST												H'0000
H'1000B046	HLST	-																H'0000
H'1000B048	HRDA	POTPGT								-								H'0100
H'1000B04A	HRDA	-			NOCP	OCPM	DT	NPS	PSM	NDP								H'0002

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
H'1000B04C	HRDB	DR																H'0000
H'1000B04E	HRDB	PPCM																H'0000
H'1000B050	HRS	DRWE	-													OCI	LPS	H'0000
H'1000B052	HRS	CRWE	-													OCIC	LPSC	H'0000
H'1000B054	HRPS1	-						LSDA	PPS	-			PRS	POCI	PSS	PES	CCS	H'0000
H'1000B056	HRPS1	-											PRSC	POCIC	PSSC	PESC	CSC	H'0000
H'1000B058	HRPS2	-						LSDA	PPS	-			PRS	POCI	PSS	PES	CCS	H'0000
H'1000B05A	HRPS2	-											PRSC	POCIC	PSSC	PESC	CSC	H'0000
Audio Codec Interface Registers																		
H'1000C000	TDR	TD15	TD14	TD13	TD12	TD11	TD10	TD9	TD8	TD7	TD6	TD5	TD4	TD3	TD2	TD1	TD0	H'XXXX
H'1000C002	TDR	TD31	TD30	TD29	TD28	TD27	TD26	TD25	TD24	TD23	TD22	TD21	TD20	TD19	TD18	TD17	TD16	H'XXXX
H'1000C004	RDR	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	H'XXXX
H'1000C006	RDR	RD31	RD30	RD29	RD28	RD27	RD26	RD25	RD24	RD23	RD22	RD21	RD20	RD19	RD18	RD17	RD16	H'XXXX
H'1000C008	CR	-	-	DMAEN	SL18	CDRT	WMRT	AC97S	SWR	PU	MS	ST	CRE	FTF	TXEN	FRF	RXEN	H'0000
H'1000C00C	SR	-	IR71	TNF	TFS1	TFS0	TFU	TFO	TDI	-	-	RNE	RFS1	RFS0	RFU	RFO	RDI	H'2000
H'1000C010	FSR	-	-	-	-	-	-	-	-	-	-	-	-	-	FS2	FS1	FS0	H'0000
H'1000C014	IER	-	-	-	-	-	-	-	-	-	IR71E	TFUIE	TFOIE	TDIE	RFUIE	RFOIE	RDIE	H'0000
H'1000C020	CSAR	CA3/SA3	CA2/SA2	CA1/SA1	CA0SA0	-	-	-	-	-	-	-	-	-	-	-	-	H'0000
H'1000C022	CSAR	-	-	-	-	-	-	-	-	-	-	-	-	RW	CA6/SA6	CA5/SA5	CA4/SA4	H'0000
H'1000C024	CDR	CD11/SD11	CD10/SD10	CD9/SD9	CD8/SD8	CD7/SD7	CD6/SD6	CD5/SD5	CD4/SD4	CD3/SD3	CD2/SD2	CD1/SD1	CD0/SD0	-	-	-	-	H'0000
H'1000C026	CDR	-	-	-	-	-	-	-	-	-	-	-	-	CD15/SD15	CD14/SD14	CD13/SD13	CD12/SD12	H'0000
H'1000C028	PCML	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C02A	PCML	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C02C	PCMR	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C02E	PCMR	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C030	L1R	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C032	L1R	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C034	PCMC	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C036	PCMC	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C038	PCMLS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C03A	PCMLS	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C03C	PCMRS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C03E	PCMRS	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C040	PCMLFE	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C042	PCMLFE	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
H'1000C044	L2R	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C046	L2R	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C048	HSET	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C04A	HSET	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C04C	IOCS	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	H'0000
H'1000C04E	IOCS	-	-	-	-	-	-	-	-	-	-	-	-	D19	D18	D17	D16	H'0000
H'1000C050	ATIER	PLSTFOVIE	PRSTFOVIE	PLFETFOVIE	L2TFOVIE	HTTFOVIE	IOCTFOVIE	PLTFUNIE	PRTFUNIE	L1TFUNIE	PCTFUNIE	PLSTFUNIE	PRSTFUNIE	PLFETFUNIE	L2TFUNIE	HTTFUNIE	IOCTFUNIE	H'0000
H'1000C052	ATIER	-	-	PLTFRQIE	PRTFRQIE	L1TFRQIE	PCTFRQIE	PLSTFRQIE	PRSTFRQIE	PLFETFRQIE	L2TFRQIE	HTTFRQIE	IOCTFRQIE	PLTFOVIE	PRTFOVIE	L1TFOVIE	PCTFOVIE	H'0000
H'1000C054	ATSR	PLSTFOV	PRSTFOV	PLFETFOV	L2TFOV	HTTFOV	IOCTFOV	PLTFUN	PRTFUN	L1TFUN	PCTFUN	PLSTFUN	PRSTFUN	PLFETFUN	L2TFUN	HTTFUN	IOCTFUN	H'0000
H'1000C056	ATSR	-	-	PLTFRQ	PRTFRQ	L1TFRQ	PCTFRQ	PLSTFRQ	PRSTFRQ	PLFETFRQ	L2TFRQ	HTTFRQ	IOCTFRQ	PLTFOV	PRTFOV	L1TFOV	PCTFOV	H'0000
H'1000C058	ARIER	HTRFRQIE	IOCSRFRQIE	PLRFOVIE	PRRFOVIE	L1RFOVIE	MICRFOVIE	L2RFOVIE	HTRFOVIE	IOCSRFOVIE	PLRFUNIE	PRRFUNIE	L1RFUNIE	MICRFUNIE	L2RFUNIE	HTRFUNIE	IOCSRFUNIE	H'0000
H'1000C05A	ARIER	-	-	-	-	-	-	-	-	-	STARYIE	STDRYIE	PLRFRQIE	PRRFRQIE	L1RFRQIE	MICRFRQIE	L2RFRQIE	H'0000
H'1000C05C	ARSR	HTRFRQ	IOCSRFRQ	PLRFOV	PRRFOV	L1RFOV	MICRFOV	L2RFOV	HTRFOV	IOCSRFOV	PLRFUN	PRRFUN	L1RFUN	MICRFUN	L2RFUN	HTRFUN	IOCSRFUN	H'0000
H'1000C05E	ARSR	-	-	-	-	-	-	-	-	-	STARY	STDRY	PLRFRQ	PRRFRQ	L1RFRQ	MICRFRQ	L2RFRQ	H'0000
H'1000C060	ACR	FPRTF	FL1TF	FPCTF	FPLSTF	FPRSTF	FPLETF	FL2TF	FHTF	FIOCTF	FPLRF	FPRRF	FL1RF	FMRF	FL2RF	FHRF	FIOSRF	H'0000
H'1000C062	ACR	VS	-	-	-	-	-	-	-	-	RXDMA_EN	TXDMA_EN	FCAF	FCDF	FSTAF	FSTDF	FPLTF	H'0000
H'1000C064	ATAGR	-	-	-	TXVS3	TXVS4	TXVS5	TXVS6	TXVS7	TXVS8	TXVS9	TXVS10	TXVS11	TXVS12	-	-	-	H'000X
H'1000C066	ATAGR	CR	RXVS1	RXVS2	RXVS3	RXVS4	RXVS5	RXVS6	RXVS7	RXVS8	RXVS9	RXVS10	RXVS11	RXVS12	-	-	-	H'000X
H'1000C068	SRAR	-	-	-	SL12RA	SL11RA	SL10RA	SL9RA	SL8RA	SL7RA	SL6RA	SL5RA	SL4RA	SL3RA	-	-	-	H'0000
KBC Registers																		
H'1000D800	CR	-	-	-	-	-	-	-	-	-	-	KCRTS1	KCRTS0	IRQ1TS1	IRQ1TS0	IRQ0TS1	IRQ0TS0	H'0000
H'1000D802	SR	-	-	-	-	-	-	-	-	-	-	-	-	-	KCR	KCIRQ1	KCIRQ0	H'0000
PS/2 Interface Registers																		
H'1000DC00	KBCSR	KBCIE	KBCOE	KBDOE	KBCD	KBDD	KBCS	KBDS	KBDP	KBD7	KBD6	KBD5	KBD4	KBD3	KBD2	KBD1	KBD0	H'0000
H'1000DC04	KBISR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	KBRDR	H'0000
H'1000DC10	MSCSR	MSCIE	MSCOE	MSDOE	MSCD	MSDD	MSCS	MSDS	MSDP	MSD7	MSD6	MSD5	MSD4	MSD3	MSD2	MSD1	MSD0	H'0000
H'1000DC14	MSISR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	MSRDF	H'0000

Reg. Add.	Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Default
A/D Converter Registers																		
H'1000E000	ADDRA	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	H'0000
H'1000E002	ADDRB	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	H'0000
H'1000E004	ADDRC	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	H'0000
H'1000E006	ADDRD	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	H'0000
H'1000E008	ADCSR	0	0	0	0	0	0	0	0	ADF	ADST	ADIS	TRGE	ADIE	SCAN	CH1	CH0	H'0000
H'1000E00A	ADCALCR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CAL	H'0000
H'1000E00C	ADCAL	0	0	0	0	0	0	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	H'0000

(2) Caution to AFE Interface

This module is the interface for the RSA Communication Inc. software modem.

(3) Standby mode

HD64465 supports several method to reduce power consumption.

Please select properly setting regarding your system design.

- Module standby mode

  Set SMSCR for each functional module except USB-HOST to enter module standby.

  Set HC Control Register to let USB-HOST enter module standby.

  To stop internal bus for USB-HOST , use Clock Gating method.
- Clock Gating

  Set SPCCR to gate clock or to stop oscilation.
- PLL stop

  Set SPLLCR to disable PLL.
- I/O setting

  Set SBCR properly to match your system design. Also check Pin Status List at each state(Power On Reset / Active / Standby).

(4) Others

PCB layout information can be prepared as IBIS model file and Boundary Scan Description Language file by Hitachi Ltd. without any kind of guarantee to work correctly.

(5) HD64465 PIN status list

Pin Name	I/O Type	Sink (mA)	POR	STB	Source	POR	STB	Destination	POR	STB	Sys. Req.	Function
TST	I		I	-	SYSTEM BOARD	D	D				D	Test Mode
TMS	I		U	-	SYSTEM BOARD	-	-					
TCK	I		U	-	SYSTEM BOARD	-	-					
TRST#	I		U	-	SYSTEM BOARD	-	-					
TDI	I		U	-	SYSTEM BOARD	-	-					
TDO	O	4	Z					SYSTEM BOARD	-	-		
CKIO	I		I	I	SH4 CKIO	O	O					CPU Interface
A25-A0	I		U	d	SH4 A[25:0]	O	Z					(VCC)
D15-D0	IO	8	I	d	SH4 D[15:0]	Z	Z	SH4 D[15:0]	Z	Z	U	(1: the function and pull up/down are decided by SH_MODE.)
CS4#	I		I	U	SH4 CS4#	H	Z					
WE0#	I		I	U	SH4 WE0#	H	Z					
WE1#/WE#	I		I	U	SH4 WE1#	H	Z					
WE2#/ICIORD#	I		I	U	SH4 WE2#	H	Z					(2: the pull up or down is decided by which CPU Interface is supported.)
WE3#/ICIOWR#	I		I	U	SH4 WE3#	H	Z					
RDWR#	I		I	U	SH4 RDWR#	H	Z					
RD#	I		I	U	SH4 RD#	H	Z					
RDY#/WAIT#[1]	O	4	Z	Z				SH4 RDY#	Z	Z	D/U	
BS#	I		I	I	SH4 BS#	H	Z				U	
DREQ[1:0]#	O	4	O	O				SH4 DREQ[1:0]#	I	I		
DRAK[1:0]	I		I	d	SH4 DRAK[1:0]	O	O					
RESET#	I		I	I	SYSTEM BOARD	O	O				U	
IRQ0#	O	4	H	H				SH4 INTR#	I	I		
SH_MODE[2]	I		I	I							D/U	
CE1B#/CS6#	I		d	d	SH4 CE1B#	H	Z					
CE2B#	I		d	d	SH4 CE2B#	I	Z					
CE1A#/CS5#	I		d	d	SH4 CE1A#	H	Z					
CE2A#	I		d	d	SH4 CE2A#	I	Z					
IOIS16#	O	4	Z	Z				SH4 IOIS16#	Z	Z	U	
PCC0CE1B#	O	8	Z	Z				PCC0 CE1#	U	U		PCMCIA0
PCC0CE2B#	O	8	Z	Z				PCC0 CE2#	U	U		(VCCA)
RDB#	O	8	Z	Z				PCC0 OE#	U	U		(3: the power of these pin is VCC)
WEB#	O	8	Z	Z				PCC0 WE#/PGM#	U	U		
PCC0ICIORDB#	O	8	Z	Z				PCC0 IORD#	U	U		
PCC0ICIOWRB#	O	8	Z	Z				PCC0 IOWR#	U	U		
PCC0RESET	O	8	Z	Z				PCC0 RESET	U	U		
PCC0WAIT#	I		d	d	PCC0 WAIT	Z	Z				U	
PCC0WP#/IOIS16#	I		d	d	PCC0 WP(IOIS16#)	Z	Z				U	

Pin Name	I/O Type	Sink (mA)	POR	STB	Source	POR	STB	Destination	POR	STB	Sys. Req.	Function
PCC0RDY/IRQ0#	I		d	d	PCC0 RDY(IREQ#)	Z	Z				U	
PCC0BVD1/STSCHG0#	I		d	d	PCC0 BVD1(STSCHG#)	Z	Z				U	
PCC0BVD2/SPKR0	I		d	d	PCC0 BVD2 (SPKR#)	Z	Z				U	
PCC0CD1#[3]	I		I	I	PCC0 CD1#	-	-				U	
PCC0CD2#[3]	I		I	I	PCC0 CD2#	-	-				U	
PCC0VS1#[3]	I		I	I	PCC0 VS1#	-	-				U	
PCC0VS2#[3]	I		I	I	PCC0 VS2#	-	-				U	
PCC0A25-A0	O	8	Z	Z				PCC0 A25-A0	D	D		
PCC0D15-D0	IO	8	d	d	PCC0 D15-D0	D	D	PCC0 D15-D0	D	D		
PCC0REG#	O	8	Z	Z				PCC0 REG#	U	U		
VCC0SEL0/DATA[3]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
VCC0SEL1/CLOCK[3]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
VCC0VPP0[3]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
VCC0VPP1/LATCH[3]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
PCC1CE1A#	O	8	Z	Z				PCC1 CE1#	U	U		PCMCIA1
PCC1CE2A#	O	8	Z	Z				PCC1 CE2#	U	U		(VCCB)
RDA#	O	8	Z	Z				PCC1 OE#	U	U		(4: the power of these pin is VCC)
WEA#	O	8	Z	Z				PCC1 WE#/PGM#	U	U		
PCC1ICIORDA#	O	8	Z	Z				PCC1 IORD#	U	U		
PCC1ICIOWRA#	O	8	Z	Z				PCC1 IOWR#	U	U		
PCC1RESET	O	8	Z	Z				PCC1 RESET	U	U		
PCC1WAIT#	I		d	d	PCC1 WAIT#	Z	Z				U	
PCC1WP#/IOIS16#	I		d	d	PCC1 WP(IOIS16#)	Z	Z				U	
PCC1RDY/IRQ1#	I		d	d	PCC1 RDY(IREQ#)	Z	Z				U	
PCC1BVD1/STSCHG1#	I		d	d	PCC1 BVD1(STSCHG#)	Z	Z				U	
PCC1BVD2/SPKR1	I		d	d	PCC1 BVD2 (SPKR#)	Z	Z				U	
PCC1CD1#[4]	I		I	I	PCC1 CD1#	-	-				U	
PCC1CD2#[4]	I		I	I	PCC1 CD2#	-	-				U	
PCC1VS1#[4]	I		I	I	PCC1 VS1#	-	-				U	
PCC1VS2#[4]	I		I	I	PCC1 VS2#	-	-				U	
PCC1A25-A0	O	8	Z	Z				PCC1 A25-A0	D	D		
PCC1D15-D0	IO	8	d	d	PCC1 D15-D0	D	D	PCC1 D15-D0	D	D		
PCC1REG#	O	8	Z	Z				PCC1 REG#	U	U		

Pin Name	I/O Type	Sink (mA)	POR	STB	Source	POR	STB	Destination	POR	STB	Sys. Req.	Function
VCC1SEL0[4]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
VCC1SEL1[4]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
VCC1VPP0[4]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
VCC1VPP1[4]	O	4	L	L				POWER SWITCH CONTROLLER	I	I		
TXD0	O	4	O	O				MAX3243				UART 0
RXD0	I		d	d	MAX3243	O	O					(VCC)
RTS0#	O	4	O	O				MAX3243	I	I		
CTS0#	I		d	d	MAX3243	O	O					
DTR0#	O	4	O	O				MAX3243	I	I		
DSR0#	I		d	d	MAX3243	O	O					
DCD0#	I		d	d	MAX3243	O	O					
RI0#	I		I	I	MAX3243	O	O					
MODSEL/RX2#	O/I		d	C/c	IR MODULE	H	H	IR MODULE	I	I		IrDA (VCC)
TXD	O	4	L	L				IR MODULE	I	I		
RX#	I		d	d	IR MODULE	H	H					
STB#	O	24	Z	Z				PP nStorbe	I	I	U	PARALLEL PORT (VCC5)
AFD#	O	24	Z	Z				PP nAutofd	I	I	U	
ERR#	I		d	d	PP nError	O	O				U	
INIT#	O	24	Z	Z				PP nInit	I	I	U	
SLIN#	O	24	Z	Z				PP nSelectin	I	I	U	
ACK#	I		d	d	PP nACK	O	O				U	
BUSY	I		d	d	PP Busy	O	O				U	
PE	I		d	d	PP PE	O	O				U	
SLCT	I		d	d	PP Select	O	O				U	
PPD7-PPD0	IO	24	d	d	PRINTER	O	O				U	
DIN	I		d	d	STLC 7546	Z	Z					AFE Interface
DOUT	O	4	Z	Z				STLC 7546	d	d		(VCC)
SCLK	I		d	d	STLC 7546	Z	Z					
HC1	O	4	Z	Z				STLC 7546	d	d		
FS	I		d	d	STLC 7546	Z	Z					
AFERST#	O	4	L	L				STLC 7546	I	I		
AFEPDN#	O	4	L	L				STLC 7546	I	I		
MCLKO	O	4	Z	Z				STLC 7546	d	d		
RING	I		I	I	DAA	O	O				U	
OFFHOOK/RLY	O	4	Z	Z				DAA	d	d		

Pin Name	I/O Type	Sink (mA)	POR	STB	Source	POR	STB	Destination	POR	STB	Sys. Req.	Function
ACCLK	O	4	Z	Z				Codec CKIN	d	d		CODEC Interface (VCC) (4: The I/O direction is determined by CODEC Mode
ACRST#	O	4	Z	Z				Codec RESET#	d	d	U	
ACPD#/ACIRQ	O/I	4	d	x	Codec IRQ	O	O	Codec PD#	d	d	U/-	
SIBCLK[4]	IO	4	d	x	Codec SCLK	Z	Z	Codec SCLK	d	d		
SIBDIN	I		d	d	Codec DOUT	Z	Z					
SIBDOUT	O	4	Z	Z				Codec DIN	d	d		
SIBSYNC[4]	IO	4	d	d	Codec SYNC	Z	Z	Codec SYNC	d	d	D	
USBOVR#	I		I	I	SYSTEM BOARD	O	O				U	USB(VCC)
USBPEN#	O	4	L	L				SYSTEM BOARD	I	I		
USBD1P	IO		I	I							D	
USBD1M	IO		I	I							D	
USBD2P	IO		I	I							D	
USBD2M	IO		I	I							D	
USBD2P	IO		I	I							D	
TSMX	I		I	I	Touch Screen	O	O					10-bit A/D (AVCC4)
TSMY	I		I	I	Touch Screen	O	O					
TSPX	I		I	I	Touch Screen	O	O					
TSPY	I		I	I	Touch Screen	O	O					
KBCS#	O	4	H	H				H8 CS#	I	I		KBC (VCC)
XIOR#	O	4	H	H				H8 IOR#	I	I		
XIOW#	O	4	H	H				H8 IOW#	I	I		
KBIRQ0	I		I	I	H8 IRQ1#	H	H					
KBIRQ1	I		I	I	H8 IRQ12#	H	H					
PA7	IO	8	U	P								IO Port A (VCC)
PA6	IO	8	U	P								
PA5	IO	8	U	P								
PA4	IO	8	U	P								
PA3	IO	8	U	P								
PA2	IO	8	U	P								
PA1	IO	8	U	P								
PA0	IO	8	U	P								
PB7	IO	8	U	P								IO Port B (VCC) (multifunction with TIMER and KBC power manager.)
PB6	IO	8	U	P								
PB5/KBRESUME	IO/I	8	U	P								
PB4/KBWAKEUP#	IO/O	8	U	P								
PB3	IO	8	U	P								
PB2	IO	8	U	P								
PB1/TMO1#	IO/O	8	U	P								
PB0/TMO0#	IO/O	8	U	P								

Pin Name	I/O Type	Sink (mA)	POR	STB	Source	POR	STB	Destination	POR	STB	Sys. Req.	Function
PC7	IO	8	U	P								IO Port C (VCC)
PC6	IO	8	U	P								
PC5	IO	8	U	P								
PC4	IO	8	U	P								
PC3	IO	8	U	P								
PC2	IO	8	U	P								
PC1	IO	8	U	P								
PC0	IO	8	U	P								
PD7	IO	8	U	P								IO Port D (VCC)
PD6	IO	8	U	P								
PD5	IO	8	U	P								
PD4	IO	8	U	P								
PD3	IO	8	U	P								
PD2	IO	8	U	P								
PD1	IO	8	U	P								
PD0	IO	8	U	P								
PE7	IO	8	U	P								IO Port E (VCC)
PE6	IO	8	U	P								
PE5	IO	8	U	P								
PE4	IO	8	U	P								
PE3	IO	8	U	P								
PE2	IO	8	U	P								
PE1	IO	8	U	P								
PE0	IO	8	U	P								
RESETPI#	I		I	I	SYSTEM BOARD	O	O				U	System Reset Interface(VCC)
RESETMI#	I		I	I	SYSTEM BOARD	O	O				U	
RESETPO#	O		O	O				SYSTEM BOARD	I	I		
RESETMO#	O		O	O				SYSTEM BOARD	I	I		
KBCK	IO	8	d	d	SYSTEM BOARD	-	-		I	I	U	System Reset Interface(VCC)
KBDATA	IO	8	d	d	SYSTEM BOARD	-	-		I	I	U	
MSCK	IO	8	d	d		-	-	SYSTEM BOARD	I	I	U	
MSDATA	IO	8	d	d		-	-	SYSTEM BOARD	I	I	U	
AFECK	I		?	?				SYSTEM BOARD	?	?		
AFECKE	O		?	?	SYSTEM BOARD	?	?					
UCK	I		?	?				SYSTEM BOARD	?	?		
UCKE	O		?	?	SYSTEM BOARD	?	?					
PWM0	O	4	O	O				SYSTEM BOARD	I	I		MISC (VCC)
PWM1	O	4	O	O				SYSTEM BOARD	I	I		
P80LE	O	4	L	L								

Note : H : output high state
L : output low state
Z : high impedance state
I : input state
U : pull-up state
D : pull-down state
P : The state is programmed by system software
C : input/output function is selected by module. When it is input pin, the state is disable state.
But it is output pin, the state is pull-up state.
c : input/output function is selected by module. When it is input pin, the state is disable state.
But it is output pin, the state is low state.
x : input/output function is selected by module. When it is input pin, the state is disable state.
But it is output pin, the state is high impedance state.
d : disable state
? : unknown state

(6) HD64465BP Module Clock & Clock Source Relationship

Signal Name	Description	Clock Source
Microprocessor Reset Interface		
RESETPO#	Power-On Reset	AFECK
RESETMO#	Manual Reset	AFECK
AFE Module		
AFE_clk	AFE Interface Clock	AFECK
AFE_cmd_clk	AFE Bus Interface Clock	UCK
Serial Codec Module		
SCDI_clk	Serial Codec Interfece Clock	AFECK/UCK (**SCDICKS** bit in SCONFR)
USB Host Module		
USB_clk	USB Host Clock	UCK
USBbus_clk	USB Host Bus Interface Clock	AFECK/CKIO (**USBCKS** bit in SCONFR)
IrDA Module		
FIR_clk	FIR Clock	UCK
SIR_clk	SIR Clock	UCK
UART Module		
UART_PP_cmd_clk	UART Bus Interface Clock	UCK
UART_clk	UART Clock	UCK
Parallel Port Module		
UART_PP_cmd_clk	PP Bus Interface Clock	UCK
PP_clk24	PP 24MHz Clock	UCK
PP_clk1p8	PP 1.8MHz Clock	UCK
KBC Module		
KBC_clk	KBC Clock	UCK
PCMCIA Module		
PCMCIA_pwr_clk	PCMCIA Power Control Clock	UCK
Internal Module Software Reset		
RST_clk	Software Reset Clock	UCK

Note : RESETPO# signal is also used as reset signal for internal module.

(7) Application note for PS/2 interface

Problem:

Cannot detect change in CLOCK line when host sends data to device.

Description:

The KBCS (bit 10) of the KBCSR (address 0x1000DC00) always reads as one when the KBCIE (bit 15) of the KBCSR is set to zero.

When using the method to transferring data to the keyboard described on the HD64465 data sheet the driver never see the KBCS change from one to zero.

A software work around exists for this problem.

Work around:

When the KBCIE bit is set to one then the CLOCK line state is available on the KBCS bit.

With the KBCIE bit set to one the driver can send the required data to the keyboard. The side affect is that the hardware shift register is also receiving the data.

To prevent this data from causing an interrupt the KBCIE bit is cleared at the stop bit time and the KBRDF (bit 0) of the KBISR (address 0x1000DC04) is set to one.

Problem:

Cannot poll for input data when device interrupts are enabled.

Description:

There is no independent interrupt enable bit for the keyboard input shift register.

This is a problem because the host must send commands to the keyboard and receive an acknowledge byte from the keyboard. Some commands require the host to receive additional data bytes.

Without a means to disable the keyboard interrupt request and poll the hardware for a response from keyboard the driver code becomes much more complicated.

Work around:

Restructure the keyboard driver to allow the interrupt service task to differentiate the keyboard scan code stream from the host to keyboard command and response stream.

Problem:

Driver looses synchronization with the message stream.

Description:

With a current production version of the Logitech or Microsoft PS/2 mouse connected the HD64465 will set the CLOCK line low before the falling edge of the tenth clock.

When the HD64465 sets the CLOCK line low before the falling edge of the tenth clock it indicates that a complete byte has been received. The mouse sees this as a host request to abort the byte being transmitted. When the host allows the CLOCK line to go high the mouse will send the aborted byte again. This resent byte causes the driver to loose message stream synchronization.

The loss synchronization does not occur with old versions of the Microsoft mouse.

Work around:

There does not appear to be a software work around for this problem. The pull up resistor value of 10K ohms may be too large. Use a lower value between 2.2K and 4.7K.

HD64465 User's Manual

Publication Date: 1st Edition, February 1999
3rd Edition, March 2001
Published by: Electronic Devices Sales & Marketing Group
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.