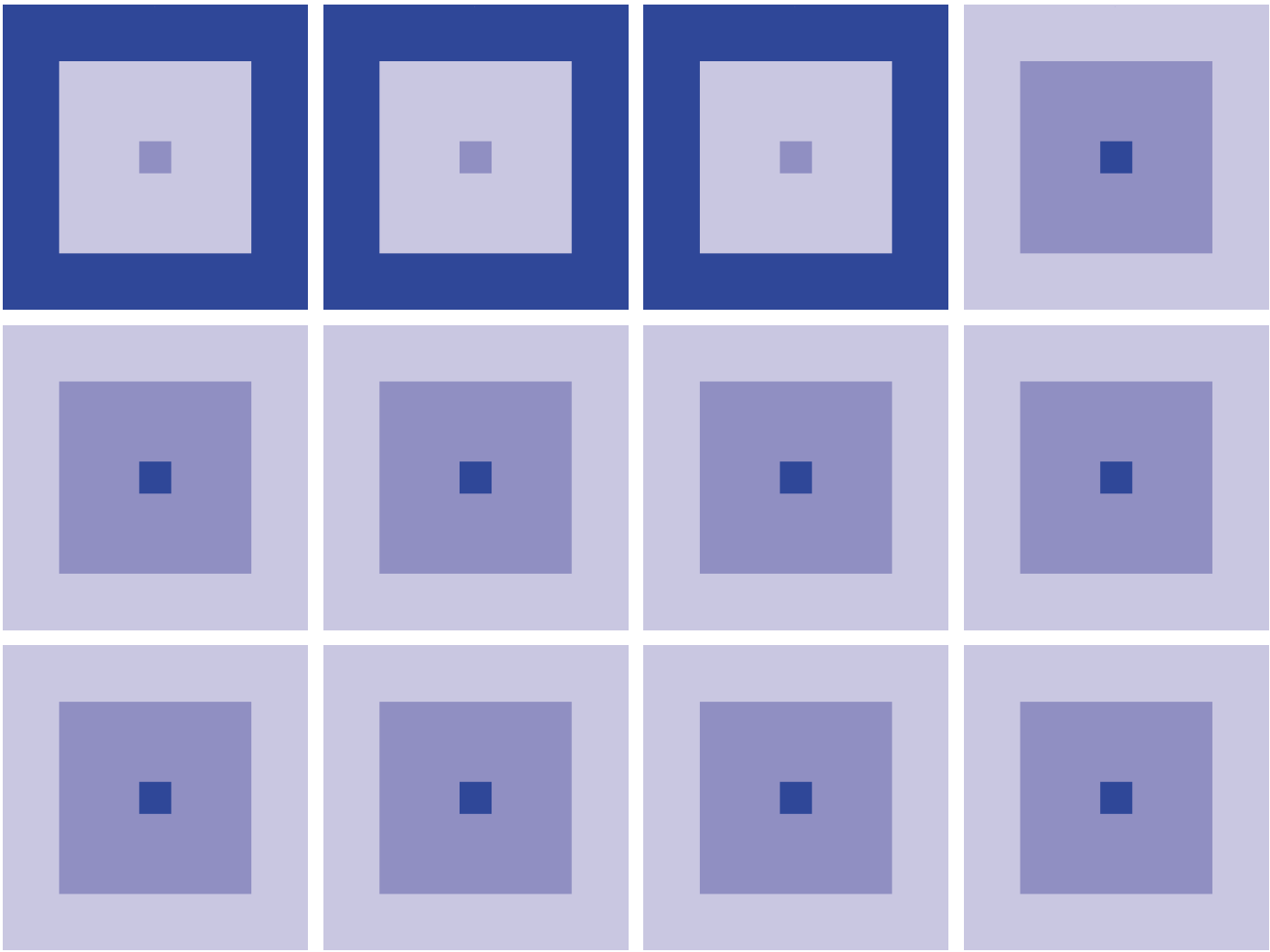
MF1530-01

EPSON

SCSI Interface Controller

S1R72105

Technical Manual



ENERGY SAVING EPSON

SEIKO EPSON CORPORATION

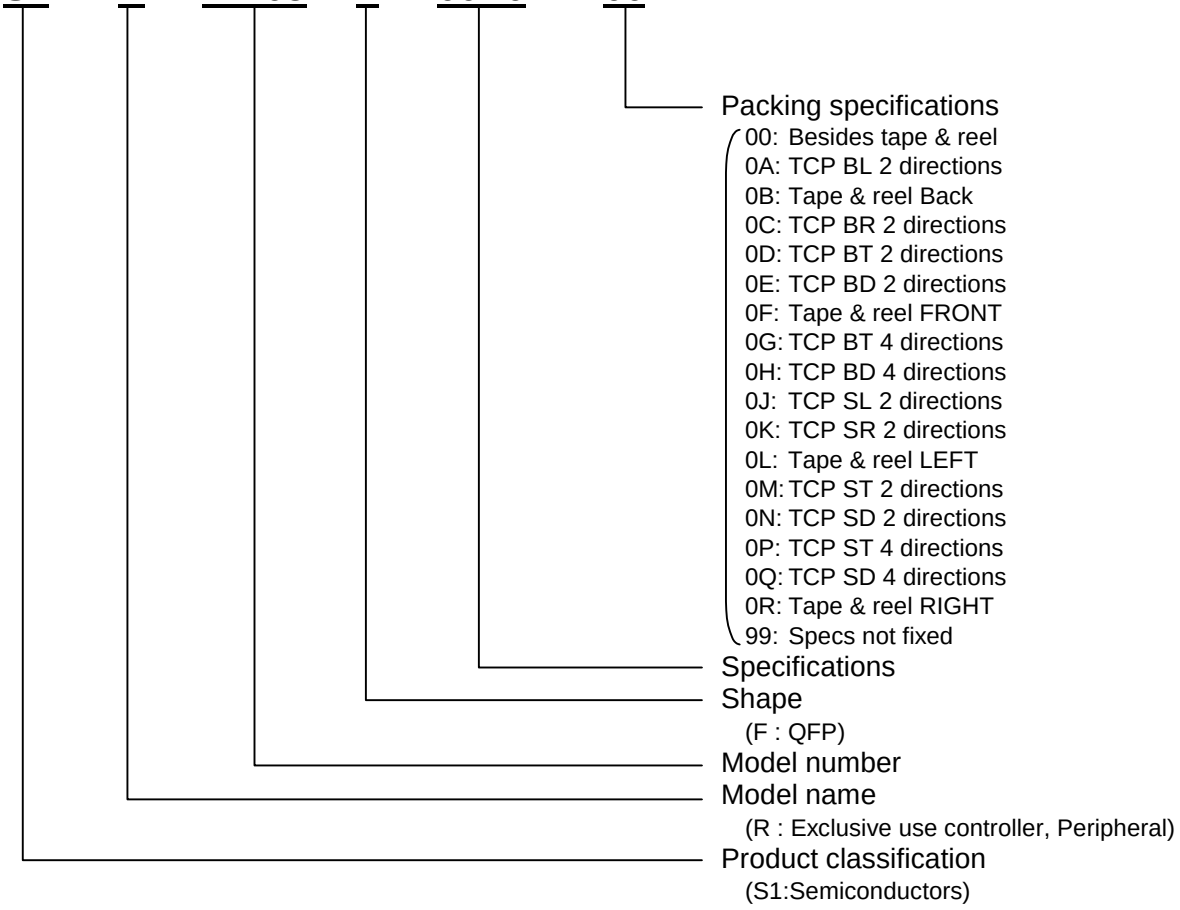
NOTICE

No part of this material may be reproduced or duplicated in any form or by any means without the written permission of Seiko Epson. Seiko Epson reserves the right to make changes to this material without notice. Seiko Epson does not assume any liability of any kind arising out of any inaccuracies contained in this material or due to its application or use in any product or circuit and, further, there is no representation that this material is applicable to products requiring high level reliability, such as, medical products. Moreover, no license to any intellectual property rights is granted by implication or otherwise, and there is no representation or warranty that anything made in accordance with this material will be free from any patent or copyright infringement of a third party. This material or portions thereof may contain technology or the subject relating to strategic products under the control of the Foreign Exchange and Foreign Trade Law of Japan and may require an export license from the Ministry of International Trade and Industry or other approval from anther government agency.

Configuration of product number

●DEVICES

S1 R 72105 F 00A0 00



CONTENTS

1. DESCRIPTION

S1R72105 contains the SCSI-3 interface controller and the USB 1.1 controller that support SCAM and FAST20 transfer, which is capable of bridging general-purpose I/O port and IDE DMA port as well as SCSI and USB interfaces.

2. FEATURES

«CPU Interface»

- It can be connected to a general-purpose CPU

«SCSI Interface»

- Compatible with SCSI-2 (10Mbps (synchronous), 5Mbps (asynchronous))
- Compatible with SCSI-3 FAST20 (20Mbps (synchronous)) transfer
- Compatible with SCAM Lv.1 (compatible with Lv.2 with firmware)
- Automatic processing of phase control
- Built-in single end driver
- Active negation I/O mounted

«USB Interface»

- Compatible with full speed mode (12Mbps) transfer
- Compatible with control transfer by endpoint 0 and bulk and interrupt transfers by three individual endpoints
- Split of the built-in SRAM (256 bytes) is programmable by user definition
  In addition to two-way endpoint 0, a maximum of three endpoints can be set

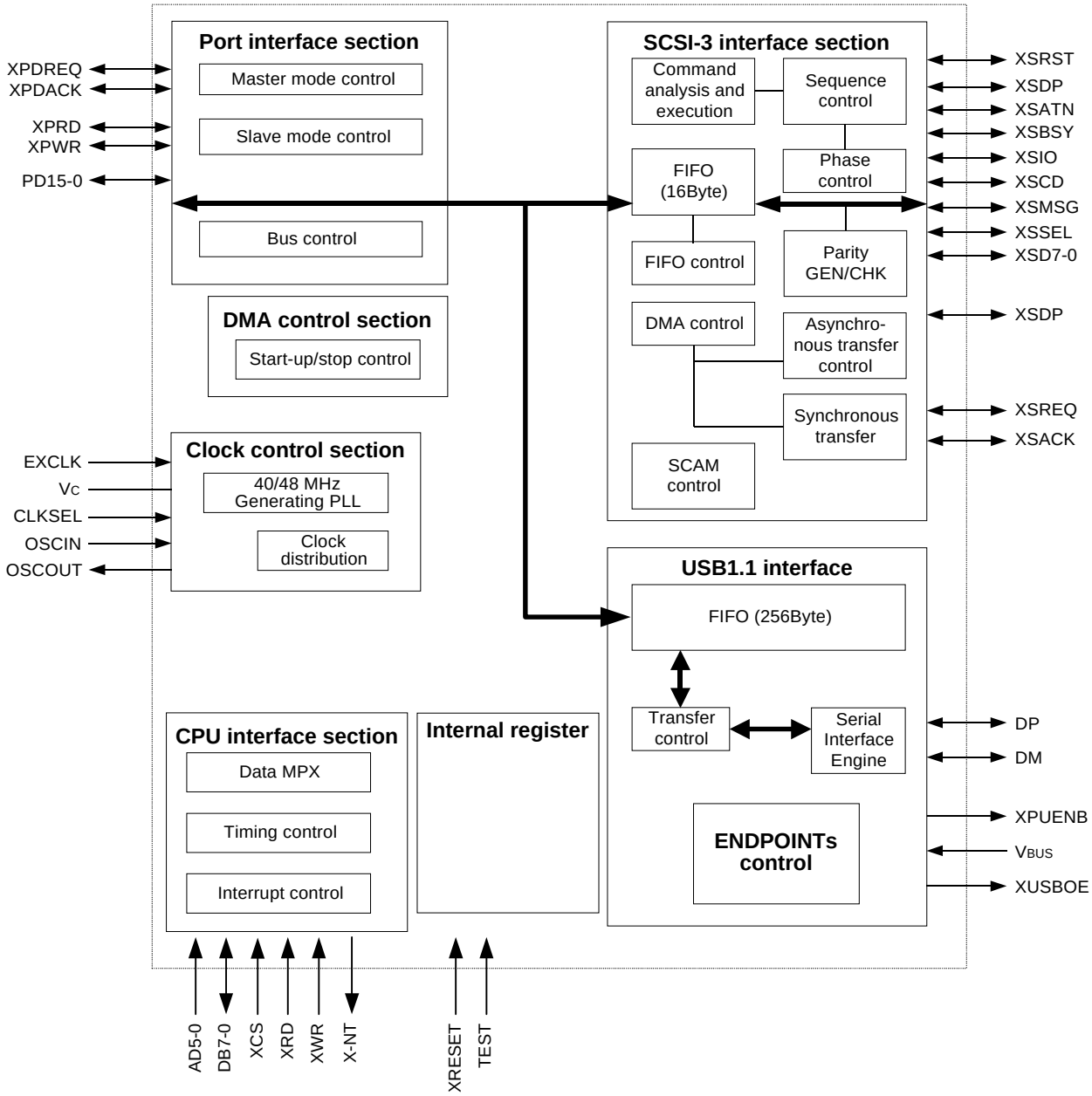
«PORT Interface»

- General-purpose 8/16 bit-selectable purpose interface
- It allows selection between master and slave
- Since it allows DMA connection to IDE(ATAPI), IDE (ATAPI) equipment can be easily turned into the SCSI and USB equipment

«Others»

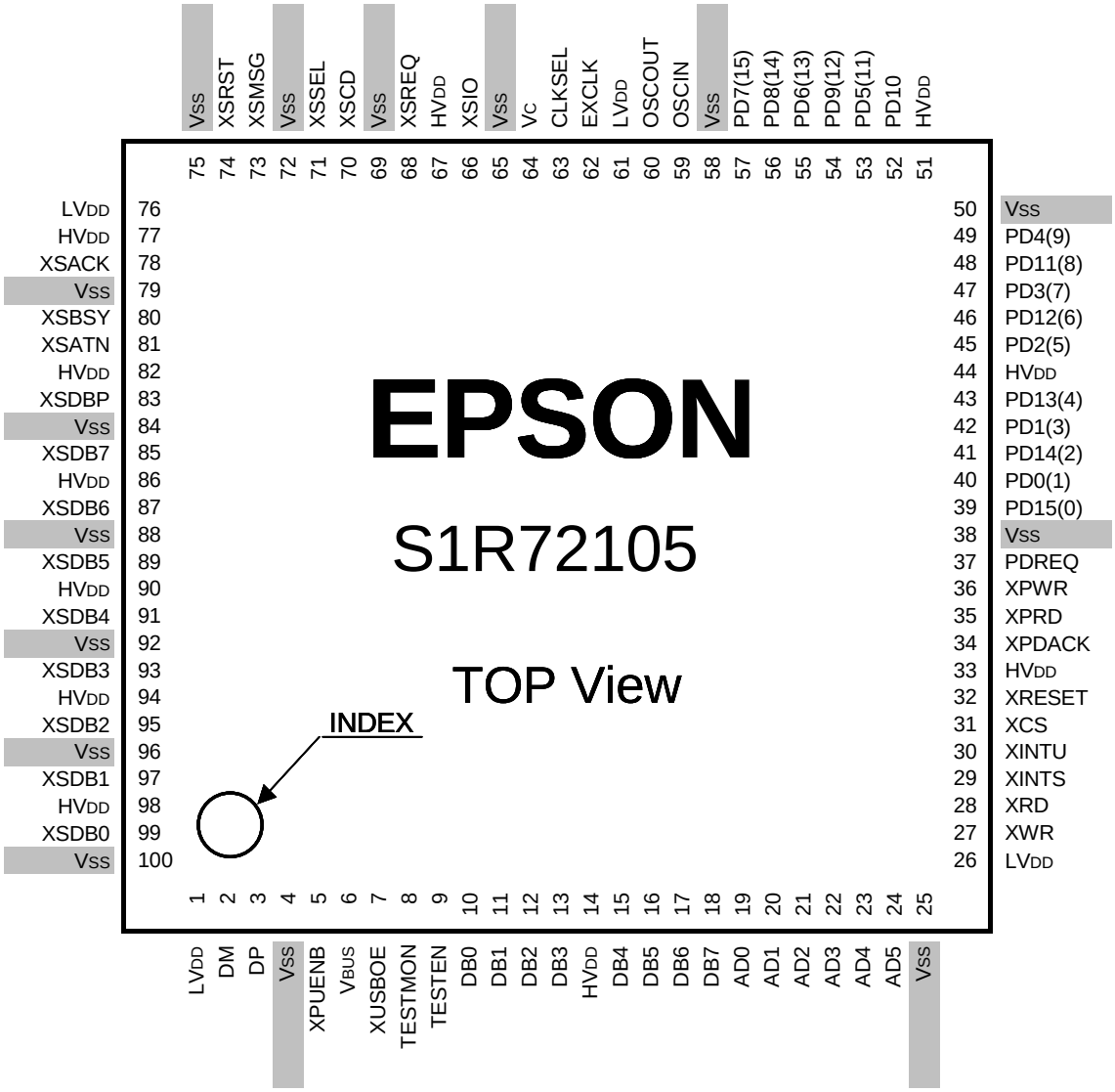
- Built-in oscillation circuit: 20MHz/40MHz
- Built-in PLL circuit (generates both USB and SCSI clocks from 20MHz oscillation)
- 100 pin QFP (0.5 mm pitch)
- Supply voltage: 5.0V±10% and 3.3V±0.3V
- No anti-radiation design

3. BLOCK DIAGRAM



4. PIN ASSIGNMENT

S1R72105F0A (QFP15-100pin)



5. PIN DESCRIPTION

The control signal with "X" at the head of a pin name is LOW-active.

Pin No.	Symbol	I/O	Functional description	Remarks
SCSI interface-related matters (18)				
99	XSDB0	Is/Otr	SCSI data signal (SD0 to SD7)	Drive capability 48mA
97	XSDB1			
95	XSDB2			
93	XSDB3			
91	XSDB4			
89	XSDB5			
87	XSDB6			
85	XSDB7			
83	XSDBP		SCSI data parity signal	Drive capability 48mA
81	XSATN	I/Ood	SCSI ATN signal	Drive capability 48mA
80	XSBSY	I/Ood	SCSI BSY signal	Drive capability 48mA
78	XSACK	Is/Otr	SCSI ACK signal	Drive capability 48mA
74	XSRST	I/Ood	SCSI RST signal	Drive capability 48mA
73	XSMSG	I/Ood	SCSI MSG signal	Drive capability 48mA
71	XSSEL	I/Ood	SCSI SEL signal	Drive capability 48mA
70	XSCD	I/Ood	SCSI C/D signal	Drive capability 48mA
68	XSREQ	Is/Otr	SCSI REQ signal	Drive capability 48mA
66	XSIO	I/Ood	SCSI I/O signal	Drive capability 48mA
USB interface-related matters (5)				
2	DM	I/O	USB data port	
3	DP	I/O	USB data port (Configure to pull up to 3.3V by a 1.5kΩ resistor externally. The register can be controlled by "XPUENB.")	
5	XPUENB	Ood	Control signal connecting a 1.5kΩ pull-up resistor to "DP" pin Ood. V_{BUS} = 5V and HIGH EnPullUp(USBCommon_b1) brings about LOW output.	Drive capability 6mA
6	V_{BUS}	Is	USB bus power detection input (Pull down externally).	
7	XUSBOE	O	LOW output while this IC outputs a signal to DP and DM pins.	Drive capability 3mA
Port interface-related matters (20)				
35	XPRD	Is/O	Port read signal	Drive capability 3mA
36	XPWR	Is/O	Port write signal	Drive capability 3mA
37	PDREQ	Is/O	Port DMA request signal (also operable in negative logic)	Drive capability 6mA
34	XPDACK	Is/O	Port DMA ACK signal	Drive capability 3mA
40	PD0	I/O	Port DMA data bus signal (PD0 to 15)	Drive capability 3mA
42	PD1			
45	PD2			
47	PD3			
49	PD4			
53	PD5			
55	PD6			
57	PD7			
56	PD8			
54	PD9			
52	PD10			
48	PD11			
46	PD12			
43	PD13			
41	PD14			
39	PD15			

Pin No.	Symbol	I/O	Functional description	Remarks
CPU interface-related matters (19)				
19	AD0	I	Address input pin (AD0 to AD5)	
20	AD1			
21	AD2			
22	AD3			
23	AD4			
24	AD5			
10	DB0	I/O	Data pin (DB0 to DB7)	Drive capability 3mA
11	DB1			
12	DB2			
13	DB3			
15	DB4			
16	DB5			
17	DB6			
18	DB7			
31	XCS	Is	Chip select signal for accessing internal register	
30	XINTU	O	USB interrupt request output signal	Drive capability 6mA
29	XINTS	O	SCSI interrupt request output signal	Drive capability 6mA
28	XRD	Is	Data read signal	
27	XWR	Is	Data write signal	
Others (17)				
59	OSCIN	I	Input to built-in oscillation circuit (40MHz or 20MHz)	
60	OSCOUT	O	Output from built-in oscillation circuit	
8	TESTMON	O	Monitor output for testing (open LOW output usually)	Drive capability 2mA
32	XRESET	Ispu	System reset input signal	
9	TESTEN	Ipd	Pin for testing (connected to LOW (GND) usually)	
63	CLKSEL	I	Input clock and PLL operation selection HIGH (LV_{DD}): Generates clock (SCSI_40MHz,USB_48MHz) in internal PLL. 20MHz oscillation in OSCIN/OUT or input 20MHz(3.3V) to OSCIN EXCLK is connected to LOW (GND) or HIGH (LV_{DD}). LOW (GND): PLL stop (power-down) , external clock in operation 40MHz oscillation in OSCIN/OUT or input 40MHz(3.3V) to OSCIN Input 48MHz(3.3V) to EXCLK	
62	EXCLK	I	External clock input pin for 3.3V level USB (Connected to LOW (GND) or HIGH (LV_{DD}) while it is not used).	
64	V_C	O	For instructions as to how to connect internal VCO control pins, refer to the description of PLL circuit.	
HV_{DD}:5V (5)				
14,33,44,51, 67,77,82,86, 90,94,98	HV_{DD}	P	Power supply for 5V interface	
LV_{DD}:3.3V (6)				
1,26,61,76	LV_{DD}	P	Power supply for internal operation	
V_{SS}:0V (17)				
4,25,38,50, 58,65,69,72, 75,79,84,88, 92,96,100	V_{SS}	P	GND	

(Note)
I : Input
Is : Schmitt input
Ipu : Pull-up input
Ispu : Pull-up Schmitt input
O : Output
Ood : Open-drain output
Otr : Try state output
Ipd : Pull-down input

6. FUNCTIONAL DESCRIPTION

6.1 CPU Interface Circuit

This block can be interfaced to a general-purpose CPU. It generally controls the interface with the CPU.
When the XCS signal from the CPU is LOW, the block can access the internal register. It decodes the address bus AD5 to AD0 to generate the address of the internal register. At this time, it generates the read/write strobe signal from the XRD/XWR signal, transferring data between the internal register. A wait signal to the CPU is not generated because of no-wait operation.

6.2 Internal Registers

Refer to the section of Register Functions as for the addresses of the internal registers and description of each bit. The main functions of this block are as follows:

(1) It generates control signals to each block according to the address, write-data and write-strobe signals generated by the CPU interface circuit.
(2) It stores the status signals from each block, and outputs data according to the address and read-strobe signals sent from the CPU interface circuit.

6.3 Port Interface Circuit

This is a block controlling the transfer to and from the external DMA port. It has the following functions:

(1) It controls the linkage operation of each functional block according to the control signal and the stop-operation signal sent from the DMA control circuit.
(2) It controls the transfer status of the external port according to PDREQ/XPDACK signals.
(3) It reads/writes data of the data bus PD15-0 of the port from/to FIFO in the SCSI-2 block. When transfer is disabled in the full/empty state of SCSI_FIFO, transfer to and from the port is temporarily halted according to the timing specified by the PDREQ/XPDACK signals.
(4) The port allows selection of bit width betweem 8 and 16.
(5) The port interface allows selection between the master and slave function (toward PDREQ/XPDACK/XPRD/XPWR direction).

6.4 DMA Control Circuit

This block controls the transfer between the DMA port and FIFO in the SCSI block and FIFO in the USB block. It has the following functions:

(1) It controls the linkage operation of each functional block according to the control signal from the internal register and the information and stop-operation signals from each block.
(2) It stores the status of each of functional blocks when their linkage operation ends, reporting it to the internal register at the specified timing.

6.5 SCSI-2 (3) Interface Circuit

This block generally controls the interfaces conforming to the SCSI-2 standard. It has the following functions:

(1) It automatically performs the SCSI protocol control on hardware.
(2) It has 16-staged off-set counter to control the off-set and transfer rate during synchronous transfer.
(3) In the command phase, it automatically distinguishes groups of commands received (in Target mode).
(4) It controls the automatic status/message transfer function. It supports the messages 00h/0Ah/0Bh. (in Target mode).
(5) It allows SCSI-3 FAST20(20Mbps) transfer.

SCAM compatibility
In addition to conventional SCSI, this LSI has SCAM (SCSI Configured Auto Magnify)-compatible functions as listed below:
These functions enable the device to operate as a SCAM Lv.1 drive.

(1) It monitors and recognizes the SCAM selection and causes an interruption.
(2) It responds to the selection response delay of 4ms or more, which enables distinction between SCAM and ordinary selections.
(3) It can operate SCSI bus's signal line directly because of its actual operation responding to the SCAM selection and sending/receiving data.

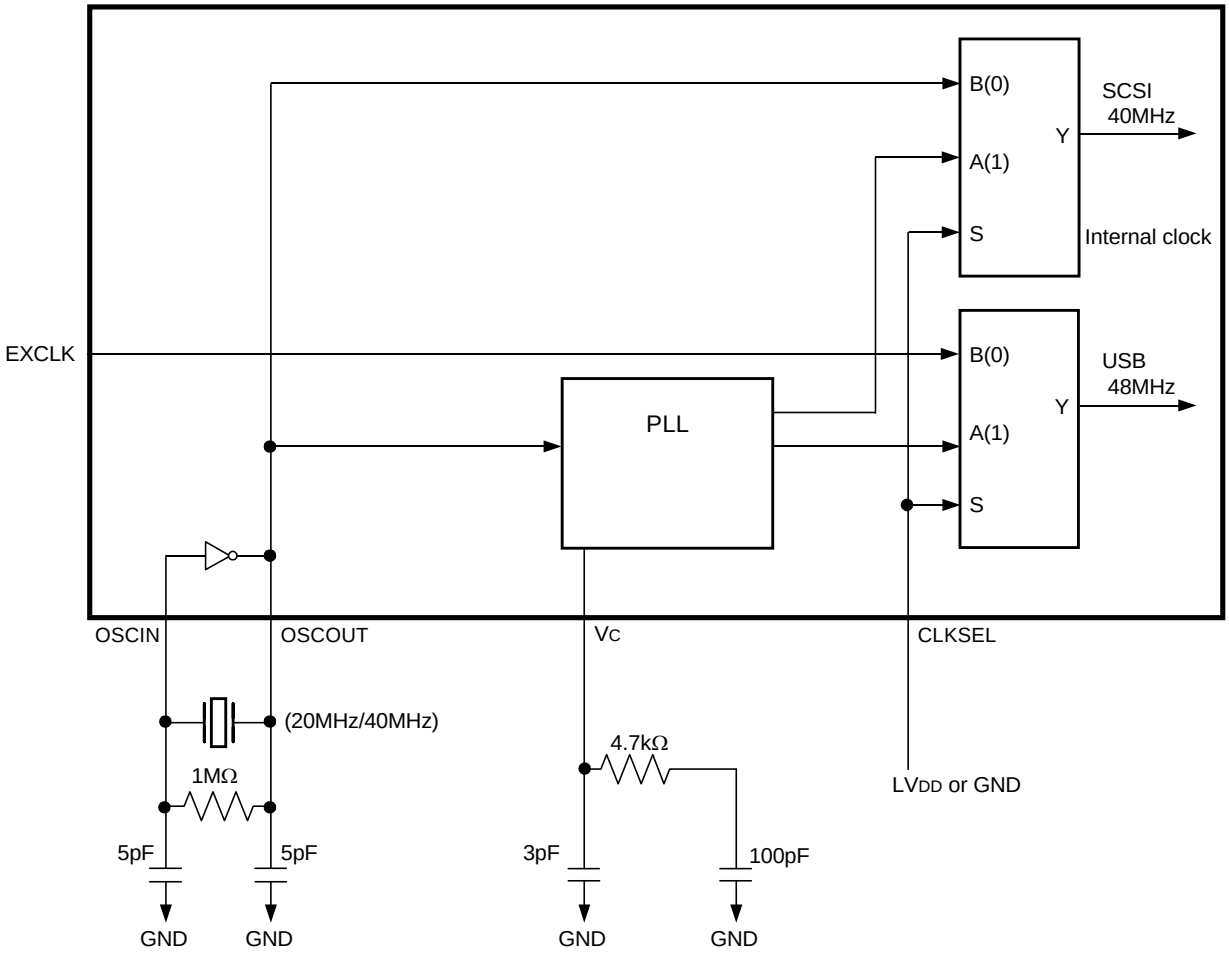
6.6 USB Interface Circuit

(1) It supports full speed device in conformity to the USB1.1 (It does not support low speed).
It supports control transfer (endpoint 0), bulk transfer, and interrupt transfer (It does not support isochronous transfer).

(2) Split of the built-in SRAM (256 bytes) is programmable by user definition.
In addition to two-way endpoint 0, a maximum of three endpoints can be set.
Three endpoints can be independently set to IN/OUT direction, any of four maximum packet lengths (8, 16, 32 or 64 bytes), any endpoint number, and buffer size (single or double).

6.7 PLL Circuit (Internal System Clock Generating Section)

This IC has the function to generate 40MHz(SCSI) and 48MHz(USB) required for the internal circuit from the clock generated by the oscillation circuit by using the PLL circuit.
The block diagram around the oscillation section is shown below:



- The IC enables easy setup of an oscillation circuit by connecting a crystal vibrator and feedback resistor (For characteristics of the crystal vibrator, contact us separately).
- It allows oscillation of 20MHz by means of the oscillation circuit mentioned above.
  In this case, 40MHz and 48MHz required for each block of SCSI and USB are generated by using the internal PLL. So connect the EXCLK pin to LV_{DD} or GND (CLKSEL = LV_{DD}).
- The IC allows operation by inputting a 40MHz external clock of 3.3V level to the OSCIN pin or 40MHz oscillation and inputting a 48MHz external clock of 3.3V level to the EXCLK pin without using the internal PLL. In this case, connect the CLKSEL to the "GND" to stop operation of the PLL block.

Depending on the usage, set the control signal as shown below:

	- When an oscillator circuit (20MHz) is used - When a 20 MHz clock of 3.3V level is input from the OSCIN pin (PLL is used)	- When an oscillator circuit (40MHz) is used - When a 40 MHz clock of 3.3V level is input from the OSCIN pin (PLL is not used)
Oscillation/input clock	20MHz	40MHz
CLKSEL	LV_{DD}	GND
EXCLK	LV_{DD} or GND	Input 48MHz(3.3V)

- PLL circuit specifications Ta=0 to 70°C LV_{DD}=3.3V±0.3V

Item	Specifications
Lock-up time	Within 1ms after oscillation was stabilized
Jitter	Within ±1ns

7. FUNCTION OF REGISTERS

7.1 List of Registers

Address	Register name	Abbreviated name
00h	Main Interrupt Status	MainIntStat
01h	Epr Interrupt Status	EPrIntStat
02h	Interrupt Status Window_0	IntStatWindow_0
03h	Interrupt Status Window_1	IntStatWindow_1
04h	Main Interrupt Enable	MainIntEnb
05h	Epr Interrupt Enable	EPrIntEnb
06h	Interrupt Enable Window_0	IntEnbWindow_0
07h	Interrupt Enable Window_1	IntEnbWindow_1
08h	Interrupt Index	IntIndex
09h	System Control	SystemCtrl
0Ah	USB Common	USBCommon
0Bh	- Reserved -	—
0Ch	- Reserved -	—
0Dh	Reset	Reset
0Eh	- Reserved -	—
0Fh	- Reserved -	—
10h	Port DMA Control	PortDMACtrl
11h	Port DMA Size_H	PortDMASize_H
12h	Port DMA Size_M	PortDMASize_M
13h	Port DMA Size_L	PortDMASize_L
14h	Port Configuration_0	PortConfig_0
15h	Port Configuration_1	PortConfig_1
16h	- Reserved -	—
17h	USB Index	USBIndex
18h	USB Window_0	USBWindow_0
19h	USB Window_1	USBWindow_1
1Ah	USB Window_2	USBWindow_2
1Bh	USB Window_3	USBWindow_3
1Ch	USB Window_4	USBWindow_4
1Dh	USB Window_5	USBWindow_5
1Eh	USB Window_6	USBWindow_6
1Fh	USB Window_7	USBWindow_7

Address	Register name	Abbreviated name
20h	Main Interrupt Status SCSI	MAININTS
21h	SCSI Interrupt status 1	SCSIINT1
22h	SCSI Interrupt status 2	SCSIINT2
23h	- Reserved -	—
24h	- Reserved -	—
25h	- Reserved -	—
26h	- Reserved -	—
27h	- Reserved -	—
28h	- Reserved -	—
29h	SCSI Mode 0	SCSIMODE0
2Ah	SCSI Mode 1	SCSIMODE1
2Bh	SCSI Control	SCSICTL
2Ch	SCSI Synchronous data transfer Mode	SCSIDATA
2Dh	SCSI DATA	SYNCMODE
2Eh	SCSI Own ID	OWNID
2Fh	SCSI Source/Destination ID	SDID
30h	SCSI Selection Timeout Counter	SELTIME
31h	SCSI FIFO Control	FIFOCTL
32h	SCSI FIFO Data	FIFODATA
33h	SCSI Non-DMA Transfer Size	NDMASIZE
34h	SCSI Command	COMMAND
35h	- Reserved -	—
36h	- Reserved -	—
37h	- Reserved -	—
38h	- Reserved -	—
39h	- Reserved -	—
3Ah	- Reserved -	—
3Bh	- Reserved -	—
3Ch	- Reserved -	—
3Dh	- Reserved -	—
3Eh	Test	TEST
3Fh	REVISION	REVISION

7.1.1 List of Register/Window Settings (USB)

- Details appearing in IntStatWindow_0(02h)

IntIndex(08h) 4 higher order bits (bit7,6,5,4)	Function of IntStatWindow_0 register	Description on display
0h	EP0IntStat	Endpoint 0 status display/clear
1h	EPaIntStat	Endpoint a status display/clear
2h	EPbIntStat	Endpoint b status display/clear
3h	EPcIntStat	Endpoint c status display/clear

- Details appearing in IntStatWindow_1(03h)

IntIndex(08h) 4 lower order bits (BIT3,2,1,0)	Function of IntStatWindow_1 register	Description on display
0h	EP0IntStat	Endpoint 0 status display/clear
1h	EPaIntStat	Endpoint a status display/clear
2h	EPbIntStat	Endpoint b status display/clear
3h	EPcIntStat	Endpoint c status display/clear

- Details appearing in IntEnbWindow_0(06h)

IntIndex(08h) higher order 4 bits (bit7,6,5,4)	Function of IntEnbWindow_0 register	Description on display
0h	EP0IntEnb	Endpoint 0 status interrupt enabled
1h	EPaIntEnb	Endpoint a status interrupt enabled
2h	EPbIntEnb	Endpoint b status interrupt enabled
3h	EPcIntEnb	Endpoint c status interrupt enabled

- Details appearing in IntEnbWindow_1(07h)

IntIndex(08h) 4 lower order bits (bit3,2,1,0)	Function of IntEnbWindow_1 register	Description on display
0h	EP0IntEnb	Endpoint 0 status interrupt enabled
1h	EPaIntEnb	Endpoint a status interrupt enabled
2h	EPbIntEnb	Endpoint b status interrupt enabled
3h	EPcIntEnb	Endpoint c status interrupt enabled

- Details appearing in USBWindow_0(18h) to USBWindow_0(1Fh)

USBIndex(17h)	Register name	Description on display
00h	USBWindow_0(18h)	USBAddress: USB address
	USBWindow_1(19h)	EP0Config_1：EP0 configuration
	USBWindow_2(1Ah)	EP0InControl: EP0 IN Transaction control
	USBWindow_3(1Bh)	EP0OutControl: EP0 OUT Transaction control
	USBWindow_4(1Ch)	(Reserved)
	USBWindow_5(1Dh)	EP0FIFOremain: EP0 FIFO counter
	USBWindow_6(1Eh)	EP0FIFOforCPU: EP0 FIFO for CPU access
	USBWindow_7(1Fh)	EP0FIFOCtrl: EP0 FIFO control
01h to 03h	USBWindow_0(18h)	EPrConfig_0:EP{r}(r=a,b,c) configuration
	USBWindow_1(19h)	EPrConfig_1:EP{r}(r=a,b,c) configuration
	USBWindow_2(1Ah)	EPrControl：EP {r} (r=a,b,c) Transaction control
	USBWindow_3(1Bh)	(Reserved)
	USBWindow_4(1Ch)	(Reserved)
	USBWindow_5(1Dh)	EPrFIFOremain：EP {r} (r=a,b,c) FIFO counter
	USBWindow_6(1Eh)	EPrFIFOforCPU：EP {r} (r=a,b,c) FIFO for CPU access
	USBWindow_7(1Fh)	EPrFIFOCtrl： EP {r} (r=a,b,c) FIFO control
08h	USBWindow_0(18h)	EP0_SETUP_0： EP0 SETUP stage receive data
	USBWindow_1(19h)	EP0_SETUP_1： EP0 SETUP stage receive data
	USBWindow_2(1Ah)	EP0_SETUP_2： EP0 SETUP stage receive data
	USBWindow_3(1Bh)	EP0_SETUP_3： EP0 SETUP stage receive data
	USBWindow_4(1Ch)	EP0_SETUP_4： EP0 SETUP stage receive data
	USBWindow_5(1Dh)	EP0_SETUP_5： EP0 SETUP stage receive data
	USBWindow_6(1Eh)	EP0_SETUP_6： EP0 SETUP stage receive data
	USBWindow_7(1Fh)	EP0_SETUP_7： EP0 SETUP stage receive data
09h	USBWindow_0(18h)	FrameNumber_H: Higher order 3 bits in the FrameNumber field of the SOF Packet received
	USBWindow_1(19h)	FrameNumber_L: Lower order 8 bits in the FrameNumber field of the SOF Packet received
	USBWindow_2(1Ah)	(Reserved)
	USBWindow_3(1Bh)	(Reserved)
	USBWindow_4(1Ch)	(Reserved)
	USBWindow_5(1Dh)	(Reserved)
	USBWindow_6(1Eh)	(Reserved)
	USBWindow_7(1Fh)	(Reserved)

7.2 List of Registers/Bits

Address	Type	Register name	Default value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00h	R/W	MainIntStat	40/00h	USB resume	USB reset	USB suspend	Detect SOF	Port DMACmp	SCSI	RcvEP0 SETUP	EPrInt Stat
01h	R/W	EPrIntStat	00h	—	—	—	—	EPcInt Stat	EPbInt Stat	EPaInt Stat	EP0Int Stat
02h	R/W	IntStatWindow_0	00h	IntStatWindow_0 (swiching by IntIndex_0)							
03h	R/W	IntStatWindow_1	00h	IntStatWindow_1 (swiching by IntIndex_1)							
04h	R/W	MainIntEnb	00h	EnUSB resume	EnUSB reset	EnUSB suspend	EnDetect SOF	EnPort DMACmp	EnSCSI	EnRcv SETUP	EnEPr IntStat
05h	R/W	EPrIntEnb	00h	—	—	—	—	EnEPc Int	EnEPb Int	EnEPa Int	EnEP0 Int
06h	R/W	IntEnbWindow_0	00h	IntEnWindow_0 (swiching by IntIndex_0)							
07h	R/W	IntEnbWindow_1	00h	IntEnWindow_1 (swiching by IntIndex_1)							
08h	R/W	IntIndex	01h	IntIndex_0				IntIndex_1			
09h	R/W	SystemCtrl	03h	—	—	—	Go Suspend	Send Wakeup	—	xINT mode1	xINT mode0
0Ah	R/W	USBCommon	xxh	V_{BUS}	—	—	—	—	IgnrTgl Mis	EnPull Up	Active USB
0Bh	R/W	—	00h	—	—	—	—	—	—	—	—
0Ch	R/W	—	00h	—	—	—	—	—	—	—	—
0Dh	R/W	Reset	00h						PORT	SCSI	USB
0Eh	R/W	—	00h	—	—	—	—	—	—	—	—
0Fh	R/W	—	00h	—	—	—	—	—	—	—	—
10h	R/W	PortDMACtrl	00h	MODE1	MODE0	—	—	—	—	S_FIFO	DTGO
11h	R/W	PortDMASize_H	00h	DBC23	DBC22	DBC21	DBC20	DBC19	DBC18	DBC17	DBC16
12h	R/W	PortDMASize_M	00h	DBC15	DBC14	DBC13	DBC12	DBC11	DBC10	DBC9	DBC8
13h	R/W	PortDMASize_L	00h	DBC7	DBC6	DBC5	DBC4	DBC3	DBC2	DBC1	DBC0
14h	R/W	PortConfig_0	00h	ACP	BUSC	PSLV	—	PRQLV	SWAP	ODS	BUS8
15h	R/W	PortConfig_1	00h	AP3	AP2	AP1	AP0	NP3	NP2	NP1	NP0
16h	—	—	—	—	—	—	—	—	—	—	—
17h	R/W	USBIndex	00h	USBIndex							
18h	R/W	USBWindow_0	00h	USBWindow_0 (swiching by USBIndex)							
19h	R/W	USBWindow_1	00h	USBWindow_1 (swiching by USBIndex)							
1Ah	R/W	USBWindow_2	00h	USBWindow_2 (swiching by USBIndex)							
1Bh	R/W	USBWindow_3	00h	USBWindow_3 (swiching by USBIndex)							
1Ch	R/W	USBWindow_4	00h	USBWindow_4 (swiching by USBIndex)							
1Dh	R/W	USBWindow_5	00h	USBWindow_5 (swiching by USBIndex)							
1Eh	R/W	USBWindow_6	00h	USBWindow_6 (swiching by USBIndex)							
1Fh	R/W	USBWindow_7	00h	USBWindow_7 (swiching by USBIndex)							

Address	Type	Register name	Default value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
20h	R/W	MAININTS	00h	GOOD	SABT	EXEC	SCSI1	SCSI2	—	DTCMP	ASCMP
21h	R/W	SCSIINT1	00h	SPERR	IDERR	SELTO	SATN	BFREE	ILPHS	SCSEL	WOATN
22h	R/W	SCSIINT2	00h	—	SRST	OFERR	UNDEF	CMDER	RESEL	SEL	LARBT
23h	—	—	—	—	—	—	—	—	—	—	—
24h	—	—	—	—	—	—	—	—	—	—	—
25h	—	—	—	—	—	—	—	—	—	—	—
26h	—	—	—	—	—	—	—	—	—	—	—
27h	—	—	—	—	—	—	—	—	—	—	—
28h	—	—	—	—	—	—	—	—	—	—	—
29h	R/W	SCSIMODE0	00h	—	—	—	ULTRA	AUTO1	AUTO2	AN_C	AN_D
2Ah	R/W	SCSIMODE1	00h	STPPE	ATNPE	STATN	AUTO	RINH	SINH	DACS	SPCEN
2Bh	R/W	SCSICTL	00h	ACK	ATN	SEL	BSY	REQ	MSG	I/O	C/D
2Ch	R/W	SCSIDATA	00h	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
2Dh	R/W	SYNCMODE	00h	RATE3	RATE2	RATE1	RATE0	OFF3	OFF2	OFF1	OFF0
2Eh	R/W	OWNID	00h	—	—	—	—	—	OID2	OID1	OID0
2Fh	R/W	SDID	xxh	—	SID2	SID1	SID0	—	DID2	DID1	DID0
30h	R/W	SELTIME	00h	ST7	ST6	ST5	ST4	ST3	ST2	ST1	ST0
31h	R/W	FIFOCTL	01h	—	—	—	—	—	FCLR	FULL	EMPTY
32h	R/W	FIFODATA	xxh	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0
33h	R/W	NDMASIZE	FFh	NSZ7	NSZ6	NSZ5	NSZ4	NSZ3	NSZ2	NSZ1	NSZ0
34h	R/W	COMMAND	00h	CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
35h	—	—	—	—	—	—	—	—	—	—	—
36h	—	—	—	—	—	—	—	—	—	—	—
37h	—	—	—	—	—	—	—	—	—	—	—
38h	—	—	—	—	—	—	—	—	—	—	—
39h	—	—	—	—	—	—	—	—	—	—	—
3Ah	—	—	—	—	—	—	—	—	—	—	—
3Bh	—	—	—	—	—	—	—	—	—	—	—
3Ch	—	—	—	—	—	—	—	—	—	—	—
3Dh	—	—	—	—	—	—	—	—	—	—	—
3Eh	R	TEST	00h	TM2	TM1	TM0	USEL1	USEL0	OFST	SCBC	DMBC
3Fh	R/W	REVISION	00h	REV7	REV6	REV5	REV4	REV3	REV2	REV1	REV0

7.2.1 List of Registers/Bits/List of Window Configuration(USB) List of Bits

- Details appearing in IntStatWindow_0(02h)

IntIndex(08h) Higher order 4 bits (BIT7,6,5,4)	IntStatWindow_0 Functions of register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0h	EP0IntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
1h	EPaIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
2h	EPbIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
3h	EPcIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv

- Details appearing in IntStatWindow_1(03h)

IntIndex(08h) Lower order 4 bits (bit3,2,1,0)	IntStatWindow_1 Functions of register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0h	EP0IntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
1h	EPaIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
2h	EPbIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
3h	EPcIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv

- Details appearing in IntEnbWindow_0(06h)

IntIndex(08h) Higher order 4 bits (bit7,6,5,4)	IntStatWindow_0 Functions of register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0h	EP0IntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
1h	EPaIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
2h	EPbIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
3h	EPcIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv

- Details appearing in IntEnbWindow_1(07h)

IntIndex(08h) Lower order 4 bits (bit3,2,1,0)	IntStatWindow_0 Functions of register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0h	EP0IntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
1h	EPaIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
2h	EPbIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv
3h	EPcIntStat	INtranACK	OUTtranACK	INtranErr	OUTtranErr	INtranNAK	OUTtranNAK	INtolkenRcv	OUTtolkenRcv

- Details appearing in USBWindow_0(18h) to USBWindow_0(1Fh)

USBIndex(17h)	Register name	Function of register	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00h	USBWindow_0 (18h)	USBAddress	—	(MSB)	USB Address [6:0]					(LSB)
	USBWindow_1 (19h)	EP0Config_1	OUTxIN	—	—	—	MaxPacketSize[3:0]			
	USBWindow_2 (1Ah)	EP0InControl	INForce NAK	InForce STALL	InEnShort Pkt	—	—	InToggle Stat	InToggle Clr	InToggle Set
	USBWindow_3 (1Bh)	EP0OutControl	OutForce NAK	OutForce STALL	—	—	—	OutToggle Stat	OutToggle Clr	OutToggle Set
	USBWindow_4 (1Ch)	(Reserved)	—	—	—	—	—	—	—	—
	USBWindow_5 (1Dh)	EP0FIFOremain	(MSB)	EP0FIFOremainCounter [7:0]						(LSB)
	USBWindow_6 (1Eh)	EP0FIFOforCPU	(MSB)	EP0FIFOdata [7:0]						(LSB)
	USBWindow_7 (1Fh)	EP0FIFOCtrl	FIFO Empty	FIFO Full	FIFO Ctr	—	—	AutoForce NAK	EnFIFO Owr	EnFIFO Ord
01h to 07h	USBWindow_0 (18h)	EPrConfig_0	—	(MSB)	USB Address [6:0]					(LSB)
	USBWindow_1 (19h)	EPrConfig_	OUTxIN	—	—	—	MaxPacketSize[3:0]			
	USBWindow_2 (1Ah)	EPrControl	INForce NAK	InForce STALL	InEnShort Pkt	—	—	InToggle Stat	InToggle Clr	InToggle Set
	USBWindow_3 (1Bh)	(Reserved)	OutForce NAK	OutForce STALL	—	—	—	OutToggle Stat	OutToggle Clr	OutToggle Set
	USBWindow_4 (1Ch)	(Reserved)	—	—	—	—	—	—	—	—
	USBWindow_5 (1Dh)	EPrFIFOremain	(MSB)	EP0FIFOremainCounter [7:0]						(LSB)
	USBWindow_6 (1Eh)	EPrFIFOforCPU	(MSB)	EP0FIFOdata [7:0]						(LSB)
	USBWindow_7 (1Fh)	EPrFIFOCtrl	FIFO Empty	FIFO Full	FIFO Ctr	—	—	AutoForce NAK	EnFIFO Owr	EnFIFO Ord
08h	USBWindow_0 (18h)	EP0_SETUP_0	(MSB)	EP0_RcvSETUPdata_0						(LSB)
	USBWindow_1 (19h)	EP0_SETUP_1	(MSB)	EP0_RcvSETUPdata_1						(LSB)
	USBWindow_2 (1Ah)	EP0_SETUP_2	(MSB)	EP0_RcvSETUPdata_2						(LSB)
	USBWindow_3 (1Bh)	EP0_SETUP_3	(MSB)	EP0_RcvSETUPdata_3						(LSB)
	USBWindow_4 (1Ch)	EP0_SETUP_4	(MSB)	EP0_RcvSETUPdata_4						(LSB)
	USBWindow_5 (1Dh)	EP0_SETUP_5	(MSB)	EP0_RcvSETUPdata_5						(LSB)
	USBWindow_6 (1Eh)	EP0_SETUP_6	(MSB)	EP0_RcvSETUPdata_6						(LSB)
	USBWindow_7 (1Fh)	EP0_SETUP_7	(MSB)	EP0_RcvSETUPdata_7						(LSB)
09h	USBWindow_0 (18h)	FrameNumber_H	—	—	—	—	—	(MSB)		
	USBWindow_1 (19h)	FrameNumber_L	FrameNumber [10:0]							(LSB)

7.3 Detailed Description of Each Register

7.3.1 Main Interrupt Status (MainIntStat) R/W

When the IC interrupts the CPU, the CPU identifies the interrupt status register responsible for interruption by reading this register first.
Following the reading of this register, the CPU identifies the bit as a source of interruption and clears it by writing the value read after appropriate interrupt processing.
When the EPrIntStat is a source of interruption, the bit should be cleared by writing the value read to the interrupt status register corresponding to each bit of the interrupt status register.

Address	Register Name	Bit Symbol	Description
00h	MainIntStat	7: USBresume	USB Resume
		6: USBreset	USB Reset
		5: USBsuspend	USB Suspend
		4: DetectSOF	Detect SOF Token
		3: PortDMACmp	Port DMA Complete
		2: SCSI	SCSI interrupt
		1: RcvEP0SETUP	Receive EP0 SETUP Transaction
		0: EPrIntStat	Epr Interrupt Status

BIT7 USB Resume
When Resume is present while the GoSuspend bit of the SystemCtrl register (09h) is being set, this bit becomes HIGH. Clearing GoSuspend clears it.
BIT6 USB Reset
At USB reset, this bit becomes HIGH.
At the same time, the USB address register shown in “7.5.2.1 USB Address (USBAddress)” is cleared.
BIT5 USB Suspend
At USB Suspend, this bit becomes HIGH.
BIT4 Detect SOF Token
When the SOF Token is detected, this bit becomes HIGH.
BIT3 Port DMA Complete
This bit becomes HIGH when a port DMA transfer activated by the PortDMACtrl register (10h) ends.
It also becomes HIGH when the transfer is forced to terminate by writing “0” to the DTGO bit of the PortDMACtrl register. In conditions of mode1:0=“00” of PortDMACtrl register, XINTU is set by this factor. In other modes, XINTU is not set by this factor.
BIT2 SCSI Interrupt
When SCSI-related interrupt occurs, this bit becomes HIGH Detailed factors appear in MAINTS/SCSIINT1/SCSIINT2 register. This does not change when accessing this register.
BIT1 Receive EP0 SETUP Transaction
This bit becomes HIGH when the endpoint 0 completes the SETUP Stage normally.
The received data appears in the USBWindow_0 register (18h) - USBWindow_7 register (1Fh) following the setting of the USBIndex register (17h) to 08h.
BIT0 EPr Interruput Status
This bit becomes HIGH when the interrupt status corresponding to each of the EPrIntStat register(01h) is a factor.

7.3.2 EPr Interrupt Status(EPrIntStat) R/W

The factor responsible for endpoint interrupt status can be identified by reading this register. When all factors of endpoint-by-endpoint interruption (interrupt factors at the main source) shown by each bit are cleared, the relevant bit is cleared.
Following the reading of this register, the bit (interrupt factor at the main source) is cleared by writing the value read in the interrupt status register (02h or 03h) corresponding to each bit of the appropriate interrupt status register.
The appropriate interrupt status register is shown in the IntStatWindow_0 register (02h) or IntStatWindow_1 register (03h) by writing a value to the IntIndex register (08h).

Address	Register Name	Bit Symbol	Description
01h	EPrIntStat	7:	Reserved
		6:	Reserved
		5:	Reserved
		4:	Reserved
		3: EPcIntStat	Endpoint c Interrupt Status
		2: EPbIntStat	Endpoint b Interrupt Status
		1: EPaIntStat	Endpoint a Interrupt Status
		0: EP0IntSta t	Endpoint 0 Interrupt Status

BIT3 Endpoint c Interrupt Status
This bit becomes HIGH when an interrupt factor related to the USB interface shown in the EPcIntStat register is present.
BIT2 Endpoint b Interrupt Status
This bit becomes HIGH when an interrupt factor related to the USB interface shown in the EPbIntStat register is present.
BIT1 Endpoint a Interrupt Status
This bit becomes HIGH when an interrupt factor related to the USB interface shown in the EPaIntStat register is present.
BIT0 Endpoint 0 Interrupt Status
This bit becomes HIGH when an interrupt factor related to the USB interface shown in the EP0IntStat register is present.

7.3.3 Interrupt Status Window 0(IntStatWindow_0) R/W

The endpoint interrupt status register appears.
The interrupt status to be displayed changes according to the value set at IntIndex_0 of the IntIndex register (08h).
For set value of the IntIndex register, refer to section “7.4 Detailed Description of Set Values of IntIndex Register.”

Address	Register Name	Bit Symbol	Description
02h	IntStatWindow_0	7: IntStatWindow_0[7]	Interrupt Status Window 0
		6: IntStatWindow_0[6]	
		5: IntStatWindow_0[5]	
		4: IntStatWindow_0[4]	
		3: IntStatWindow_0[3]	
		2: IntStatWindow_0[2]	
		1: IntStatWindow_0[1]	
		0: IntStatWindow_0[0]	

7.3.4 Interrupt Status Window 1(IntStatWindow_1) R/W

The endpoint interrupt status register appears.
The interrupt status to be displayed changes according to the value set at IntIndex_1 of the IntIndex register (08h).
For set value of the IntIndex register, refer to section "7.4 Detailed Description of Set Values of IntIndex Register."

Address	Register Name	Bit Symbol	Description
03h	IntStatWindow_1	7: IntStatWindow_1[7]	Interrupt Status Window 1
		6: IntStatWindow_1[6]	
		5: IntStatWindow_1[5]	
		4: IntStatWindow_1[4]	
		3: IntStatWindow_1[3]	
		2: IntStatWindow_1[2]	
		1: IntStatWindow_1[1]	
		0: IntStatWindow_1[0]	

7.3.5 Main Interrupt Enable (MainIntEnb) R/W

This register enables/disables interrupt factors of the MinInstStat register.
When the corresponding bit is set to HIGH an interruption to the CPU is enabled.

Address	Register Name	Bit Symbol	Description
04h	MainIntEnb	7: EnUSBresume	Enable USB Resume
		6: EnUSBreset	Enable USB Reset
		5: EnUSBsuspend	Enable USB Suspend
		4: EnDetectSOF	Enable Detect SOF
		3: EnPortDMACmp	Enable Port DMA Complete
		2: EnSCSI	Enable SCSI Interrupt
		1: EnRcvSETUP	Enable Received SETUP
		0: EnEPrIntStat	Enable EPr Interrupt Status

BIT2 Enable SCSI Interrupt

When this bit is set to HIGH all SCSI interruptions including ASCMP are enabled. For DTCMP interruption by SCSI, the BIT3 setting is valid instead of this bit.

7.3.6 EPr Interrupt Enable (EPrIntEnb) R/W

This register enables/disables interrupt factors of the EprIntStat register.
When the corresponding bit is set to HIGH an interruption to the CPU is enabled.

Address	Register Name	Bit Symbol	Description
05h	EPrIntEnb	7:	Reserved
		6:	Reserved
		5:	Reserved
		4:	Reserved
		3: EnEPcInt	Enable Endpoint c Interrupt
		2: EnEPbInt	Enable Endpoint b Interrupt
		1: EnEPaInt	Enable Endpoint a Interrupt
		0: EnEP0Int	Enable Endpoint 0 Interrupt

7.3.7 Interrupt Enable Window 0 (IntEnbWindow_0) R/W

The register enabling/disabling endpoint interruption appears.
The interrupt status to be displayed changes according to the value set at IntIndex_0 of the IntIndex register (08h).
For set value of the IntIndex register, refer to section "7.4 Detailed Description of Set Values of IntIndex Register."

Address	Register Name	Bit Symbol	Description
06h	IntEnbWindow_0	7: IntEnbWindow_0 [7] 6: IntEnbWindow_0 [6] 5: IntEnbWindow_0 [5] 4: IntEnbWindow_0 [4] 3: IntEnbWindow_0 [3] 2: IntEnbWindow_0 [2] 1: IntEnbWindow_0 [1] 0: IntEnbWindow_0 [0]	Interrupt Enable Window 0

7.3.8 Interrupt Enable Window 1(IntEnbWindow_1) R/W

The register enabling/disabling endpoint interruption appears.
The interrupt status to be displayed changes according to the value set at IntIndex_1 of the IntIndex register (08h).
For set value of the IntIndex register, refer to section "7.4 Detailed Description of Set Values of IntIndex Register."

Address	Register Name	Bit Symbol	Description
07h	IntEnbWindow_1	7: IntEnbWindow_1 [7] 6: IntEnbWindow_1 [6] 5: IntEnbWindow_1 [5] 4: IntEnbWindow_1 [4] 3: IntEnbWindow_1 [3] 2: IntEnbWindow_1 [2] 1: IntEnbWindow_1 [1] 0: IntEnbWindow_1 [0]	Interrupt Enable Window 1

7.3.9 Interrupt Index(IntIndex) R/W

Sets registers to be displayed in the IntStatWindow and IntEnbWindow registers.
For set value of the IntIndex register, refer to section "7.4 Detailed Description of Set Values of IntIndex Register."

Address	Register Name	Bit Symbol	Description
08h	IntIndex	7: IntIndex_0[3] 6: IntIndex_0[2] 5: IntIndex_0[1] 4: IntIndex_0[0]	Interrupt Index 0
		3:IntIndex_1[3] 2: IntIndex_1[2] 1: IntIndex_1[1] 0: IntIndex_1[0]	Interrupt Index 1

BIT7-4 Interrupt Index 0
Set registers to be displayed in IntStatWindow_0 and IntEnbWindow_0.
BIT3-0 Interrupt Index 1
Set values to be displayed in IntStatWindow_1 and IntEnbWindow_1.

7.3.10 System Control(SystemCtrl) R/W

Sets the system operation.

Address	Register Name	Bit Symbol	Description
09h	SystemCtrl	7: 0	Reserved
		6: 0	Reserved
		5: 0	Reserved
		4: GoSuspend	Go Suspend
		3: SendWakeup	Send Wakeup
		2: 0	Reserved
		1: xINTmode 1	xINT Mode 1
		0: xINTmode 0	xINT Mode 0

BIT4 Go Suspend

Setting this bit to HIGH places this IC in the Suspend mode.
The clock supplying to internal circuits and external elements does not stop.
This bit is used only for enabling USB Resume interrupt.

BIT3 Send Wakeup

Setting this bit to HIGH restores to the previous state from the Suspend state.

BIT1 xINT Mode1

Determines the operation mode of the XINTU signal.
0: Outputs LOW when asserting interrupt and HIGH when negating it.
1: Outputs LOW when asserting interrupt and becomes "Hi-Z" condition when negating it.

BIT0 xINT Mode0

Determines the operation mode of the XINTS signal.
0: Outputs LOW when asserting interrupt and HIGH when negating it.
1: Outputs LOW when asserting interrupt and becomes "Hi-Z" condition when negating it.

7.3.11 USB Common(USBCommon) R/W

Sets operation of the USB interface.

Address	Register Name	Bit Symbol	Description
0Ah	USBCommont	7: VBUS	VBUS
		6: 0	Reserved
		5: 0	Reserved
		4: 0	Reserved
		3: 0	Reserved
		2: IgnrTglMis	Ignore Toggle Mismatch
		1: EnPullUp	Enable Pull Up
		0: ActiveUSB	Active USB Interface

BIT7 VBUS

Shows the state of the VBUS signal.
0: Disconnected state (VBUS = LOW)
1: Connected state (VBUS = HIGH)

BIT2 IgnrTglMis

Sets the operation mode at the time of toggle mismatch in Out transaction.
0: Asserts the OUTransACK status of the relevant endpoint at the time of toggle mismatch.
1: Does not assert the OUTransACK status of the relevant endpoint at the time of toggle mismatch.

BIT1 Enable Pull Up

The XPUENB pin becomes LOW by setting this bit to HIGH after checking the state of connection of the VBUS signal.
Effective when pull up control of the USB bus is performed by XPUENB.
Combining 5V input of VBUS and the AND condition of this bit causes the XPUENB pin to become LOW.

BIT0 Active USB Interface

After resetting, the USB interface ignores all packets until this bit is set.
It activates by setting this bit to HIGH after completion of initialization.

7.3.12 Reset (Reset) R/W

Resets internal LSI. Each block is reset by writing HIGH to the corresponding bit.
Has the same effect as hard reset. Automatically returns to LOW.

Address	Register Name	Bit Symbol	Description
0Dh	Reset	7: 0	Reserved
		6: 0	Reserved
		5: 0	Reserved
		4: 0	Reserved
		3: 0	Reserved
		2: PORT	Port Block Soft Reset
		1: SCSI	SCSI Block Soft Reset
		0: USB	USB Block Soft Reset

7.3.13 Port DMA Control (PortDMACtrl) R/W

Performs transfer operation of the port DMA.

Address	Register Name	Bit Symbol	Description
10h	PortDMACtrl	7: DMAmode1	DMA mode1
		6: DMAmode0	DMA mode0
		5: 0	Reserved
		4: 0	Reserved
		3: 0	Reserved
		2: 0	Reserved
		1: S_FIFO	SCSI FIFO Control
		0: DTGO	DMA Transfer Go

BIT7-6 DMA mode1-0

Sets the DMA transfer mode.

mode1	mode0	Transfer mode
0	0	Performs DMA transfer between the port and USB
0	1	Performs DMA transfer between the port and SCSI
1	0	Performs DMA transfer between SCSI and USB
1	1	Reserve (Setting disabled)

BIT1 SCSI FIFO Control

Concurrent setting of this bit and DTGO to HIGH does not cause DMA transfer by hardware. Instead, the CPU transfers while monitoring the state of SCSI-FIFO (31-32H). The CPU must read or write data from or into SCSI-FIFO according to the FULL/EMPTY status of SCSI-FIFO. Alternatively, data may be written into SCSI-FIFO first to write HIGH into this bit and DTGO and to control, using the remaining data and FULL/EMPTY. However, the CPU must not access FIFO in the direction opposite to transfer.

This bit is valid only in mode1:0 = 0:1 or 1:0. So use it when only SCSI needs to be transferred.

BIT0 DMA Transfer Go

Starts DMA transfer. The target transfer is specified by mode1:0 BIT.

The direction of transfer is automatically determined from the OUTxIN BIT set in each endpoint for USB or from the command issued for SCSI

7.3.14 Port DMA Size High (PortDMASize_H) R/W

Sets the most significant byte of the byte-length (3 bytes) for port DMA transfer.

Address	Register Name	Bit Symbol	Description
11h	PortDMASize_H	7: PortDMASize[23] 6: PortDMASize[22] 5: PortDMASize[21] 4: PortDMASize[20] 3: PortDMASize[19] 2: PortDMASize[18] 1: PortDMASize[17] 0: PortDMASize[16]	Port DMA Size High

7.3.15 Port DMA Size Middle (PortDMASize_M) R/W

Sets the second byte of the byte-length (3 bytes) for port DMA transfer.

Address	Register Name	Bit Symbol	Description
12h	PortDMASize_M	7: PortDMASize[15] 6: PortDMASize[14] 5: PortDMASize[13] 4: PortDMASize[12] 3: PortDMASize[11] 2: PortDMASize[10] 1: PortDMASize[9] 0: PortDMASize[8]	Port DMA Size Middle

7.3.16 Port DMA Size Low (PortDMASize_L) R/W

Sets the least significant byte of the byte-length (3 bytes) for port DMA transfer.

Address	Register Name	Bit Symbol	Description
13h	PortDMASize_L	7: PortDMASize[7] 6: PortDMASize[6] 5: PortDMASize[5] 4: PortDMASize[4] 3: PortDMASize[3] 2: PortDMASize[2] 1: PortDMASize[1] 0: PortDMASize[0]	Port DMA Size Low

7.3.17 Port Config 0 (PortConfig_0) R/W

Sets the operation mode of the IC.

Address	Register Name	Bit Symbol	Description
14h	PortConfig_0	7:ActivePort	Active Port
		6: BUSC	Bus Configuration
		5:PortSlave	Port Slave
		4:0	Reserved
		3:PDREQlevel	PDREQ level
		2:Swap	Swap Port Interface Bus
		1:OddStart	Odd Byte Start
		0:Bus8	Port Interface 8 bit Bus

BIT7 Active Port

After reset, the port interface is in All Pins Input mode. Setting this bit to HIGH activates the port.

BIT6 Bus Configuration

Setting this bit to HIGH causes the listing order of DATA BUS of PORT interface in ascending order.

PIN No. / Bus Configuration	39	40	41	42	43	45	46	47	48	49	52	53	54	55	56	57
LOW	PD15	PD0	PD14	PD1	PD13	PD2	PD12	PD3	PD11	PD4	PD10	PD5	PD9	PD6	PD8	PD7
HIGH	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	PD9		PD11	PD12	PD13	PD14	PD15

BIT5 Port Slave

Determines the operation mode of the port interface.
0: Master mode (PDREQ = input, XPDACK/XPRD/XPWR = output)
1: Slave mode (PDREQ = output, XPDACK/XPRD/XPWR = input)

BIT3 PDREQ level

Determines the operation level of the PDREQ signal.
0: Positive logic
1: Negative logic

BIT2 Swap Port Interface Bus

Swaps higher order 8 bits with the lower ones when the port interface with 16 bits wide is used.
0: Higher order 1 byte data appears in PD[7:0], lower order 1 byte data in PD[15:8].
Lower order 1 byte data is transferred first.
1: Higher order 1 byte data appears in PD[15:8], lower order 1 byte data in PD[7:0].
Higher-order 1 byte data is transferred first.

BIT1 Odd Byte Start

Setting this bit to HIGH causes the 8 bit-data to be transferred first, which is due to be sent later, because of the SWAP setting when the port interface is used with 16 bits wide.
It is effective only for the first one byte only.

BIT0 Port Interface 8 bit Bus

Sets this bit to HIGH to use the port interface with 8 bits wide.
Only the lower 8 bits are valid. Connect the higher 8 bits to GND or HV_{DD} externally.

* Operation settings of the port interface
The following list shows the operational settings made by the bit setting:

1) Selecting master/slave of the port by PSLV bit

	PDREQ	XPDACK	XPRD/XPWR	Remarks
PSLV=0 (Master)	Input	Output	Output	Data input during XPRD Data output during XPWR PortConfig_1 register setting enabled XPRD/XPWR pulse width: Assert≥40ns Negate≥40ns
PSLV=1 (Slave)	Output	Input	Input	Data output during XPRD Data input during XPWR PortConfig_1 register setting disabled XPRD/XPWR pulse width: Assert≥30ns Negate≥30ns

2) Selecting operation modes by BUS8/SWAP/ODS bit

BUS8=0	SWAP=0	PD7 to 0 is transferred first. If ODS = 1, PD7 to 0 is discarded when the first one word is transferred, and only PD15 to 8 is transferred. PD7 to 0 is used when the last data to be transferred is not a word but a byte.
	SWAP=1	PD15 to 8 is transferred first. If ODS = 1, PD15 to 8 is discarded when the first one word is transferred, and only PD7 to 0 is transferred. PD15 to 8 is used when the last data to be transferred is not a word but a byte.
BUS8=1		Only PD7 to 0 is used for transfer. PD15 to 8 goes into Input mode (connect to GND or HV_{DD}).

7.3.18 Port Config 1 (PortConfig_1) R/W

Sets the operation mode of the port interface.

Address	Register Name	Bit Symbol	Description
15h	PortConfig_1	7: AssertPulseWidth[3] 6: AssertPulseWidth[2] 5: AssertPulseWidth[1] 4: AssertPulseWidth[0]	Assert Pulse Width
		3: NegatePulseWidth[3] 2: NegatePulseWidth[2] 1: NegatePulseWidth[1] 0: NegatePulseWidth[0]	Negate Pulse Width

BIT7-4 Assert Pulse Width

Sets the assert pulse width of XPRD/XPWR when the port interface operates in Master mode.
The width is the internal operation clock cycle (40 MHz) multiplied by [AssertPulseWidth + 2].
ex. 0000: 2×25ns=50ns
0001: 3×25ns=75ns

BIT3-0 Negate Pulse Width

Sets the negate pulse width of XPRD/XPWR when the port interface operates in Master mode.
The width is the internal operation clock cycle (40 MHz) multiplied by [NegatePulseWidth + 2].
ex. 0000: 2×25ns=50ns
0001: 3×25ns=75ns

7.3.19 USB Index (USBIndex) R/W

Set registers to be displayed in USBWindow_0 to USBWindow_7.
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
17h	USBIndex	7: USBIndex[7] 6: USBIndex[6] 5: USBIndex[5] 4: USBIndex[4] 3: USBIndex[3] 2: USBIndex[2] 1: USBIndex[1] 0: USBIndex[0]	USB Index

7.3.20 USB Window 0 (USBWindow_0) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
18h	USBWindow_0	7: USBWindow_0[7] 6: USBWindow_0[6] 5: USBWindow_0[5] 4: USBWindow_0[4] 3: USBWindow_0[3] 2: USBWindow_0[2] 1: USBWindow_0[1] 0: USBWindow_0[0]	USB Window 0

7.3.21 USB Window 1 (USBWindow_1) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
19h	USBWindow_1	7: USBWindow_1[7] 6: USBWindow_1[6] 5: USBWindow_1[5] 4: USBWindow_1[4] 3: USBWindow_1[3] 2: USBWindow_1[2] 1: USBWindow_1[1] 0: USBWindow_1[0]	USB Window 1

7.3.22 USB Window 2 (USBWindow_2) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
1Ah	USBWindow_2	7: USBWindow_2[7] 6: USBWindow_2[6] 5: USBWindow_2[5] 4: USBWindow_2[4] 3: USBWindow_2[3] 2: USBWindow_2[2] 1: USBWindow_2[1] 0: USBWindow_2[0]	USB Window 2

7.3.23 USB Window 3 (USBWindow_3) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
1Bh	USBWindow_3	7: USBWindow_3[7] 6: USBWindow_3[6] 5: USBWindow_3[5] 4: USBWindow_3[4] 3: USBWindow_3[3] 2: USBWindow_3[2] 1: USBWindow_3[1] 0: USBWindow_3[0]	USB Window 3

7.3.24 USB Window 4 (USBWindow_4) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
1Ch	USBWindow_4	7: USBWindow_4[7] 6: USBWindow_4[6] 5: USBWindow_4[5] 4: USBWindow_4[4] 3: USBWindow_4[3] 2: USBWindow_4[2] 1: USBWindow_4[1] 0: USBWindow_4[0]	USB Window 4

7.3.25 USB Window 5 (USBWindow_5) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
1Dh	USBWindow_5	7: USBWindow_5[7] 6: USBWindow_5[6] 5: USBWindow_5[5] 4: USBWindow_5[4] 3: USBWindow_5[3] 2: USBWindow_5[2] 1: USBWindow_5[1] 0: USBWindow_5[0]	USB Window 5

7.3.26 USB Window 6 (USBWindow_6) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
1Eh	USBWindow_6	7: USBWindow_6[7] 6: USBWindow_6[6] 5: USBWindow_6[5] 4: USBWindow_6[4] 3: USBWindow_6[3] 2: USBWindow_6[2] 1: USBWindow_6[1] 0: USBWindow_6[0]	USB Window 6

7.3.27 USB Window 7 (USBWindow_7) R/W

Displays the USB-related register.
The register to be displayed changes according to the value set at USBIndex register (17h).
For details of display by set values of the USBIndex register, refer to section "7.5 Detailed Description of Set Values of USBIndex Register."

Address	Register Name	Bit Symbol	Description
1Fh	USBWindow_7	7: USBWindow_7[7] 6: USBWindow_7[6] 5: USBWindow_7[5] 4: USBWindow_7[4] 3: USBWindow_7[3] 2: USBWindow_7[2] 1: USBWindow_7[1] 0: USBWindow_7[0]	USB Window 7

7.3.28 Main Interrupt Status SCSI (MAININTS) R/W

When the IC causes an SCSI interruption to the CPU, the CPU identifies the interrupt status register responsible for interruption by reading this register first.
Following the reading of this register, the CPU reads the interrupt status register corresponding to each bit and performs appropriate interrupt processing by identifying the bit as a source of interruption. Then, it writes the value read to the interrupt status register corresponding to each bit to clear the relevant bit.
If GOOD, SABT, or DTCMP bit is a source of interruption, the CPU writes the value read to clear the bits. The register does not need to directly clear any other bits.

Address	Register Name	Bit Symbol	Description
20h	MAININTS	7: GOOD	SCSI COMMAND NORMAL COMPLETE
		6: SABT	ABORTED SCSI COMMAND
		5: EXEC	EXECUTING SCSI COMMAND
		4: SCSI1	SCSI INTERRUPT STATUS 1
		3: SCSI2	SCSI INTERRUPT STATUS 2
		2: —	
		1: DTCMP	DMA TRANSFER COMPLETE
		0: ASCMP	AUTO SEQUENCE COMPLETE

BIT7 SCSI COMMAND NORMAL COMPLETE

When the SCSI control command is terminated normally, this bit becomes HIGH.

BIT6 ABORTED SCSI COMMAND

When the control command is forced to terminate by issuing Abort command, this bit becomes HIGH.

BIT5 EXECUTING SCSI COMMAND

This bit is HIGH during execution of the SCSI control command. This bit does not cause an interruption to the CPU. Even when it becomes HIGH no interruption is output. It is used to monitor the state of execution of the SCSI control command.

BIT4 SCSI INTERRUPT STATUS 1

This bit becomes HIGH when there is any factor responsible for an interruption related to the SCSI interface shown in the SCSIINT1 register.

BIT3 DMA INTERRUPT STATUS 2

This bit becomes HIGH when there is any factor responsible for interruption related to the SCSI interface shown in the SCSIINT2 register.

BIT1 DMA TRANSFER COMPLETE

This bit becomes HIGH when DMA data transfer activated by the DMACTL register is completed.
It also becomes HIGH when the transfer is forced to terminate by writing “0” into the DTGO bit of the DMACTL register.
After setting the DTGO bit of the DMACTL register, this bit becomes HIGH when a command is aborted by the Abort_SCSI command or when a command in the course of execution is aborted by the ATN assertion, because DMA terminates.
At times other than mode1:0=“00” of the PortDMACtrl register, XINTU is set by this factor. When mode1:0=“00”, XINTS is not set by this factor.

* This bit has the same meaning as that of BIT(PortDMACmp) of MAINT(00h). It can be cleared by using any of the registers.

BIT0 AUTO SEQUENCE COMPLETE

This bit becomes HIGH when AUTO1 or AUTO2 bit of the SCSIMODE0 register is set and the specified command is terminated.

7.3.29 SCSI Interrupt Status 1 (SCSIINT1) R/W

Shows the result of executing the SCSI control command.
The CPU identifies the source of interruption by reading this register after receiving the interrupt signal. It clears the bit by writing the value read again.

Address	Register Name	Bit Symbol	Description
21h	SCSIINT1	7: SPERR	SCSI DATA PARITY ERROR DETECTED
		6: IDERR	ID ERROR DETECTED
		5: SELTO	SELECTION TIME OUT
		4: SATN	SCSI ATN ASSERTION DETECTED
		3: BFREE	BUS FREE DETECTED
		2: ILPHS	ILLEGAL PHASE CHANGE DETECTED
		1: SCSEL	SCAM SELECTED
		0: WOATN	SELECTED WITHOUT ATTENTION

BIT7 SCSI DATA PARITY ERROR DETECTED
This bit becomes HIGH when a parity error is detected on the SCSI data bus.
BIT6 ID ERROR DETECTED
This bit becomes HIGH when an error related to the ID bit is detected during the selection or reselection phase. The errors related to the ID bit are as described below:
・Only one ID bit is asserted, or
・Three or more ID bits are asserted.
BIT5 SELECTION TIME OUT
This bit becomes HIGH when time-out is detected during the selection or reselection phase.
BIT4 SCSI ATN ASSERTION DETECTED
This bit becomes HIGH when SCSI ATN is asserted. It is not set, however, in the regular sequence of assertion of SCSI ATN. In other words, this bit is not set in the message-out phase immediately following selection.
BIT3 BUS FREE DETECTED
This bit becomes HIGH when the bus-free phase is detected during execution of the SCSI control command.
BIT2 ILLEGAL PHASE CHANGE DETECTED
This bit becomes HIGH when unexpected phase transition is detected during execution of the SCSI control command. It is valid only in Initiator mode.
BIT1 SCAM SELECTED
This bit becomes HIGH when the SCAM selection is responded to.
BIT0 SELECTED WITHOUT ATTENTION
This bit becomes HIGH when the selection, which is not asserted ATTENTION, is responded to.
Execution of the SCSI control command continues, even when this bit is set. If a command block is received continuously, however, the first byte of SCSI-FIFO has the command code.

7.3.30 SCSI Interrupt Status 2 (SCSIINT2) R/W

Shows the result of executing the SCSI control command.
The CPU identifies the source of interruption by reading this register after receiving the interrupt signal. It clears the bit by writing the value read again.

Address	Register Name	Bit Symbol	Description
22h	SCSIINT2	7: —	
		6: SRST	SCSI RST ASSERTION DETECTED
		5: OFERR	OFFSET ERROR IN SYNCHRONOUS TRANSFER
		4: UNDEF	UNDEFIND GROUP COMMAND
		3: CMDER	COMMAND ERROR
		2: RESEL	RESELECTED
		1: SEL	SELECTETD
		0: LARBT	LOST ARBITRATION

BIT7 RESERVED

BIT6 SCSI RST ASSERTION DETECTED

This bit becomes HIGH when SCSI RST is asserted.

BIT5 OFFSET ERROR IN SYNCHRONOUS TRANSFER

This bit becomes HIGH when an offset error occurs during synchronous transfer. The offset error means that the offset counter is not reset to "0" when transfer ends or that the counter overflows/underflows.

BIT4 UNDEFIND GROUP COMMAND

This bit becomes HIGH when SCSI command other than group 0, 1, 2, or 5 is received.

BIT3 COMMAND ERROR

This bit becomes HIGH when other control command is issued during execution of the command, although an undefined SCSI control command is issued.

BIT2 RESELECTED

This bit becomes HIGH when a re-selection is made from other device during execution of a command other than the SCSI control command that makes a re-selection.

BIT1 SELECTED

This bit becomes HIGH when selected from other device during execution of a command other than SCSI control command that makes a selection.

BIT0 LOST ARBITRATION

This bit becomes HIGH when defeated in the arbitration phase. When this bit is HIGH and it is not selected or re-selected by other device, the IC suspends operation of the control commands.

7.3.31 SCSI Mode Select0 (SCSIMODE0) R/W

Sets operation related to the SCSI interface.

Address	Register Name	Bit Symbol	Description
29h	SCSIMODE0	7: —	
		6: —	
		5: —	
		4: ULTRA	ULTRA SCSI
		3: AUTO1	AUTO1 (auto status)
		2: AUTO2	AUTO2 (status message stop)
		1: AN_C	ACTIVE NEGATION CONTROL
		0: AN_D	ACTIVE NEGATION DATA

BIT4 ULTRA SCSI

When this bit is HIGH, the ULTRA-SCSI transfer is enabled. It is valid only when the RATE bit of the SYNCMODE register is set at "0" or "1".

BIT3 AUTO1 (auto status)

Automatically executes STS_MSG→Busfree→Wait_SEL_CMD after executing the DMA_DATA_IN/OUT command.

At the end of execution, ASCMP interrupt occurs. The bit setting is also valid in FIFO-DMA mode.

BIT2 AUTO2 (status message stop)

Executes automatically STS_MSG after executing the DMA_DATA_IN/OUT command.

At the end of execution, ASCMP interrupt occurs. The bit setting is also valid in FIFO-DMA mode.

*AUTO: During execution of AUTO1 or 2, the AUTO bit of the SCSIMODE1 register is assumed to be "1".
The EXEC bit also becomes "1" during execution of AUTO1 or 2. Since the internal sequencer writes the command into the SCSI block in place of the CPU, the COMMAND register can read the command value in process of execution at that time.

BIT1 ACTIVE NEGATION CONTROL

When this bit is HIGH it enables the active negation function of the SCSI XSREQ/XSACK signals.

BIT0 ACTIVE NEGATION DATA

When this bit is HIGH it enables the active negation function of the SCSI data and parity signals.

7.3.32 SCSI Mode Select1 (SCSIMODE1) R/W

Sets operation related to the SCSI interface.

Address	Register Name	Bit Symbol	Description
2Ah	SCSIMODE1	7: STPPE	STOP BY PARITY ERROR
		6: ATNPE	ATN ASSERT BY PARITY ERROR
		5: STATN	STOP BY ATN ASSERT
		4: AUTO	AUTO SEND STATUS/MESSAGE
		3: RINH	RESELECTION INHIBIT
		2: SINH	SELECTION INHIBIT
		1: DIRECT	SCSI DIRECT ACCESS ENABLE
		0: SPCEN	SCSI PARITY CHECK ENABLE

BIT7 STOP BY PARITY ERROR

When this bit is HIGH it suspends the SCSI control command in process of execution if a parity error is detected on the SCSI interface.

BIT6 ATN ASSERT BY PARITY ERROR

When this bit is HIGH it asserts a target device ATN if a parity error is detected on the SCSI interface. Any setting of this bit becomes invalid when the SCSI parity check is disabled.

This bit is valid only in Initiator mode.

BIT5 STOP BY ATN ASSERT

When this bit is HIGH it suspends the SCSI control command in process of execution if assertion of ATN is detected.

This bit is valid only in Target mode.

BIT4 AUTO SEND STATUS/MESSAGE

When this bit is HIGH it puts the SCSI control command “Status_Message” into Automatic Transmission mode. In this mode, the FLAG and LINK bits of the SCSI command block which have been received are checked. When the LINK bit is LOW status 00h(GOOD) and message 00h(COMMAND COMPLETE) are automatically sent. When the LINK bit is LOW status 10h(INTERMEDIATE GOOD) and message 0Ah(LINKED COMMAND COMPLETE) are automatically sent. When both LINK bit and FLAG bit are HIGH status 10h(INTERMEDIATE GOOD) and message 0Bh(LINKED COMMAND COMPLETE WITH FLAG) are automatically sent.

BIT3 RESELECTION INHIBIT

When this bit is HIGH it disables response to re-selection.

BIT2 SELECTION INHIBIT

When this bit is HIGH it disables response to selection.

BIT1 SCSI DIRECT ACCESS ENABLE

When this bit is set to HIGH SCSI signal lines of this bit can be directly controlled from the CPU by using the SCSI data register and SCSI control register. The status of the signal lines can always be monitored regardless of the status of this bit.

BIT0 SCSI PARITY CHECK ENABLE

When this bit is HIGH parity check of the SCSI data bus is performed during the selection phase (when it is selected) and when data from SCSI is input.

7.3.33 SCSI Control (SCSICTL) R/W

This register is accessed when the CPU directly controls SCSI signal lines.
For such direct control, DIRECT (bit 1) must be set in the mode setting register (0Ah).
The status of each signal is stored as "active high".

Address	Register Name	Bit Symbol	Description
2Bh	SCSICTL	7: ACK	SCSI ACK
		6: ATN	SCSI ATN
		5: SEL	SCSI SEL
		4: BSY	SCSI BSY
		3: REQ	SCSI REQ
		2: MSG	SCSI MSG
		1: I/O	SCSI I/O
		0: C/D	SCSI C/D

7.3.34 SCSI Data (SCSIDATA) R/W

This register is accessed when the CPU directly controls the SCSI data bus. For such direct control, DIRECT (bit 1) must be set in the mode setting register (0Ah).
The status of each signal is stored as "active high". The DIRECT setting does not determine whether the parity bit is output or not. It is output if it has been output before setting DIRECT, or it is not otherwise.

Address	Register Name	Bit Symbol	Description
2Ch	SCSIDATA	7: DB7	
		6: DB6	
		5: DB5	
		4: DB4	SCSIDATA
		3: DB3	
		2: DB2	
		1: DB1	
		0: DB0	

7.3.35 Synchronize Transfer Mode (SYNCMODE) R/W

Sets the transfer rate and offset for SCSI synchronous transfer.

Address	Register Name	Bit Symbol	Description
2Dh	SYNCMODE	7: RATE3	SYNCHRONOUS TRANSFER RATE[3]
		6: RATE2	SYNCHRONOUS TRANSFER RATE[2]
		5: RATE1	SYNCHRONOUS TRANSFER RATE[1]
		4: RATE0	SYNCHRONOUS TRANSFER RATE[0]
		3: OFF3	SYNCHRONOUS OFFSET[3]
		2: OFF2	SYNCHRONOUS OFFSET[2]
		1: OFF1	SYNCHRONOUS OFFSET[1]
		0: OFF0	SYNCHRONOUS OFFSET[0]

RATE3-0	ASSERT	NEGATE	PERIOD
0000	1T(1T)	1T(1T)	2T
0001	2T(1T)	1T(2T)	3T
0010	2T	2T	4T
0011	3T	2T	5T
0100	3T	3T	6T
0101	4T	3T	7T
0110	4T	4T	8T
0111	5T	4T	9T
1000	5T	5T	10T
1001	6T	5T	11T
1010	6T	6T	12T
1011	7T	6T	13T
1100	7T	7T	14T
1101	8T	7T	15T
1110	8T	8T	16T
1111	9T	8T	17T

Note 2

OFF3-0	OFFSET
0000	Asynchronous
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Note 1) T is a cycle double the internal clock (40MHz).

Note 2) When the ULTRA bit of the SCSIMODE0 register is set, T has the same cycle as the enclosed value of the internal clock (40MHz) only if the value set for the RATE3 to 0 bit is 1 or less.

7.3.36 SCSI Own ID (OWNID) R/W

Sets the SCSI-ID of this IC itself.

Address	Register Name	Bit Symbol	Description
2Eh	OWNID	7: —	
		6: —	
		5: —	
		4: —	
		3: —	
		2: OID2	(MSB)
		1: OID1	SCSI OWN ID
		0: OID0	(LSB)

7.3.37 Source/Destination ID (SDID) R/W

Sets both the SCSI-ID of the selector side and the target SCSI-ID when selection is made.

Address	Register Name	Bit Symbol	Description
2Fh	SDID	7: —	
		6: SID2	SOURCE ID[2] (R/W)
		5: SID1	SOURCE ID[1] (R/W)
		4: SID0	SOURCE ID[0] (R/W)
		3: —	
		2: DID2	DESTINATION ID[2] (R)
		1: DID1	DESTINATION ID[1] (R)
		0: DID0	DESTINATION ID[0] (R)

In Initiator mode, sets the target SCSI-ID to be selected in DESTINATION ID.
When re-selection is received, the target SCSI-ID that makes a re-selection is set in SOURCE ID.
In Target mode, sets the initiator SCSI-ID to be re-selected in DESTINATION ID.
When selection is received, the initiator SCSI-ID that makes a selection is set in SOURCE ID.

7.3.38 Selection Timeout Counter (SLTIME) R/W

Sets time-out delay for selection and re-selection.

Address	Register Name	Bit Symbol	Description
30h	SLTIME	7: ST7	(MSB)
		6: ST6	
		5: ST5	
		4: ST4	Selection Time out Counter
		3: ST3	
		2: ST2	
		1: ST1	
		0: ST0	(MSB)

The time-out delay value is calculated according to the following formula:

Delay value = count value × 2^{15} × T × 2

Where T is an internal clock cycle (40MHz).

Detecting time-out causes the following operation of the IC:

- Suspends output ID bit.
- Negates XSSEL 4000×T×2 (about 200μs) after such suspension and outputs selection time-out interrupt.

No time-out is detected when this register is set to "0."

7.3.39 FIFO Control (FIFOCTL) R/W

Used for clearing the SCSI-FIFO data and for checking its status.

Address	Register Name	Bit Symbol	Description
31h	FIFOCTL	7: —	
		6: —	
		5: —	
		4: —	
		3: —	
		2: FCLR	CLEAR FIFO
		1: FULL	FIFO FULL
		0: EMPTY	FIFO EMPTY

BIT7,6,5,4,3 RESERVED

BIT2 CLEAR FIFO

Setting this bit to HIGH clears data stored in SCSI-FIFO.
The bit returns to LOW automatically after clearing.

BIT1 FULL

When this bit is HIGH it means that SCSI-FIFO is full. In this state, any data written into SCSI-FIFO is ignored.

BIT0 EMPTY

When this bit is HIGH it means that SCSI-FIFO is empty. In this state, any attempt to read data from SCSI-FIFO results in invalid data read out.

7.3.40 FIFO Data (FIFODATA) R/W

This register allows access to SCSI-FIFO from the CPU.

Address	Register Name	Bit Symbol	Description
32h	FIFODATA	7: FD7	(MSB)
		6: FD6	
		5: FD5	
		4: FD4	SCSI_FIFO data
		3: FD3	
		2: FD2	
		1: FD1	
		0: FD0	(LSB)

7.3.41 Non DMA Transfer Size (NDMASIZ) R/W

This register sets the number of bytes of data transfer in Non-DMA mode. In Read mode, the register allows to read out the size of data yet to be transferred.

Address	Register Name	Bit Symbol	Description
33h	NDMASIZ	7: NSZ7	(MSB)
		6: NSZ6	
		5: NSZ5	
		4: NSZ4	Non DMA Transfer Size
		3: NSZ3	
		2: NSZ2	
		1: NSZ1	
		0: NSZ0	(LSB)

7.3.42 SCSI Command (COMMAND) R/W

Sets SCSI control commands.

Address	Register Name	Bit Symbol	Description
34h	COMMAND	7: CMD7	(MSB)
		6: CMD6	
		5: CMD5	
		4: CMD4	SCSI Command
		3: CMD3	
		2: CMD2	
		1: CMD1	
		0: CMD0	(LSB)

For details of each command, refer to section "7.4 SCSI Control Commands."

7.3.43 Test (TEST) R

Used for testing an LSI. Basically, writing into this register is inhibited.

Address	Register Name	Bit Symbol	Description
3Eh	TEST	7: TM2	—
		6: TM1	—
		5: TM0	—
		4: USEL1	—
		3: USEL0	—
		2: OFST	—
		1: SCBC	—
		0: DMBC	—

7.3.44 Revision Reg.(REVISION) R

Shows the revision No. of the IC.

Address	Register Name	Bit Symbol	Description
3Fh	REVISION	7: REV7	(MSB)
		6: REV6	
		5: REV5	
		4: REV4	REVISION
		3: REV3	
		2: REV2	
		1: REV1	
		0: REV0	(LSB)

In normal Read mode, the register allows to read out the revision No. of this model.
The model name SPC number (72h, 15h) can be read by reading out twice after writing any value.

7.4 Detailed Description of Set Values of INTINDEX Register

7.4.1 Register Showing IntStatWindow_0,1 and IntEnbWindow_0,1 for Set Values of IntIndex Register

IntIndex_n	IntStatWindow_n	IntEnbWindow_n
00h	EP0IntStat	EP0IntEnb
01h	EPaIntStat	EPaIntEnb
02h	EPbIntStat	EPbIntEnb
03h	EPcIntStat	EPcIntEnb
04h	Reserved	Reserved
05h	Reserved	Reserved
06h	Reserved	Reserved
07h	Reserved	Reserved

*n=0 or 1

7.4.2 EP {r} (r=0,a,b,c) Interrupt Status (EP {r} IntStat) R/W

Shows the interrupt status of each interrupt status register endpoint displayed in IntStatWindow_0,1.
Following the reading of this register, the bit of a source of interruption is identified and it is cleared by writing the value read after appropriate interrupt processing.
(The relevant bit is cleared by writing HIGH to each bit).

IntIndex_n: 0h to 3h			
Address	**Register Name**	**Bit Symbol**	**Description**
02h,03h	EP0IntStat, EPaIntStat, EPbIntStat, EPcIntStat,	7: INtranACK	IN Transaction ACK
		6: OUTtranACK	OUT Transaction ACK
		5: INtranErr	IN Transaction Error
		4: OUTtranErr	OUT Transaction Error
		3: INtranNAK	IN Transaction NAK
		2: OUTtranNAK	OUT Transaction NAK
		1: INtokenRcv	IN Token Received
		0: OUTtokenRcv	OUT Token Received

BIT7 IN Transaction ACK
This bit becomes HIGH when ACK is received during IN Transaction.
BIT6 OUT Transaction Complete
This bit becomes HIGH when OUT Transaction is normally completed.
BIT5 IN Transaction Error
This bit becomes HIGH when IN Transaction ends abnormally.
Details can be obtained from the USBTransStatus register (0Bh).
BIT4 OUT Transaction Error
This bit becomes HIGH when OUT Transaction ends abnormally.
Details can be obtained from the USBTransStatus register (0Bh).
BIT3 IN Transaction NAK
This bit becomes HIGH when NAK is returned during IN Transaction.
BIT2 OUT Transaction NAK
This bit becomes HIGH when NAK is returned during OUT Transaction.
BIT1 IN Token Received
This bit becomes HIGH when BIT1 IN Token Received IN Token is received.
BIT0 OUT Token Received
This bit becomes HIGH when OUT Token is received.

7.4.3 EP {r} (r=0,a,b,c) Interrupt Enable (EP {r} IntEnb) R/W

Appears in IntEnbWindow 0,1. This register enables/disables endpoint interruption shown in IntStatWindow_0,1.
When the corresponding bit is set to HIGH an interruption to the CPU is enabled.

IntIndex_n: 0h to 3h			
Address	**Register Name**	**Bit Symbol**	**Description**
07h,08h	EP0IntStat, EPaIntStat EPbIntStat ,EPcIntStat	7: EnINtranACK	Enable IN Transaction ACK
		6: EnOUTtranCmp	Enable OUT Transaction Complete
		5: EnINtranErr	Enable IN Transaction Error
		4: EnOUTtranErr	Enable OUT Transaction Error
		3: EnINtranNAK	Enable IN Transaction NAK
		2: EnOUTtranNAK	Enable OUT Transaction NAK
		1: EnINtokenRcv	Enable IN Token Received
		0: EnOUTtokenRcv	Enable OUT Token Received

7.5 Detailed Description of Set Values of USBIndex Register

7.5.1 List of Registers Showing USBWindow Register (8 bytes) Corresponding to Set Values of USBIndex Register(17h)

USBIndex	Address	Register Name
00h	18h	USBAddress
	19h	EP0Config_1
	1Ah	EP0InControl
	1Bh	EP0OutControl
	1Ch	(Reserved)
	1Dh	EP0FIFOremain
	1Eh	EP0FIFOforCPU
	1Fh	EP0FIFOCtrl
01h to 07h	18h	EPrConfig_0
	19h	EPrConfig_1
	1Ah	EPrControl
	1Bh	(Reserved)
	1Ch	(Reserved)
	1Dh	EP{r}FIFOremain
	1Eh	EP{r}FIFOforCPU
	1Fh	EP{r}FIFOCtrl
08h	18h	EP0_SETUP [0]
	19h	EP0_SETUP [1]
	1Ah	EP0_SETUP [2]
	1Bh	EP0_SETUP [3]
	1Ch	EP0_SETUP [4]
	1Dh	EP0_SETUP [5]
	1Eh	EP0_SETUP [6]
	1Fh	EP0_SETUP [7]
09h	18h	FrameNumber_H
	19h	FrameNumber_L
	1Ah	(Reserved)
	1Bh	(Reserved)
	1Ch	(Reserved)
	1Dh	(Reserved)
	1Eh	(Reserved)
	1Fh	(Reserved)

*{r}=a,b,c

7.5.2 Description of Registers by Set Value of USBIndex

7.5.2.1 USB Address (USBAddress) R/W

Sets the USB Address.
This setting is cleared (set 00h) when the USB BusReset is detected.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
18h	USBAddress	7: 0	Reserved
		6: USBAddress[6]	USB Address
		5: USBAddress[5]	
		4: USBAddress[4]	
		3: USBAddress[3]	
		2: USBAddress[2]	
		1: USBAddress[1]	
		0: USBAddress[0]	

BIT7 RESERVED
BIT6-0
Sets the USB Address.
This setting is cleared (set 00h) when the USB BusReset is detected.
Set the address specified by the host when control transfer of the SetAddress request is completed.

7.5.2.2 EP0 Config 1 (EP0Config_1) R/W

Sets operation of the Endpoint 0.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
19h	EP0Config_1	7: OUT × IN	OUT × IN
		6: 0	Reserved
		5: 0	Reserved
		4: 0	Reserved
		3: MaxPacketSize[3]	Max Packet Size
		2: MaxPacketSize[2]	
		1: MaxPacketSize[1]	
		0: MaxPacketSize[0]	

BIT7 OUT × IN
Sets the direction of transfer of Endpoint 0.
Set the direction of transfer in data stage by interpreting the request in SETUP stage.
After completion of setting of the direction of the data stage, it can be executed by clearing the InForceNAK bit or OutForceNAK bit to LOW in either of the EP0InControl or EP0OutControl register, which controls the direction of data stage.
0: IN direction
1: OUT direction
BIT3-0 Max Packet Size
Sets MaxPacketSize of Endpoint 0.
The list corresponding to these bits is as follows:
After setting this field, AllFIFOClr in the EP0FIFOCtrl register must be set to HIGH once.
(bit3, bit2, bit1, bit0)

0	0	0	1	: 8 bytes
0	0	1	0	: 16 bytes
0	0	0	0	: 32 bytes
1	0	0	0	: 64 bytes

7.5.2.3 EP0 In Transaction Control (EP0InControl) R/W

Sets operation to IN Transaction.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Ah	EP0InControl	7: InForceNAK	IN Transaction Force NAK
		6: InForceSTALL	IN Transaction Force STALL
		5: InEnShortPkt	IN Transaction Short Packet Enable
		4: 0	Reserved
		3: 0	Reserved
		2: InToggleStat	IN Transaction Toggle Status
		1: InToggleClr	IN Transaction Toggle Clear
		0: InToggleSet	IN Transaction Toggle Set

BIT7 IN Transaction Force NAK

Setting this bit to HIGH returns NAK to IN Transaction.

When the RcvEP0SETUP bit of the MainIntStat register is set to HIGH as a result of completing the SETUP stage, this bit is set to HIGH while the RcvEP0SETUP bit is HIGH. Therefore, the RcvEP0SETUP bit must be cleared to LOW to clear this bit to LOW.

If the direction of transfer of data stage is IN, the data stage can be executed by clearing this bit to LOW after setting the direction by the OUTxIN bit of the EP0Config_1 register.

If the direction of transfer of data stage is OUT, the data stage can be executed by clearing this bit to LOW after the status stage is ready.

When there is a transaction that is being executed, setting of this bit a fixed period of time after starting transaction becomes valid from the next transaction.

BIT6 IN Transaction Force STALL

When this bit is set to HIGH it becomes valid, taking precedence over the setting of the INForceNAK bit.

Setting this bit to HIGH returns STALL to IN Transaction.

When the RcvEP0SETUP bit of the MainIntStat register is set to HIGH as a result of completing the SETUP stage, this bit is set to LOW while the RcvEP0SETUP bit is HIGH. Therefore, the RcvEP0SETUP bit must be cleared to LOW to set this bit to HIGH.

When there is a transaction that is being executed, setting of this bit a fixed period of time after starting transaction becomes valid from the next transaction.

BIT5 IN Transaction Short Packet Enable

Setting this bit to HIGH returns data packet to IN Transaction independent of amount of data of FIFO.

When packet transfer is completed after setting this bit to HIGH this bit returns to LOW.

BIT2 IN Transaction Toggle Status

Shows the state of Toggle Sequence bit during IN Transaction.

It is set to 1 when SETUP Token is received.

BIT1 IN Transaction Toggle Clear

Clears the Toggle Sequence bit during IN Transaction to 0.

BIT0 IN Transaction Toggle Set

Sets the Toggle Sequence bit during IN Transaction to 1.

7.5.2.4 EP0 OUT Transaction Control (EP0OutControl) R/W

Sets operation to OUT Transaction.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Bh	EP0OutControl	7: OutForceNAK	IN Transaction Force NAK
		6: OutForceSTALL	IN Transaction Force STALL
		5: 0	Reserved
		4: 0	Reserved
		3: 0	Reserved
		2: OutToggleStat	IN Transaction Toggle Status
		1: OutToggleClr	IN Transaction Toggle Clear
		0: OutToggleSet	IN Transaction Toggle Set

BIT7 OUT Transaction Force NAK

Setting this bit to HIGH returns NAK to IN Transaction.

When the RcvEP0SETUP bit of the MainIntStat register is set to HIGH as a result of completing the SETUP stage, this bit is set to HIGH while the RcvEP0SETUP bit is HIGH. Therefore, the RcvEP0SETUP bit must be cleared to LOW to clear this bit to LOW.

If the direction of transfer of data stage is IN, the data stage can be executed by clearing this bit to LOW after setting the direction by the OUTxIN bit of the EP0Config_1 register.

If the direction of transfer of data stage is OUT, the data stage can be executed by clearing this bit to LOW after the status stage is ready.

When there is a transaction that is being executed, setting of this bit a fixed period of time after starting transaction becomes valid from the next transaction.

BIT6 OUT Transaction Force STALL

When this bit is set to HIGH it becomes valid, taking precedence over the setting of the INForceNAK bit.

Setting this bit to HIGH returns STALL to IN Transaction.

When the RcvEP0SETUP bit of the MainIntStat register is set to HIGH as a result of completing the SETUP stage, this bit is set to LOW while the RcvEP0SETUP bit is HIGH. Therefore, the RcvEP0SETUP bit must be cleared to LOW to set this bit to HIGH.

When there is a transaction that is being executed, setting of this bit a fixed period of time after starting transaction becomes valid from the next transaction.

BIT2 OUT Transaction Toggle Status

Shows the state of Toggle Sequence bit during OUT Transaction.

It is set to 1 when SETUP Token is received.

BIT1 OUT Transaction Toggle Clear

Clears the Toggle Sequence bit during IN Transaction to 0.

BIT0 OUT Transaction Toggle Set

Sets the Toggle Sequence bit during IN Transaction to 1.

7.5.2.5 EP0 FIFO remain Counter (EP0FIFOremain) R

Shows the number of bytes of remaining data in FIFO of Endpoint 0.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Dh	EP0FIFOremain	7: EP0FIFOremainCount[7] 6: EP0FIFOremainCount[6] 5: EP0FIFOremainCount[5] 4: EP0FIFOremainCount[4] 3: EP0FIFOremainCount[3] 2: EP0FIFOremainCount[2] 1: EP0FIFOremainCount[1] 0: EP0FIFOremainCount[0]	EP0 FIFO remain Counter

7.5.2.6 EP0 FIFO for CPU (EP0FIFOforCPU) R/W

This register allows access to FIFO of Endpoint 0 from the CPU.
When the EnFiFOwr bit of the EP0FIFOCtrl register is set, writing is enabled, when the EnFIFOrd bit is set, reading is enabled.
It should be noted that reading or writing decreases the number of data in FIFO.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Eh	EP0FIFOforCPU	7: EP0FIFOdata[7] 6: EP0FIFOdata[6] 5: EP0FIFOdata[5] 4: EP0FIFOdata[4] 3: EP0FIFOdata[3] 2: EP0FIFOdata[2] 1: EP0FIFOdata[1] 0: EP0FIFOdata[0]	EP0 FIFO for CPU

7.5.2.7 EP0 FIFO Control (EP0FIFOCtrl) R/W

Checks the status and sets operation of FIFO of Endpoint 0.

USBIndex : 00h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Fh	EP0FIFOCtrl	7: FIFOEmpty	FIFO Empty
		6: FIFOFull	FIFO Full
		5: FIFOClr	FIFO Clear
		4: ALLFIFOClr	ALL FIFO Clear
		3: 0	Reserved
		2: AutoForceNAK	AutoForceNAK
		1: EnFIFOwr	Enable FIFO Write
		0: EnFIFOrd	Enable FIFO Read

BIT7 FIFO Empty

When this bit is HIGH it indicates that FIFO is empty.
If a reading operation of FIFO is carried out in this state, invalid data is read out.

BIT6 FIFO Full

When this bit is HIGH it indicates that FIFO is full.
If a writing operation into FIFO is carried out in this state, the data is ignored.

BIT5 FIFO Clear

Setting this bit to HIGH clears data stored in FIFO.
The bit returns to LOW automatically after clearing.

BIT4 ALL FIFO Clear

Setting this bit to HIGH clears FIFO of all Endpoints.
When the MaxPacketSize filed or DoubleBuf bit of each Endpoint is set, be sure to set this bit to HIGH once after completion of setting.
The bit returns to LOW automatically after clearing FIFO.

BIT2 AutoForceNAK

Setting this bit to HIGH sets the InForceNAK bit of the EP0InControl register and the OutForceNAK bit of the EP0OutControl register when transaction is completed normally.

BIT1 Enable FIFO Write

Setting this bit to HIGH enables writing data into FIFO from the CPU.

BIT0 Enable FIFO Read

Setting this bit to HIGH enables reading data in FIFO from the CPU.

7.5.2.8 EP {r}(r=a,b,c) Config 0 (EPrConfig_0) R/W

Sets operation of Endpoint a, b, and c.

USBIndex : 01h to 03h			
Address	**Register Name**	**Bit Symbol**	**Description**
18h	EPrConfig_0	7: EndPointNumber[3]	
		6: EndPointNumber[2]	Endpoint Number
		5: EndPointNumber[1]	
		4: EndPointNumber[0]	
		3: OUTxIN	OUT x IN
		2: 0	Reserved
		1: JoinPortDMA	Join Port DMA
		0: Really Used	Really Used

BIT7-4 Endpoint Number

Sets any Endpoint number of 1h to Fh.

BIT3 OUT x IN

Sets the direction of transfer of Endpoint.

0: IN direction

1: OUT direction

BIT1 Join Port DMA

Connects the endpoint to the port DMA.

Connects the endpoint to the port DMA.

The port DMA is connected to the endpoint that last set this bit to HIGH.

When there is no endpoint set to HIGH the port DMA is connected to Endpoint c.

Immediately after resetting, the JointPortDMA bit becomes LOW at all endpoints.

BIT0 Really Used

Setting this bit to HIGH allows to use the endpoints.

When this bit is LOW it ignores access to the endpoint.

Set in accordance with the SetConfiguration request from the host.

7.5.2.9 EP {r}(r=a,b,c) Config 1 (EPrConfig_1) R/W

Sets operation of Endpoints a, b, and c.

USBIndex : 01h to 03h			
Address	**Register Name**	**Bit Symbol**	**Description**
19h	EPrConfig_1	7: DoubleBuf	Double Buffer
		6: 0	Reserved
		5: 0	Reserved
		4: 0	Reserved
		3: MaxPacketSize[3]	
		2: MaxPacketSize[2]	Max Packet Size
		1: MaxPacketSize[1]	
		0: MaxPacketSize[0]	

BIT7 Double Buffer

Setting this bit to HIGH makes FIFO a double buffer, securing the area doubling the value of the size set for MaxPacketSize.

BIT3-0 Max Packet Size

Sets MaxPacketSize of endpoint.

After setting MaxPacketSize and DoubleBuf of all endpoints, be sure to set the ALLFIFOCLR bit to HIGH.

The list corresponding to these bits is as follows:

(bit3, bit2, bit1, bit0)

0 0 0 0 : Not in active use (no area is reserved)
0 0 0 1 : 8 bytes
0 0 1 0 : 16 bytes
0 1 0 0 : 32 bytes
1 0 0 0 : 64 bytes

7.5.2.10 EP {r}(r=a,b,c) Control (EPrControl) R/W

Sets operation for transaction in the direction of the endpoint set in the EPrConfig_0 register.

USBIndex : 01h to 03h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Ah	EPrControl	7: ForceNAK	Force NAK
		6: ForceSTALL	Force STALL
		5: EnShortPkt	Short Packet (IN Transaction Only)
		4: 0	Reserved
		3: ToggleMode	Toggle Mode (IN Transaction Only)
		2: ToggleStat	Toggle Status
		1: ToggleClr	Toggle Clear
		0: ToggleSet	Toggle Set

BIT7 Force NAK

Setting this bit to HIGH returns NAK to transaction.

When there is a transaction that is being executed, setting of this bit a fixed period of time after starting transaction becomes valid from the next transaction.

BIT6 Force STALL

Setting this bit to HIGH returns STALL to transaction.

Setting this bit to HIGH gives a STALL response to transaction.

When there is a transaction that is being executed, setting of this bit a fixed period of time after starting transaction becomes valid from the next transaction.

BIT5 Short Packet (IN Transaction Only)

Setting this bit to HIGH returns packet to IN Transaction independent of amount of data of FIFO.

When packet transfer is completed after setting this bit to HIGH this bit returns to LOW.

BIT3 Toggle Mode (IN Transaction Only)

Sets the Toggle Mode.

0: Normal Mode

1: Fake Mode

BIT2 Toggle Status

Shows the Toggle Sequence bit during transaction.

Updated when transfer is completed by returning Short Packet for IN Transaction or when ACK is received from the host for OUT Transaction.

BIT1 Toggle Clear

Sets the Toggle Sequence bit during transaction to 0.

BIT0 Toggle Set

Sets the Toggle Sequence bit during transaction to 1.

7.5.2.11 EP {r}(r=a,b,c) FIFO remain Counter (EPrFIFOremain) R

Shows the number of bytes of remaining data in FIFO of the endpoint.

USBIndex : 01h to 03h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Dh	EPrFIFOremain	7: EPrFIFOremainCount[7] 6: EPrFIFOremainCount[6] 5: EPrFIFOremainCount[5] 4: EPrFIFOremainCount[4] 3: EPrFIFOremainCount[3] 2: EPrFIFOremainCount[2] 1: EPrFIFOremainCount[1] 0: EPrFIFOremainCount[0]	EPr FIFO remain Counter

7.5.2.12 EPr FIFO for CPU (EPrFIFOforCPU) R/W

This register allows access to FIFO of the endpoint from the CPU.
When the EnFiFOwr bit of the EP0FIFOCtrl register is set, writing is enabled, when the EnFIFOrd bit is set, reading is enabled.
It should be noted that reading or writing causes changes in the number of data in FIFO, as it is similar to access from the USB.

USBIndex : 01h to 03h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Eh	EPrFIFOforCPU	7: EPrFIFOdata[7] 6: EPrFIFOdata[6] 5: EPrFIFOdata[5] 4: EPrFIFOdata[4] 3: EPrFIFOdata[3] 2: EPrFIFOdata[2] 1: EPrFIFOdata[1] 0: EPrFIFOdata[0]	EPr FIFO for CPU

7.5.2.13 EPr FIFO Control (EPrFIFOCtrl) R/W

Checks the status and sets operation of FIFO of the endpoint.

USBIndex : 01h to 03h			
Address	**Register Name**	**Bit Symbol**	**Description**
1Fh	EPrFIFOCtrl	7: FIFOEmpty	FIFO Empty
		6: FIFOFull	FIFO Full
		5: FIFOClr	FIFO Clear
		4: 0	Reserved
		3: 0	Reserved
		2: AutoForceNAK	AutoForceNAK
		1: EnFIFOwr	Enable FIFO Write
		0: EnFIFOrd	Enable FIFO Read

BIT7 FIFO Empty

When this bit is HIGH it indicates that FIFO is empty.
If a reading operation of FIFO is carried out in this state, invalid data is read out.

BIT6 FIFO Full

When this bit is HIGH it indicates that FIFO is full.
If a writing operation into FIFO is carried out in this state, the data is ignored.

BIT5 FIFO Clear

Setting this bit to HIGH clears data stored in FIFO.
The bit returns to LOW automatically after clearing.

BIT2 AutoForceNAK

Setting this bit to HIGH sets the InForceNAK bit of the EP0InControl register and the OutForceNAK bit of the EP0OutControl register when transaction is completed normally.

BIT1 Enable FIFO Write

Setting this bit to HIGH enables writing data into FIFO from the CPU.

BIT0 Enable FIFO Read

Setting this bit to HIGH enables reading data in FIFO from the CPU.

7.5.2.14 EP0 SETUP [n](n=0,1,2,3,4,5,6,7) (EP0SETUP[n]) R

Displays data received in the SETUP stage.

USBIndex : 08h			
Address	**Register Name**	**Bit Symbol**	**Description**
18h to 1Fh	Ep0SETUP[0]	7: Ep0_RcvSETUPdata[7]	
		6: Ep0_RcvSETUPdata[6]	
	Ep0SETUP[7]	5: Ep0_RcvSETUPdata[5]	
		4: Ep0_RcvSETUPdata[4]	Ep0_RcvSETUPdata
		3: Ep0_RcvSETUPdata[3]	
		2: Ep0_RcvSETUPdata[2]	
		1: Ep0_RcvSETUPdata[1]	
		0: Ep0_RcvSETUPdata[0]	

7.5.2.15 Frame Number H (FrameNumber_H) R

Displays higher order 3 bits in the FrameNumber field of the SOF Packet received.

USBIndex : 09h			
Address	**Register Name**	**Bit Symbol**	**Description**
18h	FrameNumber_H	7: 0	Reserved
		6: 0	Reserved
		5: 0	Reserved
		4: 0	Reserved
		3: 0	Reserved
		2: FrameNumber_[10]	Frame Number High
		1: FrameNumber_[9]	
		0: FrameNumber_[8]	

7.5.2.16 Frame Number L (FrameNumber_L) R

Displays lower order 8 bits in the FrameNumber field of the SOF Packet received.

USBIndex : 09h			
Address	**Register Name**	**Bit Symbol**	**Description**
19h	FrameNumber_L	7: FrameNumber_[7]	Frame Number Low
		6: FrameNumber_[6]	
		5: FrameNumber_[5]	
		4: FrameNumber_[4]	
		3: FrameNumber_[3]	
		2: FrameNumber_[2]	
		1: FrameNumber_[1]	
		0: FrameNumber_[0]	

7.6 SCSI Control Commands

7.6.1 Control Commands and Command Codes

Code	Command names	Summary of commands
00h		Reserved
01h	Abort_SCSI	SCSI Abort command
02h		Reserved
03h		
04h	Assert_RST	SCSI Bus Clear command
05h	Busfree	
06h		Reserved
07h	Assert_ATN	
08h	SEL_MSG_clear	SCAM control commands
09h	Select_WithoutATN	Connection system commands
0Ah	Select_WithATN_Commannd	
0Bh	SelectWithoutATN_Command	
0Ch	Wait_Selection_Command	
0Dh	Reselection	
0Eh	Wait_Reselection	
0Fh	Wait_SCAM_Selection_Command	
10h		Reserved
11h	Negate_ACK	Transfer system commands
12h	Command_Out	
13h	DMA_Data_Out	
14h	Non-DMA_Data_Out	
15h	DMA_Data_In	
16h	Non-DMA_Data_In	
17h	Status_In	
18h	Message_In	
19h	Message_Out	
1Ah	Status_Message	

7.6.2 Description of Each Control Command

●Abort_SCSI (01H)
Aborts the SCSI control command in process of execution. After aborting the process,
・The SABT bit of the MAININT register is set.
・The status block is set.
It causes an interruption. If this command is issued in the state of not operating, it is ignored.

●Assert_RST (04H)
Asserts the SCSI RST signal (XSRST) for 768 × T × 2 (about 46μs), and then negates it.
This command releases all the signals it asserts, causing the busfree condition. The inside of the IC is not initialized.
It sets the SCSIINT1 SRST bit after negating the RST signal, causing an interruption.
When the SCSI control command is being executed, it is forced to terminate. Only RST bit is set, though.
When the SCSI DMA command is being executed, the status block is set.

●Busfree (05H)
Executes busfree.
The command is valid only in Target mode. If issued in Initiator mode, it is ignored.
It is invalid if issued while other SCSI-type command is in execution, causing a command error and a command error interruption.

●Assert_ATN (07H)
Asserts the SCSI ATN signal (XSATN).
The command is valid only in Initiator mode. If issued in Target mode, it is ignored.
Also, it is not asserted in the busfree condition, however, no error occurs.
It causes no interruption after its execution.
Any other command being executed continues execution.
Negation of ATN occurs in any of the following cases:
・When the last byte is ACK-negated after the Message_Out command is issued.
・When busfree is detected.
・When the Assert_RST command is executed.
・When chip-reset is done.

●SEL_MSG_clear (08H)
Negates the SCSI SEL/MSG signal (XSSEL/XSMSG).
When SCAM is selectted by Wait_SCAM_Selection_Command, the SCAM protocol is processed in Direct mode, with SEL/MSG asserted left inside. This command is used to clear it. After issuing this command, release Direct mode.
It causes no interruption after its execution.
Any other command being executed continues execution.

●Select_WithoutATN (09H)
Executes selection without asserting the SCSI ATN signal.
This command is valid in both disconnected and connected conditions. Issuing this command while any other command in execution causes a command error.
After the command is issued, operation of the IC is as follows:
・Waits until the SCSI bus becomes busfree.
・Enters arbitration after detecting busfree.
・If it beats arbitration, it asserts XSSEL and ID bit to data bus, going into the selection phase.
・It terminates selection and operation when its counterpart asserts XSBSY.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
After that, the IC goes into Initiator mode.

●Select_WithATN_Command (0AH)
Asserts SCSI ATN signal, executes selection, and then executes the message-out command phase.
This command is valid in both disconnected and connected condition. Issuing this command while any other command is in execution causes a command error.

The CPU sets the message byte number in the NON-DMA data-size register before issuing this command.
Then, the CPU write message data in FIFO. After transferring the message, it sets the number of byte of command in the NON-DMA data-size register and writes the command data into FIFO.
The IC operates as follows:
・Waits for busfree.
・After detecting busfree, enters arbitration.
・When it beats arbitration, it asserts XSSEL and ID bit and then goes into the selection phase.
Asserts XSATN at this time.
・After selection, it checks message-out at the timing when XSREQ is asserted and transfers messages in FIFO.
・Negates XSATN after asserting XSREQ and before asserting XSACK at the last byte of the messages.
・After transferring all the messages, it checks the command phase, detects data accumulated in FIFO, and transfers the command data in FIFO according to the byte number newly set.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
After that, the IC goes into Initiator mode.

Note: Be sure to write data into FIFO after setting the number of bytes of transfer.

●Select_WithoutATN_Command (0BH)
Executes selection while SCSI ATN is being negated and continues to execute the command phase.
This command is valid in both disconnected and connected condition. Issuing this command while any other command is in execution causes a command error.

The CPU issues this command after setting the number of bytes of command in the NON_DMA data-size register.
The command data is written into FIFO.
The IC operates as follows:
・Waits for busfree.
・Enters arbitration after detecting busfree.
・When it beats arbitration, it asserts XSSEL and ID bit and then goes into the selection phase.
・After completing selection, it checks the command phase at the timing when XSREQ is asserted and transfers command data from FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
After that, the IC goes into Initiator mode.

●Wait_Select_Command (0CH)
Waits for the selection phase and executes the command phase after selection.
Valid only when it is not connected.
Issuing this command in the connected condition sets SCSIINT2 and CMDER bits and causes an interruption.
Any other command being executed continues execution.

When this command is issued, set STATN(bit5) of the SCSIMODE register and clear it when the command is terminated.
After issuing the command, the IC operates as follows:
(1) Waits for the selection phase.
(2) When selected, checks XSATN. If it is not asserted, the IC operates as mentioned in (5).
If it is asserted, the IC sets the message-out phase and receives a message.
(3) If the message received is other than "Identify", the IC operates as mentioned in (6) (The CPU checks the message in FIFO and responds to it).
(4) When XSATN remains to be asserted after 1-byte message ("Identify") is received, the IC terminates its operation by setting the SATN bit of the SCSIINT1 register and causes an interruption.
If XSATN is negated,
(5) The command phase is set to receive a command.
The IC distinguishes command groups and determines the number of bytes received automatically.
(6) It sets the GOOD bit of the MAININT register and causes an interruption.
After that, the IC goes into Target mode.

●Reselect (0DH)
Executes the re-selection phase.
Valid only when it is not connected.
Issuing this command in the connected condition sets the SCSIINT2 and CMDER bits and causes an interruption. Any other command being executed continues execution.

The IC operates as follows:
・Waits for busfree.
・Enters arbitration after detecting busfree.
・When it beats arbitration, it asserts the XSSEL, XSIO and ID bits and then goes into the re-selection phase.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
After that, the IC goes into Target mode.

●Wait_Reselect(0EH)
Waits for the re-selection phase.
Valid only when it is not connected.
Issuing this command in the connected condition sets the SCSIINT2 and CMDER bits and causes an interruption. Any other command being executed continues execution.

The IC operates as follows:
・Enters a state of waiting for re-selection.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
After that, the IC goes into Initiator mode.

●Wait_SCAM_Selection_Command (0FH)
Waits for SCAM selection and causes an interruption after the selection is made.
Valid only when it is not connected.
Issuing this command in the connected condition sets the SCSIINT2 and CMDER bits and causes an interruption. Any other command being executed continues execution.

When issuing this command, set STATN (bit5) of the SCSIMODE register and clear it when the command is terminated.
After issuing the command, the IC operates as follows:
・Waits for the SCAM/normal selection phase.
・Does not respond to, but ignores the SCSI selection with the selection time-out delay less than 4ms.
・Responds to the SCAM selection and causes an interruption.
・If no SCAM selection occurs, the IC responds to the selection which continues for 4ms or longer and after that it operates as in the case of Wait_Select_Command (0Ch).
After that, the IC goes into Target mode.

●Negate_ACK (11H)
Clears ACK left asserted by the last message transfer in Initiator mode when the LSI stops operation.

●Command_Out (12H)
Executes the SCSI command phase.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
The IC operates as follows:
Enters this command into FIFO after setting the number of bytes of command in the NON-DMA data-size register.
This control command does not distinguish command groups automatically. It is used for receiving a group command which has undefined command block length.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
The CPU reads a command from FIFO.

○In Initiator mode
The CPU issues this command after setting the number of bytes of the command in the NON-DMA data-size register.
Then it writes the command data into FIFO.
The IC operates as follows:
・At the start of execution, negates XSACK if it is asserted.
・Transfers the command data in FIFO after checking the command phase at the timing of assertion of XSREQ. When FIFO is empty, places the REQ-ACK handshake on hold until data is accumulate in FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.

If any other phase is found when the command phase is checked, the IC sets ILPHS of SCSIINT1 and causes an interruption.

Note: Be sure to set the number of bytes of transfer before writing data in FIFO.

●DMA_Data_Out (13H)
Executes the data-out phase that carries out transfer between port and SCSI.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
Combining this command issued and the AND condition of the DTGO bit of the DMACTL register causes DMA transfer to be started.
When transfer of the count value set in the DTBC register is completed, the command is terminated, the GOOD and DTCMP bits of the MAININT register are set, and an interruption is caused.

○In Initiator mode
At the start of execution, negates XSACK if it is asserted.
After the data-out phase is checked at the timing of assertion of XSREQ, the AND condition of the DTGO bit of the DMACTL register causes actual DMA transfer to start. When a transfer of the count value set in the DTBC register is completed, the command is terminated, the GOOD and DTCMP bits of the MAININT register are set, and an interruption is caused.

If any other phase is found when the data-out phase is checked, ILPHS of SCSIINT1 is set and an interruption is caused.

●Non-DMA_Data_Out (14H)
Valid only when the command is connected, which executes the data-out phase between SCSI and CPU interface. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
Sets the number of transfer in the NON-DMA data-size register and issues this command.
The CPU reads data from FIFO by checking the status of FIFO.
The IC operates as follows:
・Enters data equivalent to the number of bytes set into FIFO after setting the data-out phase.
When FIFO is full, the REQ-ACK handshake is held until when there is free space available in FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.

○In Initiator mode
Sets the number of transfer in the NON-DMA data-size register and issues this command.
The CPU writes data into FIFO by checking the status of FIFO.
The IC operates as follows:
・At the start of execution, negates XSACK if it is asserted.
Transfers data equivalent to the number of bytes set from FIFO after checking the data-out phase at the timing when XSREQ is asserted. When FIFO is empty, the REQ-ACK handshake is put on hold until when data is accumulated in FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.

Note: Be sure to set the number of bytes of transfer before writing data into FIFO.

●DMA_Data_In (15H)
Executes the data-in phase between SCSI and buffer.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
Combining this command issued and the AND condition of the DTGO bit of the DMACTL register causes DMA transfer to be started.
When transfer equivalent to the count value set in the DTBC register is completed, the command is terminated, the GOOD and DTCMP bits of the MAININT register are set, and an interruption is caused.

○In Initiator mode
At the start of execution, negates XSACK if it is asserted.
After the data-out phase is checked at the timing of assertion of XSREQ, the AND condition of the DTGO bit of the DMACTL register causes actual DMA transfer to start. When a transfer equivalent to the count value set in the DTBC register is completed, the command is terminated, the GOOD and DTCMP bits of the MAININT register are set, and an interruption is caused.

If any other phase is found when the data-in phase is checked, the IC sets ILPHS of SCSIINT1 and causes an interruption.

●Non-DMA_Data_In (16H)
Valid only when the command is connected, which executes the data-in phase between SCSI and CPU interface. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
Sets the number of transfer in the NON-DMA data-size register and issues this command.
The CPU writes data into FIFO by check the status of FIFO.
The IC operates as follows:
・Outputs data equivalent to the number of bytes set from FIFO after setting the data-in phase.
When FIFO is empty, the REQ-ACK handshake is put on hold until when data is accumulated in FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.

Note: Be sure to set the number of bytes of transfer before writing data into FIFO.

○In Initiator mode
Sets the number of transfer in the NON-DMA data-size register and issues this command.
The CPU reads data from FIFO by checking the status of FIFO.
The IC operates as follows:
・At the start of execution, negates XSACK if it is asserted.
・Enters data into FIFO after checking the data-in phase at the timing of assertion of XSREQ.
When FIFO is full, the REQ-ACK handshake is put on hold until when there is free space available in FIFO.

If any other phase is found when the data-in phase is checked, the IC sets ILPHS of SCSIINT1 and causes an interruption.

●Status_In (17H)
Executes the status phase.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
The CPU issues this command after writing the status byte into SCSI FIFO.
The IC transfers the status in FIFO after setting the status phase.
It sets the GOOD bit of MAININT register and causes an interruption.

○In Initiator mode
At the start of execution, negates XSACK if it is asserted.
Enters 1-byte status into SCSI FIFO after checking the status phase at the first timing when XSREQ is asserted.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
The CPU checks the interrupt status and reads the status byte from SCSI FIFO if terminated normally.

●Message_In (18H)
Executes the message-in phase.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
The CPU sets the number of bytes of the message to be sent in the NON-DMA data-size register and issues this command. It writes messages to be transferred into FIFO.
The IC sets the message phase and then sends data equivalent to the number of bytes set in FIFO.
When FIFO is empty, the REQ-ACK handshake is put on hold until when data is accumulated in FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.

Note: Be sure to set the number of bytes of transfer before writing data into FIFO.

○In Initiator mode
The CPU sets the number of bytes of the message to be received in the NON-DMA data-size register before issuing this command.
The IC operates as follows:
・At the start of execution, negates XSACK if it is asserted.
・Enters the message equivalent to the number of bytes into FIFO after checking the message-in phase at the timing when REQ is asserted. When FIFO is full, the REQ-ACK handshake is put on hold until when there is free space available in FIFO.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
Usually, the number of bytes of a message is unknown beforehand. So set the number of transfer to “1” when the command is issued first and then determine the number of bytes to be received from the second byte by checking the message code received.

●Message_Out (19H)
Executes the message-out phase.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
The CPU sets the number of bytes of the message to be received in the NON-DMA data-size register before issuing this command.
After setting the message-out phase, the IC enters the message equivalent to the number of bytes set into FIFO.
When FIFO becomes full, the REQ-ACK handshake put on hold until when the CPU reads out the message from FIFO to make some space available in it.
・After completion, it sets the GOOD bit of the MAININT register.
・It causes an interruption.
Usually, the number of bytes of a message is unknown beforehand. So set the number of transfer to “1” when

the command is issued first and then determine the number of bytes to be received from the second byte by checking the message code received.

○In Initiator mode
The CPU sets the number of bytes of a message to be sent in the NON-DMA data-size register before issuing this command.
The CPU writes the message to be transferred into FIFO.
The IC operates as follows:
Asserts XSATN.
・At the start of execution, negates XSACK if it is asserted.
・Sends data in FIFO after checking the message phase at the timing when XSREQ is asserted.
When FIFO is empty, the REQ-ACK handshake is put on hold until when data is accumulated in FIFO.
・Negates XSATN after sending the number of bytes to be transferred.
・After completion, it sets the GOOD bit of the MAINT register.
・It causes an interruption.

Note: Be sure to set the number of bytes of transfer before writing data into FIFO.

●Status_Message (1AH)
Executes the message-in phase after executing the status phase.
Valid only in the connected condition. It can be issued in either Target or Initiator mode.
Issuing this command in the disconnected condition sets the SCSIINT2 and CMDER bits and causes an interruption.

○In Target mode
Writes the status and message to be sent into FIFO and issues this command.
The status and message may be written after issuing the command.
The IC operates as follows:
・Sets the status phase, fetches 1-byte status byte from FIFO, and transfers it.
・Sets the message-in phase, fetches 1-byte message byte from FIFO and transfers it.
When FIFO is empty, the REQ-ACK handshake is put on hold until data is accumulated in FIFO.
・After completion, it sets the GOOD bit of the MAINT register.
・It causes an interruption.

○In Initiator mode
When this command is issued, 1 byte each of status and message is fetched into FIFO.
The CPU reads 1-byte status byte and then 1-byte message byte from FIFO.
The IC operates as follows:
・At the start of execution, negates XSACK if it is asserted.
・Enters the status into FIFO after checking the status phase at the timing of assertion of XSREQ.
・After receiving the status, it enters the message into FIFO after checking the message-in phase at the timing of assertion of XSREQ.
・After completion, it sets the GOOD bit of the MAINT register.
・It causes an interruption.

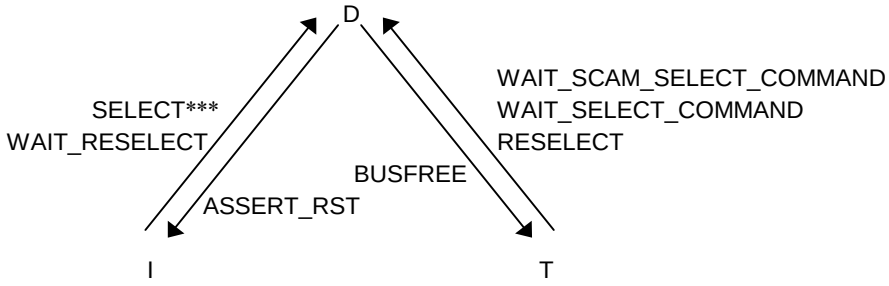
Note: The message length is fixed at one byte.

7.6.3 Command Execution and Change of State

The IC goes through the following three states from the viewpoint of execution of SCSI-type commands:

- Disconnected state (D)
- Connected state in Initiator mode (I)
- Connected state in Target mode (T)

Changes in these states caused by specific commands are as shown in the figure below:



Change from I to D takes place implicitly except that by the Assert_RST command.
It means that there is no command that causes such change explicitly. If busfree is found when the next connection-type command is issued, it is assumed to be the disconnected state and the command is executed. If busfree is not found, the disconnected state is waited for and the connection-related command is put on hold.

The connection-type command can be executed in the disconnected state. It may be issued in such state.
If it is issued while the SCSI control command is being executed, however, a command error occurs.
A transfer-type command can be executed in the connected state. If it is issued in the disconnected state, a command error occurs.
Both connection- and transfer-type commands can be executed when the IC is in the condition where no SCSI control command is being executed.
If they are issued while SCSI control command is in execution, a command error occurs.

7.7 Others and Cautions in Operation

●Operation responding to the selection without the SCSI-1 arbitration phase

The IC operates as mentioned below in response to the selection of only target ID of SCSI-1. Note that there occurs no (automatic) transition to the message or command phase after selection, as in the usual cases after the Wait_selection command.

(1) If only a target ID is selected after the Wait_select_cmd command is issued, an IDERR interrupt occurs and the command is terminated.
The inside is in the condition where connection is complete, though. So message_out/command_out and other commands can be issued, as in the usual case the selection is made with an initiator/target ID (except that message_out/command_out is not executed automatically).

(2) If an ID of 3 bits or more is selected, an IDERR interrupt occurs. This distinguishes whether selection of 1 bit is completed or IDERR with an ID of 3 bits or more is selected.
If 1 bit is selected, IDERR and SEL interrupts occur. If ATN is not asserted here, a WOATN interrupt occurs at the same time.

* The firm is asked to check that the SCSI-1 selection has occurred by observing the SEL interrupt at the same time when an IDERR interrupt occurs.
Also, issue message_out/command_out manually while observing the state of WOATN interrupt, because IDERR terminates the Wait_selection command.

●Parity error in the SCSI data phase or command stop operation by detecting ATN

The following points should be noted when the port interface is used as slave:
If a setting has been made that a parity error or detection of ATN in SCSI data phase stops the operation of a command (STATN/STPPE/SPCEN bit of SCSIMODE register), the occurrence of such factor and subsequent command stop causes negation of PDREQ being output to the port interface at the internal timing of the IC.
Accordingly, use such setting after checking that it causes no problem in handshake on the LSI side connected to the IC.
In such a case, no problem occurs in the internal sequence of the IC if XPDACK or XPRD/XPWR may come from the port side. Though, data transfer to and from FIFO may be obstructed depending on timing
(The data may not be written in or read from FIFO).
In such a case, FIFO terminates in uncompleted manner, so it requires clearing before going to the status phase.

8. ELECTRICAL CHARACTERISTICS

8.1 Absolute Maximum Ratings

V_{SS} = 0[V]

Item	Symbol	Ratings	Unit
Supply voltage	HV_{DD}	-0.3 to +6.0	V
	LV_{DD}	-0.3 to +4.6	V
Input voltage	HV_{IN}	-0.3 to HV_{DD} +0.5	V
	LV_{IN}	-0.3 to LV_{DD} +0.5	V
Output voltage	HV_{OUT}	-0.3 to HV_{DD} +0.5	V
	LV_{OUT}	-0.3 to LV_{DD} +0.5	V
Output current/pins (SCSI output pins)	I_{OUT1}	50	mA
Output current/pins (Other than SCSI output pins)	I_{OUT2}	±30	mA
Storage temperature	T_{stg}	-65 to +150	°C

8.2 Recommended Operating Conditions

V_{SS} = 0[V]

Item	Symbol	Min.	Typ.	Max.	Unit
Supply voltage	HV_{DD}	4.50	5.00	5.50	V
	LV_{DD}	3.00	3.30	3.60	
Input voltage	HV_{IN}	V_{SS}	—	HV_{DD}	V
	LV_{IN}	V_{SS}	—	LV_{DD}	V
Operating temperature	T_{opr}	0	25	70	°C
Input signal rise time Normal input	t_{ri}	—	—	50	ns
Input signal fall time Normal input	t_{fi}	—	—	50	ns
Input signal rise time Schmitt input	t_{ri}	—	—	5	ms
Input signal fall time Schmitt input	t_{fi}	—	—	5	ms

8.3 DC Characteristics

(1) I/O characteristics in the DC condition(Ta = 0 to 70°C, V_{SS}=0V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Static current	HI_{DDS}	HV_{DD}=5.0V±10%	—	—	80	μA
	LI_{DDS}	LV_{DD}=3.3V±0.3V	—	—	220	
Input leak current	I_{LI}	HV_{DD}=5.5V LV_{DD}=3.6V HV_{IH}=HV_{DD} LV_{IH}=LV_{DD} V_{IL}=V_{SS}	-1	—	1	μA
Input pins capacitance	C_I	f=1MHz HV_{DD}=0V	—	—	10	pF
Output pins capacitance	C_O	f=1MHz HV_{DD}=0V	—	—	10	pF
Input/output pins capacitance	C_{IO}	f=1MHz HV_{DD}=0V	—	—	10	pF

(2) TTL input characteristics (Ta = 0 to 70°C, V_{SS}=0V)
Names of signals covered: AD0 to 5, DB0 to 7, TESTEN, PD0 to 15, XSATN, XSBSY, XSRST, XSMSG, XSSEL, XSCD, XSIO

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level Input voltage	V_{IH}	HV_{DD}=5.5V	2.0	—	—	V
LOW level Input voltage	V_{IL2H}	HV_{DD}=4.5V	—	—	0.8	V

(3) CMOS input characteristics (Ta = 0 to 70°C, V_{SS} = 0V)
Names of signals covered: CLKSEL, OSCIN, EXCLK

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level Input voltage	V_{IHIL}	LV_{DD}=3.6V	1.9	—	—	V
LOW level Input voltage	V_{ILIL}	LV_{DD}=3.0V	—	—	0.8	V

(4) TTL Schmitt input characteristics (Ta = 0 to 70°C, V_{SS} = 0V)
Names of signals covered: XRESET, XSREQ, XSACK, XSDB0 to 7, XSDBP, XCS, XRD, XWR, XPRD, XPWR, XPDACK, PDREQ

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level trigger Input voltage	V_{T2+}	HV_{DD}=5.5V LV_{DD}=3.6V	1.2	—	2.4	V
LOW level trigger Input voltage	V_{T2-}	HV_{DD}=4.5V LV_{DD}=3.0V	0.6	—	1.8	V
Hysteresis voltage	ΔV_H	HV_{DD}=5.0V LV_{DD}=3.3V	0.1	—	—	V

(5) TTL Schmitt input characteristics (USB) (Ta = 0 to 70°C, V_{SS} = 0V)
Names of signals covered: DP, DM

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level trigger Input voltage	V_{T+} (USB)	LV_{DD}=3.6V	1.1	—	1.8	V
LOW level trigger Input voltage	V_{T-} (USB)	LV_{DD}=3.0V	1.0	—	1.5	V
Hysteresis voltage	ΔV_H (USB)	LV_{DD}=3.3V	0.1	—	—	V

(6) USB differential input characteristics (Ta = 0 to 70°C, V_{SS} = 0V)
Names of signals covered: A pair of DP and DM

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Differential input sensitivity	V_{DS}	LV_{DD}=3.0V Differential input voltage 0.8 to 2.5V	—	—	0.2	V

(7) Characteristics of pull-up and pull-down input (Ta = 0 to 70°C, V_{SS}=0V)
Names of signals covered: XRESET, TESTEN

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Pull-up resistor	R_{pu}	V_I=0V HV_{DD}=5.0V	50	100	200	kΩ
Pull-down resistor	R_{pd}	V_I= HV_{DD} HV_{DD}=5.0V	50	100	200	kΩ

(8) Output characteristics (Ta = 0 to 70°C, V_{SS} = 0V) (I_{OL} = 2mA)
Names of signals covered: TESTMON, XUSBOE

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level Output voltage	V_{OH}	LV_{DD}=3.0V I_{OH}=-2mA	LV_{DD} -0.4	—	—	V
LOW level Output voltage	V_{OL}	LV_{DD}=MIN I_{OL}=-2mA	—	—	0.4	V

(9) Output characteristics (Ta = 0 to 70°C,V_{SS} = 0V) (I_{OL} = 3mA)
Names of signals covered: XPDACK, PD0 to 15, DB0 to 7, XPRD, XPWR

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level Output voltage	V_{OH}	HV_{DD}=5.0V I_{OH}=-1.5mA	HV_{DD} -0.4	—	—	V
LOW level Output voltage	V_{OL}	HV_{DD}=4.5V I_{OL}=3mA	—	—	0.4	V

(10) Output characteristics (Ta = 0 to 70°C, V_{SS} = 0V) (I_{OL} = 6mA)
Names of signals covered: XINTU, XINTS, PDREQ

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level Output voltage	V_{OH}	HV_{DD}=5.0V I_{OH}=-3mA	HV_{DD} -0.4	—	—	μA
LOW level Output voltage	V_{OL}	HV_{DD}=4.5V I_{OL}=6mA	—	—	0.4	V

(11) Output characteristics (Ta = 0 to 70°C, V_{SS} = 0V) (Open drain I_{OL} = 6mA)
Names of signals covered: XPUENB

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
OFF-state Leak current	I_{OZ}	HV_{DD}=Max.	-1	—	1	μA
LOW level Output voltage	V_{OL5}	HV_{DD}=Min. I_{OL}=6mA	—	—	0.4	V

(12) Output characteristics (Ta = 0 to 70°C,V_{SS} = 0V) (Open drain I_{OL} = 48mA)
Names of signals covered: XSATN, XSBSY, XSRST, XSMSG, XSSEL, XSCD, XSIO

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
OFF-state Leak current	I_{OZ}	HV_{DD}=Max.	-1	—	1	μA
LOW level Output voltage	V_{OL5}	HV_{DD}=Min. I_{OL}=48mA	—	—	0.4	V

(13) Output characteristics (Ta = 0 to 70°C, V_{SS} = 0V) (I_{OL} = 48mA)
Names of signals covered: XSDB0 to 7, XSDBP, XSREQ, XSACK

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit	
HIGH level Output voltage	V_{OH}	HV_{DD}=Min. I_{OH}=-20mA	1.5	—	—	V	*1
LOW level Output voltage	V_{OL}	HV_{DD}=Min. I_{OL}=48mA	—	—	0.4	V	

*1: V_{OH} of the active negation cell meets Active negation Current vs. Voltage defined in the American National Standard X3T10/1071D for Information Systems - SCSI-3 Fast20.

(14) Output characteristics (Ta = 0 to 70°C, V_{SS} = 0V)
Names of signals covered: DP, DM

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
HIGH level Output voltage	V_{OH}	LV_{DD}=Min. I_{OH}=-0.5mA	2.8	—	—	V
LOW level Output voltage	V_{OL}	LV_{DD}=Min. I_{OL}=3.0mA	—	—	0.3	V

8.4 AC Characteristics

Measurement conditions of AC characteristics:

・Ta = 0 to 70°C HV_{DD} = 5V±10% LV_{DD} = 3.3V±0.3V
V_{SS} = 0V

・DC level to determine input
0.8V to 2.4V

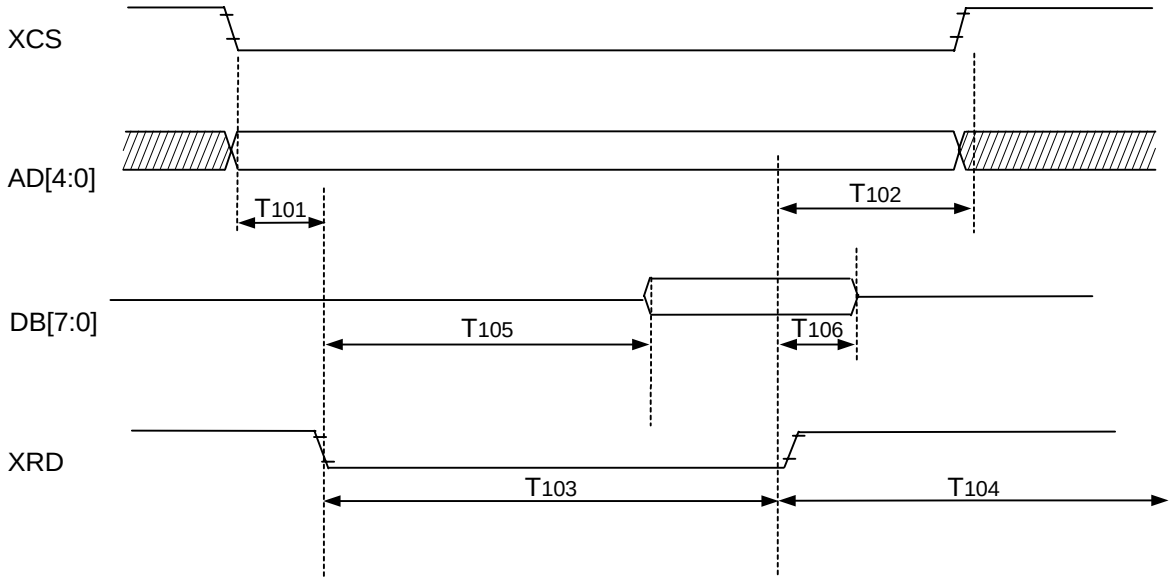
・Operating clock
f_{oscin}=20MHz (internal 40/48MHz operation):PLL operation

・Loading conditions of output pins except SCSI pins
Drives load capacitance of 50pF and 1TTL.

・Load capacitance of SCSI pins
Load capacitance = 100pF, pull-up resistance = 110Ω/pull-down resistance = 165Ω

8.4.1 CPU Interface

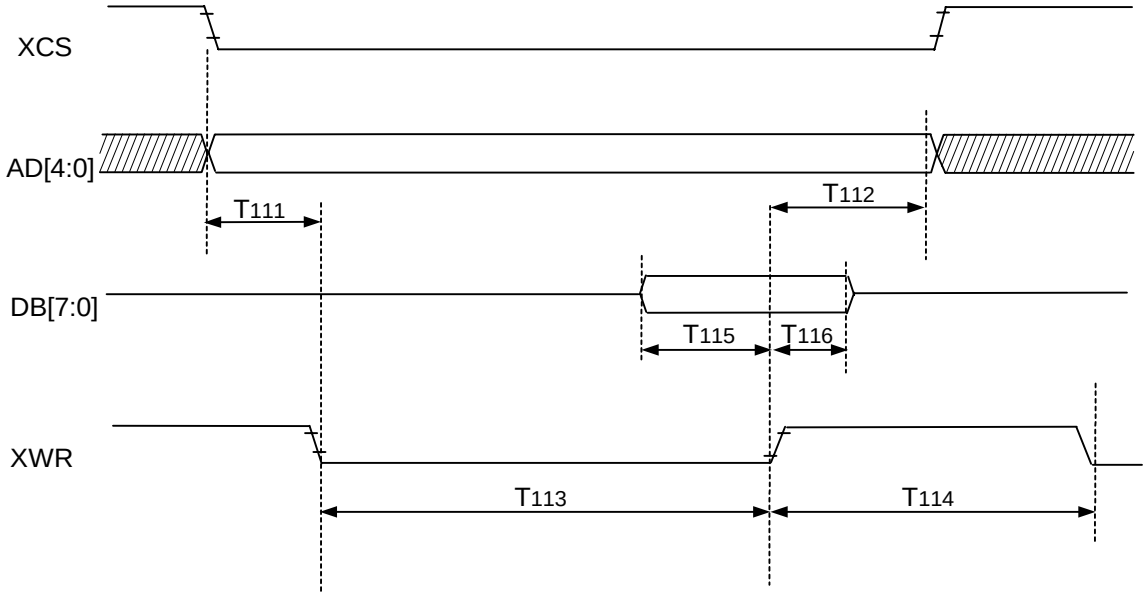
8.4.1.1 Register Read Timing



Symbol	Item		Min.	Typ.	Max.	Unit
T_{101}	XCS fall → XRD fall AD [4:0] Valid→ XRD fall		0	—	—	ns
T_{102}	XRD rise → AD [4:0] Invalid XRD rise → XCS rise		0	—	—	ns
T_{103}	XRD LOW level pulse width		65	—	—	ns
T_{104}	XRD HIGH level pulse width		45	—	—	ns
T_{105}	XRD fall→ DB[7:0] output	Normal (Registers other than those below)	2	—	65	ns
		FIFO area for USB *1			120	
T_{106}	XRD rise → DB[7:0] hold		2	—	15	ns

*1: The FIFO area for USB indicates acess to USBWindow_6(1Eh) in case of USBIndex(17h) = 00h to 03h.

8.4.1.2 Register Write Timing



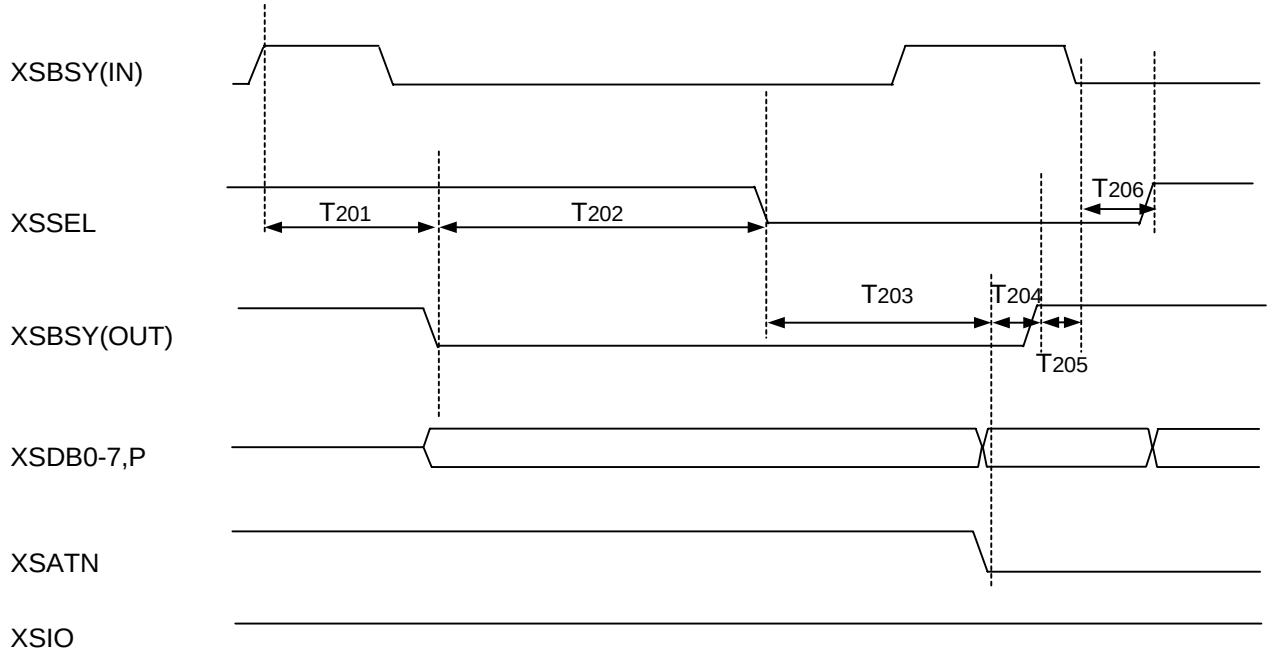
Symbol	Item		Min.	Typ.	Max.	Unit
T_{111}	XCS fall → XWR fall AD [4:0] Valid → XWR fall		0	—	—	ns
T_{112}	XWR rise → AD [4:0] Invalid XWR rise → XCS rise		0	—	—	ns
T_{113}	XWR LOW level pulse width		40	—	—	ns
T_{114}	XWR HIGH level pulse width	Normal (Registers other than those below)	45	—	—	ns
		FIFO area_A for USB *1	80			
		FIFO area_B for USB *2	120			
T_{115}	DB[7:0] valid → XWR rise		10	—	—	ns
T_{116}	XWR rise → DB[7:0] hold		0	—	—	ns

*1: When written into FIFO area for USB (see the previous page) and the next access also writes into FIFO area for USB.

*2: When written into FIFO area for USB (see the previous page) and the next access reads FIFO area for USB or reads the FIFO full/Empty bit for USB.

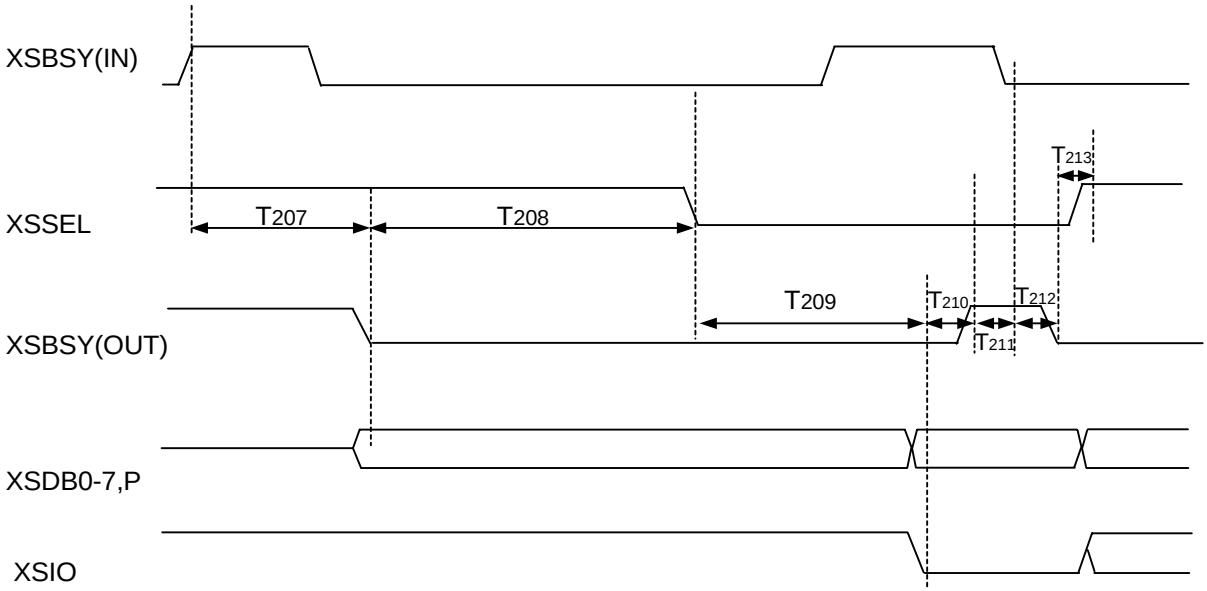
8.4.2 SCSI Interface

8.4.2.1 Selection Timing



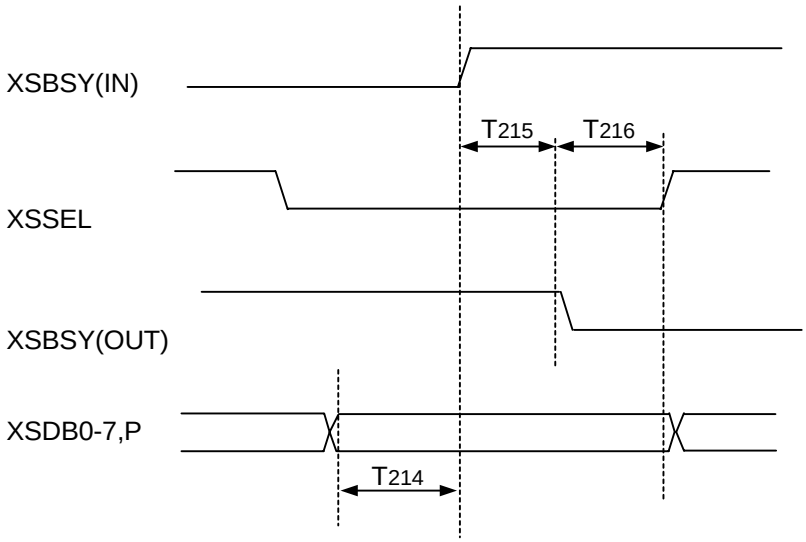
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{201}	XSBSY(IN) ↑ - XSBSY(OUT) ↓, OWNID valid	1600	—	—	ns
T_{202}	XSBSY(OUT) ↓ - XSSEL ↓	3000	—	—	ns
T_{203}	XSSEL ↓ - SELID valid	1500	—	—	ns
T_{204}	SELID valid - XSBSY(OUT) ↑	150	—	—	ns
T_{205}	XSBSY(OUT) ↑ - XSBSY(IN) ↓	500	—	—	ns
T_{206}	XSBSY(IN) ↓ - XSSEL ↑	250	—	—	ns

8.4.2.2 Re-selection Timing



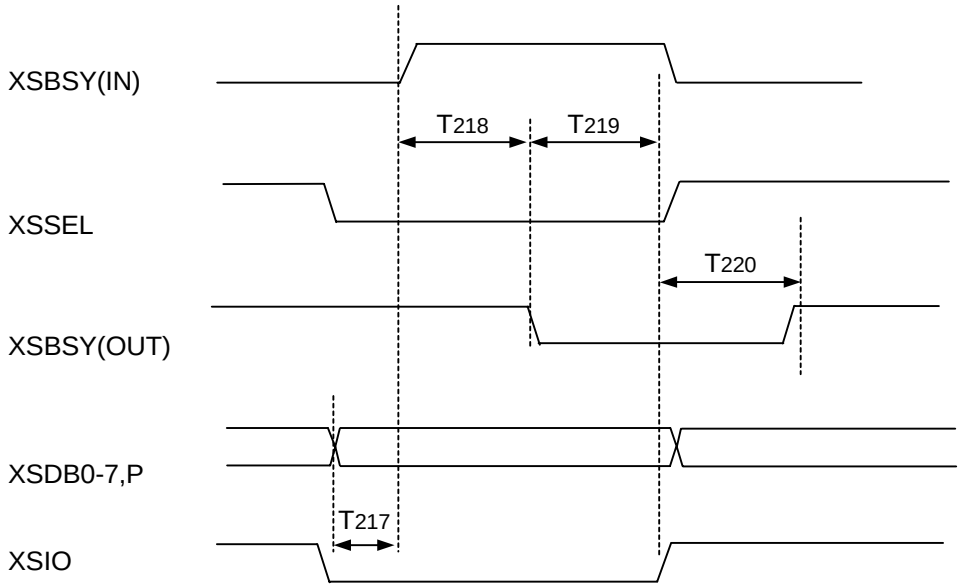
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{207}	XSBSY(IN) ↑ - XSBSY(OUT) ↓, OWNID valid	1600	—	—	ns
T_{208}	XSBSY(OUT) ↓ - XSSEL ↓	3000	—	—	ns
T_{209}	XSSEL↓ -SELID valid, XSIO ↓	1500	—	—	ns
T_{210}	SELID valid - XSBSY(OUT) ↑	150	—	—	ns
T_{211}	XSBSY(OUT) ↑ - XSBSY(IN) ↓	500	—	—	ns
T_{212}	XSBSY(IN) ↓ - XSBSY(OUT) ↓	100	—	—	ns
T_{213}	XSBSY(OUT) ↓ - XSSEL ↑	150	—	—	ns

8.4.2.3 Timing of Being Selected



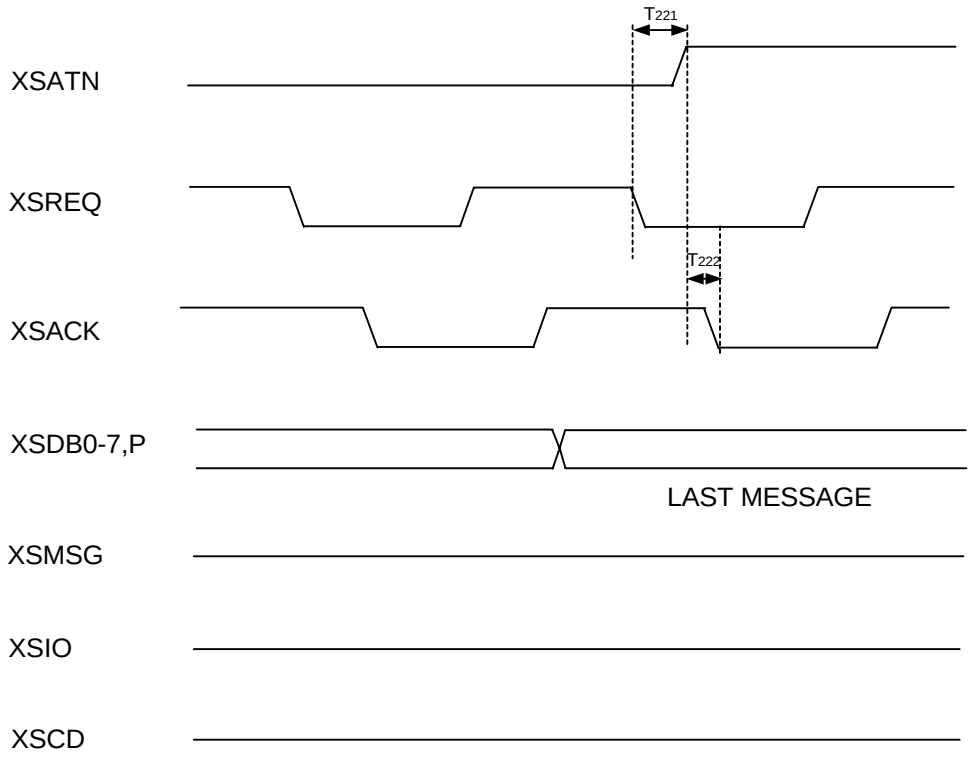
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{214}	SELID valid - XSBSY(IN) ↑	0	—	—	ns
T_{215}	XSBSY(IN) ↑ - XSBSY(OUT) ↓	800	—	—	ns
T_{216}	XSBSY(OUT) ↓ - XSSEL ↑	0	—	—	ns

8.4.2.4 Timing of Being Selected



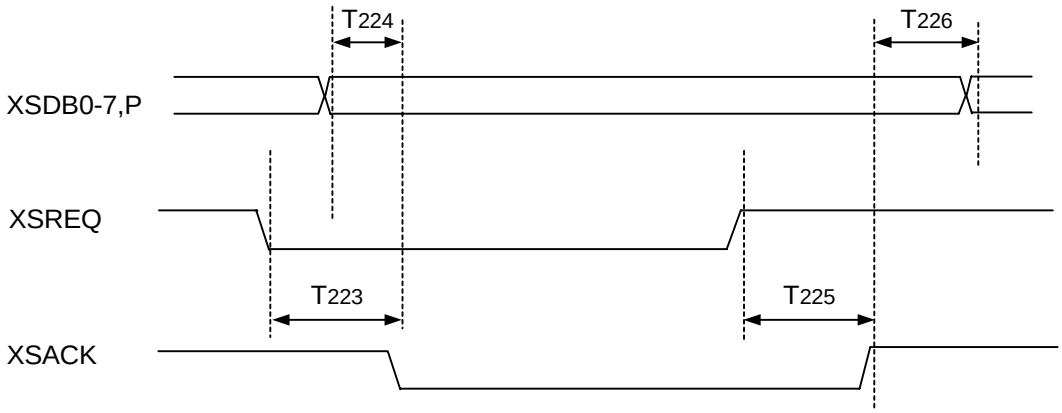
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{217}	SELID valid - XSBSY(IN) ↑	0	—	—	ns
T_{218}	XSBSY(IN) ↑ - XSBSY(OUT) ↓	800	—	—	ns
T_{219}	XSBSY(OUT) ↑ - XSSEL ↑	0	—	—	ns
T_{220}	XSSEL ↓ - XSBSY(OUT) ↑	—	—	200	ns

8.4.2.5 XSATN Output Timing



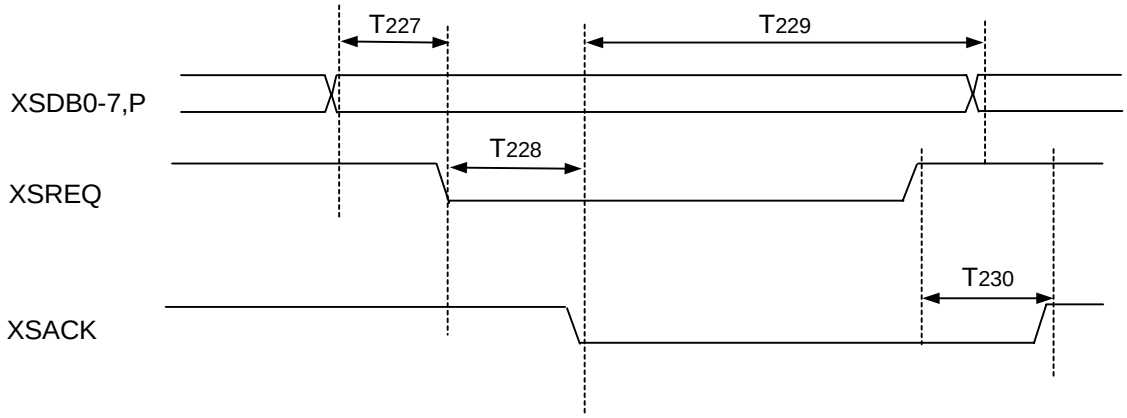
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{221}	XSREQ ↓ - XSATN ↑	25	—	—	ns
T_{222}	XSATN ↑ - XSACK ↓	150	—	—	ns

8.4.2.6 Initiator Asynchronous Data-out Timing (Data output)



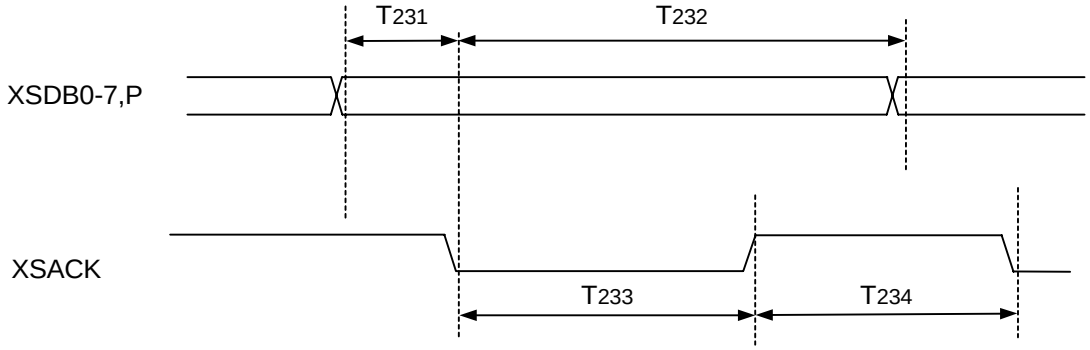
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{223}	XSREQ ↓ - XSACK ↓	25	—	—	ns
T_{224}	XSDB valid - XSACK ↓	100	—	—	ns
T_{225}	XSREQ ↑ - XSACK ↑	25	—	90	ns
T_{226}	XSACK ↑ - XSDB invalid	50	—	—	ns

8.4.2.7 Initiator Asynchronous Data-in Timing (Data input)



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{227}	XSDB valid - XSREQ ↓	30	—	—	ns
T_{228}	XSREQ ↓ - XSACK ↓	25	—	—	ns
T_{229}	XSACK ↓ - XSDB invalid	0	—	—	ns
T_{230}	XSREQ ↑ - XSACK ↑	25	—	90	ns

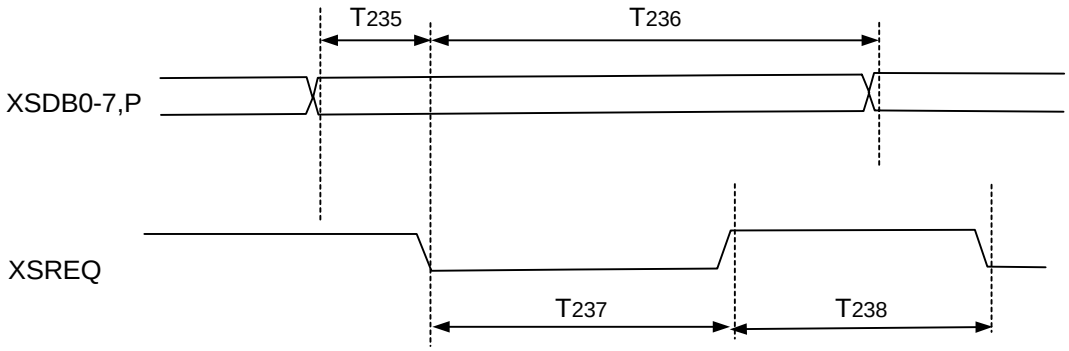
8.4.2.8 Initiator Synchronous Data-out Timing (Data output)



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{231}	XSDB valid - XSACK ↓	25	—	—	ns
T_{232}	XSACK ↓ - XSDB invalid	25	—	—	ns
T_{233}	XSACK ↓ - XSACK ↑	25	—	—	ns
T_{234}	XSACK ↑ - (NEXT) XSACK ↓	25	—	—	ns

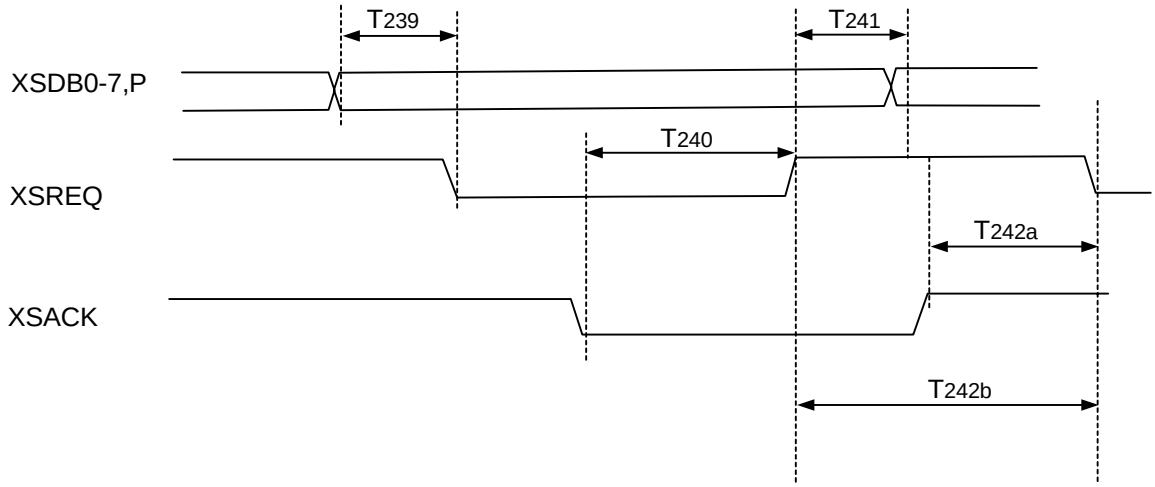
Note: Value when RATE3 to 0bit is "0000."
The timing of switching data is the same as in the case of XSREQ rise.

8.4.2.9 Initiator Synchronous Data-in Timing (Data input)



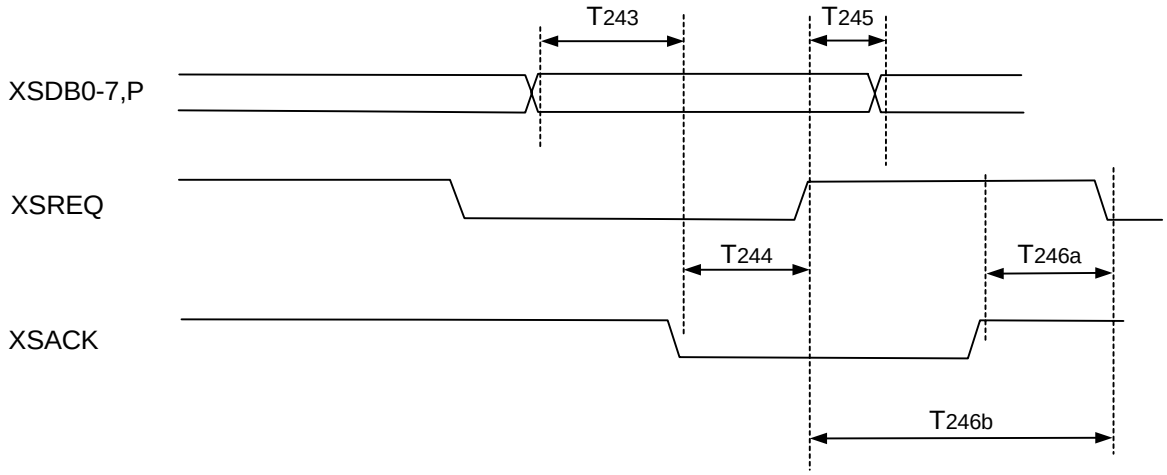
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{235}	XSDB0-7, P valid - XSREQ ↓	6.5	—	—	ns
T_{236}	XSREQ ↓ - XSDB0-7, P invalid	5	—	—	ns
T_{237}	XSREQ ↓ - XSREQ ↑	11	—	—	ns
T_{238}	XSREQ ↑ - XSREQ ↓	11	—	—	ns

8.4.2.10 Target Asynchronous Data-in Timing (Data output)



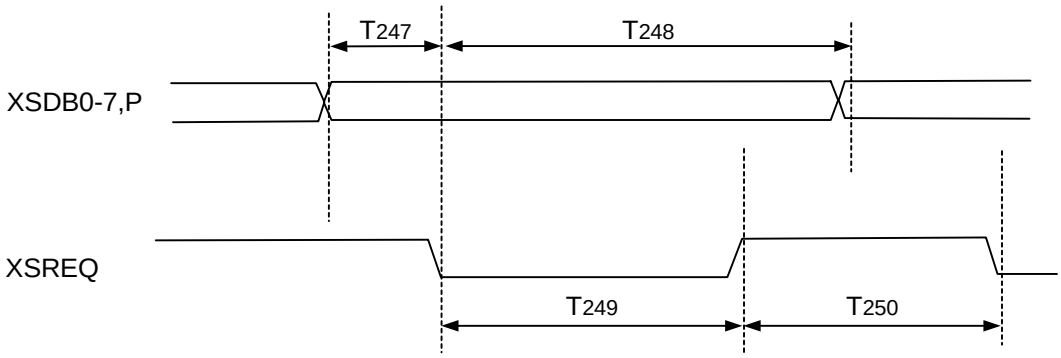
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{239}	XSDB valid - XSREQ ↓	100	—	—	ns
T_{240}	XSACK ↑ - XSREQ ↑	25	—	90	ns
T_{241}	XSREQ ↑ - XSDB invalid	50	—	—	ns
T_{242a}	XSACK ↑ - (NEXT) XSREQ ↓	25	—	—	ns
T_{242b}	XSREQ ↑ - XSREQ ↓	150	—	—	ns

8.4.2.11 Target Asynchronous Data-out Timing (Data input)



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{243}	XSDB valid - XSACK ↓	30	—	—	ns
T_{244}	XSACK ↓ - XSREQ ↑	25	—	90	ns
T_{245}	XSREQ ↑ - XSDB invalid	0	—	—	ns
T_{246a}	XSACK ↑ - XSREQ ↓	25	—	—	ns
T_{246b}	XSREQ ↑ - XSREQ ↓	150	—	—	ns

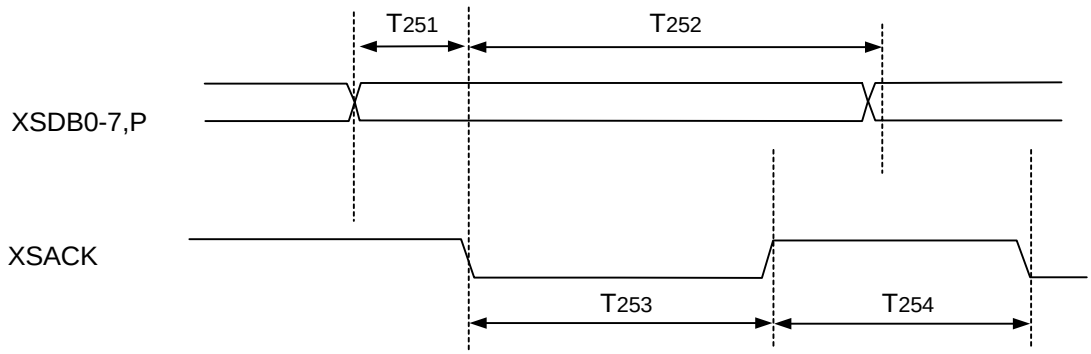
8.4.2.12 Target Synchronous Data-in Timing (Data output)



Symbol	Specification	Min.	Typ.	Max.	Unit
T247	XSDB valid - XSREQ ↓	25	—	—	ns
T248	XSREQ ↓ - XSDB invalid	25	—	—	ns
T249	XSREQ ↓ - XSREQ↑	25	—	—	ns
T250	XSREQ ↑ - XSREQ ↓	25	—	—	ns

Note: Value when RATE3 to 0bit is "0000."
The timing of switching data is the same as in the case of XSREQ rise.

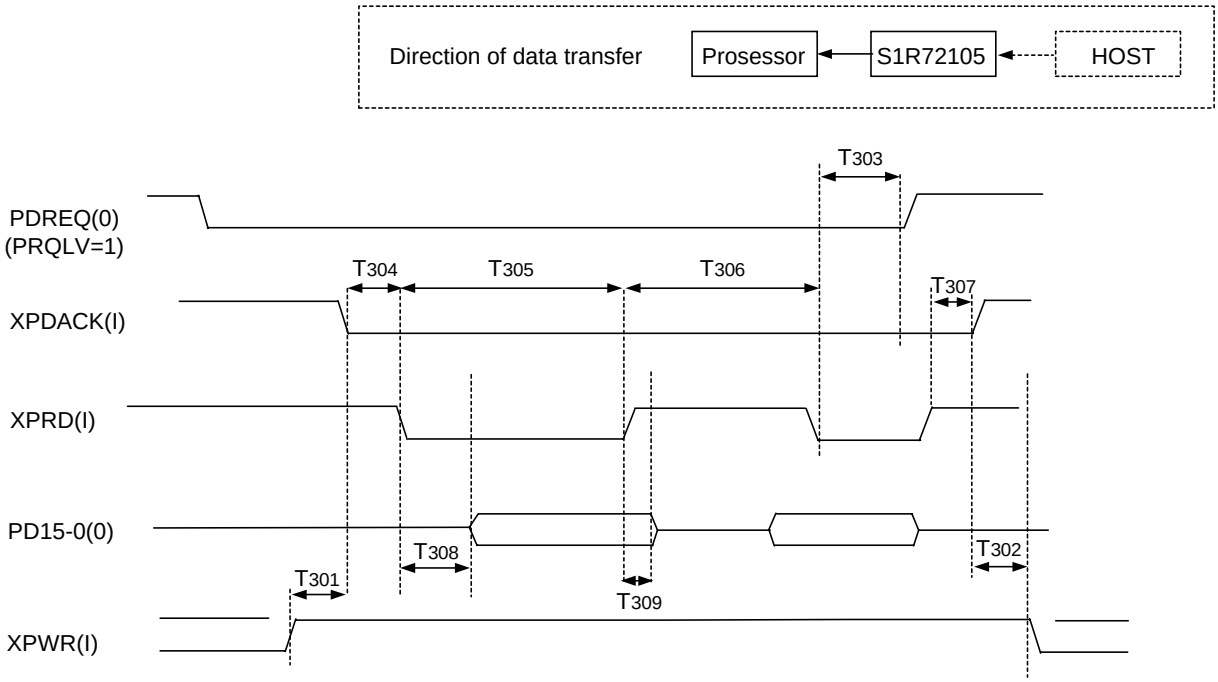
8.4.2.13 Target Synchronous Data-out Timing (Data input)



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{251}	XSDB valid - XSACK ↓	6.5	—	—	ns
T_{252}	XSACK ↓ - XSDB invalid	5	—	—	ns
T_{253}	XSACK ↓ - XSACK ↑	11	—	—	ns
T_{254}	XSACK ↑ - XSACK ↓	11	—	—	ns

8.4.3 Port Interface

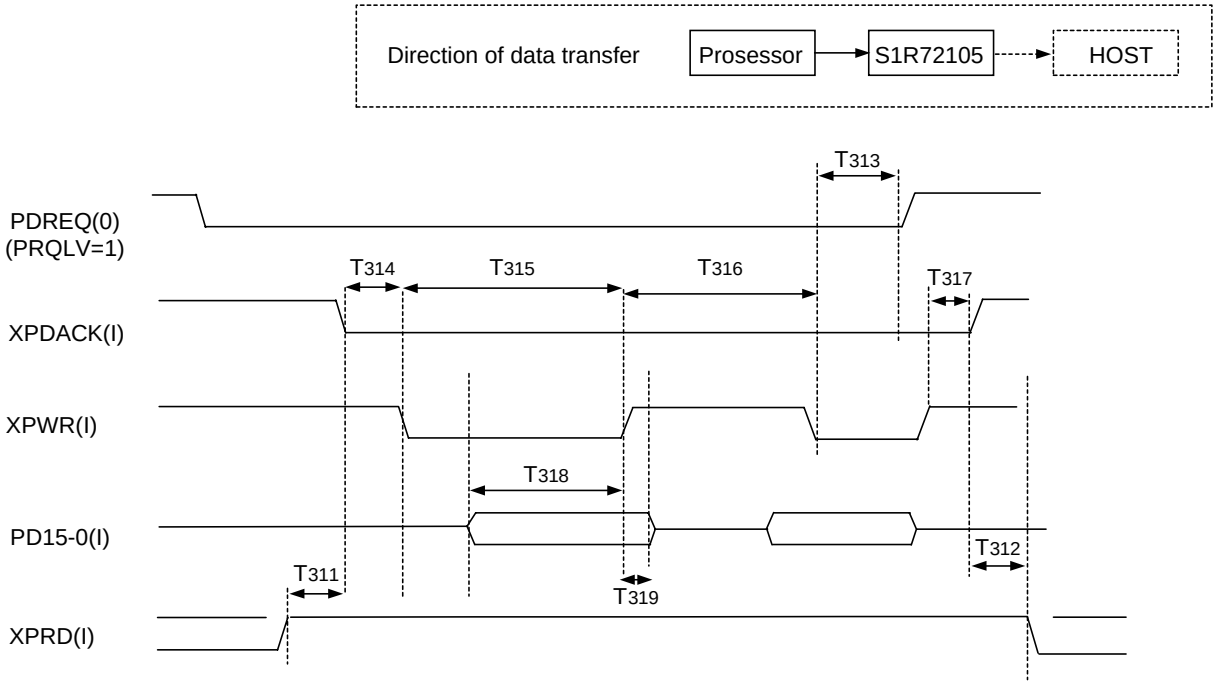
8.4.3.1 DMA Read (PSLV=1: Slave mode)



Symbol	Specification	Min.	Typ.	Max.	Unit
T301	XPWR → XPDACK ↓ XPDACK setup time	5	—	—	ns
T302	XPDACK ↑ → XPWR XPDACK hold time	5	—	—	ns
T303	XPRD ↓ → PDREQ negate PDREQ negate delay time	10	—	37	ns
T304	XPDACK ↓ → XPRD ↓ XPRD setup time	0	—	—	ns
T305	XPRD ↓ → XPRD ↑ XPRD assert pulse width	30	—	—	ns
T306	XPRD ↑ → XPRD ↓ XPRD negate pulse width	30	—	—	ns
T307	XPRD ↑ → XPDACK ↑ XPRD hold time	0	—	—	ns
T308	XPRD ↓ → PD Data output delay time Note 1	0	—	25	ns
T309	XPRD ↑ → PD(Hi-Z) Data bus negate time Note 1	6	—	40	ns

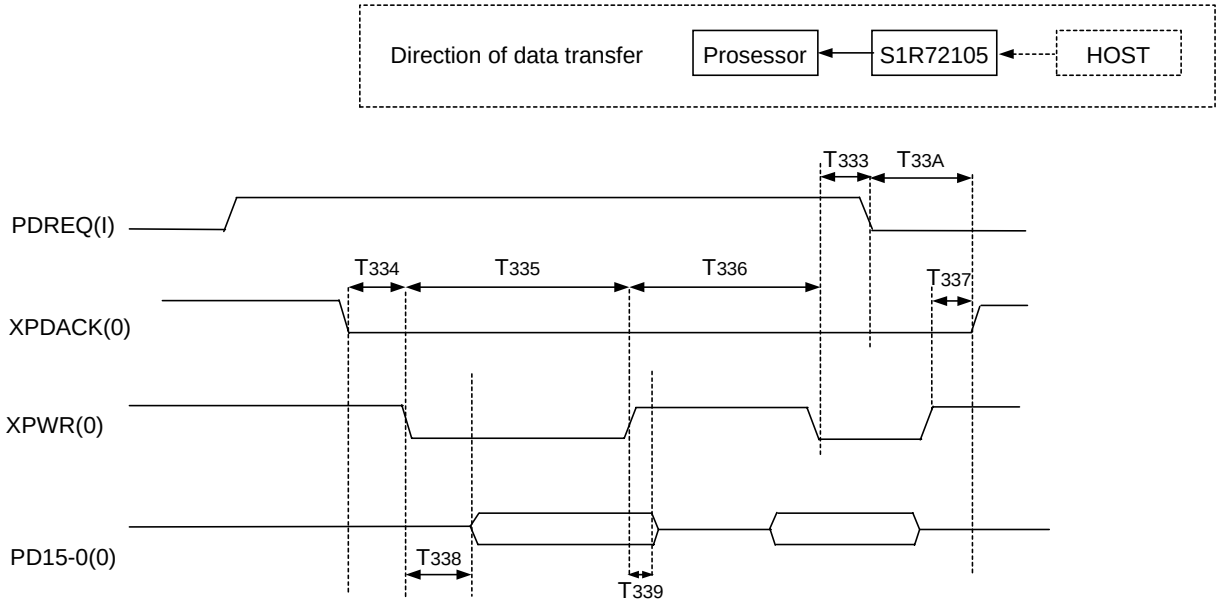
Note 1: Data is output to PD only while both XPDACK and XPRD are asserted.
PD is always in Input mode except such time.

8.4.3.2 DMA Write (PSLV=1: Slave mode)



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{311}	XPRD → XPDACK ↓ XPDACK setup time	5	—	—	ns
T_{312}	XPDACK ↑ → XPRD XPDACK hold time	5	—	—	ns
T_{313}	XPWR ↓ → PDREQ negate PDREQ negate delay time	10	—	37	ns
T_{314}	XPDACK ↓ → XPWR ↓ XPWR setup time	0	—	—	ns
T_{315}	XPWR ↓ → XPWR ↑ XPWR assert pulse width	30	—	—	ns
T_{316}	XPWR ↑ → XPWR ↓ XPWR negate pulse width	30	—	—	ns
T_{317}	XPWR ↑ → XPDACK ↑ XPWR hold time	0	—	—	ns
T_{318}	PD → XPWR ↑ Data input delay time	10	—	—	ns
T_{319}	XPWR ↑ → PD Data hold time	0	—	—	ns

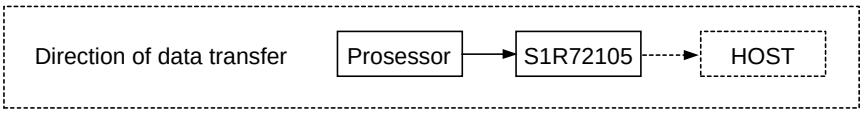
8.4.3.3 DMA Write (PSLV=0: Master mode)

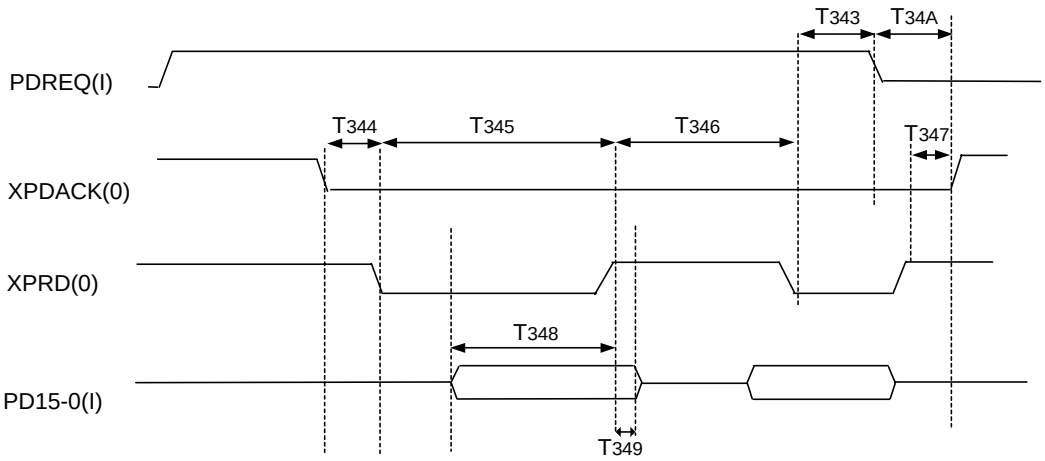


Symbol	Specification	Min.	Typ.	Max.	Unit
T_{333}	XPWR ↓ → PDREQ negate PDREQ negate delay time	0	—	30	ns
T_{334}	XPDACK ↓ → XPWR ↓ XPWR setup time	0	—	5	ns
T_{335}	XPWR ↓ → XPWR ↑ XPWR assert pulse width	—	(AP+2)×25	—	ns
T_{336}	XPWR ↑ → XPWR ↓ XPWR negate pulse width	—	(NP+2)×25	—	ns
T_{337}	XPWR ↑ → XPDACK ↑ XPWR hold time	0	—	5	ns
T_{338}	XPWR ↓ → PD Data output delay time Note 1	0	—	25	ns
T_{339}	XPWR ↑ → PD(Hi-Z) Data bus negate time Note 1	5	—	40	ns
T_{33A}	PDREQ negate → XPDACK ↑ XPDACK setup time	5	—	—	ns

Note 1: Data is output to PD only while both XPDACK and XPWR are asserted.
PD is always in Input mode except such time.

8.4.3.4 DMA Read (PSLV=0: Master mode)



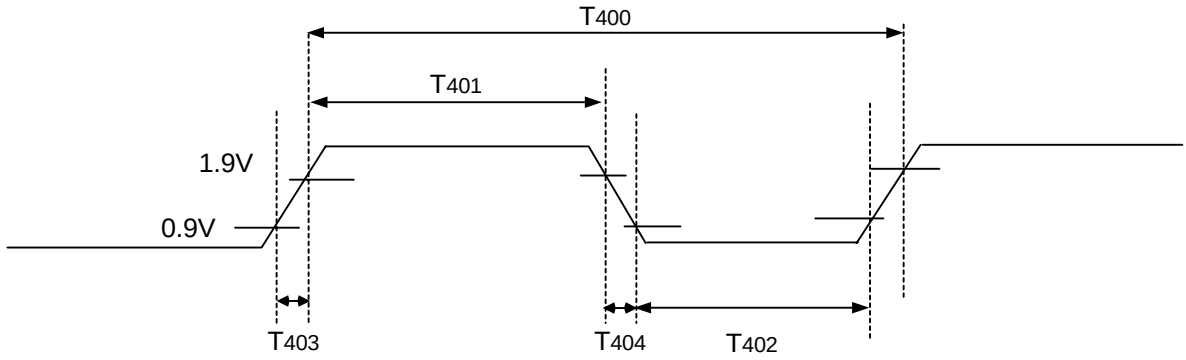


Symbol	Specification	Min.	Typ.	Max.	Unit
T_{343}	XPRD ↓ → PDREQ negate PDREQ negate delay time	0	—	30	ns
T_{344}	XPDACK ↓ → XPRD ↓ XPRD setup time	0	—	5	ns
T_{345}	XPRD ↓ → XPRD ↑ XPRD assert pulse width	—	(AP+2)×25	—	ns
T_{346}	XPRD ↑ → XPRD ↓ XPRD negate pulse width	—	(NP+2)×25	—	ns
T_{347}	XPRD ↑ → XPDACK ↑ XPRD hold time	0	—	5	ns
T_{348}	PD → XPRD ↑ Data input delay time	10	—	—	ns
T_{349}	XPRD ↑ → PD Data hold time	0	—	—	ns
T_{34A}	PDREQ negate → XPDACK ↑ XPDACK setup time	5	—	—	ns

8.4.4 Others

8.4.4.1 OSCIN Input Clock (ex.40MHz)

Note: Maximum input voltage: LV_{DD}



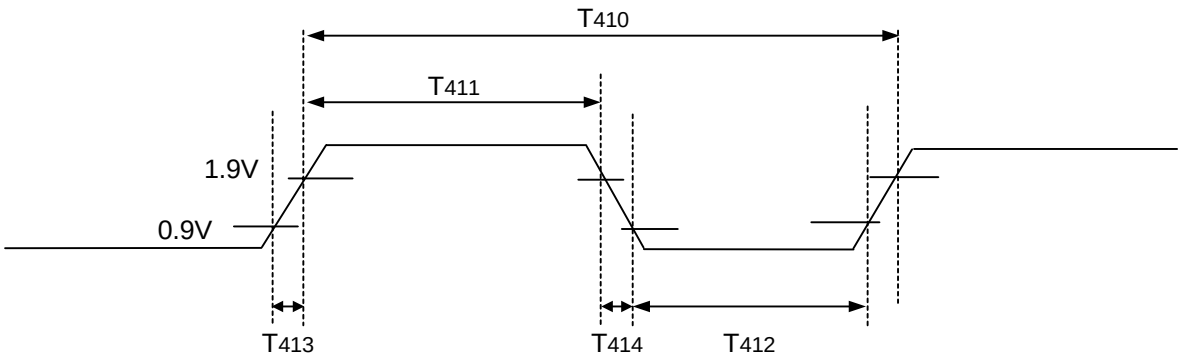
Symbol	Specification	Min.	Typ.	Max.	Unit
T_{400}	CLK cycle *1	0	25 (1/f)	—	ns
T_{401}	CLK HIGH width *1	10 (1/f)×0.4	—	15 (1/f)×0.6	ns
T_{402}	CLK LOW width *1	10 (1/f)×0.4	—	15 (1/f)×0.6	ns
T_{403}	CLK rise time	—	—	5	ns
T_{404}	CLK fall time	—	—	5	ns

*1 $T_{401+402}=T_{400}$

*1 Specified in the same rate also in any cases other than CLK input = 20MHz.

8.4.4.2 EXCLK Input Clock (48MHz)

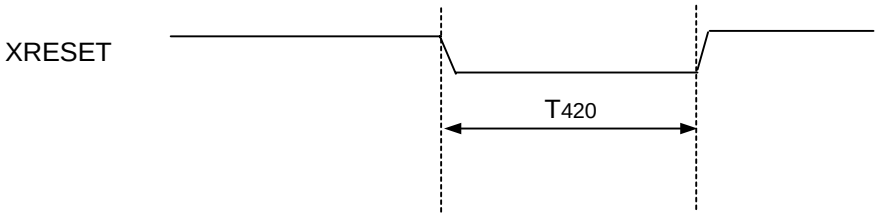
Note: Maximum input voltage: LV_{DD}



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{410}	CLK cycle *1	—	20.83	—	ns
T_{411}	CLK HIGH width *1	8.33	—	12.5	ns
T_{412}	CLK LOW width *1	8.33	—	12.5	ns
T_{413}	CLK rise time	—	—	5	ns
T_{414}	CLK fall time	—	—	5	ns

*1 $T_{411+412}=T_{410}$

8.4.4.3 XRESET Input Pulse Width



Symbol	Specification	Min.	Typ.	Max.	Unit
T_{420}	XRESET LOW width	150	—	—	ns

8.4.4.4 USB Interface Access Timing

Conformity to the USB 1.1 Specification.
For the USB1.1 Specification, visit http://www.usb.org/developers/docs.html.

9. EXAMPLES OF CONNECTION

(When 20MHz OSC oscillation is used)

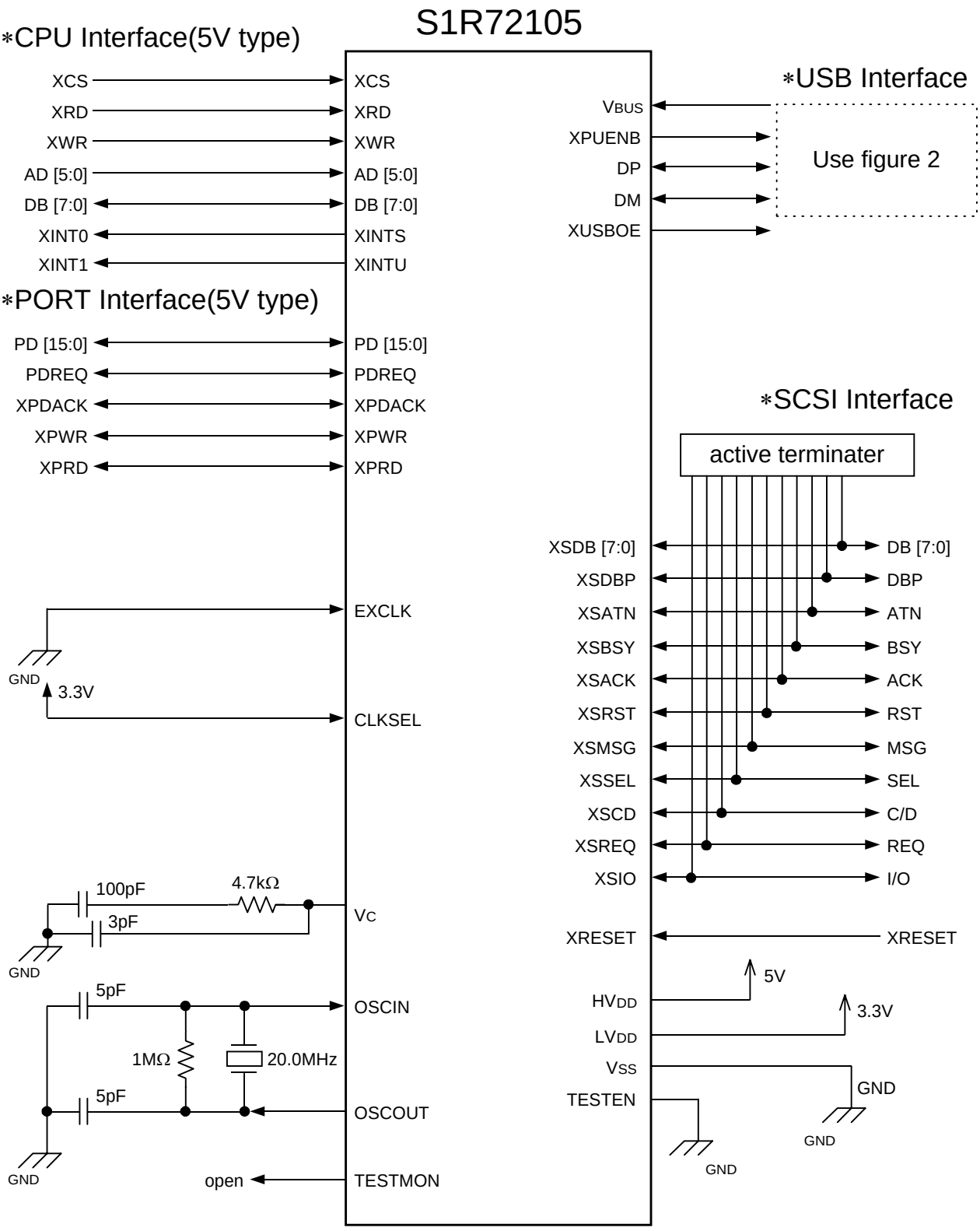


Fig.1. General View

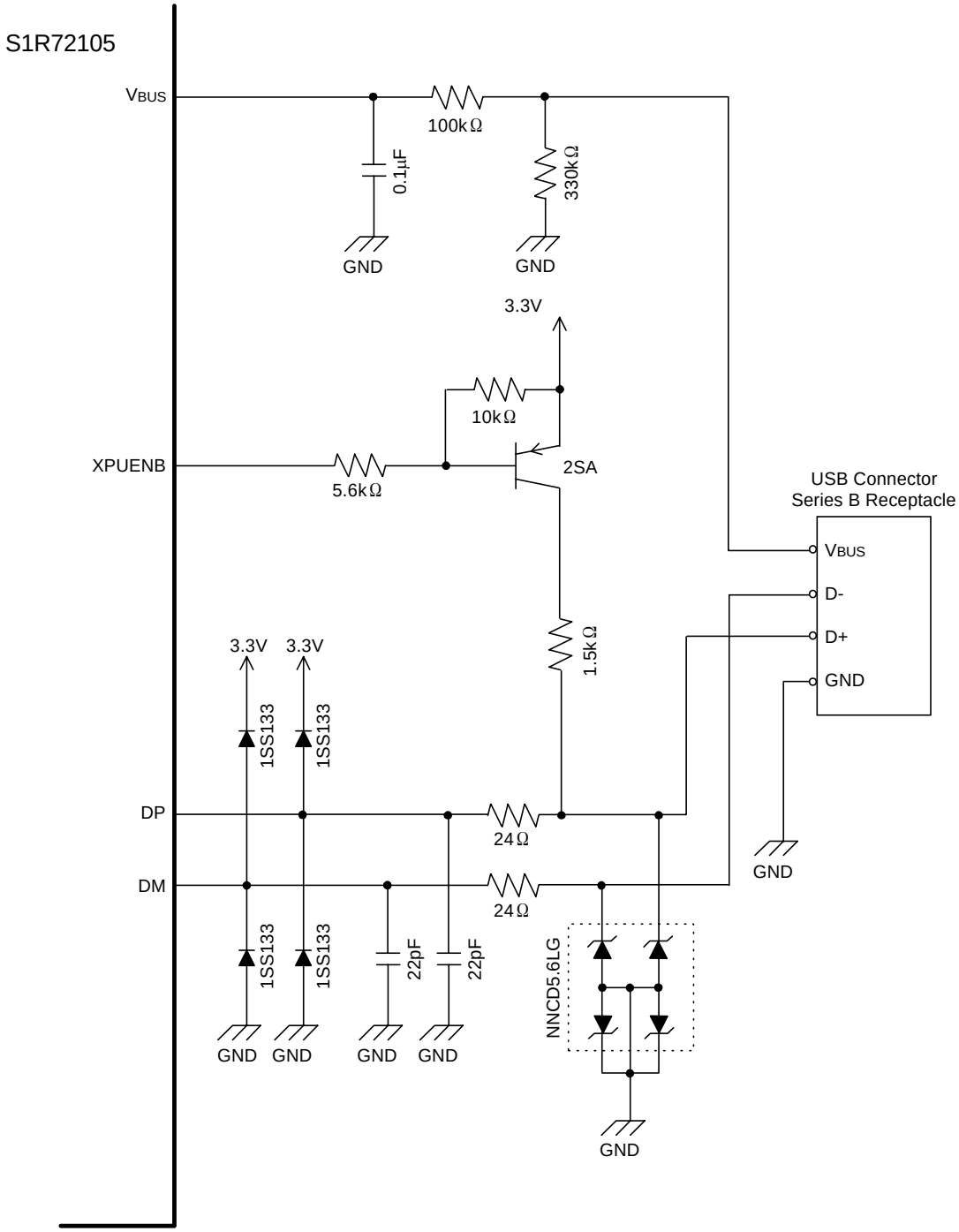
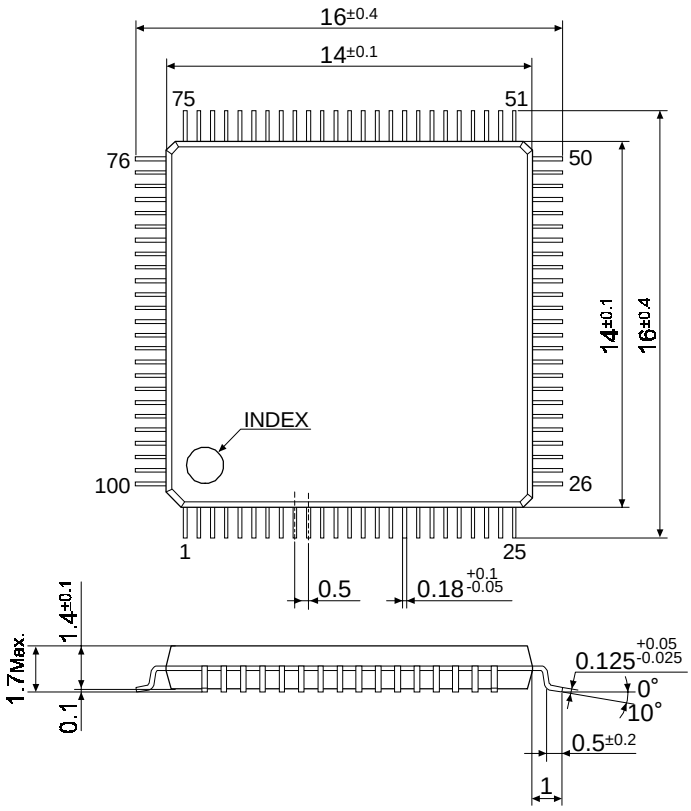


Fig.2. USB Connection Diagram

1) Connect the VBUS pin to the VBUS pin of the connector by providing chattering prevention circuit.
2) The XPUENB outputs LOW by inputting 5V in VBUS pin and setting the EnPullUp(bit1) bit of the USBCommon register (0Ah).
   XPUENB is asserted when USB is ready for operation after checking that VBUS is HIGH (5V).
3) How to handle DP and DM pins.
   - Add Zener diode for electrostatic protection onto the connector side.
   - Add 24 Ω resistor for adjusting impedance.
   - Add 22pF condensor for adjusting edge rate.
   - Add a diode onto the pin side to protect from a short in the wiring of the USB cable.

10. EXTERNAL DIMENSIONS DRAWING

Plastic QFP15-100 pin



Unit:mm

EPSON

International Sales Operations

AMERICA

EPSON ELECTRONICS AMERICA, INC.
HEADQUARTERS
150 River Oaks Parkway
San Jose, CA 95134, U.S.A.
Phone: +1-408-922-0200 FAX: +1-408-922-0238

SALES OFFICES
West
1960 E.Grand Avenue
El Segundo, CA 90245, U.S.A.
Phone: +1-310-955-5300 FAX: +1-310-955-5400

Central
101 Virginia Street, Suite 290
Crystal Lake, IL 60014, U.S.A.
Phone: +1-815-455-7630 FAX: +1-815-455-7633

Northeast
301 Edgewater Place, Suite 120
Wakefield, MA 01880, U.S.A.
Phone: +1-781-246-3600 FAX: +1-781-246-5443

Southeast
3010 Royal Blvd. South, Suite 170
Alpharetta, GA 30005, U.S.A.
Phone: +1-877-EEA-0020 FAX: +1-770-777-2637

EUROPE

EPSON EUROPE ELECTRONICS GmbH
HEADQUARTERS
Riesstrasse 15
80992 Munich, GERMANY
Phone: +49-(0)89-14005-0 FAX: +49-(0)89-14005-110

DÜSSELDORF BRANCH OFFICE
Altstadtstrasse 176
51379 Leverkusen, GERMANY
Phone: +49-(0)2171-5045-0 FAX: +49-(0)2171-5045-10

UK & IRELAND BRANCH OFFICE
Unit 2.4, Doncastle House, Doncastle Road
Bracknell, Berkshire RG12 8PE, ENGLAND
Phone: +44-(0)1344-381700 FAX: +44-(0)1344-381701

FRENCH BRANCH OFFICE
1 Avenue de l' Atlantique, LP 915 Les Conquerants
Z.A. de Courtaboeuf 2, F-91976 Les Ulis Cedex, FRANCE
Phone: +33-(0)1-64862350 FAX: +33-(0)1-64862355

BARCELONA BRANCH OFFICE
Barcelona Design Center
Edificio Testa, Avda. Alcalde Barrils num. 64-68
E-08190 Sant Cugat del Vallès, SPAIN
Phone: +34-93-544-2490 FAX: +34-93-544-2491

Scotland Design Center
Integration House, The Alba Campus
Livingston West Lothian, EH54 7EG, SCOTLAND
Phone: +44-1506-605040 FAX: +44-1506-605041

ASIA

EPSON (CHINA) CO., LTD.
23F, Beijing Silver Tower 2# North RD DongSanHuan
ChaoYang District, Beijing, CHINA
Phone: 64106655 FAX: 64107319

SHANGHAI BRANCH
7F, High-Tech Bldg., 900, Yishan Road,
Shanghai, 200233, CHINA
Phone: 86-21-5423-5577 FAX: 86-21-5423-4677

EPSON HONG KONG LTD.
20/F., Harbour Centre, 25 Harbour Road
Wanchai, Hong Kong
Phone: +852-2585-4600 FAX: +852-2827-4346
Telex: 65542 EPSCO HX

EPSON TAIWAN TECHNOLOGY & TRADING LTD.
10F, No. 287, Nanking East Road, Sec. 3
Taipei
Phone: 02-2717-7360 FAX: 02-2712-9164
Telex: 24444 EPSONTB

HSINCHU OFFICE
13F-3, No. 295, Kuang-Fu Road, Sec. 2
HsinChu 300
Phone: 03-573-9900 FAX: 03-573-9169

EPSON SINGAPORE PTE., LTD.
No. 1 Temasek Avenue, #36-00
Millenia Tower, SINGAPORE 039192
Phone: +65-6337-7911 FAX: +65-6334-2716

SEIKO EPSON CORPORATION
KOREA OFFICE
50F, KLI 63 Bldg., 60 Yoido-dong
Youngdeungpo-Ku, Seoul, 150-763, KOREA
Phone: 02-784-6027 FAX: 02-767-3677

GUMI OFFICE
6F, Good Morning Securities Bldg., 56 Songjeong-Dong,
Gumi-City, Seoul, 730-090, KOREA
Phone: 054-454-6027 FAX: 054-454-6093

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

IC Marketing Department
IC Marketing & Engineering Group
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5816 FAX: +81-(0)42-587-5624

ED International Marketing Department
421-8, Hino, Hino-shi, Tokyo 191-8501, JAPAN
Phone: +81-(0)42-587-5814 FAX: +81-(0)42-587-5117

In pursuit of **“Saving” Technology**, Epson electronic devices.
Our lineup of semiconductors, displays and quartz devices
assists in creating the products of our customers’ dreams.
Epson IS energy savings.

S1R72105
Technical Manual

SEIKO EPSON CORPORATION
ELECTRONIC DEVICES MARKETING DIVISION

■ EPSON Electronic Devices Website

http://www.epsondevice.com/

First issue October, 2002
Printed in Japan Ⓗ A