1 Overview

1.1 Overview

The M66592 is a USB 2.0 peripheral controller that is compliant with USB (Universal Serial Bus) specification Rev. 2.0 for both Hi-Speed and Full-Speed transfers. This controller has a built-in USB transceiver and supports all of the transfer types defined by the USB specification. The compact package and low power consumption make it ideal for use in mobile devices.

The M66592 has a 5 kB built-in buffer memory for data transfers and enables use of up to eight pipes. For pipes 1 to 7, any end point numbers can be assigned, based on the user's system. The M66592 can be connected to the CPU using either a separate bus or a multiplex bus. Moreover, a split bus interface (dedicated DMA interface) is provided independent of the CPU bus interface, making this an ideal choice for systems that require transfer of large volumes of data at high speed.

1.2 Features

1.2.1 USB Rev. 2.0 Hi-Speed supported

- Compliant with USB specification Rev. 2.0
- Both Hi-Speed transfer(480 Mbps)and Full-Speed transfer (12 Mbps) are supported
- Built-in Hi-Speed / Full-Speed USB transceiver
- Can be operated as a Hi-Speed / Full-Speed peripheral controller

1.2.2 Reduced power consumption

- 1.5 V core power supply
- Low power consumption makes this ideal for mobile devices
- Low-power mode (power-saving sleep state) supported to reduce power consumption during suspended operation

1.2.3 Space-saving installation supported

- Few external elements are used, so less space is required for mounting
  - VBUS signal can be connected directly to the controller pin
  - Built-in D+ pull-up resistor
  - Built-in D+ and D- terminating resistors (for Hi-Speed operation)
  - Built-in D+ and D- output resistors (for Full-Speed operation)
- Compact 64-pin package used

1.2.4 Isochronous transfer supported

- All types of USB transfers supported
  - Control transfers
  - Bulk transfers
  - Interrupt transfers (High-Bandwidth transfers are not supported)
  - Isochronous transfers (High-Bandwidth transfers are not supported)

1.2.5 Bus interfaces

- The user can select either a 1.8 V or 3.3 V bus interface power supply
- 16-bit CPU bus interface
  - 16-bit separate bus and 16-bit multiplex bus supported
  - 8-bit and 16-bit DMA interface (slave function) supported
- 8-bit split bus (dedicated external DMA interface) supported
- DMA interface has two channels built into it.
- DMA transfer enables high-speed access of 40 MB/sec.

1.2.6 Pipe configuration

- Internal 5 KB buffer memory for USB communication built in
- Up to 8 pipes(endpoints) can be selected (including the default control pipe for endpoint 0)
- Programmable pipe configuration
- End point numbers can be assigned flexibly to PIPE1to PIPE7.
- Transfer conditions that can be set for theeach pipe
  - Pipe 0: Control transfer, continuous transfer mode, 256-byte fixed single buffer
  - PIPE1 and PIPE2: Bulk transfer / isochronous transfer, continuous transfer mode, programmable buffer size (up to 2 KB; double buffer can be specified)
  - PIPE 3 to PIPE5: Bulk transfer, continuous transfer mode, programmable buffer size (up to 2 KB; double buffer can be specified)
  - PIPE6 and PIPE7: Interrupt transfer, 64-byte fixed single buffer

1.2.7 Other functions

- Automatic recognition of Hi-Speed operation or Full-Speed operation based on automatic response to the reset handshake
- Byte endian swap function when using 16-bit data transfers
- Transaction count function when using DMA transfers
- DMA transfer termination function using external trigger (DEND pin)
- Control transfer stage control function
- Device state control function
- Auto response function for SET_ADDRESS request
- SOF interpolation function
- SOF pulse output function
- Three types of input clocks can built into the PLL and are available for selection
  - Input clocks of 48 MHz / 24 MHz / 12 MHz can be selected
- Zero-Length packet addition function (**DEZPM**) when ending DMA transfers using the DEND pin
- BRDY interrupt event notification timing change function (**BFRE**)
- Function that automatically clears the buffer memory after the data for the pipe specified at the DxFIFO port has been read (**DCLRM**)
- Function to automatically supply a clock from the low-power sleep state (**ATCKM**)
- NAK setting function for response PID generated by end of transfer (**SHTNAK**)
- NAK response assignment function (**NRDY**)

1.2.8 Applications

Digital video cameras, digital still cameras, printers, external storage devices, portable information terminals, USB audio devices
Also: GeneralOrdinary PC peripheral devices equipped with Hi-Speed USB

1.3 Pin layout diagram

Figure 1.1 shows the pin layout diagram (top view) of the controller.

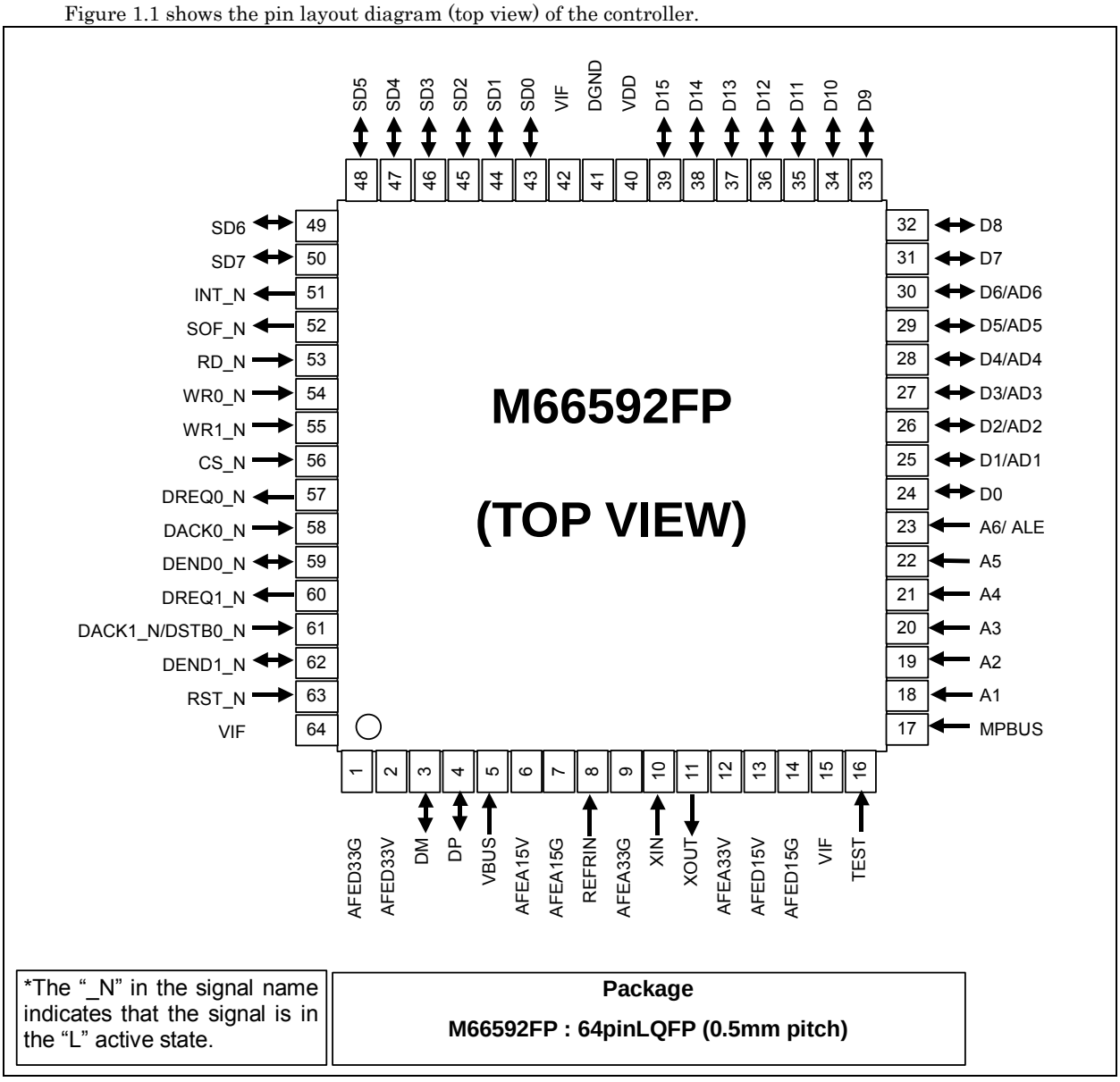


Figure 1.1 Pin layout diagram of M66592FP

M66592WG(TOP VIEW)

	A	B	C	D	E	F	G	H
8	SD6	SD4	SD2	DGND	VDD	D13	D10	D9
7	SD7	SD5	SD3	VIF	D15	D12	D8	D7
6	RD_N	SOF_N	INT_N	SD0	D14	D11	D6/AD6	D5/AD5
5	CS_N	WR1_N	WR0_N	SD1	D2/AD2	D1/AD1	D4/AD4	D3/AD3
4	DEND0_N	DREQ1_N	DREQ0_N	DACK0_N	A5	A2	A6/ALE	D0
3	DACK1_N /DSTB0_N	VIF	DEND1_N	AFEA15V	AFEA33G	AFEA33V	A3	A4
2	RST_N	AFED33V	VBUS	AFEA15G	XOUT	AFED15G	TEST	A1
1	AFED33G	DM	DP	REFRIN	XIN	AFED15V	VIF	MPBUS

*The "_N" in the signal name indicates that the signal is in the "L" active state.

Package

M66592WG : 64pin FBGA (0.8mm pitch)

Figure 1.2 Pin layout diagram of M66592WG

1.4 Description of pins

Table 1.1 describes the controller pins.

Table 1.1 Pin descriptions

Category	Pin name	Name	I/O	Function	Pin count (Pin nos.)	State of pin *7)		
						RST_N="L"	RST_N goes "H"	PCUT=1
CPU bus interface	D15-0	Data Bus	I/O	This is a 16-bit data bus.	16 (24-39)	*4)	*4)	Input (Hi-z)
	AD6-1	Multiplex Address Bus	I/O	When a multiplex bus is specified, this group of pins is used on a time-shared basis for some of the data buses (D6-D1), or for 6 bits of the address bus (A6-A1).				
	A6-1	Address Bus	IN	This is a 6-bit address bus. Because the data bus consists of 16 bits, there is no A0.	6 (18-23)	Input *5)	Input *5)	Input (Hi-z) Input
	ALE	Address Latch Enable	IN	When a multiplex bus is specified, the A6 pin is used as the ALE signal.		Input	Input	
	CS_N	Chip Select	IN	Setting this to the "L" level selects this controller.	1 (56)	Input *6)	Input *6)	Input
	RD_N	Read Strobe	IN	Setting this to the "L" level reads data from the controller registers.	1 (53)	Input	Input	Input
	WR0_N	D7-0 Byte Write Strobe	IN	At the rising edge, D7-D0 are written to the registers of the controller.	1 (54)	Input *6)	Input *6)	Input
	WR1_N	D15-8 Byte Write Strobe	IN	At the rising edge, D15-D8 are written to the registers of the controller.	1 (55)	Input *6)	Input *6)	Input
	MPBUS*3	Bus Mode Selection	IN	Setting this to the "L" level selects a separate bus. Setting this to the "H" level selects a multiplex bus. This should be fixed at either the "H" or "L" level.	1 (17)	Input *3)	Input *3)	Input *3)
Split bus interface	SD7-0	Split Data Bus	I/O	If a split bus is selected, this functions as the data bus for the split bus.	8 (43-50)	Input (Hi-z)	Input (Hi-z)	Input (Hi-z)
DMA bus interface	DREQ0_N*1 DREQ1_N*1	DMA Request	OUT	This notifies the system of a D0FIFO port or D1FIFO port DMA transfer request.	2 (57, 60)	H	H	H/L *8)
	DACK0_N*1 DACK1_N*1	DMA Acknowledge	IN	Input the DMA Acknowledge signal for the D0FIFO or D1FIFO port.	2 (58, 61)	Input	Input	Input
	DSTB0_N*2	Data Strobe 0	IN	This functions as the data strobe signal for the D0FIFO port. Because it is also used for the DMA Acknowledge signal of the D1FIFO port, the DSTB0_N function cannot be used if the DACK1_N function is being used.				
	DEND0_N*1 DEND1_N*1	DMA Transfer End	I/O	<In the FIFO port access writing direction> This receives the Transfer End signal from another peripheral chip or the CPU as an input signal. <In the FIFO port access reading direction> This indicates the transfer end data as an output signal.	2 (59, 62)	Input (Hi-z)	Input (Hi-z)	Input (Hi-z)
Interrupt/ SOF output	INT_N	Interrupt	OUT	In the "L" active state, this notifies the system of various types of interrupts relating to USB communication.	1 (51)	H	H	H
	SOF_N	SOF pulse output	OUT	When an SOF is detected in the "L" active state, an SOF pulse is output.	1 (52)	H	H	H
Clock	XIN	Oscillation input	IN	A crystal oscillator should be connected between XIN and XOUT. When using external clock input, the external clock signal should be connected to XIN, and XOUT should be open.	1 (10)			
	XOUT	Oscillation output	OUT		1 (11)			

Category	Pin name	Name	I/O	Function	Pin count (Pin nos.)	State of pin *7)		
						RST_N="L"	RST_N goes "H"	PCUT=1
System control	RST_N	Reset signal	IN	At "L" level, the controller is initialized.	1 (63)	Input (L)	Input (H)	Input (H)
	TEST	Test signal	IN	This should be fixed at "L" or open.	1 (16)			
USB bus interface	DP	USB D+ data	I/O	This should be connected to the D+ pin of the USB bus.	1 (4)	Input (Hi-z)	Input (Hi-z)	Input (Hi-z)
	DM	USB D- data	I/O	This should be connected to the D- pin of the USB bus.	1 (3)	Input (Hi-z)	Input (Hi-z)	Input (Hi-z)
VBUS monitor input	VBUS	VBUS input	IN	This should be connected directly to the Vbus of the USB bus. The connected or disconnected state of the Vbus can be detected. If This pin is not connectted with Vbus of a USB bus, connect it with 5V.	1 (5)	Input (Hi-z)	Input (Hi-z)	Input (Hi-z)
Reference resistance	REFRIN	Reference input	IN	This should be connected to AFEA33G through a 5.6 kΩ±1% resistance.	1 (8)			
Power supply / GND	AFEA33V	Transceiver unit analog power supply	-	This should be connected to 3.3 V.	1 (12)			
	AFEA33G	Transceiver unit analog GND	-		1 (9)			
	AFED33V	Transceiver unit digital power supply	-	This should be connected to 3.3 V.	1 (2)			
	AFED33G	Transceiver unit digital GND	-		1 (1)			
	AFEA15V	Transceiver unit analog 1.5 V power supply	-	This should be connected to 1.5 V.	1 (6)			
	AFEA15G	Transceiver unit analog GND	-		1 (7)			
	AFED15V	Transceiver unit digital 1.5 V power supply	-	This should be connected to 1.5 V.	1 (13)			
	AFED15G	Transceiver unit digital GND	-		1 (14)			
	VDD	Core power supply	-	This should be connected to 1.5 V.	1 (40)			
	VIF	IO power supply	-	This should be connected to 3.3 V or 1.8 V.	3 (15, 42, 64)			
	DGND	Digital GND	-		1 (41)			

***1)** The "L" active and "H" active states of these pins can be specified using the control program for the user system. "_N" indicates that the "L" active state is the default state.

***2)** DSTB0_N and DACK1_N are assigned to the same pin, so the functions of one or the other are valid.

***3)** The input level of the MPBUS pin needs to be established just before the end of H/W reset. Also, this should not be switched during operation.

***4)** When CS_N and RD_N are "L", these pins output "H" or "L".

***5)** If MPBUS is "H", these pins can be made to open.

***6)** CS_N, WR0_N, and WR1_N should be kept as (a) or (b) during RST_N="L" (from RST_N goes "L" to right after RST_N goes "H").
(a) CS_N="H"
(b) WR0_N="H" and WR1_N="H"

***7)** Discription of "State of pin"
(a) Input : Pins are Hi-z state. Please do not make it "open" on a board.
(b) Input(Hi-z) : Pins are Hi-z state. Pins can be "open" on a board.
(c) H, L, H/L : Output states is shown.

***8)** These pins are in an inactive state.

Table 1.2 The example of not used pins

Category	Pin name	Description
SPLIT bus interface	SD7-0	“Open”
DMA bus interface	DREQ0_N	“Open”
	DREQ1_N	“Open”
	DACK0_N	Fix to “H” *1)
	DACK1_N/DSTB0_N	Fix to “H” *1)
	DEND0_N	“Open” *2)
	DEND1_N	“Open” *2)
SOF output	SOF_N	“Open”
System control	TEST	Fix to “L” or “Open”
VBUS monitor input	VBUS	Fix to 5V *3)

***9)** When DACKn_N pin is not used, please set DACKA bit of DMAnCFG register as "0" (n=0,1).
***10)** When DENDx_N pin is not used, please set DENDA bit of DMAnCFG register as "0" (n=0,1).
***11)** If this pin is not connected with Vbus of a USB bus, fix to 5V.

1.5 Pin function configuration

Figure 1.3 shows a diagram of the pin function configuration of the controller.

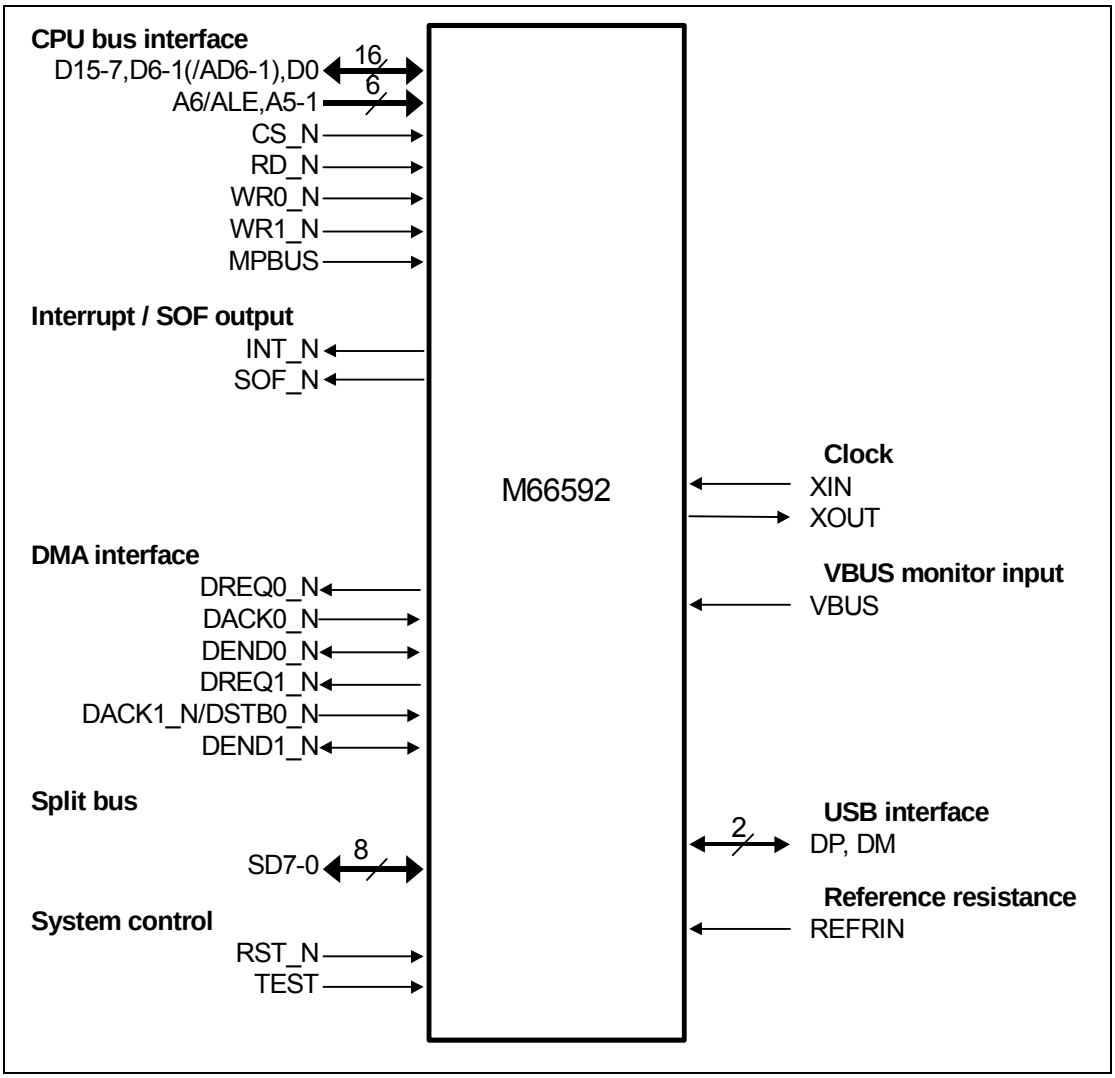


Figure 1.3 Pin function configuration diagram

1.6 Block diagram

The controller is configured with an analog front end unit (AFE), a protocol engine unit (Prtcl_Eng) that includes an SIE, a pipe control unit (Pipe_Ctrl), a FIFO port unit (FIFO_Port), a buffer memory unit (Buf_Mem), an interrupt control unit (Int_Ctrl), a bus interface unit (BIU), and a CPU interface register unit (CPU_IF_Reg). Figure 1.4 shows a block diagram of the controller.

When data is being sent and received between the controller and a host controller connected on the USB bus, a buffer memory assigned to each of the pipes is used. Two-way communication is possible by the controller changing data stored in the buffer memory into USB data packets and outputting them to the USB bus using serial output, and by inputting data packets on the USB bus which are then stored in the buffer memory.

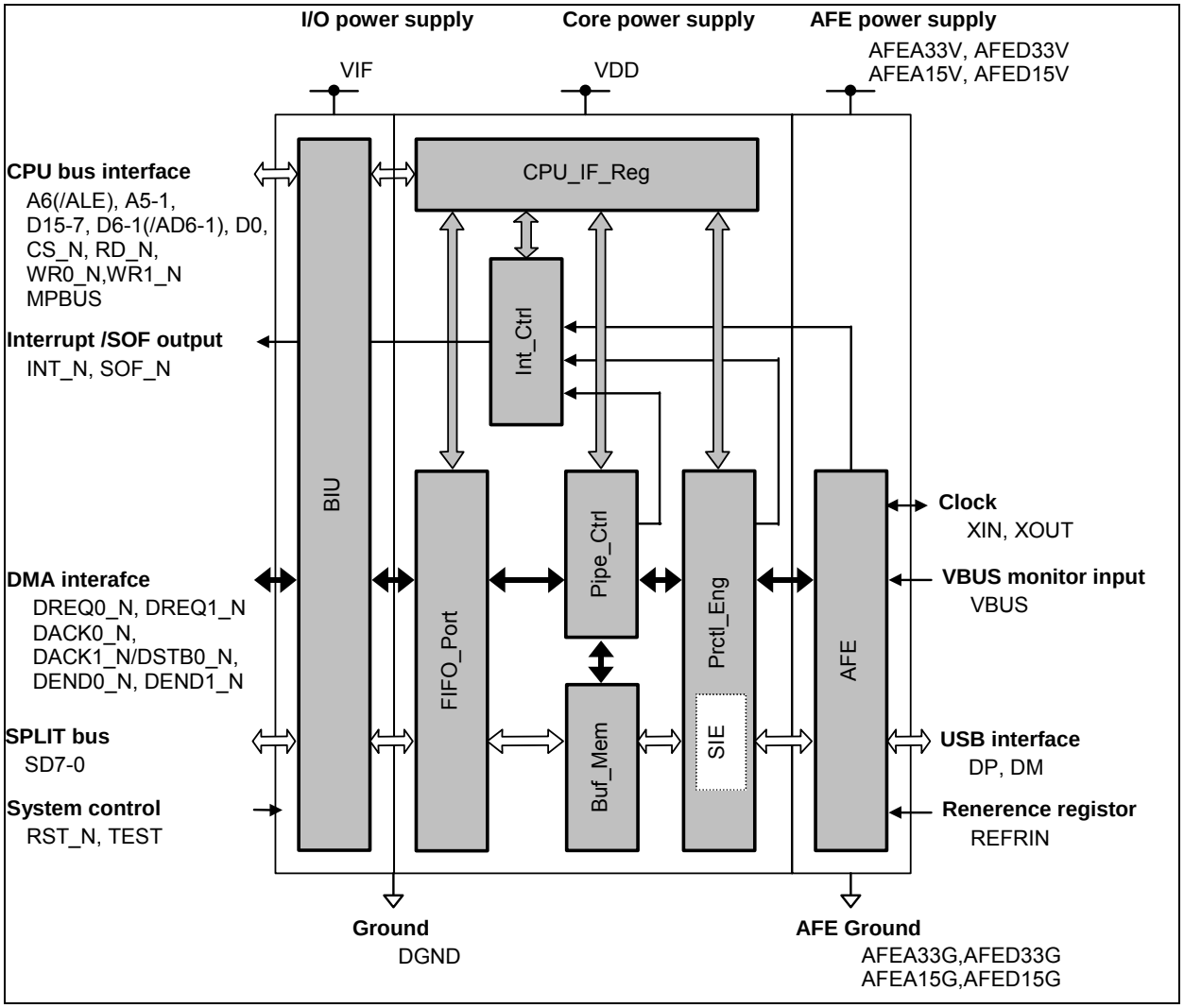


Figure 1.4 Block diagram

1.7 An overview of functions

1.7.1 Bus interfaces

The controller supports the bus interfaces noted below.

1.7.1.1 External bus interface

The controller uses a CPU bus interface to access control registers.

The bus interface with the CPU supports the two access methods noted below. The Chip Select pin (CS_N) and the three strobe pins (RD_N, WR0_N and WR1_N) should be used for access.

(1) (16-bit separate bus
The six address buses (A6-1) and the 16 data buses (D15-0) are used.

(2) 16-bit multiplex bus
The ALE pin (ALE) and the 16 data buses (D15-0) are used. The data buses are used for addresses and data on a time-shared basis.

The separate bus and multiplex bus are selected based on the MPBUS pin signal level when the H/W reset is canceled.

1.7.1.2 Accessing the buffer memory

The controller supports the two methods described below to access the USB data transfer buffer memory.

(1) CPU access
Addresses and control signals should be used to write data to the buffer memory or read it from the buffer memory.

(2) DMA access
Data should be written to the buffer memory of the controller, or read from the buffer memory, from the DMAC in the CPU or a dedicated DMAC.

USB data communication is done using Little Endian. There is a byte Endian swap function for FIFO port access, and when using 16-bit access, the Endian can be switched using the register settings.

1.7.1.3 DMA access methods

When using DMA access to access the buffer memory, the two access methods noted below can be selected.

(1) Method using a bus shared with the CPU

(2) Method using a dedicated bus (split bus)

1.7.2 USB events

The controller notifies the user's system of USB operation events by means of interrupts. Moreover, with a pipe for which the DMA interface has been selected, the system is notified that the buffer memory of the controller can be accessed by asserting the DREQ signal.

There are eight types of interrupts and 35 causes for interrupts being generated. The user can select whether or not interrupt notification is permitted for each type and each cause, using settings in the control program for the user system.

1.7.3 USB data transfers

The controller is capable of all types of transfers: USB communication control transfers, bulk transfers, and interrupt transfers, as well as isochronous data transfers. The pipes noted below can be used with data transfers for various types of communication.

(1) Dedicated control transfer pipe
(2) Two dedicated interrupt transfer pipes
(3) Three dedicated bulk transfer pipes
(4) Two pipes for which bulk transfers or isochronous transfers can be selected

The settings necessary for USB transfers, such as the transfer type, end point number, and maximum packet size, should be set for each pipe, in conjunction with the user system.

Also, the controller has a built-in 5 kB buffer memory. For a dedicated bulk transfer pipe and the pipes for which bulk transfers or isochronous transfers can be selected, settings such as the buffer memory assignment and buffer operation mode which are based on the user system should be entered. The buffer operation mode setting can be set to enable high-speed data transfers with few interrupts, using the double-buffer configuration and data packet continuous transfer function.

Access to the buffer memory from the control CPU of the user's system and the DMA controller is done through the three FIFO port registers.

1.7.4 DMA interface

The DMA (Direct Memory Access) interface consists of data transfers between the user system and the controller using the DxFIFO port, and is a type of data transfer in which the CPU is not involved. The controller is equipped with a 2-channel DMA interface and has the following functions.

(1) A transfer end notification function using the Transfer End signal (DEND signal)
(2) An auto-clear function activated when a Zero-Length packet is received
(3) A "send addition" function used to send a Zero-Length packet based on input of the Transfer End signal (DEND signal)
(4) A transfer end function using a transaction counter function

The controller supports the two types of DMA interfaces noted below.

(1) Cycle steal transfer
With this type of transfer, the DREQ pin is repeatedly asserted and negated each time a data transfer (1 byte / 1 word) is carried out.
(2) Burst transfer
With this type of transfer, the DREQ pin remains asserted for the pipe buffer area assigned to the pertinent FIFO port, or until the transfer is ended by the DEND signal, without ever being negated.

Also, the following can be selected as the DMA interface handshake signal (pin): CS_N, RD_N, or WR_N, or DACK_N. With DMA transfers using a split bus, high-speed DMA transfers are possible by changing the data setup timing, by operating the **OBUS** bit of the **DMAxCFG** register.

1.7.5 SOF pulse output function

An SOF pulse output function is provided that notifies the system of the timing at which SOF packets are received. This function outputs pulses at fairly regular intervals, using an SOF interpolation timer, even if an SOF packet is damaged.

1.7.6 External elements integration

The controller has the following external elements built into it. Also, because the VBUS pin can withstand 5 V, the user system can input the VBUS signal directly to the controller.

(1) Resistors necessary for D+ and D- line control
This makes it possible to configure the system without adding an external resistor.

(2) 48 MHz and 480 MHz PLL
One of three external clocks (12 MHz / 24 MHz / 48 MHz) can be selected and Hi-Speed and Full-Speed operation carried out.

Providing this many external elements in the controller and using a 64-pin compact package mean that less space is required for mounting in the user system.

1.7.7 Low-power sleep state function

The controller is equipped with a low-power sleep state that reduces current consumption.
The low-power sleep state functions effectively under the following circumstances.

(1) When there is no host controller connected
(2) When the device state is shifted to the suspended state, and USB data transfer is not necessary

The system is returned from the low-power sleep state to the normal operating state using a designated interrupt, or by dummy writing to the controller.

2 Registers

Reading the table of registers

① Bit no. — Each register is connected to a 16-bit internal bus.
Odd-numbered addresses will use b15 to b8, and even-numbered addresses b7 to b0.

② Status after reset — This indicates the default state of the register immediately after a reset operation, and after recovering from the low-power sleep state.
H/W Reset is the default state when an external reset signal has been input from the RST_N pin.
S/W Reset is the default state when the user system has carried out a bit operation using the USBE bit.
USB Reset is the default state when the controller has detected a USB bus reset.
Low-power Sleep is the default state when the controller has recovered from the low-power sleep state.
Items that require particular attention during a reset operation are noted under " Notes".
"-" indicates a state in which there is no operation by the controller, and the user setting is retained.
"?" indicates that a value is undecided.

④ S/W Access Condition — This is the condition in effect if the user system control program is accessing a register.
⑤ H/W Access Condition — This is the condition in effect if the controller is accessing a register during any operation other than a reset.

R ..... Read Only
W ..... Write Only
R/W ..... Read / Write
R(0) ..... "0"Read Only
W(1) ..... "1"Write Only

⑥ Note — This is the number of detailed explanations and the number of notes.
⑦ Bit Name — This indicates the bit symbol and bit name.
⑧ Function Description — This describes active items and notes.

<Example of table notation>
Nothing is placed in shaded sections. These should be fixed at "0".

① Bit number →	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit symbol →		A bit	B bit	C bit												
②H/W reset →	?	0	0	0												
S/W reset →	?	0	0	-												
USB reset →	?	0	-	-												
Low-power sleep state →	?	0	0	0												

Bit	Name	Function	S/W	H/W	Note
15	Nothing is placed here. It should be fixed at "0".				
14	A bit AAA enabled	0 : Operation disabled 1 : Operation enabled	R/W	R	2.3.1 *1
13	B bit BBB operation	0 : "L"output 1 : "H"output	R	W	2.3.2 *1
12	C bit CCC control	0 : ..... 1 :	R(0)/ W(1)	R	2.3.2
	⑦	⑧	④	⑤	⑥

«Note»

***1)** If the A bit and B bit are being accessed in succession for writing, an access cycle of 300 ns is necessary.

2.1 Table of registers

Table 2.1 shows the controller registers.

Table 2.1 Registers

Address	Symbol	Name	Index
00	SYSCFG	System configuration control register	2.3
02	SYSSTS	System configuration status register	2.3
04	DVSTCTR	Device state control register	2.4
06	TESTMODE	Test mode register	2.4
08			
0A	PINCFG	Data pin configuration register	2.5
0C	DMA0CFG	DMA0 pin configuration register	2.5
0E	DMA1CFG	DMA1 pin configuration register	2.5
10	CFIFO	CFIFO port register	2.6
12			
14	D0FIFO	D0FIFO port register	2.6
16			
18	D1FIFO	D1FIFO port register	2.6
1A			
1C			
1E	CFIFOSEL	CFIFO port selection register	2.6
20	CFIFOCTR	CFIFO port control register	2.6
22	CFIFOSIE	CFIFO port SIE register	2.6
24	D0FIFOSEL	D0FIFO port selection register	2.6
26	D0FIFOCTR	D0FIFO port control register	2.6
28	D0FIFOTRN	D0 transaction counter register	2.6
2A	D1FIFOSEL	D1FIFO port selection register	2.6
2C	D1FIFOCTR	D1FIFO port control register	2.6
2E	D1FIFOTRN	D1 transaction counter register	2.6
30	INTENB0	Interrupt enable register 0	2.7
32	INTENB1	Interrupt enable register 1	2.7
34			
36	BRDYENB	BRDY interrupt enable register	2.7
38	NRDYENB	NRDY interrupt enable register	2.7
3A	BEMPENB	BEMP interrupt enable register	2.7
3C	SOFCFG	SOF pin configuration register	2.8
3E			
40	INTSTS0	Interrupt status register 0	2.9
42			
44			
46	BRDYSTS	BRDY interrupt status register	2.9
48	NRDYSTS	NRDY interrupt status register	2.9
4A	BEMPSTS	BEMP interrupt status register	2.9
4C	FRMNUM	Frame number register	2.10
4E	UFRMNUM	Micro frame number register	2.10
50	RECOVER	USB address / low-power status recovery register	2.11
52			
54	USBREQ	USB request type register	2.12
56	USBVAL	USB request value register	2.12
58	USBINDX	USB request index register	2.12
5A	USBLENG	USB request length register	2.12
5C	DCPCFG	DCP configuration register	2.13
5E	DCPMAXP	DCP maximum packet size register	2.13

Address	Symbol	Name	Index
60	DCPCTR	DCP control register	2.13
62			
64	PIPESEL	Pipe window selection register	2.14
66	PIPECFG	Pipe configuration register	2.14
6E	PIPEBUF	Pipe buffer setting register	2.14
6A	PIPEMAXP	Pipe maximum packet size register	2.14
6C	PIPEPERI	Pipe period control register	2.14
6E			
70	PIPE1CTR	Pipe 1 control register	2.14
72	PIPE2CTR	Pipe 2 control register	2.14
74	PIPE3CTR	Pipe 3 control register	2.14
76	PIPE4CTR	Pipe 4 control register	2.14
78	PIPE5CTR	Pipe 5 control register	2.14
7A	PIPE6CTR	Pipe 6 control register	2.14
7C	PIPE7CTR	Pipe 7 control register	2.14
7E			

Nothing is placed in addresses that are shaded. These addresses should not be accessed.

2.2 Table of bit symbols

Table 2.2shows the controller bit symbols.

Table 2.2 Bit symbols

Addr	Register name	Odd-numbered addresses								Even-numbered addresses							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
00	SYSCFG	XTAL		XCKE	RCKE	PLLC	SCKE		ATCKM	HSE			DPRPU			PCUT	USBE
02	SYSSTS															LNST	
04	DVSTCTR								WKUP							RHST	
06	TESTMODE														UTST		
08																	
0A	PINCFG	LDRV							BIGEND								
0C	DMA0CFG		DREQA	BURST			DACKA	DFORM			DENDA	PKTM	DENDE		OBUS		
0E	DMA1CFG		DREQA	BURST			DACKA	DFORM			DENDA	PKTM	DENDE		OBUS		
10	CFIFO	CFPORT															
12																	
14	D0FIFO	D0FPORT															
16																	
18	D1FIFO	D1FIPORT															
1A																	
1C																	
1E	CFIFOSEL	RCNT	REW				MBW					ISEL			CURPIPE		
20	CFIFOCTR	BVAL	BCLR	FRDY		DTLN											
22	CFIFOSIE	TGL	SCLR	SBUSY													
24	D0FIFOSEL	RCNT	REW	DCLRM	DREQE		MBW	TRENB	TRCLR	DEZPM					CURPIPE		
26	D0FIFOCTR	BVAL	BCLR	FRDY		DTLN											
28	D0FIFOTRN	TRNCNT															
2A	D1FIFOSEL	RCNT	REW	DCLRM	DREQE		MBW	TRENB	TRCLR	DEZPM					CURPIPE		
2C	D1FIFOCTR	BVAL	BCLR	FRDY		DTLN											
2E	D1FIFOTRN	TRNCNT															
30	INTENB0	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
32	INTENB1														BRDYM	INTL	PCSE
34																	
36	BRDYENB	PIPEBRDYE															
38	NRDYENB	PIPENRDYE															
3A	BEMPENB	PIPEBEMPE															
3C	SOFCFG													SOFM			
3E																	
40	INTSTS0	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ			VALID	CTSQ		
42																	
44																	
46	BRDYSTS	PIPEBRDY															
48	NRDYSTS	PIPENRDY															
4A	BEMPSTS	PIPEBEMP															
4C	FRMNUM	OVRN	CRCE			SOFRM	FRNM										
4E	UFRMNUM														UFRNM		
50	RECOVER						STSRECOV				USBADDR						
52																	
54	USBREQ	bRequest								bmRequestType							
56	USBVAL	wValue															
58	USBINDX	wIndex															
5A	USBLENG	wLength															
5C	DCPCFG								CNTMD								
5E	DCPMAXP										MXPS						
60	DCPCTR	BSTS							SQCLR	SQSET	SQMON				CCPL	PID	
62																	

Addr	Register name	Odd-numbered addresses								Even-numbered addresses							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
64	PIPESEL														PIPESEL		
66	PIPECFG	TYPE					BFRE	DBLB	CNTMD	SHTNAK			DIR	EPNUM			
68	PIPEBUF		BUFSIZE							BUFNMB							
6A	PIPEMAXP						MXPS										
6C	PIPEPERI				IFIS										IITV		
6E																	
70	PIPE1CTR	BSTS	INBUFM					ACLRM	SQCLR	SQSET	SQMON					PID	
72	PIPE2CTR	BSTS	INBUFM					ACLRM	SQCLR	SQSET	SQMON					PID	
74	PIPE3CTR	BSTS	INBUFM					ACLRM	SQCLR	SQSET	SQMON					PID	
76	PIPE4CTR	BSTS	INBUFM					ACLRM	SQCLR	SQSET	SQMON					PID	
78	PIPE5CTR	BSTS	INBUFM					ACLRM	SQCLR	SQSET	SQMON					PID	
7A	PIPE6CTR	BSTS						ACLRM	SQCLR	SQSET	SQMON					PID	
7C	PIPE7CTR	BSTS						ACLRM	SQCLR	SQSET	SQMON					PID	
7E																	

2.3 System control

◆ System configuration control register [**SYSCFG**] <Address : 00H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTAL		XCKE	RCKE	PLLC	SCKE		ATCKM	HSE			DPRPU			PCUT	USBE
0	0	0	0	0	0	?	0	0	?	?	0	?	?	0	0
-	-	-	-	-	-	?	-	-	?	?	-	?	?	-	-
-	-	-	-	-	-	?	-	-	?	?	-	?	?	-	-
-	-	1	0	0	0	?	-	-	?	?	-	?	?	0	-

Bit	Name	Function	S/W	H/W	Note
15-14	XTAL Clock selection	00: 12 MHz input 01: 24 MHz input 10: 48 MHz input 11: Reserved	R/W	R	3.1.5
13	XCKE Oscillation buffer enabled	0: Oscillation buffer operation disabled 1: Oscillation buffer operation enabled	R/W	R/W(1)	3.1.5 *2)
12	RCKE Reference clock enabled	0: Reference clock supply stopped 1: Reference clock supply enabled	R/W	R	3.1.5
11	PLLC PLL operation enabled	0: PLL operation disabled 1: PLL operation enabled	R/W	R	3.1.5
10	SCKE Internal clock enabled	0: Internal clock supply stopped 1: Internal clock supply enabled	R/W	R	3.1.5
9	Nothing is placed here. It should be fixed at “0”.				
8	ATCKM Auto clock supply function enabled	The clock is supplied from the low-power sleep state or clock stop state. 0: Auto clock supply function disabled 1: Auto clock supply function enabled	R/W	R	3.1.6.6
7	HSE Hi-Speed operation enabled	This enables Hi-Speed operation. 0: Hi-Speed operation disabled (Full-Speed) 1: Hi-Speed operation enabled (detected by controller)	R/W	R	3.1.3 *1)
6-5	Nothing is placed here. These should be fixed at “0”.				
4	DPRPU D+ line pull-up control	Issues notification of connection to host controller. 0: Pull-up disabled 1: Pull-up enabled	R/W	R	3.1.4
3-2	Nothing is placed here. These should be fixed at “0”.				
1	PCUT Low-power sleep state enabled	0: Normal operation state 1: Low-power sleep state	R/W(1)	R/W(0)	3.1.6
0	USBE USB block operation enabled	0: USB block operation disabled (S/W Reset) 1: USB block operation enabled	R/W	R	2.3.1 3.1.1

«Notes»

*1) The Hi-Speed operation enabled((**HSE**) bit should be set beforethe **DPRPU** bit is set to “1”.

*2) When the system returns from the low-power sleep state to the normal operation state, the controller sets “XCKE = 1”.

◆ System configuration status register [**SYSSTS**] <Address: 02H>

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

Bit	Name	Function	S/W	H/W	Note
15-2	Nothing is placed here. These should be fixed at "0".				
1-0	LNST USB data line status	Please see the detailed explanation concerning this item.	R	W	2.3.2

«Notes»

None in particular

2.3.1 USB block operation enabled

The **USBE** bit of the **SYSCFG** register should be used to enable USB block operation.

The same bit can be used to carry out an S/W reset of the controller. When software is set to "USBE=0", the controller resets the register targeted for S/W reset initialization to the default setting value. As long as "USBE=0" is set, no data can be written by software to the bit targeted for S/W reset initialization. "USBE=1" should be set following an S/W reset to enable controller operation.

2.3.2 Line status monitor

Table 2.3 shows the USB data bus line statuses of the controller. The controller monitors the line status (D+ line and D- line) of the USB data bus using the **LNST** bit of the **SYSSTS** register. The **LNST** bit is configured of two bits. For the meaning of each of the bits, please refer to the table below. The timing at which the **LNST** bit becomes valid differs depending on the selected controller function. In the normal operating state, the line status can be monitored on an ongoing basis, but in the low-power sleep state, the line status cannot be monitored.

Table 2.3 USB data bus line statuses

LNST [1]	LNST [0]	During Full-Speed operation	During Hi-Speed operation	During chirp operation
0	0	SE0	Squelch	Squelch
0	1	J State	not Squelch	Chirp J
1	0	K State	Invalid	Chirp K
1	1	SE1	Invalid	Invalid

Chirp: The reset handshake protocol is being executed in the Hi-Speed operation enabled state (HSE = "1").
Squelch: SE0 or Idle state
not Squelch: Hi-Speed J state or Hi-Speed K state
Chirp J: Chirp J state
Chirp K: Chirp K state

2.4 USB signal control

◆ Device state control register [**DVSTCTR**] <Address: 04H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							WKUP							RHST	
?	?	?	?	?	?	?	0	?	?	?	?	?	?	0	0
?	?	?	?	?	?	?	0	?	?	?	?	?	?	0	0
?	?	?	?	?	?	?	0	?	?	?	?	?	?	-	-
?	?	?	?	?	?	?	0	?	?	?	?	?	?	0	0

Bit	Name	Function	S/W	H/W	Note
15-9	Nothing is placed here. These should be fixed at "0".				
8	WKUP Wakeup output	0: Non-output 1: Remote wakeup signal output	R/W(1)	R/W(0)	2.4.1 *1), *2)
7-2	Nothing is placed here. These should be fixed at "0".				
1-0	RHST Reset handshake	00: Communication speed not decided 01: Reset handshake being processed 10: Full-Speed operation established 11: Hi-Speed operation established	R	W	2.4.2

«Notes»

*1) "1" should never be written to the **WKUP** bit unless "Suspended" is set for the device state ("DVSQ = 1x") and a remote wakeup from the USB host is enabled.

*2) When the **WKUP** bit is set to "1", software should not disable oscillation buffer operation.

◆ Test mode register [**TESTMODE**] <Address: 06H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													UTST		
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	-	-	-
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-4	Nothing is placed here. These should be fixed at "0".				
3-0	UTST Test mode	Please see the detailed explanation concerning this item.	R/W	R	2.4.3 *3)

«Note»

*3) The **UTST** bit is valid only during Hi-Speed operation. Check to make sure the "RHST=11" before using it.

2.4.1 USB data bus control

Each bit of the **DVSTCTR** register can be used to control and confirm the state of the USB data bus based on the user system.

The **WKUP** bit handles control of remote wakeup signal output to the USB bus. The controller controls the output time for remote wakeup signals. 2ms after a software has set "1" for the **WKUP** bit, the controller outputs a 10 ms "K-State" then it transfers the bus state to idle. When the bus state is transferred to the idle state, the controller sets "WKUP=0".

According to the USB specification, USB idle state must be kept longer than 5ms. Thus if the software set "WKUP=1" right after detection of Suspend state, the controller will assert "K-State" after 2ms.

2.4.2 Communication speed discrimination

Software is able to confirm the USB speed at which communication is being carried out with the host controller (the communication bit rate), using the **RHST** bit.

If Hi-Speed operation has been set to the disabled state ("HSE=0") by software, the controller immediately establishes Full-Speed operation ("RHST=10") after a USB bus reset has been detected, without executing the reset handshake protocol. If Hi-Speed operation has been set to the enabled state ("HSE=1"), the controller executes the reset handshake protocol ("RHST=01" during execution of the protocol), and feeds back the execution results to the **RHST** bit ("RHST=11" : Hi-Speed operation, or "RHST=10" : Full-Speed operation).

2.4.3 Test mode

Table 2.4 shows the test mode operation of the controller. The **UTST** bit of the **TESTMODE** register controls the USB test signal output during Hi-Speed operation.

Table 2.4 Test mode operation

Test mode	UTST bit setting
Normal operation	000
Test_J	001
Test_K	010
Test_SE0_NAK	011
Test_Packet	100
Reserved	101-111

2.5 External input/output control

◆ Data pin configuration register [**PINCFG**] <Address: 0AH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LDRV							BIGEND								
0	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?
-	?	?	?	?	?	?	-	?	?	?	?	?	?	?	?
-	?	?	?	?	?	?	-	?	?	?	?	?	?	?	?
-	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Note
15	LDRV Output pins drive current control	0: When VIF=1.6-2.0 V 1: When VIF=2.7-3.6 V	R/W	R	2.5.1
14-9	Nothing is placed here. These should be fixed at “0”.				
8	BIGEND FIFO port Endian	0: Little Endian 1: Big Endian	R/W	R	2.5.2 *1)
7-0	Nothing is placed here. These should be fixed at “0”.				

«Note»

*1) The **BIGEND** bit is common to all of the FIFO ports and available for the FIFO ports only. The **BIGEND** bit doesn’t affects register acess.

The **DMA0CFG** register controls the input/output pins used for the DMA0 interface and the D0FIFO port, and the **DMA1CFG** register controls the input/output pins used for the DMA1 interface and the D1FIFO port.

◆ DMA0 pin configuration register [**DMA0CFG**] <Address: 0CH>
◆ DMA1 pin configuration register [**DMA1CFG**] <Address: 0EH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DREQA	BURST			DACKA	DFORM			DENDA	PKTM	DENDE		OBUS		
?	0	0	?	?	0	0	0	0	0	0	0	?	0	?	?
?	-	-	?	?	-	-	-	-	-	-	-	?	-	?	?
?	-	-	?	?	-	-	-	-	-	-	-	?	-	?	?
?	-	0	?	?	0	0	0	0	0	0	0	?	0	?	?

Bit	Name	Function	S/W	H/W	Note
15	Nothing is placed here. This should be fixed at “0”.				
14	DREQA DREQx_N signal polarity selection	This specifies the active state for the DREQx_N pin. 0: Low active 1: High active	R/W	R	-
13	BURST Burst mode	0: Cycle steal transfer 1: Burst transfer	R/W	R	2.5.3
12-11	Nothing is placed here. These should be fixed at “0”.				
10	DACKA DACKx_N signal polarity selection	This specifies the active state for the DACKx_N pin. 0: Low active 1: High active	R/W	R	-
9-7	DFORM DMA transfer signal selection	011: Only the DACKx_N signal is used (CPU bus). 000: The Address signal + the RD_N/WRx_ signals are used (CPU bus). 010: The DACKx_N + the RD_N/WRx_N signals are used (CPU bus). 100: The DACKx_N signal is used (split bus). 110: The DACK0_N + the DSTB0_N signal are used (split bus). 001, 101, 111: Reserved	R/W	R	3.4.3.2 *3)
6	DENDA DEND0_N signal polarity selection	This specifies the active state of the DENDx_N pin. 0: Low active 1: High active	R/W	R	-
5	PKTM Packet mode	0: The DENDx_N signal is asserted in transfer units. 1: The DENDx_N signal is asserted each time an amount of data corresponding to the buffer size is transferred.	R/W	R	2.5.3 3.4.3.4 *2)
4	DENDE DENDx_N signal enabled	0: The DENDx_N signal is disabled (Hi-Z output). 1: The DENDx_N signal is enabled.	R/W	R	2.5.3 3.4.3.4
3	Nothing is placed here. It should be fixed at “0”.				
2	OBUS OBUS operation disabled	0: The OBUS mode is enabled. 1: The OBUS mode is disabled.	R/W	R	3.5
1-0	Nothing is placed here. These should be fixed at "0".				

«Notes»

*2) The **PKTM** bit is valid only when the data receiving direction (reading from the buffer memory) is set. If the DxFIFO port is being used in the data writing direction, “PKTM=0” should be set.

*3) The “DFORM=110” setting is valid only when the DMA channel 0 is set.
Also, the following should not be set: "DFORM=001”,"DFORM=101” and"DFORM=111”.

2.5.1 Output pins drive current control

The output pins drive capability should be set using the **LDRV** bit of the **PINCFG** register, to match the VIF power supply.

The output pins are the **SD7-0, D15-0, INT_N, DREQx_N, DENDx_N**, and **SOF_N** pins.

2.5.2 FIFO port access Endian

Table 2.5 and Table 2.6 show the byte Endian operation of the controller. (The controller uses Little Endian.) When the CPU of user-system is Big-endian, software should be set the **BIGEND** bit to "1" of the **PINCFG** register.

Table 2.5 Endian operation when using 16-bit access

BIGEND	b15 – b8	b7 – b0
0	Odd-numbered addresses	Even-numbered addresses
1	Even-numbered addresses	Odd-numbered addresses

Table 2.6 Endian operation when using 8-bit access

BIGEND	b15 – b8	b7 – b0
0	Writing: invalid Reading: invalid	Writing: valid Reading: valid
1	Writing: valid Reading: valid	Writing: invalid Reading: invalid

2.5.3 DMA signal control

When transferring data using the DMA interface, the DMA operations (assertion and negation of the **DREQx_N** and **DENDx_N** signals, and the DMA transfer mode) should be specified to match the user system, using the **BURST**, **PKTM**, **DENDE**, and **OBUS** bits of the **DMAxCFG** register. The DMA signals are valid for the selected pipe(s) as long as DMA transfers are enabled using the **DREQE** bit of the **DxFIFOSEL** register, which will be explained later. The **DREQx_N** pin is asserted when the buffer memory of the pipe is in the Buffer Ready (BRDY) state.

2.6 FIFO ports

The transmission and reception buffer memory of the controller uses the FIFO configuration. The FIFO port registers should be used to access the buffer memory. There are three FIFO ports: the CFIFO port, D0FIFO port, and D1FIFO port. Each FIFO port is configured of a port register that handles reading of data from the buffer memory, as well as writing of data to the memory, a selection register used to select the pipe assigned to the FIFO port, a control register, and registers used specifically for port functions (an SIE register used exclusively for the CFIFO port, and a transaction counter register used exclusively for the DxFIFO port).

The Notes noted below apply to each of the FIFO ports. For more detailed information, please refer to Chapter 3.4, Buffer memory.

(1) The DCP buffer memory can only be accessed through the CFIFO port.
(2) Accessing the buffer memory using DMA transfer can be done only through the DxFIFO port.
(3) Accessing the DxFIFO port using the CPU has to be done in conjunction with the functions and restrictions of the DxFIFO port. (Using the transaction counter, etc.)
(4) When using functions specific to the FIFO port, the selected pipe cannot be changed. (Using the transaction counter, signal input/output through DMA-related pins, etc.)
(5) Registers corresponding to a FIFO port never affect other FIFO ports.
(6) The same pipe should not be assigned to two or more FIFO ports.
(7) There are two sorts of buffer memory states; the access right is on the CPU side and it is on the SIE side. When the buffer memory access right is on the SIE side, the memory cannot be properly accessed from the CPU.
(8) The pipe configuration,i.e. **PIPECFG**, **PIPEBUF**, **PIPEMAXP**, **PIPEPERI**, **PIPE1CTR** registers of the pipe selected by the **CURPIPE** bit should not be changed.

◆ CFIFO port register [**CFIFO**] <Address: 10H>
◆ D0FIFO port register [**D0FIFO**] <Address: 14H>
◆ D1FIFO port register [**D1FIFO**] <Address: 18H>

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

Bit	Name	Function	S/W	H/W	Note
15-0	FIFOPORT FIFO port	This handles reading of received data from the buffer memory, or writing of the sent data to the buffer memory.	R/W	R/W	3.4 *1)

«Note»
*1) Only the CFIFO port can be used for DCP access of the buffer memory.
Accessing the buffer memory using DMA transfers can only be done through the D0FIFO and D1FIFO ports.

- CFIFO port selection register [**CFIFOSEL**] <Address: 1EH>
- D0FIFO port selection register [**D0FIFOSEL**] <Address: 24H>
- D1FIFO port selection register [**D1FIFOSEL**] <Address: 2AH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCNT	REW	DCLRM	DREQE		MBW	TRENB	TRCLR	DEZPM		ISEL			CURPIPE		
0	0	0	0	?	0	0	0	0	?	0	?	?	0	0	0
0	0	0	0	?	0	0	0	0	?	0	?	?	0	0	0
-	-	-	-	?	-	-	-	-	?	-	?	?	-	-	-
0	0	0	0	?	0	0	0	0	?	0	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	RCNT Read Count mode	0: The DTLN bit is cleared when all of the reception data has been read. 1: The DTLN bit is decremented when the reception data is read.	R/W	R	3.4.2
14	REW Buffer pointer rewind	0:Invalid. 1: The buffer pointer is rewound.	R(0)/W	R/W(0)	3.4.2
13	DCLRM This is the Auto Buffer Memory clear mode accessed after the data for the specified pipe has been read.	0: The Auto Buffer Clear mode is disabled. 1: The Auto Buffer Clear mode is enabled.	R/W	R	3.4.3 *2)
12	DREQE DREQ signal output enabled	0: Output is disabled. 1: Output is enabled.	R/W	R	3.4.3 *2),
11	Nothing is placed here. This should be fixed at “0”.				
10	MBW FIFO port access bit width	0: 8-bit width 1: 16-bit width	R/W	R	3.4.2 *4)
9	TRENB Transaction counter enabled	0: The transaction counter function is invalid. 1: The transaction counter function is valid.	R/W	R	3.4.2 *2)
8	TRCLR Transaction counter clear	0: Invalid 1: The current count is cleared.	R(0)/ W(1)	R	3.4.2 *2)
7	DEZPM Zero-Length Packet Added mode	0: No packet is added. 1: The packet is added.	R/W	R	3.4.3 *2)
6	Nothing is placed here. This should be fixed at “0”.				
5	ISEL Access direction of the FIFO port when DCP is selected	0: This selects reading from the buffer memory. 1: This selects writing to the buffer memory.	R/W	R	3.4 *3)
4-3	Nothing is placed here. These should be fixed at “0”.				
2-0	CURPIPE FIFO port access pipe specification	000: DCP / No specification 001: Pipe 1 010: Pipe 2 011: Pipe 3 100: Pipe 4 101: Pipe 5 110: Pipe 6 111: Pipe 7	R/W	R	3.4 *5)

«Notes»

***2)** The **DCLRM**, **DREQE**, **TRENB**, **TRCLR** and **DEZPM** bits are valid for the **D0/D1FIFOSEL** registers.
The **DCLRM**, **TRENB** and **TRCLR** bits are valid when the receiving direction (reading from the buffer memory) has been set for the pipe specified by the **CURPIPE** bit.
The **DEZPM** bit is valid when the sending direction (writing to the buffer memory) has been set for the pipe specified by the **CURPIPE** bit.

***3)** The **ISEL** bit is valid only when DCP is selected using the **CFIFO** port selection register.
Software should set the **ISEL** bit according to the folowing (a) or (b)
(a) The setting to **CURPIPE** bit to DCP (“CURPIPE=”0”) and setting to **ISEL** bit should be done at the same time.
(b) First software sets **CURPIPE** bit to DCP (“CURPIPE=”0”), then it sets **ISEL** bit after 200ns or more.

***4)** Once reading from the buffer memory has begun, the access bit width of the FIFO port cannot be changed until all of the data has been read. Also, the bit width cannot be changed from the 8-bit width to the 16-bit width while data is being written to the buffer memory.

***5)** Specifying"CURPIPE=0"using the **D0/D1FIFOSEL** register will be interpreted as no pipe having been specified. Also, the pipe number should not be changed while **DREQ** output is enabled.

◆ CFIFO port control register [**CFIFOCTR**] <Address: 20H>
◆ D0FIFO port control register [**D0FIFOCTR**] <Address: 26H>
◆ D1FIFO port control register [**D1FIFOCTR**] <Address: 2CH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BVAL	BCLR	FRDY		DTLN											
0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	?	-	-	-	-	-	-	-	-	-	-	-	-
0	0	0	?	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	BVAL Buffer Memory Valid flag	0: Invalid 1: Writing ended	R/ W(1)	R/W	3.4.2 *6)
14	BCLR CPU Buffer Clear	0: Invalid 1: Clears the buffer memory on the CPU side.	R(0)/ W(1)	R/W(0)	3.4 *7), *8)
13	FRDY FIFO Port Ready	0: FIFO port access is disabled. 1: The FIFO port can be accessed.	R	W	3.4 *9)
12	Nothing is placed here. This should be fixed at "0".				
11-0	DTLN Reception Data Length	The length of the reception data can be confirmed.	R	W	3.4.2 3.4.4 *7)

«Notes»

***6)** Writing "1" to the **BVAL** bit is valid when the direction of the data packet is the sending direction (when data is being written to the buffer memory). When the direction is the receiving direction, "BVAL=0" should be set.

***7)** The **BCLR** bit and **DTLN** bit are valid for the buffer memory on the CPU side. Software should set "BCLR=1" or refer **DTLN** bit after making sure that "FRDY=1".

***8)** Using the **BCLR** bit to clear the buffer should be done with the pipe invalid state by the pipe configuration ("PID=NAK"). When DCP is selected, the **BCLR** bit has the same function as the **ACLRM** bit of the **PIPExCTR** register.

***9)** The **FRDY** bit requires an access cycle of at least 450 ns after the pipe has been selected.

◆ CFIFO port SIE register [**CFIFOSIE**] <Address: 22H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TGL	SCLR	SBUSY													
0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?
0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?
-	-	-	?	?	?	?	?	?	?	?	?	?	?	?	?
0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Note
15	TGL Access Right Switch	0: Invalid 1: Switches access right	R(0)/ W(1)	R/W(0)	3.4.2 *10)
14	SCLR SIE Buffer Clear	0: Invalid 1: Clears buffer memory on SIE side	R(0)/ W(1)	R/W(0)	3.4 *11)
13	SBUSY SIE Buffer Busy	0: SIE is not being accessed. 1: SIE is being accessed.	R	W	3.4.2
12-0	Nothing is placed here. These should be fixed at "0".				

«Note»

***10)** The function of the **TGL** bit is to set the buffer memory on the SIE side to the CPU side. Set "PID=NAK" and check the **SBUSY** bit to make sure the SIE is not accessing the buffer ("SBUSY=0"). Then write the **TGL** bit (toggle operation). This bit is valid only for pipes for which the reception direction (reading from the buffer memory) has been set.

***11)** The function of the **SCLR** bit is to clear the buffer memory on the SIE side. Set "PID=NAK" and check the **SBUSY** bit to make sure the SIE is not accessing the buffer ("SBUSY=0"). Then clear the buffer. This bit is valid only for pipes for which the sending direction (writing to the buffer memory) has been set.

◆ D0 transaction counter register [**D0FIFOTRN**] <Address: 28H>
◆ D1 transaction counter register [**D1FIFOTRN**] <Address: 2EH>

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

Bit	Name	Function	S/W	H/W	Note
15-0	TRNCNT Transaction counter	Writing: Sets the number of DMA transfer transactions. Reding: Reads the number of transactions.	R/W	R	3.4.2 *12)

«Note»

***12)** The transaction counter is valid when data is being read from the buffer memory.
The number of transactions can be read while counting is taking place only if the **TRENB** bit of the **DxFIFOSEL** register is “1”. If “TRENB=0”, the set number of transactions can be read.

2.7 Interrupts enabled

◆ Interrupts enabled register 0[**INTENB0**] <Address: 30H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	URST	SADR	SCFG	SUSP	WDST	RDST	CMPL	SERR
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	VBSE VBUS interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.8 2.7.1 2.7.4
14	RSME Resume interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.9 2.7.1 2.7.4
13	SOFE Frame number refresh interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.7 2.7.1
12	DVSE Device state transition interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.5 2.7.1 2.7.2
11	CTRE Control transfer stage transition interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.6 2.7.1 2.7.3
10	BEMPE Buffer Empty interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.4 2.7.1
9	NRDYE Buffer Not Ready response interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.3 2.7.1
8	BRDYE Buffer Ready interrupts enabled	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.2 2.7.1
7	URST Default state transition notifications enabled	0: DVST interrupt disabled at transition to default state 1: DVST interrupt enabled at transition to default state	R/W	R	3.2.5 2.7.2
6	SADR Address state transition notifications enabled	0: DVST interrupt disabled at transition to address state 1: DVST interrupt enabled at transition to address state	R/W	R	3.2.5 2.7.2
5	SCFG Configuration state transition notifications enabled	0: DVST interrupt disabled at transition to configuration state 1: DVST interrupt enabled at transition to configuration state	R/W	R	3.2.5 2.7.2
4	SUSP Suspend state transition notifications enabled	0: DVST interrupt disabled. at transition to suspend state 1: DVST interrupt enabled at transition to suspend state	R/W	R	3.2.5 2.7.2
3	WDST Control write transfer status stage transition notifications enabled	0: CTST interrupt disabled at transition to status stage of control write transfer 1: CTST interrupt enabled at transition to status stage of control write transfer	R/W	R	3.2.6 2.7.3
2	RDST Control read transfer status stage transition notifications enabled	0: CTST interrupt disabled at transition to status stage of control read transfer 1: CTST interrupt enabled at transition to status stage of control read transfer	R/W	R	3.2.6 2.7.3
1	CMPL Control transfer end notifications enabled	0: CTST interrupt are disabled at end of control transfer 1: CTST interrupt enabled at end of control transfer	R/W	R	3.2.6 2.7.3
0	SERR Control transfer sequence error notifications enabled	0: CTST interrupt disabled at detection of control transfer sequence error 1: CTST interrupt enabled at detection of control transfer sequence error	R/W	R	3.2.6 2.7.3

«Note»

None in particular

◆ Interrupt enabled register 1[**INTENB1**] <Address: 32H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													BRDYM	INTL	PCSE
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	-	-	0
?	?	?	?	?	?	?	?	?	?	?	?	?	-	-	-
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	-

Bit	Name	Function	S/W	H/W	Note
15-3	Nothing is placed here. These should be fixed at “0”.				
2	BRDYM BRDY interrupt status clear timing control	0: Software clears BRDY interrupt status 1: The controller clears BRDY interrupt status	R/W	R	3.2.2 *1)
1	INTL Interrupt output sensing control	0: Edge sensing 1: Level sensing	R/W	R	3.2.1
0	PCSE returning fact selection from low-power sleep mode	0: USB resume detectionm, VBUS interrupt detection during suspend, or CS_N signal input 1: USB resume detectionm, or VBUS interrupt detection during suspend	R/W	R	3.2.5 *2)

«Note»

*1) When software sets "BRDYM=1", it will set “INTL=1” also.

*2) **PCSE** bit shoud be set after “USBE=1” setting.

◆ BRDY interrupt enabled register [**BRDYENB**] <Address: 36H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIPEBRDYE							
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	-	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-8	Nothing is placed here. These should be fixed at “0”.				
7-0	PIPEBRDYE BRDY interrupts for each pipeis enabled.	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.2 2.7.1 *3)

«Note»

3) The bit numbers correspond to the pipe numbers.

◆ NRDY interrupt enabled register [**NRDYENB**] <Address: 38H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIPENRDYE							
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	-	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-8	Nothing is placed here. These should be fixed at “0”.				
7-0	PIPENRDYE NRDY interrupts for each pipe is enabled.	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.3 2.7.1 *4)

«Note»

4) The bit numbers correspond to the pipe numbers.

◆ BEMP interrupt enabled register [**BEMPENB**] <Address: 3AH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIPEBEMPE							
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	-	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-8	Nothing is placed here. These should be fixed at “0”.				
7-0	PIPEBEMPE BEMP interrupts for each pipe is enabled.	0: Interrupt output disabled 1: Interrupt output enabled	R/W	R	3.2.4 2.7.1 *5)

«Note»

5) The bit numbers correspond to the pipe numbers.

2.7.1 Interrupt masks

The **VBSE**, **RSME**, **SOFE**, **DVSE**, **CTRE**, **BEMPE**, **NRDYE**, and **BRDYE** bits of the **INTENB0** register operate as interrupt mask bits. Each of the bits should be used to specify whether interrupt signal output is enabled or disabled for the **INT_N** pin.

The **BRDYENB** register, **NRDYENB** register and **BEMPENB** register operate as the **BRDY** interrupt mask bit, the **NRDY** interrupt mask bit, and the **BEMP** interrupt mask bit, respectively, for each corresponding pipe. For detailed information, please refer to 3.2 Interrupt functions.

2.7.2 Device state transition interrupts

The **URST**, **SADR**, **SCFG**, and **SUSP** bits of the **INTENB0** register operate as interrupt mask bits for the device state transition interrupt (**DVST**).

If a factor is disabled, no device state transition interrupt is issued in response to the pertinent factor. However, the device state (**DVSQ**) transits in keeping with the circumstances. For detailed information, please refer to 3.2 Interrupt functions.

2.7.3 Control transfer stage transition interrupts

The **WDST**, **RDST**, **CMPL** and **SERR** bits of the **INTENB0** register operate as interrupt mask bits for the control transfer stage transition interrupt (**CTRT**).

If a factor is disabled, no control transfer stage transition interrupts are issued in response to the pertinent factor. For detailed information, please refer to 3.2 Interrupt functions.

2.7.4 Operations in the low-power sleep state

For **VBSE** and **RSME**, interrupts are issued in the low-power sleep state as well.

2.8 SOF control register

◆ SOF pin configuration register [**SOFCFG**] <Address: 3CH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
												SOFM			
?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?
?	?	?	?	?	?	?	?	?	?	?	?	-	-	?	?
?	?	?	?	?	?	?	?	?	?	?	?	-	-	?	?
?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?

Bit	Name	Function	S/W	H/W	Note
15-4	Nothing is placed here. These should be fixed at "0".				
3-2	SOFM SOF pin function setting	This selects the SOF pulse output mode. 00: SOF output disabled 01: SOF output in units of 1 ms 10: uSOF output in units of 125 us 11: Reserved	R/W	R	3.10.1 *1), *2)
1-0	Nothing is placed here. These should be fixed at "0".				

«Notes»

***1)** With Full-Speed operation, (when "HSE=0" has been set, or the **RHST** bit indicates "RHST=0" as the result of the reset handshake), "SOFM=10" should not be set.

***2)** This bit should be set after a reset handshake has been completed, or when recovering from the low-power sleep state, and should not be changed during subsequent USB communication.

2.9 Interrupt statuses

◆ Interrupt status register 0[**INTSTS0**] <Address: 40H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ			VALID	CTSQ		
0	0	0	0	0	0	0	0	?	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0
-	-	-	1	-	-	-	-	-	0	0	1	-	-	-	-
-	-	0	0	0	0	0	0	?	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	VBINT VBUS interrupt status	0: VBUS interrupts not issued 1: VBUS interrupts issued	R/W	W	3.2.8 *3)
14	RESM Resume interrupt status	0: Resume interrupts not issued 1: Resume interrupts issued	R/W	W	3.2.9 *3)
13	SOFR Frame number refresh interrupt status	0: SOF interrupts not issued 1: SOF interrupts issued	R/W(0)	W	3.2.7 *3)
12	DVST Device state transition interrupt status	0: Device state transition interrupts not issued 1: Device state transition interrupts issued	R/W(0)	W	3.2.5 *3)
11	CTRT Control transfer stage transition interrupt status	0: Control transfer stage transition interrupts not issued 1: Control transfer stage transition interrupts issued	R/W(0)	W	3.2.6 *3)
10	BEMP Buffer Empty interrupt status	0: BEMP interrupts not issued 1: BEMP interrupts issued	R	W	3.2.4 *1)*2)
9	NRDY Buffer Not Ready interrupt status	0: NRDY interrupts not issued 1: NRDY interrupts issued	R	W	3.2.3 *1)
8	BRDY Buffer Ready interrupt status	0: BRDY interrupts not issued 1: BRDY interrupts issued	R	W	3.2.2 *1)
7	VBSTS VBUS input status	0: VBUS pin is “L” level 1: VBUS pin is “H” level	R	W	3.2.8 *2)
6-4	DVSQ Device state	000: Powered state 001: Default state 010: Address state 011: Configured state 1xx: Suspended state	R	W	3.2.5
3	VALID Setup packet reception	0: Not detected 1: Setup packet reception	R/W(0)	W	3.6.1
2-0	CTSQ Control transfer stage	000: Idle or setup stage 001: Control read data stage 010: Control read status stage 011: Control write data stage 100: Control write status stage 101: Control write (NoData) status stage 110: Control transfer sequence error 111: Reserved	R	W	3.2.6

«Note»

*1) The **BEMP**, **BRDY** and **NRDY** bits are cleared when all of the factors for each pipe on correcponding registers have been eliminated, i.e. **BEMPSTS**, **BRDYSTS** and **NRDYSTS**.

*2) The VBUS input status based on the **VBSTS** bit requires that chattering be eliminated using software.

*3) If multiple factors are being generated among the **VBINT**, **RESM**, **SOFR**, **DVST**, and **CTRT** bits, an access cycle of at least 100 ns is required in order to clear the bits in succession, rather than simultaneously.

◆ BRDY interrupt status register [**BRDYSTS**] <Address: 46H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIPEBRDY							
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	-	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-8	Nothing is placed here. These should be fixed at "0".				
7-0	PIPEBRDY BRDY interrupt status for theeach pipe	0: Interrupts are not issued. 1: Interrupts are issued.	R/W(0)	W(1)	3.2.2 *4)

«Note»

***4)** The bit numbers correspond to the pipe numbers. Also, if factors are being generated for more than one pipe, an access cycle of at least 100 ns is required in order to clear the bits in succession, rather than simultaneously.

◆ NRDY interrupt status register [**NRDYSTS**] <Address: 48H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIPENRDY							
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	-	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-8	Nothing is placed here. These should be fixed at "0".				
7-0	PIPENRDY NRDY interrupt for the each pipe	0: Interrupts are not issued. 1: Interrupts are issued.	R/W(0)	W(1)	3.2.3 *5)

«Note»

***5)** The bit numbers correspond to the pipe numbers. Also, if factors are being generated for more than one pipe, an access cycle of at least 100 ns is required in order to clear the bits in succession, rather than simultaneously.

◆ BEMP interrupt status register [**BEMPSTS**] <Address: 4AH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								PIPEBEMP							
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0
?	?	?	?	?	?	?	?	-	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-8	Nothing is placed here. These should be fixed at "0".				
7-0	PIPEBEMP BEMP interrupt for the each pipe	0: Interrupts are not issued. 1: Interrupts are issued.	R/W(0)	W(1)	3.2.4 *6)

«Note»

***6)** The bit numbers correspond to the pipe numbers. Also, if factors are being generated for more than one pipe, an access cycle of at least 100 ns is required in order to clear the bits in succession, rather than simultaneously.

2.10 Frame number register

◆ Frame number register [**FRMNUM**] <Address: 4CH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVRN	CRCE			SOFRM	FRNM										
0	0	?	?	0	0	0	0	0	0	0	0	0	0	0	0
0	0	?	?	0	0	0	0	0	0	0	0	0	0	0	0
-	-	?	?	-	-	-	-	-	-	-	-	-	-	-	-
0	0	?	?	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	OVRN Overrun / Underrun	0: No error 1: Error issued	R/W(0)	W	2.10.1
14	CRCE Reception data error	0: No error 1: Error issued	R/W(0)	W	2.10.1
13-12	Nothing is placed here. These should be fixed at “0”.				
11	SOFRM Frame number update interrupt output mode	0: Interrupt asserted upon SOF reception and timer interpolation. 1: Interrupt asserted if SOF is damaged or missing.	R/W	R	2.10.2 3.2.7 *1)
10-0	FRNM Frame number	The frame number can be confirmed.	R	W	2.10.2

«Note»

***1)** Frame number update interrupts are not issued for uSOF packet detection other than “UFRNM=0”.

◆ micro frame number register [**UFRMNUM**] <Address: 4EH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													UFRNM		
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	-	-	-
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-3	Nothing is placed here. These should be fixed at "0".				
2-0	UFRNM micro frame	The micro frame number can be confirmed.	R	W	2.10.2 3.10 *2)

«Note»

***2)** When using Full-Speed operation, “000” is normally read with this bit.

2.10.1 Isochronous errors

With this controller, data transfer errors that occur in isochronous transfers can be confirmed using the **OVRN** bit and the **CRCE** bit of the **FRMNUM** register. In isochronous transfers, error notification by the **NRDY** interrupt can be differentiated using the **OVRN** bit and the **CRCE** bit between data buffer errors and packet errors.

Table 2.7and Table 2.8 show the conditions under which the **OVRN** bit and **CRCE** bit areset to "1".

Table 2.7 Error information when an NRDY interrupt is issued in an isochronous out transfer

Bit status	Issued when:	Issue conditions	Detected error	Operation
"OVRN=1"	Data packet is received	A new data packet is received before reading of buffer memory is completed	Reception data buffer overrun	The new data packet is thrown out
"CRCE=1"	Data packet is received	A CRC error, or, a bit stuffing error is detected	Received packet error	The new data packet is thrown out

Table 2.8 Error information when an NRDY interrupt is issued in an isochronous in transfer

Bit status	Issued when:	Issue conditions	Detected error	Operation
"OVRN=1"	IN token is received	An in-token is received before writing to buffer memory is completed	Transmission data buffer underrun	Zero-Length packet transmission
"CRCE=1"	Not issued			

2.10.2 SOF interrupts and frame numbers

The **SOFR** interrupt operation mode should be selected using the **SOFRM** bit of the **FRMNUM** register. Also, the current frame number can be confirmed using the **FRNM** bit of the **FRMNUM** register and the **UFRNM** bit of the **UFRNUM** register.

With this controller, the frame numbers are refreshed at the timing at which SOF packets are received. If the controller is unable to detect an SOF packet because the packet has beencorrupted, or for another reason, the **FRNM** value is retained until a new SOF packet is received.

At that point, the **FRNM** bit based on the SOF interpolation timer is not refreshed. Also, the **UFRNM** bit is incremented in response to a uSOF packet being received.

2.11 USB address((low-power recovery)

◆ USB address/low-power status recovery register [**RECOVER**] <Address: 50H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					STSRECOV				USBADDR						
?	?	?	?	?	0	0	0	?	0	0	0	0	0	0	0
?	?	?	?	?	-	-	-	?	0	0	0	0	0	0	0
?	?	?	?	?	-	-	-	?	0	0	0	0	0	0	0
?	?	?	?	?	0	0	0	?	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-11	Nothing is placed here. These should be fixed at “0”.				
10-8	STSRECOV Status recovery	Status recovery after the low-power sleep state 000: reserved 001: Full-Speed Default state 010: Full-Speed Address state 011: Full-Speed Configured state 100: reserved 101: Hi-Speed Default state 110: Hi-Speed Address state 111: Hi-Speed Configured state	R/W	R	3.1.5 *1)
7	Nothing is placed here. This should be fixed at "0".				
6-0	USBADDR USB address	USB address confirmation and recovery	R/W	R/W	3.1.5 *1)

«Note»

*1) When a recovery has been made from the low-power sleep state to the normal mode, the communication speed, device state and USB address have to be returned to the values backed up bysoftware. “STSRECOV=x00” should not be set.

2.12 USB request register

The USB request register is used to store setup requests for control transfers. The values of USB requests that have been received are stored here.

◆ USB request type register [**USBREQ**] <Address: 54H>

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

Bit	Name	Function	S/W	H/W	Note
15-8	bRequest Request	The USB request bRequest value is stored here.	R	W	3.6.1
7-0	bmRequestType Request type	The USB request bmRequestType is stored here.	R	W	3.6.1

«Note»

None in particular

◆ USB request value register [**USBVAL**] <Address: 56H>

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

Bit	Name	Function	S/W	H/W	Note
15-0	wValue Value	The USB request wValue value is stored here.	R	W	3.6.1

«Note»

None in particular

◆ USB request index register [**USBINDX**] <Address: 58H>

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

Bit	Name	Function	S/W	H/W	Note
15-0	wIndex Index	The USB request wIndex value is stored here.	R	W	3.6.1

«Note»

None in particular

◆ USB request length register [**USBLENG**] <Address: 5AH>

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

Bit	Name	Function	S/W	H/W	Note
15-0	wLength Length	The USB request wLength value is stored here.	R	W	3.6.1

«Note»

None in particular

2.13 DCP configuration

When data communication is being carried out using control transfers, the default control pipe should be used.

◆ DCP configuration register [**DCPCFG**] <Address: 5CH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							CNTMD								
?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	-	?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	?

Bit	Name	Function	S/W	H/W	Note
15-9	Nothing is placed here. These should be fixed at “0”.				
8	CNTMD Continuous transfer mode	0: Non-continuous transfer mode 1: Continuous transfer mode	R/W	R	3.4.1 *1)
7-0	Nothing is placed here. These should be fixed at “0”.				

«Note»

*1) Because the DCP buffer memory is used for both control read transfers and control write transfers, the **CNTMD** bit will serve as the bit common to both, regardless of the transfer direction.

◆ DCP maximum packet size register [**DCPMAXP**] <Address: 5EH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
									MXPS						
?	?	?	?	?	?	?	?	?	1	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	1	0	0	0	0	0	0
?	?	?	?	?	?	?	?	?	-	-	-	-	-	-	-
?	?	?	?	?	?	?	?	?	1	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-7	Nothing is placed here. These should be fixed at “0”.				
6-0	MXPS Maximum packet size	This specifies the maximum packet size for the DCP.	R/W	R	3.3.3 *2)

«Note»

*2) This should not be set to anything other than the USB specification. Also, because b2-b0 are fixed at “0”, writing to these is invalid.

◆ DCP control register [**DCPCTR**] <Address: 60H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSTS							SQCLR	SQSET	SQMON				CCPL	PID	
0	?	?	?	?	?	?	0	0	1	?	?	?	0	0	0
0	?	?	?	?	?	?	0	0	1	?	?	?	0	0	0
-	?	?	?	?	?	?	-	-	-	?	?	?	0	0	0
0	?	?	?	?	?	?	0	0	1	?	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	BSTS Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R	W	3.4.1 *3)
14-9	Nothing is placed here. These should be fixed at “0”.				
8	SQCLR Toggle Bit Clear	0: Invalid 1: Specifies DATA0	R(0)/ W(1)	R	3.3.6 *3), *5), *6)
7	SQSET Toggle Bit Set	0: Invalid 1: Specifies DATA1	R(0)/ W(1)	R	3.3.6 *3), *5), *6)
6	SQMON Toggle Bit Confirm	0: DATA0 1: DATA1	R	W	3.3.6 *3), *5), *7)
5-3	Nothing is placed here. These should be fixed at “0”.				
2	CCPL Control Transfer End enabled	0: Invalid 1: The control transfer is ended.	R(0)/ W(1)	R/W(0)	3.6 *4)
1-0	PID Response PID	00: NAK response 01: BUF response (in keeping with the buffer state) 10: STALL response 11: STALL response	R/W	R/W	3.3.4 *8)

«Notes»

***3)** The direction of buffer access, writing or reading, is depend on setting of **ISEL** bit. For detailed information, please refer to Chapter 3.

***4)** The **CCPL** bit is cleared to “0” right after the SETUP token has been received.

***5)** If the **SQSET** bits and the **SQCLR** bits of the **DCPCTR** register and the **PIPExCTR** registers are being changed in succession (the PID sequence toggle bits of multiple pipes are being changed in succession), an access cycle of at least 200 ns is required.

***6)** The **SQCLR** bit and **SQSET** bit should not both be set to “1” at the same time. Before operating either bit, “PID=NAK” should be set.

***7)** The **SQMON** bit is initialized to “1” by the controller right after the SETUP token of the control transfer has been received.

***8)** The **PID** bit is cleared to “00” right after the SETUP token has been received.

2.14 Pipe configuration register

The PIPE1-7 settings should be set using the **PIPESEL**, **PIPECFG**, **PIPEBUF**, **PIPEMAXP**, **PIPEPERI**, and **PIPExCTR** registers.

After selecting the pipe using the **PIPESEL** register, set the functions of the pipe using the **PIPECFG**, **PIPEBUF**, **PIPEMAXP**, and **PIPEPERI** registers. The **PIPExCTR** register can be set separately from the pipe selection specified with the **PIPESEL** register, with no relation between them.

For an H/W reset, S/W reset, USB bus reset, and when shifting to the low-power sleep state, the pertinent bits for not only the selected pipe, but all of the pipes are initialized.

◆ Pipe window selection register [**PIPESEL**] <Address: 64H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
													PIPESEL		
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	?	?	?	?	?	?	?	?	?	?	-	-	-
?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-3	Nothing is placed here. These should be fixed at “0”.				
2-0	PIPESEL Pipe window selection	000: Not selected 001: Pipe 1 010: Pipe 2 011: Pipe 3 100: Pipe 4 101: Pipe 5 110: Pipe 6 111: Pipe 7	R/W	R	3.3 *1)

«Note»

***1)** When"PIPESEL=000"is set, “0” is read from all of the bits of the five related registers noted above.

◆ Pipe configuration register [**PIPECFG**] <Address: 66H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE					BFRE	DBLB	CNTMD	SHTNAK			DIR	EPNUM			
0	0	?	?	?	0	0	0	0	?	?	0	0	0	0	0
0	0	?	?	?	0	0	0	0	?	?	0	0	0	0	0
0	0	?	?	?	-	-	-	-	?	?	-	-	-	-	-
0	0	?	?	?	0	0	0	0	?	?	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-14	TYPE Transfer type	00: Pipe use disabled 01: Bulk transfer 10: Interrupt transfer 11: Isochronous transfer	R/W	R	3.3.1 *2)
13-11	Nothing is placed here. These should be fixed at “0”.				
10	BFRE BRDY interrupt operation specified	0: BRDY interrupt upon sending or receiving of data 1: BRDY interrupt upon reading of data	R/W	R	3.2.2 *3)
9	DBLB Double buffer mode	0: Single buffer 1: Double buffer	R/W	R	3.4.1 *4)
8	CNTMD Continuous transfer mode	0: Non-continuous transfer mode 1: Continuous transfer mode	R/W	R	3.4.1 *5)
7	SHTNAK Pipe disabled at end of transfer	0: Pipe continued at end of transfer 1: Pipe disabled at end of transfer	R/W	R	3.3.7
6-5	Nothing is placed here. These should be fixed at “0”.				
4	DIR Transfer direction	0: Receiving (OUT transfer) 1: Sending (IN transfer)	R/W	R	3.4.1
3-0	EPNUM End point number	Specifies the end point number for the pertinent pipe	R/W	R	3.3.2

«Notes»

***2)** The valid value for the **TYPE** bit depends on the setting for the **PIPESEL** bit of the **PIPESEL** register. For detailed information, please refer to 3.3.1.

***3)** If "BFRE=1” is set, **BRDY** interrupts are not generated when the buffer is set to the data writing direction.

***4)** The **DBLB** bit is valid when PIPE1-5 are selected.
The procedure to change the **DBLB** bit for a PIPE is as following;
(a) Single buffer to double buffer (“DBLB=0” to “DBLB=”1”);
Set the **PID** bit to “NAK” for the pertinent pipe → “ACLRM=1” → (wait at least 100ns) → “ACLRM=0” → “DBLB=”1” → Set the **PID** bit to “BUF” for the pipe
(b) Double buffer to singlee buffer (“DBLB=1” to “DBLB=”0”);
Set the **PID** bit to “NAK” for the pertinent pipe → “DBLB=”0” → “ACLRM=1” → (wait at least 100ns) → “ACLRM=0” → Set the **PID** bit to “BUF” for the pipe

***5)** The **CNTMD** bit is valid when bulk transfer (“TYPE=01”) is selected using PIPE1-5. “CNTMD=1” should not be set when isochronous transfer has been selected (“TYPE=11”).
The **CNTMD** bit should not be set “1” for PIPE6-7.

◆ Pipe buffer setting register [**PIPEBUF**] <Address: 68H>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BUFSIZE							BUFNMB							
?	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0
?	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0
?	-	-	-	-	-	?	?	-	-	-	-	-	-	-	-
?	0	0	0	0	0	?	?	0	0	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15	Nothing is placed here. This should be fixed at “0”.				
14-10	BUFSIZE Buffer Size	Specifies the size of the pertinent pipe. (from 0: 64 bytes to 0x1F: 2KB)	R/W	R	3.4 *6)
9-8	Nothing is placed here. These should be fixed at “0”.				
7-0	BUFNMB Buffer Number	Specifies the buffer number for the pertinent pipe. (From 0x4 to 0x4F)	R/W	R	3.4 *7)

«Notes»

*6) The valid value for the **BUFSIZE** bit depends on the selected PIPE by the **PIPESEL** bit of the **PIPESEL** register.
When using PIPE1-5, any value from 0 to 0x1F is valid.
When using PIPE6-7, writing to this bit is invalid.

*7) The **BUFNMB** bit can be set to match the user system when PIPE1-5 are selected.
"BUFNMB=0-3" is used exclusively for the DCP. "BUFNMB=4-5" is allocated to PIPE6-7.
When using PIPE6, writing to this bit is invalid, and "BUFNMB=4” is always used for reading.
When using PIPE7, writing to this bit is invalid, and “BUFNMB=5” is always used for reading.

◆ Pipe maximum packet size register [**PIPEMAXP**] <Address: 6AH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
					MXPS										
?	?	?	?	?	0	0	0	0	0(1) *9)	0	0	0	0	0	0
?	?	?	?	?	0	0	0	0	0(1)	0	0	0	0	0	0
?	?	?	?	?	-	-	-	-	-	-	-	-	-	-	-
?	?	?	?	?	0	0	0	0	0(1)	0	0	0	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-11	Nothing is placed here. These should be fixed at “0”.				
10-0	MXPS Maximum Packet Size	Specifies the maximum packet size for the pertinent pipe.	R/W	R	3.3.3 *8), *9)

«Note»

***8)** The **MXPS** bit should be set to"0x00”, or the setting defined by the USB specification should be used.

***9)** The default value of **MXPS** bit is "0x0" when "PIPSEL=0", and "0x40" when "PIPESEL>0".

◆ Pipe timing control register [**PIPEPERI**] <Address: 6CH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			IFIS										IITV		
?	?	?	0	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	0	?	?	?	?	?	?	?	?	?	0	0	0
?	?	?	-	?	?	?	?	?	?	?	?	?	-	-	-
?	?	?	0	?	?	?	?	?	?	?	?	?	0	0	0

Bit	Name	Function	S/W	H/W	Note
15-13	Nothing is placed here. These should be fixed at “0”.				
12	IFIS Isochronous IN buffer flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W	R	3.9.5
11-3	Nothing is placed here. These should be fixed at “0”.				
2-0	IITV Interval error detection spacing	Specifies the interval timing as IITV-th power of 2.	R/W	R	3.9 *10)

«Note»

***10)** The **IITV** bit is valid only when isochronous transfer is selected. In other words, it can be set only when PIPE1-2 are selected.
For OUT-direction: An interval error occurs upon a **NRDY** interrupt caused by a token not having been issued.
For IN-direction: When the controller doesn't receive IN-token until the time indicated by **IITV** bit, it detects an interval error and flushs the buffer.

- PIPE1 control register [**PIPE1CTR**] <Address: 70H>
- PIPE2 control register [**PIPE2CTR**] <Address: 72H>
- PIPE3 control register [**PIPE3CTR**] <Address: 74H>
- PIPE4 control register [**PIPE4CTR**] <Address: 76H>
- PIPE5 control register [**PIPE5CTR**] <Address: 78H>
- PIPE6 control register [**PIPE6CTR**] <Address: 7AH>
- PIPE7 control register [**PIPE7CTR**] <Address: 7CH>

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BSTS	INBUFM					ACLRM	SQCLR	SQSET	SQMON					PID	
0	0	?	?	?	?	0	0	0	0	?	?	?	?	0	0
0	0	?	?	?	?	0	0	0	0	?	?	?	?	0	0
-	-	?	?	?	?	-	-	-	-	?	?	?	?	0	0
0	0	?	?	?	?	0	0	0	0	?	?	?	?	0	0

Bit	Name	Function	S/W	H/W	Note
15	BSTS Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R	W	3.4.1 *11)
14	INBUFM IN buffer monitor	0: No data in IN-direction buffer 1: There is sending data in IN-direction buffer	R	W	*12), *13)
13-10	Nothing is placed here. These should be fixed at “0”.				
9	ACLRM Auto Buffer Clear mode	0: Disabled 1: Enabled (all buffers are initialized)	R(0)/W(1)	R/W(0)	3.4.1 *14)
8	SQCLR Toggle Bit Clear	0: Invalid 1: Specifies DATA0	R(0)/W(1)	R	3.3.6 *15), *16)
7	SQSET Toggle Bit Set	0: Invalid 1: Specifies DATA1	R(0)/W(1)	R	3.3.6 *15), *16)
6	SQMON Toggle Bit Confirm	0: DATA0 1: DATA1	R	W	3.3.6 *15)
5-2	Nothing is placed here. These should be fixed at “0”.				
1-0	PID Response PID	00: NAK response 01: BUF response (in keeping with the buffer state) 10: STALL response 11: STALL response	R/W	R/W	3.3.4 *16), *17)

«Notes»

***11)** The direction of buffer access, writing or reading, is depend on setting of the **DIR** bit of the **PIPECFG** register. For detailed information, please refer to Chapter 3.

***12)** The **INBUFM** bit is valid when softwware sets the **DIR** bit to Sending-direction.

***13)** The **INBUFM** bit is valid for PIPE1-5.

***14)** Software should not set “ACLRM=1” for the PIPE whith selected by the **PIPESEL** bit of the **PIPESEL** register.

***15)** If the **SQCLR** bits and the **SQSET** bits of the **DCPCTR** register and the PIPExCTR registers are being used to change the data PID sequence toggle bit for several pipes in succession, an access cycle of at least 200 ns is required.

***16)** The **SQCLR** bit and **SQSET** bit should not both be set to “1” at the same time. Before operating either bit, “PID=NAK” should be set.

***17)** If an excessive packet size error is detected, “PID=STALL” is set by the controller.

3 Description of Operation

3.1 System control and oscillation control

This chapter describes the register operations that are necessary to the default settings of the controller, and the registers necessary for power consumption control.

3.1.1 Resets

Table 3.1 shows a table of controller resets. For information on the initialized states of the registers following the various reset operations, please refer to Chapter 2, Registers.

Table 3.1 Types of resets

Name	Operation
H/W reset	"L" level input from the **RST_N** pin
S/W reset	Operation using the **USBE** bit of the **SYSCFG** register
USB bus reset	Automatically detected by the controller from the D+ and D- lines

3.1.2 Bus interface settings

Table 3.2 shows the bus interface settings for the controller.

Table 3.2 Bus interface settings

Register name	Bit name	Setting contents
PINCFG	LDRV	Control setting for the drive current
PINCFG	BIGEND	Byte Endian setting for the CPU being connected
DMAxCFG	DREQA	Active setting for the DREQ_N pin
DMAxCFG	DACKA	Active setting for the DACK_N pin
DMAxCFG	DENDA	Active setting for the DEND_N pin
DMAxCFG	OBUS	OBUS mode setting
INTENB1	INTL	Output sensing setting for the INT_N pin

3.1.3 Enabling Hi-Speed operation

With this controller, either Hi-Speed operation or Full-Speed operation can be selected as the USB communication speed (communication bit rate), using software. To enable Hi-Speed operation for the controller, set the **HSE** bit of the **SYSCFG** register to "1". Changing the **HSE** bit should be done with the internal clock stopped ("SCKE=0").

If Hi-Speed operation has been enabled, the controller executes the reset handshake protocol, and the USB communication speed is set automatically. The results of the reset handshake can be confirmed using the **RHST** bit of the **DVSTCTR** register.

If Hi-Speed operation has been disabled, the controller will use Full-Speed operation.

3.1.4 USB data bus resistor control

Figure 3.1 shows a diagram of the connections between the controller and the USB connectors.

The controller has a built-in pull-up resistor for the D+ signal. "1" should be set for the **DPRPU** bit of the **SYSCFG** register, then the D+ line is pulled up. The pull-up power supply is **AFE33V**.

Also, the controller has a built-in terminal resistor for use when the D+ and D- signals are operating at Hi-Speed, and a built-in output resistor for Full-Speed operation. The controller automatically switches the built-in resistor after connection with the PC, by means of reset handshake, suspended state and resume detection. If a disconnection from the PC is detected, the H/W should be initialized by means of an S/W reset (USBE=0).

If "0" is set for the **DPRPU** bit of the **SYSCFG** register, the pull-up resistor (or the terminal resistor) of the USB data line is disabled, making it possible to notify the host controller of the device disconnection.

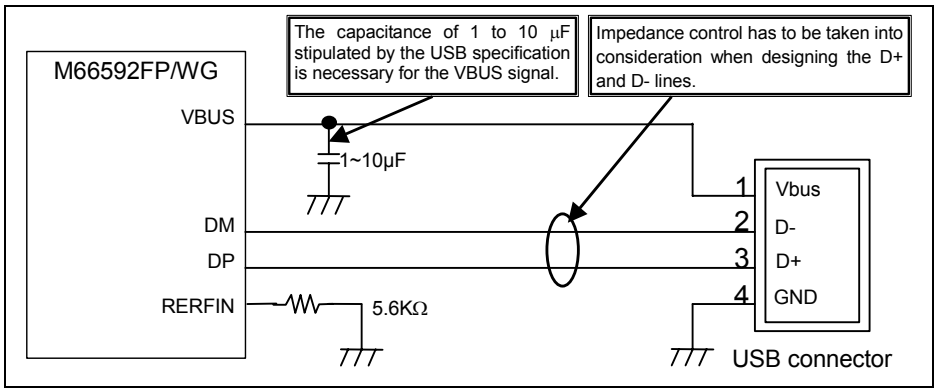


Figure 3.1 UBS connector connections

3.1.5 Clock supply control

Figure 3.2 shows a block diagram of the controller clock control. Frequency of the input clock for the XIN pin should be selected using the **XTAL** bit of the **SYSCFG** register, while the oscillation buffer is enabled using the **XCKE** bit and the clock supply is controlled using the **RCKE**, **PLLC**, and **SCKE** bits. For information on the register control timing, please refer to 3.1.7, State transition timing.

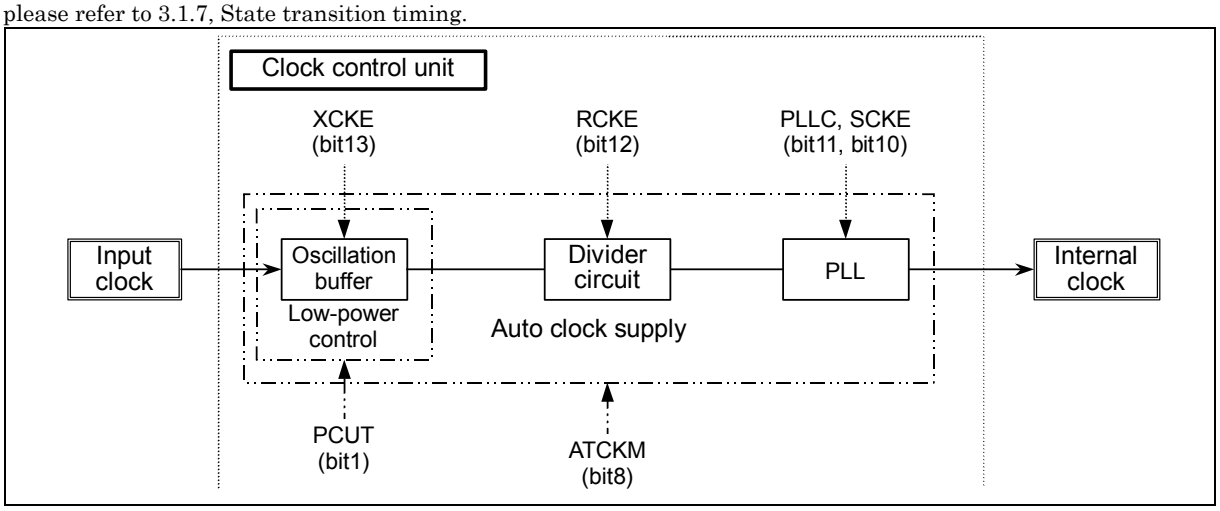


Figure 3.2 Clock control block

3.1.6 Low power consumption control

3.1.6.1 Overview of low-power sleep state

In order to reduce power consumption, the controller is equipped with a function for setting a low-power sleep state.

Controlling the clock and the low-power sleep state enables reduced power consumption when communication is not being carried out, such as in the suspended state or disconnected state. In order to coordinate the relationship between the controller clock supply being enabled and disabled in low-power sleep state, the values shown in Table 3.3 indicate the correspondence between the controller state and the value of the **SYSCFG** register, while Figure 3.3 shows the transitions in the controller state.

For the timing at which the transitions between the various states take place, and the register control timing, please refer to 3.1.7.

Table 3.3 Correspondence between the controller state and SYSCFG register value

Controller state	Values of the various SYSCFG register bits	Explanation
H/W reset	XTAL=0, XCKE=0, RCKE=0, PLLC=0, SCKE=0, ATCKM=0, HSE=0, DPRPU=0, PCUT=0, USBE=0	
Normal operating state	XTAL=xx *1), XCKE=1, RCKE=1, PLLC=x*1), SCKE=1, ATCKM=x*1), HSE=x*1), DPRPU=x*1), PCUT=0, USBE=1	In this state, the clock is supplied to the controller, and USB communication is enabled.
Low-power sleep state	XTAL=xx *1), XCKE=0, RCKE=0, PLLC=0, SCKE=0, ATCKM=x *1), HSE= x *1), DPRPU=x *1), PCUT=1, USBE=1	In this state, USB communication is not carried out, such as when communication is suspended or a cable is disconnected.

***1)** x indicates that the value is set by the user is retained.

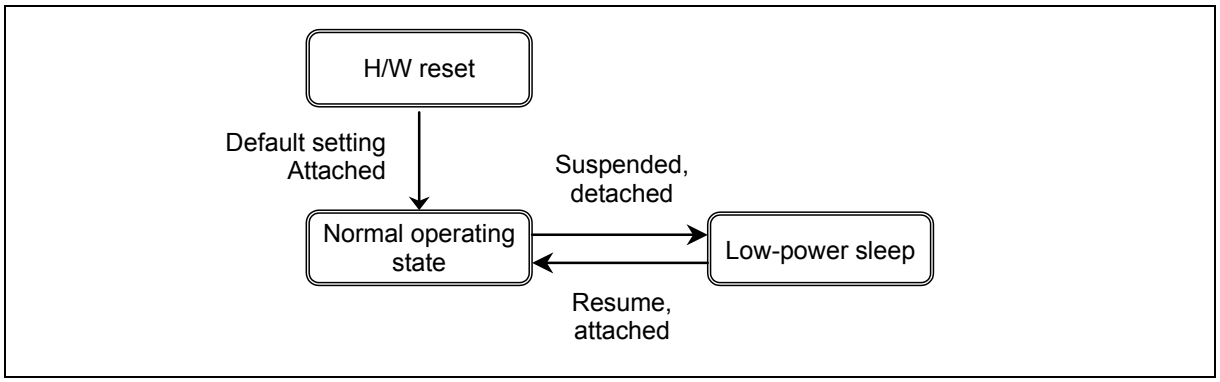


Figure 3.3 Controller state transitions (using low-power sleep state)

3.1.6.2 Overvew of Clock stop state

This controller is equipped with the setting function of the low power consumption state by clock stop as well as M66291, M66591, and M66592. Software is transplantable for this controller from M66291, M66591, and M66592 by fewer changes.

In the states where the controller is not communicating such as suspend and a not connect state, low power consumption by clock stop is realized. Correspondence of the state of the controller and the value of a **SYSCFG** register is shown in Table 3.4. The transitions in the controller state is shown inFigure 3.4**.**

For the timing at which the transitions between the various states take place, and the register control timing, please refer to 3.1.7.

Moreover, when you use the low power consumption state by clock stop, please use the auto clock supply function ("ATCKM=1").

Table 3.4 Correspondence between the controller state and SYSCFG register value

Controller state	Values of the various SYSCFG register bits	Explanation
H/W reset	XTAL=0, XCKE=0, RCKE=0, PLLC=0, SCKE=0, ATKCM=0, HSE=0, DPRPU=0, PCUT=0, USBE=0	
Normal operating state	XTAL=xx *1), XCKE=1, RCKE=1, PLLC=x *1), SCKE=1, ATKCM=x *1), HSE=x *1), DPRPU=x *1), PCUT=0, USBE=1	In this state, the clock is supplied to the controller, and USB communication is enabled.
Clock stop state	XTAL=xx *1), XCKE=0, RCKE=0, PLLC=0, SCKE=0, ATKCM=1, HSE= x *1), DPRPU=x *1), PCUT=0, USBE=1	In this state, USB communication is not carried out, such as when communication is suspended or a cable is disconnected.

***1)** x indicates that the value is set by the user is retained.

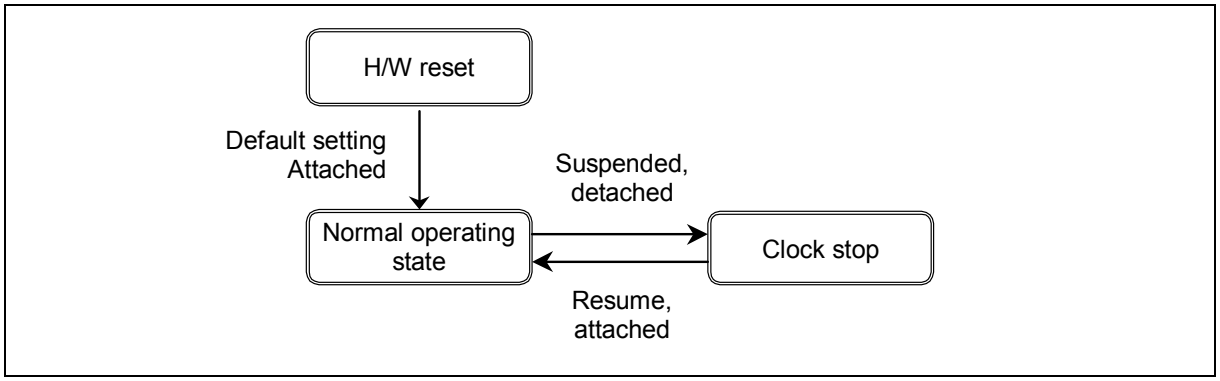


Figure 3.4 Controller state transitions (using clock stop state)

3.1.6.3 Low-power sleep state

The low-power sleep state is set by setting “1” for the **PCUT** bit of the **SYSCFG** register. For information on the sequence in which settings are entered for the low-power sleep state, please refer to 3.1.7.2, and for information on register control timing, please refer to the timing diagram noted later (Figure 3.6 Transition control timing).

In the low-power sleep state, of the registers set bysoftware, only registers other than those noted below are initialized. After returning to the normal operating state, the settings must be re-entered usingsoftware. Table 3.5 shows the registers that are not initialized when the controller is in the low-power sleep state.

Table 3.5 Registers that are not initialized in the low-power sleep state

Register	Bit	Description
SYSCFG	XTAL	This is retained as system information.
	ATCKM	This is retained as system information.
	HSE	This is retained as system information.
	DPRPU	This is retained as system information.
	USBE	This is retained as system information.
PINCFG	LDRV	The state of the output pins drive current settings is retained.
DMAxCFG	DREQA	The polarity of the DREQ0_N pin and the DREQ1_N pin are retained.
INTENB0/ INTSTS0	VBSE / VBINT	When "VBSE=1", if there was any change to the VBUS signal in the low-power sleep state, the INT_N pin is asserted and notification is made to the CPU.
	RSME / RESM	When "RSME=1", if there was any change to the USB data bus in the low-power sleep state, the INT_N pin is asserted and notification is made to the CPU.

3.1.6.4 Recovering from the low-power sleep state

If any of the events noted below occurs from the low-power sleep state, the controller notifies the CPU through the **INT_N** pin. The interrupt factor related to those events should be enabled, before sofware sets the controller to the low-power sleep state.

(1) VBUS detection : If a change in the **VBUS** pin was detected in the low-power sleep state
(2) RESUME detection : If a change in the state of the USB bus (J-State to K-State or SE0) was detected when the state shifted to the low-power sleep state during the suspended state.

When the **PCSE** bit of **INTENB1** register is set to "0", the low-power sleep state is also canceled by the operations noted below, and the controller returns to the normal operating state.

(1) Dummy writing to the 0x7E address of the controller (no actual writing is done to this address).

When the system has returned from the low-power sleep state to the normal state, some of the controller registers need to be returned to the values in effect prior to the transition to the low-power sleep state. Of the registers for which the settings have to be returned, special registers are available that are used for re-setting data in the read-only registers.

Table 3.6 shows the re-settings for the read-only registers for which the settings have to be returned.

Table 3.6 Re-settings for read-only registers for which settings have to be returned

Register	Bit	Method for re-setting registers
DVSTCTR	RHST	Setting the USB communication speed and device state using the **STSRECOV** bit of the **RECOVER** register before shifting to the low-power sleep state recovers the values for the **RHST** bit and **DVSQ** bit.
INTSTS0	DVSQ	
RECOVER	USBADDR	The USB device address prior to the shift to the low-power sleep state is set in the USBADDR bit of the RECOVER register.
PIPExCTR	SQMON	The sequence toggle bits for the various pipes prior to the low-power sleep state are set using the SQSET bit or the SQCLR bit of PIPExCTR. *1)

***1)** The **SQMON** bit of the **DCPCTR** register is initialized when the SETUP stage ends, so it is not necessary to return to the state in effect prior to the normal operating state.

3.1.6.5 Recovering from the clock stop state

If any of the events noted below occurs from the clock stop state, the controller notifies the CPU through the **INT_N** pin. The interrupt factor related to those events should be enabled, before sofware sets the controller to the low-power sleep state.

(1) VBUS detection : If a change in the **VBUS** pin was detected in the clock stop state

(2) RESUME detection : If a change in the state of the USB bus (J-State to K-State or SE0) was detected when the state shifted to the low-power sleep state during the suspended state.

3.1.6.6 Auto clock supply function

This controller is equipped with an auto clock supply function. With the auto clock supply function, the controller automatically implements a series of sequence control operations, from the oscillation stabilization standby timing to the supply of the internal clock, when the system is returning from the low-power sleep state or from the clock stop mode to the normal operating state. This function is enbaled by setting "1" for the **ATCKM** bit of the **SYSCFG** register. For information on specific register control, please refer to3.1.7.3.

3.1.7 State transition timing

3.1.7.1 Starting the internal clock supply (from the H/W reset state to the normal operating state)

Figure 3.5 shows a diagram of the clock supply start control timing of the controller. The transition from the H/W reset state to the normal operating state should be done through operation of the bits at the timing noted below.

(1) Software enables the oscillation buffer. "XCKE=1"
(2) Software waits for oscillation to stabilize. (The oscillation stabilization time varies depending on the oscillator.)
(3) Software enables the reference clock suppliance and the PLL operation "RCKE=1", "PLLC=1"
(4) Software waits for the PLL to lock. (A waiting time of at least 8.3 us is necessary.)
(5) Software enables the internal clock suppliance. "SCKE=1"

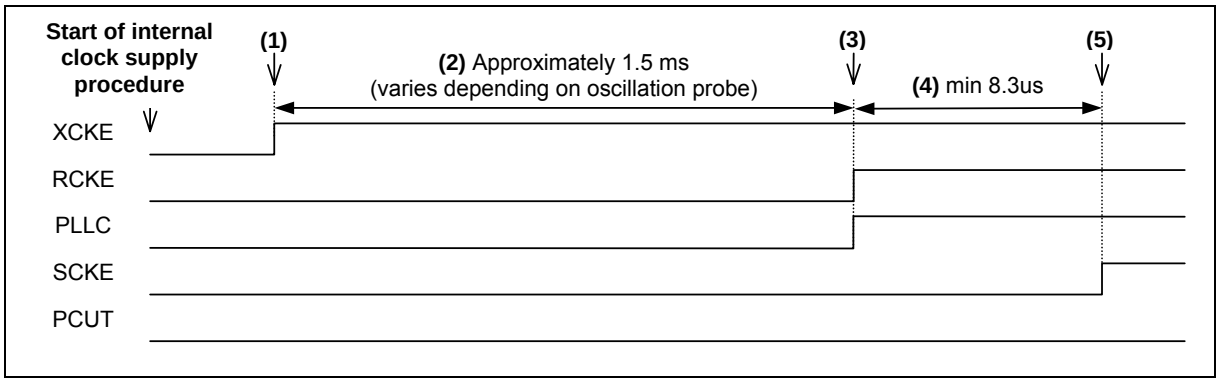


Figure 3.5 Clock supply start control timing

3.1.7.2 Stopping the internal clock supply (from the normal operating state to the low-power sleep state)

Figure 3.6 shows a diagram of the low-power control timing from the normal operating state to the low-power sleep state. The transition from the normal operating state to the low-power sleep state should be done through operation of the bits at the timing noted below.

(1) Software disables the internal clock suppliance. "SCKE=0"
(2) Software waits until the internal clock stops. (A waiting time of at least 300 ns is necessary.)
(3) Software disables the PLL. "PLLC=0"
(4) Software waits for the PLL to stop. (A waiting time of at least 300 ns is necessary.)
(5) Software disables reference clock suppliance. "RCKE=0"
(6) Software waits until the reference clock stops. (A waiting time of at least 300 ns is necessary.)
(7) Software sets the bit for low-power sleep state. "PCUT=1"
(8) The controller disables the oscillation buffer. "XCKE=0(H/W)" *1)

***1)** Software must not set the **XCKE** bit to "0".

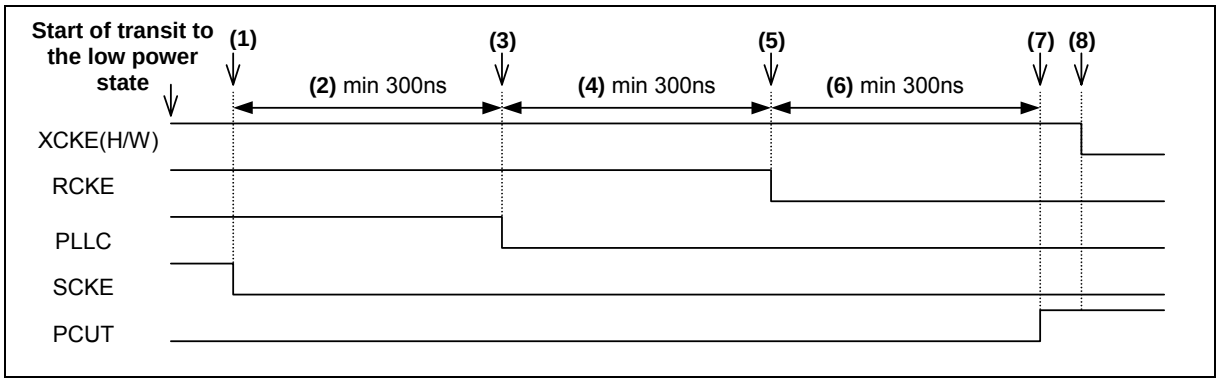


Figure 3.6 Transition control timing to the low-power sleep state

3.1.7.3 Starting the internal clock supply (from the low-power sleep state to the normal operating state)

Figure 3.7 shows a diagram of the timing at which the transition from the low-power sleep state to the normal operating state takes place when the auto clock supply function is enabled ("ATCKM=1"). When the auto clock supply function is enabled, the controller carries out register control, so after an interrupt is generated, the transition to the normal operating state is completed simply by waiting for the amount of time that access is disabled. No operation of the registers using software is necessary.

When operation has been resumed from the suspended state using the USB Bus Reset signal, it is necessary to recover to the normal operating state within 3 ms after the data line change has been detected so that the controller can begin the reset handshake protocol. When the auto clock supply function is enabled, the controller waits automatically for oscillation to stabilize and then carries out clock supply control and handles the reset handshake. Because there is a signal output time of 10 ms for the USB Bus Reset signal and of 20 ms for the Resume signal, software is provided with plenty of allowance to process the recovery to the normal state.

The recovery sequence when the auto clock supply function is enabled is as shown below.

(1) An interrupt is generated to recover from the low-power sleep state, and the **INT_N** pin is asserted. *1)
(Or, the control program writes dummy data to the 0x7E address to cause the controller to recover.)
(2) At the same time, the controller automatically enables the oscillation buffer. "XCKE=1(H/W)"
(3) Software waits until access is enabled. (A waiting time of at least 2.5 ms is necessary.)
(4) The controller automatically enables **RCKE**, **PLLC**, and **SCKE**.
(5) Software resets the registers that have been in the held state before going into the low-power sleep state. *2)

***1)** When the system has recovered from the low-power sleep state to the normal operating state, the USB communication speed and the device state recovery settings have to be set in the **STSRECOV** bit of the **RECOVER** register, and the USB address in the **USBADDR** bit of that register, for recovery to take place.

***2)** If the auto clock supply function has been enabled, however, the recovery settings for the above bits should be entered after the DVSQ bit has been confirmed. This is because, if recovery has been made using the USB Bus Reset signal, there is a possibility that the controller has initialized the device state and the USB address to the default state, in which case rewriting the register values to the waiting state will cause erroneous operation.

The recovery settings are written to the **RECOVER** register using the procedure outlined below.

(a) If "DVSQ=000", recovery is made by a method other than the USB Bus Reset signal. The USB communication speed, device state, and USB address should be returned to the state they were in prior to shifting to the low-power sleep state, by writing to the **RECOVER** register.
(b) If "DVSQ=001", recovery is made using the USB Bus Reset signal. In this case, software should not restore backup value to the **RECOVER** register.

Also, in the low-power sleep state, there are registers that are initialized by the controller. When recovery has been made to the normal operating state, the initialized registers should be reset to match the user system.

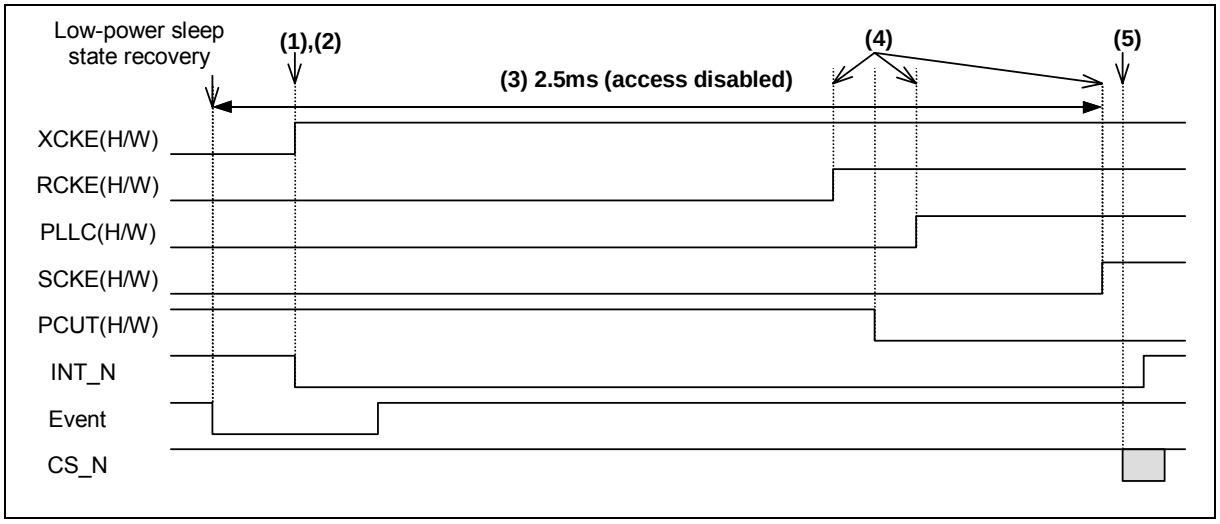


Figure 3.7 Recovery control timing from the low power sleep state with “ATCKM=1”

3.1.7.4 Stopping the internal clock supply (from the normal operating state to the clock stop state)

The timing diagram of the transition from the normal operating state to the clock stop state is shown in Figure 3.8. The transition should be operated according to the following sequence.

(1)	Software disables the internal clock suppliance.	"SCKE=0"
(2)	Software waits until the internal clock stops.	(A waiting time of at least 300 ns is necessary.)
(3)	Software disables the PLL.	"PLLC=0"
(4)	Software waits for the PLL to stop.	(A waiting time of at least 300 ns is necessary.)
(5)	Software disables reference clock suppliance.	"RCKE=0"
(6)	Software waits until the reference clock stops.	(A waiting time of at least 300 ns is necessary.)
(7)	Software disables the oscillation buffer.	"XCKE=0"

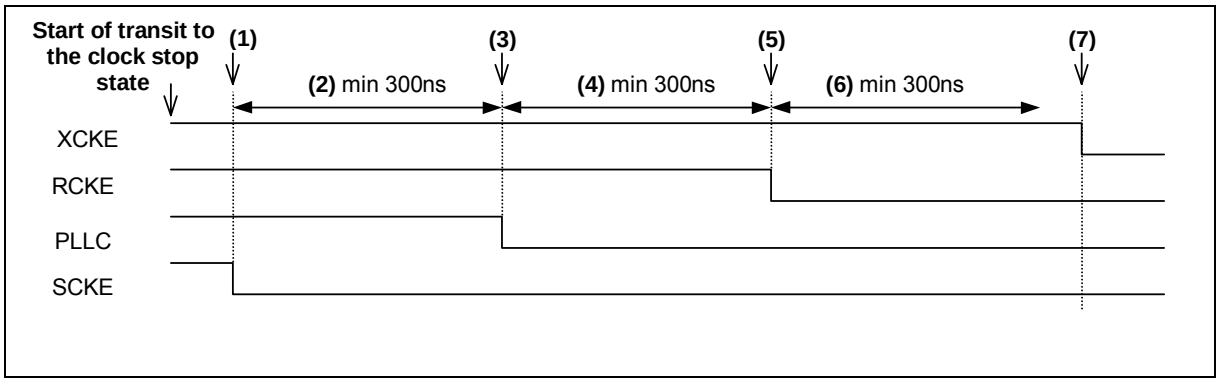


Figure 3.8 Transition control timing to the clock stop state

3.1.7.5 Starting the internal clock supply (from the clock stop state to the normal operating state : with "ATCKM=1")

The timing diagram from the clock stop state to the normal operation state is shown in Figure 3.9. The diagram is in the case of the auto clock supply function is enabled ("ATCKM=1"; recommended setting).

In this case, the controller operates registers when resume signal is detected. The controller changes to the normal operation state by waiting for access prohibition time after resume interrupt is generated. The register operation by software is not required.

If VBUS interrupt is occurs, the softwear need to enable the oscillation, “XCKE=1”

(1)	The controller detects the resume signal or VBUS changes on a USB bus by detachment the cable, and the **INT_N** pin is asserted.	
(2)	When resume signal is received, the controller automatically enables the oscillation buffer. When VBUS change occurs, softwear needs to enable the oscillation buffer.	"XCKE=1(H/W)" "XCKE=1"
(3)	Software waits until access is enabled.	(A waiting time of at least 2.5 ms is necessary.)
(4)	The controller automatically enables **RCKE**, **PLLC**, and **SCKE** during (3).	
(5)	Software operates the process depending on the interrupt factor, resume or attachment.	

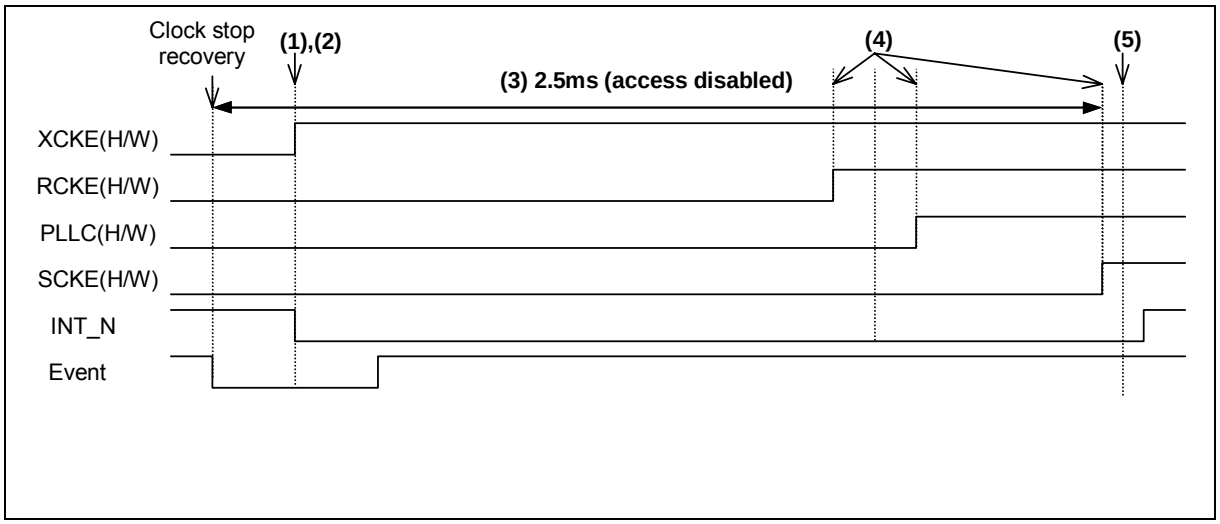


Figure 3.9 Recovery control timing from the clock stop state with "ATCKM=1"

3.1.7.6 Starting the internal clock supply (from the clock stop state to the normal operating state : with "ATCKM=0")

The timing diagram from the clock stop state to the normal operation state is shown in Figure 3.10. The diagram is in the case of the auto clock supply function is disabled ("ATCKM=0"). When the auto clock supply function is disabled, register control is performed by software. Software should operate registers according to the following sequence.

.

(1)	The controller detects the resume on a USB bus or attachment of the USB cable, and the **INT_N** pin is asserted	
(2)	At the same time, the controller automatically enables. the oscillation buffer	"XCKE=1(H/W)"
(3)	Software waits for oscillation to stabilize.*1)	(The oscillation stabilization time varies depending on the oscillator.)
(4)	Software enables the reference clock suppliance and the PLL operation	"RCKE=1", "PLLC=1"
(5)	Software waits for the PLL to lock.	(A waiting time of at least 8.3 us is necessary.)
(6)	Software enables the internal clock suppliance.	"SCKE=1"
(7)	Software performs processing depending on the interrupt factor, resume or attachment.	

***1)** When it returns with a USB bus reset signal from the suspend state, it is necessary to return to the normal operation state less than 3ms and the controller start a reset handshake protocol. For this reason, when an auto clock supply function is disabled, it is necessary to perform a processings to oscillation stability waiting and clock supply by software within 3ms.

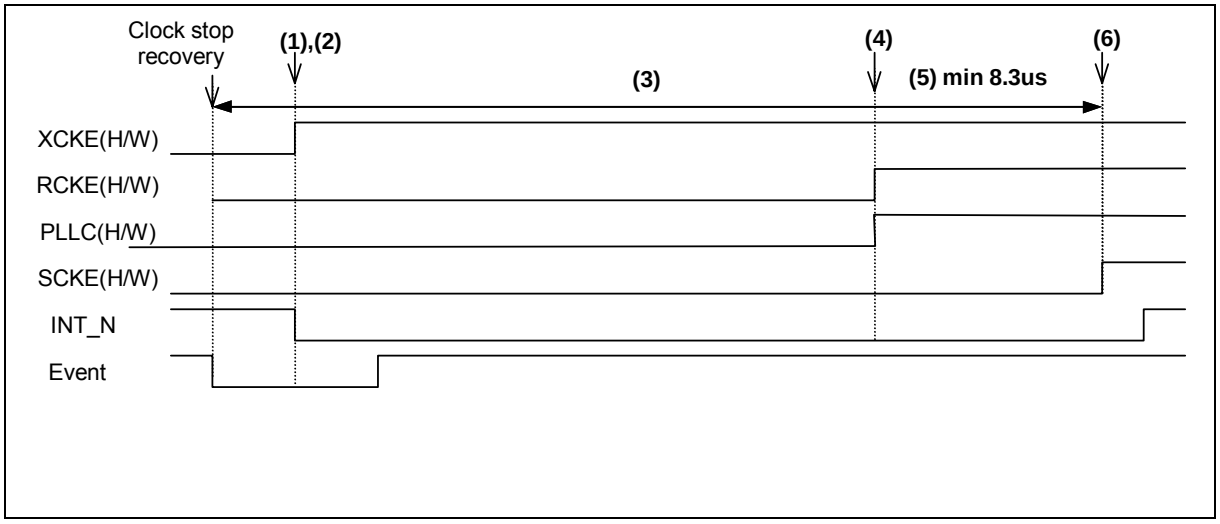


Figure 3.10 Recovery control timing from the clock stop stete with "ATCKM=0"

3.2 Interrupt functions

3.2.1 An overview of interrupt functions

Table 3.7 shows the interrupt functions of the controller.

Table 3.7 Interrupt functions

Bit	Interrupt name	Cause of interrupt	Related status	Note
VBINT	VBUS interrupt	When a change in the state of the VBUS input pin has been detected (change in bothedge, "L"→"H", "H" →"L")	VBSTS	3.2.8
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-State→K-State or J-State→SE0)	-	3.2.9
SOFR	Frame No. Refresh interrupt	When "SOFRM=0" : When an SOF packet with a different frame number has been received When "SOFRM=1" : When the controller detects a corruption of an SOF packet	-	3.2.7
DVST	Device State Transition interrupt	When a device state transition has been detected USB bus reset detected Suspend state detected Set Address request received Set Configuration request received	DVSQ	3.2.5
CTRT	Control Transfer Stage Transition interrupt	When a stage transition has been detected in a control transmission Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed Control transfer sequence error occurred	CTSQ	3.2.6
BEMP	Buffer Empty interrupt	When transmission of all of the data in the buffer memory has been completed When an excessive maximum packet size error has been detected	PIPEBEMP	3.2.4
NRDY	Buffer Not Ready interrupt	When an IN token has been received and there is no data that can be sent to the buffer memory When an OUT token has been received and there is no area in which data can be stored in the buffer memory, so reception is not possible When a CRC error or bit stuffing error occurred in isochronous transfer	PIPENRDY	3.2.3
BRDY	Buffer Ready interrupt	When the buffer is ready (reading or writing is enabled)	PIPEBRDY	3.2.2

Table 3.8 shows the INT_N pin operations of the controller. If multiple interrupt causes have occurred, the method used for **INT_N** pin output can be set using the **INTL** bit of the **INTENB1** register. The operation setting for the **INT_N** pin should be set to match the user system.

Table 3.8 INT_N pin operations

INT_N pin operation / INTL setting	When one interrupt cause occurred	When multiple interrupt causes occurred
Edge sensing (“INTL=0”)	“L” level output until the cause has been eliminated	When one cause is eliminated, the 32 clock time is negated (“H” pulse output) at 48 MHz.
Level sensing (“INTL=1”)	“L” level output until the cause has been eliminated	“L” level is output until all of the causes have been eliminated.

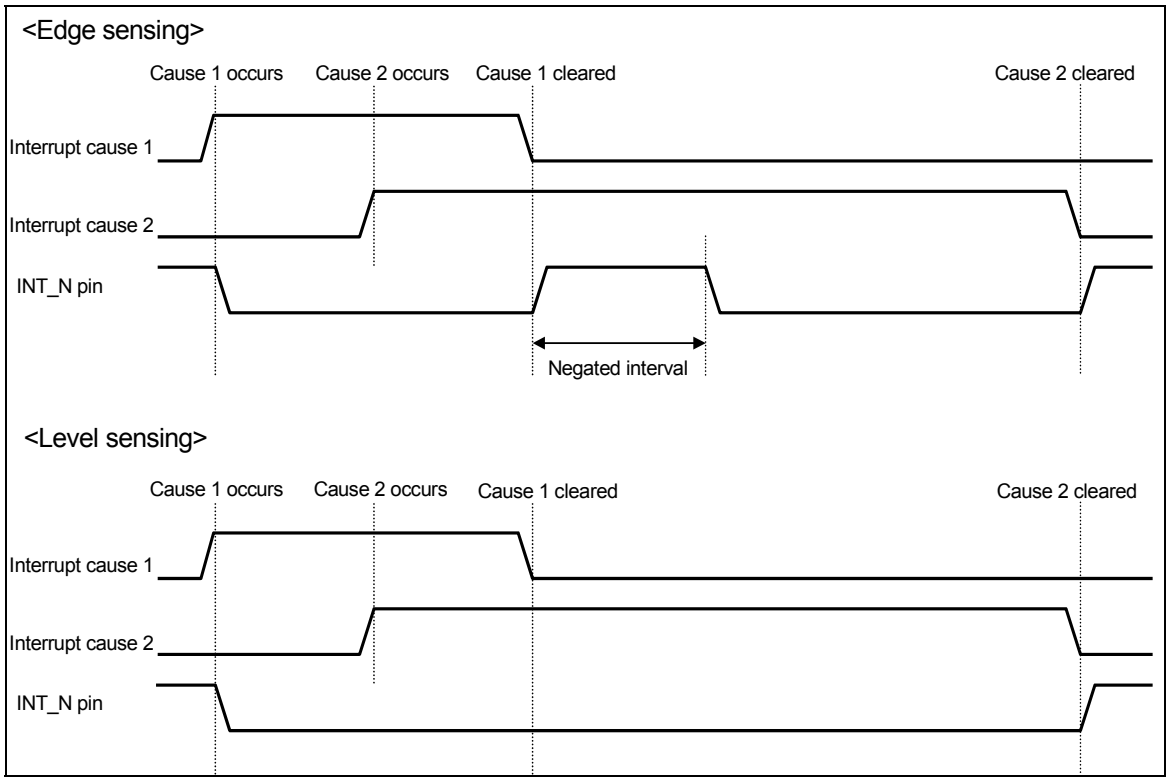


Figure 3.11 INT_N pin operation

Figure 3.12 shows a diagram relating to controller interrupts.

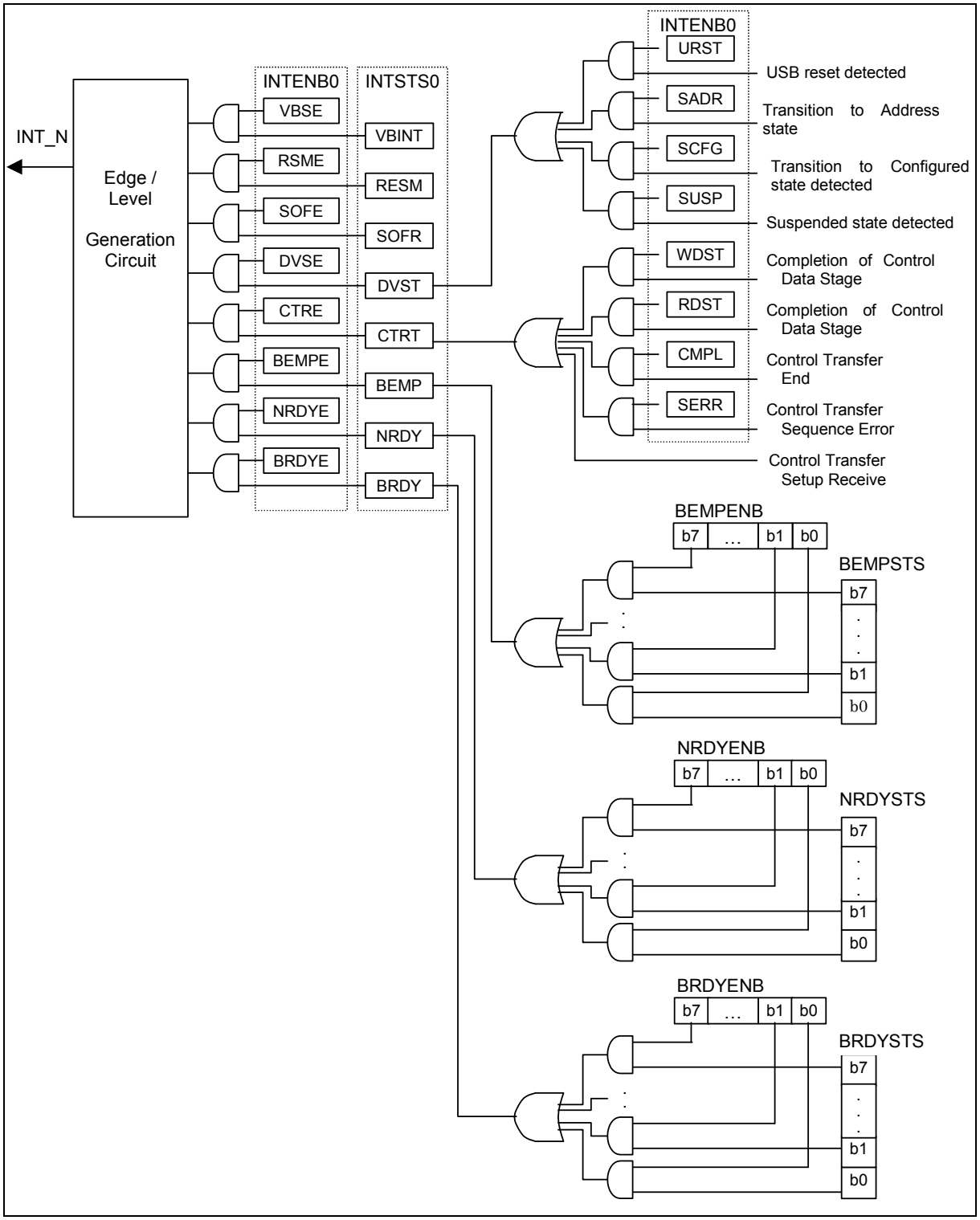


Figure 3.12 Items relating to interrupts

3.2.2 BRDY interrupt

Table 3.9 shows the conditions under which the controller sets "1" to a pertinent bit of the **BRDYSTS** register. Under above condition, the controller generats **BRDY** interrupt, if software enables **PIPERDYE** bit of the **BRDYENB** register and **BRDYE** bit of the **INTENB0** register. Figure 3.13 shows the timing at which **BRDY** interrupts are generated.

The conditions for elimination the **BRDY** bit of the **INTSTS0** register are depend on the setting of the **BRDYM** bit of the **INTENB1** register. Table 3.10 shows the conditions.

Under the conditions noted below, a Zero-Length packet is sent for an IN token, and the **BRDY** interrupt is not generated.

(1) When "0x000" is set in the **MXPS** bit of the **PIPEMAXP** register for a pipe, and the transfer type of the pertinent pipe is bulk IN.

Table 3.9 Conditions under which a BRDY interrupt is generated

Access direction	Transfer direction	Pipe	BFRE	DBLB	Conditions under which a BRDY interrupt is generated
Reading	Receive	DCP	-	0	(1) or (2) bellow; (1) Short packet reception including , Zero-Length packet reception (2) Buffer is full by reception
		1-7	0	0	(1) or (2) bellow; (1) Short packet reception including , Zero-Length packet reception (2) Buffer is full by reception (3) Transaction Counter End when buffer is not full
				1	(1), (2) or (3) bellow (1) One of (a) to (c) conditions is occur when both buffers are waiting for receiption (a) a short packet reception including a Zero-Length packet reception (b) One buffer of two is full by reception (c) Transaction Counter End when a buffer is not full (2) Reading is complete of a buffer, when both buffer are waiting for reading (3) Software set "BCLR=1" to clear a buffer, when both buffer are waiting for reading
			1	Don't Care	(1), (2) or (3) bellow (1) Zero-Length packet reception (2) After a short packet reception, reading data of the packet is complete. (3) After Transaction Counter End reading data of the last packet is complete,.
Writing	Transmit	DCP	-		Doesn't take place
		1-7	0	0	(1), (2), (3) or (4) bellow; (1) Software set direction of transfer to transmitting (2) Packet transmission is completed (3) Software set "ACLRM=1", when there are data waiting to transmitted (4) Software set "SCLR=1", when there are data waiting to transmitte
		1-7	0	1	(1), (2), (3), (4) or (5) bellow; ♣ (1) Software set direction of transfer to transmitting (2) Data is enabled to be transmitted, when there are no buffer waiting to be transmitted. (3) Data is enabled to be transmitted,, when there are no buffer waiting to be transmitted. (a) A buffer is full by writing (b) Software set "BVAL=1" to enable the buffer is ready to tarnsmit (c) DMAC asserts DEND signal to make a buffer be ready to transmit (4) Software set "ACLRM=1", when there are data waiting to transmitted (5) Software set "SCLR=1", when there are data waiting to transmitte
			1	Don't Care	Doesn't take place

With PIPE1-PIPE7, if DMA transfer is being carried out in the reading direction, interrupts can be generated in transfer units, by setting the **BFRE** bit of the **PIPECFG** register. Also, if a Zero-Length packet has been received, the pertinent bit of the **BRDYSTS** register goes to "1", but the data of the pertinent packet cannot be read. The buffer should be cleared ("BCLR=1") after clearing the **BRDYSTS** register.

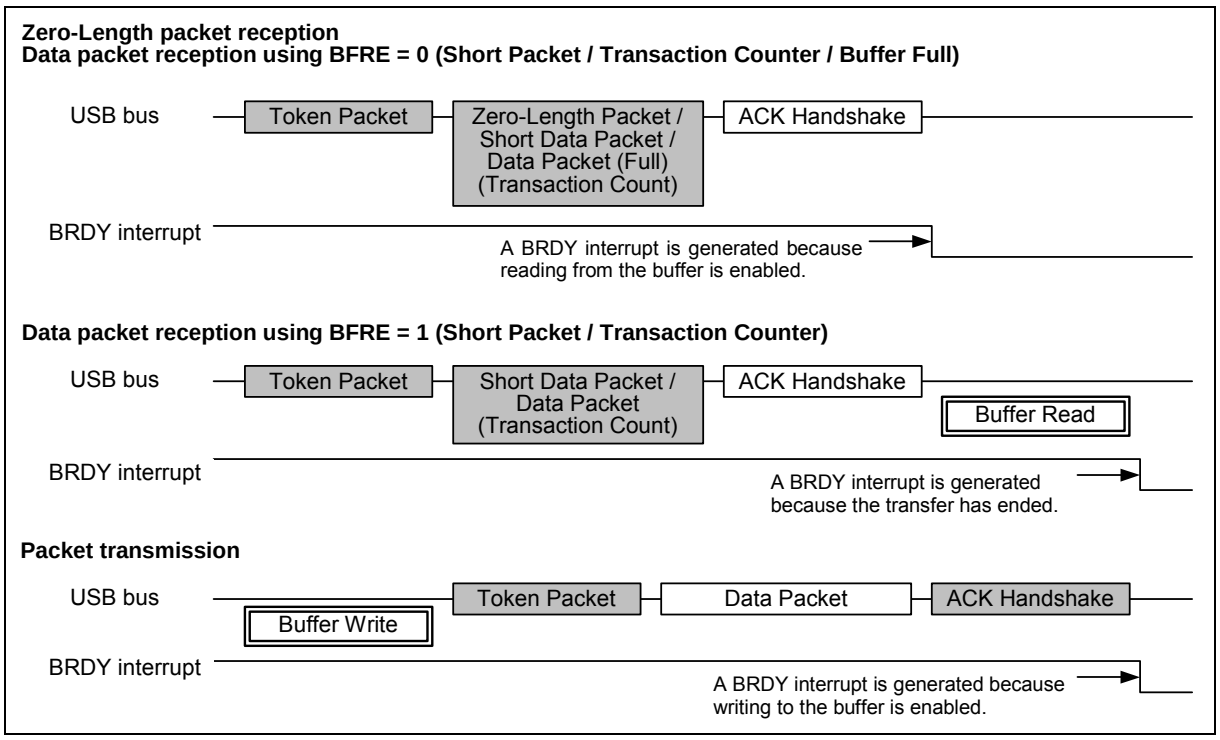


Figure 3.13 Timing at which BRDY interrupts are generated

Table 3.10 **Conditions for elimination of the BRDY bit**

BRDYM	Conditions for elimination of the **BRDY** bit
0	When software clears all enabled bits of the **BRDYSTS** register, the controller clears the **BRDY** bit.
1	When the controller clears all **BSTS** bits which corresponding to BRDY interrupt enebled pipe, the controller clears the **BRDY** bit.

3.2.3 NRDY interrupt

If a pipe is under the conditions such as (1), (2)(a), or (2)(b) bellow, the controller sets "1" to a pertinent bit of the **NRDYSTS** register. In this case, the controller generats **NRDY** interrupt, if software enables **PIPENRDYE** bit of the **NRDYENB** register and **NRDYE** bit of the **INTENB0** register. When software clears all enabled bits of the **NRDYSTS** register, the controller clears the **NRDY** bit.

(1) For data transmission
If an IN token has been received (data underrun) when the the **PID** bit of the **PIPExCTR** register is in the "PID=BUF" and there is no data to be sent in the buffer memory

(6) For data is reception
(a) If an OUT token or a PING token has been received (data overrun) when the **PID** bit of the **PIPExCTR** register is in the "PID=BUF" and there is no area in the buffer memory where data can be stored
(b) In a bulk transfer, when the maximum packet size has not been set ("MXPS=0") and an OUT token or a PING token has been received
(c) When a CRC error or bit stuffing error has occurred during an isochronous transfer

Figure 3.14 shows the timing at which **NRDY** interrupts are generated.

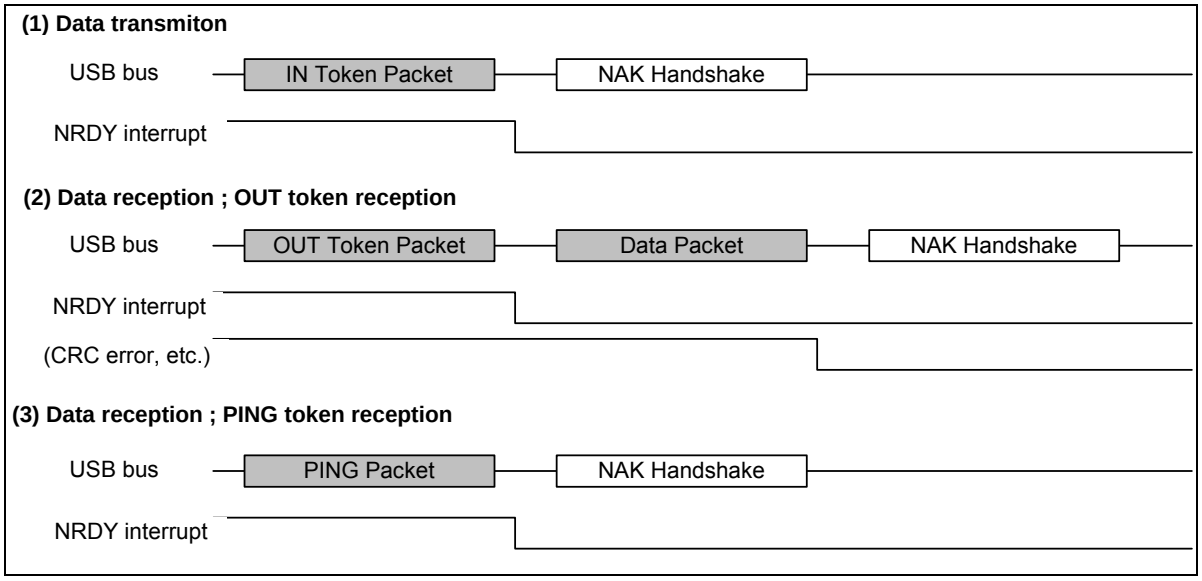


Figure 3.14 Timing at which NRDY interrupts are generated

3.2.4 BEMP interrupt

The table below shows the conditions under which **BEMP** interrupts are generated. The cause of a **BEMP** for the various pipes should be confirmed using the pertinent bit of the **BEMPSTS** register. If an interrupt has been disabled using the **BEMPE** bit of the **INTENB0** register, the interrupt request is set in the pertinent bit of the **BEMPSTS** register. When all of the bits of the **BEMPSTS** register are cleared using the user system control program, the controller clears the **BEMP** bit of the **INTSTS0** register. If a pipe is under the conditions such as (1)(a), (1)(b), or (2) bellow, the controller sets "1" to a pertinent bit of the **BEMPSTS** register. In this case, the controller generats **BEMP** interrupt, if software enables **PIPENBEMPE** bit of the **BEMPENB** register and **BEMPE** bit of the **INTENB0** register. When software clears all enabled bits of the **BEMPSTS** register, the controller clears the **BEMP** bit.

(1) When the sending direction (writing to the buffer memory) has been set
When all of the data stored in the buffer memory has been sent
If a double buffer is being used for the buffer memory, however, the following conditions are observed.
(a) A **BEMP** interrupt is generated if the buffer on one side is empty and sending of data from the buffer on the opposite side has been completed.
(b) A **BEMP** interrupt is generated if data consisting of less than eight bytes is being written to the buffer on one side and sending of data from the buffer on the opposite side has been completed.
(c) A **BEMP** interrupt is not generated if data consisting of eight bytes or more is being written to the buffer on one side and sending of data from the buffer on the opposite side has been completed.

(2) When the receiving direction (reading of the buffer memory) has been set
If the size of the data packet that was received exceeded the maximum packet size
At this point, if the other maximum packet size parameters were set to a value other than "0" ("MXPS≠0"), the controller sets the **PID** bit of the pertinent pipe to "STALL".

Figure 3.15 shows the timing at which **BEMP** interrupts are generated.

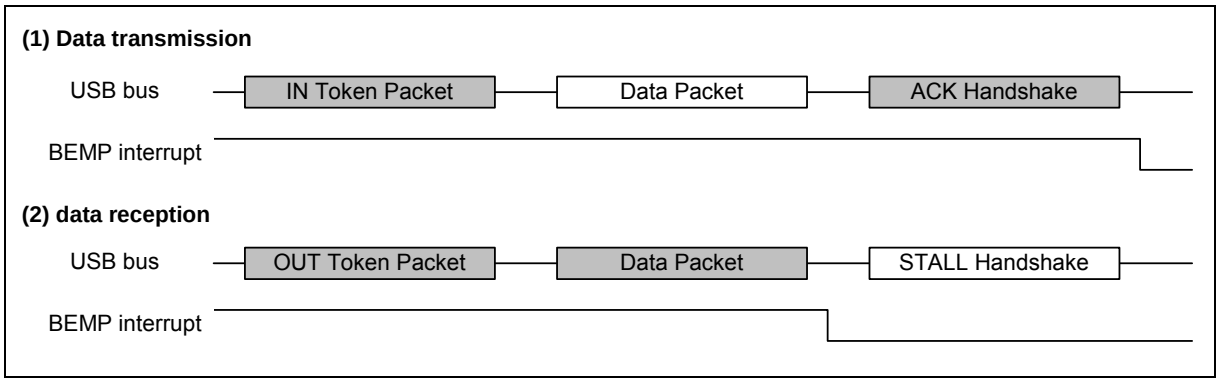
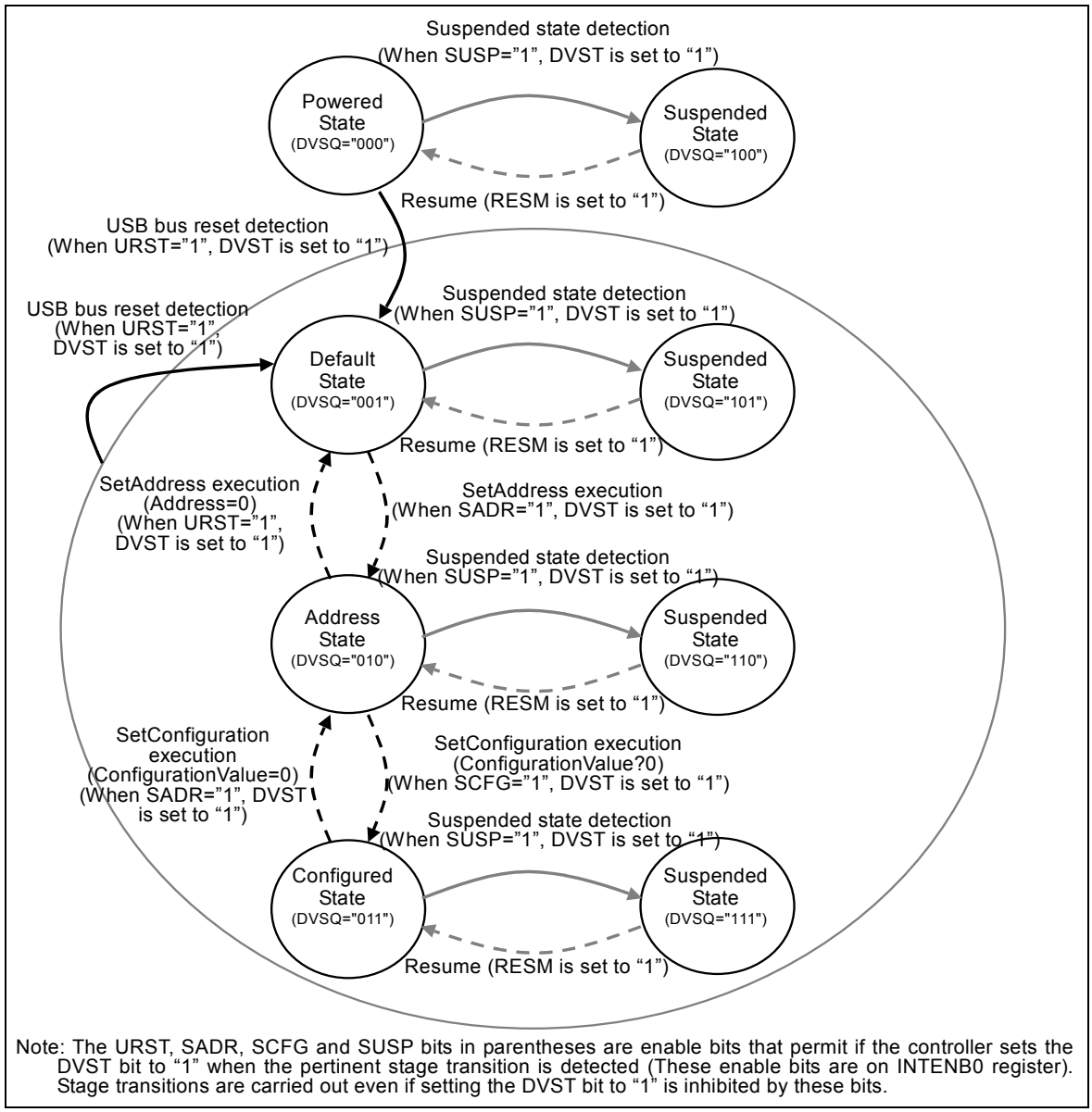


Figure 3.15 Timing at which BEMP interrupts are generated

3.2.5 Device state transition interrupt

Figure 3.16 shows a diagram of the controller device state transitions. The controller controls device states and generates device state transition interrupts. However, recovery from the suspended state (Resume signal detection) is detected by means of the Resume interrupt. The device state transition interrupt can be set when interrupts are enabled or disabled individually, using the **INTENB0** register. Also, the device state that underwent a transition can be confirmed using the **DVSQ** bit of the **INTSTS0** register.

When making a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.



Note: The URST, SADR, SCFG and SUSP bits in parentheses are enable bits that permit if the controller sets the DVST bit to "1" when the pertinent stage transition is detected (These enable bits are on INTENB0 register). Stage transitions are carried out even if setting the DVST bit to "1" is inhibited by these bits.

Figure 3.16 Device state transitions

3.2.6 Control transfer stage transition interrupt

Figure 3.17 shows a diagram of how the controller handles the control transfer stage transition. The controller controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled individually, using the **INTENB0** register. Also, the transfer stage that underwent a transition can be confirmed using the **CTSQ** bit of the **INTSTS0** register.

The control transfer sequence errors are noted below. If an error occurs, the **PID** bit of the **DCPCTR** register goes to "1X" (STALL).

(1) During control read transfers
  (a) At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all
  (b) An IN token is received at the status stage
  (c) A packet is received at the status stage for which the data packet is "DATAPID=DATA0"

(2) During control write transfers
  (a) For the OUT token of the data stage, when there have been no ACK responses at all, the IN token is received
  (b) A packet is received at the data stage for which the first data packet is "DATAPID=DATA0"
  (c) At the status stage, an OUT or PING token is received

(3) During no-data control transfers
  (a) At the status stage, an OUT or PING token is received

At the control write transfer stage, if the number of received data elements exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. Also, at the control read transfer status stage, packets other than Zero-Length packets are received by an ACK response being carried out, and the transfer ends normally.

If a **CTRT** interrupt occurs in response to a sequence error ("SERR=1"), the "CTSQ=110" value is held until "CTRT=0" is written from the user system (the interrupt status is cleared). Because of this, while "CTSQ=110" is being held, the **CTRT** interrupt that ends the setup stage will not be generated even if a new USB request is received. (The controller holds the setup stage end, and after the interrupt status has been cleared bysoftware, a setup stage end interrupt is generated.)

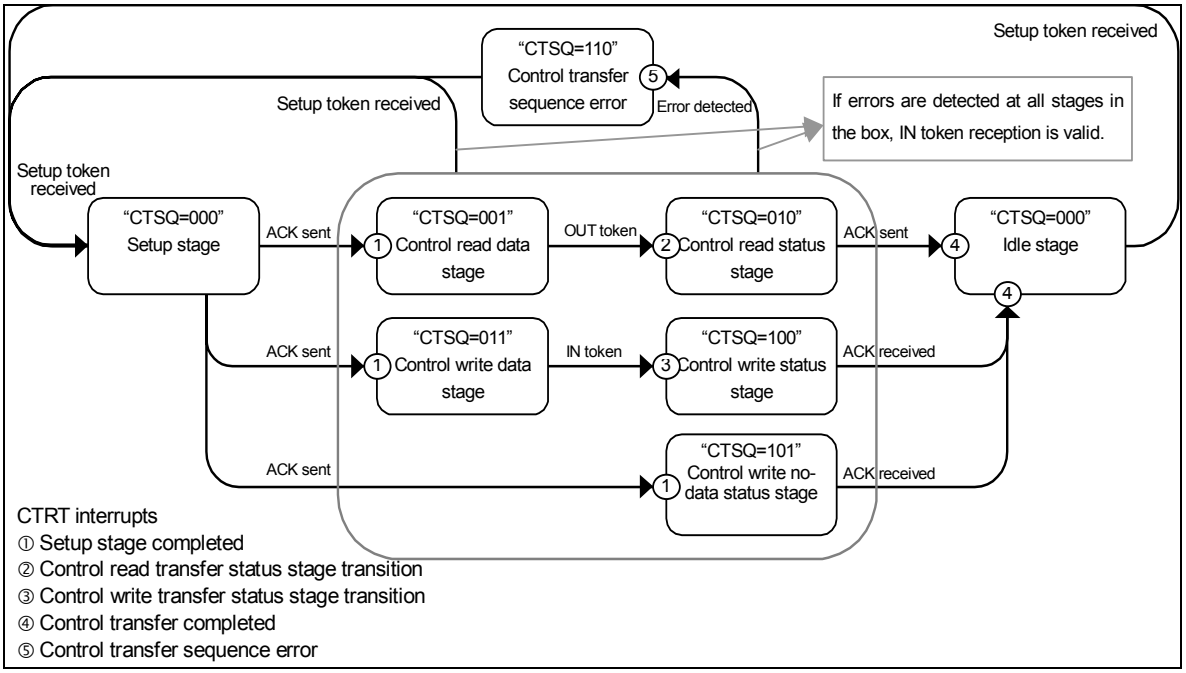


Figure 3.17 Control transfer stage transitions

3.2.7 Frame refresh interrupt

Figure 3.18 shows an example of the **SOFR** interrupt output timing of the controller. When the frame number is refreshed, or a damaged SOF packet is detected, the **SOFR** interrupt is generated. The interrupt operation should be specified using the **SOFRM** bit of the **FRMNUM** register.

(1) When "SOFRM=0" is selected
The **SOFR** interrupt is generated at the timing at which the frame number is refreshed (intervals of approximately 1 ms). Interrupts are generated by the internal interpolation function even if an SOF packet is damaged or missing. During Hi-Speed communication as well, interrupts are generated at the timing at which the frame number is refreshed (intervals of approximately 1 ms).

(2) When "SOFRM=1" is selected
The **SOFR** interrupt is generated when SOF packets are damaged and when they are missing. During Hi-Speed communication, the interrupt is generated only if the first packet of a uSOF packet with the same frame number is damaged or missing.
(Corrupted and missing SOFs are recognized by the SOF interpolation function. For detailed information, please refer to Chapter 3.10, SOF interpolation function.)

If the controller detects a new SOF packet during Full-Speed operation, it refreshes the frame number and generates an **SOFR** interrupt. However, if the system does not enter the µSOF lock state during Hi-Speed operation, the frame number is not refreshed, and no **SOFR** interrupt is generated. Also, the SOF interpolation function is not activated. The µSOF lock state is the state in which uSOF packets with different frame numbers are received twice in succession without an error occurring.

The conditions under which µSOF lock monitoring begins, and under which µSOF lock monitoring stops, are as noted below.

(1) Conditions under which µSOF lock monitoring begins
"USBE=1" and the internal clock (SCKE) is being supplied

(2) Conditions under which µSOF lock monitoring stops
"USBE=0 (S/W reset)" or a USB bus reset is received, or a suspended state is detected

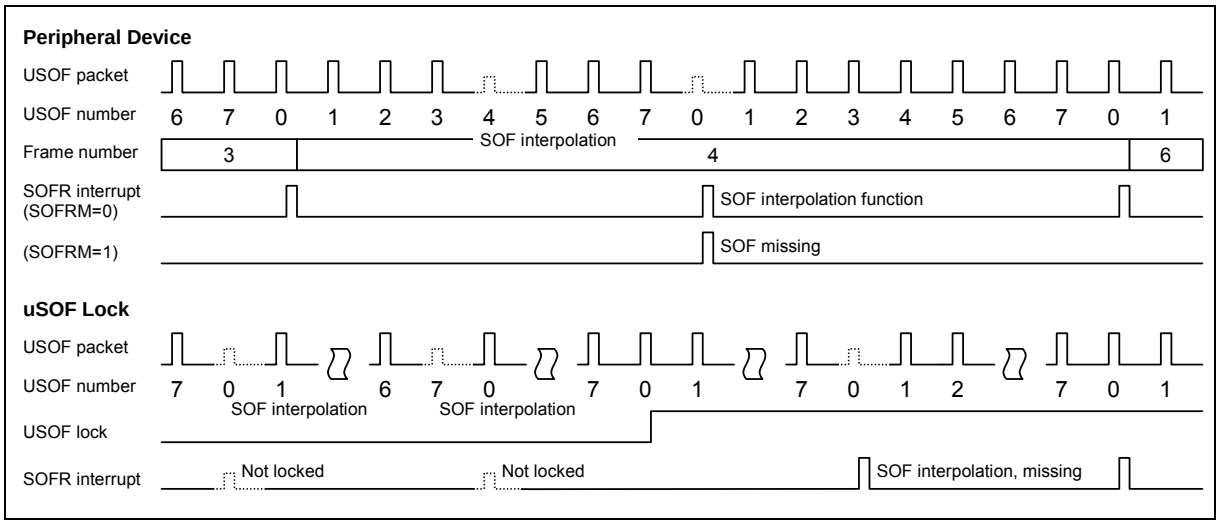


Figure 3.18 Example of SOFR interrupt output timing

3.2.8 VBUS interrupt

If there has been a change in the **VBUS** pin, the **VBUS** interrupt is generated. The level of the **VBUS** pin can be detected using the **VBSTS** bit of the **INTSTS0** register. Confirmation can be made of whether the host controller is connected or disconnected using the **VBUS** interrupt. However, if the user system is booted with the host controller connected, the first **VBUS** interrupt is not generated, because there is no change in the **VBUS** pin.

3.2.9 Resume interrupt

The **RESM** interrupt is generated if the device state is the suspended state, and the USB bus state has changed (from the J-State to the K-State, or from the J-State to SE0). Recovery from the suspended state is detected by means of the Resume interrupt.

3.3 Pipe control

Table 3.11 shows the pipe setting items of the controller. With USB data transfers, data communication has to be carried out using the logic pipe called the end point. This controller has eight pipes that are used for data transfers. Settings should be entered for each of the pipes in conjunction with the specifications of the user system.

Table 3.11 Pipe setting items

Register name	Bit name	Setting contents	Note
DCPCFG PIPECFG	TYPE	Specifies the transfer type	Please refer to 3.3.1.
	BFRE	Selects the BRDY interrupt mode	PIPE1-5: Can be set Please refer to 3.4.3.5 and 3.4.3.6.
	DBLB	Selects a single or double buffer	PIPE1-5: Can be set Please refer to 3.4.1.5.
	CNTMD	Selects continuous transfer or non-continuous transfer	DCP: Can be set PIPE1-2: Can be set (can be set only when bulk transfer has been selected) PIPE3-5: Can be set With continuous transmission and reception, the buffer size should be set to an integer multiple of the payload. Please refer to 3.4.1.6.
	DIR	Selects transfer direction (reading or writing)	IN or OUT can be set. Please refer to 3.4.2.1 (DCP is controlled by ISEL).
	EPNUM	End point number	Please refer to 3.3.2.
	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE1-2: Can be set (can be set only when bulk transfer has been selected) PIPE3-5: Can be set Please refer to 3.3.7.
PIPEBUF	BUFSIZE	Buffer memory size	DCP: Cannot be set (fixed at 256 bytes) PIPE1-5: Can be set (can be specified up to a maximum of 2 KB in 64-byte units) PIPR6-7: Cannot be set (fixed at 64 bytes) Please refer to 3.4.1.
	BUFNMB	Buffer memory number	DCP: Cannot be set (areas fixed at 0-3) PIPE1-5: Can be set (can be specified in areas 6-4F) PIPE6-7: Cannot be set (areas fixed at 4-5) Please refer to 3.4.1.
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	Please refer to Chapter 3.3.3.
PIPEPERI	IFIS	Buffer Flush	PIPE1-2: Can be set (only when isochronous transfer has been selected) PIPE3-7: Cannot be set Please refer to 3.9.5.
	IITV	Interval Counter	PIPE1-2: Can be set (only when isochronous transfer has been selected) PIPE3-7: Cannot be set Please refer to 3.9.3.
DCPCTR PIPExCTR	BSTS	Buffer Status	Please refer to 3.4.1.1 (also related to DIR / ISEU)
	INBUFM	IN Buffer Monitor	Please refer to 3.4.1.1 (also related to DIR / ISEU)
	ACLRM	Auto Buffer Clear	Enabled / disabled setting can be set when buffer memory reading is set. Please refer to 3.4.1.4.
	SQCLR	Sequence Clear	Clears the data toggle bit. Please refer to 3.3.6.
	SQSET	Sequence Set	Sets the data toggle bit. Please refer to 3.3.6.
	SQMON	Sequence Confirm	Confirms the data toggle bit. Please refer to 3.3.6.
	PID	Response PID	Please refer to 3.3.4.

3.3.1 Transfer types

The **TYPE** bit of the **PIPEPCFG** register is used to specify the type of transfer for the various pipes. The types of transfer that can be set for the pipes are noted below.

(1) DCP: No setting is necessary (fixed at control transfer).
(2) PIPE1-2: These should be set to bulk transfer or isochronous transfer.
(3) PIPE3-5: These should be set to bulk transfer.
(4) PIPE6-7: These should be set to interrupt transfer.

3.3.2 End point number

The **EPNUM** bit of the **PIPEPCFG** register is used to set the end point numbers for the various pipes. DCP is fixed at end point 0. The other pipes can be set from end point 1 to end point 15.

(1) DCP: No setting is necessary (fixed at end point 0).
(2) PIPE1-7: "1" to "15" should be selected and set.
However, these should be set so that the combination of the **DIR** bit and the **EPNUM** bit is a unique combination.

3.3.3 Maximum packet size setting

The **MXPS** bit of the **DCPMAXP** register and the **PIPEMAXP** register is used to set the maximum packet size for the various pipes. DCP and PIPE1-5 can be set to any of the maximum pipe sizes defined by USB specification. For PIPE6-7, 64 bytes is the upper limit for the maximum packet size. The maximum packet size should be set before beginning the transfer ("PID=BUF").

(1) DCP: "64" should be set when using Hi-Speed operation.
(2) DCP: Select and set "8", "16", "32" or "64" when using Full-Speed operation.
(3) PIPE1-5: "0" or "512" should be set when using Hi-Speed bulk transfer.
(4) PIPE1-5: Select and set "0", "8", "16", "32" or "64" when using Full-Speed bulk transfer.
(5) PIPE1-2: Set a value between "1" and "1024" when using Hi-Speed isochronous transfer.
(6) PIPE1-2: Set a value between "1" and "1023" when using Full-Speed isochronous transfer.
(7) PIPE6-7: Set a value between "1" and "64".

The High-Bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

Also, setting "MXPS=0" for pipes when bulk transfer is being used results in the following operations.

(1) Bulk IN:
Data cannot be written to the buffer memory.
When "PID=BUF" is set, a Zero-Length packet is sent in response to an IN token.
The **BRDY**, **NRDY**, and **BEMP** interrupts are not generated.
(2) Bulk OUT:
The data in the received data packets cannot be stored in the buffer memory.
When "PID=BUF" is set, the NAK response is sent in response to an OUT token.
In this case, **NRDY** interrupts are generated, but **BRDY** and **BEMP** interrupts are not.

3.3.4 Response PID

The PID bit of the DCPCTR register and PIPExCTR register is used to set the response PID for the various pipes.
Operation with the various settings

(1) NAK setting: The "NAK response" is always returned in response to the generated transaction.
(2) BUF setting: Responses are made to transactions based on the status of the buffer memory.
(3) STALL setting: The "STALL" response is always returned in response to the generated transaction.

Also, for setup transactions, an "ACK response" is always returned, regardless of the PID setting, and the USB request is stored in the register.

The controller may carry out writing to the **PID** bit, depending on the results of the transaction.
If writing to the PID bit is generated by the controller

(1) NAK setting:
  (a) When the SETUP token is received normally (DCP only)
  (b) When the transaction counter has ended or a short packet is received, if the **SHTNAK** bit of the **PIPECFG** register has been set to "1" for bulk transfers
(2) BUF setting: There is no BUF writing by the controller.
(3) STALL setting:
  (a) When an error has been detected in a received data packet indicating that the data size exceeds the maximum packet size
  (b) When a control transfer stage transition error has been detected

3.3.5 Registers that should not be set in the USB communication enabled ("PID=BUF") state

ISEL bit of the **CFIFOSEL** register (applies only when DCP is selected)
TGL and **SCLR** bits of the **CFIFOSIE** register
DCLRM, **TRENB**, **TRCLR**, and **DEZPM** bits of the **DxFIFOSEL** register
TRNCNT bit of the **DxFIFOTRN** register
The various bits of the **DCPCFG** and **DCPMAXP** registers
The various bits of the **DCPCTR** register (except for the **CCPL** bit)
The various bits of the **PIPECFG**, **PIPEBUF**, and **PIPEMAXP** registers
The various bits of the **PIPEPERI** and **PIPExCTR** registers

3.3.6 Data PID sequence bit

The controller toggles the data PID sequence bit when data is transferred normally. Next, the sequence bit of the data PID that was sent can be used to confirm the **SQMON** bit of the **DCPCTR** register and the **PIPExCTR** register. When data is sent, the sequence bit switches at the timing at which the ACK handshake is received, and when data is received, the sequence bit switches at the timing at which the ACK handshake is sent. Also, the **SQCLR** bit and the **SQSET** bit of the **DCPCTR** register and the **PIPExCTR** register can be used to change the data PID sequence bit.

With pipes for which isochronous transfer has been set, sequence bit operation cannot be carried out using the **SQSET** bit.

3.3.7 Response PID=NAK function

The controller has a function that disables pipe operation ("Response PID=NAK") at the timing at which the final data packet of a transaction is received (the controller automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the **SHTNAK** bit of the **PIPECFG** register to "1".

When a double buffer is being used for the buffer memory, using this function enables reception of data packets in transfer units. Also, if pipe operation has been disabled, the pipe has to be set to the enabled state again ("Response PID=BUF") usingsoftware.

This function can be used for operation only when using bulk transfers.

3.4 Buffer memory

3.4.1 Buffer memory allocation

Figure 3.19 shows an example of a buffer memory map for the controller. The buffer memory is an area shared by the user system control CPU and the controller. In the buffer memory status, there are times when the access right to the buffer memory is allocated to the user system (CPU side), and times when it is allocated to the controller (SIE side).

The buffer memory sets independent areas for each pipe. In the memory areas, 64 bytes comprise one block, and the memory areas are set using the first block number of the number of blocks (specified using the **BUFNMB** and **BUFSIZE** bits of the **PIPEBUF** register). Moreover, three FIFO ports are used for access to the buffer memory (reading and writing data). A pipe is assigned to the FIFO port by specifying the pipe number using the **CURPIPE** bit of the **C/DxFIFOSEL** register.

The buffer statuses of the various pipes can be confirmed using the **BSTS** bit of the **DCPCTR** register and **PIPExCTR** register. Also, the access right of the FIFO port can be confirmed using the **FRDY** bit of the **C/DxFIFOCTR** register.

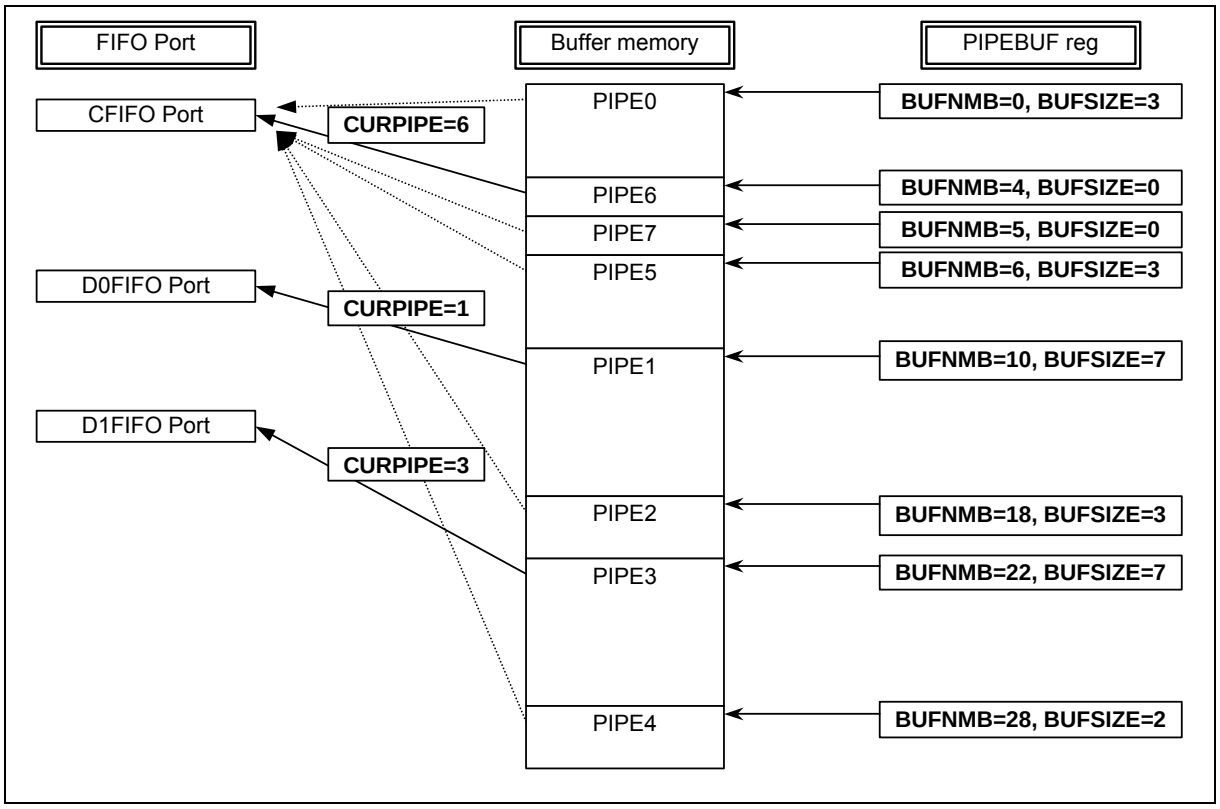


Figure 3.19 Example of buffer memory map

3.4.1.1 Buffer status

Table 3.12 shows the buffer status. The buffer memory status can be confirmed using the **BSTS** bit and the **INBUFM** bit. The access direction for the buffer memory can be specified using either the **DIR** bit of the **PIPExCFG** register or the **ISEL** bit of the **CFIFOSEL** register (when DCP is selected).

the **INBUFM** bit is valid for IN direction pipe ("DIR=1")

For a IN pipe uses double buffer, software can refer the BSTS bit to monitor the buffer memory status of CPU side and the **INBUFM** bit to monitor the buffer memory status of SID side. In the case like the **BEMP** interrupt may not show bufer empty status because the CPU (DMAC) writes data slowly, software can use the **INBUFM** bit to tell the end of sending.

Table 3.12 Buffer statuses and the BSTS bit

ISEL or DIR	BSTS	Buffer memory state
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the CPU is inhibited.
0 (receiving direction)	1	There is received data, or a Zero-Length packet has been received. Reading from the CPU is allowed. However, because reading is not possible when a Zero-Length packet is received, the buffer must be cleared.
1 (sending direction)	0	The transmission has not been finished. Writing to the CPU is inhibited.
1 (sending direction)	1	Writing to the CPU is allowed. (1) "DBLB=0"(Single buffer) ; The transmission has been finished. (2) "DBLB=1"(Double buffer) ; The transmission for one side of the buffer has been finished.

Table 3.13 Buffer statuses and the INBUFM bit

I DIR	INBUFM	Buffer memory state
0 (receiving direction)	invaild	invaild
1 (sending direction)	0	The transmission has not been finished. There is no waiting data to be sent.
1 (sending direction)	1	There is data to be sent, because CPU(DMAC) has written data to the buffer.

3.4.1.2 Buffer clearing

Table 3.14 shows the clearing of the buffer memory by the controller. The buffer memory can be cleared using the four bits indicated below.

Table 3.14 Buffer clearing

Bit name	BCLR	SCLR	DCLRM	ACLRM
Register	CFIFOCTR register DxFIFOCTR register	CFIFOSIE register	DxFIFOSEL register	PIPExCTR register
Function	Clears the buffer memory on the CPU side	Clears the buffer memory on the SIE side	In this mode, after the data of the specified pipe has been read, the buffer memory is cleared automatically. See 3.4.3.5.	This is the Auto Buffer Clear mode, in which all of the received packets are destroyed. See 3.4.1.4.
Clearing method	Cleared by writing "1"	Cleared by writing "1"	"1": Mode valid "0": Mode invalid	"1": Mode valid "0": Mode invalid

3.4.1.3 Buffer areas

Table 3.15 shows the FIFO buffer memory map of the controller. The buffer memory has special fixed areas to which pipes are assigned in advance, and user areas that can be set by the user. The buffer for the DCP is a special fixed area that is used both for control read transfers and control write transfers. The PIPE6-7 area is assigned in advance, but the area for pipes that are not being used can be assigned to PIPE1-5 as a user area. The settings should ensure that the areas of the various pipes do not overlap. Also, the buffer size should not be specified using a value that is less than the maximum packet size.

Table 3.15 Buffer memory map

Buffer memory no.	Buffer size	Pipe setting	Note
0 – 3	256 bytes	DCP special fixed area	Single buffer, continuous transfers enabled
4	64 bytes	Fixed area for PIPE6	Single buffer
5	64 bytes	Fixed area for PIPE7	Single buffer
6 – 4F	4736 bytes	PIPE1-5 user area	Double buffer can be set, continuous transfers enabled

3.4.1.4 Auto Buffer Clear mode function

With this controller, all of the received data packets are discarded if the **ACLRM** bit of the **PIPExCTR** register is set to "1". If a normal data packet has been received, however, the ACK response is returned to the host controller. This function can be set only in the buffer memory reading direction.

Also, if the **ACLRM** bit is set to "1" and then to "0", the buffer memory of the pertinent pipe can be cleared regardless of the access direction.

An access cycle of at least 100 ns is required between "ACLRM=1" and "ACLRM=0".

3.4.1.5 Buffer memory specifications (single / double setting)

Either a single or double buffer can be selected for PIPE1-5, using the **DBLB** bit of the **PIPExCFG** register. The double buffer is a function that assigns two memory areas specified with the **BUFSIZE** bit of the **PIPEBUF** register to the same pipe. Figure 3.20 shows an example of buffer memory settings for the controller.

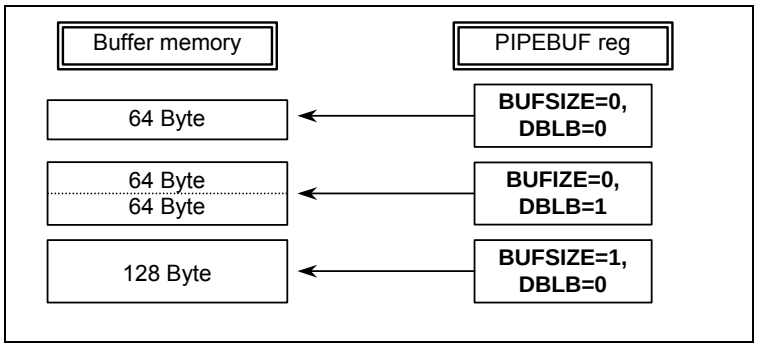


Figure 3.20 Example of buffer memory settings

3.4.1.6 Buffer memory operation (continuous transfer setting)

Either the continuous transfer mode or the non-continuous transfer mode can be selected, using the **CNTMD** bit of the **DCPCFG** register and the **PIPExCFG** register. This selection is valid for pipes 0-5.

The continuous transfer mode function is a function that sends and receives multiple transactions in succession. When the continuous transfer mode is set, data can be transferred without interrupts being issued to the CPU, up to the buffer sizes assigned for each of the pipes.

In the continuous sending mode, the data being written is divided into packets of the maximum packet size and sent. If the data being sent is less than the buffer size (short packet, or the integer multiple of the maximum packet size is less than the buffer size), "BVAL=1" must be set after the data being sent has been written.

In the continuous reception mode, interrupts are not issued during reception of packets up to the buffer size, until the transaction counter has ended, or until a short packet is received.

Figure 3.21 shows an example of buffer memory operation for the controller.

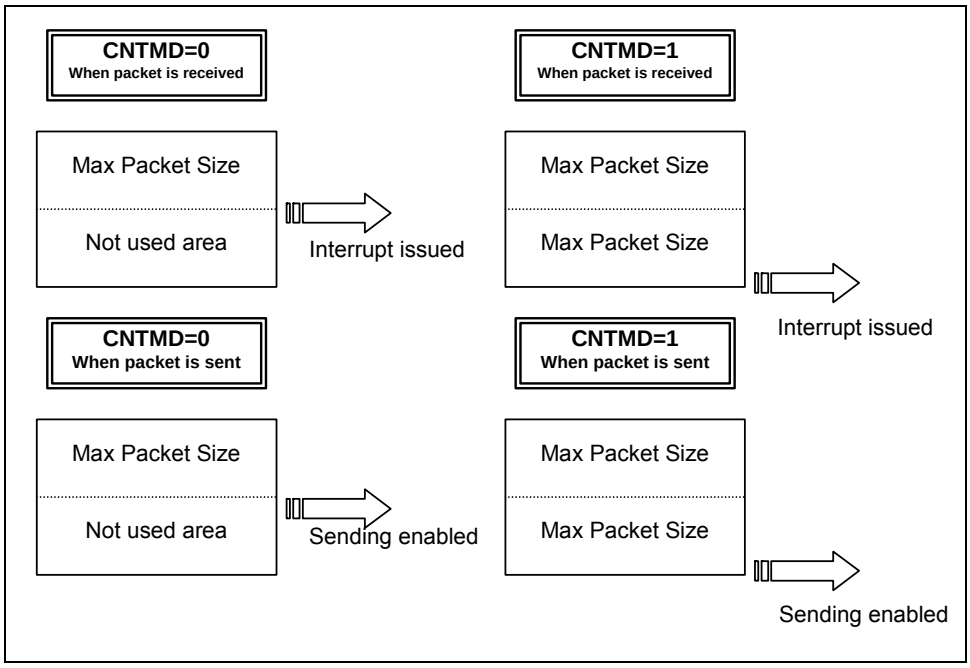


Figure 3.21 Example of buffer memory operation

3.4.2 FIFO port functions

Table 3.16 shows the settings for the FIFO port functions of the controller. When data writing is being accessed, writing data until the buffer is full (or the maximum packet size for non-continuous transfers) automatically enables sending of the data. To enable sending of data before the buffer is full (or before the maximum packet size for non-continuous transfers), the **BVAL** bit of the **C/DxFIFOCTR** register (or the **DEND** signal when using DMA transfers) must be set to end the writing. Also, to send a Zero-Length packet, the **BCLR** bit of the same register must be used to clear the buffer and then the **BVAL** bit set in order to end the writing.

When accessing reading, reception of new packets is automatically enabled if all of the data has been read. However, data cannot be read when a Zero-Length packet is being received (DTLN=0), so the **BCLR** bit of the register must be used to release the buffer. The length of the data being received can be confirmed using the **DTLN** bit of the **C/DxFIFOCTR** register.

Table 3.16 FIFO port function settings

Register name	Bit name	Function	Reference	Note
C/DxFIFOSEL	REW	Buffer memory rewind (re-read, re-write)	3.4.2.2	
	DCLRM	Automatically clears data received for a specified pipe after the data has been read	3.4.1.2 3.4.3.4 3.4.4	For DxFIFO only DMA transfer assumed
	DREQE	Asserts DREQ signal	3.4.3	For DxFIFO only
	MBW	FIFO port access bit width	3.4.2.1	
	TRENB	Enables transaction counter operation	3.4.2.5	For DxFIFO only
	TRCLR	Clears the current number of transactions	3.4.2.5	For DxFIFO only
	DEZPM	Zero-Length packet addition mode	3.4.3.3	For DMA only
	ISEL	FIFO port access direction	3.4.2.1	For DCP only
C/DxFIFOCTR	BVAL	Ends writing to the buffer memory	3.4.2	
	BCLR	Clears the buffer memory on the CPU side	3.4.1.2	
	DTLN	Confirms the length of received data	3.4.2	
DxFIFOTRN	TRNCNT	Sets the received transaction count	3.4.2.5	For DxFIFO only
CFIFOSIE (excluding DCP)	TGL	CPU / SIE buffer toggle	3.4.2.3	For CFIFO only
	SCLR	Clears the buffer memory on the SIE side	3.4.2.4	For CFIFO only
External pin	DEND	Ends writing to the buffer memory	3.4.3.4	For DMA transfer only

3.4.2.1 FIFO port selection

Table 3.17 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed is selected using the **CURPIPE** bit of the **C/DxFIFOSEL** register. After the pipe has been selected, "FRDY=1" should be confirmed before accessing the FIFO port.

Also, the bus width to be accessed should be selected using the **MBW** bit. The buffer memory access direction conforms to the **DIR** bit of the **PIPExCFG** register. However, the **ISEL** bit determines this only for the DCP.

Table 3.17 FIFO port access categorized by pipe

Pipe	Access method	Port that can be used
DCP	CPU access	CFIFO port register
PIPE1~PIPE7	CPU access	CFIFO port register DxFIFO port register
	DMA access	DxFIFO port register

3.4.2.2 REW bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing using the current pipe once again. The **REW** bit of the **C/DxFIFOSEL** register is used for this.

If a pipe is selected when the **REW** bit is set to "1" and at the same time the **CURPIPE** bit of the **C/DxFIFOSEL** register is set, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. Also, if a pipe is selected with "0" set for the **REW** bit, data can be read and written in continuation of the previous selection, without the pointer used for reading from and writing to the buffer memory being reset.

To access the FIFO port, "FRDY=1" must be confirmed after selecting a pipe.

3.4.2.3 Reading the buffer memory on the SIE side (CFIFO port reading direction)

Even in the "FRDY=0" state, when data cannot be read from the buffer memory, confirming the **SBUSY** bit in the **CFIFOSIE** register and setting "1" for the **TGL** bit makes it possible for the controller to read and access data on the SIE side. "PID=NAK" should be set and "SBUSY=0" confirmed, and then "TGL=1" written. The controller is then able to read data from the **CFIFO** register. This function can be used only in the buffer memory reading direction. Also, the **BRDY** interrupt is generated by operation of the **TGL** bit.

"1" should not be written for the **TGL** bit in the following circumstances.

(1) When DCP is selected
(2) While the buffer memory is being read
(3) Pipes in the buffer memory writing direction

3.4.2.4 Clearing the buffer memory on the SIE side (CFIFO port writing direction)

The controller can cancel data that is waiting to be sent, by confirming the **SBUSY** bit of the **CFIFOSIE** register and setting "1" for the **SCLR** bit.

"PID=NAK" should be set and "SBUSY=0" confirmed, and then "SCLR=1" written. The controller is then able to write new data from the **CFIFO** register. This function can be used only in the buffer memory writing direction. Also, the **BRDY** interrupt is generated by operation of the **SCLR** bit.

"1" should not be written for the **SCLR** bit in the following circumstances.

(1) When DCP is selected
(2) While data is being written to the buffer memory
(3) Pipes in the buffer memory reading direction

3.4.2.5 Transaction counter (DxFIFO port reading direction)

When the specified number of transactions have been completed in the data packet receiving direction, the controller is able to recognize that the transfer has ended. The transaction counter is a function that operates when the pipe selected by means of the **DxFIFO** port has been set in the direction of reading data from the buffer memory. The transaction counter has a **TRNCNT** register that specifies the number of transactions and a current counter that counts the transactions internally. When the current counter matches the number of transactions specified in the **TRNCNT** register, reading is enabled for the buffer memory. The current counter of the transaction counter function is initialized by the **TRCLR** bit, so that the transactions can be counted again starting from the beginning. The information read by the **TRNCNT** register differs depending on the setting of the **TRENB** bit.

TRENB=0: The set transaction counter value can be read.
TRENB=1: The value of the current counter that counts the transactions internally can be read.

The conditions for changing the **CURPIPE** bit are as noted below.

(1) The **CURPIPE** bit should not be changed until the transaction for the specified pipe has ended.
(2) The **CURPIPE** bit cannot be changed if the current counter has not been cleared.

The operation conditions for the **TRCLR** bit are as noted below.

(1) If the transactions are being counted and "PID=BUF", the current counter cannot be cleared.
(2) If there is any data left in the buffer, the current counter cannot be cleared.

3.4.3 DMA transfers (DxFIFO port)

3.4.3.1 An overview of DMA transfers

For pipes 1 to 7, the FIFO port can be accessed using the DMAC.

For DMA transfers, there are two modes that can be selected. One is the cycle steal transfer mode, in which the **DREQ** signal is asserted each time a data element (8 or 16 bits) is transferred. The other is the burst transfer mode, in which the **DREQ** signal continues to be asserted until all of the data in the buffer memory has been transferred. For information regarding the timing, please refer to Chapter4, Electrical characteristics.

The unit of transfer to the FIFO port (8 or 16 bits) should be selected using the **MBW** bit of the **DxFIFOSEL** register, and the pipe targeted for the DMA transfer should be selected using the **CURPIPE** bit. The selected pipe should not be changed during the DMA transfer.

3.4.3.2 Selecting the DMA control signal

The **DFORM** bit of the **DMAxCFG** register should be used to select the pin to be used in the DMA transfer. Table 3.18 shows the DMA control pins of the controller.

Table 3.18 DMA control pins

Access method	Register				Pin						Note
	DREQE	DFORM			DATA BUS	DREQ	DACK	RD/WR	ADDR +CS	DSTB	
CPU bus 0	0	0	0	0	CPU	-	-	v	v	-	CPU access
CPU bus 1	1	0	0	0	CPU	v	-	v	v	-	DMA with CPU bus
CPU bus 2	1	0	1	0	CPU	v	v	v	*2)	-	DMA with CPU bus
CPU bus 3	1	0	1	1	CPU	v	v	-	*2)	-	DMA with CPU bus
SPLIT bus 1	1	1	1	0	SPLIT	v	v	-	-	v	Split bus *1)
SPLIT bus 2	1	1	0	0	SPLIT	v	v	-	-		Split bus

***1)** This access method can be set only in relation to the **D0FIFO** port. Also, if using the **D0FIFO** port with this setting and also using the **D1FIFO** port, the **D1FIFO** port should be used with the setting "DFROM=000".

***2)** When this access method is set CS_N pin shoul be held inactive (should be held in the high state) while the DMAC accsesses to the **DxFIFO** port.

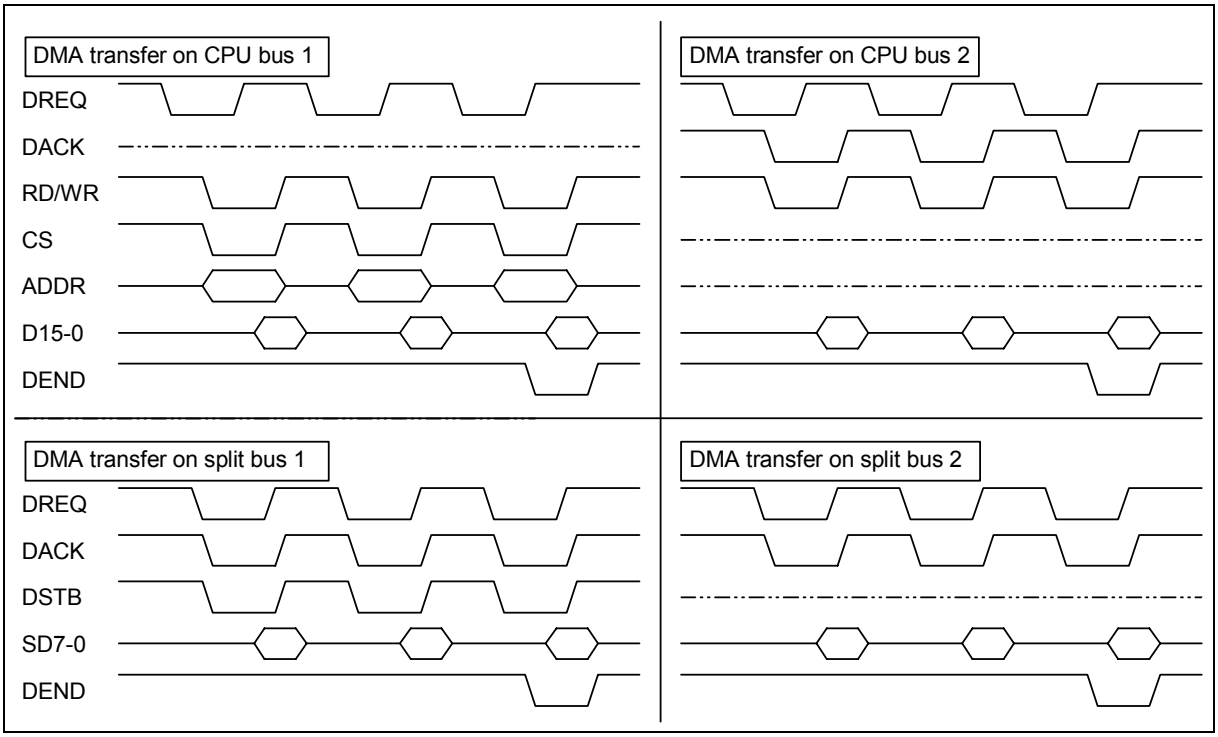


Figure 3.22 DMA control pins operation by FIFO port accessing methods

3.4.3.3 Zero-Length packet addition mode (DxFIFO port writing direction)

With this controller, it is possible to add and send one Zero-Length packet after all of the data has been sent, under the conditions noted below, by setting "1" for the **DEZPM** bit of the **DxFIFOSEL** register. This function can be set only if the buffer memory writing direction has been set (a pipe in the sending direction has been set for the **CURPIPE** bit).

(1) If the number of data bytes written to the buffer memory is a multiple of the integer for the maximum packet size when the **DEND** signal is received

3.4.3.4 DEND pin

The controller is able to terminate DMA transfers that used the **DEND** pin. The **DEND** pin has separate input and output functions, depending on the USB data transfer direction.

(1) Buffer memory reading direction

The **DEND** pin becomes an output pin, making it possible to notify the external DMA controller of the final data transfer. The conditions under which the **DEND** signal is asserted can be set using the **PKTM** bit of **DMAxCFG** register.

Table 3.19 shows the **DEND** pin assertion conditions for the controller.

Table 3.19 DEND pin assertions

Event / PKTM	Transaction count ended	BRDY generated upon reception of packet	Reception of short packet other than Zero-Length packet	Reception of Zero-Length packet when buffer is not empty	Reception of Zero-Length packet when buffer is empty *1)
0	Asserted	Not asserted	Asserted	Asserted	Asserted
1	Asserted	Asserted	Asserted	Asserted	Not asserted

***1)** With reception of a Zero-Length packet when the buffer is empty, the DREQ signal is not asserted.

(1) Buffer memory writing direction

The **DEND** pin becomes the input pin, and data can be sent from the buffer memory (the same situation as when "BVAL=1" is set).

3.4.3.5 DxFIFO auto clear mode (DxFIFO port reading direction)

If "1" is set for the **DCLRM** bit of the **DxFIFOSEL** register, the controller automatically clears the buffer memory of the pertinent pipe when reading of the data from the buffer memory has been completed.

Table 3.20 shows the packet reception and buffer memory clearing processing for each of the various settings.

Using the **DCLRM** bit eliminates the need for the buffer to be cleared by software even if a situation occurs that necessitates clearing of the buffer. This makes it possible to carry out DMA transfers without involvingsoftware. This function can be set only in the buffer memory reading direction.

Table 3.20 Packet reception and buffer memory clearing processing

Register setting / Buffer status when packet is received	DCLRM = "0"		DCLRM = "1"	
	BFRE=0	BFRE=1	BFRE=0	BFRE=1
Buffer full	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Zero-Length packet reception	Needs to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Normal short packet reception	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared
Transaction count ended	Doesn't need to be cleared	Needs to be cleared	Doesn't need to be cleared	Doesn't need to be cleared

3.4.3.6 BRDY interrupt timing selection function

By setting the **BFRE** bit of the **PIPECFG** register, it is possible to keep the **BRDY** interrupt from being generated when a data packet consisting of the maximum packet size is received.

When using DMA transfers, this function can be used to generate an interrupt only when the last data item has been received. The "last data item" refers to the reception of a short packet, or the ending of the transaction counter. When a short packet has been received, the **BRDY** interrupt is generated after the received data has been read. When the **BRDY** interrupt is generated, the length of the data received in the last data packet to have been received can be confirmed.

Table 3.21 shows the timing at which the **BRDY** interrupts are generated by the controller.

Table 3.21 Timing at which BRDY interrupts are generated

Register setting / Buffer state when packet is received	BFRE = "0"	BFRE = "1"
Buffer full	When packet is received	Not generated
Zero-Length packet received	When packet is received	When packet is received
Normal packet received	When packet is received	When reading of the received data from the buffer memory has been completed
Transaction counter ended	When packet is received	When reading of the received data from the buffer memory has been completed

***1)** This function is valid only in the direction of reading from the buffer memory. In the writing direction, the **BFRE** bit should be fixed at "0".

3.4.4 Timing at which the FIFO port can be accessed

3.4.4.1 Timing at which the FIFO port can be accessed when switching pipes

Figure 3.23 shows a diagram of the timing up to the point where the **FRDY** bit and **DTLN** bit are determined when the pipe specified by the FIFO port has been switched (the **CURPIPE** bit of the **C/DxFIFOSEL** register has been changed).

If the **CURPIPE** bit has been changed, access to the FIFO port should be carried out after waiting 450 ns after writing to the **C/DxFIFOSEL** register.

The same timing applies with respect to the CFIFO port, when the **ISEL** bit is changed.

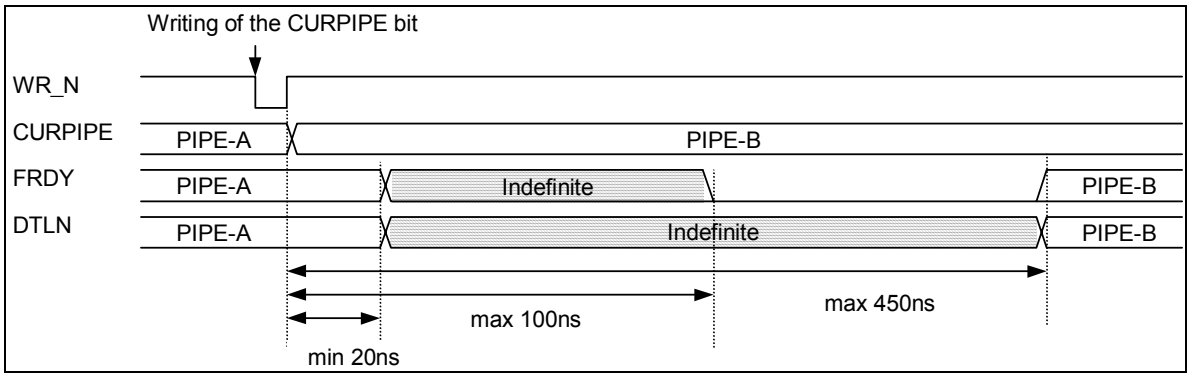


Figure 3.23 Timing at which the FRDY and DTLN bits are determined after changing a pipe

3.4.4.2 Timing at which the FIFO port can be accessed after reading/writing has been completed when using a double buffer

Figure 3.24 shows the timing at which, when using a pipe with a double buffer, the other buffer can be accessed after reading from or writing to one buffer has been completed.

When using a double buffer, access to the FIFO port should be carried out after waiting 300 ns after the access made just prior to toggling.

The same timing applies when a short packet is being sent based on the "BVAL=1" setting using the IN direction pipe.

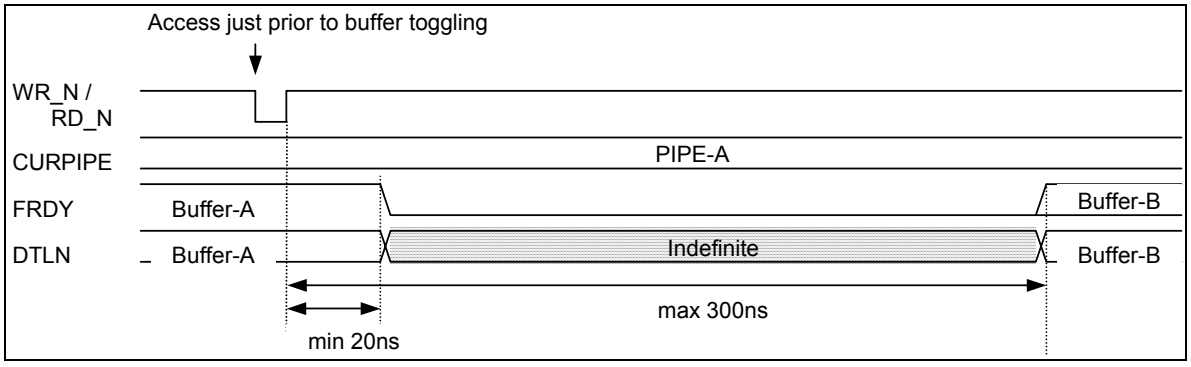


Figure 3.24. Timing at which the FRDY and DTLN bits are determined after reading from or writing to a double buffer has been completed

3.5 Data setup timing

This section describes the **OBUS** bit used to select the timing of split bus.

With this controller, the timing of the SD0-7 and DEND pin can be changed as shown in Table 3.22 , using the **OBUS** bit of the **DMAxCFG** register The **OBUS** bit is a function that is valid only for DMA transfers using a split bus. When using the CPU bus for DMA transfers, the setting of the **OBUS** bit is ignored.

Table 3.22 Differences in operation based on the value set for the OBUS bit

Direction	OBUS bit setting	Operation
Reading	0	The SD0-7 and DEND signals are output on an ongoing basis, regardless of the control signal*1). The next data is output when the control signal is negated. This assures data setup time for the DMAC and enables high-performance DMA transfers.
	1	The SD0-7 and DEND signals are output after the control signal has been asserted. The SD0-7 and DEND signals go to the Hi-Z state when the control signal is negated.
Writing	0	The SD0-7 and DEND signals can be input on an ongoing basis, regardless of the DACKx_N signal. The DMAC can output the next data before the DACKx_N signal is asserted. This assures data setup time for the controller and enables high-performance DMA transfers.
	1	The SD0-7 and DEND signals can be input only if the DACKx_N signal is asserted. The SD0-7 and DEND signals are ignored if the DACKx_N signal is negated.

***1)** *1 "Control signal" refers to the **DACKx_N** signal if the **DFORM**[9-7] of the **DMAxCFG** register is "100". If the **DFORM**[9-7] is "110", it refers to both **DACK0_N** and **DSTRB0_N**. In this case, "assertion of the control signal" means the state in which either **DACK0_N** or **DSTRB0_N** is asserted.

If "OBUS=0" is set in the reading direction, the SD0-7 and DEND signals are output on an ongoing basis, so please be aware that sharing the bus with another device can cause the signals to collide.

If "OBUS=0" is set in the writing direction, the SD0-7 and DEND signals can be input on an ongoing basis, so the user should make sure that the signals are not set to an intermediate potential.

Figure 3.25 shows a schematic diagram of the data setup timing based on the OBUS bit.

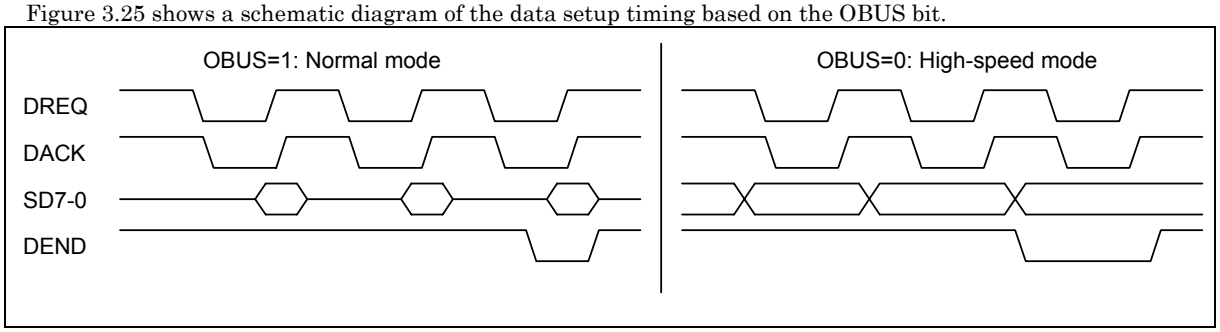


Figure 3.25 Schematic diagram of data setup timing

3.6 Control transfers (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 256-byte single buffer, and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed through the **CFIFO** port.

3.6.1 Setup stage

The controller always sends an ACK response in response to a setup packet that is normal with respect to the controller. The operation of the contoller in the setup stage is noted below.

(1) When a new USB request is received, the controller set following registers:
  (a) Set the **VALID** bit of the **INTSTS0** register to "1".
  (b) Set the **PID** bit of the **DCPCTR** register to "NAK".
  (c) Set the **CCPL** bit of the **DCPCTR** register to "0".
(2) When a data packet is received right after the SETUP packet, the USB request parameters are stored in the **USBREQ**, **USBVAL**, **USBINDX** and **USBLENG** registers.

Response processing with respect to the controller should always be carried out after first setting "VALID=0". In the "VALID=1" state, "PID=BUF" cannot be set, and the data stage cannot be terminated.

Using the function of the **VALID** bit, the controller is able to interrupt the processing of a request currently being processed if a new USB request is received during a control transfer, and can send a response in response to the newest request.

Also, the controller automatically judges the direction bit (bit 8 of the bmRequestType) and the request data length (wLength) of the USB request that was received, and then distinguishes between control read transfers, control write transfers, and no-data control transfers, and controls the stage transition. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the control program is notified. For information on the stage control of the controller, please refer to Figure 3.17, Control transfer stage transitions.

3.6.2 Data stage

Data transfers corresponding to USB requests that have been received should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the **ISEL** bit of the **CFIFOSEL** register.

If the data being transferred is larger than the size of the DCP buffer memory, the data transfer should be carried out using the **BRDY** interrupt for control write transfers and the **BEMP** interrupt for control read transfers.

With control write transfers during Hi-Speed operation, the NYET handshake response is carried out based on the state of the buffer memory. For information on the NYET handshake, please refer to 3.7.1 NYET handshake control.

3.6.3 Status stage

Control transfers are terminated by setting the **CCPL** bit to "1" with the **PID** bit of the **DCPCTR** register set to "PID=BUF".

After the above settings have been entered, the controller automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

(1) For control read transfers:
  The Zero-Length packet is received from the USB host, and the controller sends an ACK response.
(2) For control write transfers and no-data control transfers:
  The controller sends a Zero-Length packet and receives the ACK response from the USB host.

3.6.4 Control transfer auto response function

The controller automatically responds to a normal SET_ADDRESS request. If any of the following errors occur in the SET_ADDRESS request, a response from the S/W is necessary.

(1)	Any transfer other than a control read transfer:	bmRequestType	≠ "0x00"
(2)	If a request error occurs:	wIndex	≠ "0x00"
(3)	For any transfer other than a no-data control transfer:	wLength	≠ "0x00"
(4)	If a request error occurs:	wValue	> "0x7F"
(5)	Control transfer of a device state error:	DVSQ	= "011(Configured)"

For all requests other than the SET_ADDRESS request, a response is required from the correspondingsoftware.

3.7 Bulk transfers (PIPE1-5)

The user can select the buffer memory specifications for bulk transfers (single / double buffer setting, or continuous / non-continuous transfer mode setting). The maximum size that can be set for the buffer memory is 2 KB. The buffer memory state is controlled by the controller, with a response sent automatically for a PING packet / NYET handshake. If "MXPS=0" has been set, the interrupt specifications are different from those of the other pipes. For detailed information, please refer to 3.3.3 Maximum packet size setting.

3.7.1 NYET handshake control

Table 3.23 shows the NYET handshake responses of the controller. The NYET response of the controller is made in conformance with the conditions noted below. When a short packet is received, however, the response will be an ACK response rather than a NYET packet response. The same applies to data stages of control write transfers.

Table 3.23 NYET handshake responses

Value set for **PID** bit	Buffer memory state *1)	Token	Response	Note
NAK/STALL	-	SETUP	ACK	-
	-	IN/OUT/PING	NAK/STALL	-
BUF	-	SETUP	ACK	-
	RCV-BRDY*1	OUT/PING	ACK	If an OUT token is received, a data packet is received.
	RCV-BRDY*2	OUT	NYET	Notification of whether a data packet is received or cannot be received
	RCV-BRDY*2	OUT (Short)	ACK	Notification of whether a data packet is received or cannot be received
	RCV-BRDY*2	PING	ACK	Notification that reception is not possible
	RCV-NRDY	OUT / PING	NAK	Notification that reception is not possible
	TRN-BRDY	IN	DATA0 / 1	Data packet transmission
	TRN-NRDY	IN	NAK	TRN-NRDY

***1)** Buffer memory is state is as following;
RCV-BRDY*1: When an OUT/PING token is received, there is space in the buffer memory for two or more packets.
RCV-BRDY*2: When an OUT token is received, there is only enough space in the buffer memory for one packet.
RCV-NRDY: When a PING token is received, there is no space in the buffer memory.
TRN-BRDY: When an IN token is received, there is data to be sent in the buffer memory.
TRN-NRDY: When an IN token is received, there is no data to be sent in the buffer memory.

3.8 Interrupt transfers(PIPE6-7)

The controller carries out interrupt transfers in accordance with the timing controlled by the host controller. For interrupt transfers, PING packets are ignored (no response is sent), and the ACK, NAK and STALL responses are carried out without an NYET handshake response being made. The controller does not support High-Bandwidth transfers of interrupt transfers.

3.9 Isochronous transfers(PIPE1-2)

The controller is equipped with the following functions pertaining to isochronous transfers.

(1) Notification of isochronous transfer error information
(2) Interval counter (specified by the **IITV** bit)
(3) Isochronous IN transfer data setup control (**IDLY** function)
(4) Isochronous IN transfer buffer flush function (specified by the **IFIS** bit)
(5) SOF pulse output function

The controller does not support the High-Bandwidth transfers of isochronous transfers.

3.9.1 Error detection with isochronous transfers

The controller has a function for detecting the error information noted below, so that when errors occur in isochronous transfers, software can control them. Table 3.24 and Table 3.25 show the order in which errors are confirmed, and the interrupts that are generated.

(1) PID errors
If the PID is illegal
(2) CRC errors and bit stuffing errors
If an error occurs in the CRC of the packet being received, or the bit stuffing is illegal
(3) Maximum packet size exceeded
The maximum packet size exceeded the set value.
(4) Overrun and underrun errors
When using isochronous IN transfers, the data transmission was not in time for the IN token.
When using isochronous OUT transfers, the OUT token was received, but the buffer memory was not empty.
(5) Interval errors
During an isochronous IN transfer, the token could not be received during the interval frame.

Table 3.24 Error detection when a token is received

Detection priority order	Error	Generated interrupt and status
1	PID errors	No interrupt generated (ignored)
2	CRC error and bit stuffing errors	No interrupt generated (ignored)
3	Overrun and underrun errors	**NRDY** interrupt **OVRN** bit set
4	Interval errors	**NRDY** interrupt

Table 3.25 Error detection when a data packet is received

Detection priority order	Error	Generated interrupt and status
1	PID errors	No interrupt generated (ignored)
2	CRC error and bit stuffing errors	**NRDY** interrupt generated **CRCE** bit set
3	Maximum packet size exceeded error	**BEMP** interrupt **PID** set to "STALL"

3.9.2 DATA-PID

Because High-Bandwidth transfers are not supported, the DATA-PID added with the USB 2.0 standard is supported as indicated below.

(1) IN direction:
  (a) DATA0: Sent as data packet PID
  (b) DATA1: Not sent
  (c) DATA2: Not sent
  (d) mDATA: Not sent
(2) OUT direction(when using Full-Speed operation):
  (a) DATA0: Received normally as data packet PID
  (b) DATA1: Received normally as data packet PID
  (c) DATA2: Packet is ignored
  (d) mDATA: Packet is ignored
(3) OUT direction(when using Hi-Speed operation):
  (a) DATA0: Received normally as data packet PID
  (b) DATA1: Received normally as data packet PID
  (c) DATA2: Received normally as data packet PID
  (d) mDATA: Received normally as data packet PID

3.9.3 Interval counter

3.9.3.1 An overview of operation

The isochronous interval can be set using the **IITV** bit of the **PIPEPERI** register. The interval counter enables the functions noted below.

Table 3.26 Function of the interval counter

Transfer Direction	Function	Conditions for detecition
IN	IN buffer flush function	When a token cannot be normally received in the interval frame during an isochronous IN transfer
OUT	Notification of a token not being received	When a token cannot be normally received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is 2^{IITV} (u)frames.

3.9.3.2 Counter initialization

The controller initializes the interval counter under the following conditions.

(1) H/W reset
  The **IITV** bit is initialized.
(2) S/W reset
  The **IITV** bit is initialized.
(3) USB bus reset (the count is stopped, and is then begun once again after USB bus reset.)
  The **IITV** bit is initialized.
(4) Buffer memory initialization using the **ACLRM** bit
  The **IITV** bit is not initialized but count value. The count is begun once again when the software set the **ACLRM** bit to "0")

After the interval counter has been initialized, the counter is started under the following confitions (1) or (2) , when a packet has been transferred normally.

(1) An SOF is received following transmission of data in response to an IN token, in the "PID-BUF" state
(2) An SOF is received after data following an OUT token is received in the "PID=BUF" state

The interval counter is not initialized under the conditions noted below.

(1) The pipe is disabled (the "PID=NAK/STALL" setting temporarily stops the count, and the count resumes upon "PID=BUF")
(2) The USB bus reset or the USB is suspended
  The count stops temporarily and then continues when operation is resumed

3.9.4 Setup of data to be transmitted using isochronous transfer

With isochronous data transmission using this controller, after data has been written to the buffer memory, a data packet can be sent with the next frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

If a double buffer is being used for the buffer memory, transmission will be enabled for only one of the two buffers even after the writing of data to both buffers has been completed, that buffer memory being the one to which the data writing was completed first. For this reason, even if multiple IN tokens are received, the only buffer memory that can be sent is one packet's worth of data.

When an IN token is received, if the buffer memory is in the transmission enabled state, the controller transmits the data . If the buffer memory is not in the transmission enabled state, however, a Zero-Length packet is sent and an underrun error occurs.

Figure 3.26 shows an example of transmission using the isochronous transfer transmission data setup function with the controller, when "IITV=0 (every frame)" has been set. Sending of a Zero-Length packet is displayed in the illustration as "Null", in a shaded box.

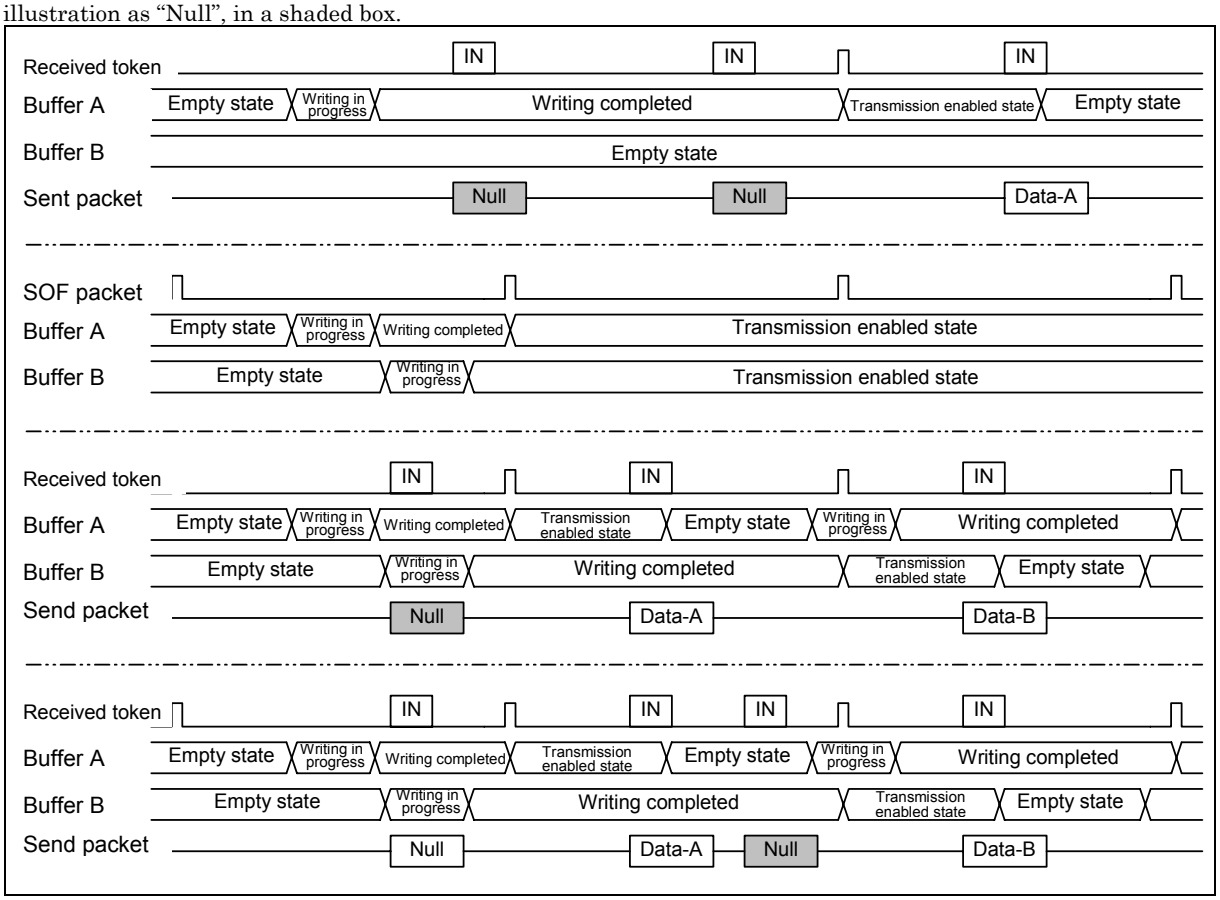


Figure 3.26 Example of data setup function operation

3.9.5 Isochronous transfer transmission buffer flush

If a (u) SOF packet is received without an IN token having been received in the interval frame during isochronous data transmission, the controller operates as a IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer is being used at that time and writing has been finished to both buffers, the buffer memory that was cleared is seen as the data having been sent at the same interval frame, and transmission is enabled for the other buffer memory.

The timing at which the operation of the buffer flush function begins varies depending on the value set for the **IITV** bit.

(1) If IITV=0
The buffer flush operation starts from the next frame after the pipe becomes valid.

(2) In any case other than IITV=0
The buffer flush operation is carried out subsequent to the first normal transaction.

Figure 3.27 shows an example of the buffer flush function of the controller. When an unanticipated token prior to the interval frame is received, the controller sends the written data or a Zero-Length packet accordance with buffer state.

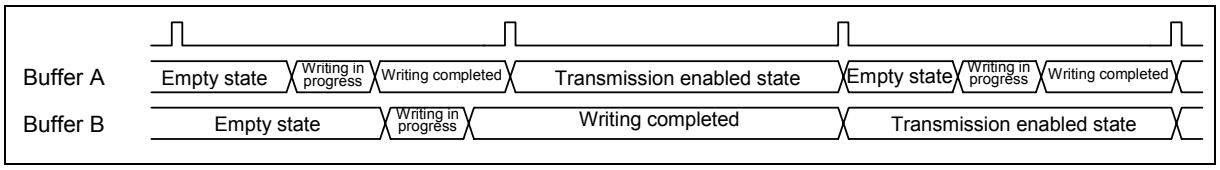


Figure 3.27 Example of buffer flush function operation

Figure 3.28 shows an example of the controller generating an interval error. There are five types of interval errors, as noted below. The interval error is generated at the timing indicated by ① in the illustration, and the IN buffer flush function is activated.

If the interval error occurs during an IN transfer, the buffer flush function is activated, and if it occurs during an OUT transfer an **NRDY** interrupt is generated.

The **OVRN** bit should be used to distinguish between **NRDY** interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the illustration, responses occur based on the buffer memory status.

(1) IN direction:
  (a) If the buffer is in the transmission enabled state, the data is transferred as a normal response.
  (b) If the buffer is in the transmission disabled state, a Zero-Length packet is sent and an underrun error occurs.

(2) OUT direction:
  (a) If the buffer is in the reception enabled state, the data is received as a normal response.
  (b) If the buffer is in the reception disabled state, the data is decarded and an overrun error occurs.

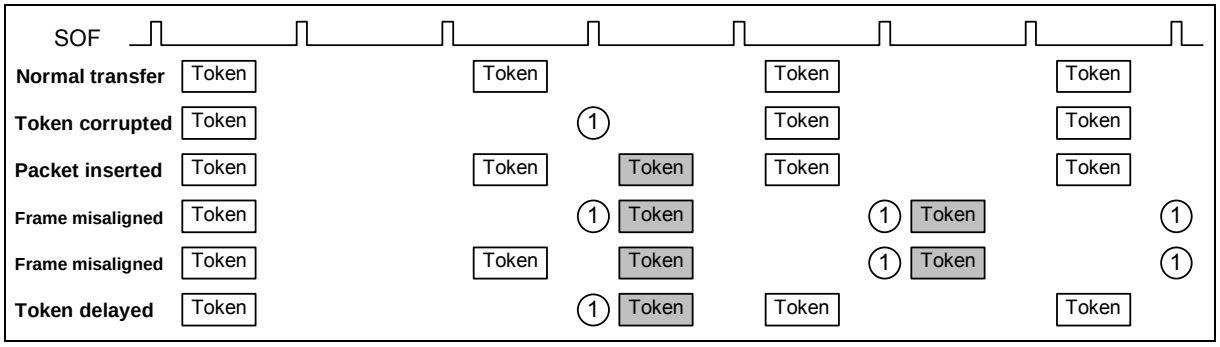


Figure 3.28 Example of interval error being generated when "IITV=1"

3.10 SOF interpolation function

If data could not be received at intervals of 1 ms (when using Full-Speed operation) or 125 µs (when using Hi-Speed operation) because an SOF packet was corrupted or missing, the controller interpolates the SOF. The SOF interpolation operation begins when "USBE=1", "SCKE=1" and an SOF packet is received. The interpolation function is initialized under the following conditions.

(1) H/W reset
(2) S/W reset
(3) USB bus reset
(4) Suspend state detected

Also, the SOF interpolation operates under the following specifications.

(1) 125 µs/1 ms conforms to the results of the reset handshake protocol.
(2) The interpolation function is not activated until an SOF packet is received.
(3) After the first SOF packet is received, either 125 µs or 1 ms is counted at an internal clock of 48 MHz, and interpolation is carried out.
(4) After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
(5) Interpolation is not carried out in the suspended state or while a USB bus reset is being received. (With suspended transitions during Hi-Speed operation, interpolation continues for 3 ms after the last packet is received.

The contorller supports the following functions based on the SOF detection. Those functions also operate normally with SOF interpolation, if the SOF packet was corrupted.

(1) Refreshing of the frame number and micro-frame number
(2) SOFR interrupt timing and uSOF lock
(3) SOF pulse output
(4) Isochronous transfer interval count

If an SOF packet is missing when Full-Speed operation is being used, the **FRNM** bit of the **FRMNUM0** register is not refreshed. If a µSOF packet is missing during Hi-Speed operation, the **UFRNM** bit of the **FRMNUM1** register is refreshed.

However, if a µSOF packet for which "µFRNM=000" is missing, the **FRNM** bit is not refreshed. If this happens, the **FRNM** bit is not refreshed even if successive µSOF packets other than "µFRNM=000" are received normally.

3.10.1 SOF pulse output

When SOF output is enabled, the controller is able to output SOF signals at the timing at which the SOFs are received. When the value of the **SOFM** bit of the **SOFCFG** register is "01" (1 ms SOF) or "10" (125µs SOF), pulses are output from the SOF_N pin in the "L" active state. These are called "SOF signals". For information on pulse timing, please see Figure 3.29. The controller outputs SOF output based on SOF packet reception events or SOF interpolation events at uniform intervals.

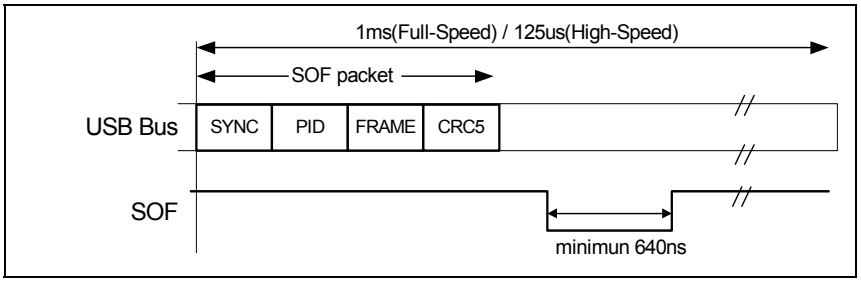


Figure 3.29 SOF output timing

4 Electrical characteristics

4.1 Absolute maximum ratings

Symbol	Item		Rated value	Unit
VDD	Core power supply voltage		-0.3~+2.4	V
VIF	IO power supply voltage		-0.3~+4.0	V
AFEA33V	USB transceiver analog 3.3V supply voltage		-0.3~+4.0	V
AFED33V	USB transceiver digital 3.3V supply voltage		-0.3~+4.0	V
AFEA15V	USB transceiver analog 1.5V supply voltage		-0.3~+2.4	V
AFED15V	USB transceiver digital 1.5V supply voltage		-0.3~+2.4	V
VBUS	VBUS input voltage		-0.3~+5.5	V
V_I (IO)	System interface input voltage		-0.3 ~VIF+0.3	V
V_O (IO)	System interface output voltage		-0.3~VIF+0.3	V
Pd	Power consumption		400	mW
Tstg	Storage temperature	M66592FP (LQFP)	-55~+150	°C
		M66592WG (FBGA)	-55~+125	°C

4.2 Recommended operating conditions

Symbol	Item		Rated value			Unit
			Min.	Typ.	Max.	
VDD	Core power supply voltage		1.35	1.5	1.65	V
VIF	IO power supply voltage	1.8V supported	1.6	1.8	2.0	V
		3.3V supported	2.7	3.3	3.6	V
AFEA33V	USB transceiver analog 3.3V supply voltage		3.0	3.3	3.6	V
AFED33V	USB transceiver digital 3.3V supply voltage		3.0	3.3	3.6	V
AFEA15V	USB transceiver 1.5V analog supply voltage		1.35	1.5	1.65	V
AFED15V	USB transceiver digital 1.5V supply voltage		1.35	1.5	1.65	V
AFEA33G	USB transceiver analog supply GND			0		V
AFED33G	USB transceiver digital supply GND			0		V
AFEA15G	USB transceiver analog supply GND			0		V
AFED15G	USB transceiver digital supply GND			0		V
DGND	Power supply GND			0		V
V_I (IO)	System interface input voltage		0		VIF	V
V_I (VBUS)	Input voltage (VBUS input only)		0		5.25	V
V_O (IO)	System interface output voltage		0		VIF	V
Topr	Ambient operating temperature		-20	+25	+85	°C
tr, tf	Input rise, fall times	Normal input			500	ns
		Schmitt trigger input			5	ms

4.3 Electrical characteristics(ratings for VIF = 2.7~3.6V, VDD = 1.35~1.65V)

Symbol	Item		Measurement conditions		Rated value			Unit
					Min.	Typ.	Max.	
V_{IH}	High input voltage	Xin	AFEA33V = 3.6V		2.52		3.6	V
V_{IL}	Low input voltage		AFEA33V = 3.0V		0		0.9	V
V_{IH}	High input voltage	Note 1	VIF = 3.6V		0.7VIF		3.6	V
V_{IL}	Low input voltage		VIF = 2.7V		0		0.3VIF	V
VT+	Threshold voltage in positive direction	Note 2	VIF = 3.3V		1.4		2.4	V
VT-	Threshold voltage in negative direction				0.5		1.65	V
VTH	Hysteresis voltage					0.8		V
V_{OH}	High output voltage	Xout	AFEA33V = 3.0V	I_{OH} = -50 uA	2.6			V
V_{OL}	Low output voltage			I_{OL} = 50 uA			0.4	V
V_{OH}	High output voltage	Note 3	VIF = 2.7V	I_{OH} = -2 mA	VIF-0.4			V
V_{OL}	Low output voltage			I_{OL} = 2 mA			0.4	V
V_{OH}	High output voltage	Note 4	VIF = 2.7V	I_{OH} = -4 mA	VIF-0.4			V
V_{OL}	Low output voltage			I_{OL} = 4 mA			0.4	V
VT+	Forward direction threshold voltage	Note 5	AFED33V = 3.3V		1.4		2.4	V
VT-	Reverse direction threshold voltage				0.5		1.65	V
I_{IH}	High input current		VIF = 3.6V	V_I = VIF			10	uA
I_{IL}	Low input current			V_I = GND			-10	uA
I_{OZH}	High output current in off status	Note 4	VIF = 3.6V	V_O = VIF			10	uA
I_{OZL}	Low output current in off status			V_O = GND			-10	uA
Rdv	Pull-down resistance	Note 5				500		kΩ
Rdt	Pull-down resistance	Note 6				50		kΩ
Icc (A)	Average supply current at Hi-Speed operation	Note 7	f(Xin) = 48 MHz VDD = 1.65V, VIF = 3.6V, AFEA33V, AFED33V = 3.6V, AFEA15V, AFED15V = 1.65V			40		mA
Icc (A)	Average supply current at Full-Speed operation	Note 7	f(Xin) = 48 MHz VDD = 1.65V, VIF = 3.6V, AFEA33V, AFED33V = 3.6V, AFEA15V, AFED15V = 1.65V			18		mA
Icc (S)	Supply current in static mode	Note 7	USB suspend state VIF = 3.6V			0.27		mA
			USB cable detached VIF = 3.6V			0.07		mA
C_{IN}	Pin capacitance (input)					7		pF
C_{OUT}	Pin capacitance (input / output)	Note 8				7		pF
C_{OUT}	Pin capacitance (D+, D-)					15		pF

Note 1: A6/ALE, A5-1, TEST, MPBUS input pin, and D15-7, D6/AD6-D1/AD1, D0, SD7-0, DEND0-1_N input / output pins
Note 2: CS_N, RD_N, WR0-1_N, DACK0_N, DACK1_N/DSTB0_N, RST_N input pins
Note 3: INT_N, SOF_N, DREQ0-1_N output pins, and DEND0-1_N input / output pin
Note 4: D15-7, D6/AD6-D1/AD1, D0, SD7-SD0 input / output pins
Note 5: VBUS input pin
Note 6: TEST input pin
Note 7: The supply current is the total of the VDD, VIF, AFEA33V, AFED33V, AFEA15V, and AFED15V currents
Note 8: Except D+ and D-

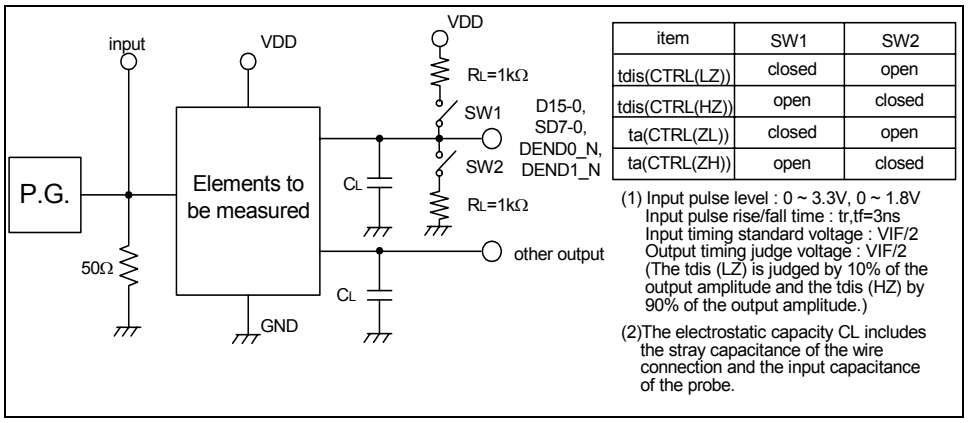
4.4 Electrical characteristics (ratings for VIF = 1.6~2.0V, VDD = 1.35~1.65V)

Symbol	Item		Measurement conditions		Rated value			Unit
					Min.	Standard	Max.	
V_{IH}	High input voltage	Xin	AFEA33V = 3.6V		2.52		3.6	V
V_{IL}	Low input voltage		AFEA33V = 3.0V		0		0.9	V
V_{IH}	High input voltage	Note 1	VIF = 2.0V		0.7VIF		2.0	V
V_{IL}	Low input voltage		VIF = 1.6V		0		0.3VIF	V
VT+	Threshold voltage in positive direction	Note 2	VIF = 1.8V		0.7		1.4	V
VT-	Threshold voltage in negative direction				0.2		0.8	V
VTH	Hysteresis voltage					0.5		V
V_{OH}	High output voltage	Xout	AFEA33V = 3.0V	I_{OH} = -50 uA	2.6			V
V_{OL}	Low output voltage			I_{OL} = 50 uA			0.4	V
V_{OH}	High output voltage	Note 3	VIF = 1.6V	I_{OH} = -2 mA	VIF-0.4			V
V_{OL}	Low output voltage			I_{OL} = 2 mA			0.4	V
V_{OH}	High output voltage	Note 4	VIF = 1.6V	I_{OH} = -4 mA	VIF-0.4			V
V_{OL}	Low output voltage			I_{OL} = 4 mA			0.4	V
VT+	Forward direction threshold voltage	Note 5	AFED33V=3.3V		1.4		2.4	V
VT-	Reverse direction threshold voltage				0.5		1.65	V
I_{IH}	High input current		VIF = 2.0V	V_I= VIF			10	uA
I_{IL}	Low input current			V_I = GND			-10	uA
I_{OZH}	High output current in off status	Note 4	VIF = 2.0V	Vo = VIF			10	uA
I_{OZL}	Low output current in off status			Vo = GND			-10	uA
Rdv	Pull-down resistance	Note 5				500		kΩ
Rdt	Pull-down resistance	Note 6				50		kΩ
Icc(A)	Average supply current at Hi-Speed operation	Note 7	f(Xin) = 48 MHz VDD = 1.65V, VIF = 2.0V, AFEA33V, AFED33V = 3.6V, AFEA15V, AFED15V = 1.65V			40		mA
Icc(A)	Average supply current at Full-Speed operation	Note 7	f(Xin) = 48 MHz VDD = 1.65V, VIF = 2.0V, AFEA33V, AFED33V = 3.6V, AFEA15V, AFED15V = 1.65V			18		mA
Icc(S)	Supply current in static mode	Note 7	USB suspend state VIF = 2.0V			0.27		mA
			USB cable detached VIF = 2.0V			0.07		mA
C_{IN}	Pin capacitance (input)					7		pF
C_{OUT}	Pin capacitance (input / output)	Note 8				7		pF
C_{OUT}	Pin capacitance (D+, D-)					15		pF

Note 1: A6/ALE, A5-1, TEST, MPBUS input pins, and D15-7, D6/AD6-D1/AD1, D0, SD7-0, DEND0-1_N input / output pins
Note 2: CS_N, RD_N, WR0-1_N, DACK0_N, DACK1_N/DSTB0_N, RST_N input pins
Note 3: INT_N, SOF_N, DREQ0-1_N output pin, and DEND0-1_N input / output pin
Note 4: D15-7, D6/AD6-D1/AD1, D0, SD7-SD0 input / output pins
Note 5: VBUS input pin
Note 6: TEST input pin
Note 7: The supply current is the total of the VDD, VIF, AFEA33V, AFED33V, AFEA15V, AFED15V currents.
Note 8: Except D+ and D-

4.5 Measurement circuit

4.5.1 Pins except for USB buffer block

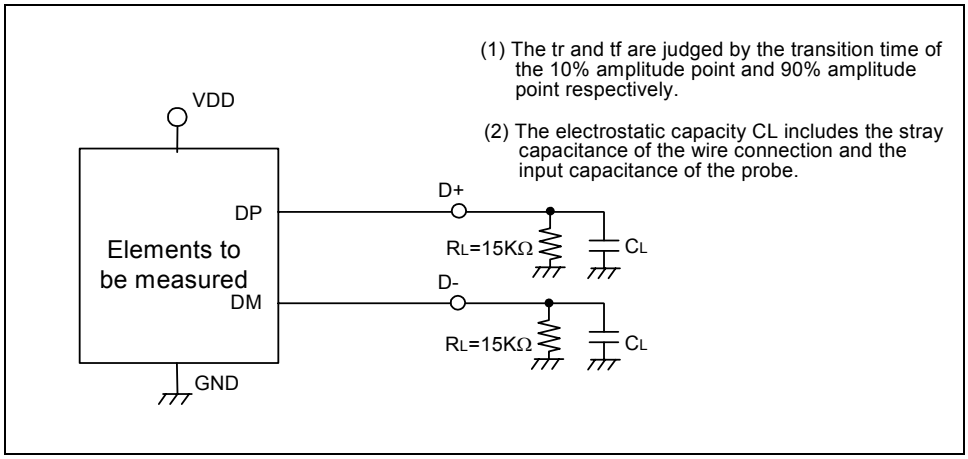


item	SW1	SW2
tdis(CTRL(LZ))	closed	open
tdis(CTRL(HZ))	open	closed
ta(CTRL(ZL))	closed	open
ta(CTRL(ZH))	open	closed

(1) Input pulse level : 0 ~ 3.3V, 0 ~ 1.8V
Input pulse rise/fall time : tr,tf=3ns
Input timing standard voltage : VIF/2
Output timing judge voltage : VIF/2
(The tdis (LZ) is judged by 10% of the output amplitude and the tdis (HZ) by 90% of the output amplitude.)

(2)The electrostatic capacity CL includes the stray capacitance of the wire connection and the input capacitance of the probe.

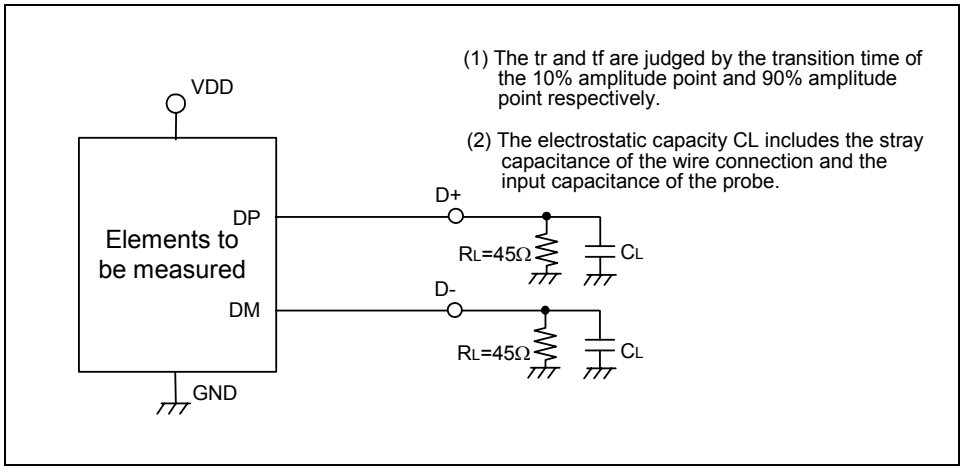
4.5.2 USB buffer block (Full-Speed)



(1) The tr and tf are judged by the transition time of the 10% amplitude point and 90% amplitude point respectively.

(2) The electrostatic capacity CL includes the stray capacitance of the wire connection and the input capacitance of the probe.

4.5.3 USB buffer block (Hi-Speed)



(1) The tr and tf are judged by the transition time of the 10% amplitude point and 90% amplitude point respectively.

(2) The electrostatic capacity CL includes the stray capacitance of the wire connection and the input capacitance of the probe.

4.6 Electrical characteristics (D+/D-)

4.6.1 DC characteristics

Symbol	Item	Measurement conditions		Rated value			Unit
				Min.	Typ.	Max.	
R_{REF}	Reference resistance			5.544	5.6	5.656	kΩ
R_o	FS driver output impedance	HS operation		40.5	45	49.5	Ω
		FS operation		28	36	44	Ω
R_{pu}	D+ pull-up resistance	Idle status		0.9		1.575	kΩ
		Transmitting and receiving status		1.425		3.09	kΩ
Input characteristics for Full-Speed operation							
V_{IH}	High input voltage			2.0			V
V_{IL}	Low input voltage					0.8	V
V_{DI}	Differential input sensitivity	|(D+)-(D-)|		0.2			V
V_{CM}	Differential common mode range			0.8		2.5	V
Output characteristics for Full-Speed operation							
V_{OL}	Low output voltage	AFEAVDD = 3.0V	From 1.5KΩ RL 3.6V			0.3	V
V_{OH}	High output voltage		From 15KΩ RL GND	2.8		3.6	V
V_{SE}	Single-ended receiver threshold voltage			0.8		2.0	V
V_{ORS}	Output signal crossover voltage	CL=50 pF		1.3		2.0	V
Input characteristics for Hi-Speed operation							
V_{HSSQ}	Squelch detection threshold voltage (differential)			100		150	mV
V_{HSCM}	Common mode range			-50		500	mV
Output characteristics for Hi-Speed operation							
V_{HSOI}	Idle state			-10.0		10	mV
V_{HSOH}	High output voltage			360		440	mV
V_{HSOL}	Low output voltage			-10.0		10	mV
V_{CHIRPJ}	Chirp J output voltage (differential)			700		1100	mV
V_{CHIRPK}	Chirp K output voltage (differential)			-900		-500	mV

4.6.2 AC characteristics (Full-Speed)

Symbol	Item	Measurement conditions		Rated value			Unit
				Min.	Standard	Max.	
Tr	Rise time	Data signal: 10% → 90% of amplitude	CL=50 pF	4		20	ns
Tf	Fall time	Data signal: 90% → 10% of amplitude	CL=50 pF	4		20	ns
TRFM	Rise / fall time ratio	tr/tf		90		111.11	%

4.7 Switching characteristics (VIF = 2.7~3.6V, or 1.6~2.0V)

Symbol	Item	Measurement conditions / other	Rated value			Unit	Ref. no.
			Min.	Typ.	Max.		
ta (A)	Address access time	CL=50 pF			40	ns	①
tv (A)	Time that data is valid after address	CL=10 pF	2			ns	②
ta (CTRL - D)	Time that data can be accessed after control	CL=50 pF			30	ns	③
tv (CTRL - D)	Time that data is valid after control	CL=10 pF	2			ns	④
ten (CTRL - D)	Time that data output is enabled after control		2			ns	⑤
tdis (CTRL - D)	Time that data output is disabled after control	CL=50 pF			30	ns	⑥
ta (CTRL - DV)	Time that data can be accessed after control when split bus (DMA Interface) Obus=0	CL=30 pF			30	ns	⑨
tv (CTRL – DV)	Time that data can is valid after control when split bus (DMA Interface) Obus=0	CL=10 pF	2			ns	⑩
ta (CTRL - DendV)	Time that DEND output can be accessed after control when split bus (DMA Interface) Obus=0	CL=30 pF			30	ns	⑪
tv (CTRL - DendV)	Time that DEND output is valid after control when CPU bus and split bus (DMA Interface) Obus=0	CL=10 pF	2			ns	⑫
ta (CTRL - Dend)	Time that DEND output can be accessed after control when split bus (DMA Interface) Obus=1	CL=30 pF			30	ns	⑬
tv (CTRL – Dend)	Time that DEND output is valid after control when CPU bus and split bus (DMA Interface) Obus=1	CL=10 pF	2			ns	⑭
ten (CTRL – Dend)	Time that DEND output is enabled after control when CPU bus and split bus (DMA Interface) Obus=1		2			ns	⑮
tdis (CTRL-Dend)	Time that DEND output is disabled after control when CPU bus and split bus (DMA Interface) Obus=1	CL=30 pF			30	ns	⑯
tdis (CTRL – Dreq)	Time that DREQ is disabled after control				70	ns	⑰
tdis (CTRLH –Dreq)	Time that DREQ is disabled after control				70	ns	⑱
ten (CTRL – Dreq)	Time that DREQ is enabled after control		30			ns	⑲
twh (Dreq)	DREQ output "H" pulse width		20		50	ns	⑳
td (CTRL - INT)	INT output negated delay time				250	ns	㉑
twh (INT)	INT output “H” pulse width		650			ns	㉒
td (DREQ - DV)	Data access after DREQ begins to be asserted when split bus (DMA Interface) Obus=0				0	ns	㉓
td (DREQ - DendV)	Time that DEND can be accessed after DREQ begins to be asserted when split bus (DMA Interface) Obus=0				0	ns	㉔

Key ta: Access time, tv: Valid time, ten: Output enabled time, tdis: Output disabled time, td: propagation delay
(A): Address, (D): Data, (Dend): DEND, (CTRL): Control, (V): Obus=0

4.8 Required timing conditions (VIF = 2.7~3.6V, or 1.6~2.0V)

Symbol	Item		Measurement conditions / other	Rated value			Unit	Ref. no.
				Min.	Typ.	Max.		
tsuw (A)	Address write setup time		CL=50 pF	30			ns	(30)
tsur (A)	Address read setup time			0			ns	(31)
tsu (A - ALE)	Address setup time when using multiplex bus			10			ns	(32)
thw (A)	Address write hold time			0			ns	(33)
thr (A)	Address read hold time			30			ns	(34)
th (A - ALE)	Address setup hold time when using multiplex bus			0			ns	(35)
tw (ALE)	ALE pulse width when using multiplex bus			10			ns	(36)
tdwr (ALE - CTRL)	Write / read delay time when using multiplex bus			7			ns	(37)
trec (ALE)	ALE recovery time when using multiplex bus			0			ns	(38)
tw (CTRL)	Control pulse width (write)			30			ns	(39)
trec (CTRL)	Control recovery time (FIFO)			30			ns	(40)
trecr (CTRL)	Control recovery time (REG)			12			ns	(41)
twr (CTRL)	Control pulse width (read)			30			ns	(42)
tsu (D)	Data setup time			20			ns	(43)
th (D)	Data hold time			0			ns	(44)
tsu (Dend)	DEND input setup time			30			ns	(45)
th (Dend)	DEND input hold time			0			ns	(46)
tw (cycle)	FIFO access cycle time	8-bit FIFO access		30			ns	(47)
		16-bit FIFO access		50			ns	
		8- / a6-bit FIFO access when using multiplex bus		84			ns	
tw (CTRL_B)	Control pulse width when using burst transfers	When using split bus, and Obus=0		12			ns	(48)
		When using split bus, and Obus=1 *1)		30			ns	
		When using DMA transfers with CPU bus		30			ns	
trec (CTRL_B)	Control recovery time for burst transfers			12			ns	(49)
tsud (A)	DMA address write setup time			15			ns	(50)
thd (A)	DMA address write hold time			0			ns	(51)
tw (RST)	Reset pulse width time			100			ns	(52)
tst (RST)	Control starts time after reset			500			ns	(53)

Key tsuw: Write setup time, tsur: Read setup time, tsu: Setup time
thw: Write hold time, thr: Read hold time, th: Hold time, tw: Pulse width, twr: Read pulse width
tdwr: Read / write delay time, trec: Recovery time, trecr: Register recovery time
tsud: DMA setup time, thd: DMA hold time, tst: Start time
(A): Address, (D): Data, (CTRL): Control, (CTRL_B): Burst control, (ALE): ALE

*1) Only for data writing, when the DACK0_N signal is assuring an active period of at least 30 ns, the DSTB0_N signal can be accessed at a minimum of 12 ns.

4.9 Timing diagrams

Table 4.1 Index for register access timing diagram

Bus specification	access	R/W	INDEX	Note
Separate bus	CPU	WRITE	4.9.1.1	CPU bus 0
Separate bus	CPU	READ	4.9.1.2	CPU bus 0
Multiplex bus	CPU	WRITE	4.9.2.1	CPU bus 0
Multiplex bus	CPU	READ	4.9.2.2	CPU bus 0

Table 4.2 Index for FIFO port access

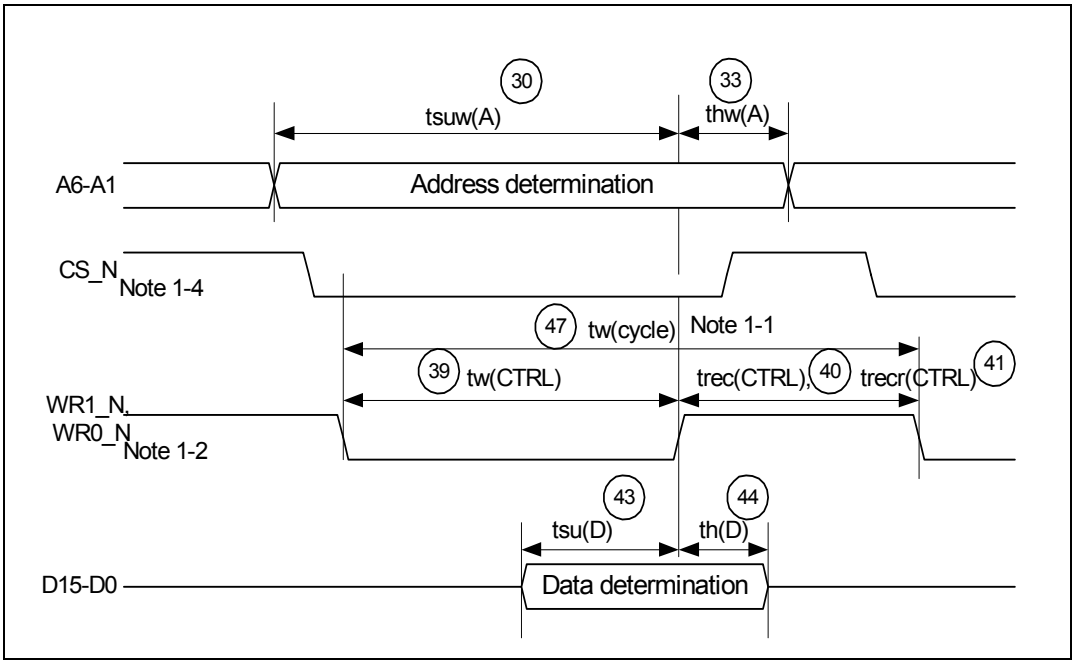
Access	Bus I/F specifications *2)	I/F specifications when operating	DFORM	OBUS	R/W	Note	INDEX
CPU	CPU bus 0	Separate bus	-		WRITE	-	4.9.1.1
CPU	CPU bus 0	Separate bus	-		READ	-	4.9.1.2
CPU	CPU bus 0	Multiplex bus	-		WRITE	-	4.9.2.1
CPU	CPU bus 0	Multiplex bus	-		READ	-	4.9.2.2
DMA	CPU bus 2	ACK+RD/WR	010		WRITE	Cycle steal transfer	4.9.3.1*1)
DMA	CPU bus 2	ACK+RD/WR	010		READ	Cycle steal transfer	4.9.3.2*1)
DMA	SPLIT bus 1	ACK+STB	110	1	WRITE	Cycle steal transfer	4.9.3.3*1)
DMA	SPLIT bus 1	ACK+STB	110	1	READ	Cycle steal transfer	4.9.3.4 *1)
DMA	SPLIT bus 1	ACK+STB	110	0	WRITE	Cycle steal transfer	4.9.3.3 *1)
DMA	SPLIT bus 1	ACK+STB	110	0	READ	Cycle steal transfer	4.9.3.5 *1)
DMA	CPU bus 1	Separate bus	000		WRITE	Cycle steal transfer	4.9.3.6
DMA	CPU bus 1	Separate bus	000		READ	Cycle steal transfer	4.9.3.7
DMA	SPLIT bus 2	ACK only	100	1	WRITE	Cycle steal transfer	4.9.3.8 *1)
DMA	SPLIT bus 2	ACK only	100	1	READ	Cycle steal transfer	4.9.3.9*1)
DMA	SPLIT bus 2	ACK only	100	0	WRITE	Cycle steal transfer	4.9.3.8*1)
DMA	SPLIT bus 2	ACK only	100	0	READ	Cycle steal transfer	4.9.3.10*1)
DMA	CPU bus 3	ACK only	011		WRITE	Cycle steal transfer	4.9.3.11*1)
DMA	CPU bus 3	ACK only	011		READ	Cycle steal transfer	4.9.3.12*1)
DMA	CPU bus 1	Multiplex bus	000		WRITE	Cycle steal transfer	4.9.4.1
DMA	CPU bus 1	Multiplex bus	000		READ	Cycle steal transfer	4.9.4.2
DMA	CPU bus 2	ACK+RD/WR	010		WRITE	Burst transfer	4.9.5.1*1)
DMA	CPU bus 2	ACK+RD/WR	010		READ	Burst transfer	4.9.5.2*1)
DMA	SPLIT bus 1	ACK+STB	110	1	WRITE	Burst transfer	4.9.5.3*1)
DMA	SPLIT bus 1	ACK+STB	110	1	READ	Burst transfer	4.9.5.4*1)
DMA	SPLIT bus 1	ACK+STB	110	0	WRITE	Burst transfer	4.9.5.3*1)
DMA	SPLIT bus 1	ACK+STB	110	0	READ	Burst transfer	4.9.5.5*1)
DMA	CPU bus 1	Separate bus	000		WRITE	Burst transfer	4.9.5.6
DMA	CPU bus 1	Separate bus	000		READ	Burst transfer	4.9.5.7
DMA	SPLIT bus 2	ACK only	100	1	WRITE	Burst transfer	4.9.5.8*1)
DMA	SPLIT bus 2	ACK only	100	1	READ	Burst transfer	4.9.5.9*1)
DMA	SPLIT bus 2	ACK only	100	0	WRITE	Burst transfer	4.9.5.8*1)
DMA	SPLIT bus 2	ACK only	100	0	READ	Burst transfer	4.9.5.10*1)
DMA	CPU bus 3	ACK only	011		WRITE	Burst transfer	4.9.5.11*1)
DMA	CPU bus 3	ACK only	011		READ	Burst transfer	4.9.5.12*1)
DMA	CPU bus 1	Multiplex bus	000		WRITE	Burst transfer	4.9.6.1
DMA	CPU bus 1	Multiplex bus	000		READ	Burst transfer	4.9.6.2

***1)** Because the address signal is not used, the timing will be the same for the separate bus and multiplex bus.
***2)** For the bus I/F specifications, please refer to 3.4.4.2, DMA control signal selection.

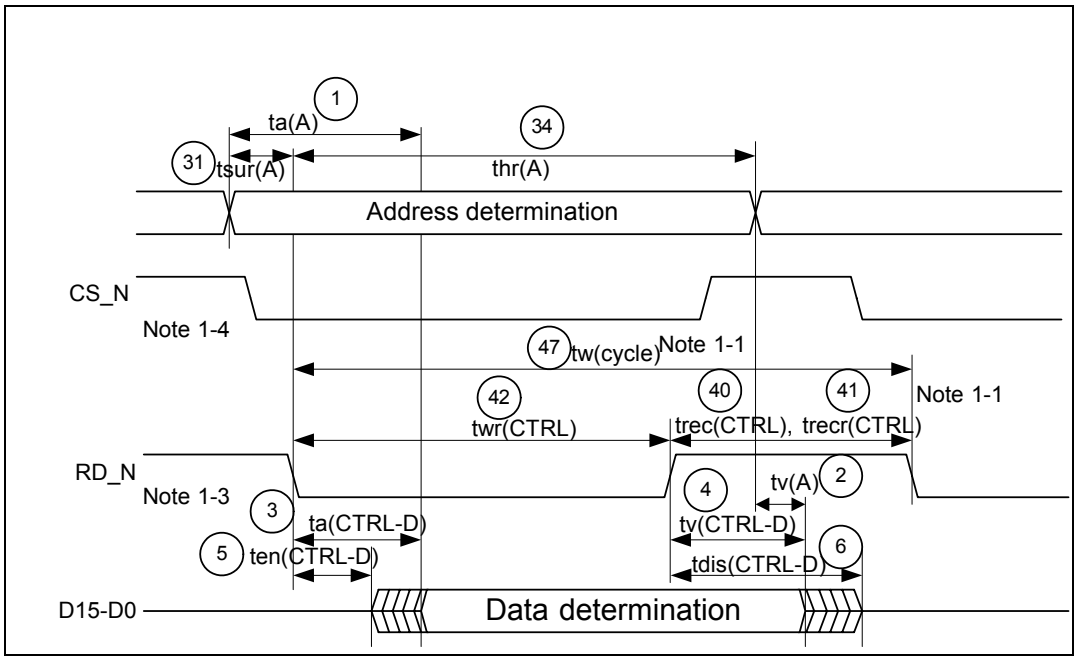
► The reading and writing timing are carried out using control signal. If the control signal is configured of a combination of multiple signals, the ratings from the falling edge will be valid starting from when the active delay signal changes.
The ratings from the rising edge will be valid starting from the change in signals that become inactive more quickly.

4.9.1 CPU access timing(when a separate bus is set)

4.9.1.1 CPU access write timing (when a separate bus is set)



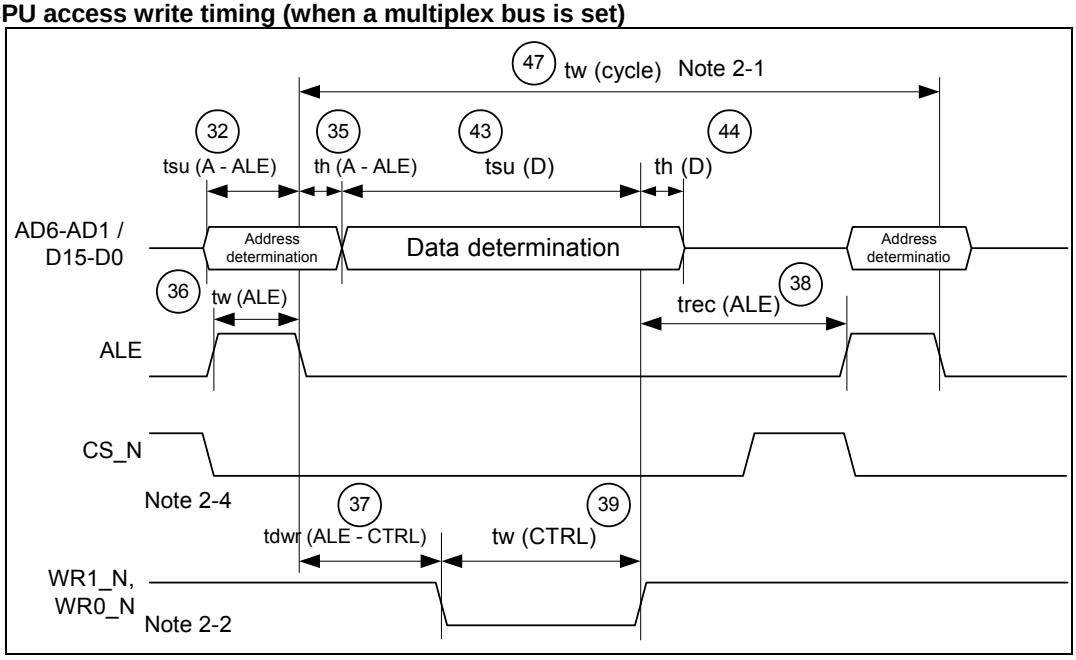
4.9.1.2 CPU access read timing (when a separate bus is set)



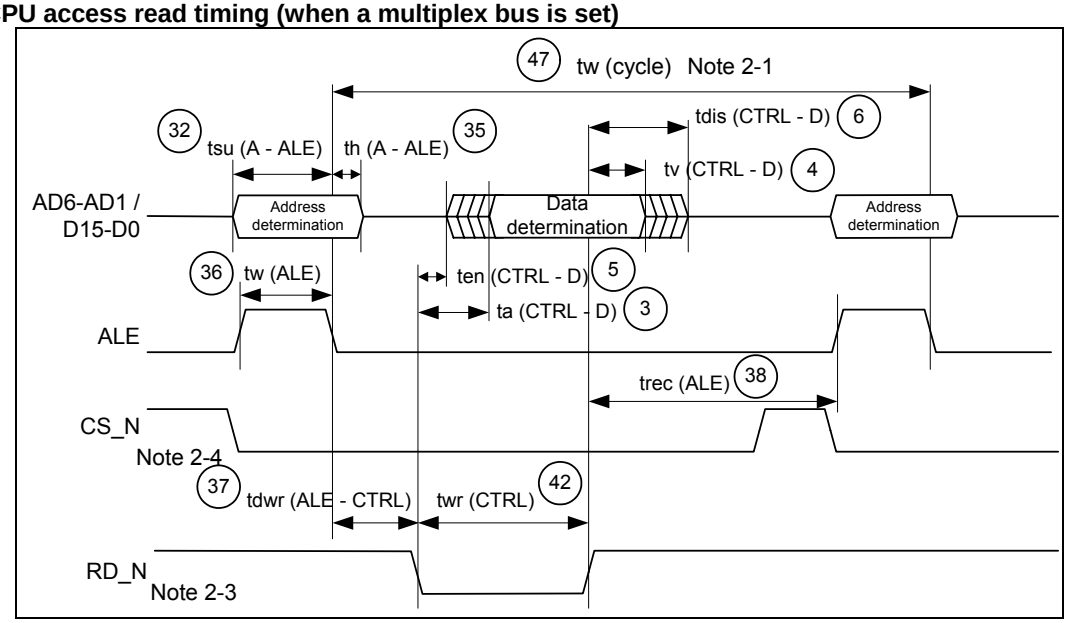
Note 1-1: tw (cycle) and trec (CTRL) are necessary when accessing the FIFO.
Note 1-2: The control signal when writing data is a combination of CS_N, WR1_N, and WR0_N.
Note 1-3: The control signal when reading data is a combination of CS_N and RD_N.
Note 1-4: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that CS_N is rising. Similarly, CS_N should not be timed to fall at the same timing that WR0_N and WR1_N are rising. In the instances noted above, an interval of at least 10 ns must be left open.

4.9.2 CPU access timing (when a multiplex bus is set)

4.9.2.1 CPU access write timing (when a multiplex bus is set)



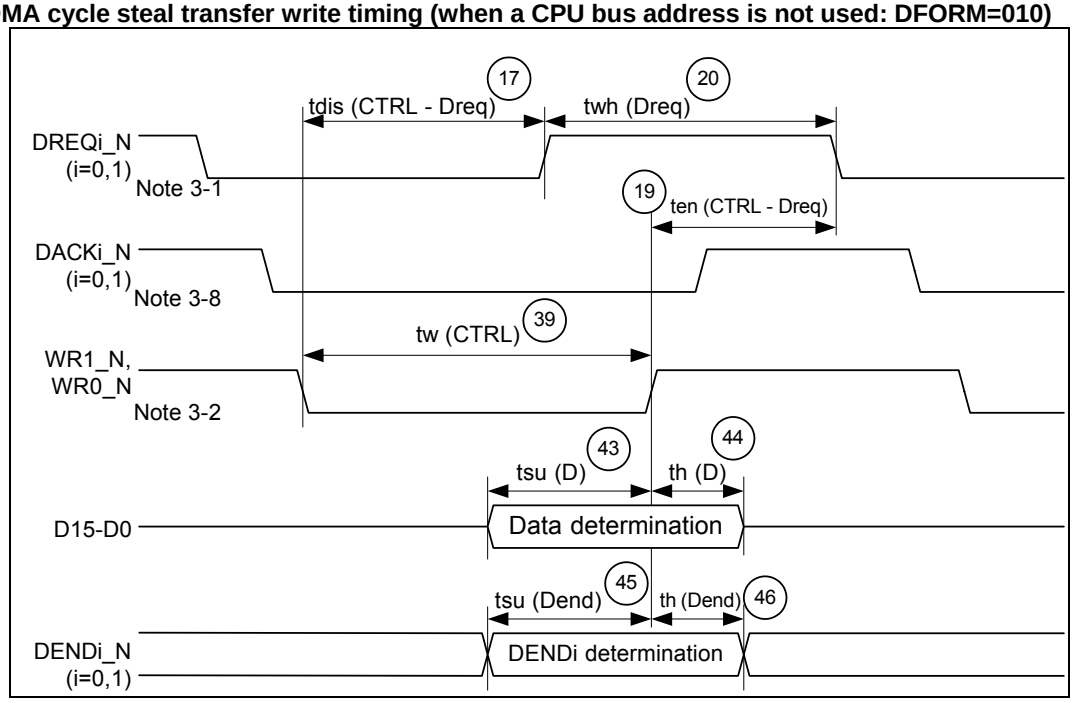
4.9.2.2 CPU access read timing (when a multiplex bus is set)



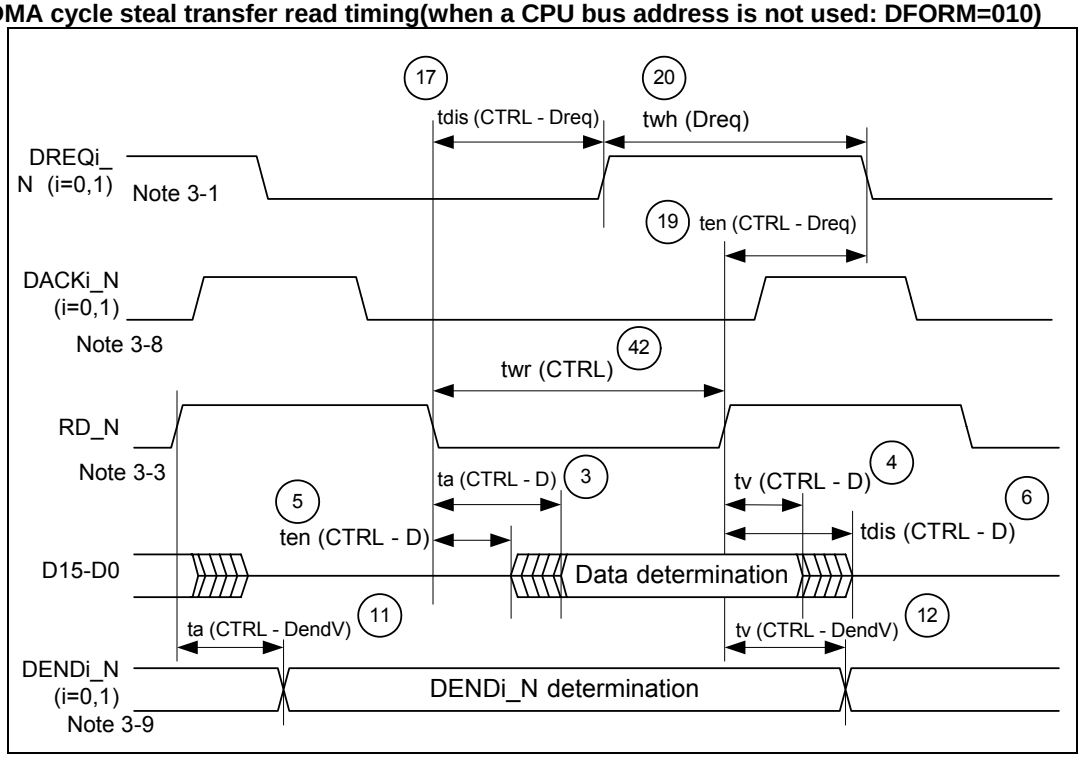
Note 2-1: tw (cycle) and trec (CTRL) are necessary when accessing the FIFO.
Note 2-2: The control signal when writing data is a combination of CS_N, WR1_N, and WR0_N.
Note 2-3: The control signal when reading data is a combination of CS_N and RD_N.
Note 2-4: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that CS_N is rising. Similarly, CS_N should not be timed to fall at the same timing that RD_N, WR0_N and WR1_N are rising. In the instances noted above, an interval of at least 10 ns must be left open.

4.9.3 DMA access timing(when a cycle steal transfer and separate bus are set)

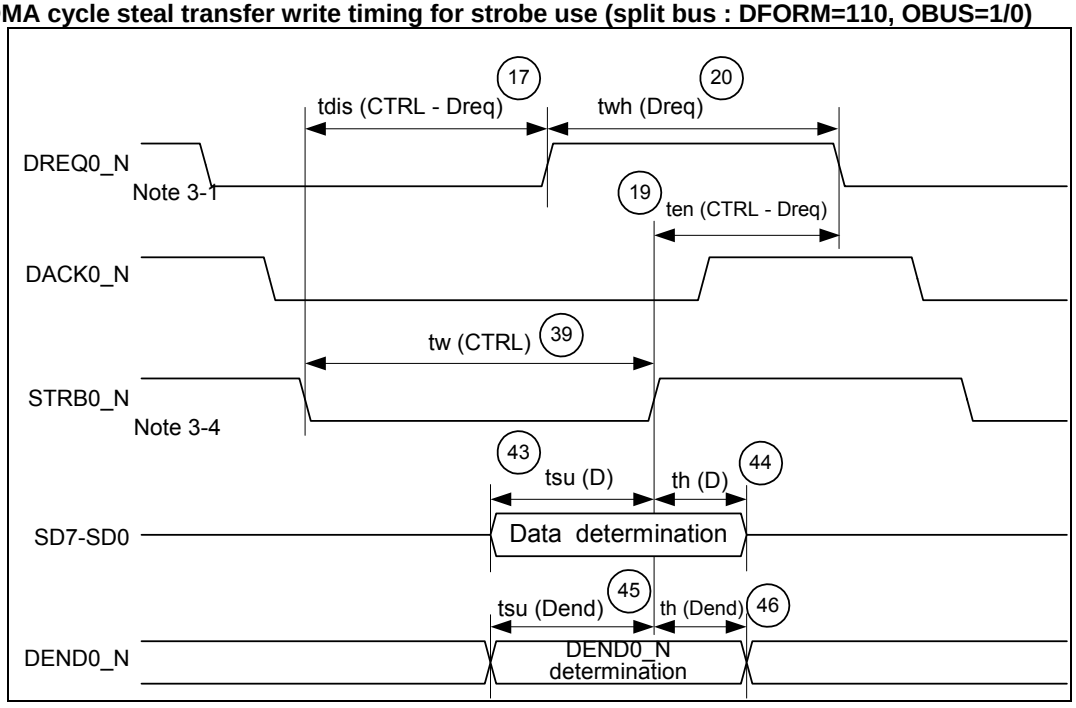
4.9.3.1 DMA cycle steal transfer write timing (when a CPU bus address is not used: DFORM=010)



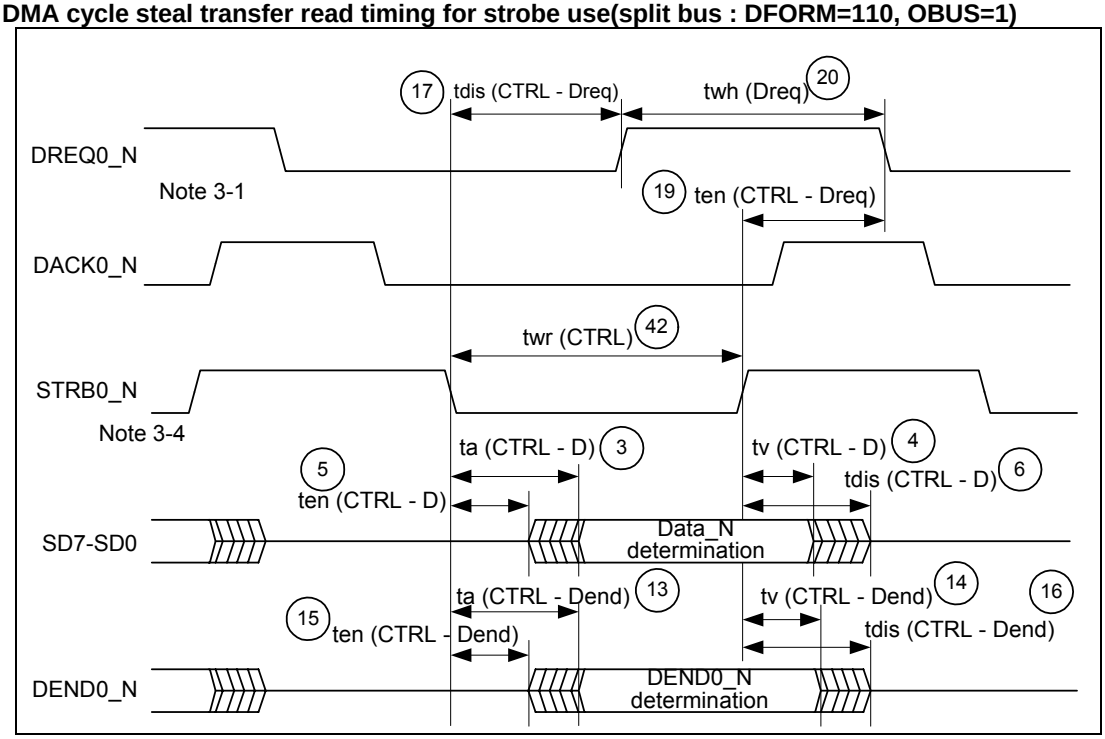
4.9.3.2 DMA cycle steal transfer read timing(when a CPU bus address is not used: DFORM=010)



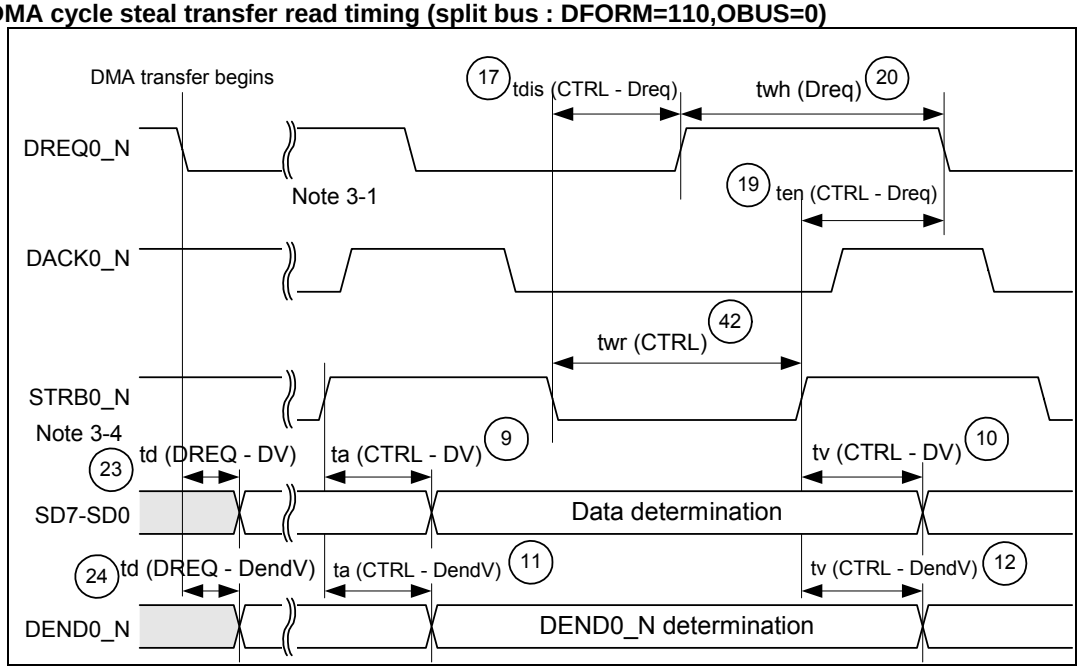
4.9.3.3 DMA cycle steal transfer write timing for strobe use (split bus : DFORM=110, OBUS=1/0)



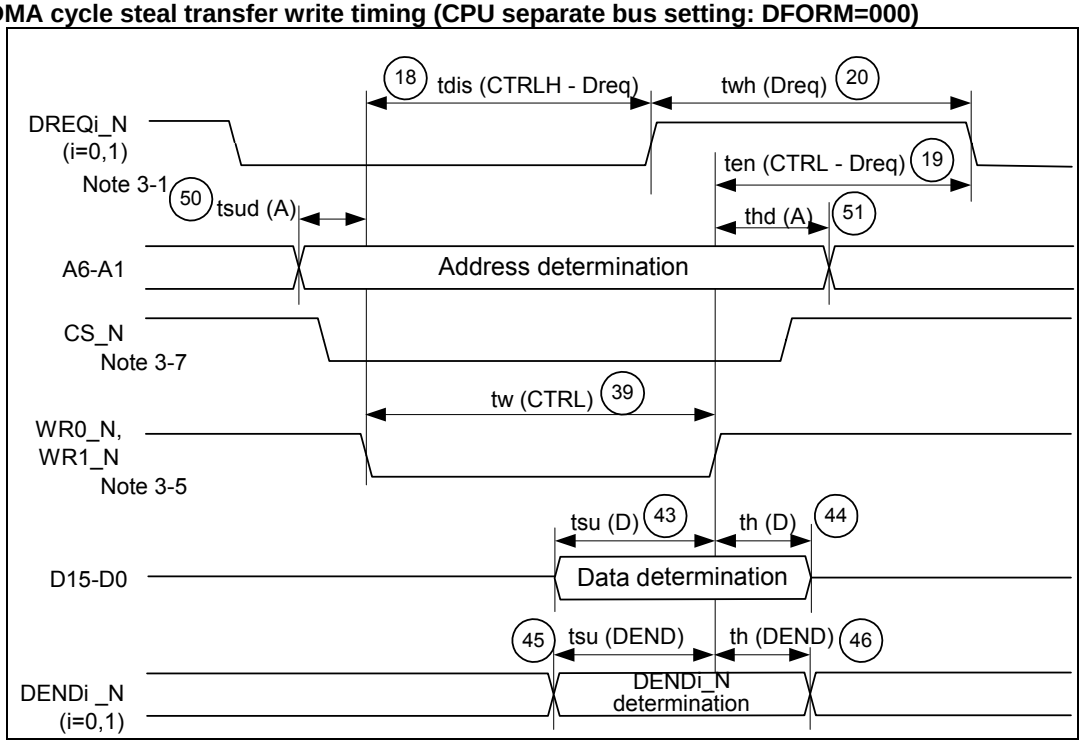
4.9.3.4 DMA cycle steal transfer read timing for strobe use(split bus : DFORM=110, OBUS=1)



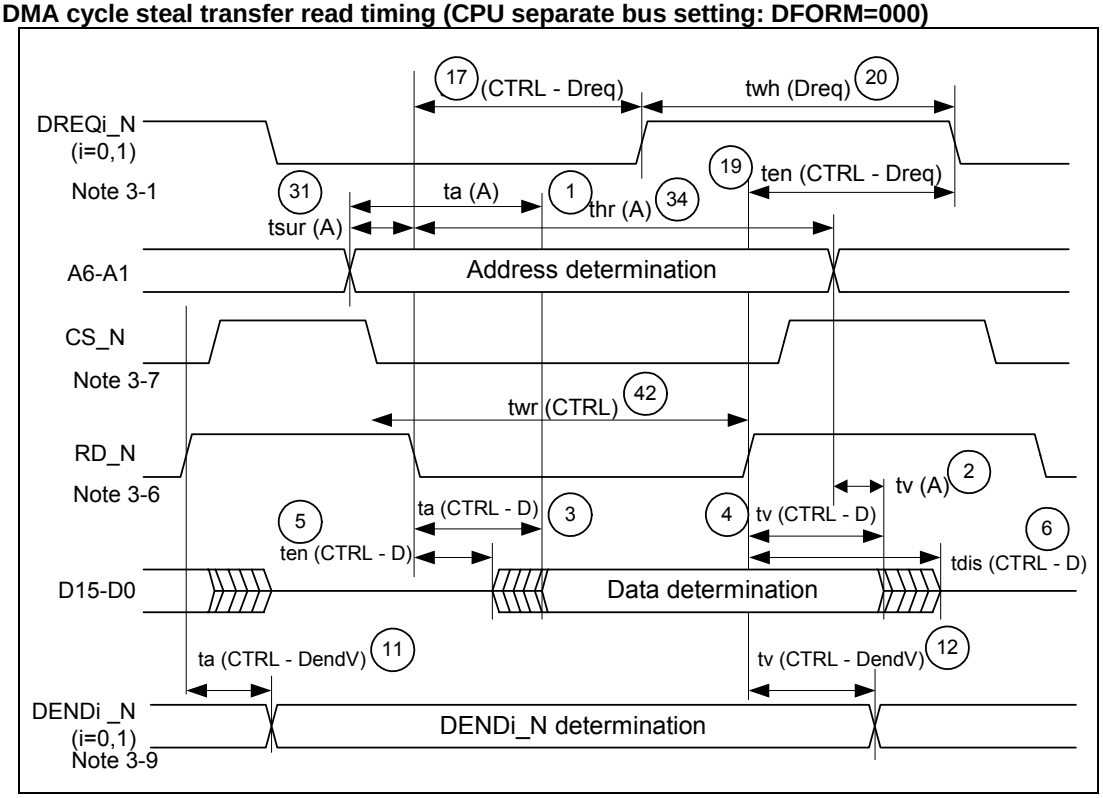
4.9.3.5 DMA cycle steal transfer read timing (split bus : DFORM=110,OBUS=0)



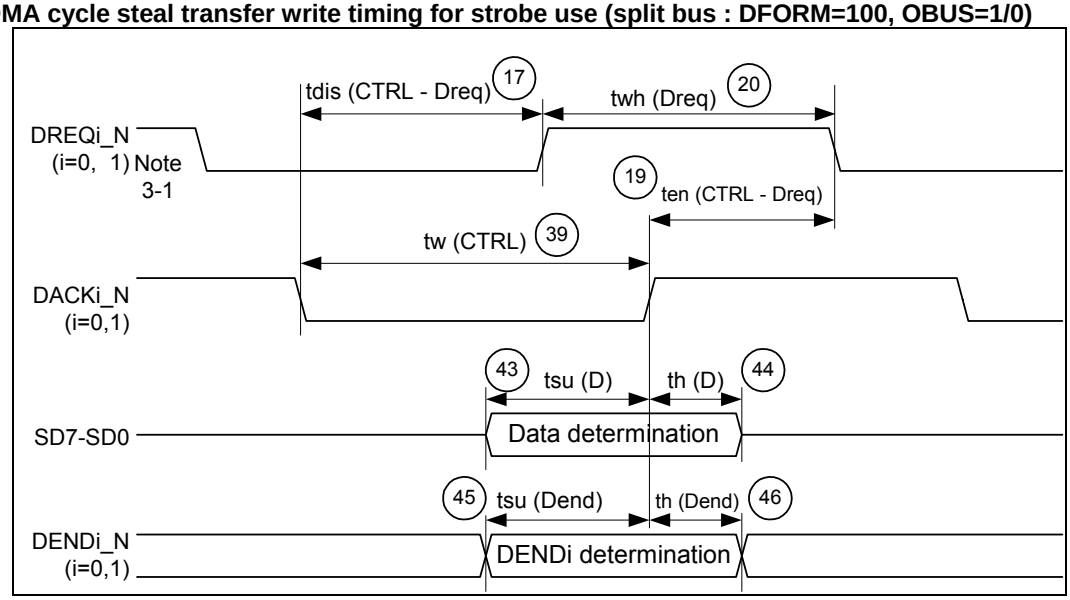
4.9.3.6 DMA cycle steal transfer write timing (CPU separate bus setting: DFORM=000)



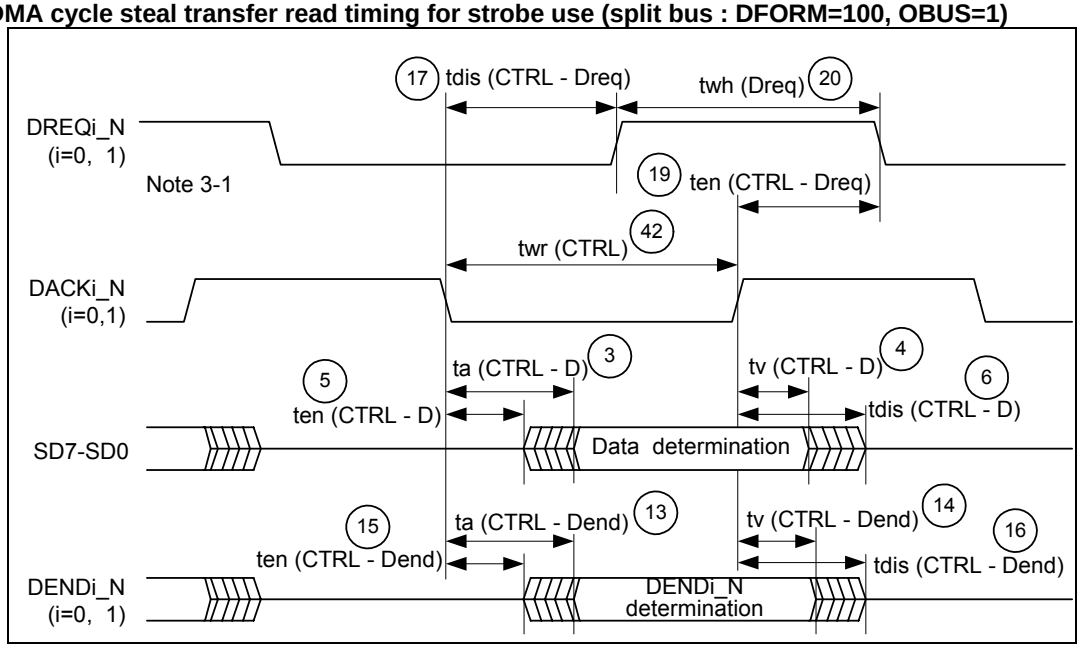
4.9.3.7 DMA cycle steal transfer read timing (CPU separate bus setting: DFORM=000)



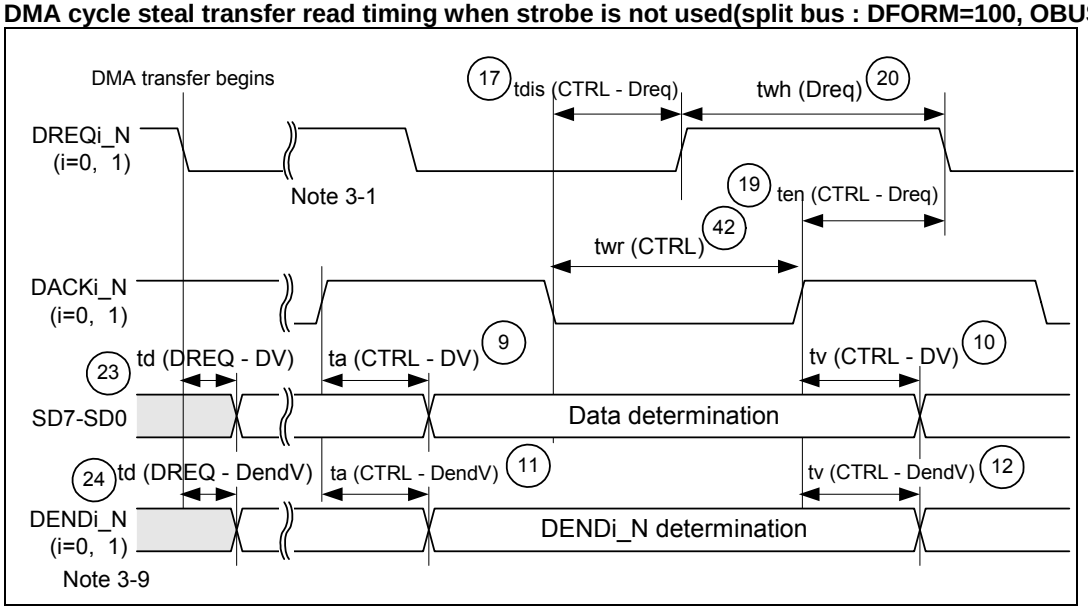
4.9.3.8 DMA cycle steal transfer write timing for strobe use (split bus : DFORM=100, OBUS=1/0)



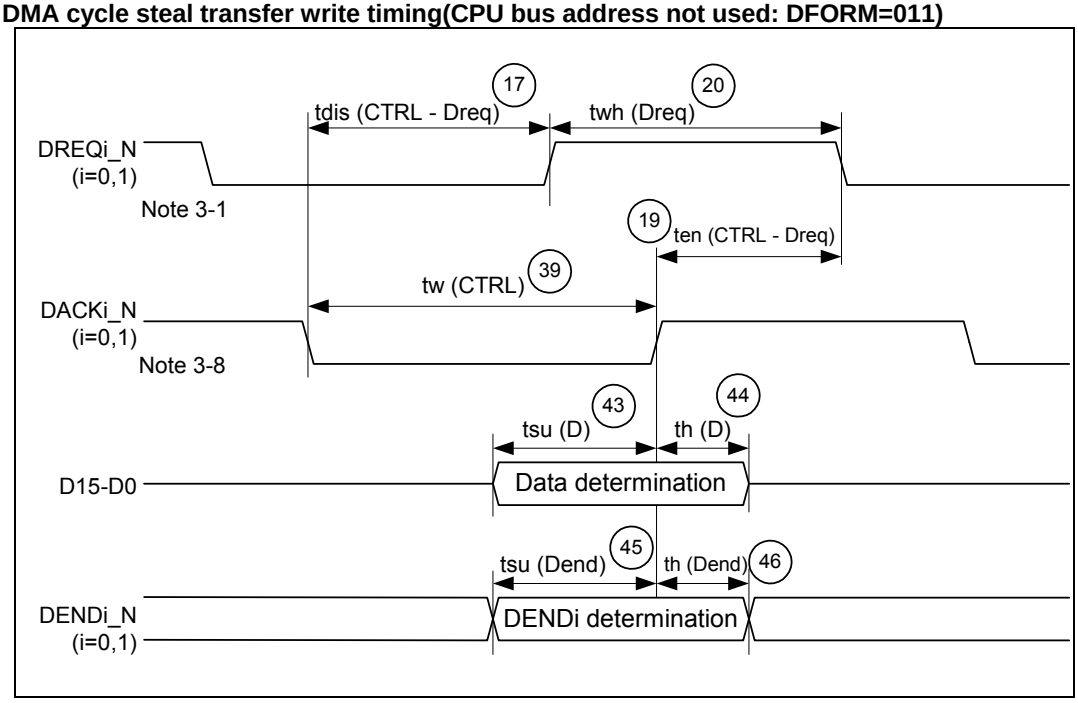
4.9.3.9 DMA cycle steal transfer read timing for strobe use (split bus : DFORM=100, OBUS=1)



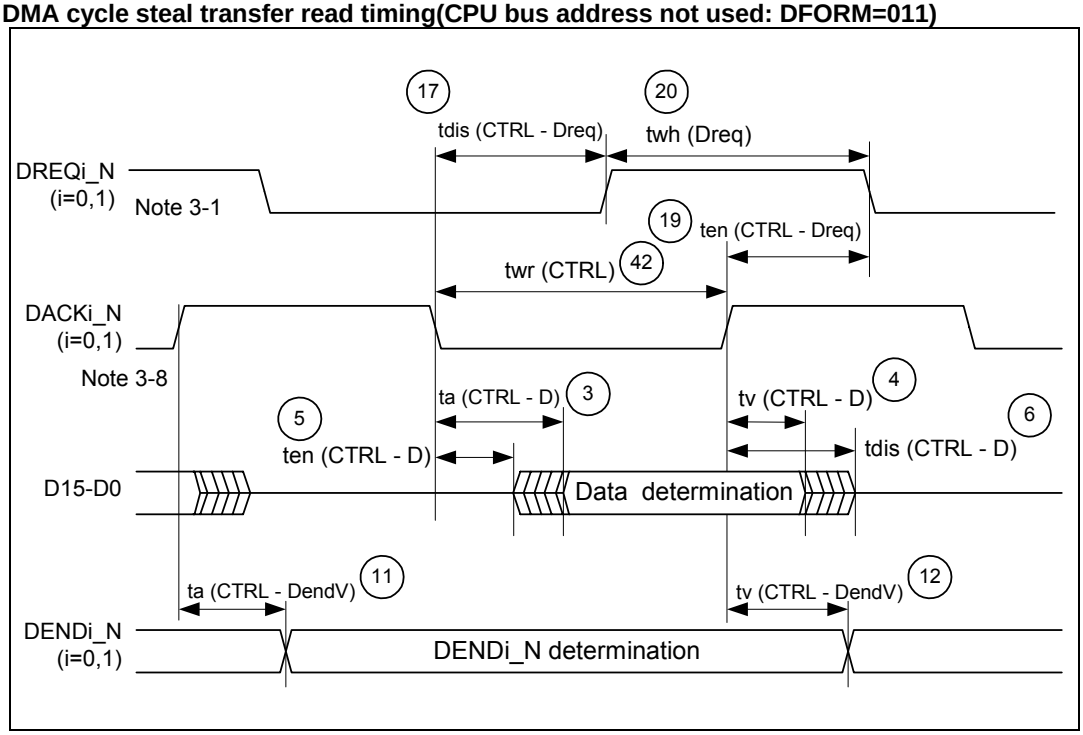
4.9.3.10 DMA cycle steal transfer read timing when strobe is not used(split bus : DFORM=100, OBUS=0)



4.9.3.11 DMA cycle steal transfer write timing(CPU bus address not used: DFORM=011)



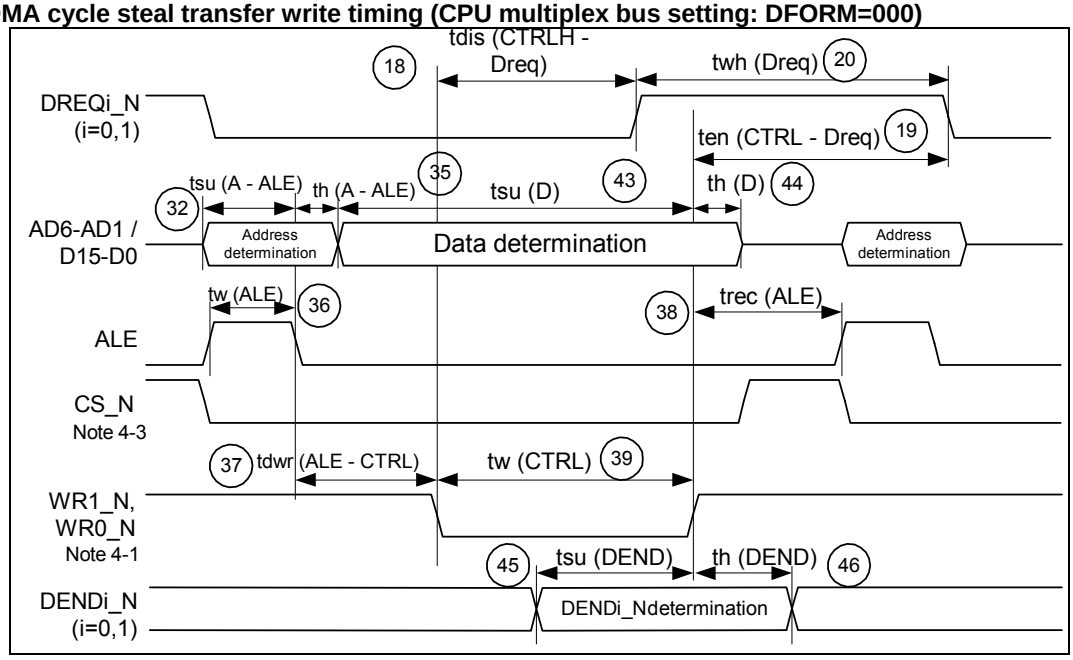
4.9.3.12 DMA cycle steal transfer read timing(CPU bus address not used: DFORM=011)



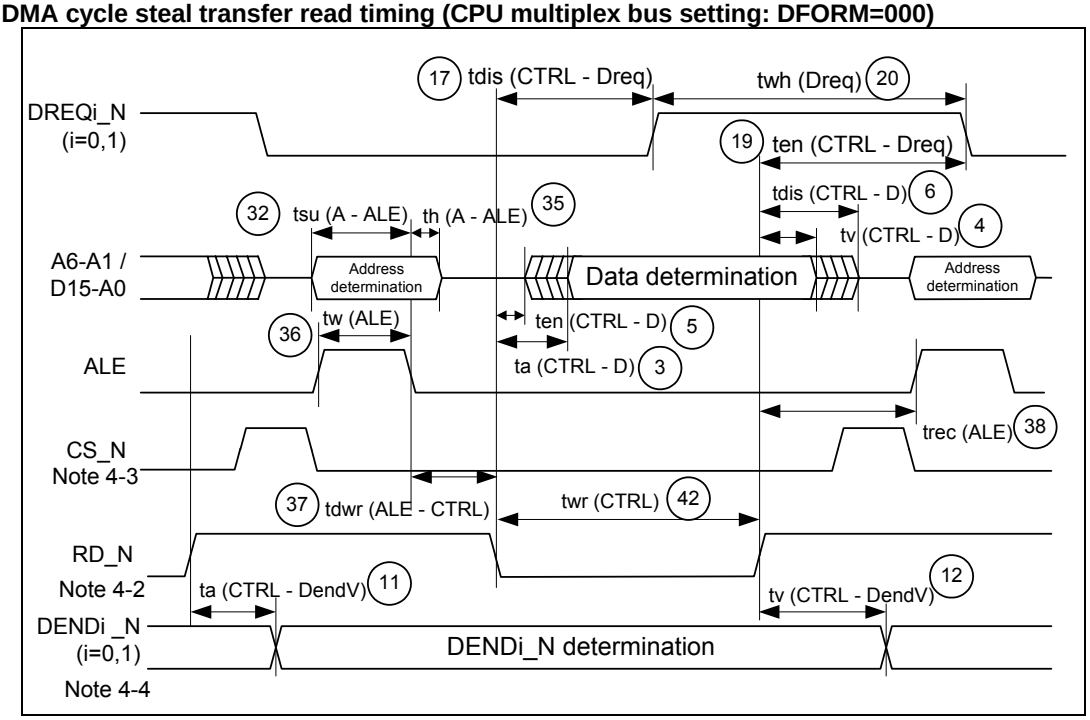
Note 3-1: The inactive condition for DREQi_N (i=0, 1) is the control signal. If there is a next DMA transfer, the delay ratings for twh (Dreq) and ten (CTRL-Dreq) will be valid for the time until DREQi_N becomes active is twh (Dreq).
Note 3-2: The control signal when writing data is a combination of DACKi_N, WR1_N, and WR0_N.
Note 3-3: The control signal when reading data is a combination of DACKi_N and RD_N.
Note 3-4: The control signal when writing data is a combination of DACK0 and DSTRB0_N.
Note 3-5: The control signal when writing data is a combination of CS_N, WR0_N and WR1_N.
Note 3-6: The control signal when reading data is a combination of CS_N and RD_N.
Note 3-7: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that CS_N is rising. Similarly, CS_N should not be timed to fall at the same timing that RD_N or WR0_N and WR1_N are rising. In the instances noted above, an interval of at least 10 ns must be left open.
Note 3-8: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that DACKi_N is rising (or falling). Similarly, DACK should not be timed to fall (or rise) at the same timing that RD_N or WR0_N and WR1_N are rising. In the instances noted above, an interval of at least 10 ns must be left open.
Note 3-9: When the receipt data is one byte, the data determined time is "(23)td(DREQ-DV)" and the DEND determined time is "(24)td(DREQ-DendV)".

4.9.4 DMA access timing(cycle steal transfer, when a multiplex bus is set)

4.9.4.1 DMA cycle steal transfer write timing (CPU multiplex bus setting: DFORM=000)



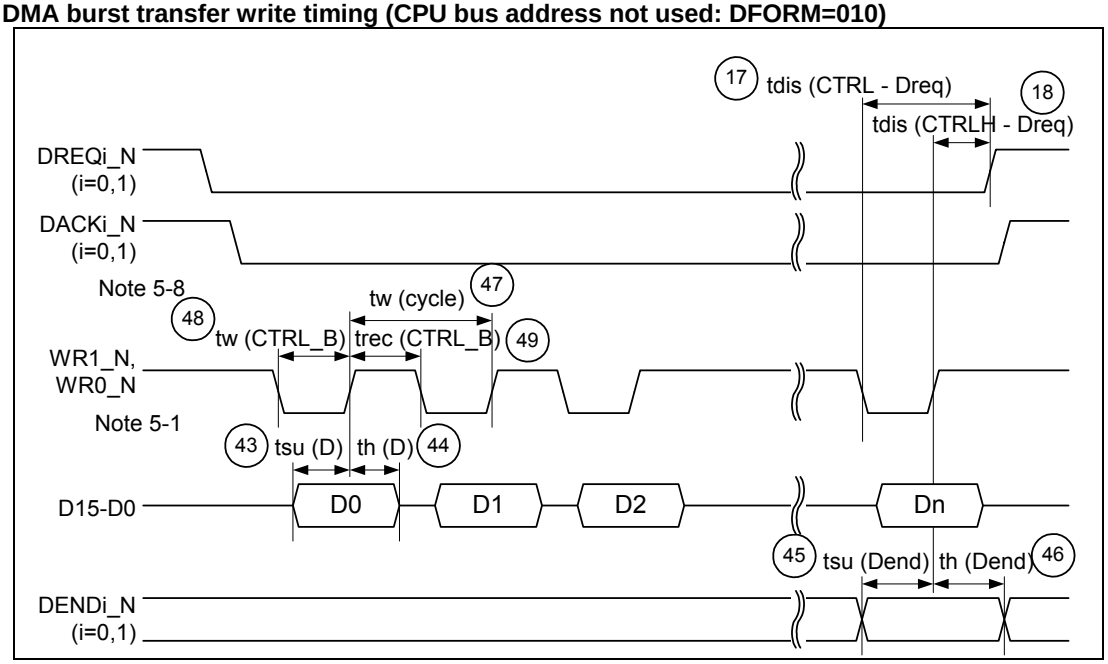
4.9.4.2 DMA cycle steal transfer read timing (CPU multiplex bus setting: DFORM=000)



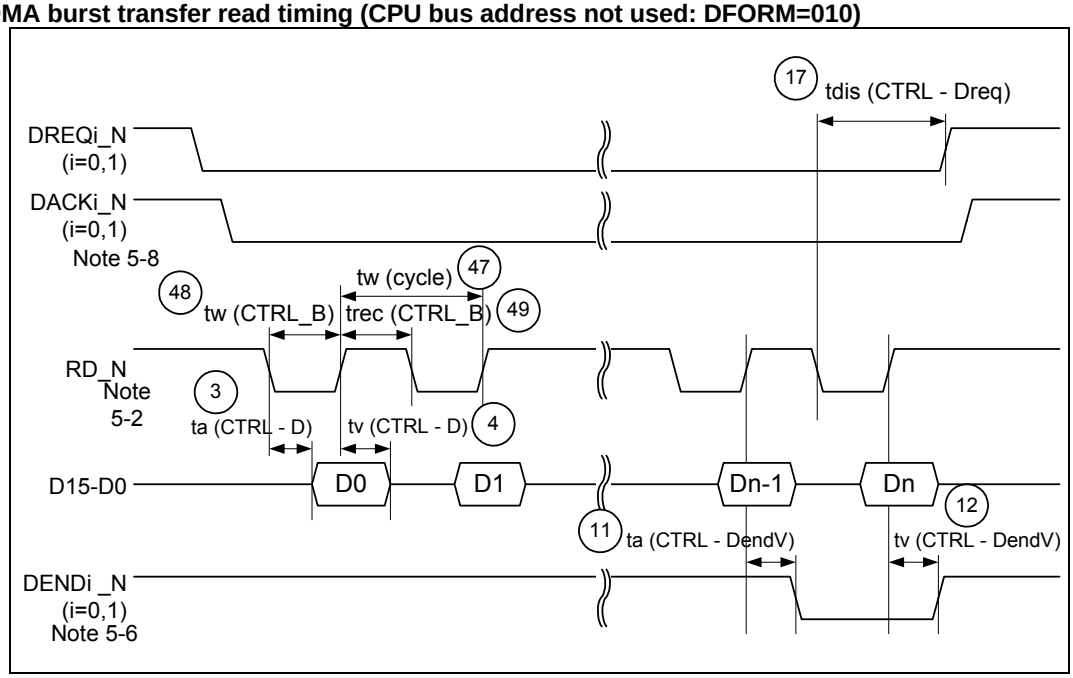
Note 4-1: The control signal when writing data is a combination of CS_N, WR0_N, and WR1_N.
Note 4-2: The control signal when reading data is a combination of CS_N and RD_N.
Note 4-3: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that CS_N is rising. Similarly, CS_N should not be timed to fall (or rise) at the same timing that RD_N or WR0_N and WR1_N are rising. In the instances noted above, an interval of at least 10 ns must be left open.
Note 4-4: When the receipt data is one byte, the DEND determined time is "(24)td(DREQ-DendV)".

4.9.5 DMA access timing (when burst transfer and separate bus are set)

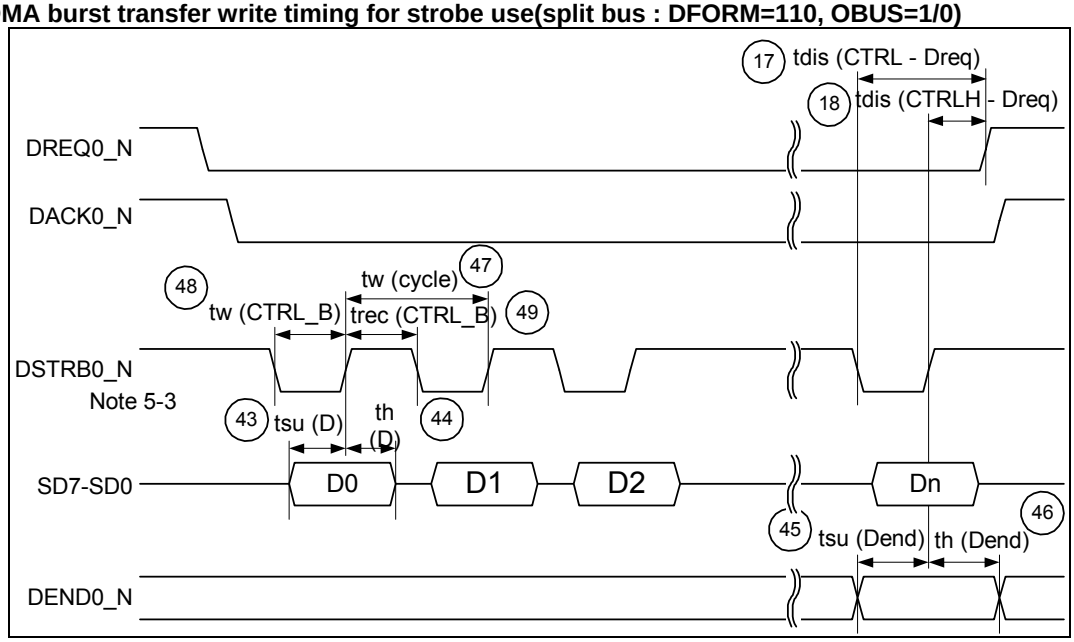
4.9.5.1 DMA burst transfer write timing (CPU bus address not used: DFORM=010)



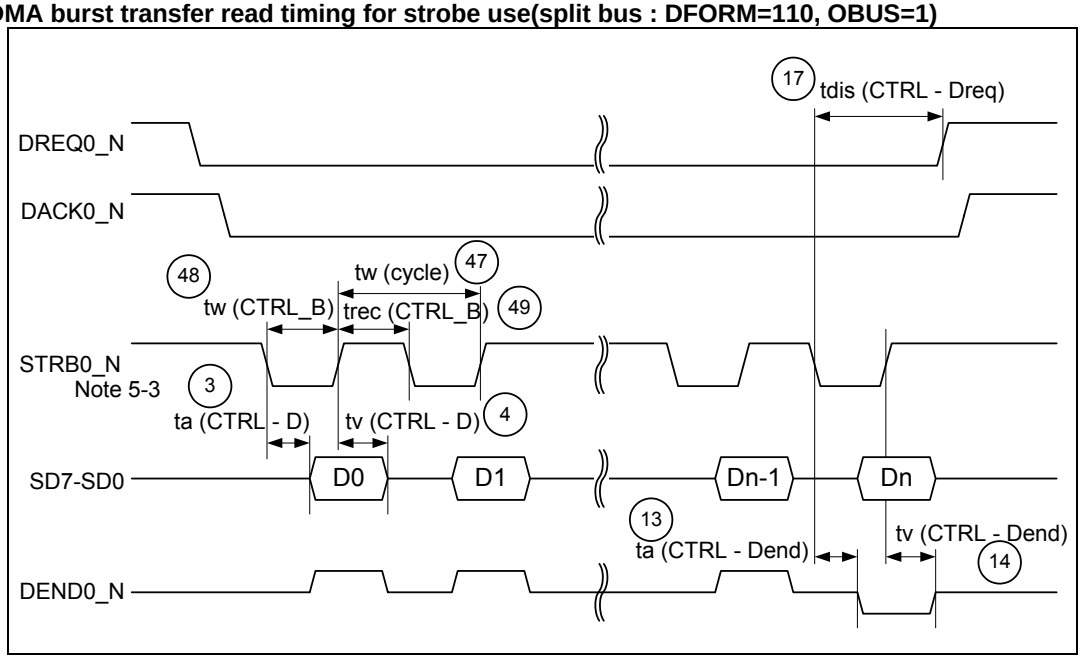
4.9.5.2 DMA burst transfer read timing (CPU bus address not used: DFORM=010)



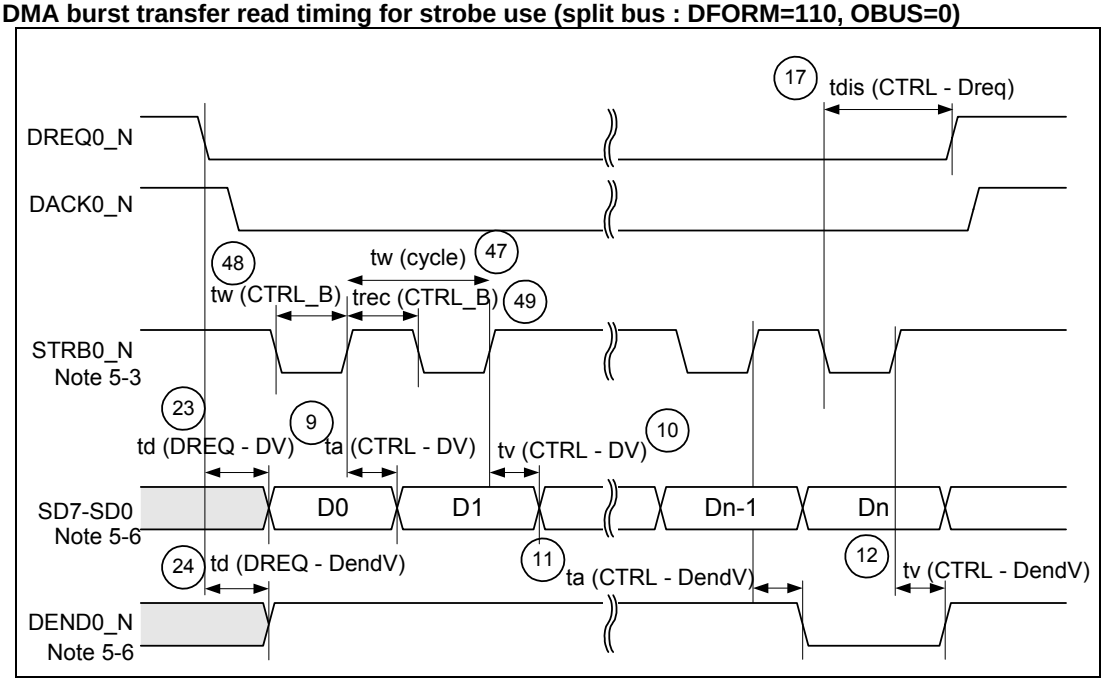
4.9.5.3 DMA burst transfer write timing for strobe use(split bus : DFORM=110, OBUS=1/0)



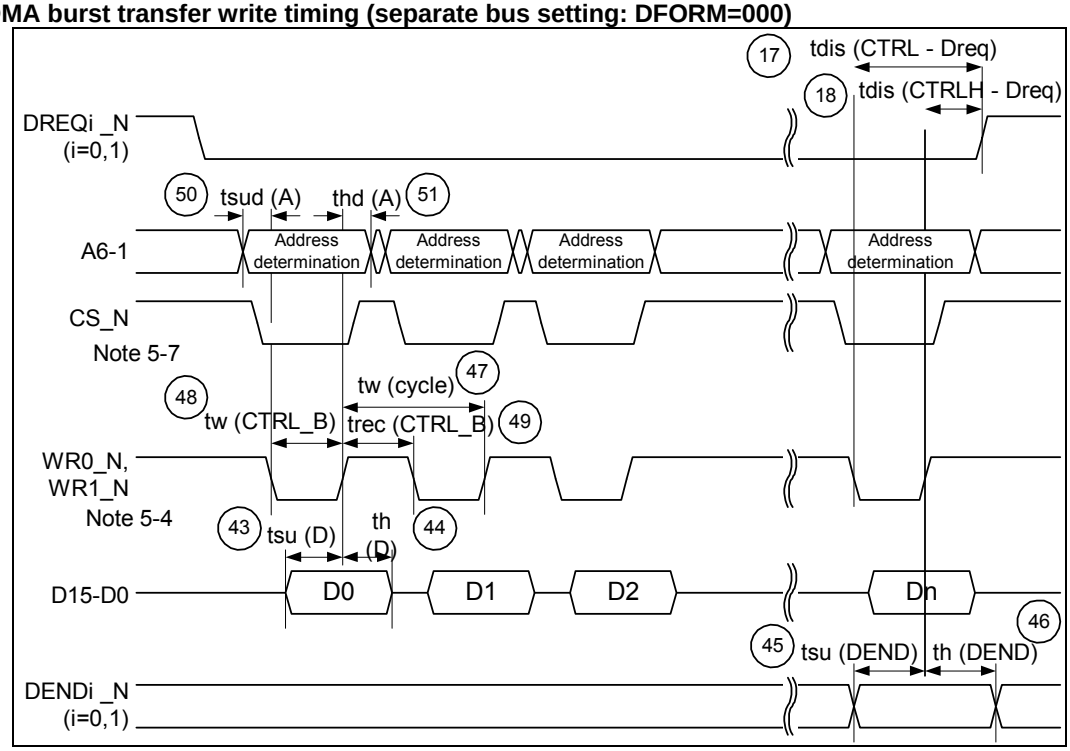
4.9.5.4 DMA burst transfer read timing for strobe use(split bus : DFORM=110, OBUS=1)



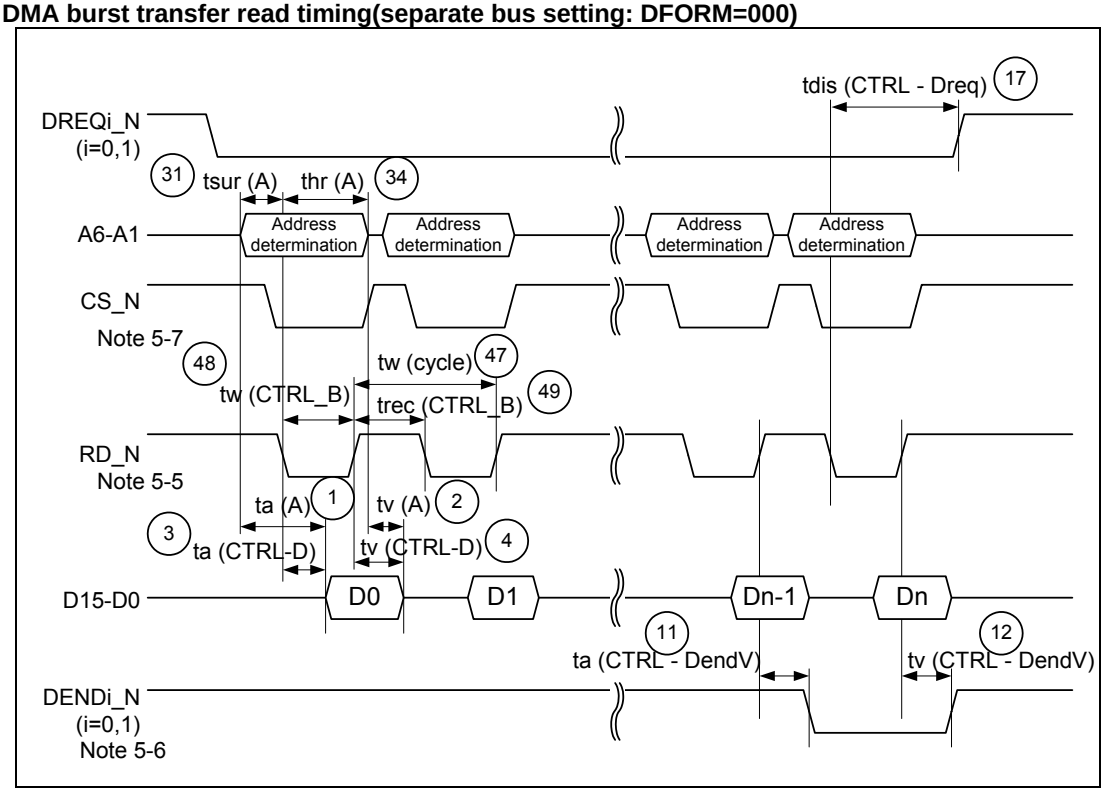
4.9.5.5 DMA burst transfer read timing for strobe use (split bus : DFORM=110, OBUS=0)



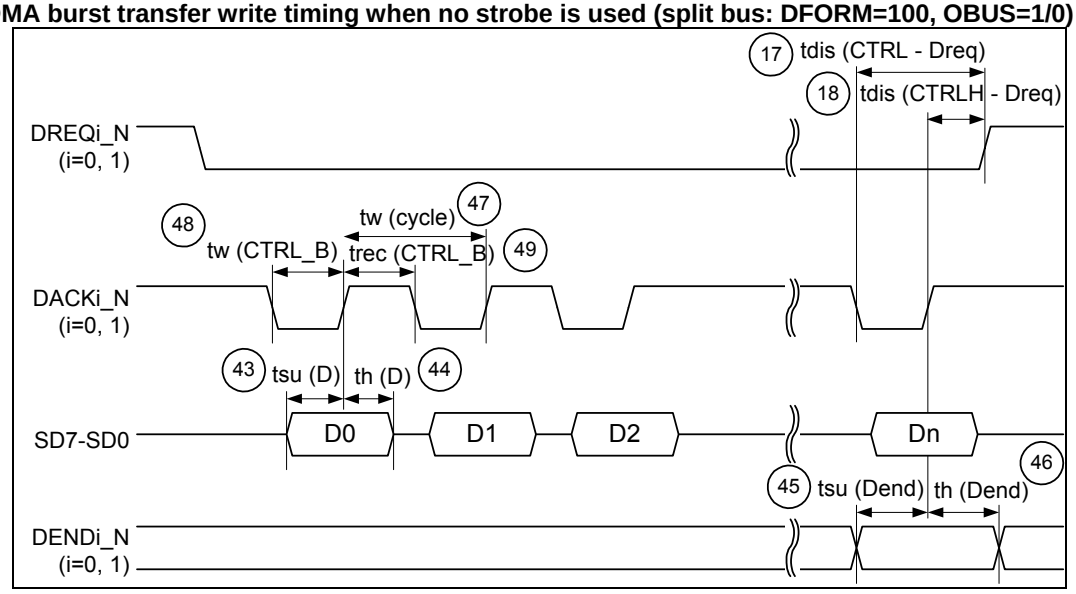
4.9.5.6 DMA burst transfer write timing (separate bus setting: DFORM=000)



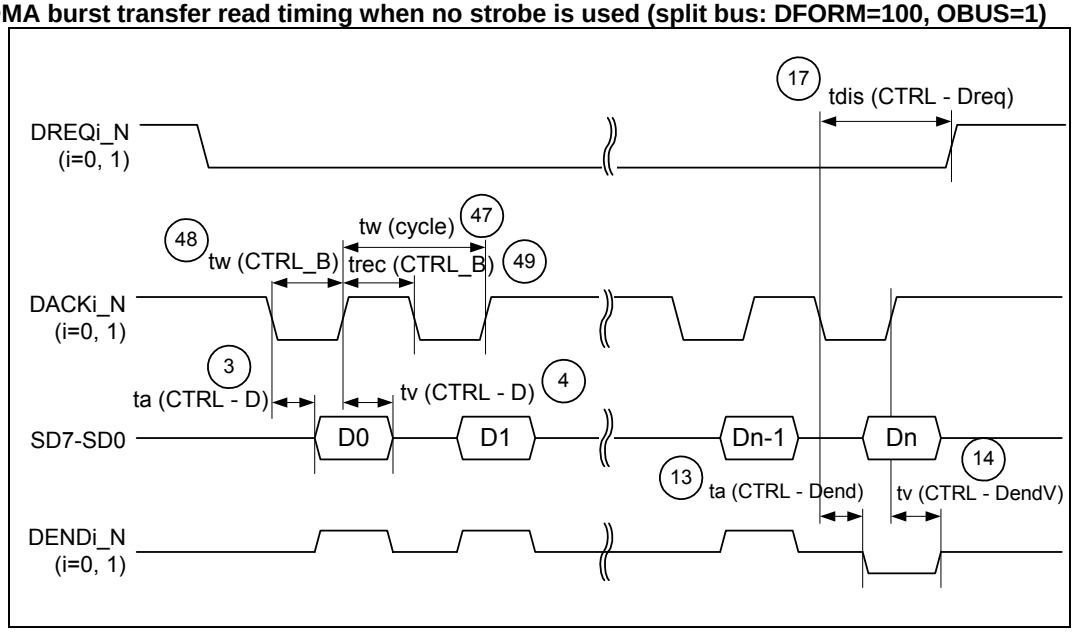
4.9.5.7 DMA burst transfer read timing(separate bus setting: DFORM=000)



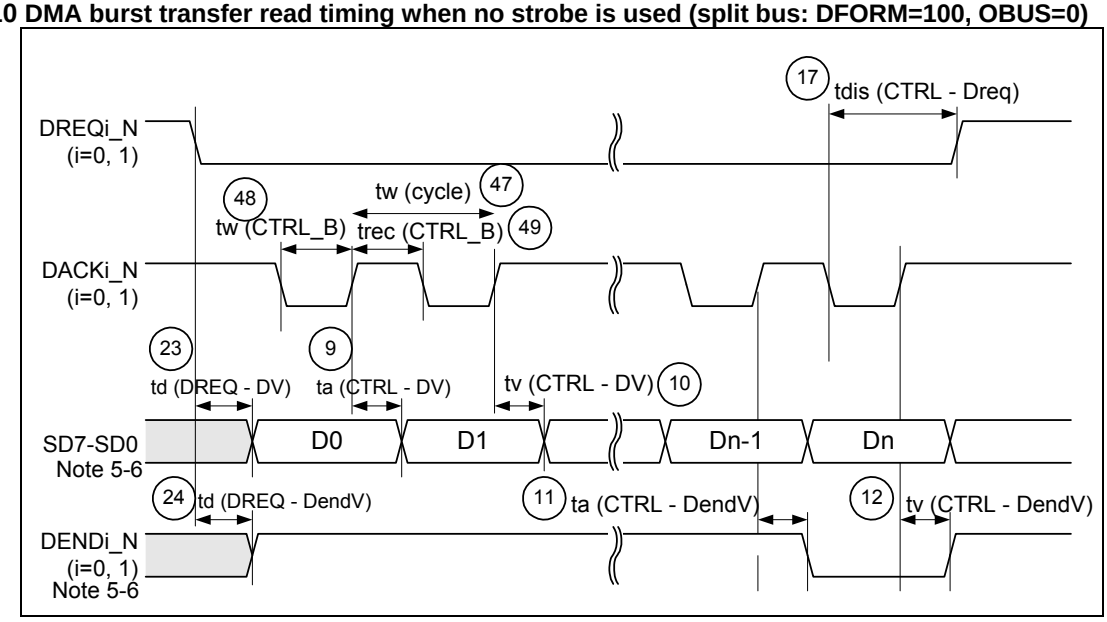
4.9.5.8 DMA burst transfer write timing when no strobe is used (split bus: DFORM=100, OBUS=1/0)



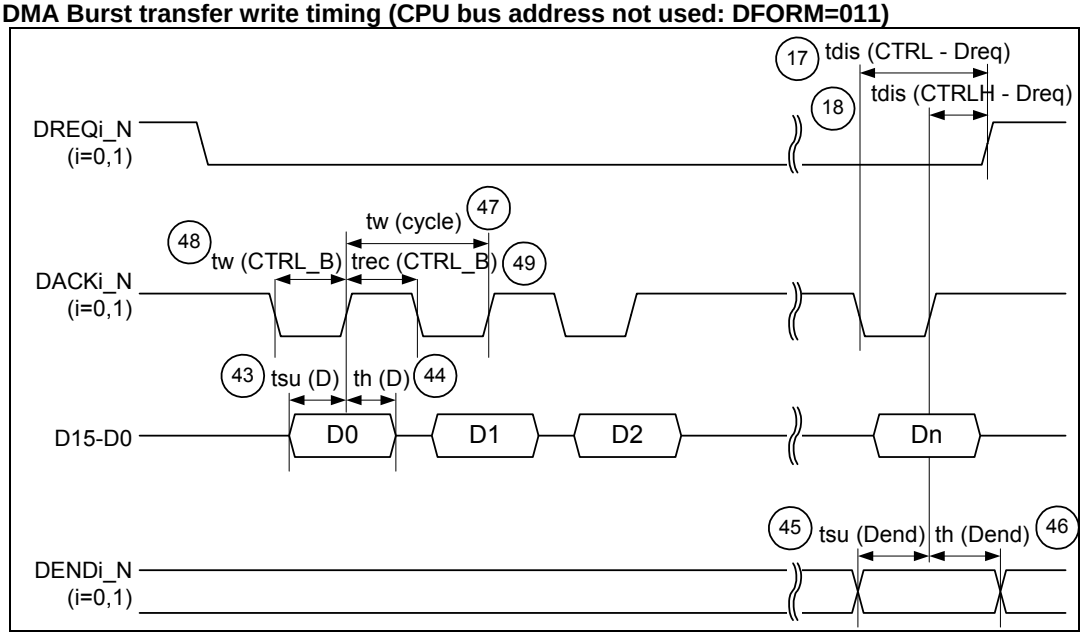
4.9.5.9 DMA burst transfer read timing when no strobe is used (split bus: DFORM=100, OBUS=1)



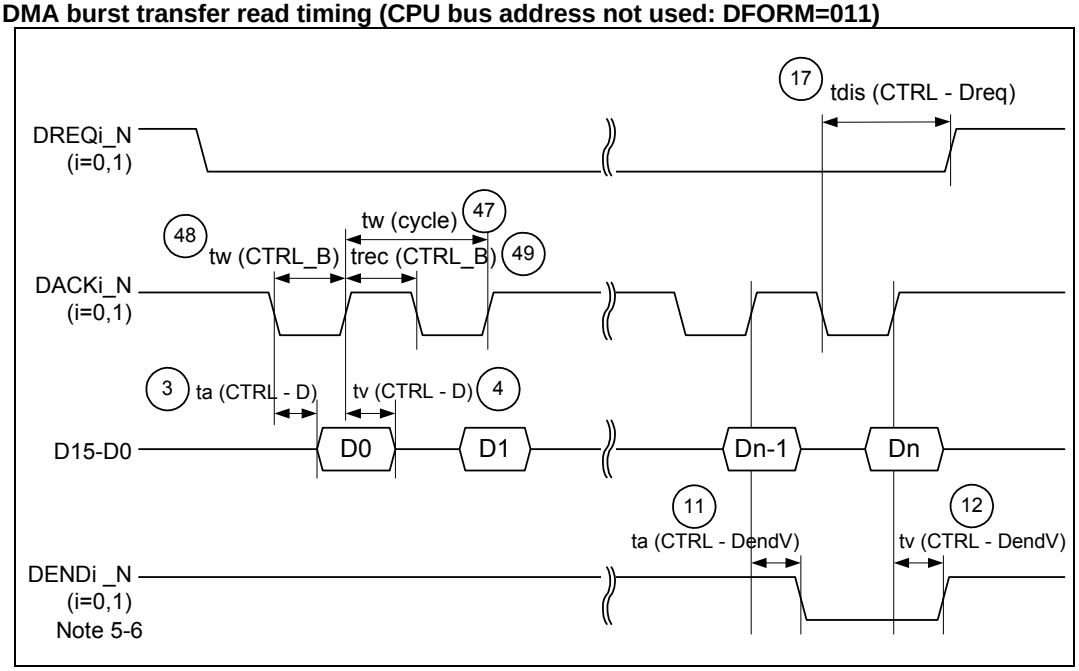
4.9.5.10 DMA burst transfer read timing when no strobe is used (split bus: DFORM=100, OBUS=0)



4.9.5.11 DMA Burst transfer write timing (CPU bus address not used: DFORM=011)



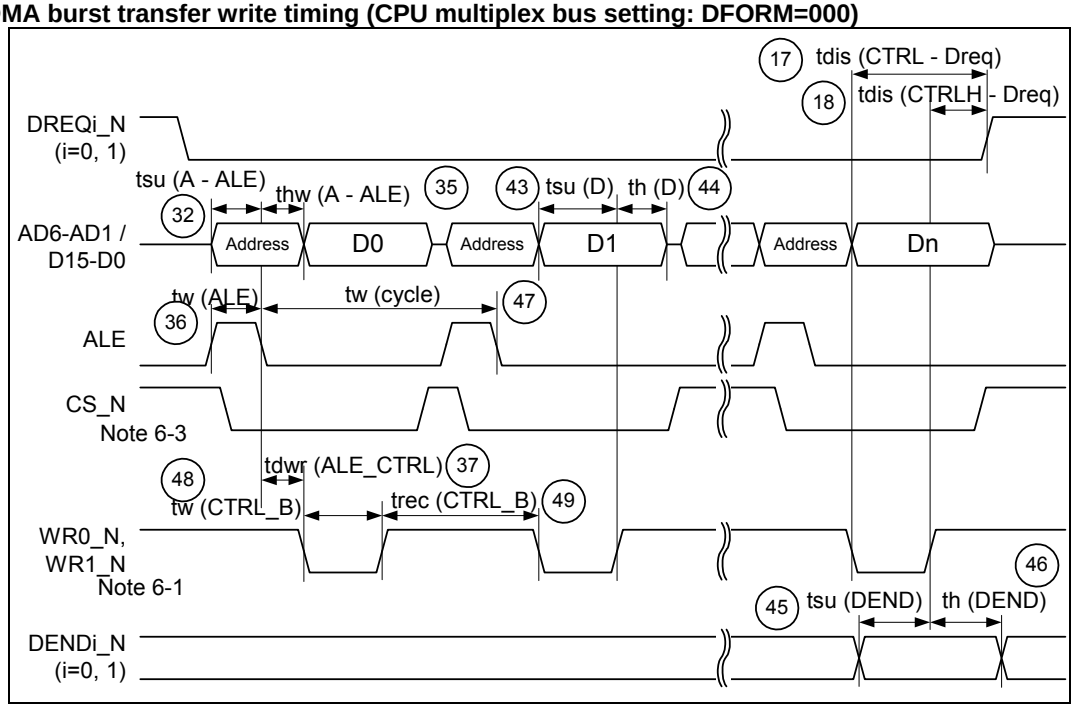
4.9.5.12 DMA burst transfer read timing (CPU bus address not used: DFORM=011)



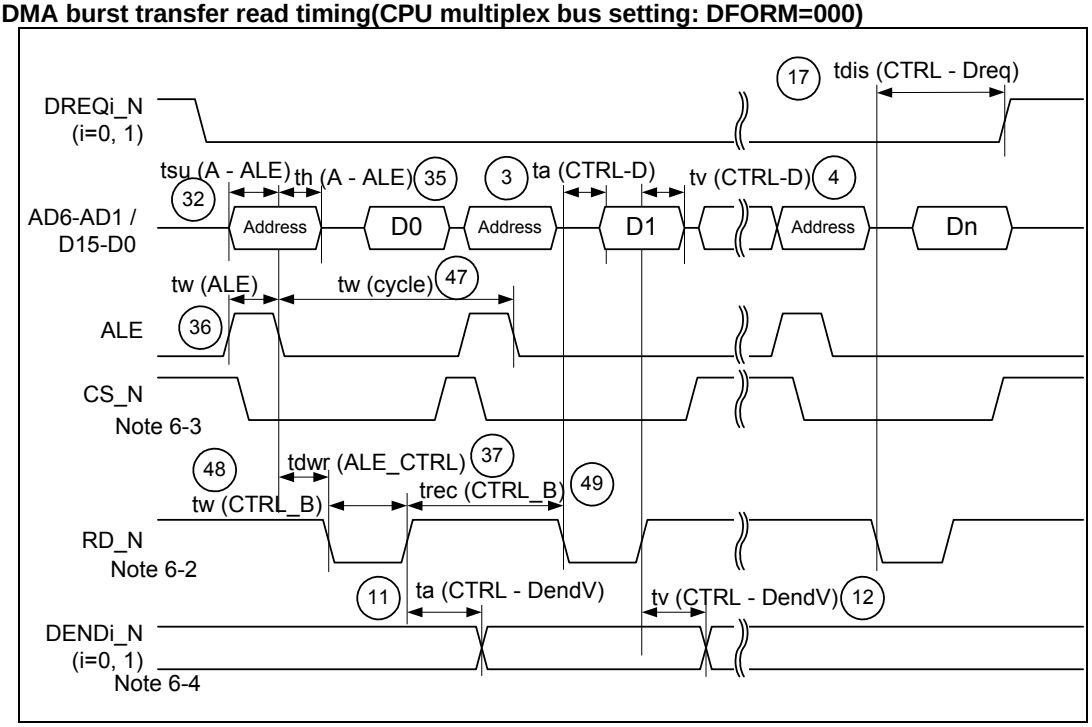
Note 5-1: The control signal when writing data is a combination of DACKi_N(i=0, 1), WR0_N and WR1_N.
Note 5-2: The control signal when reading data is a combination of DACKi_N and RD_N.
Note 5-3: The control signal when writing data is a combination of DACK0 and DSTRB0_N.
Note 5-4: The control signal when writing data is a combination of CS_N, WR0_N and WR1_N.
Note 5-5: The control signal when reading data is a combination of CS_N and RD_N.
Note 5-6: When the receipt data is one byte, the data determined time is "(23)td(DREQ-DV)" and the DEND determined time is "(24)td(DREQ-DendV)".
Note 5-7: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that CS_N is rising. Similarly, CS_N should not be timed to fall at the same timing that RD_N, WR0_N and WR1_N are rising. In the instances noted above, an interval must be needed at least 10ns.
Note 5-8:. RD_N, WR0_N and WR1_N should not be timed to fall at the same time that DACKi_N is rising. Similarly, DACKi_N should not be timed to fall at the same timing that RD_N, WR0_N and WR1_N are rising. In the instances noted above, an interval must be needed at least 10ns.

4.9.6 DMA access timing(burst transfer, when a multiplex bus is set)

4.9.6.1 DMA burst transfer write timing (CPU multiplex bus setting: DFORM=000)

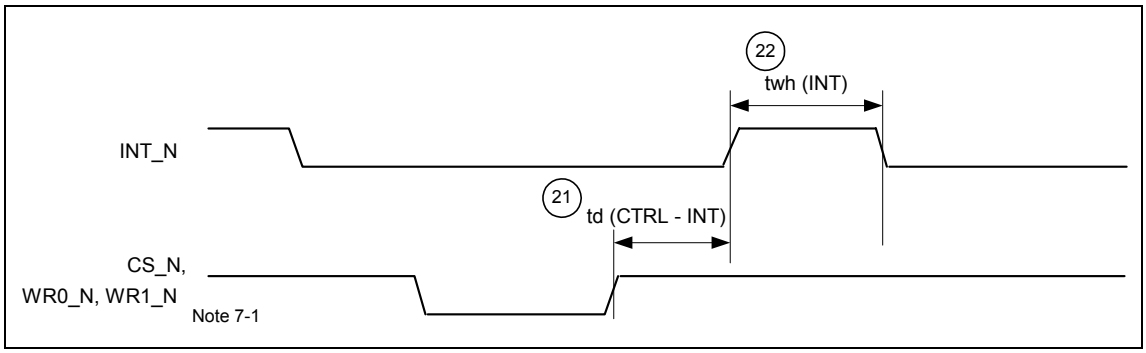


4.9.6.2 DMA burst transfer read timing(CPU multiplex bus setting: DFORM=000)



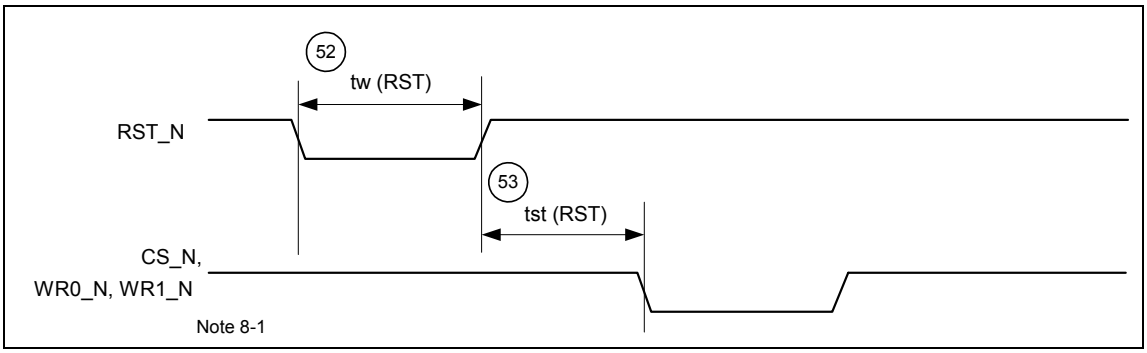
Note 6-1: The control signal when writing data is a combination of CS_N, WR0_N and WR1_N.
Note 6-2: The control signal when reading data is a combination of CS_N and RD_N.
Note 6-3: RD_N, WR0_N and WR1_N should not be timed to fall at the same time that CS_N is rising. Similarly, CS_N should not be timed to fall at the same timing that RD_N or WR0_N and WR1_N are rising. In the instances noted above, an interval of at least 10 ns must be left open.
Note 6-4: When the receipt data is one byte, the DEND determined time is "(24)td(DREQ-DendV)".

4.10 Interrupt timing



Note 7-1: Writing using the combination of CS_N, WR0_N and WR1_N takes place during the active (“L”) overlap period. The ratings from the rising edge are valid starting from the earliest change in the inactive signal.

4.11 Reset timing



Note 8-1: Writing using the combination of CS_N, WR0_N and WR1_N takes place during the active (“L”) overlap period. The ratings from the rising edge are valid starting from the earliest change in the inactive signal.

REVISION HISTORY	M66592 Data Sheet

Rev.	Date	Description	
		Page	Summary
0.80	Jul 12, 2004	-	Preliminary edition issued
1.00	Oct 1, 2004	-	First edition issued
		11 (1.7.5)	Deleted: description; " It is also possible to output pulses only when an SOF packet is damaged."
		16-17 (Table2.2)	Added: the **BRDYM** bit and the **PCSE** bit of the **INTENB1** register the **INBUFM** bit of the **PIPEnCTR** (n=1,2,...,5)
		20 (2.4)	Added: note *2) of the **DVSTCTR** register
		26 (2.4)	Deleted: note *3) for the **DREQE** bit
		31 (2.7)	Added: the **BRDYM** bit and the **PCSE** bit of the **INTENB1** register
		47 (2.14)	Added: the **INBUFM** bit of the **PIPEnCTR** (n=1,2,...,5)
		47 (2.14)	Added and modified: notes of the **PIPEnCTR** (n=1,2,...,5)
		50 (3.1.6.1)	Corrected: 2nd row, 1st column of the Table 3.3 ; DPRPU=0 → DPRPU=x
		51 (3.1.6.2)	Added: 3.1.6.2 "Overview of Clock stop state"
		53 (3.1.6.5)	Added: 3.1.6.5 "Recovering from the clock stop state"
		55 (3.1.7.3)	Modified: 3.1.7.3 (1)
		57 (3.1.7.4)	Added: 3.1.7.4 "Stopping the internal clock supply (from the normal operating state to the clock stop state)"
		57 (3.1.7.5)	Added: 3.1.7.5 "Starting the internal clock supply (from the clock stop state to the normal operating state : with "ATCKM=1")"
		59 (3.1.7.6)	Added: 3.1.7.6 "Starting the internal clock supply (from the clock stop state to the normal operating state : with "ATCKM=0")"
		- (3.1.7)	Deleted: 3.1.7.4(previous) "Starting the internal clock supply (from the low-power sleep state to the normal operating state: Auto clock supply function disabled)
		64 (Table 3.7)	Added: Description of SORF interrupt
		64 (Table 3.9)	Added: Description was added to "Conditions under which a BRDY interrupt is generated"
		72 (Table 3.11)	Added: the **SHTNAK** bit and the **INBUFM** bit
		76 (Table 3.13)	Added: Table 3.13 "Buffer statuses and the INBUFM bit"
		77 (3.4.1.4)	Added: description; "An access cycle of at least 100 ns is required between "ACLRM=1" and "ACLRM=0"."
		81 (Table3.18)	Added: Note *2)
		86 (3.6.3)	Corrected: Description of (a) and (b)
		97,98	Corrected: Average supply current at Hi-Speed operation: 60mA → 40mA

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500 Fax: <1> (408) 382-7501

Renesas Technology Europe Limited.
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, United Kingdom
Tel: <44> (1628) 585 100, Fax: <44> (1628) 585 900

Renesas Technology Europe GmbH
Dornacher Str. 3, D-85622 Feldkirchen, Germany
Tel: <49> (89) 380 70 0, Fax: <49> (89) 929 30 11

Renesas Technology Hong Kong Ltd.
7/F., North Tower, World Finance Centre, Harbour City, Canton Road, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2375-6836

Renesas Technology Taiwan Co., Ltd.
FL 10, #99, Fu-Hsing N. Rd., Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 2713-2999

Renesas Technology (Shanghai) Co., Ltd.
26/F., Ruijin Building, No.205 Maoming Road (S), Shanghai 200020, China
Tel: <86> (21) 6472-1001, Fax: <86> (21) 6415-2952

Renesas Technology Singapore Pte. Ltd.
1, Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Colophon .1.0