

28.3 Operation

28.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

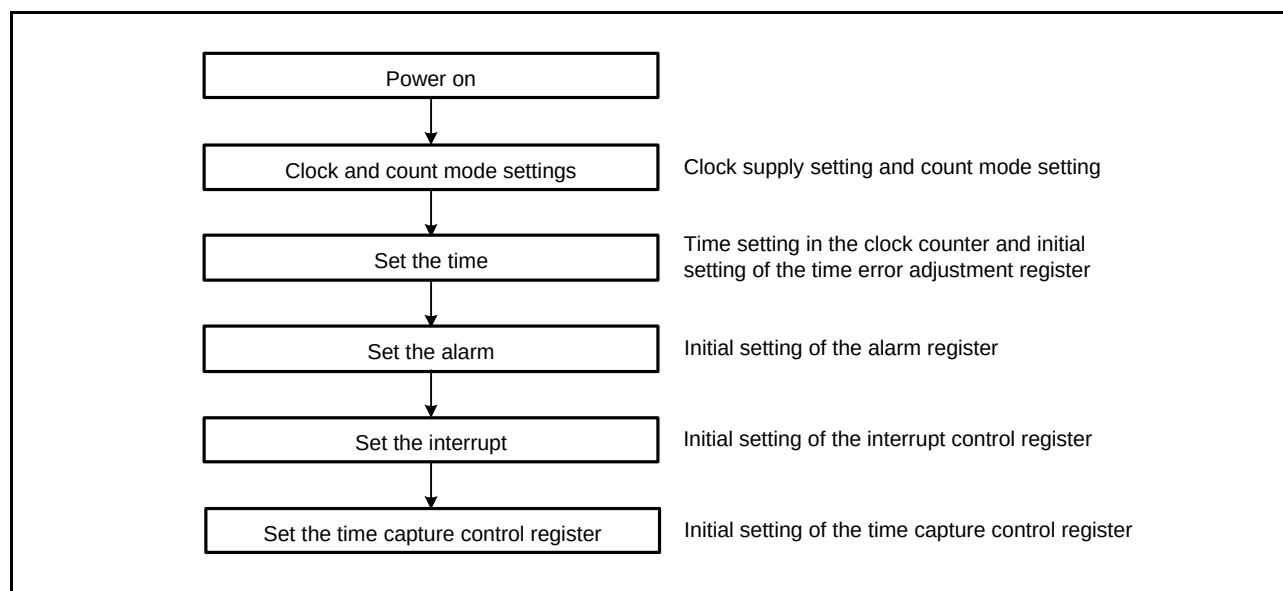


Figure 28.2 Outline of Initial Settings after Power On

28.3.2 Clock and Count Mode Setting Procedure

Figure 28.3 shows how to set the clock and the count mode.

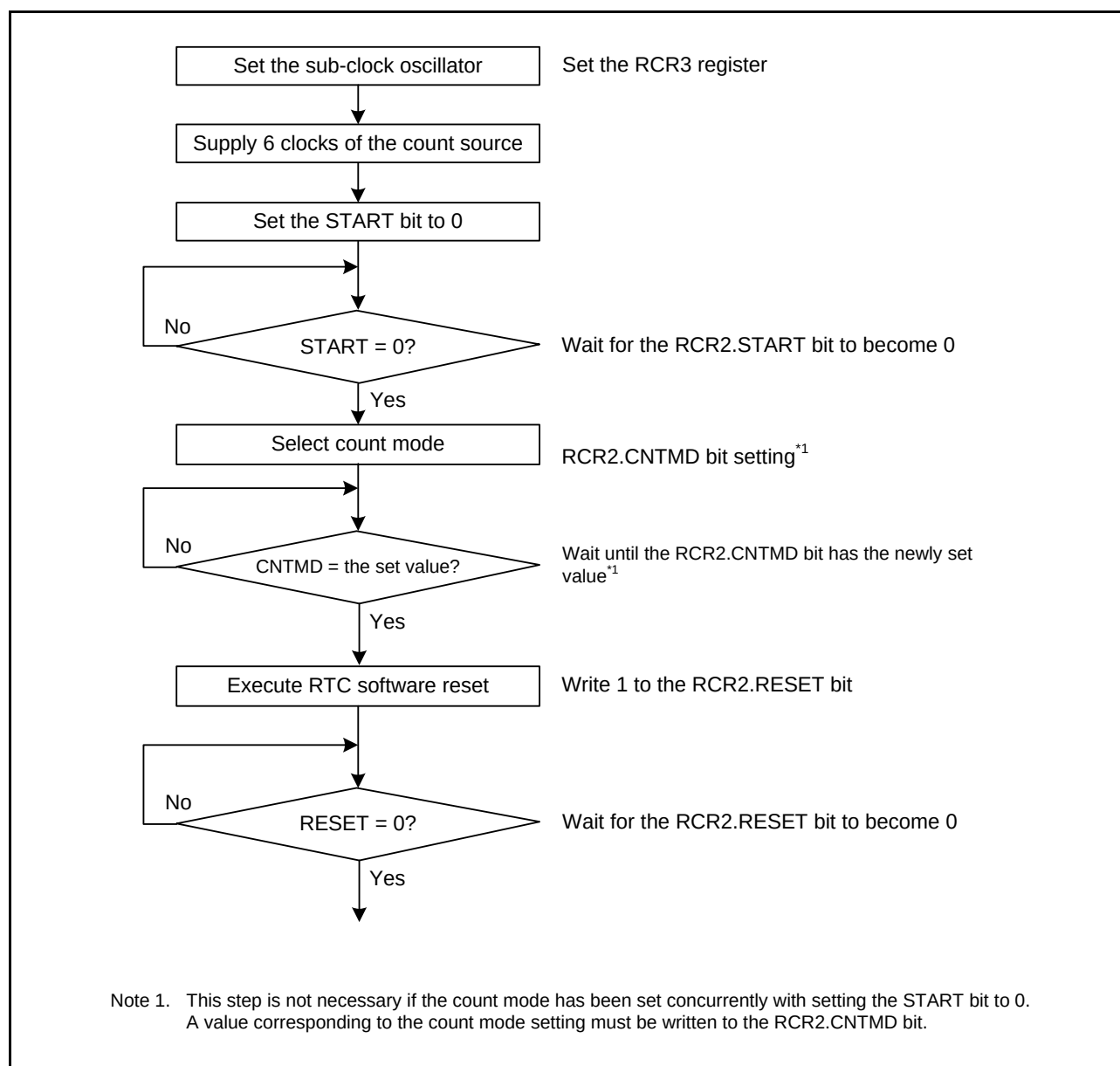


Figure 28.3 Clock and Count Mode Setting Procedure

28.3.3 Setting the Time

Figure 28.4 shows how to set the time.

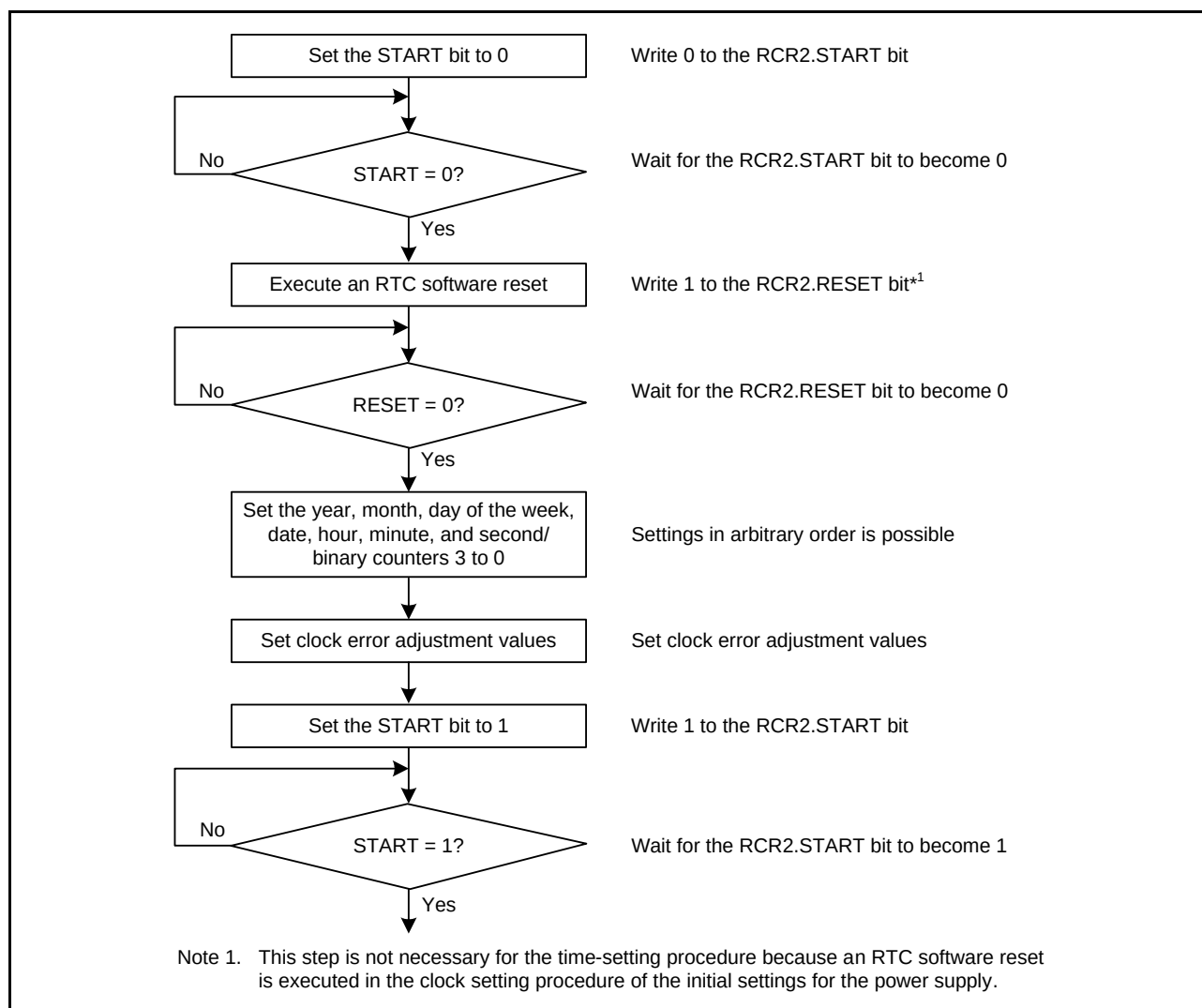


Figure 28.4 Setting the Time

28.3.4 30-Second Adjustment

Figure 28.5 shows how to execute 30-second adjustment.

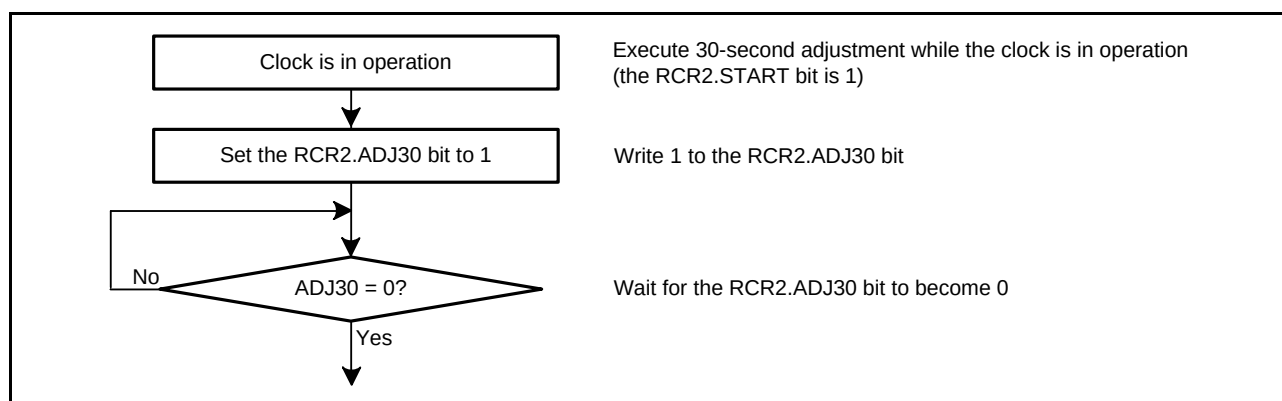


Figure 28.5 30-Second Adjustment

28.3.5 Reading 64-Hz Counter and Time

Figure 28.6 shows how to read the 64-Hz counter and time.

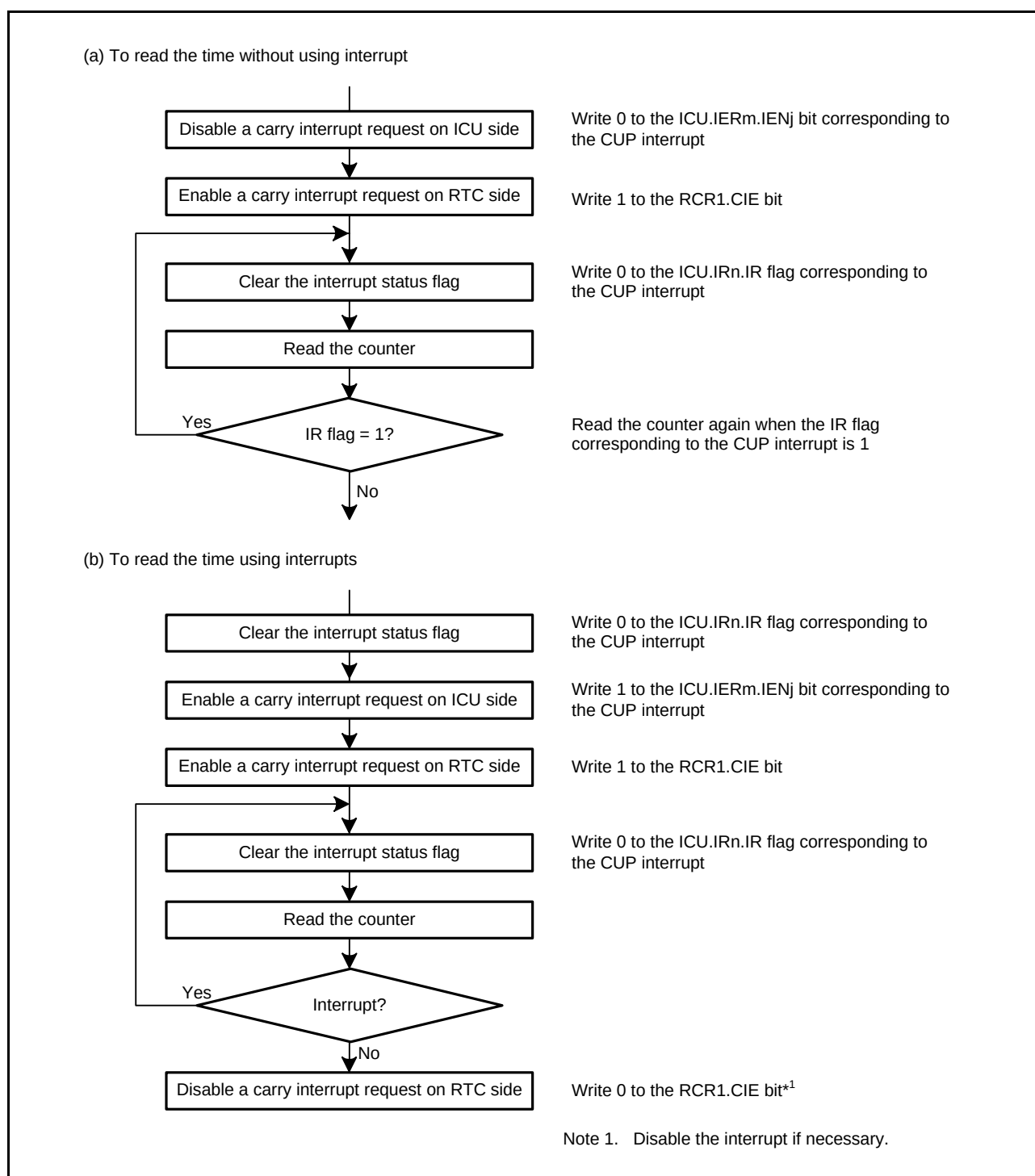


Figure 28.6 Reading Time

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 28.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

28.3.6 Alarm Function

Figure 28.7 shows how to use the alarm function.

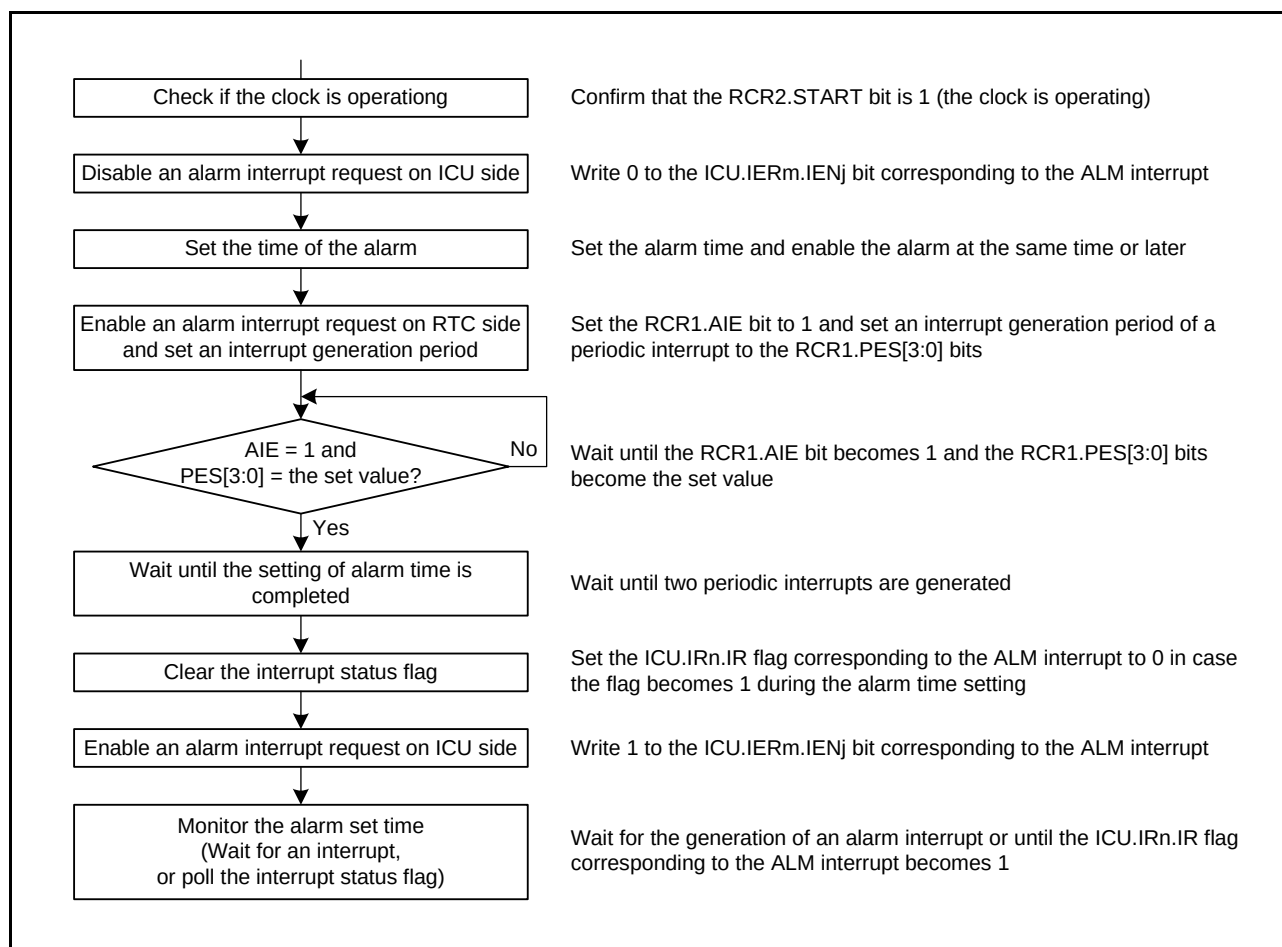


Figure 28.7 Using Alarm Function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR flag corresponding to the ALM interrupt is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 sets the IR flag corresponding to the ALM interrupt to 0.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state.

28.3.7 Procedure for Disabling Alarm Interrupt

Figure 28.8 shows the procedure for disabling the enabled alarm interrupt request.

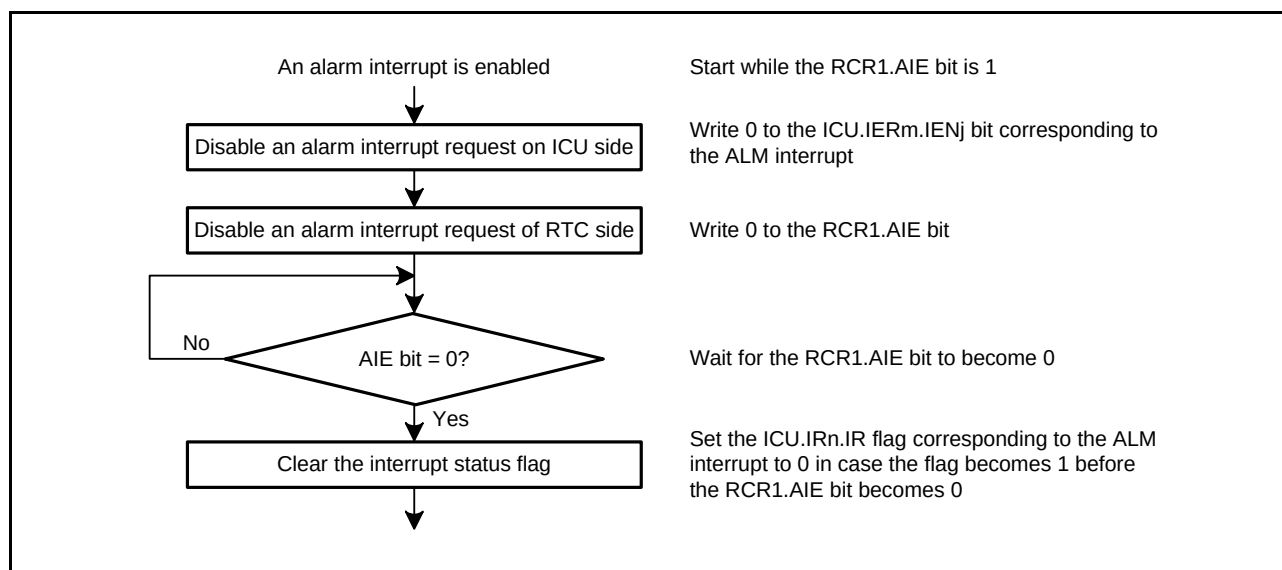


Figure 28.8 Procedure for Disabling Alarm Interrupt Request

28.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.

Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software. Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

28.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.

Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.

Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz

Adjustment procedure:

When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz

Adjustment procedure:

At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

28.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.

Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.

An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)

This is written to the RADJ register once per 1-second interrupt.

28.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

- (1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
- (2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
- (3) Use the RCR2.AADJP bit to select the period of adjustment.
- (4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from adjustment by software to automatic adjustment:

- (1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
- (2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
- (3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

28.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

28.3.9 Time Capture Function

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in Figure 28.9 and operation when the noise filter is on is shown in Figure 28.10.

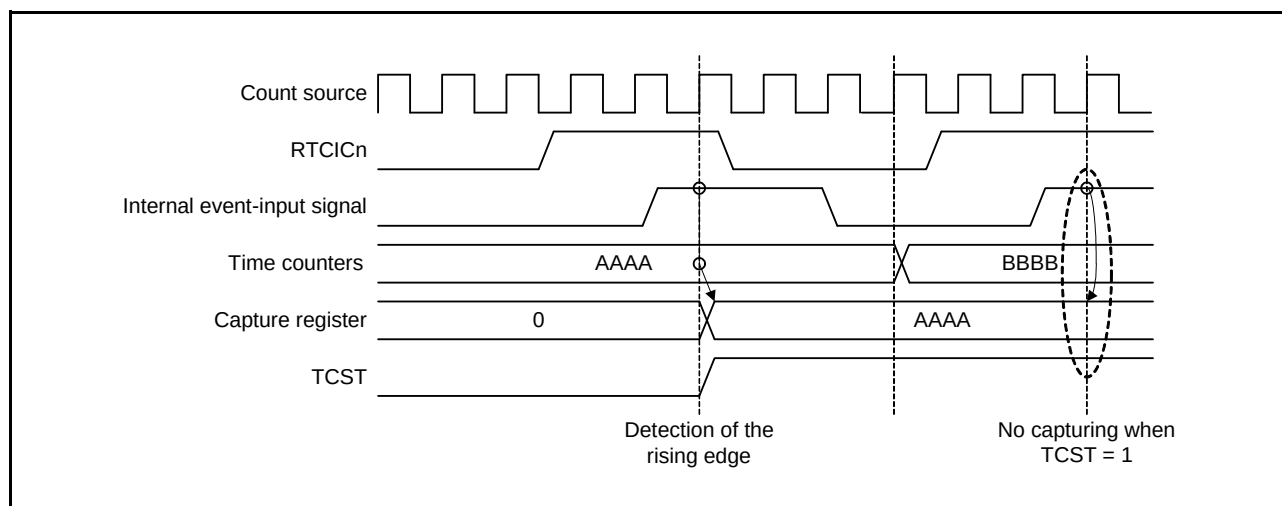


Figure 28.9 Timing of a Time Capture Operation (with the Filter Off) (n = 0, 1)

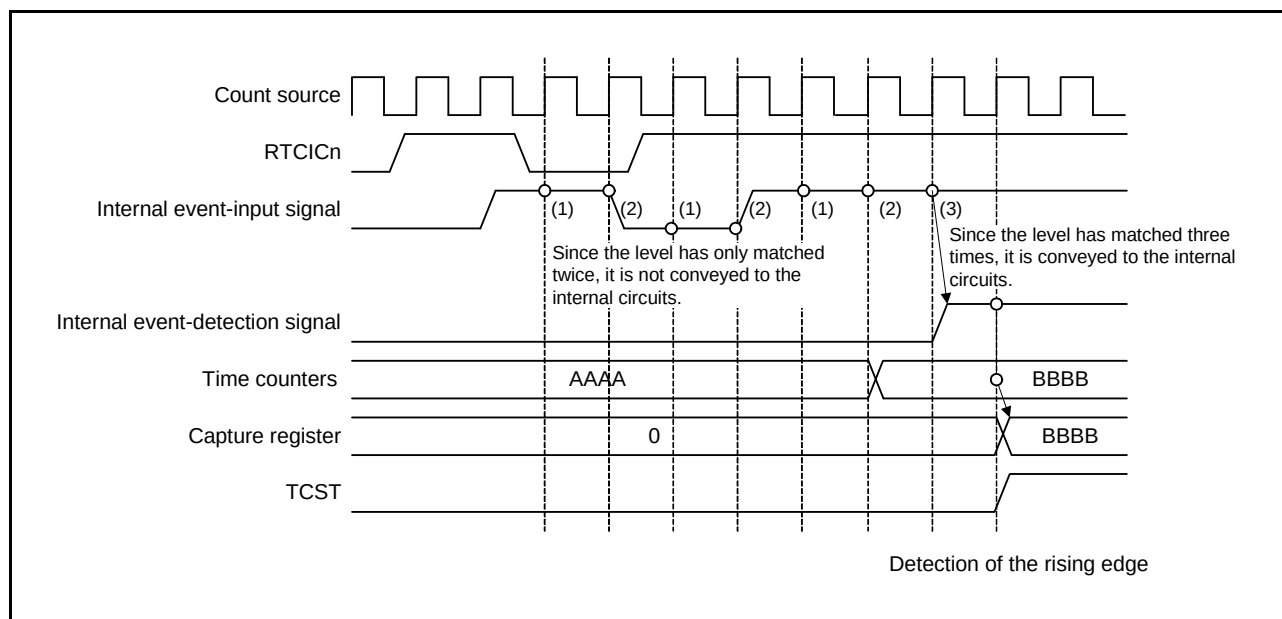


Figure 28.10 Timing of a Time Capture Operation (with the Filter On) (n = 0, 1)

28.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 28.3 lists interrupt sources for the RTC.

Table 28.3 RTC Interrupt Sources

Name	Interrupt Sources
ALM	Alarm interrupt
PRD	Periodic interrupt
CUP	Carry interrupt

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to section 28.3.6, Alarm Function).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and set the IR flag corresponding to the ALM interrupt to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.

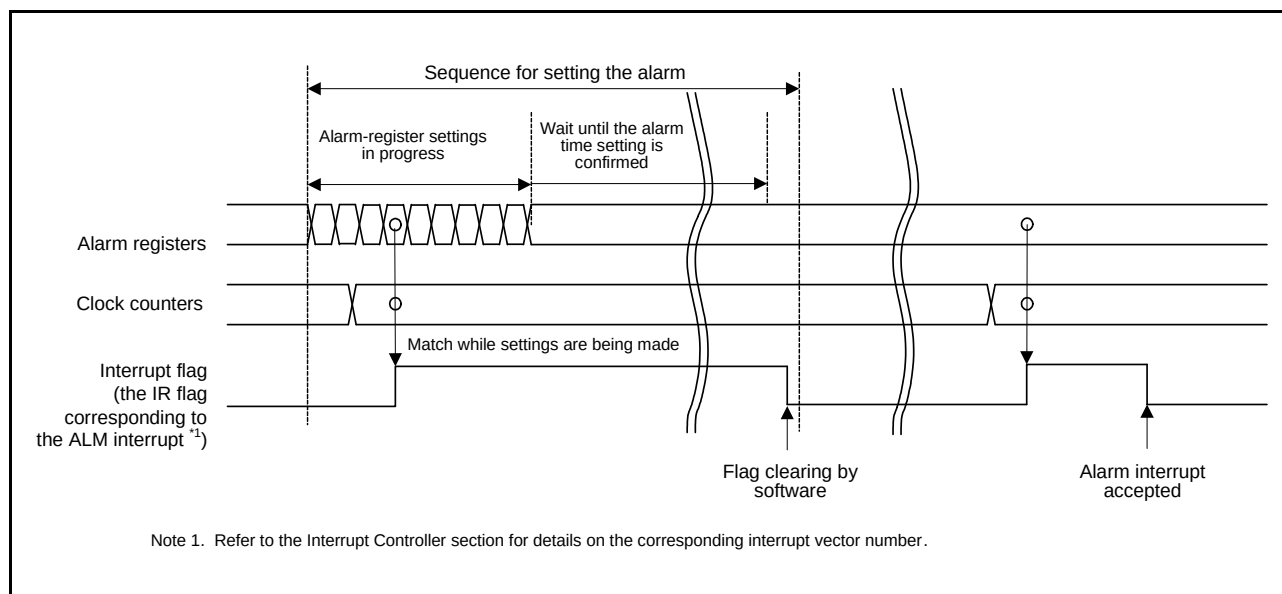


Figure 28.11 Timing Chart for the Alarm Interrupt (ALM)

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

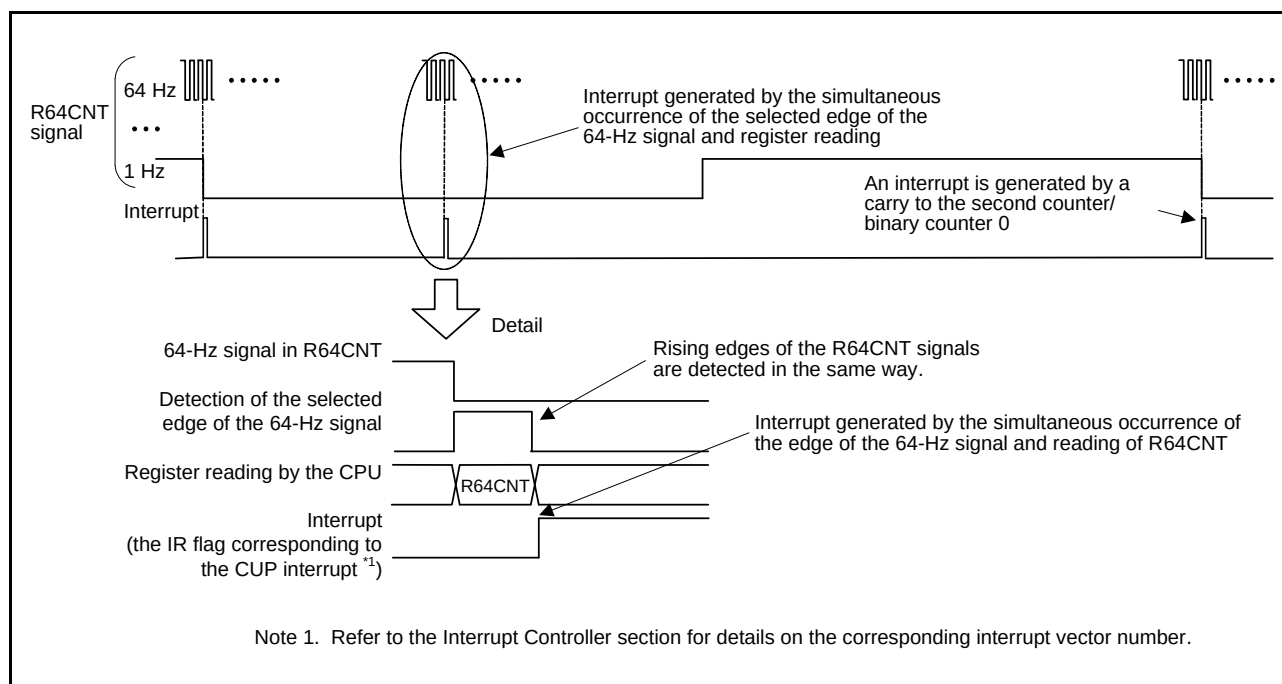


Figure 28.12 Carry Interrupt (CUP) Timing Chart

28.5 Event Link Output

The RTC outputs the following event signals for the event link controller (ELC), and these can be used to initiate operations by other modules selected in advance.

(1) Periodic event output

The periodic event signal is output at the interval selected from among 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by the setting of the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is to be used, only make the ELC settings after making the RTC settings (initialization, time settings, etc.). Making the RTC settings after the ELC settings can lead to the output of unexpected event signals.

28.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during software standby, the periodic event signals for the ELC are not output.

28.6 Usage Notes

28.6.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24

The counter must be stopped before writing to any of the above registers.

28.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 28.13.

The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.

Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

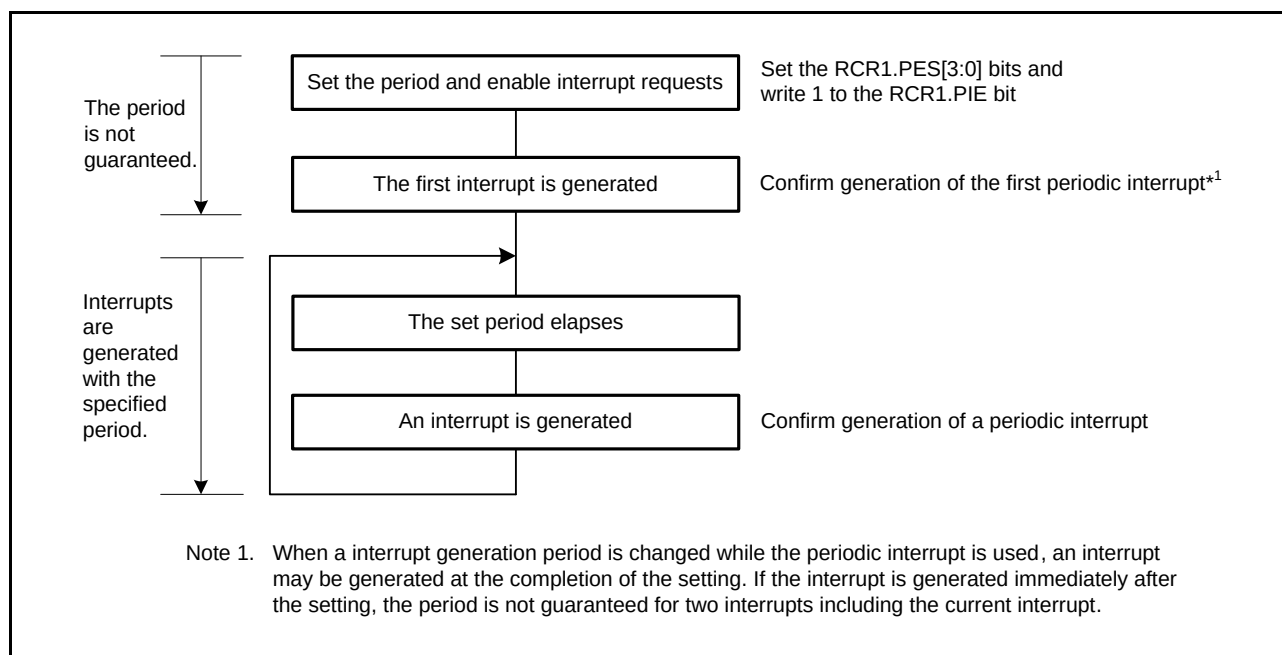


Figure 28.13 Using Periodic Interrupt Function

28.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

28.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, or battery backup) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

28.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, or RCR3 register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, software standby mode, or the battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the count source have elapsed.

28.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 28.3.1, Outline of Initial Settings of Registers after Power On.

28.6.7 Initialization Procedure When the Realtime Clock is Not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt request or operation of the counter may lead to increased power consumption.

For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 28.14.

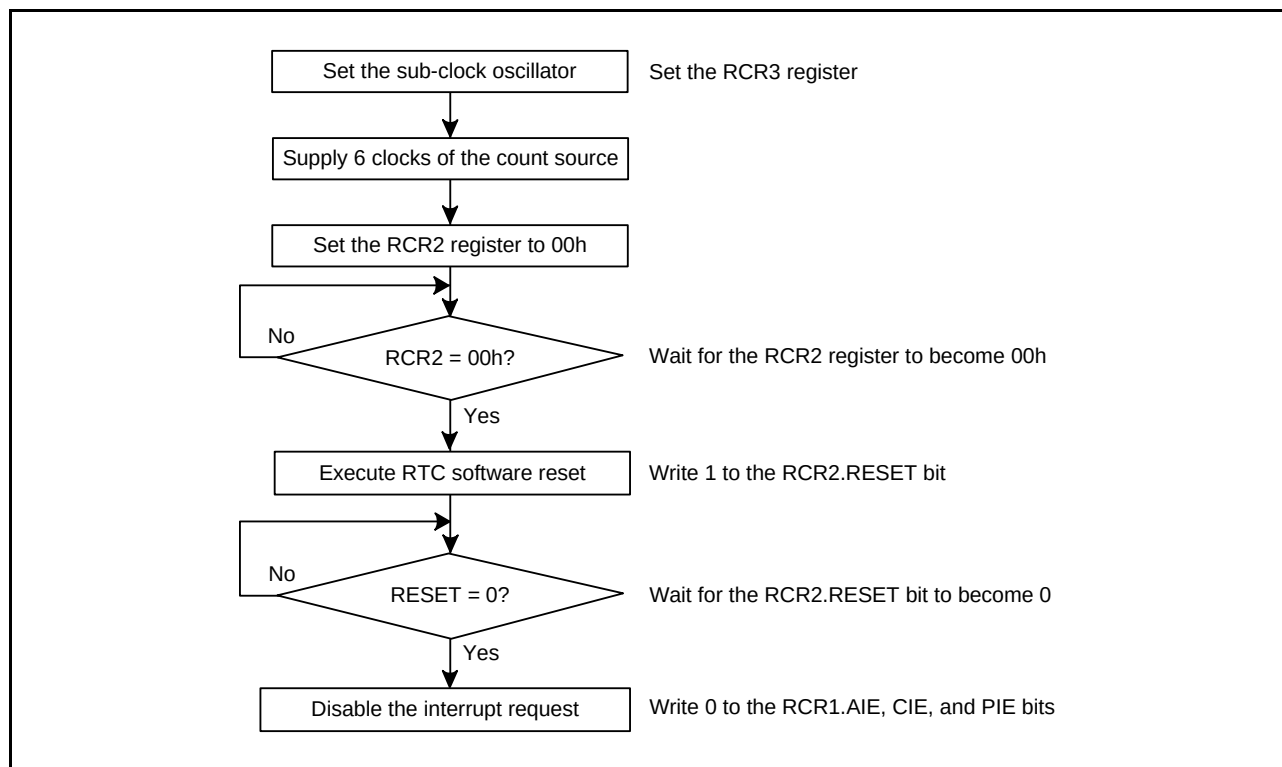


Figure 28.14 Initialization Procedure

29. Low-Power Timer (LPT)

29.1 Overview

This MCU integrates a low-power timer (LPT) that consists of a single-channel 16-bit timer. The LPT uses a sub-clock or IWDG-dedicated clock as the clock source, and can continue counting operation even in software standby mode. A compare match signal can be used to return from software standby mode to normal operating mode. Table 29.1 lists the specifications of the LPT and Figure 29.1 shows a block diagram of the LPT.

Table 29.1 LPT Specifications

Item	Description
Clock source	Sub-clock or IWDG-dedicated clock
Clock division ratio	Divided by 2, 4, 8, 16, or 32
Counting operation	- Count up using the 16-bit up-counter - Counting operation can be continued even in software standby mode
Compare match	Compare match 0 (a compare match signal is generated only in software standby mode)
Event link function (output)	Compare match 0 (a compare match signal is generated only in software standby mode)

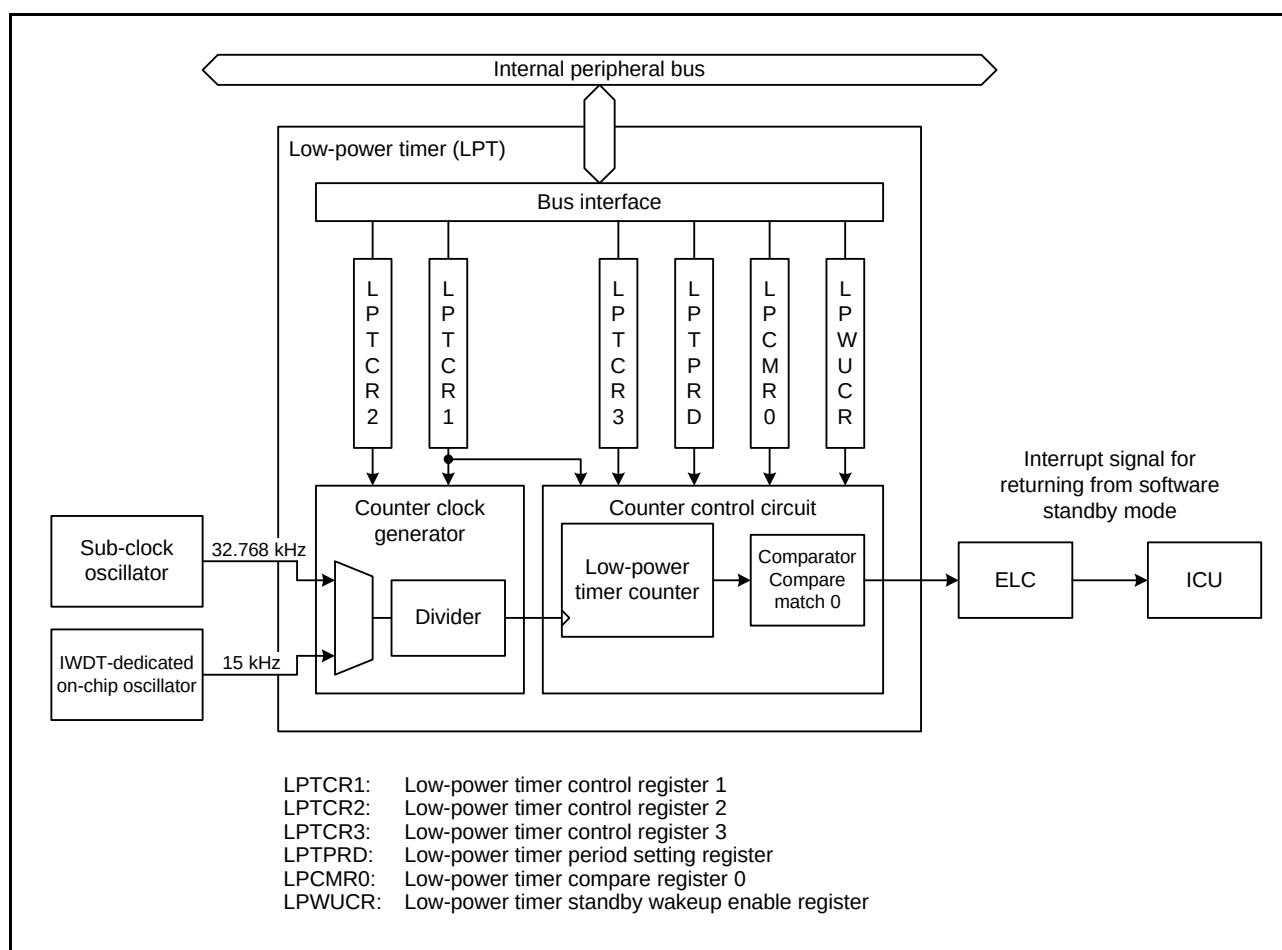


Figure 29.1 LPT Block Diagram

29.2 Register Descriptions

29.2.1 Low-Power Timer Control Register 1 (LPTCR1)

Address(es): LPT.LPTCR1 0008 00B0h

b7	b6	b5	b4	b3	b2	b1	b0
—	LPCMR E0	—	LPCNT CKSEL	—	LPCNTPSSEL[2:0]		
Value after reset:	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	LPCNTPSSEL[2:0]	Clock Division Ratio Select*1	b2 b0 0 0 1: Divided by 2 0 1 0: Divided by 4 0 1 1: Divided by 8 1 0 0: Divided by 16 1 0 1: Divided by 32 Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	LPCNTCKSEL	Clock Source Select*1, *2	0: Sub-clock 1: IWDT-dedicated clock (IWDTCLK)*3	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	LPCMRE0	Compare Match 0 Enable*4	0: Compare match 0 is disabled 1: Compare match 0 is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

Note 1. Rewrite these bits while the LPTCR2.LPCNTSTP bit is 1 (supply of clock to the low-power timer is stopped).

Note 2. Satisfy that the frequency of the system clock (ICLK) and peripheral module clock (PCLKB) $\geq 4 \times$ (the frequency of the clock source).

Note 3. A clock generated by the IWDT-dedicated on-chip oscillator (IWDTCLK) is supplied to the low-power timer. When modifying this bit, make sure that the IWDT-dedicated on-chip oscillator is oscillating stably.

When the IWDTCLK is used as the clock source for the low-power timer, set the OFS0.IWDTSLCSTP bit to 0 (counting stop is disabled) in IWDT auto-start mode operation, and set the IWDTCSSTPR.SLCSTP bit to 0 (count stop is disabled) in other modes. Without this setting, the IWDT-dedicated on-chip oscillator is stopped in software standby mode.

Note 4. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTCR1 register is used to control the low-power timer.

LPCNTCKSEL Bit (Clock Source Select)

This bit is used to select the sub-clock or IWDT-dedicated clock as the clock source for the low-power timer.

LPCMRE0 Bit (Compare Match 0 Enable)

This bit enables or disables low-power timer compare match 0.

When the low-power timer is put into operation and the MCU makes a transition to software standby mode while this bit and the LPWUCR.LPWKUPEN bit are set to 1 (wake up from software standby mode using low-power timer is enabled), the MCU returns from software standby mode to normal operating mode through the event link controller (ELC) when the value of the low-power timer counter matches the value of the LPCMR0 register.

Settings for the interrupt and ELC are necessary to use a compare match 0 as a trigger source to return from software standby mode.

Refer to section 20, Event Link Controller (ELC) for details on the ELC settings, and refer to section 15, Interrupt Controller (ICUb) for details on the interrupt settings.

An interrupt request at compare match 0 is generated only in software standby mode. It is not generated in normal operating mode, sleep mode, and deep sleep mode.

29.2.2 Low-Power Timer Control Register 2 (LPTCR2)

Address(es): LPT.LPTCR2 0008 00B1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LPCNT STP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	LPCNTSTP	Clock Supply Control	0: Clock is supplied to the low-power timer. 1: Supply of clock to the low-power timer is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

The LPTCR2 register is used to control supply of the clock to be used for the low-power timer.

LPCNTSTP Bit (Clock Supply Control)

This bit is used to supply or stop the clock to be used for the low-power timer. When this bit is set to 0, the clock signal is supplied to the low-power timer counter and divider.

29.2.3 Low-Power Timer Control Register 3 (LPTCR3)

Address(es): LPT.LPTCR3 0008 00B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LPCNT RST	LPCNT EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LPCNTEN	Low-Power Timer Counter Operation Control	0: Low-power timer counter stops 1: Low-power timer counter operates	R/W
b1	LPCNTRST	Low-Power Timer Counter Clear*1, *2	- When writing 0: Has no effect 1: Clears divider and counter - When reading 0: Clearing is completed 1: Clearing is in progress	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

Note: Rewrite this register while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer).

Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

Note 2. When clearing the low-power timer counter successively, confirm that the LPCNTRST bit becomes 0, wait for at least one cycle of the clock selected by the LPTCR1.LPCNTCKSEL bit, and then write 1 to the LPCNTRST bit again.

The LPTCR3 register controls operations of and clears the low-power timer counter and divider.

LPCNTEN Bit (Low-Power Timer Counter Operation Control)

This bit is used to operate or stop the low-power timer counter and divider.

When this bit is set to 1 while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer), the low-power timer counter and divider start operating.

Do not write 1 to the LPCNTRST bit while this bit is 1.

LPCNTRST Bit (Low-Power Timer Counter Clear)

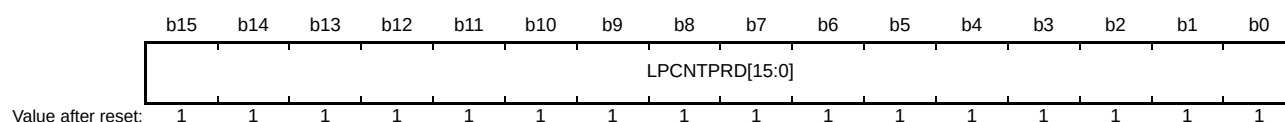
This bit is used to clear the low-power timer counter and divider.

When this bit is set to 1 while the LPTCR2.LPCNTSTP bit is 0 (clock is supplied to the low-power timer), the low-power timer counter and divider are cleared in synchronization with the clock used for the low-power timer. Once clearing is complete, this bit automatically becomes 0.

When 1 is written to this bit, confirm that its value becomes 0 before executing the next processing.

29.2.4 Low-Power Timer Period Setting Register (LPTPRD)

Address(es): LPT.LPTPRD 0008 00B4h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LPCNTPRD[15:0]	Low-Power Timer Period Setting	Set the period of the low-power timer. Setting range: 0001h to FFFFh	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

Note: Rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPTPRD register is used to set the period of the low-power timer.

The period of the low-power timer is proportional to “LPTPRD + 1” and calculated by the following formula:

$$\text{Period of low-power timer} = \text{period of clock source} \times \text{division ratio} \times (\text{LPTPRD} + 1)$$

When the value of the low-power timer counter matches the set value, the counter is cleared to 0000h and continues counting.

Do not set this register to 0000h.

Table 29.2, and Table 29.3 list examples of setting the periods of the low-power timer. These examples show values most approximate to the periods.

Table 29.2 Example of Low-Power Timer Period Settings for IWDTCCLK

Division ratio	Divided by 2			Divided by 4			Divided by 8		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)
1	0006h	0.93	-6.67	0003h	1.07	6.67	0001h	1.07	6.67
2	000Dh	1.87	-6.67	0006h	1.87	-6.67	0003h	2.13	6.67
5	0024h	4.93	-1.33	0011h	4.80	-4.00	0008h	4.80	-4.00
10	004Ah	10.00	0.00	0024h	9.87	-1.33	0011h	9.60	-4.00
20	0095h	20.00	0.00	004Ah	20.00	0.00	0024h	19.73	-1.33
50	0176h	50.00	0.00	00BAh	49.87	-0.27	005Ch	49.60	-0.80
100	02EDh	100.00	0.00	0176h	100.00	0.00	00BAh	99.73	-0.27
200	05DBh	200.00	0.00	02EDh	200.00	0.00	0176h	200.00	0.00
500	0EA4h	499.87	-0.03	0751h	499.73	-0.05	03A8h	499.73	-0.05
1000	1D4Ah	999.87	-0.01	0EA4h	999.73	-0.03	0751h	999.47	-0.05
2000	3A96h	1999.87	-0.01	1D4Ah	1999.73	-0.01	0EA4h	1999.47	-0.03
5000	927Bh	5000.00	0.00	493Dh	5000.00	0.00	249Eh	5000.00	0.00
10000	—	—	—	—	—	—	493Dh	10000.00	0.00
20000	—	—	—	—	—	—	927Bh	20000.00	0.00
50000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)
1	—	—	—	—	—	—
2	0001h	2.13	6.67	—	—	—
5	0004h	5.33	6.67	0001h	4.27	-14.67
10	0008h	9.60	-4.00	0004h	10.67	6.67
20	0011h	19.20	-4.00	0008h	19.20	-4.00
50	002Dh	49.07	-1.87	0016h	49.07	-1.87
100	005Ch	99.20	-0.80	002Dh	98.13	-1.87
200	00BAh	199.47	-0.27	005Ch	198.40	-0.80
500	01D3h	499.20	-0.16	00E9h	499.20	-0.16
1000	03A8h	999.47	-0.05	01D3h	998.40	-0.16
2000	0751h	1998.93	-0.05	03A8h	1998.93	-0.05
5000	124Eh	4999.47	-0.01	0926h	4998.40	-0.03
10000	249Eh	10000.00	0.00	124Eh	9998.93	-0.01
20000	493Dh	20000.00	0.00	249Eh	20000.00	0.00
50000	B71Ah	50000.00	0.00	5B8Ch	49998.93	0.00

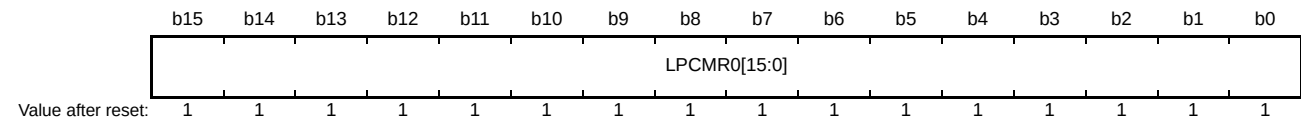
Table 29.3 Example of Low-Power Timer Period Settings for Sub-Clock

Division ratio	Divided by 2			Divided by 4			Divided by 8		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)
1	000Fh	0.98	-2.34	0007h	0.98	-2.34	0003h	0.98	-2.34
2	001Fh	1.95	-2.34	000Fh	1.95	-2.34	0007h	1.95	-2.34
5	0050h	4.94	-1.12	0027h	4.88	-2.34	0013h	4.88	-2.34
10	00A2h	9.95	-0.51	0050h	9.89	-1.12	0027h	9.77	-2.34
20	0146h	19.96	-0.21	00A2h	19.90	-0.51	0050h	19.78	-1.12
50	0332h	49.99	-0.02	0198h	49.93	-0.15	00CBh	49.80	-0.39
100	0665h	99.98	-0.02	0332h	99.98	-0.02	0198h	99.85	-0.15
200	0CCBh	199.95	-0.02	0665h	199.95	-0.02	0332h	199.95	-0.02
500	1FFFh	500.00	0.00	0FFFh	500.00	0.00	07FFh	500.00	0.00
1000	3FFFh	1000.00	0.00	1FFFh	1000.00	0.00	0FFFh	1000.00	0.00
2000	7FFFh	2000.00	0.00	3FFFh	2000.00	0.00	1FFFh	2000.00	0.00
5000	—	—	—	9FFFh	5000.00	0.00	4FFFh	5000.00	0.00
10000	—	—	—	—	—	—	9FFFh	10000.00	0.00
20000	—	—	—	—	—	—	—	—	—
50000	—	—	—	—	—	—	—	—	—

Division ratio	Divided by 16			Divided by 32		
Target period (ms)	Set Value	Actual period (ms)	Error (%)	Set Value	Actual period (ms)	Error (%)
1	0001h	0.98	-2.34	—	—	—
2	0003h	1.95	-2.34	0001h	1.95	-2.34
5	0009h	4.88	-2.34	0004h	4.88	-2.34
10	0013h	9.77	-2.34	0009h	9.77	-2.34
20	0027h	19.53	-2.34	0013h	19.53	-2.34
50	0065h	49.80	-0.39	0032h	49.80	-0.39
100	00CBh	99.61	-0.39	0065h	99.61	-0.39
200	0198h	199.71	-0.15	00CBh	199.22	-0.39
500	03FFh	500.00	0.00	01FFh	500.00	0.00
1000	07FFh	1000.00	0.00	03FFh	1000.00	0.00
2000	0FFFh	2000.00	0.00	07FFh	2000.00	0.00
5000	27FFh	5000.00	0.00	13FFh	5000.00	0.00
10000	4FFFh	10000.00	0.00	27FFh	10000.00	0.00
20000	9FFFh	20000.00	0.00	4FFFh	20000.00	0.00
50000	—	—	—	C7FFh	50000.00	0.00

29.2.5 Low-Power Timer Compare Register 0 (LPCMR0)

Address(es): LPT.LPCMR0 0008 00B8h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	LPCMR0[15:0]	Low-Power Timer Compare 0	Set the value of compare match 0 for comparison with the low-power timer counter.	R/W

- Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.
- Note: Rewrite this register while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPCMR0 register is used to set the value of compare match 0 for comparison with the low-power timer counter. Set the LPCMR0 register to a value smaller than or equal to the value of the LPTPRD register.

29.2.6 Low-Power Timer Standby Wakeup Enable Register (LPWUCR)

Address(es): LPT.LPWUCR 0008 00BCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LPWKUPEN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	LPWKUPEN	Low-Power Timer Standby Wakeup Enable*1	0: Wakeup from software standby mode using low-power timer is disabled 1: Wakeup from software standby mode using low-power timer is enabled	R/W

Note: Set the PRCR.PRC2 bit to 1 (write enabled) before rewriting this register.

Note 1. Rewrite this bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

The LPWUCR register is used to enable the function that allows to return from software standby mode to normal operating mode when compare match 0 occurs in the low-power timer.

LPWKUPEN Bit (Low-Power Timer Standby Wakeup Enable)

This bit enables or disables the function that allows to return from software standby mode to normal operating mode when compare match 0 occurs in the low-power timer.

29.3 Operation

29.3.1 Periodic Counting Operation

The low-power timer is a 16-bit up-counter that operates regardless of the MCU operating mode*1.

When the LPTCR3.LPCNTEN bit is set to 1 (low-power timer counter operates) after setting the LPTCR1.LPCNTPSSEL[2:0] bits to select the division ratio, the LPTCR1.LPCNTCKSEL bit to select the clock source, and the LPTCR2.LPCNTSTP bit to 0 (clock is supplied to the low-power timer), the low-power timer counter starts counting with the selected clock.

When the value of the low-power timer counter matches the value of the LPTPRD register, the counter restarts counting from 0000h.

When the value of the low-power timer counter matches the value of the LPCMR0 register in software standby mode while the LPTCR1.LPCMRE0 bit is set to 1 (Compare match 0 is enabled) and the LPWUCR.LPWKUPEN bit is set to 1 (wakeup from software standby mode using low-power timer is enabled), the MCU returns from software standby mode to normal operating mode by the function of the event link controller (ELC).

Figure 29.2 shows operation of the low-power timer and Figure 29.3 shows an example of procedure for the initial settings.

Note 1. When the LPTCR1.LPCNTCKSEL bit is set to 1 (IWDT-dedicated clock), the counter stops because the IWDTCLK stops in the low-power consumption mode under the following settings:

The OFS0.IWDTSLCSTP bit is set to 1 (counting stop is enabled) while IWDT is activated in auto-start mode, or the IWDTCSTPR.SLCSTP bit is set to 1 (counting stop is enabled) while IWDT is activated in register start mode.

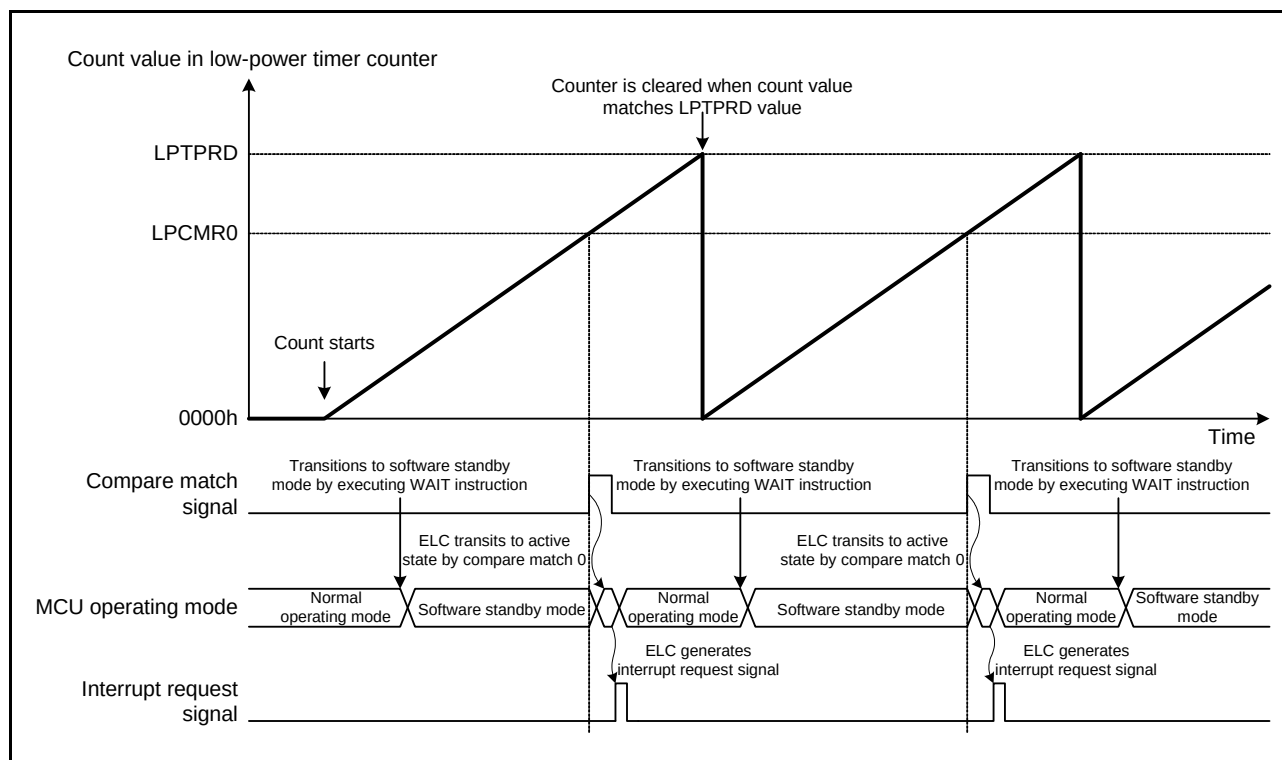


Figure 29.2 Operation of Low-Power Timer

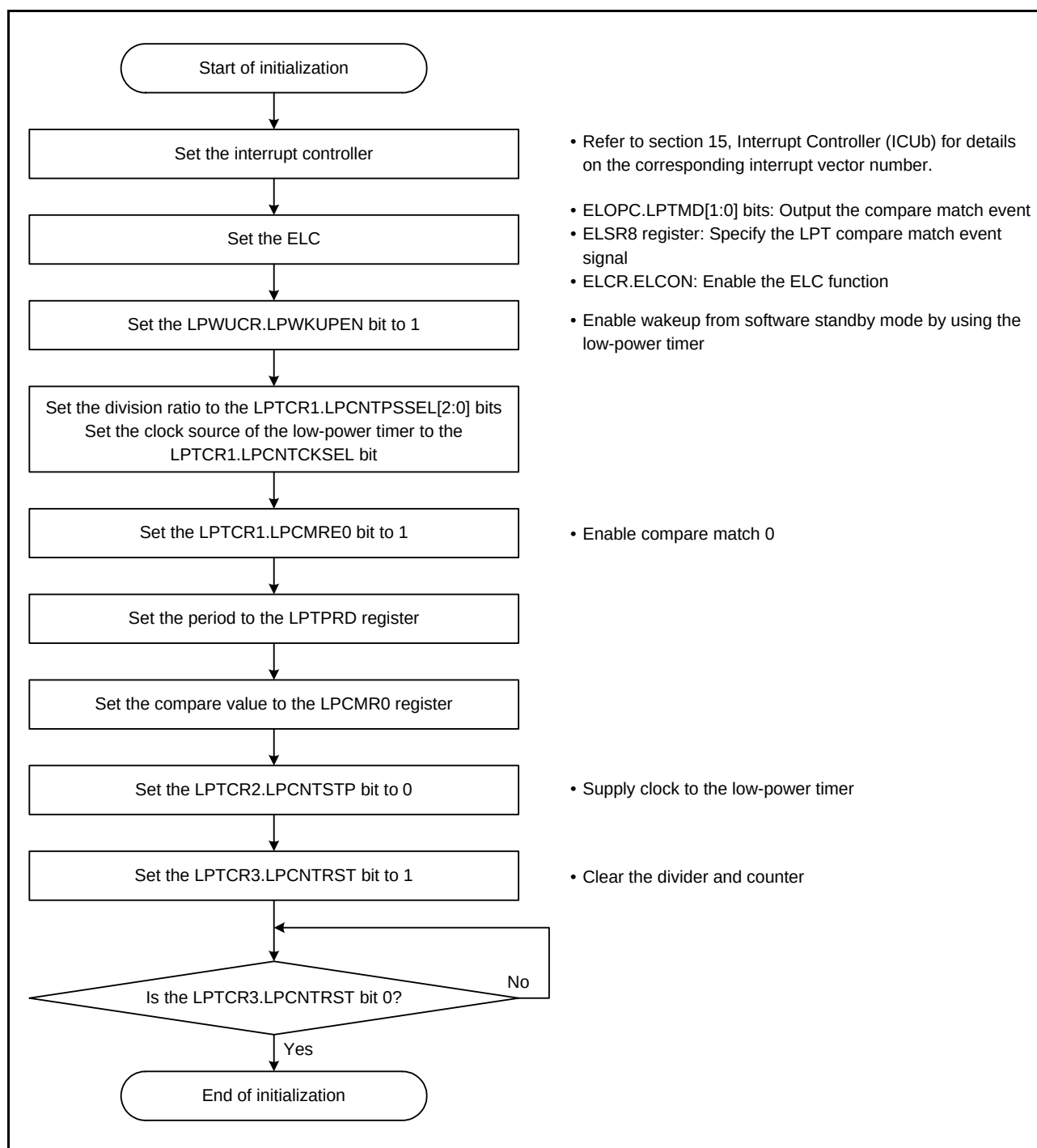


Figure 29.3 Example of Initial Settings

29.3.2 Count Timing of Low-Power Timer Counter

The LPTCR1.LPCNTPSSEL[2:0] bits are used to select the counter clock to be input to the low-power timer counter from among five clocks derived from dividing the clock source selected by the LPTCR1.LPCNTCKSEL bit by 2 to 32, respectively.

Figure 29.4 shows the count timing of the low-power timer counter in this case.

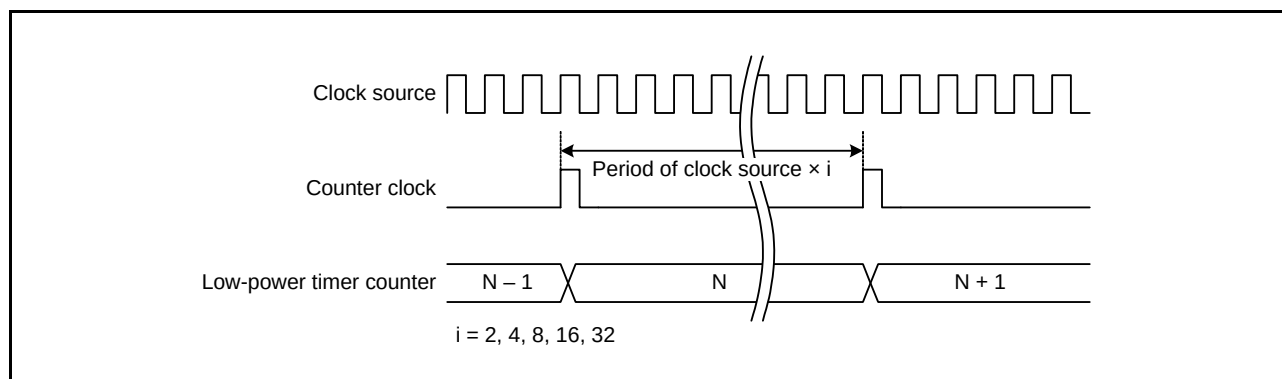


Figure 29.4 Count Timing of Low-Power Timer Counter

29.3.3 Clearing Timing of Low-Power Timer Counter

Writing 1 to the LPTCR3.LPCNTRST bit*1 clears the low-power timer counter.

This bit automatically becomes 0 when the clearing of the counter is completed.

Figure 29.5 shows the clearing timing of the low-power timer counter in this case.

Note 1. Write to the LPTCR3.LPCNTRST bit while the LPTCR3.LPCNTEN bit is 0 (low-power timer counter stops).

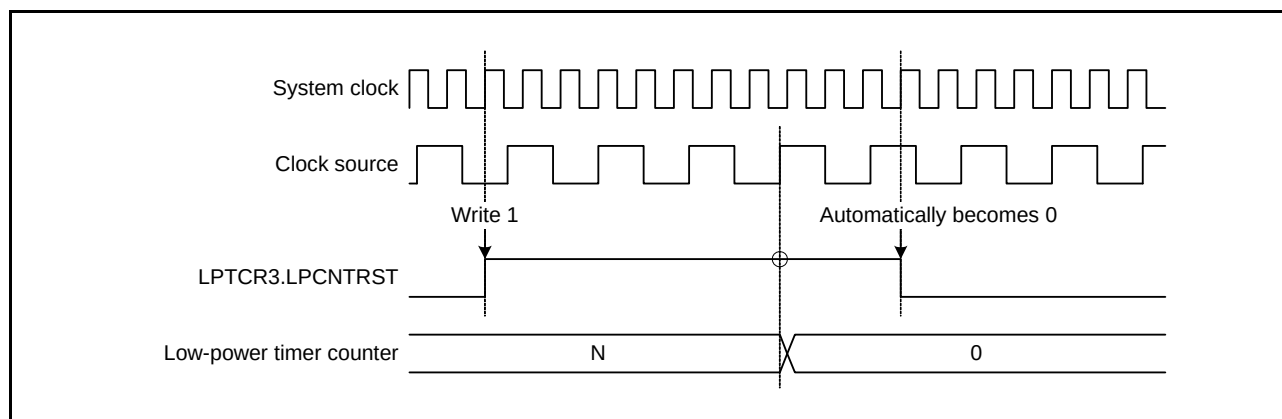


Figure 29.5 Clearing Timing of Low-Power Timer Counter

29.4 Wakeup from Software Standby Mode by an Interrupt through the Event Link Controller (ELC)

The low-power timer generates an event signal upon LPT compare match 0 to the event link controller (ELC) only in software standby mode.

Setting the ELOPC.LPTMD[1:0] bits to 00b (output the compare match 0 event to ICU as an interrupt request) and the ELSR8 register to 32h (LPT compare match 0) leads to the generation of an interrupt from the event signal, and the MCU returns from software standby mode to normal operating mode.

29.5 Usage Notes

29.5.1 Notes on Transition to Software Standby Mode

When the MCU is to re-enter to software standby mode after returning from software standby mode to normal operating mode, wait for at least one cycle of the clock selected by the LPTCR1.LPCNTCKSEL bit before executing the WAIT instruction.

30. Watchdog Timer (WDTA)

The watchdog timer (WDT) is a 14-bit down-counter. It can be used to reset this MCU when the counter underflows because its value cannot be refreshed due to the system being out of control.

In addition, a non-maskable interrupt can be generated by an underflow.

The refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control.

In this section, “PCLK” is used to refer to PCLKB.

30.1 Overview

Table 30.1 lists the specifications of the WDT and Figure 30.1 shows a block diagram of the WDT.

Table 30.1 WDT Specifications

Item	Specifications
Count source	Peripheral module clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	- Auto-start mode: Counting automatically starts after a reset is released - Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the WDTRR register)
Conditions for stopping the counter	- Reset (the down-counter and other registers return to their initial values) - In low power consumption states - A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer Reset sources	- Down-counter underflows - Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	- Down-counter underflows - Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.

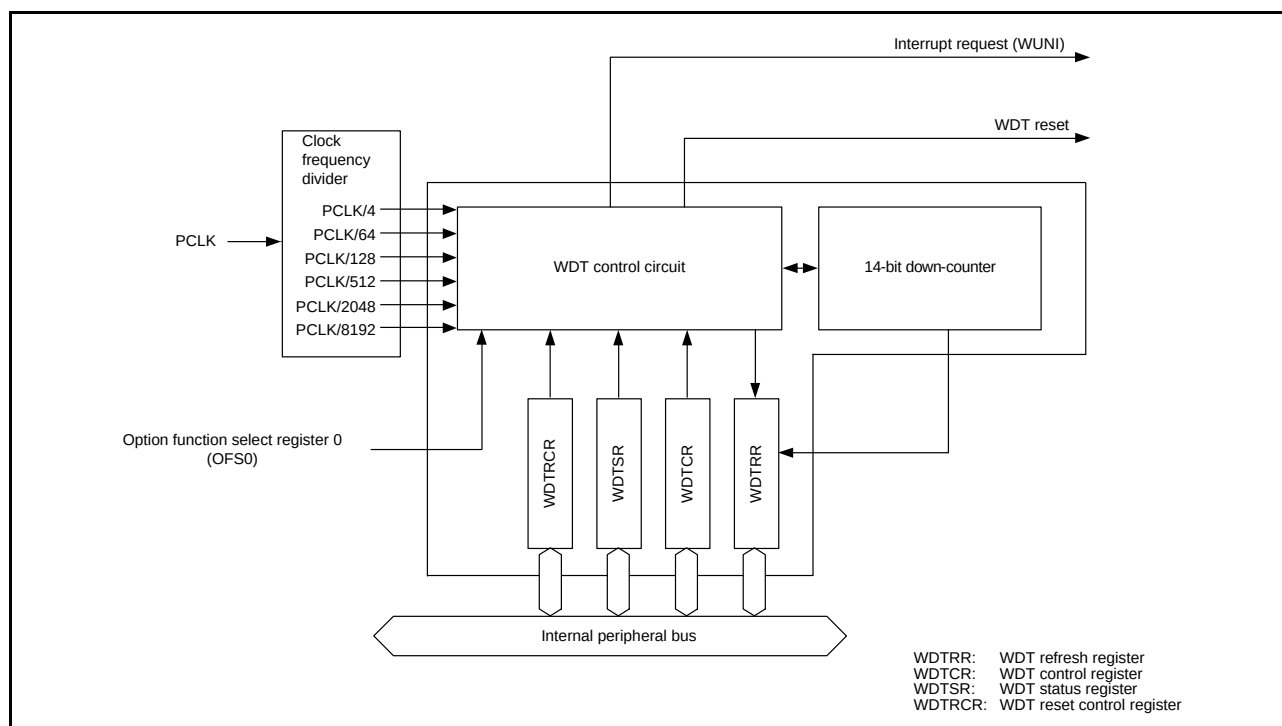
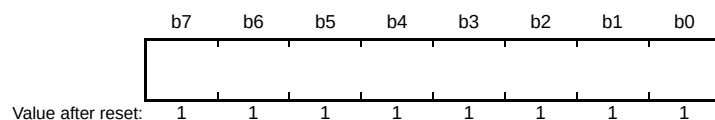


Figure 30.1 WDT Block Diagram

30.2 Register Descriptions

30.2.1 WDT Refresh Register (WDTRR)

Address(es): 0008 8020h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

WDTRR refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 00h and then writing FFh to WDTRR (refresh operation) within the refresh-permitted period.

After the down-counter has been refreshed, it starts counting down from the value selected by setting the WDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the WDTCR.TOPS[1:0] bits.

When 00h is written, the read value is 00h, when a value other than 00h is written, the read value is FFh.

For details of the refresh operation, refer to section 30.3.3, Refresh Operation.

30.2.2 WDT Control Register (WDTCR)

Address(es): 0008 8022h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	RPSS[1:0]	—	—	RPES[1:0]	—	—	CKS[3:0]	—	—	—	—	—	TOPS[1:0]	—
0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Setting other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0 and cannot be modified.	R

There are some restrictions on writing to the WDTCR register. For details, refer to section 30.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the settings in the WDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in OFS0 register. For details, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Selection)

These bits select the timeout period (period until the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLK cycles are listed in Table 30.2.

Table 30.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Divide-by-4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Divide-by-512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Divide-by-2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Divide-by-8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] Bits (Clock Division Ratio Selection)

These bits specify the division ration of the clock used for the down-counter. The division ration can be selected from among the peripheral module clock (PCLK) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLK clock can be selected for the WDT.

RPES[1:0] Bits (Window End Position Selection)

These bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] Bits (Window Start Position Selection)

These bits specify the window start position that indicates the refresh-permitted period. 25%, 50%, 75%, or 100% of the timeout period can be selected for the window end position. The window start position should be set to a value greater the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Table 30.3 lists the counter values for the window start and end positions and Figure 30.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 30.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

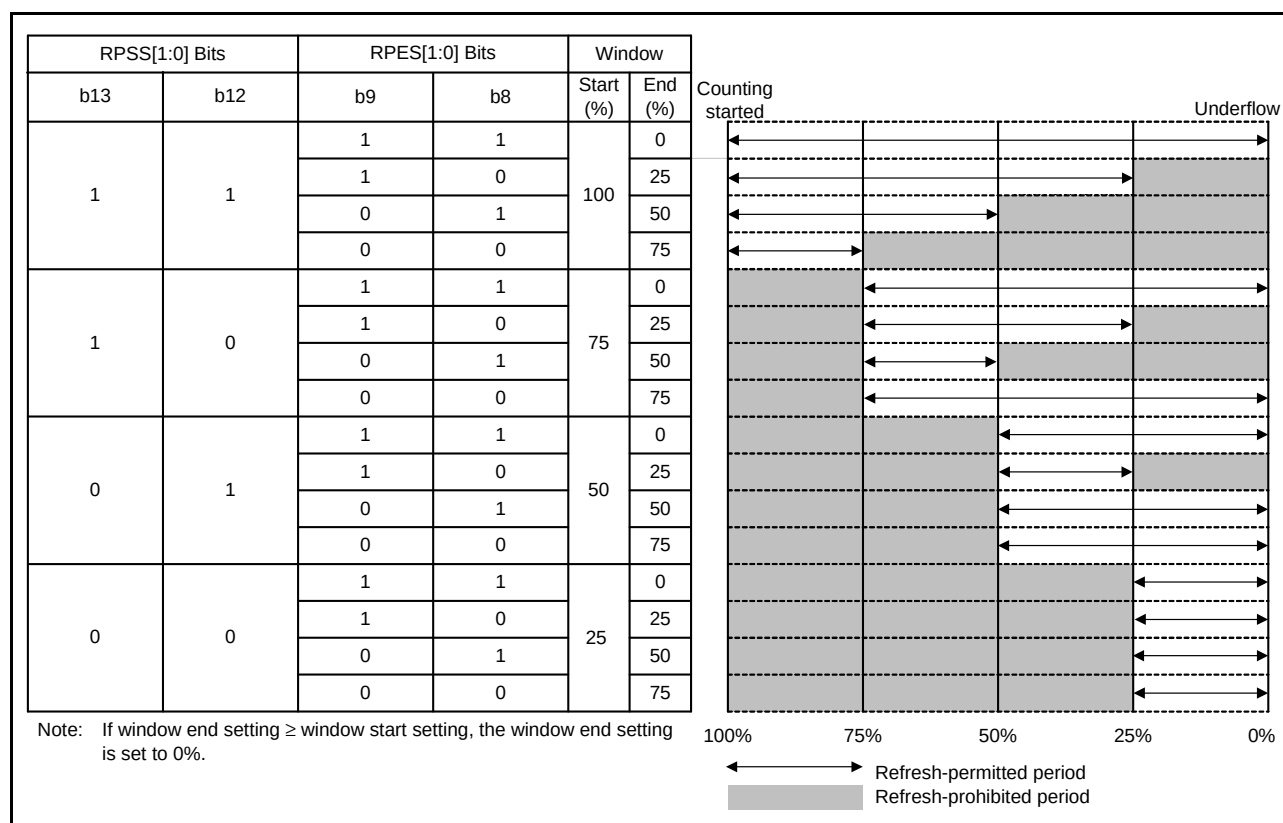
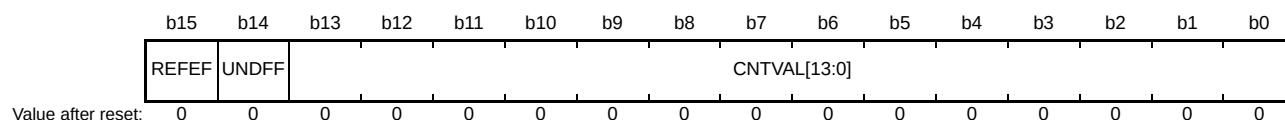


Figure 30.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

30.2.3 WDT Status Register (WDTSR)

Address(es): 0008 8024h



Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R(W) *1

Note 1. Only 0 can be written to clear the flag.

CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

Read this flag to confirm whether or not an underflow has occurred in the down-counter.

The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

Read this flag to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.

The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

30.2.4 WDT Reset Control Register (WDTRCR)

Address(es): 0008 8026h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIR QS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request output is enabled 1: Reset output is enabled	R/W

There are some restrictions on writing to the WDTRCR register. For details, refer to section 30.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the WDTRCR register settings are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

30.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 30.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

30.3 Operation

30.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after a reset is released, and register start mode, in which counting is started by refresh operation (writing to the register).

In auto-start mode, counting automatically starts after a reset is released in accordance with the settings in option function select register 0 (OFS0) in the ROM.

In register start mode, counting is started by refresh operation (writing to the register) after the respective registers are set after a reset is released.

Select auto-start mode or register start mode by setting the OFS0.WDTSTRT bit.

When the auto-start mode is selected, the settings in the WDTCCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.

On the other hand, when the register start mode is selected, the setting of the OFS0 register is disabled, and the settings of the WDTCCR and WDTRCR registers are enabled.

30.3.1.1 Register Start Mode

When the OFS0.WDTSTRT bit is 1, register start mode is selected, and the WDTCCR and WDTRCR registers are enabled.

After a reset is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCCR register, and the reset output or interrupt request output in the WDTRCR register. Then, refresh the down-counter to start counting down from the value set by the WDTCCR.TOPS[1:0] bits.

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or a non-maskable interrupt request (WUNI). Reset output or interrupt request output can be selected by setting the WDTRCR.RSTIRQS bit.

Figure 30.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCCR.RPES[1:0] = 10b)

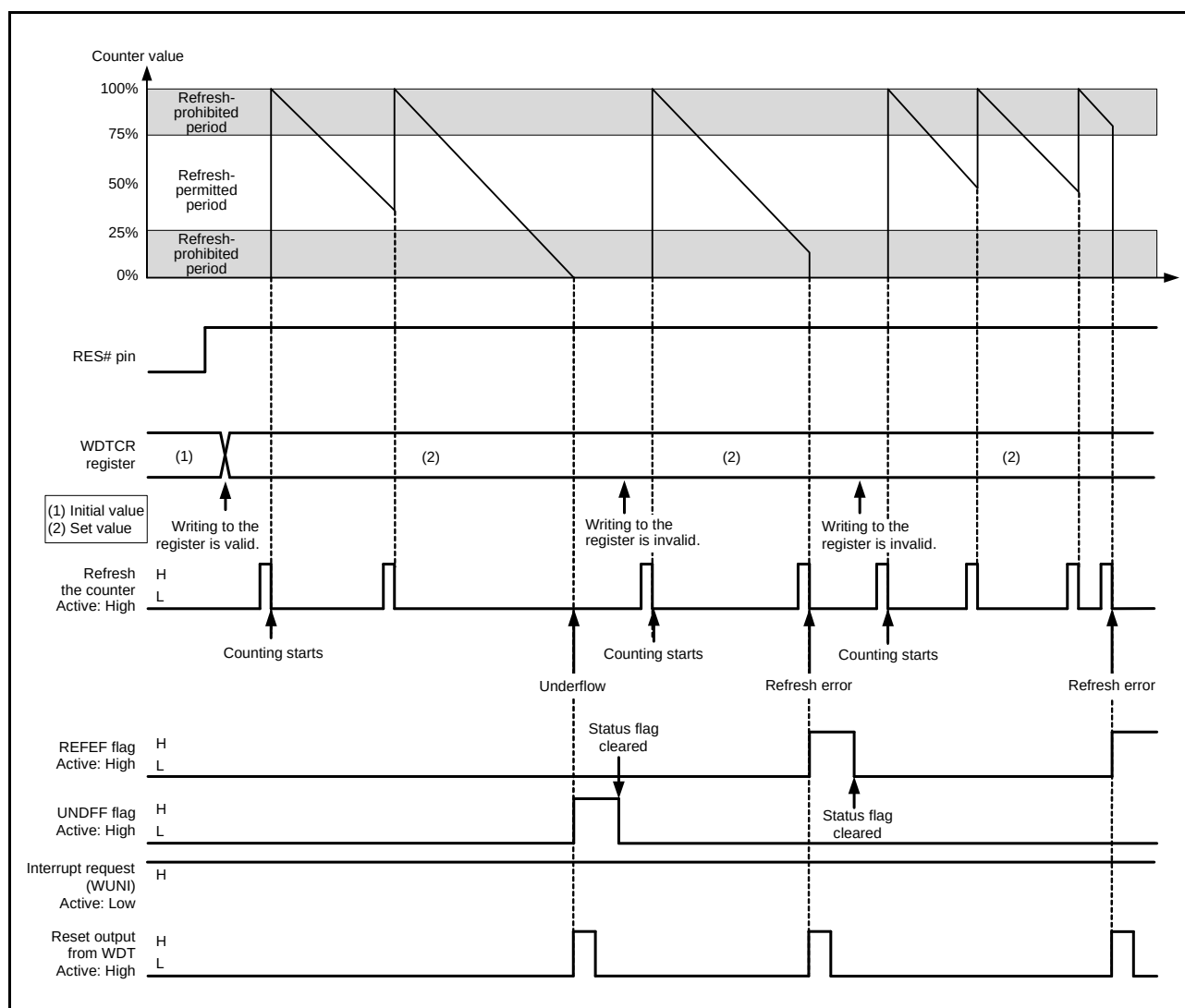


Figure 30.3 Operation Example in Register Start Mode

30.3.1.2 Auto-Start Mode

When the WDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.

Within the reset state, the setting values (clock division ratio, window start and end positions, timeout period, and reset output or interrupt request) of the OFS0 register are set in the WDT registers.

When the reset is released, the down-counter automatically starts counting down from the value set by the OFS0.WDTTOPS[1:0] bits.

After that, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues.

However, if the down-counter underflows because refreshing of the down-counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request (WUNI).

After the reset signal or non-maskable interrupt request is output of for one cycle of counting, the value of the timeout period is set in the down-counter counting is restarted.

Reset output or interrupt request output can be selected by setting the OFS0.WDTRSTIRQS bit.

Figure 30.4 shows an example of operation (non-maskable interrupt) under the following conditions.

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)

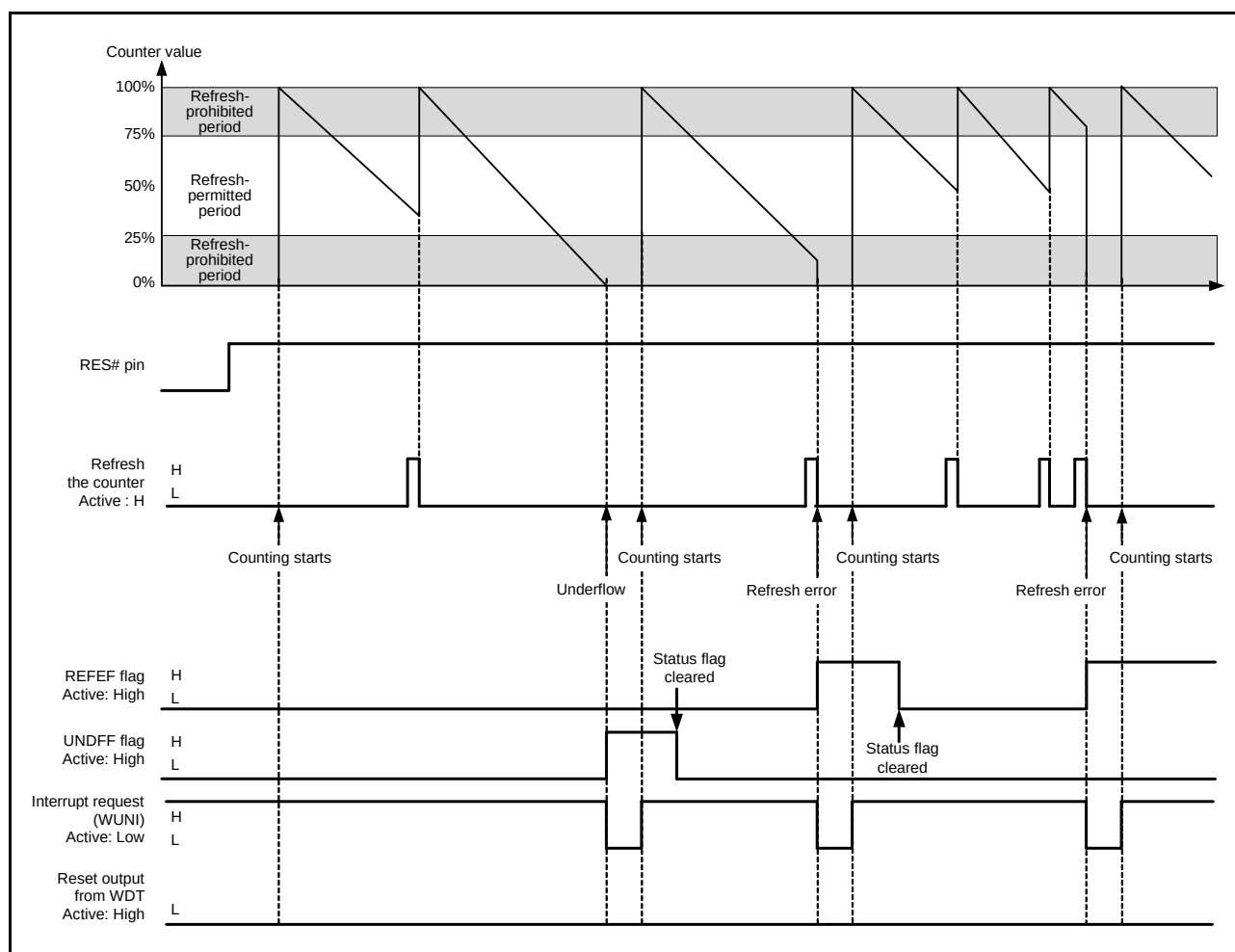


Figure 30.4 Operation Example in Auto-Start Mode

30.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDTCR or WDTRCR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the WDTCR or WDTRCR register, the protection signal in the WDT becomes 1 to protect the WDTCR and WDTRCR registers against subsequent attempts at writing.

This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 30.5 shows control waveforms produced in response to writing to the WDTCR register.

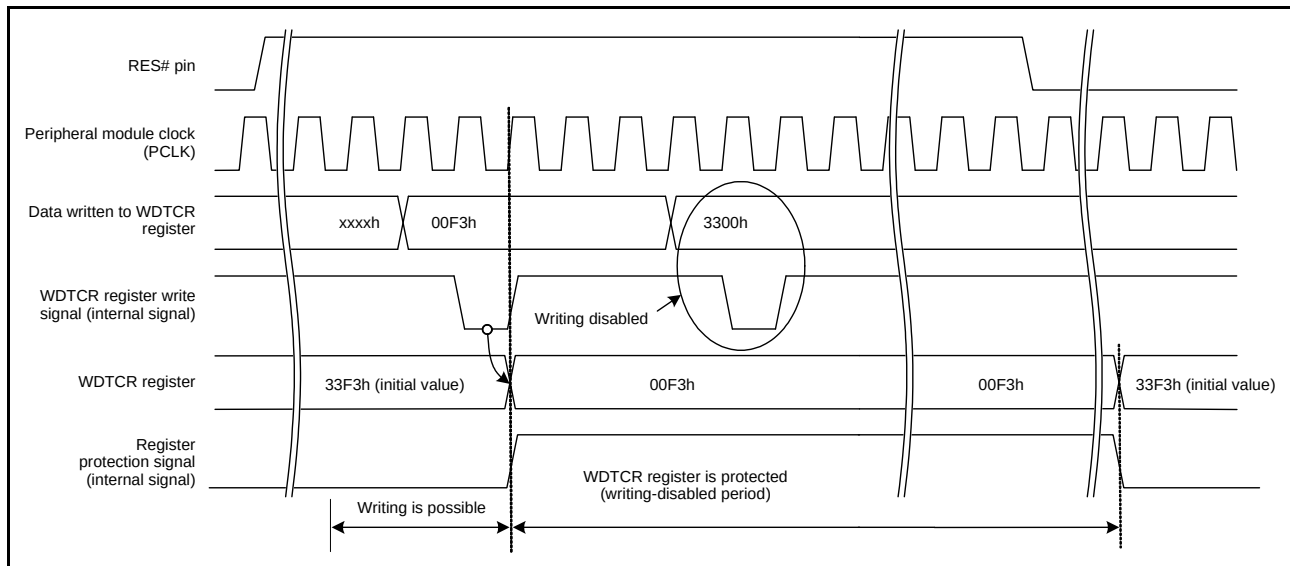


Figure 30.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

30.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and then FFh to the WDTRR register. If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing to 00h and then FFh to the WDTRR register.

Even if a register other than the WDTRR register is accessed or the WDTRR register is read between writing 00h and writing FFh to the WDTRR register, correct refreshing will be done.

Writing to refresh the counter must be performed within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when writing FFh. For this reason, correct refreshing will be done even if 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n-1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the WDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

After FFh is written to the WDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR register should be completed four-count cycles before the down-counter underflows.

Figure 30.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

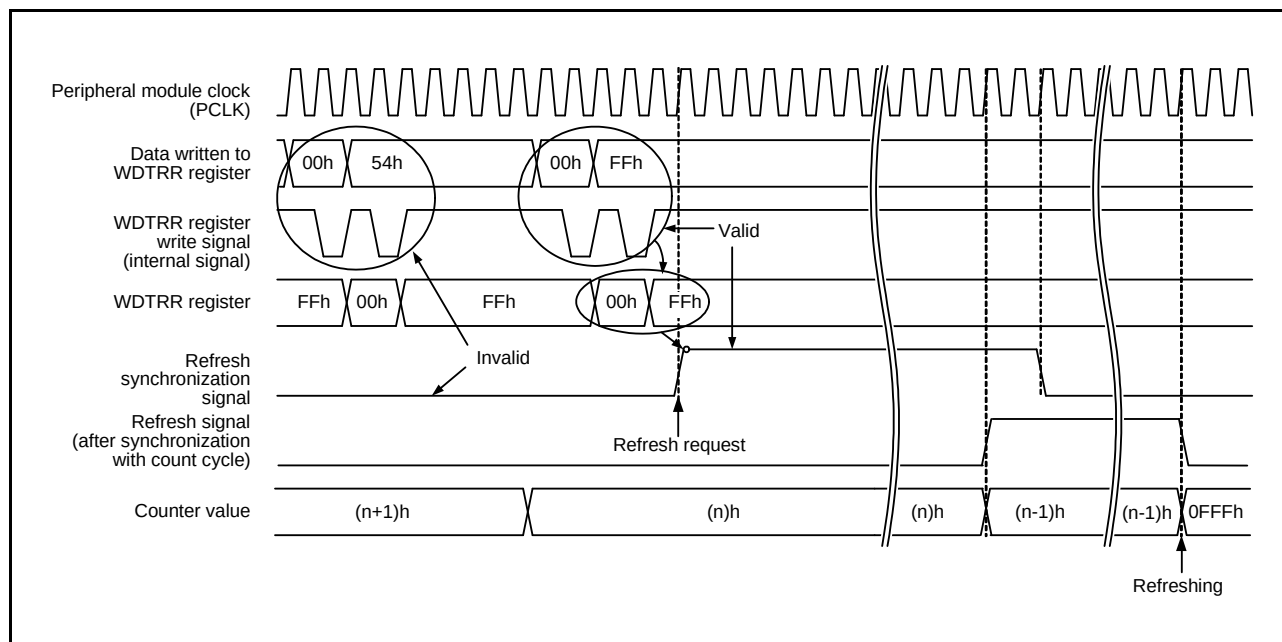


Figure 30.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

30.3.4 Reset Output

When the WDTCR.RSTIRQS bit is set to 1 in register start mode or when the WDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.

In auto-start mode, counting down automatically starts after the reset is released.

30.3.5 Interrupt Source

When the WDTCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.WDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUb).

Table 30.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

30.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the WDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 30.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

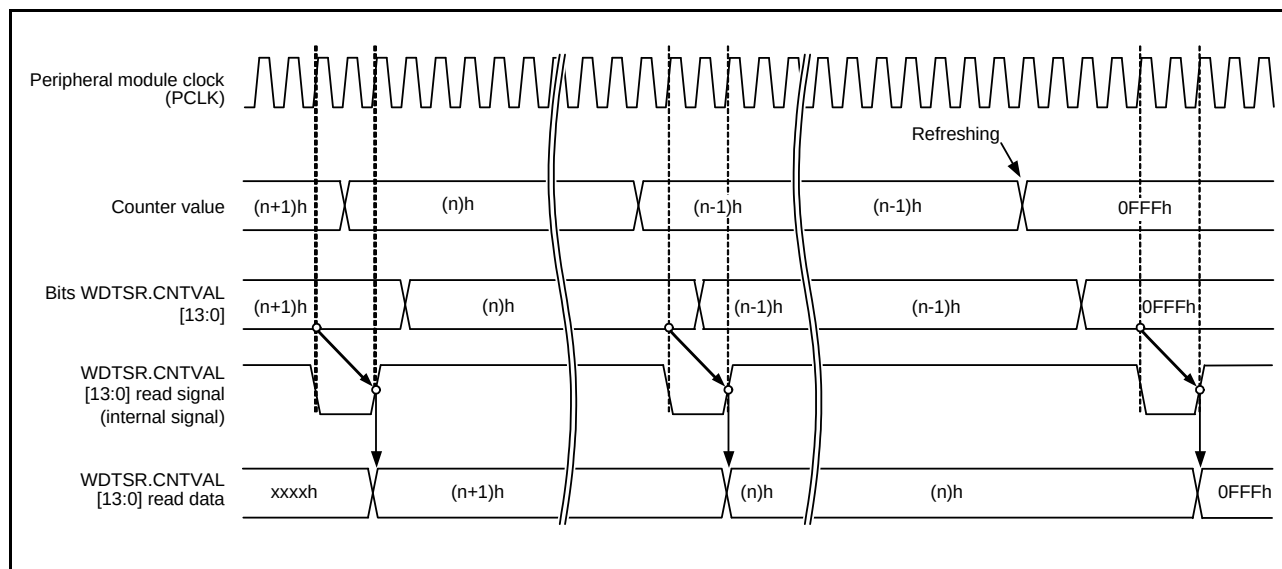


Figure 30.7 Processing for Reading WDT Down-Counter Value
(WDTCR.CKS[3:0] = 0100b, WDTCR.TOP[1:0] = 01b)

30.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 30.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during WDT operation.

For details on the OFS0 register, refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 30.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Enabled in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTPSS[1:0]	WDTCR.TOP[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTCR.RSTIRQS

31. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.

The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.

The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, or deep sleep mode, the IWDTCSLSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 31.1 lists the specifications of the IWDT and Figure 31.1 shows a block diagram of the IWDT.

Table 31.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • In low power consumption states (depends on the register setting*2) • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (IWDTCSLSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) $\geq 4 \times$ (the frequency of the count source after divide).

Note 2. When the OFS0.IWDTSLCSTP bit is 1 in auto-start mode, and when the IWDTCSLSTP bit is 1 in register start mode.

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.

Figure 31.1 is a block diagram of the IWDT.

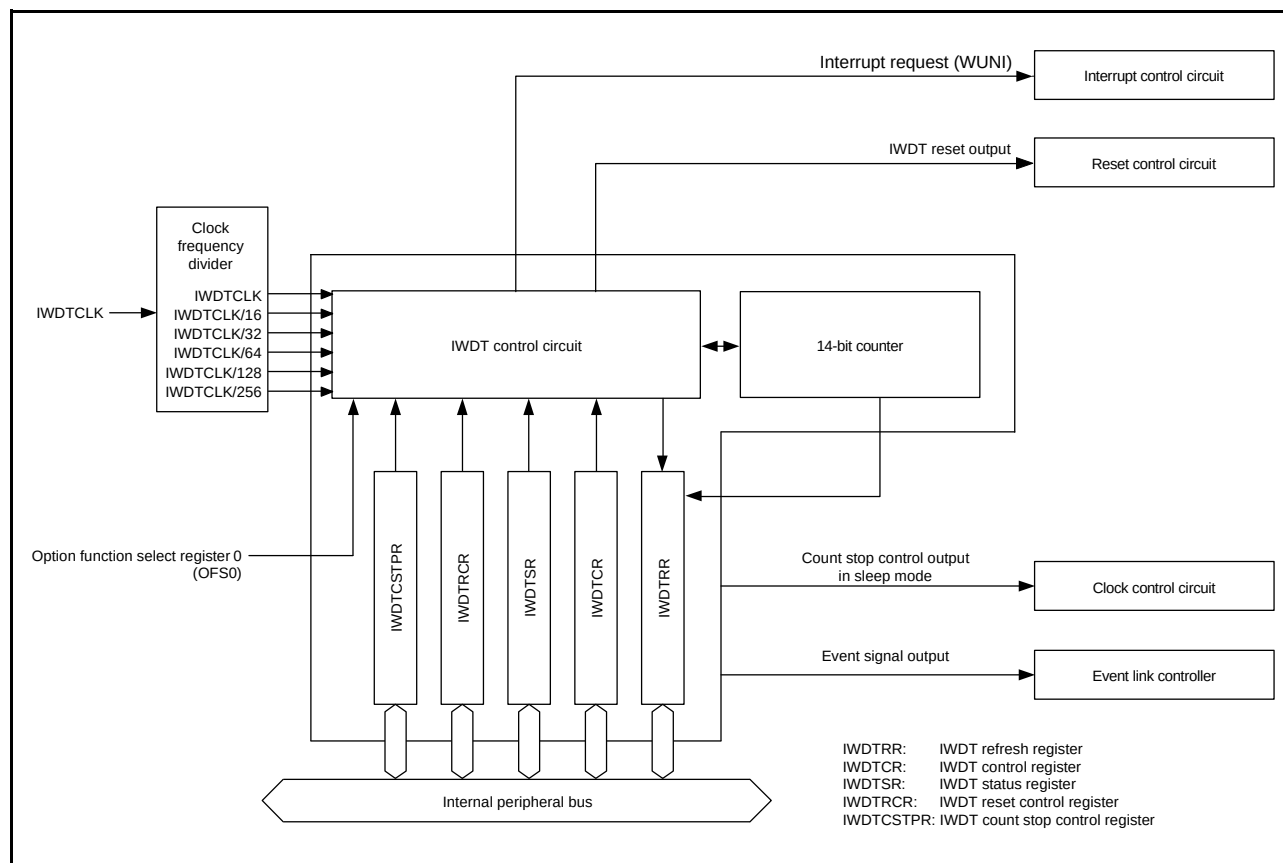
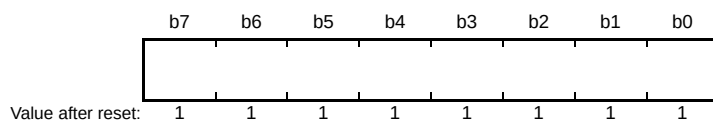


Figure 31.1 IWDT Block Diagram

31.2 Register Descriptions

31.2.1 IWDt Refresh Register (IWDTRR)

Address(es): IWDt.IWDTRR 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDt.

The counter of the IWDt is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.

After the counter has been refreshed, it starts counting down from the value selected by the IWDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the IWDTCR.TOPS[1:0] bits in the first refresh operation after a reset is released.

When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.

For details of the refresh operation, refer to section 31.3.3, Refresh Operation.

31.2.2 IWDt Control Register (IWDTCR)

Address(es): IWDt.IWDTCR 0008 8032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b6 b5 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDt Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCCLK cycles are listed in Table 31.2.

Table 31.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCCLK
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	No division	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	Divide-by-16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	Divide-by-32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	Divide-by-64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	Divide-by-128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	Divide-by-256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCCLK clock can be selected for the IWDTC.

RPES[1:0] Bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.

Table 31.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

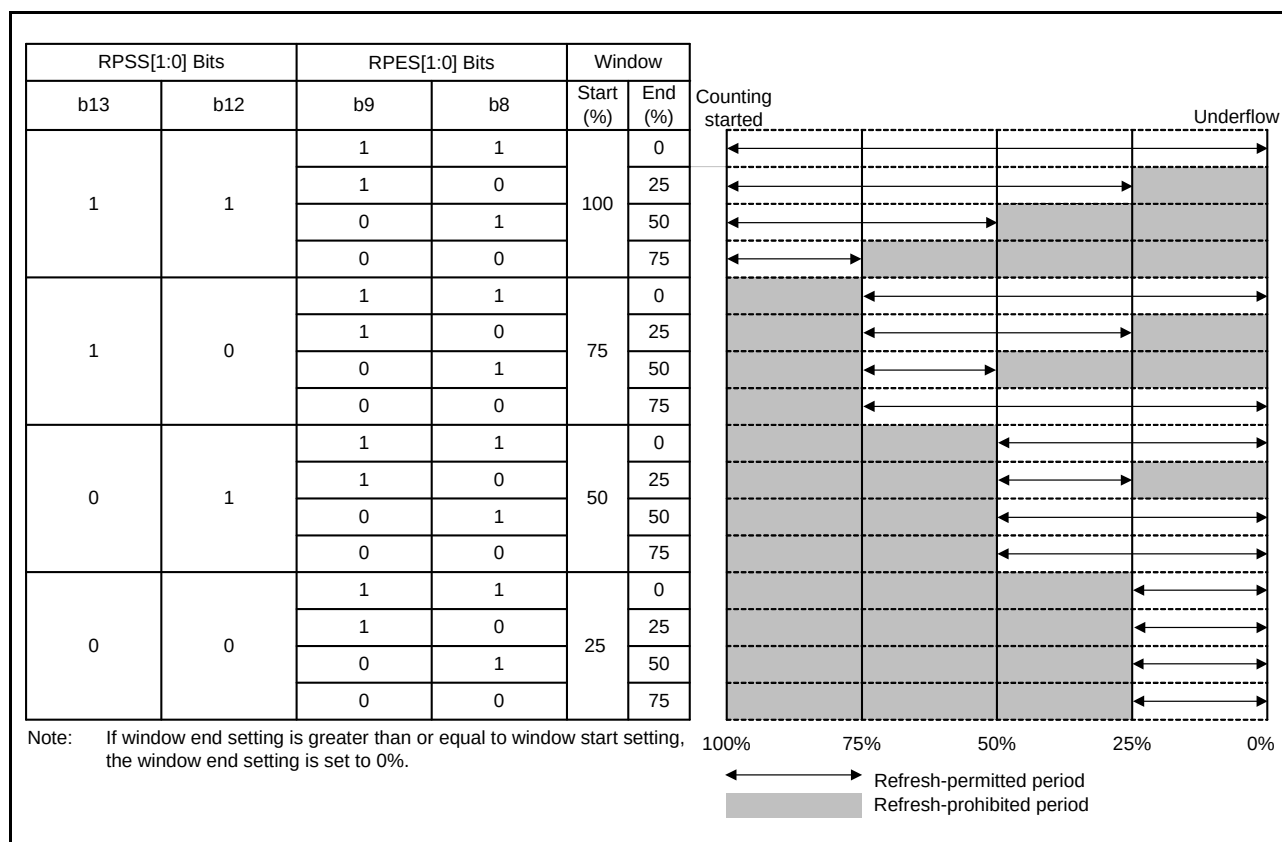
Table 31.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

RPSS[1:0] Bits (Window Start Position Select)

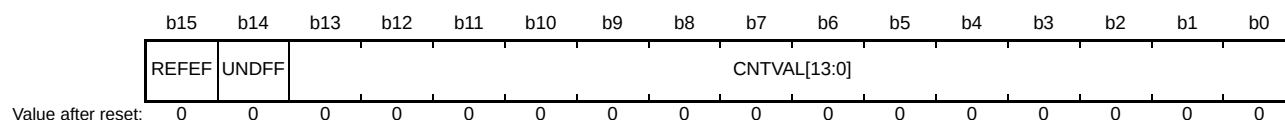
These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.

Figure 31.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

**Figure 31.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period**

31.2.3 IWDt Status Register (IWDTSR)

Address(es): IWDt.IWDTSR 0008 8034h



Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

The IWDTSR register is initialized by the reset source of the IWDt. The IWDTSR register is not initialized by other reset sources.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.

The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed. Write 0 to the UNDF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).

The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred. Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

31.2.4 IWDt Reset Control Register (IWDTRCR)

Address(es): IWDt.IWDTRCR 0008 8036h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIR QS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDt Registers.

31.2.5 IWDT Count Stop Control Register (IWDTCSSTPR)

Address(es): IWDT.IWDTCSSTPR 0008 8038h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCST P	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, or deep sleep mode.	R/W

The IWDTCSSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSSTPR register. For details, refer to section 31.3.2, Control over Writing to the IWDTCSR, IWDTRCR, and IWDTCSSTPR Registers.

In auto-start mode, the IWDTCSSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSSTPR register can also be made in the OFS0 register. For details, refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or deep sleep mode.

31.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 31.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

31.3 Operation

31.3.1 Count Operation in Each Start Mode

Select the IWDt start mode by setting the IWDtSTRT bit in option function select register 0 (OFS0).

When the OFS0.IWDtSTRT bit is 1 (register start mode), the IWDtCR, IWDtRCR, and IWDtCSTPR registers are enabled, and counting is started by refresh operation (writing) to the IWDtRR register. When the OFS0.IWDtSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

31.3.1.1 Register Start Mode

When the OFS0.IWDtSTRT bit in option function select register 0 is 1, register start mode is selected, and the IWDtCR, IWDtRCR, and IWDtCSTPR registers are enabled.

After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDtCR register, the reset output or interrupt request output in the IWDtRCR register, and the counter stop control at transitions to low power consumption states in the IWDtCSTPR register. Then refresh the counter to start counting down from the value selected by setting the IWDtCR.TOPS[1:0] bits.

Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDt does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDt outputs a reset signal or a non-maskable interrupt request (WUNI). Set the IWDtRCR.RSTIRQS bit to select either reset output or interrupt request output.

Figure 31.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.IWDtSTRT = 1)
- Reset output is enabled (IWDtRCR.RSTIRQS = 1)
- The window start position is 75% (IWDtCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDtCR.RPES[1:0] = 10b)

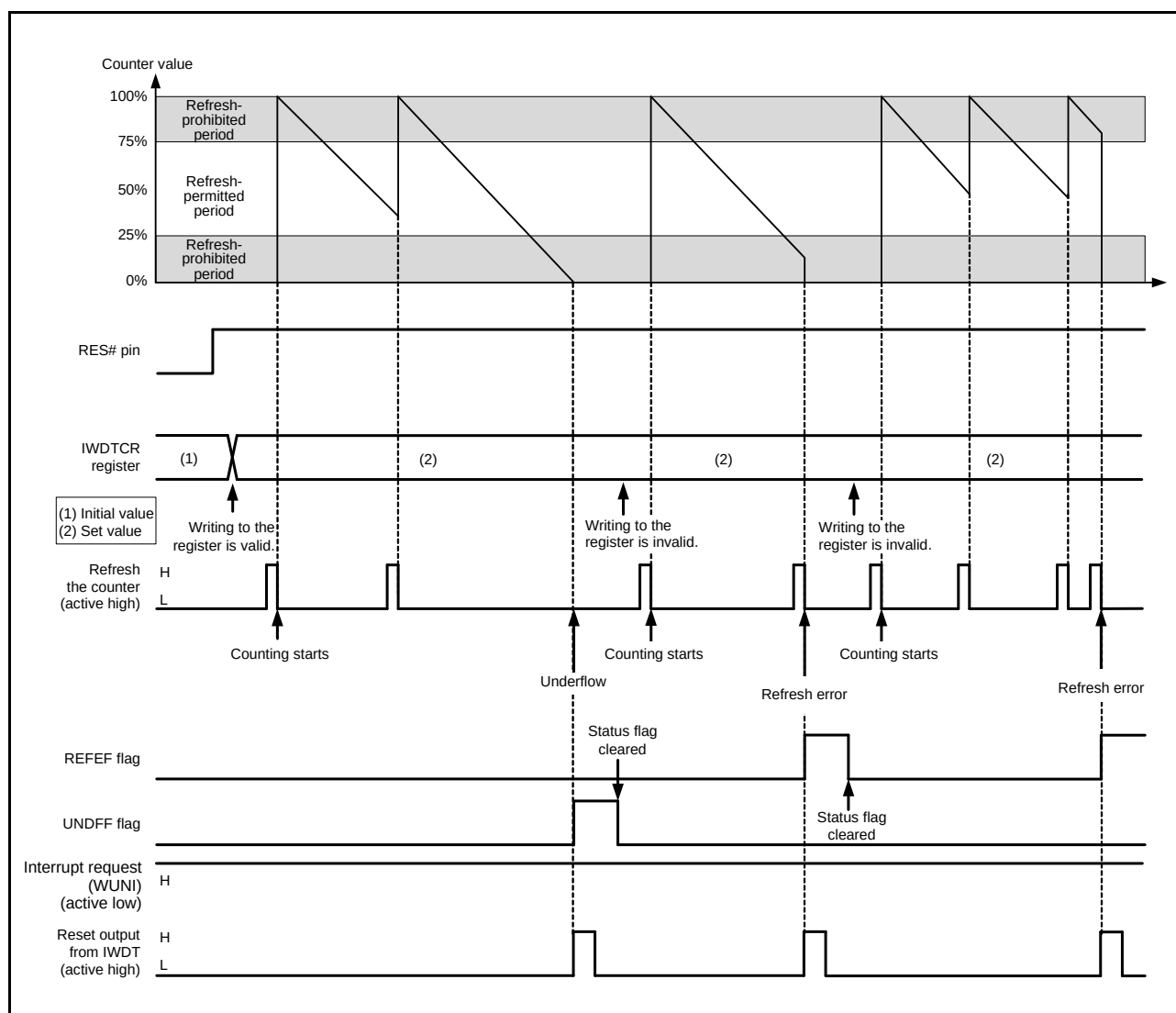


Figure 31.3 Operation Example in Register Start Mode

31.3.1.2 Auto-Start Mode

When the IWDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are disabled.

Within the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in the OFS0 register. When the reset is released, the counter automatically starts counting down from the value selected by the OFS0.IWDTTOPS[1:0] bits.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDt does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDt outputs the reset signal or non-maskable interrupt request (WUNI). After the reset signal or non-maskable interrupt request (WUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the OFS0.IWDTRSTIRQS bit to select either reset output or interrupt request output.

Figure 31.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

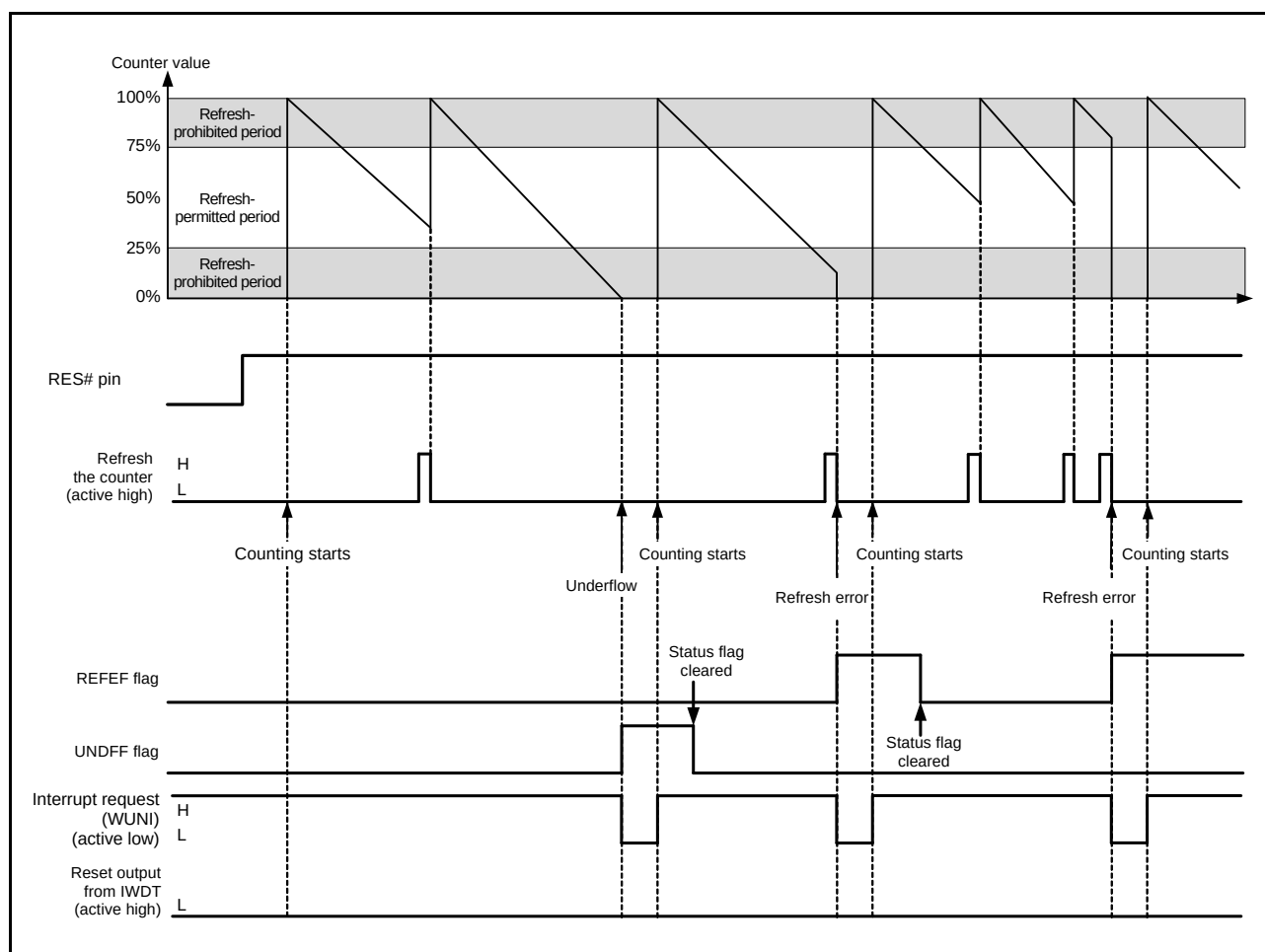


Figure 31.4 Operation Example in Auto-Start Mode

31.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDTCR, IWDTRCR, or IWDTCSTPR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDt becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.

This protection is released by the reset source of the IWDt. With other reset sources, the protection is not released.

Figure 31.5 shows control waveforms produced in response to writing to the IWDTCR register.

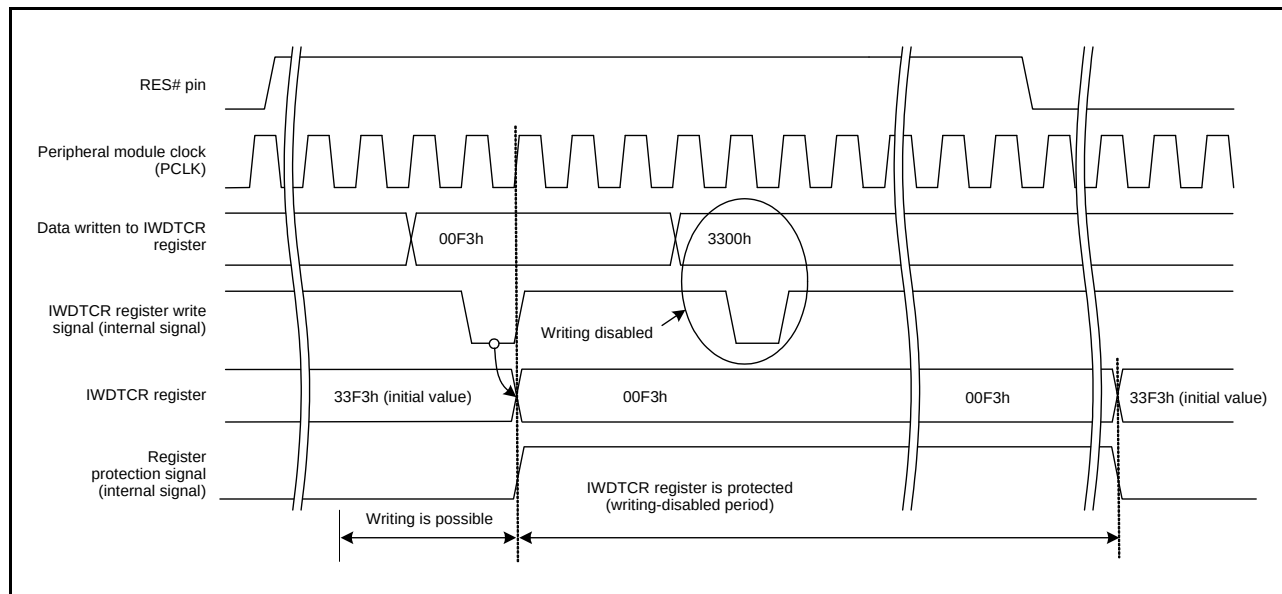


Figure 31.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

31.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDTRR register.

When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n-1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n-1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.

After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the IWDt-dedicated clock (IWDtCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the IWDTSR.CNTVAL[13:0] bits.

[Sample refreshing timings]

- When the window start position is set to 03FFh, even if 00h is written to the IWDTRR register before 03FFh is reached (0402h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 03FFh.
- When the window end position is set to 03FFh, refreshing is done if 0403h (four-count cycles before 03FFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 31.6 shows the IWDT refresh-operation waveforms when $PCLK > IWDTCCLK$ and clock divide ratio = $IWDTCCLK$.

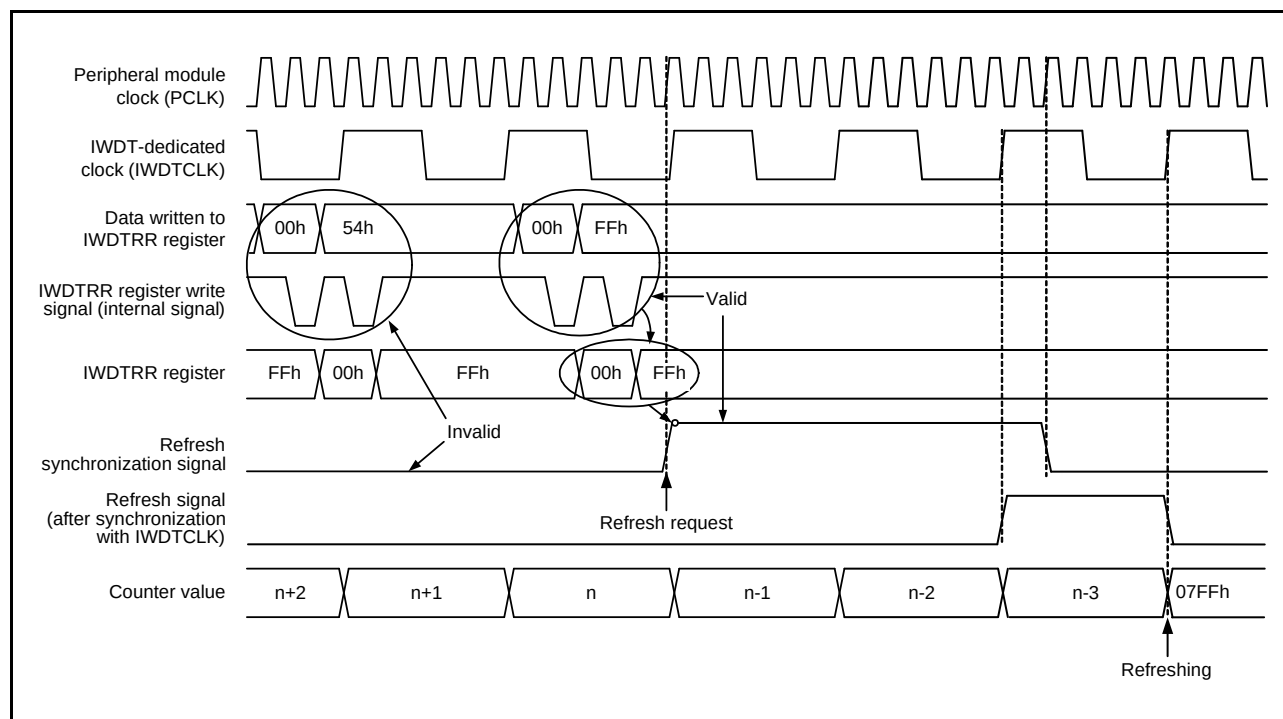


Figure 31.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

31.3.4 Status Flags

The IWDTSR.REFEF and IWDTSR.UNDF flags retain the source of the reset signal output from the IWDt or the source of the interrupt request from the IWDt.

Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDF flags to check for the reset or interrupt source.

For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDt, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.

After 0 is written to each flag, up to three IWDtCLK cycles and two PCLK cycles are required before the value is reflected.

31.3.5 Reset Output

When the IWDTRCR.RSTIRQS bit is set to 1 in register start mode or when the IWDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.

In register start mode, the counter is initialized (0000h) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.

In auto-start mode, counting down automatically starts after the reset output.

31.3.6 Interrupt Sources

When the IWDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.IWDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUb).

Table 31.4 IWDt Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Counter underflow Refresh error	Not possible	Not possible

31.3.7 Reading the Counter Value

As the counter in IWDt-dedicated clock (IWDtCLK), the counter value cannot be read directly. The IWDt synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the IWDtSR.CNTVAL[13:0] bits. Thus, the counter value can be checked indirectly through the IWDtSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 31.7 shows the processing for reading the IWDt counter value when $PCLK > IWDtCLK$ and clock divide ratio = IWDtCLK.

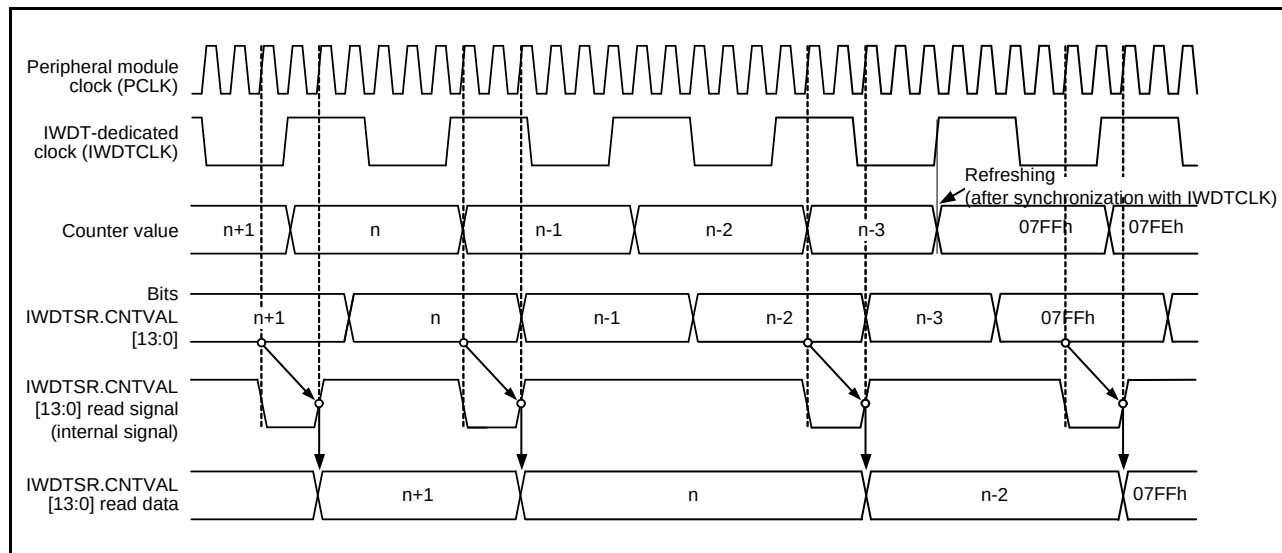


Figure 31.7 Processing for Reading IWDt Counter Value
(IWDtCR.CKS[3:0] = 0000b, IWDtCR.TOPS[1:0] = 11b)

31.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 31.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during IWDT operation.

For details on the OFS0 register, refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 31.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDRSTIRQS	IWDTCCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCTPR.SLCSTP

31.4 Link Operation by ELC

The event link controller (ELC) can use the interrupt request signal generated by the IWDT as the event signal.

Therefore, the ELC generates an event to the module specified previously when the IWDT outputs an interrupt request.

The event signal is output by the counter underflow and refresh error.

An event signal is output regardless of the setting of the IWDTCCR.RSTIRQS bit in register start mode or the OFS0.IWDRSTIRQS bit in auto-start mode. An event signal can also be output upon generation of the next interrupt source while the IWDTSR.REFEF or IWDTSR.UNDF flag is 1.

For details, see section 20, Event Link Controller (ELC).

31.5 Usage Notes

31.5.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

31.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock ($PCLK \geq 4 \times$ (the frequency of the count source after divide)).

32. USB 2.0 Host/Function Module (USBc)

32.1 Overview

This MCU incorporates a USB 2.0 host/function module (USB0).

The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0. It also supports Battery Charging Specification Revision 1.2.

The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.

Table 32.1 shows the specifications of the USB.

Table 32.1 USB Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (one channel) • The host controller and the function controller can be switched by software. • Self-power mode or bus power mode can be selected. • BC1.2 (Battery Charging Specification Revision 1.2) is supported. When the host controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers When the function controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• Buffer memory for USB communication is provided. • Up to 10 pipes can be selected (including the default control pipe). • Pipes 1 to 9 can be assigned any endpoint number. Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer, 64-byte single buffer • Pipes 1 and 2: 64-byte double buffer can be specified for bulk transfer 256-byte double buffer for isochronous transfer • Pipes 3 to 5: Bulk transfer, 64-byte double buffer • Pipes 6 to 9: Interrupt transfer, 64-byte single buffer
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK) • On-chip pull-up and pull-down resistors of D+/D-
Low power consumption function	Module-stop state can be set.

Figure 32.1 shows a block diagram of the USB.

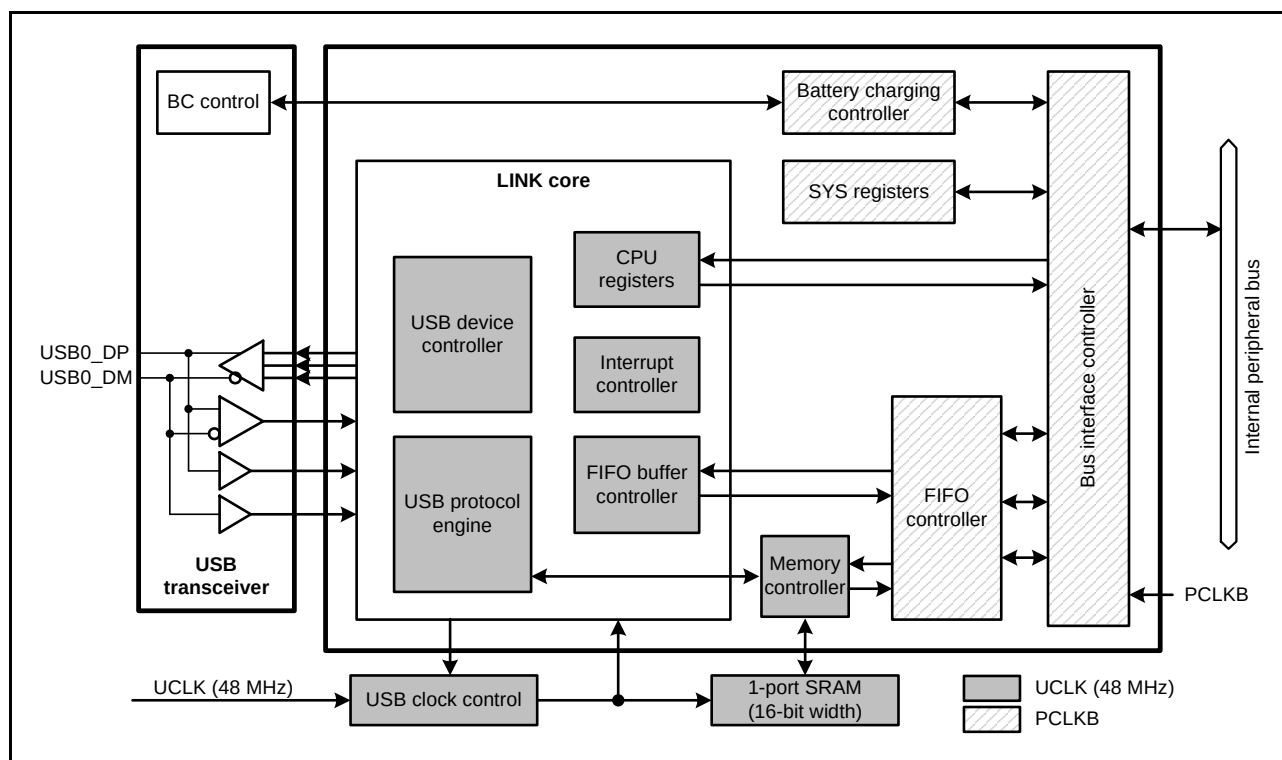


Figure 32.1 USB Block Diagram

Table 32.2 lists the I/O pins of the USB.

Table 32.2 USB Pin Configuration

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB0_DM	I/O	D- I/O pin of the USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
	USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	miniAB connector ID input signal should be connected to this pin during operation in OTG mode.
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

32.2 Register Descriptions

32.2.1 System Configuration Control Register (SYSCFG)

Address(es): USB0.SYSCFG 000A 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	CNEN	—	DCFM	DRPD	DPRPU	DMRPU	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	DMRPU	D– Line Resistor Control*1	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b4	DPRPU	D+ Line Resistor Control*1	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CNEN	CNEN Single End Receiver Enable	0: Single end receiver operation is disabled. 1: Single end receiver operation is enabled.	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*2	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not enable the DMRPU and DPRPU bits at the same time.

Note 2. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 32.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stabilized.

Table 32.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	—	The value is retained when the host controller is selected.
	USBINDX	—	The value is retained when the host controller is selected.
	USBLENG	—	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.

DMRPU Bit (D– Line Resistor Control)

The DMRPU bit enables or disables pulling up the D– line when the function controller is selected.

When the DMRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D– line to notify the USB host of connection as a low-speed device. Modifying the DMRPU bit from 1 to 0 allows the USB to release the D– line, thus notifying the USB host of disconnection.

This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.

When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection as a full-speed device. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.

This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.

This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.

This bit should be modified when the DMRPU, DPRPU, and DRPD bits are all 0.

CNEN Bit (CNEN Single End Receiver Enable)

Setting the CNEN bit to 1 allows the USB module to enable the single end receiver and set the LNST[1:0] flags to monitor the status of the D+ and D– lines.

The CNEN bit is used when the USB module operates as a portable device for battery charging.

SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.

When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

32.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USB0.SYSSTS0 000A 0004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVCMON[1:0]	—	—	—	—	—	—	—	—	HTACT	—	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Refer to Table 32.4.	R
b2	IDMON	ID Input Pin Monitor Flag	0: USB0_ID pin is low 1: USB0_ID pin is high	R
b5 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b6	HTACT	USB Host Sequencer Status Monitor Flag	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	OVRCURA/OVRCURB Input Pin Monitor Flag	The OVCMON[1] flag indicates the status of the USB0_OVRCURA pin. The OVCMON[0] flag indicates the status of the USB0_OVRCURB pin.	R

Note 1. Depends on the status of the USB0_OVRCURA/USB0_OVRCURB and USB0_ID pins.

LNST[1:0] Flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines (D+ and D– lines). Refer to Table 32.4.

The LNST[1:0] flags should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

HTACT Flag (USB Host Sequencer Status Monitor Flag)

The HTACT flag is 0 when the host sequencer of the USB is completely stopped.

OVCMON[1:0] Flags (OVRCURA/OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of overcurrent from an external power supply chip.

Table 32.4 Status of USB Data Bus Lines (D+ Line, D– Line)

LNST[1:0] Flags	During Low-Speed Operation	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

32.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USB0.DVSTCTR0 000A 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HNPBTOA	EXICEN	VBUSEN	WKUP	RWUPE	USBRS	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	- When the host controller is selected b2 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection - When the function controller is selected b2 b0 0 0 0: Communication speed not determined 0 0 1: USB bus reset in progress or low-speed connection 0 1 0: USB bus reset in progress or full-speed connection	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRS	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	VBUSEN Output Pin Control	0: USB0_VBUSEN pin outputs low. 1: USB0_VBUSEN pin outputs high.	R/W
b10	EXICEN	EXICEN Output Pin Control	0: USB0_EXICEN pin outputs low. 1: USB0_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

RHST[2:0] Flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the status of the USB bus reset.

When the host controller is selected, the RHST[2:0] flags indicate 100b after the USBRST bit has been set to 1 by software.

The USB fixes the value of the RHST[2:0] flags when 0 is written to the USBRST bit by software and the USB completes SE0 driving.

When the function controller is selected, the RHST[2:0] flags indicate 010b (connection while DPRPU = 1) or 001b (disconnection while DMRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.

With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.

This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.

With this bit set to 0, the USB enters the idle state after outputting SOF packets.

The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).

This bit should be set to 0 if the function controller is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.

Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.

The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.

The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.

This bit should be set to 1 in the suspended state.

Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

This bit should be set to 0 if the function controller is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.

When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.

The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.

Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.

Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

This bit should be set to 0 if the function controller is selected.

RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.

With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 μ s) from the downstream port device and performs the resume processing (drives the port to the K-state).

With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.

While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).

This bit should be set to 0 if the function controller is selected.

WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.

The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.

According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).

This bit should be set to 0 if the host controller is selected.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.

After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

32.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

Address(es): USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): USB0.CFIFO.L 000A 0014h, USB0.D0FIFO.L 000A 0018h, USB0.D1FIFO.L 000A 001Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following precautions.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA/DTC transfer function is used, etc.).
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right for a FIFO buffer can be on the CPU side or on the Serial Interface Engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bit

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY flag in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 32.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the lower byte in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the lower byte in bytes.

Table 32.5 Endian Operation in 16-Bit Access

MDE.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

32.2.5 CFIFO Port Select Register (CFIFOSEL), D0FIFO Port Select Register (D0FIFOSEL), D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): USB0.CFIFOSEL 000A 0020h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	CFIFO Port Access Pipe Specification	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected. 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewind. 1: The buffer pointer is rewind.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] flags (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] flags are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.

After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.

Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.

Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.

When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.

When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

- D0FIFOSEL, D1FIFOSEL

Address(es): USB0.D0FIFOSEL 000A 0028h, USB0.D1FIFOSEL 000A 002Ch

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE [3:0]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewind. 1: The buffer pointer is rewind.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] flags (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN bit Value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] flags are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.

After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.

Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.

When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.

When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.

When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables the DMA/DTC transfer request to be issued.

Before setting the DREQE bit to 1 to enable the DMA/DTC transfer request to be issued, set the CURPIPE[3:0] bits.

When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data in the selected pipe has been read.

With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.

When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit.

When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

32.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USB0.CFIFOCTR 000A 0022h, USB0.D0FIFOCTR 000A 002Ah, USB0.D1FIFOCTR 000A 002Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length Flag	Indicate the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] flags shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W ¹
b15	BVAL	Buffer Memory Valid	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

Registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

DTLN[8:0] Flags (Receive Data Length Flag)

The DTLN[8:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] flags indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1) value as described below.

- RCNT = 0

The USB sets the DTLN[8:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data from a single FIFO buffer plane.

While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.

- RCNT = 1

The USB decrements the value indicated by the DTLN[8:0] flags each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)

The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

FRDY Flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.

In the following cases, the USB sets the FRDY flag to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only the FIFO buffer on the CPU side even when both planes are read-enabled.

When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) before setting the BCLR bit to 1.

When the selected pipe is in the transmitting direction, if 1 is written to the BVAL bit and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY flag in the FIFO port control register is 1 (set by the USB).

BVAL Bit (Buffer Memory Valid)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).

When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

Writing 1 to the BVAL bit should be done while the FRDY flag is 1 (set by the USB).

When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

32.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USB0.INTENB0 000A 0030h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected. Set these bits to 0 when the host controller is selected.

On detecting the interrupt corresponding to the bit in the INTENB0 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS0 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB0 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS0 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB0 register is modified from 0 to 1 by software.

32.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USB0.INTENB1 000A 0032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCH E	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	PDDETINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINTE0	Portable Device Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH E	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in the INTENB1 register can be set to 1 only when the host controller is selected. Set these bits to 0 when the function controller is selected.

The INTENB1 register specifies the interrupt masks when the host controller is selected, and for the setup transaction. On detecting the interrupt corresponding to the bit in the INTENB1 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS1 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB1 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS1 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB1 register is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

32.2.9 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USB0.BRDYENB 000A 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B RDYE	PIPE8B RDYE	PIPE7B RDYE	PIPE6B RDYE	PIPE5B RDYE	PIPE4B RDYE	PIPE3B RDYE	PIPE2B RDYE	PIPE1B RDYE	PIPE0B RDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY flag to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in the BRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY flag (n = 0 to 9) and the INTSTS0.BRDY flag. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY flag indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in the BRDYENB register is modified from 0 to 1 by software.

32.2.10 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USB0.NRDYENB 000A 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDYE	PIPE8NRDYE	PIPE7NRDYE	PIPE6NRDYE	PIPE5NRDYE	PIPE4NRDYE	PIPE3NRDYE	PIPE2NRDYE	PIPE1NRDYE	PIPE0NRDYE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY flag to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in the NRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPE_nNRDY flag (n = 0 to 9) and the INTSTS0.NRDY flag. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPE_nNRDY flag indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in the NRDYENB register is modified from 0 to 1 by software.

32.2.11 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USB0.BEMPENB 000A 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B EMPE	PIPE8B EMPE	PIPE7B EMPE	PIPE6B EMPE	PIPE5B EMPE	PIPE4B EMPE	PIPE3B EMPE	PIPE2B EMPE	PIPE1B EMPE	PIPE0B EMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP flag to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in the BEMPENB register to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP flag (n = 0 to 9) and the INTSTS0.BEMP flag. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP flag in the BEMPSTS register indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in the BEMPENB register is modified from 0 to 1 by software.

32.2.12 SOF Output Configuration Register (SOFCFG)

Address(es): USB0.SOFCFG 000A 003Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TRNENSEL	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor Flag*1	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

EDGESTS Flag (Edge Interrupt Output Status Monitor Flag)

The EDGESTS flag indicates 1 when the edge interrupt output signal is in the middle of the edge processing.

Confirm that this flag is 0 before stopping the clock supply to the USB.

BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.

Set the TRNENSEL bit to 1 when a low-speed device is connected.

The TRNENSEL bit is valid only when the host controller is selected.

This bit should be set to 0 if the function controller is selected.

32.2.13 Interrupt Status Register 0 (INTSTS0)

Address(es): USB0.INTSTS0 000A 0040h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]		VALID		CTSQ[2:0]		
Value after reset:	0	0	0	0/1*1	0	0	0	0*2	0*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
b3	VALID	USB Request Reception Flag	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State Flag	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: USB0_VBUS pin is low. 1: USB0_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status Flag	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status Flag	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status Flag	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*5	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status Flag*5	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: SOF interrupts are not generated. 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status Flag*5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status Flag*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.

Note 2. The value is 1 when the USB0_VBUS pin is high and 0 when the USB0_VBUS pin is low.

Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.

Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID flag, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to the status flags indicating 0.

Note 5. The status of the RESM, DVST, and CTRT flags are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.

Note 6. A change in the status indicated by the VBINT and RESM flags can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

CTSQ[2:0] Flags (Control Transfer Stage Flag)

When the host controller is selected, the read value is invalid.

VALID Flag (USB Request Reception Flag)

When the host controller is selected, the read value is invalid.

DVSQ[2:0] Flags (Device State Flag)

The DVSQ[2:0] flags are initialized by a USB bus reset.

When the host controller is selected, the read value is invalid.

BRDY Flag (Buffer Ready Interrupt Status Flag)

Indicates the BRDY interrupt status.

The USB sets the BRDY flag to 1 when at least one PIPEnBRDY flag ($n = 0$ to 9) is set to 1 among the PIPEnBRDY flags. These bits correspond to the BRDYENB.PIPEnBRDYE bits ($n = 0$ to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.

For the conditions for PIPEnBRDY status assertion, refer to section 32.3.3.1, BRDY Interrupt.

The USB sets the BRDY flag to 0 when 0 is written by software to all the PIPEnBRDY flags corresponding to the PIPEnBRDYE bits that have been set to 1.

The BRDY flag cannot be set to 0 even if 0 is written to this bit by software.

NRDY Flag (Buffer Not Ready Interrupt Status Flag)

The USB sets the NRDY flag to 1 when at least one PIPEnNRDY flag ($n = 0$ to 9) is set to 1 among the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits ($n = 0$ to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).

For the conditions for PIPEnNRDY status assertion, refer to section 32.3.3.2, NRDY Interrupt.

The USB sets the NRDY flag to 0 when 0 is written by software to all the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits that have been set to 1.

The NRDY flag cannot be set to 0 even if 0 is written to this bit by software.

BEMP Flag (Buffer Empty Interrupt Status Flag)

The USB sets the BEMP flag to 1 when at least one PIPEnBEMP flag ($n = 0$ to 9) is set to 1 among the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits ($n = 0$ to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).

For the conditions for PIPEnBEMP status assertion, refer to section 32.3.3.3, BEMP Interrupt.

The USB sets the BEMP flag to 0 when 0 is written by software to all the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits that have been set to 1.

The BEMP flag cannot be set to 0 even if 0 is written to this bit by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a change in the control transfer stage.

When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.

When the host controller is selected, the read value is invalid.

DVST Flag (Device State Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST flag to 1 on detecting a change in the device state.

When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.

When the host controller is selected, the read value is invalid.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) When the host controller is selected

The USB sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)

(2) When the function controller is selected

The USB sets the SOFR flag to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)

The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Flag (Resume Interrupt Status Flag)

When the function controller is selected, the USB sets the RESM flag to 1 on detecting the falling edge of the signal on the USB0_DP pin in the suspended state (DVSQ[2:0] = 1xxb).

When the host controller is selected, the read value is invalid.

VBINT Flag (VBUS Interrupt Status Flag)

The USB sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB0_VBUS pin input value. The USB sets the VBSTS flag to indicate the USB0_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS flag until the same value is read three or more times, and eliminate chattering.

32.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USB0.INTSTS1 000A 0042h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	PDDETINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDDETINT0	Portable Device Detection Interrupt Status Flag	0: PDDETINT interrupts are not generated. 1: PDDETINT interrupts are generated.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status Flag	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag *2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCCR	Overcurrent Input Change Interrupt Status Flag*2	0: OVRCCR interrupts are not generated. 1: OVRCCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. A change in the status indicated by the OVRCCR or BCHG flag can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.

No interrupts other than those indicated by the BCHG and OVRCCR flags can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

The INTSTS1 register is used to confirm the status of each interrupt when the host controller is selected.

The various status change interrupts indicated by the flags in the INTSTS1 register should be enabled only when the host controller is selected.

PDDETINT0 Flag (Portable Device Detection Interrupt Status Flag)

Indicates the status of the portable device detection interrupt when the host controller is selected.

This bit is set to 1 when the USB module detects when a level change (high to low or low to high) occurs in the input value to the VDPDET pin of the USB physical layer transceiver (PHY). The USB module sets the PDDETSTS0 flag to indicate the VDPDET input value.

When the PDDETINT interrupt is generated, use software to repeat reading the PDDETSTS0 flag until the same value is read three or more times, and eliminate chattering.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.

The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.

When the function controller is selected, the read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

Indicates the status of the setup transaction error interrupt when the host controller is selected.

The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.

Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

EOFERR Flag (EOF Error Detection Interrupt Status Flag)

Indicates the status of the EOFERR interrupt when the host controller is selected.

The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.

After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

ATTCH Flag (ATTCH Interrupt Status Flag)

Indicates the status of the ATTCH interrupt when the host controller is selected.

The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 μ s, and sets the ATTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 μ s.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 μ s.

When the function controller is selected, the read value is invalid.

DTCH Flag (USB Disconnection Detection Interrupt Status Flag)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.

The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

The USB detects bus disconnection based on USB Specification 2.0.

After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

BCHG Flag (USB Bus Change Interrupt Status Flag)

Indicates the status of the USB bus change interrupt.

The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

The USB sets the LNST[1:0] flags to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] flags until the same value is read three or more times, and eliminate chattering.

A change in the USB bus state can be detected even while the internal clock supply is stopped.

When the function controller is selected, the read value is invalid.

OVRCCR Flag (Overcurrent Input Change Interrupt Status Flag)

Indicates the status of the USB0_OVRCURA and USB0_OVRCURB input pin change interrupt.

The USB detects the OVRCCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USB0_OVRCURA and USB0_OVRCURB pins, and sets the OVRCCR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

32.2.15 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USB0.BRDYSTS 000A 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B RDY	PIPE8B RDY	PIPE7B RDY	PIPE6B RDY	PIPE5B RDY	PIPE4B RDY	PIPE3B RDY	PIPE2B RDY	PIPE1B RDY	PIPE0B RDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BRDY	BRDY Interrupt Status Flag for Pipe 0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status Flag for Pipe 1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status Flag for Pipe 2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status Flag for Pipe 3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status Flag for Pipe 4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status Flag for Pipe 5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status Flag for Pipe 6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status Flag for Pipe 7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status Flag for Pipe 8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status Flag for Pipe 9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the flags in the BRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

32.2.16 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USB0.NRDYSTS 000A 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0NRDY	NRDY Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the NRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

32.2.17 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USB0.BEMPSTS 000A 004Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B EMP	PIPE8B EMP	PIPE7B EMP	PIPE6B EMP	PIPE5B EMP	PIPE4B EMP	PIPE3B EMP	PIPE2B EMP	PIPE1B EMP	PIPE0B EMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the BEMPSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

32.2.18 Frame Number Register (FRMNUM)

Address(es): USB0.FRMNUM 000A 004Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRN	CRCE	—	—	—						FRNM[10:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error Flag	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overflow/Underflow Detection Status Flag	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

FRNM[10:0] Flags (Frame Number Flag)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] flags at the SOF packet reception.

CRCE Flag (Receive Data Error Flag)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.

The CRCE flag can be set to 0 by writing 0 to the CRCE flag by software. Here, 1 should be written to the other bits in FRMNUM.

On detecting a CRC error, the USB generates the internal NRDY interrupt request.

OVRN Flag (Overflow/Underflow Detection Status Flag)

Indicates whether an overflow/underflow error has been detected in the pipe during isochronous transfer.

The OVRN flag can be set to 0 by writing 0 to the OVRN flag by software. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

(2) When the function controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

32.2.19 USB Request Type Register (USBREQ)

Address(es): USB0.USBREQ 000A 0054h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBREQ register stores setup requests for control transfers.

When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.

When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.

The USBREQ register is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:
Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

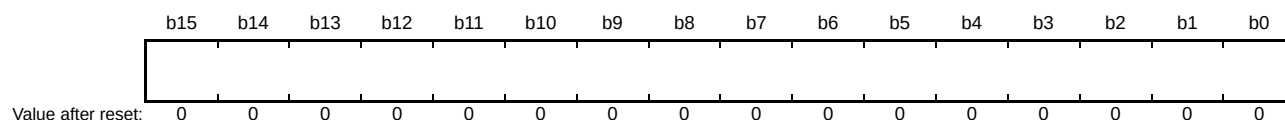
BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:
Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

32.2.20 USB Request Value Register (USBVAL)

Address(es): USB0.USBVAL 000A 0056h



When the function controller is selected, the value of wValue that has been received is stored in the USBVAL register.

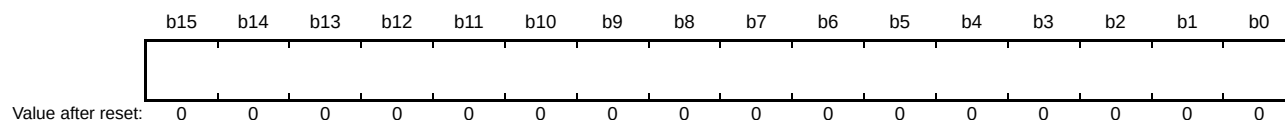
When the host controller is selected, the value of wValue to be transmitted is set.

The USBVAL register is initialized by a USB bus reset.

- When the host controller is selected:
Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not modify the value of the USBVAL register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the USBVAL register has no effect.

32.2.21 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 000A 0058h



The USBINDX register stores setup requests for control transfers.

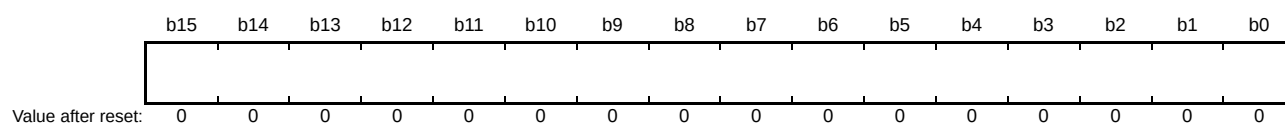
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.

The USBINDX register is initialized by a USB bus reset.

- When the host controller is selected:
Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not modify the value of the USBINDX register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the USBINDX register has no effect.

32.2.22 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 000A 005Ah



The USBLENG register stores setup requests for control transfers.

When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.

The USBLENG register is initialized by a USB bus reset.

- When the host controller is selected:
Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not modify the value of the USBLENG register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the USBLENG register has no effect.

32.2.23 DCP Configuration Register (DCPCFG)

Address(es): USB0.DCPCFG 000A 005Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	SHTNA K	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while DCPCTR.PID[1:0] bits are 00b (NAK). Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.

When the function controller is selected, the DIR bit should be set to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.

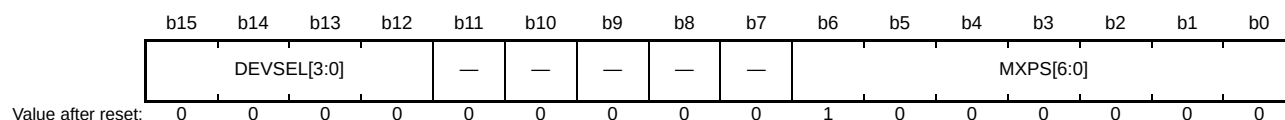
The SHTNAK bit is valid when the selected pipe is in the receiving direction.

When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

32.2.24 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USB0.DCPMAXP 000A 005Eh



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP. b6 b0 0 0 0 1 0 0 0: 8 bytes 0 0 1 0 0 0 0: 16 bytes 0 0 1 1 0 0 0: 24 bytes 0 1 0 0 0 0 0: 32 bytes 0 1 0 1 0 0 0: 40 bytes 0 1 1 0 0 0 0: 48 bytes 0 1 1 0 0 0 0: 56 bytes 1 0 0 0 0 0 0: 64 bytes 1 0 0 1 0 0 0: 72 bytes 1 0 1 0 0 0 0: 80 bytes 1 0 1 1 0 0 0: 88 bytes 1 1 0 0 0 0 0: 96 bytes 1 1 0 1 0 0 0: 104 bytes 1 1 1 0 0 0 0: 112 bytes 1 1 1 1 0 0 0: 120 bytes Settings other than above are prohibited.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID[1:0] bits are 00b (NAK) and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID[1:0] = 01b (BUF) while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

32.2.25 DCP Control Register (DCPCTR)

Address(es): USB0.DCPCTR 000A 0060h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	SUREQ	—	—	SUREQ CLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.

Note 2. Write 1 to the SQSET and SQCLR bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from 00b (NAK) to 01b (BUF) using the following procedure.

- When the transmitting direction is set
Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then write 01b (BUF response). After PID[1:0] have been set to 01b (BUF), the USB executes the OUT transaction.
- When the receiving direction is set
Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then set PID[1:0] bits to 01b (BUF). After PID[1:0] bits have been set to 01b (BUF), the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID[1:0] bits to 11b (STALL) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to 00b (NAK) on receiving the setup packet. Here, the USB sets the INTSTS0.VALID flag to 1. The setting of the PID[1:0] bits cannot be modified until the VALID flag is set to 0 by software.
- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 1xb (STALL) on detecting the control transfer sequence error.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.

The PID[1:0] bits are initialized by a USB bus reset.

CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.

When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to 01b (BUF), the USB completes the control transfer status stage.

During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.

The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.

1 cannot be written to the CCPL bit by software while the INTSTS0.VALID flag is 1.

The CCPL bit is initialized by a USB bus reset.

When the host controller is selected, be sure to write 0 to the CCPL bit.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from 01b (BUF) to 00b (NAK).

The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.

Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

For details, refer to section 32.3.4.1, Pipe Control Register Switching Procedures.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.

The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

When the function controller is selected, the USB sets the SQMON flag to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.

When the function controller is selected, the USB does not reference the SQMON flag during the IN/OUT transaction of the status stage, and does not allow the SQMON flag to toggle upon normal completion.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.

Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.

Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.

When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.

After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.

The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to 00b (NAK). After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).

Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.

When the function controller is selected, be sure to write 0 to the SUREQ bit.

BSTS Flag (Buffer Status Flag)

Indicates whether DCP FIFO buffer access is enabled or disabled.

The meaning of the BSTS flag depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS flag indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS flag indicates whether the data to be transmitted can be written to the buffer.

32.2.26 Pipe Window Select Register (PIPESEL)

Address(es): USB0.PIPESEL 000A 0064h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PIPESEL[3:0]	Pipe Window Select	b3b0 0 0 0 0: No pipe selected 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Pipes 1 to 9 should be set using registers PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using registers PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to registers PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from registers PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in registers PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

32.2.27 Pipe Configuration Register (PIPECFG)

Address(es): USB0.PIPECFG 000A 0068h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000b means an unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe assignment continued at the end of transfer 1: Pipe assignment disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 2. Modify the BFRE, DBLB, and DIR bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), checking the PIPEnCTR.PBUSY flag through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID[1:0] bits and CURPIPE[3:0] bits are in the state described in the above note 2.

The PIPECFG register specifies the transfer type, buffer memory access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.

Setting 0000b means an unused pipe.

Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 0000b can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify the PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.

The SHTNAK bit is valid when the selected pipe is pipe 1 to 5 in the receiving direction.

When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.

The DBLB bit is valid when the selected pipe is pipe 1 to 5.

BFRE Bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.

When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.

When the BRDY interrupt is generated with the above conditions, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.

For details, refer to section 32.3.3.1, BRDY Interrupt.

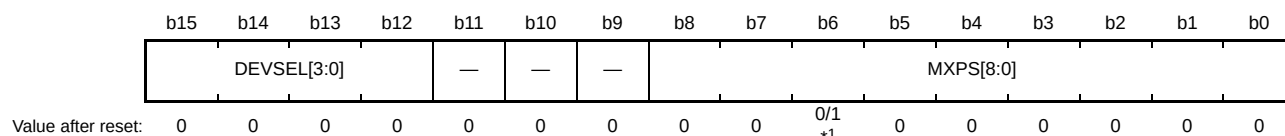
TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).

Before setting the PID[1:0] bits to 01b (BUF) for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

32.2.28 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USB0.PIPEMAXP 000A 006Ch



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	- Pipes 1 and 2: 1 byte (001h) to 256 bytes (100h) - Pipes 3 to 5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] are not provided.) - Pipes 6 to 9: 1 byte (001h) to 64 bytes (040h) (Bits [8:7] are not provided.)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

Note 2. Modify the MXPS[8:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEMAXP register specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] Bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of pipes 1 and 2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

32.2.29 Pipe Cycle Control Register (PIPEPERI)

Address(es): USB0.PIPEPERI 000A 006Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval *1	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEPERI register selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set the PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are invalid for pipes 3 to 5; set the IITV[2:0] bits to 000b for pipes 3 to 5.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

32.2.30 Pipe n Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): USB0.PIPE1CTR 000A 0070h, USB0.PIPE2CTR 000A 0072h, USB0.PIPE3CTR 000A 0074h, USB0.PIPE4CTR 000A 0076h, USB0.PIPE5CTR 000A 0078h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set* ²	0: Write disabled 1: Specifies DATA1.	R/W* ¹
b8	SQCLR	Sequence Toggle Bit Clear* ²	0: Write disabled 1: Specifies DATA0.	R/W* ¹
b9	ACLRM	Auto Buffer Clear Mode* ³	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode* ²	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status Flag	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read.

Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the ACLRM bit while PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEnCTR register can be set regardless of the pipe selection in the PIPESEL register.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.

The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 32.6 and Table 32.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.

After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 00b (NAK) on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.

The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.

Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

For details, refer to section 32.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 32.8 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.

When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.

When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.

(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB transmits a zero-length packet in response to the IN token.

The USB updates (allows toggling of) the sequence toggle bit (data PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).

In this case, the USB does not generate the BRDY or BEMP interrupt.

(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB returns NAK in response to the OUT token and generates the NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.

When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.

When the host controller is selected, be sure to set the ATREPM bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.

When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM flag to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.

The USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.

The INBUFM flag indicates the same value as the BSTS flag when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 32.9.

Table 32.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 32.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR bit = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

Table 32.8 Information Cleared by USB by Setting ACLRM = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	When the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	FIFO buffer toggle control	When the PIPECFG.DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

Table 32.9 Operation of BSTS Flag

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Flag Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

- PIPEnCTR (n = 6 to 9)

Address(es): USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh, USB0.PIPE9CTR 000A 0080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Sequence Toggle Bit Confirmation Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Invalid 1: Specifies DATA0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.

Note 2. Write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the ACLRM bit while the PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.

The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 32.6 and Table 32.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.

After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.

The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.

Reading the PBUSY flag after the PID[1:0] bits has been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

SQMON Flag (Sequence Toggle Bit Confirmation Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 32.10 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 32.9.

Table 32.10 Information Cleared by USB by Setting the ACLRM Bit = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

32.2.31 Pipe n Transaction Counter Enable Register (PIPEnTRE) (n = 1 to 5)

Address(es): USB0.PIPE1TRE 000A 0090h, USB0.PIPE2TRE 000A 0094h, USB0.PIPE3TRE 000A 0098h, USB0.PIPE4TRE 000A 009Ch, USB0.PIPE5TRE 000A 00A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Modify each bit in the PIPEnTRE register while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN register through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the PIPEnTRN register.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID[1:0] bits to 00b (NAK) for the corresponding pipe on having received the number of packets equal to the setting of the PIPEnTRN register.
- While the PIPECFG.BFRE bit is 1, the USB asserts the BRDY interrupt on having received the number of packets equal to the setting of the PIPEnTRN register and then reading the last received data.

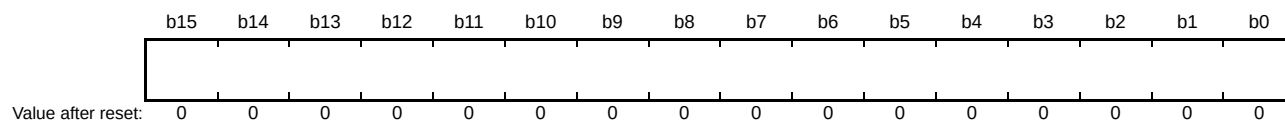
For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the PIPEnTRN register before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

32.2.32 Pipe n Transaction Counter Register (PIPEnTRN) (n = 1 to 5)

Address(es): USB0.PIPE1TRN 000A 0092h, USB0.PIPE2TRN 000A 0096h, USB0.PIPE3TRN 000A 009Ah, USB0.PIPE4TRN 000A 009Eh, USB0.PIPE5TRN 000A 00A2h



The PIPEnTRN register retains the setting by a USB bus reset.

- When written to:
This register specifies the total of packets (number of transactions) to be received in corresponding pipe.
- When read from:
This register indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0.
This register indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.

The USB increments the value of the PIPEnTRN register by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value $\neq$ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB sets the value of the PIPEnTRN register to 0000h when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the PIPEnTRN register to 0000h.

When the transaction counter is not used, set the PIPEnTRN register to 0000h.

Setting the number of transactions to be transferred to the PIPEnTRN register is only enabled when the PIPEnTRE.TRENB bit is 0. To modify the number of transactions to be transferred, set the TRCLR bit to 1 (to clear the current counter value) before setting the PIPEnTRE.TRENB bit to 1.

32.2.33 Device Address n Configuration Register (DEVADDn) (n = 0 to 5)

Address(es): USB0.DEVADD0 000A 00D0h, USB0.DEVADD1 000A 00D2h, USB0.DEVADD2 000A 00D4h, USB0.DEVADD3 000A 00D6h, USB0.DEVADD4 000A 00D8h, USB0.DEVADD5 000A 00DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	USBSPD[1:0]	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device which is the communication target for pipes 0 to 9.

When the host controller is selected, the bits in the DEVADDn register should be set before starting communication using each pipe.

The bits in the DEVADDn register should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- The DEVADDn register is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 01b (BUF) for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.

Set these bits to 01b when a low-speed device is connected, whereas set them to 10b when a full-speed device is connected.

When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.

When the function controller is selected, set these bits to 00b.

32.2.34 USB Module Control Register (USBMC)

Address(es): USB0.USBMC 000A 00CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDDUS BE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	VDDUSBE	USB Power Supply Circuit On/Off Control	0: USB power supply circuit off 1: USB power supply circuit on	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

VDDUSBE Bit (USB Power Supply Circuit On/Off Control)

The USB power supply circuit generates the reference voltage for battery charging. Set this bit to 1 when using the battery charging function.

32.2.35 BC Control Register 0 (USBBCCTRL0)

Address(es): USB0.USBBCCTRL0 000A 00B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PDDETSTS0	CHGDETSTS0	BATCHGE0	—	VDMSRCE0	IDPSINKE0	VDPSRCE0	IDMSINKE0	IDPSRCE0	RPDME0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPDME0	D– Pin Pull-Down Control	0: Pull-down off 1: Pull-down on	R/W
b1	IDPSRCE0	D+ Pin IDPSRC Output Control	0: Stop 1: 10 μ A output	R/W
b2	IDMSINKE0	D– Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b3	VDPSRCE0	D+ Pin VDPSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output	R/W
b4	IDPSINKE0	D+ Pin 0.6 V Input Detection (Comparator and Sink) Control	0: Detection off 1: Detection on (comparator and sink current on)	R/W
b5	VDMSRCE0	D– Pin VDMSRC (0.6 V) Output Control	0: Stop 1: 0.6 V output	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	BATCHGE0	BC (Battery Charger) Function Ch0 General Enable Control	0: Disabled 1: Enabled	R/W
b8	CHGDETSTS0	D– Pin 0.6 V Input Detection Status Flag*1	0: Not detected 1: Detected	R
b9	PDDETSTS0	D+ Pin 0.6 V Input Detection Status Flag*2	0: Not detected 1: Detected	R
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Valid when IDMSINKE0 = 1.

Note 2. Valid when IDPSINKE0 = 1.

RPDME0 Bit (D– Pin Pull-Down Control)

When using the battery charging function, set this bit to 1 to control the pull-down resistor of the D– pin.

IDPSRCE0 Bit (D+ Pin IDPSRC Output Control)

With this bit set to 1, when the function controller is selected, current output is enabled upon detection of the connection of the data pin and the D+ pin is pulled up.

IDMSINKE0 Bit (D– Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1, when the function controller is selected, the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– upon primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

VDPSRCE0 Bit (D+ Pin VDPSRC (0.6 V) Output Control)

With this bit set to 1, when the function controller is selected, output is enabled upon primary detection and VDPSRC (0.6 V) is applied to D+.

IDPSINKE0 Bit (D+ Pin 0.6 V Input Detection (Comparator and Sink) Control)

With this bit set to 1, when the function controller selected, the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– is connected to the function's D+ (DCP) via the host. When the host controller is selected, the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ upon primary detection is connected.

VDMSRCE0 Bit (D– Pin VDMSRC (0.6 V) Output Control)

With this bit set to 1, when the function controller selected, output is enabled upon secondary detection and VDMSRC (0.6 V) is applied to D–. When the host controller is selected, output is enabled upon primary detection and VDMSRC (0.6 V) is applied to D–.

CHGDETSTS0 Flag (D– Pin 0.6 V Input Detection Status Flag)

When the host controller is selected, this flag is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the host to D– during primary detection is connected, or whether VDPSRC (0.6 V) that is output from the function to D+ is connected to the function's D– via the host.

PDDETSTS0 Flag (D+ Pin 0.6 V Input Detection Status Flag)

When the function controller is selected, this flag is set to 1 if the USB module detects whether VDMSRC (0.6 V) that is output from the function to D– during secondary detection is connected to the function's D+ (DCP) via the host. When the host controller is selected, this bit is set to 1 if the USB module detects whether VDPSRC (0.6 V) that is output from the function to D+ during primary detection is connected.

32.3 Operation

32.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

32.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBFE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

32.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pull-up resistors of the D+ and D– lines and pull-down resistors of the D+ and D– lines are disabled (the SYSCFG.DPRPU bit = 0, DMRPU bit = 0, and DRPD bit = 0).

32.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU, DMRPU, and DRPD bits.

When the function controller is selected, confirm that connection to the USB host is made, then set the DPRPU or DMRPU bit to 1 and pull up the D+ line (during full-speed) and D– line (during low-speed).

When the DPRPU or DMRPU bit is set to 0 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.

When the host controller is selected, set the DRPD bit and pull down the D+ and D– lines.

Table 32.11 Controlling USB Data Bus Resistors

Settings			Controlling USB Data Bus Resistors		
DRPD	DPRPU	DMRPU	D–	D+	Remarks
0	0	0	Open	Open	When not used
0	1	0	Open	Pull-up	When operating as the function controller (full-speed)
0	0	1	Pull-up	Open	When operating as the function controller (low-speed)
1	0	0	Pull-down	Pull-down	When operating as the host controller
Other than above			—	—	Setting prohibited

32.3.1.4 Example of USB External Connection Circuit

Figure 32.2 shows an example of OTG connection of the USB connector in the self-powered state.

The USB controls the signals for enabling pull-up resistors for the D+ and D– signals and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU, DMRPU, and DRPD bits.

When the function controller is selected and the DPRPU or DMRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

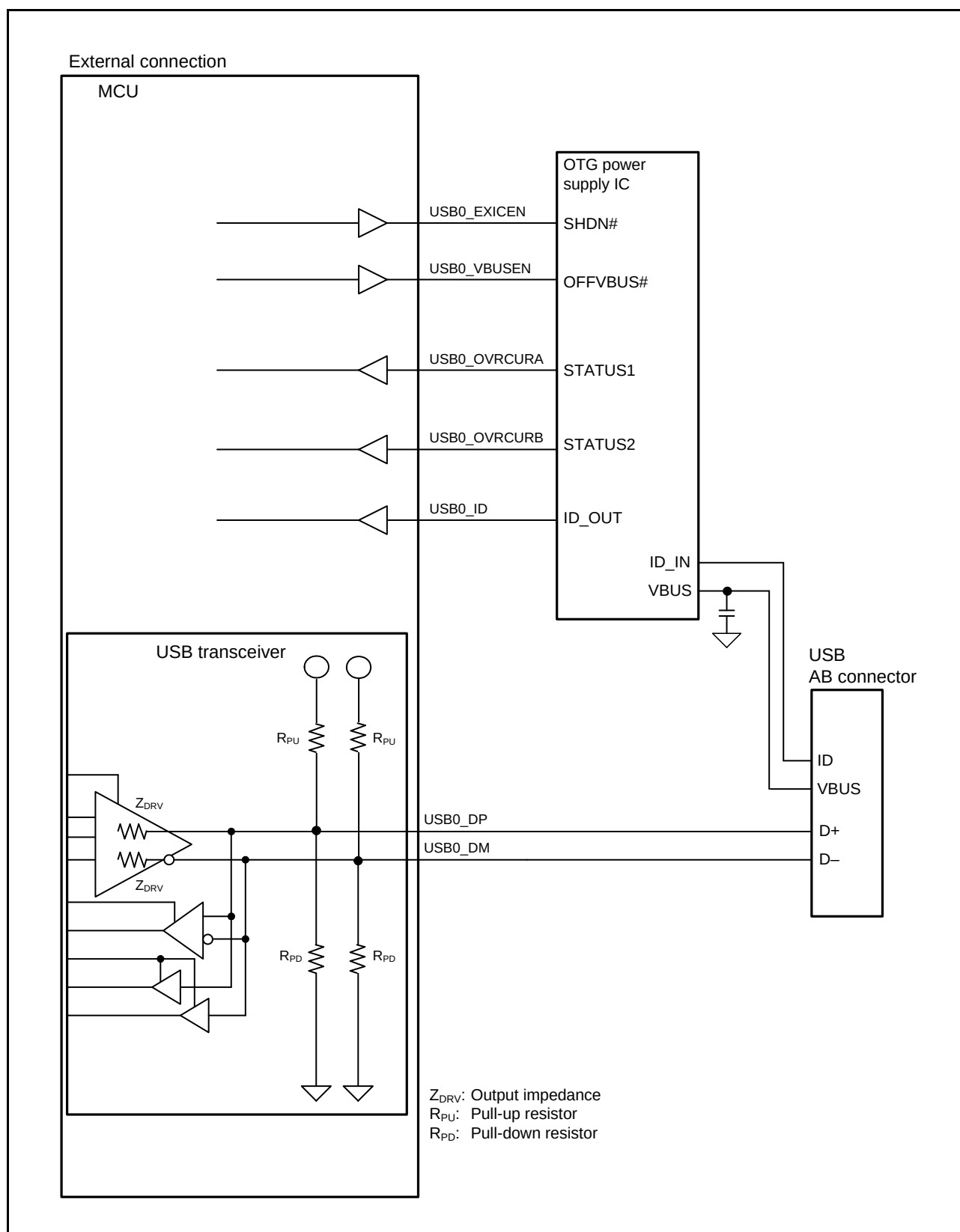


Figure 32.2 Sample OTG Connection of USB Connector in Self-Powered State

Figure 32.3 shows an example of functional connection of the USB connector in the self-powered state.

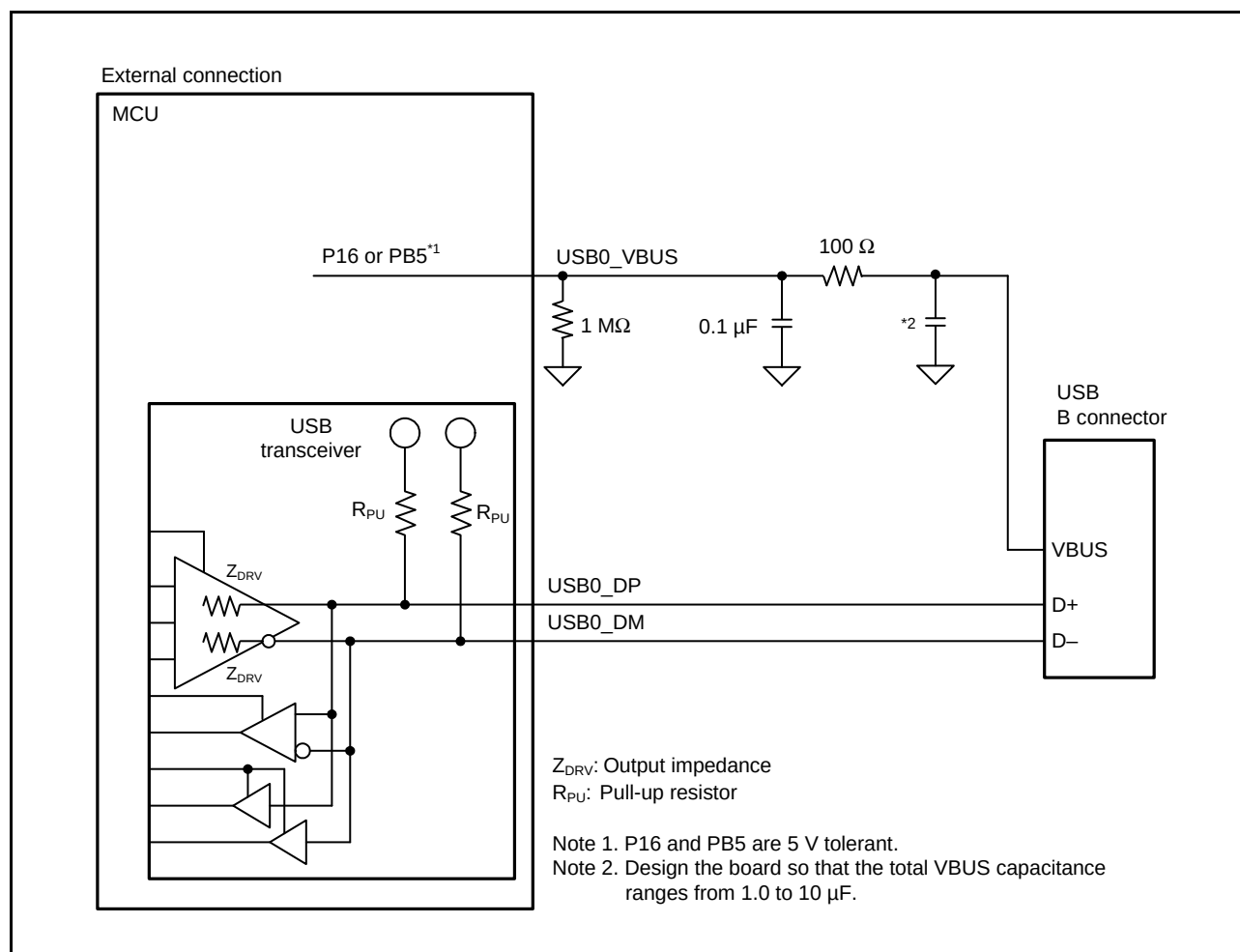


Figure 32.3 Functional Connection of USB Connector in Self-Powered State

[illegible]

Figure 32.4 Functional Connection Sample of USB Connector with Battery Charging Specification Rev.1.2 Supported

Figure 32.5 shows an example of host connection of the USB connector.

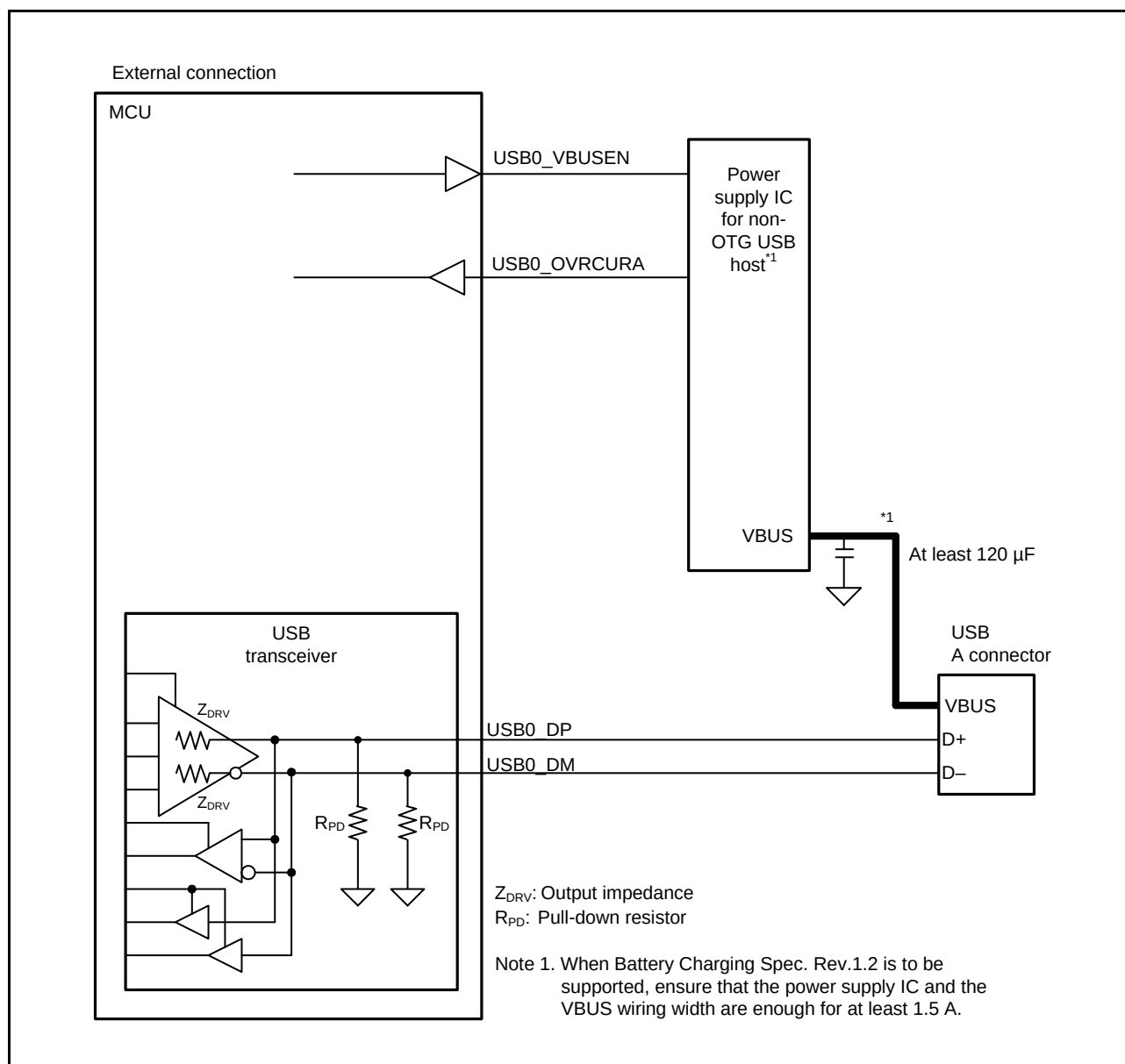


Figure 32.5 Sample Host Connection of USB Connector

Figure 32.6 shows an example of functional connection of the USB connector in bus powered state.

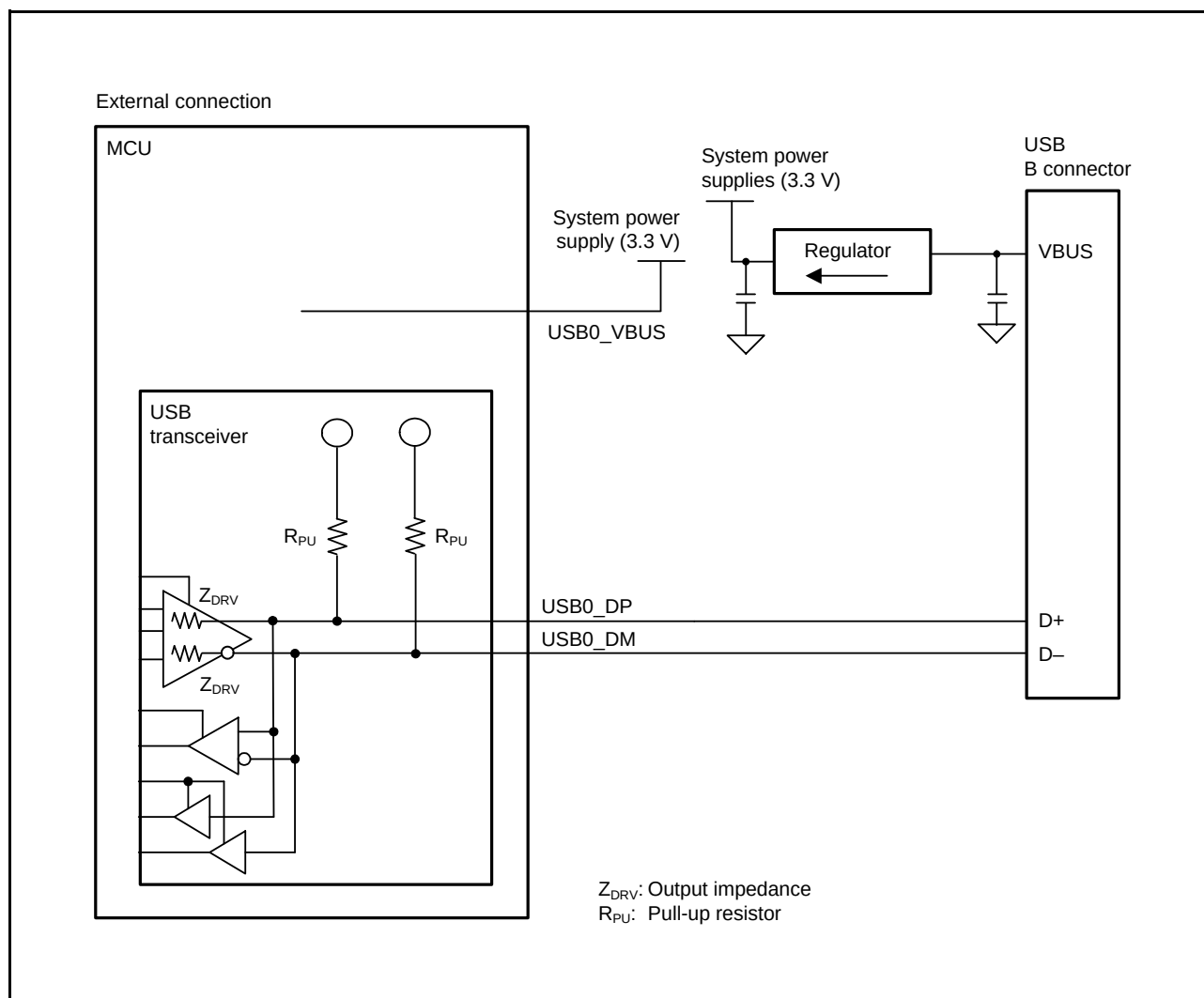


Figure 32.6 Functional Connection Sample of USB Connector in Bus Powered State

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

32.3.2 Interrupt Sources

Table 32.12 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

Table 32.12 Interrupt Sources

Bit to be Set	Name	Interrupt Source	Function That Generates the Interrupt	Status Flag
VBINT	VBUS interrupt	When a change in the state of the USB0_VBUS input pin has been detected (low to high or high to low)	Host/function ¹	INTSTS0.VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	Function	—
SOFR	Frame number update interrupt	[Host controller is selected] - When an SOF packet with a different frame number has been transmitted [Function controller is selected] - When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	- When a device state transition has been detected (any of the following conditions) - A USB bus reset detected - Suspended state detected - SET_ADDRESS request received - SET_CONFIGURATION request received	Function	INTSTS0.DVSTQ[2:0]
CTRT	Control transfer stage transition interrupt	- When a stage transition has been detected in control transfer (any of the following conditions) - Setup stage completed - Control write transfer status stage transition - Control read transfer status stage transition - Control transfer completed - A control transfer sequence error occurred	Function	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	- When transmission of all data in the buffer memory has been completed and the buffer has become empty - When a packet larger than the maximum packet size has been received	Host/function	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	[Host controller is selected] - When STALL has been received from the peripheral device for the issued token - When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) - When an overrun/underrun occurred during isochronous transfer [Function controller is selected] - When NAK has been returned for an IN or OUT token while the PID[1:0] bits are 01b (BUF) - When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer - When an overrun/underrun occurred during data reception in isochronous transfer	Host/function	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	When the buffer has become ready (reading or writing is enabled)	Host/function	BRDYSTS.PIPEnBRDY
OVRCCR	Overcurrent input change interrupt	When a change in the state of the USB0_OVRCURA or USB0_OVRCURB input pin has been detected (low to high or high to low)	Host	INTSTS1.OVRCCR
BCHG	Bus change interrupt	When a change of USB bus state has been detected	Host/function	SYSSTS0.LNST[1:0]
DTCH	Disconnection detection during full-speed operation	When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connection detection	When J-state or K-state is detected on the USB port for 2.5 μs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—
PDDTINT0	Portable device detection interrupt	When connection of the portable device has been detected	Host	INTSTS1.PDDTINT0

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 32.7 shows the circuits related to the interrupts in the USB.

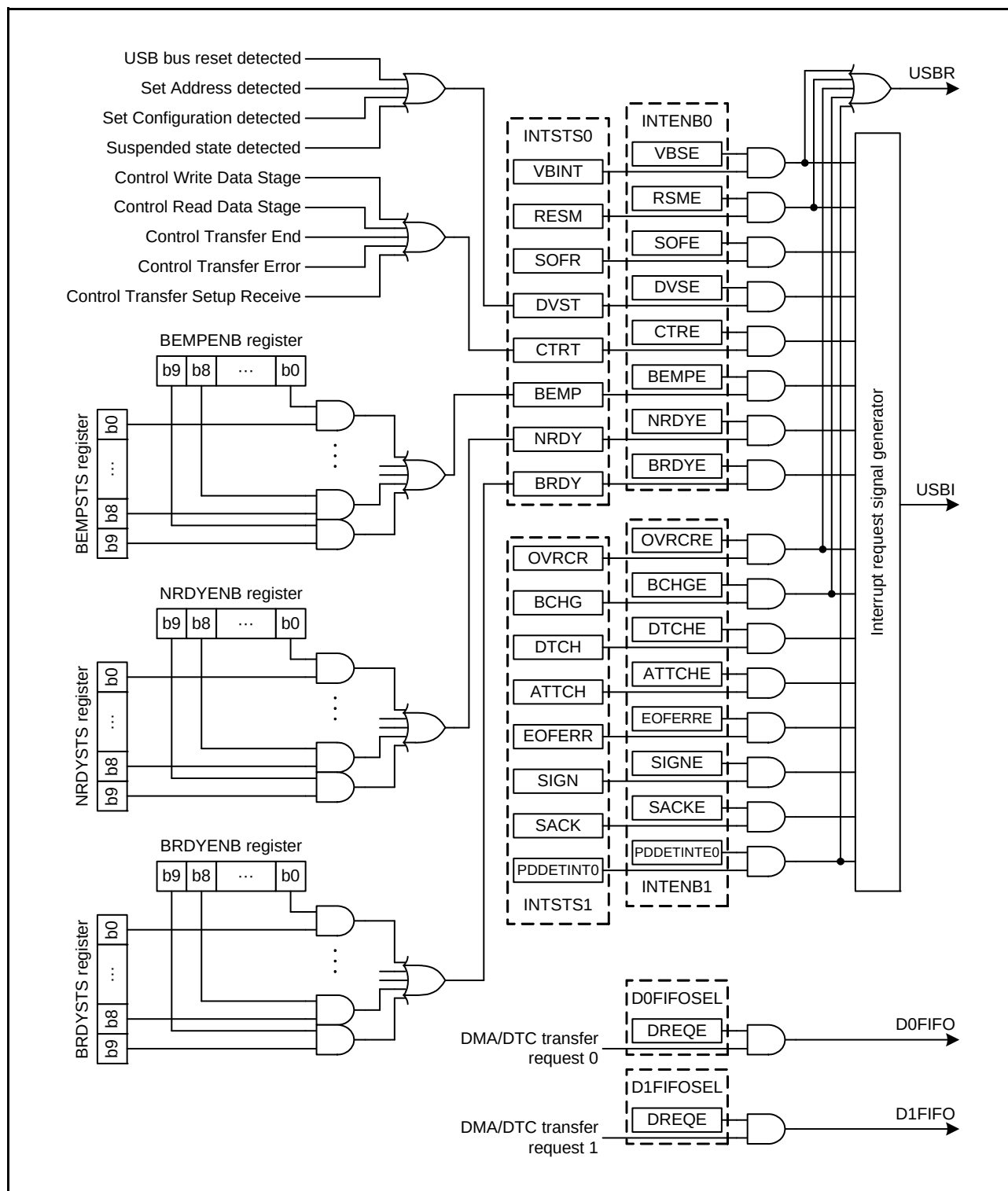


Figure 32.7 Circuits Related to Interrupts in USB

Table 32.13 shows the interrupts generated in the USB0.

Table 32.13 USB Interrupts

Interrupt Name	Interrupt Status Flag	DTC Activation	DMAC Activation
D0FIFO	DMA/DTC transfer request 0	Possible	Possible
D1FIFO	DMA/DTC transfer request 1	Possible	Possible
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, setup error, and portable device detection interrupt	Not possible	Not possible
USBR	VBUS interrupt, resume interrupt, overcurrent input change interrupt, bus change interrupt, and portable device detection interrupt	Not possible	Not possible

32.3.3 Interrupt Descriptions

32.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.

The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY flag corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

(b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
No request trigger is generated for the transaction in which data PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers. The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes. Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in the BRDYSTS register corresponding to the pertinent pipe. On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the pipe n transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN register are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[8:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.

With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.

The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.

In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLR bit.

(3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPE_nBRDY values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status flags (PIPE_nBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.

However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.

With this setting, the PIPE_nBRDY flag cannot be set to 0 through software.

When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Figure 32.8 shows the timing of BRDY interrupt generation.

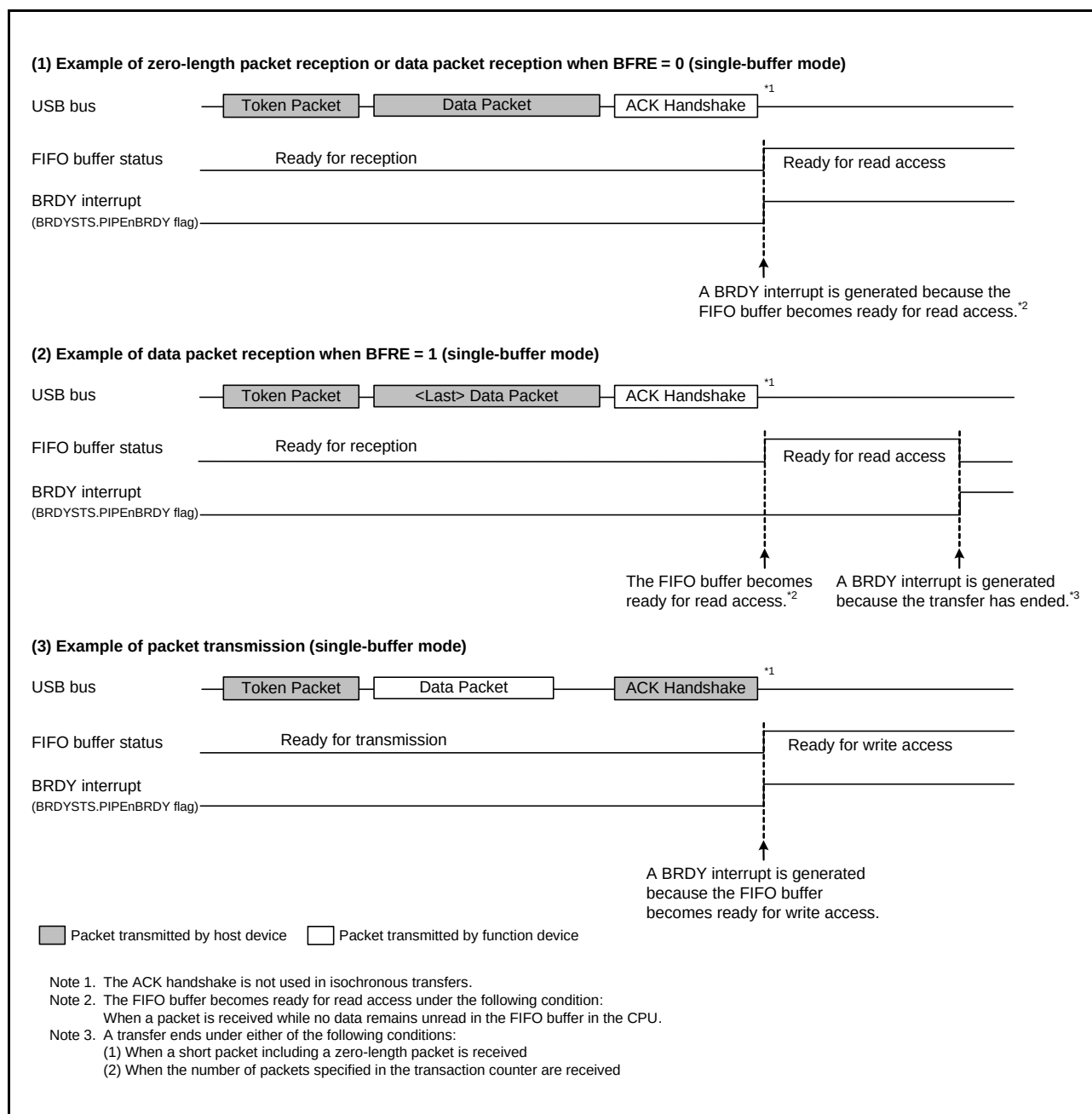


Figure 32.8 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting. Table 32.14 shows the condition for clearing the BRDY flag.

Table 32.14 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing BRDY Flag
0	The USB sets the BRDY flag to 0 when all bits in the BRDYSTS register have been set to 0 by software.
1	The USB sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

32.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding NRDYSTS.PIPE_nNRDY flag to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB sets the INTSTS0.NRDY flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.

Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

(1) When Host Controller is Selected

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPE_nNRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
In this case, the USB sets the bit corresponding to the PIPE_nNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
In this case, the USB sets the bit corresponding to the PIPE_nNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
In this case, the USB discards the received data for the IN token and sets the PIPE_nNRDY flag corresponding to the pipe and the OVRN flag to 1.
When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE flag to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
In this case, the USB sets the PIPE_nNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
In this case, the USB sets the PIPE_nNRDY flag corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.

In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe and the CRCE flag to 1.

- When the STALL handshake is received.

In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(2) When Function Controller is Selected

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.

In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY flag to 1.

For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.

For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY flag to 1 and OVRN flag to 1.

For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY flag to 1.

However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.

- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.

In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY flag to 1.

Figure 32.9 shows the timing of NRDY interrupt generation when the function controller is selected.

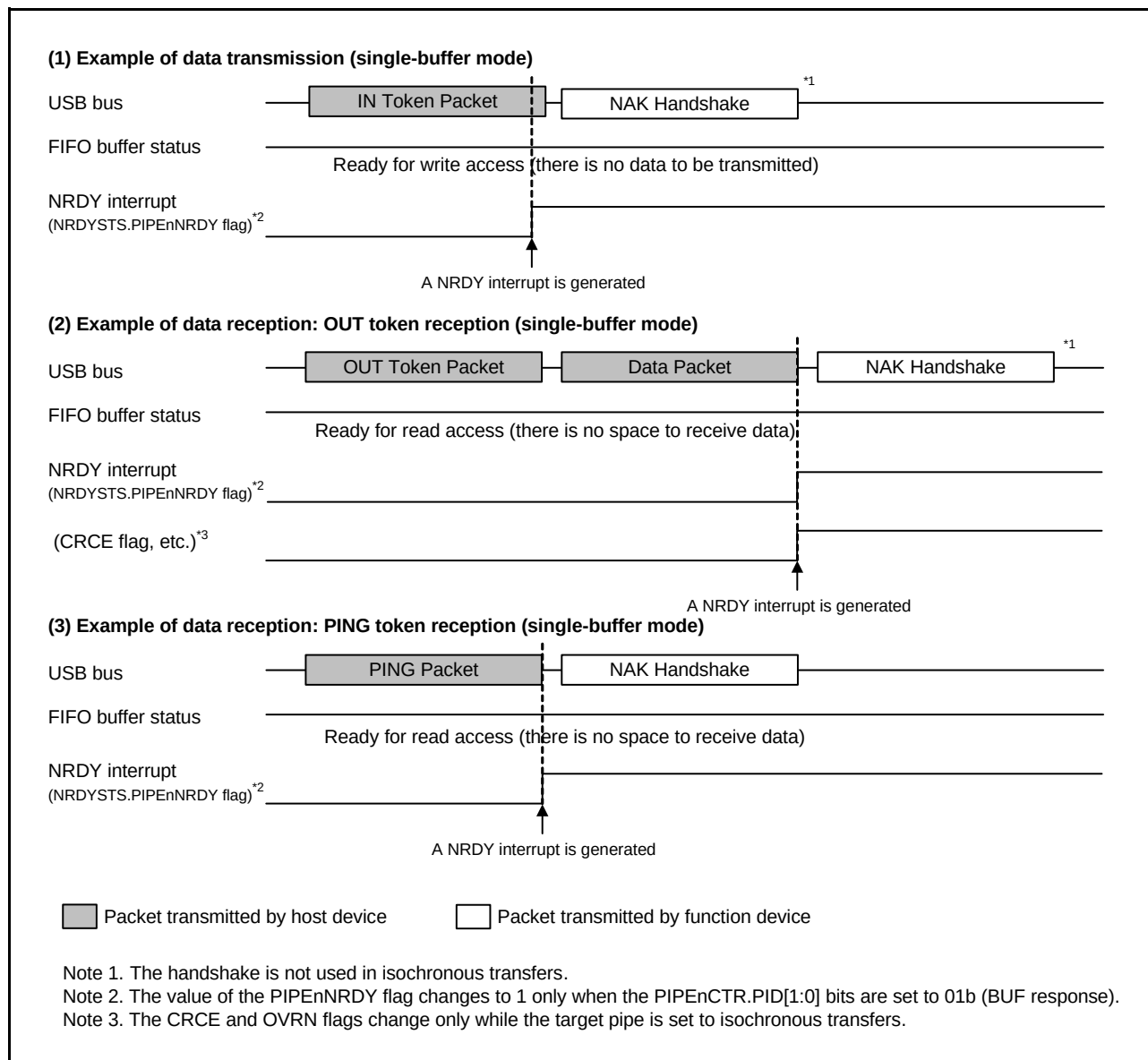


Figure 32.9 Timing of NRDY Interrupt Generation (When Function Controller is Selected)

32.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP flag to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).

In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.

In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.

However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
Writing 0 to the BEMPSTS.PIPEnBEMP flag clears the status.
Writing 1 to the BEMPSTS.PIPEnBEMP flag has no effect.

Figure 32.10 shows the timing of BEMP interrupt generation when the function controller is selected.

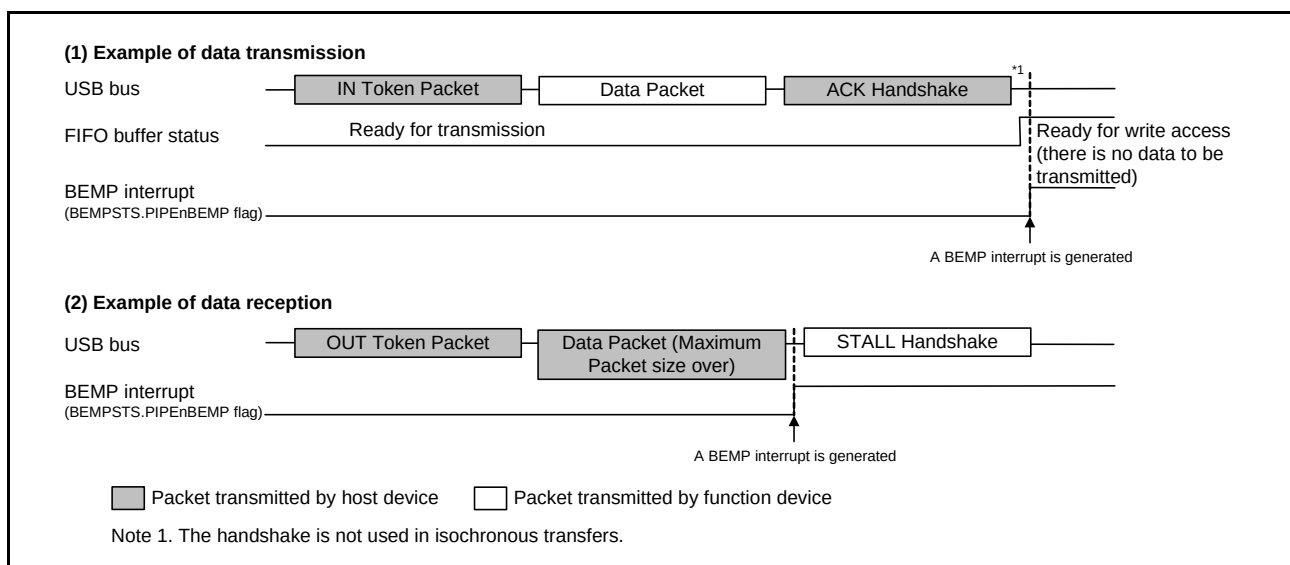


Figure 32.10 Timing of BEMP Interrupt Generation (When Function Controller is Selected)

32.3.3.4 Device State Transition Interrupt

Figure 32.11 is a diagram of device state transitions in the USB. The USB controls device state 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 interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.

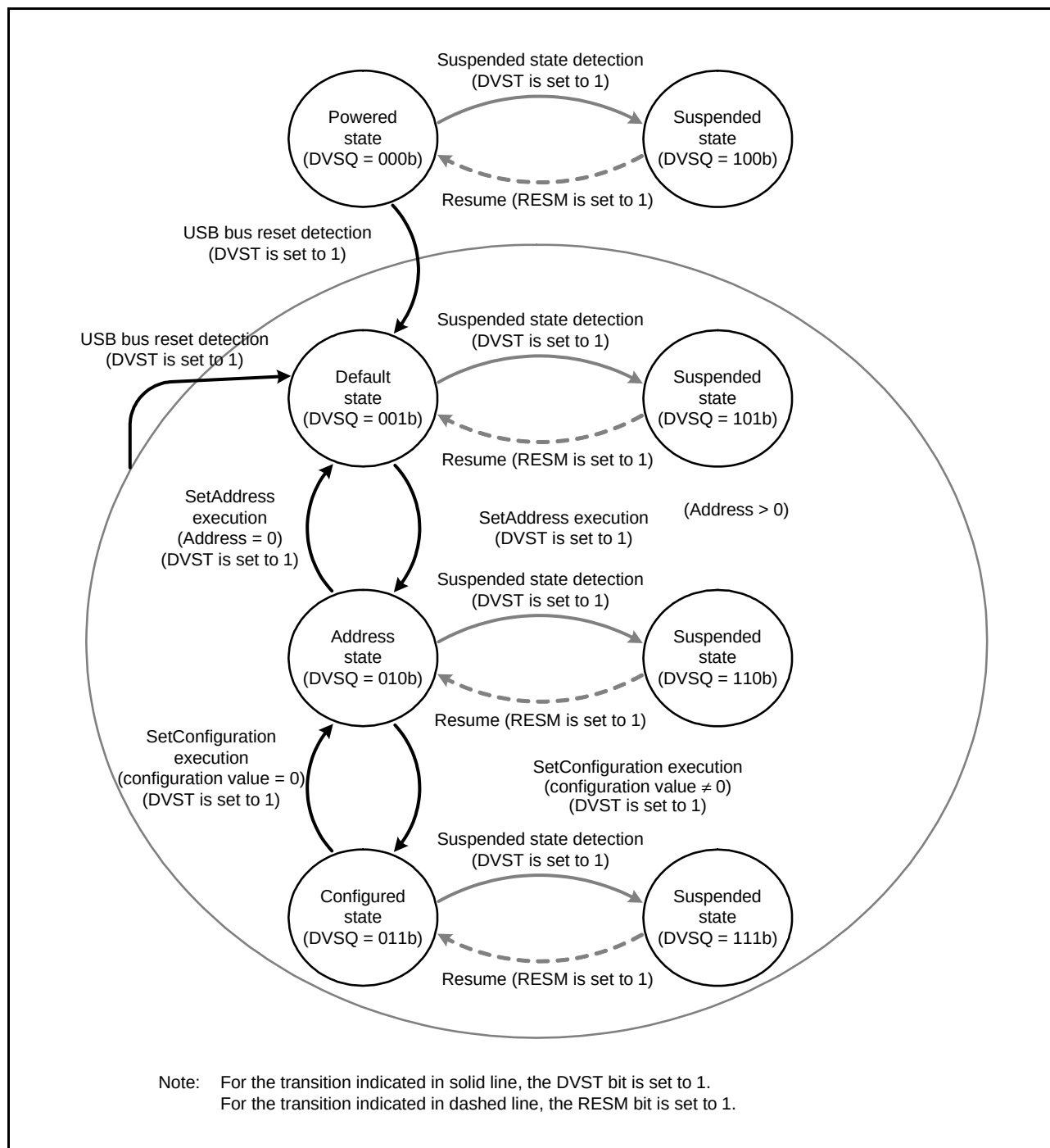


Figure 32.11 Device State Transitions

32.3.3.5 Control Transfer Stage Transition Interrupt

Figure 32.12 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the INTSTS0.CTSQ[2:0] flags.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with PID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with PID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT flag = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

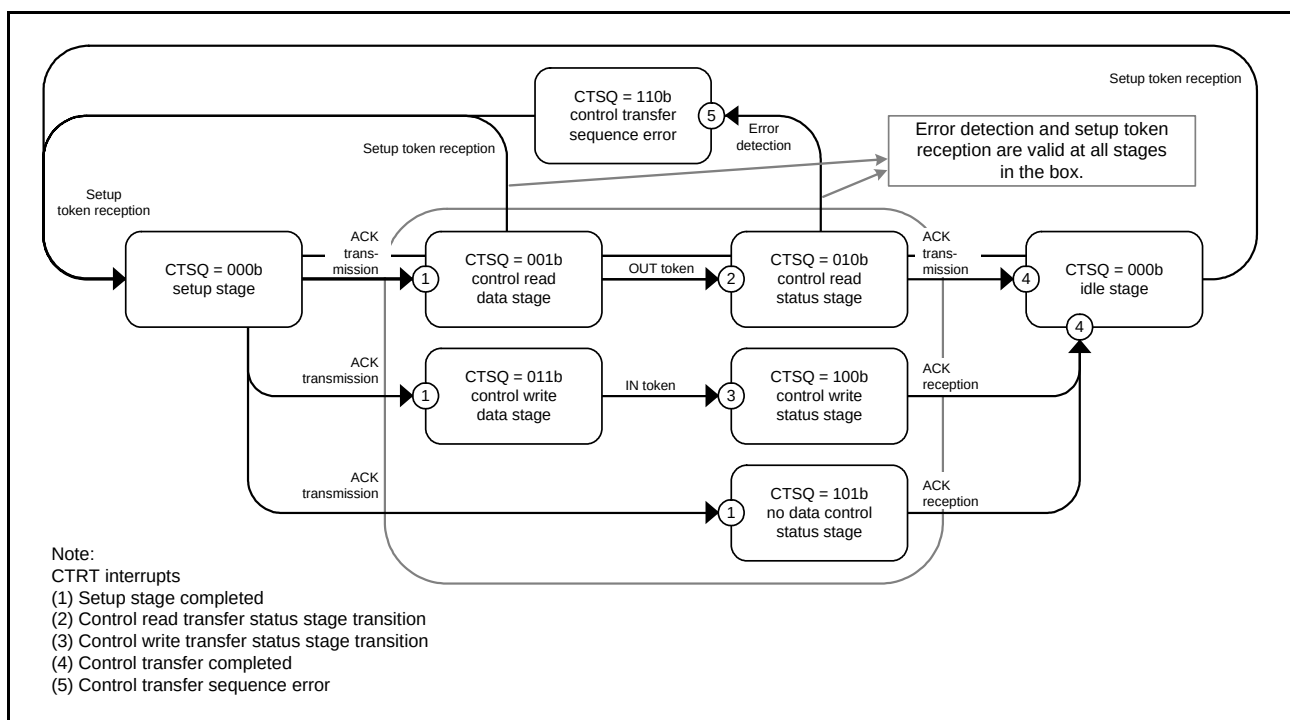


Figure 32.12 Control Transfer Stage Transitions

32.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.

When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

32.3.3.7 VBUS Interrupt

When the USB0_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB0_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USB0_VBUS pin level.

32.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

32.3.3.9 OVRCCR Interrupt

An OVRCCR interrupt is generated when the USB0_OVRCURA or USB0_OVRCURB pin level has changed. The levels of the USB0_OVRCURA and USB0_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCCR interrupt.

For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCCR interrupt.

32.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

32.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.

After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

32.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

32.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

32.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 μ s with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 μ s.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 μ s.

32.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.

After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

32.3.3.16 Portable Device Detection Interrupt

A portable device detection interrupt is generated when the USB module detects a level change (high to low or low to high) in the PDDET output from the USB-PHY. When a portable device detection interrupt is generated, use software to repeat reading the PDDETSTS0 flag until the same value is read three or more times, and perform debouncing.

32.3.4 Pipe Control

Table 32.15 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.

Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 32.15 Pipe Settings

Register Name	Bit Name	Setting	Remarks	
DCPCFG PIPECFG	TYPE[1:0]	Specifies the transfer type	Pipes 1 to 9:	Can be set
	BFRE	Selects the BRDY interrupt mode	Pipes 1 to 5:	Can be set
	DBLB	Selects double buffer mode	Pipes 1 to 5:	Can be set
	DIR	Selects transfer direction	IN or OUT can be set	
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9:	Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2:	Can be set (only when bulk transfer has been selected)
			Pipes 3 to 5:	Can be set
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selects a device	Referenced only when the host controller is selected.	
	MXPS[8:0]	Maximum packet size	Compliant with USB Specification 2.0.	
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2:	Can be set (only when isochronous transfer has been selected)
			Pipes 3 to 9:	Cannot be set
	IITV[2:0]	Interval counter	Pipes 1 and 2:	Can be set (only when isochronous transfer has been selected)
			Pipes 3 to 5:	Cannot be set
			Pipes 6 to 9:	Can be set (only when the host controller has been selected)
DCPCTR PIPECTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.	
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.	
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.	
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.	
	ATREPM	Auto response mode	Pipes 1 to 5:	Can be set Can be set only when the function controller has been selected.
	ACLRM	Auto buffer clear	Pipes 1 to 9:	Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.	
	SQSET	Sequence set	Sets the data toggle bit.	
	SQMON	Sequence monitor	Monitors the data toggle bit.	
	PBUSY	Pipe busy status		
	PID[1:0]	Response PID	Refer to section 32.3.4.6, Response PID.	
PIPEntRE	TRENB	Transaction counter enable	Pipes 1 to 5:	Can be set
	TRCLR	Current transaction counter clear	Pipes 1 to 5:	Can be set
PIPEntRN	—	Transaction counter	Pipes 1 to 5:	Can be set

32.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] = 00b (NAK)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] = 01b (BUF)).

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR register
- Bits in registers PIPECFG, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID[1:0] = 01b (BUF)) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to 00b (NAK).
3. Wait until the corresponding PBUSY flag is set to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in the DCPCFG and DCPMAXP register
- Bits in registers PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

32.3.4.2 Transfer Types

The PIPECFG.TYPE[1:0] bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

32.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

32.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined by USB Specification 2.0. For pipes 6 to 9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID[1:0] = 01b (BUF)).

- DCP: Set 8, 16, 32, or 64.
- Pipes 1 to 5: Set 8, 16, 32, or 64 when using bulk transfer.
- Pipes 1 and 2: Set a value between 1 and 256 when using isochronous transfer.
- Pipes 6 to 9: Set a value between 1 and 64.

32.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID[1:0] = 01b (BUF), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

32.3.4.6 Response PID

The PID[1:0] bits in the DCPCTR and PIPEnCTR registers are used to set the response PID for each pipe. The following shows the USB operation with various response PID settings:

(1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

(2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and issuing of tokens is automatically stopped:
When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.
(For details, refer to section 32.3.3.2, NRDY Interrupt.)
- If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and issuing of tokens is automatically stopped:
When STALL is received in response to the transmitted token.
When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and NAK is returned in response to transactions:
When the SETUP token is received normally (DCP only).
If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and STALL is returned in response to transactions:
When a maximum packet size exceeded error is detected in the received data packet.
When a control transfer sequence error has been detected (DCP only).

32.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON flag in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The DCPCTR.SQCLR bit and the PIPEnCTR.SQSET bit can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

32.3.4.8 Response PID = NAK Function

The USB 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 USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

32.3.4.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (the PIPECFG.DIR bit = 0), OUT-NAK mode is entered, and during an IN transfer (the DIR bit = 1), null auto response mode is entered.

32.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

32.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 μ s) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

32.3.5 FIFO Buffer Memory

32.3.5.1 FIFO Buffer Memory

The USB has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB (SIE side).

(1) Buffer Status

Table 32.16 and Table 32.17 show the buffer status in the USB. The buffer memory status can be confirmed using the DCPCTR.BSTS flag and the PIPEnCTR.INBUFM flag. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM flag is valid for pipe 0 to pipe 5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS flag to monitor the buffer memory status on the CPU side and the INBUFM flag to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DMAC/DTC is slow, software can use the INBUFM flag to confirm the end of transmission.

Table 32.16 Buffer Status Indicated by the BSTS Flag

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 32.17 Buffer Status Indicated by the INBUFM Flag

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

32.3.5.2 FIFO Buffer Clearing

Table 32.18 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.

Either a single or double buffer configuration can be selected for pipes 1 to 5, using the PIPECFG.DBLB bit.

Table 32.18 List of Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.

If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

32.3.5.3 FIFO Port Functions

Table 32.19 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.

In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] flags = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] flags in the port control register.

Table 32.19 FIFO Port Function Settings

Register Name	Bit Name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA/DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN	Checks the length of receive data.

(1) FIFO Port Selection

Table 32.20 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY flag in a port control register = 1 is checked.

In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

Table 32.20 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, 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. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the FRDY flag in the port control register = 1 should be checked after selecting a pipe.

32.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the DnFIFOSEL.MBW bit and the pipe targeted for the DMA transfer should be selected using the DnFIFOSEL.CURPIPE[3:0] bits. The selected pipe should not be changed during the DMA transfer.

(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USB automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.

Table 32.21 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 32.21, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 32.21 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary
Zero-length packet reception	Clearing is necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Normal short packet reception	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Transaction count end	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary

32.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).

The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

32.3.6.1 Control Transfers When the Host Controller is Selected

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.

After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPPD[1:0] bits set appropriately.

After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in the DEVADD2 register; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in the DEVADD5 register.

When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK flag in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.

A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON flag.

(2) Data Stage

Data is transferred using the DCP buffer memory.

The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID[1:0] bits = 01b (BUF). Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.

For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.

For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

32.3.6.2 Control Transfers When the Function Controller is Selected

(1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.

When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID flag to 1.
- Set the DCPCTR.PID[1:0] bits to 00b (NAK).
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in registers USBREQ, USBVAL, USBINDEX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID flag = 0. In the VALID flag = 1 state, PID[1:0] = 01b (BUF) cannot be set, and the data stage cannot be terminated.

Using the function of the VALID flag, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.

In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, refer to Figure 32.12.

(2) Data Stage

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

If the transfer data 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.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF).

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

- For control read transfers
A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
A zero-length packet is transmitted and an ACK response is received from the USB host.

(4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

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

32.3.7 Bulk Transfers (Pipes 1 to 5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 32.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1)
- Transaction count function
(PIPEnTRE.TRENB and TRCLR bits and PIPEnTRN register: refer to section 32.3.4.5, Transaction Counter (For Pipes 1 to 5 in Reading Direction))
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 32.3.4.8, Response PID = NAK Function)
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 32.3.4.9, Auto Response Mode)

32.3.8 Interrupt Transfers (Pipes 6 to 9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.

When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

32.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL).
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

32.3.9 Isochronous Transfers (Pipes 1 and 2)

The USB has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDLX function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

32.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 32.22 and Table 32.23 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

(a) PID errors

- If the PID of the received packet is illegal.

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

(d) Overrun and underrun errors

- When the host controller is selected
When the buffer memory is full at the token sending timing in the IN (receiving) direction.
When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

(e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

Table 32.22 Error Detection When a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN flag to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected.

Table 32.23 Error Detection When a Data Packet is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to 1xb (STALL) in both cases when the host controller is selected and the function controller is selected.

32.3.9.2 Data PID

When the function controller is selected, the USB operates as follows in response to the received PID.

IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mData: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mData: Packets are ignored

32.3.9.3 Interval Counter

The isochronous transfer interval can be set using the PIPEPERI.IITV[2:0] bits. The interval counter enables the functions shown in Table 32.24 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 32.24 Interval Counter Function When the Function Controller is Selected

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully 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 the $2^{IITV[2:0]}$ frames.

(1) Counter Initialization When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token while the PID[1:0] bits are 01b (BUF).
2. An SOF is received after reception of data of an OUT token while the PID[1:0] bits are 01b (BUF).

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL)
The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control When the Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.

The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF) by software.

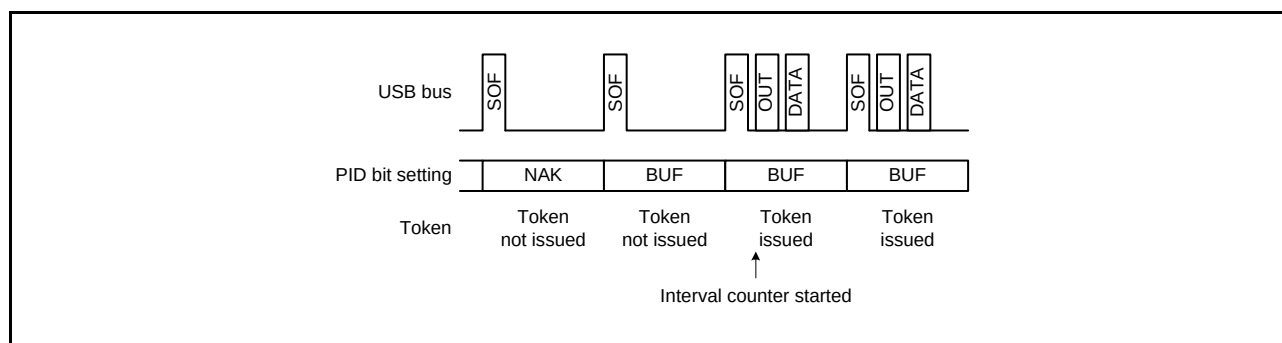


Figure 32.13 Token Issuance When IITV[2:0] = 000b

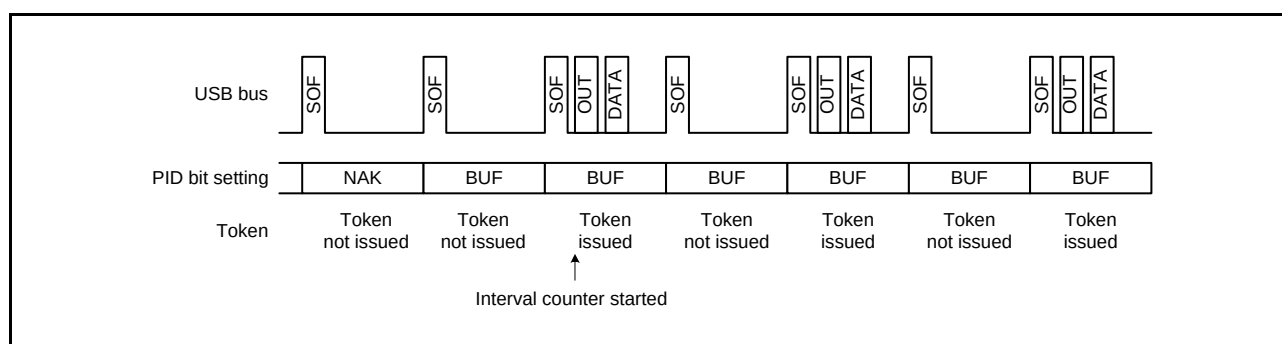


Figure 32.14 Token Issuance When IITV[2:0] = 001b

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).

The USB sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).

The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
(The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.

The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.

The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.

However, when the IITV[2:0] bits are set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.

When the PID[1:0] bits are set to 00b (NAK) by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.

The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] = 000b: The interval counting starts when software has set the PID[1:0] bits for the selected pipe to 01b (BUF).

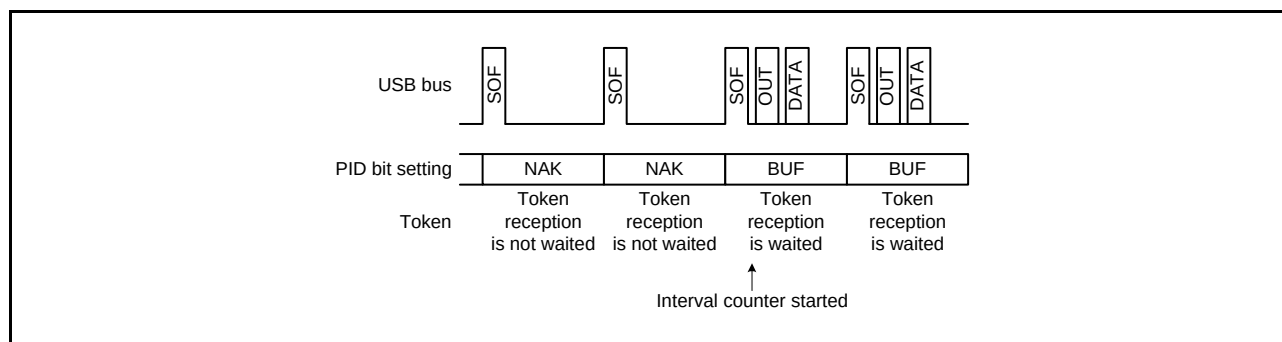


Figure 32.15 Relationship between Frames and Expected Token Reception When IITV[2:0] = 000b

- When the IITV[2:0] $\neq$ 000b: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF).

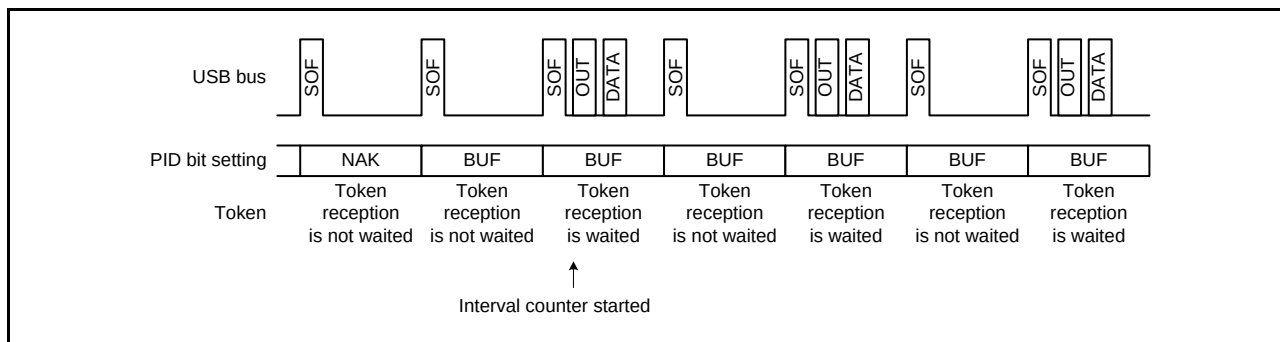


Figure 32.16 Relationship between Frames and Expected Token Reception When IITV[2:0] $\neq$ 000b

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.

When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.

The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.

The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).

The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the 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.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 32.17 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV[2:0] = 000b (every frame) has been set.

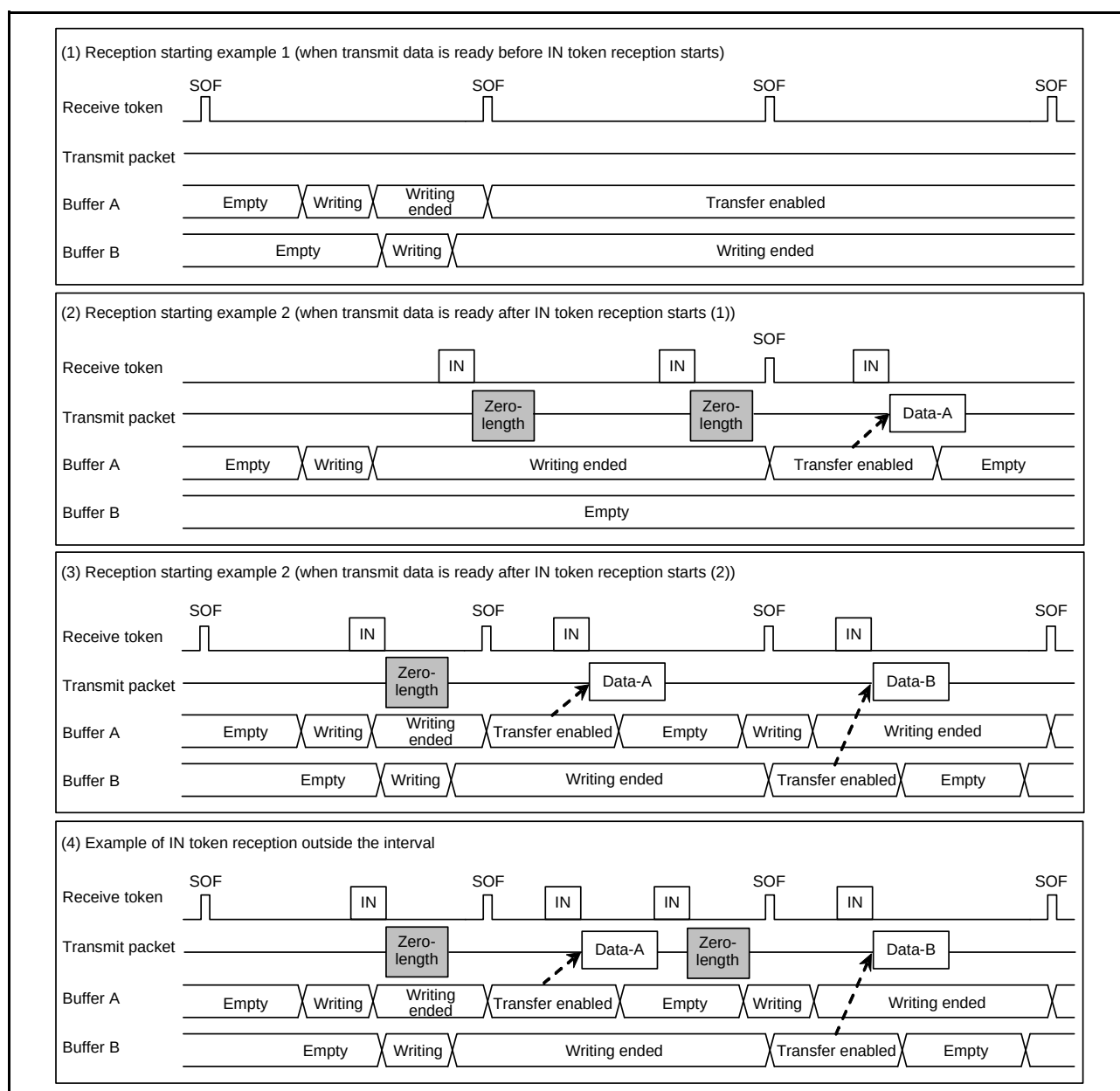


Figure 32.17 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an 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 configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.

The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV[2:0] = 000b
The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV[2:0] ≠ 000b
The buffer flush operation is carried out after the first successful transaction.

Figure 32.18 shows an example of the buffer flush function in the USB. When an unanticipated token is received before the interval frame, the USB sends the write data or a zero-length packet as an underrun error according to the data setup state.

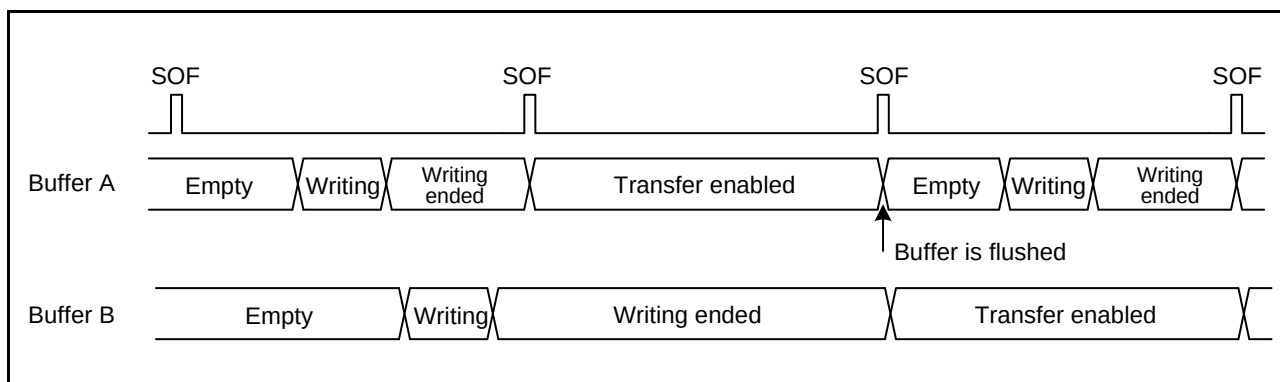


Figure 32.18 Example of Buffer Flush Operation

Figure 32.19 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by ① in the figure, and the buffer flush function is activated.

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

The FRMNUM.OVRN flag 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 figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

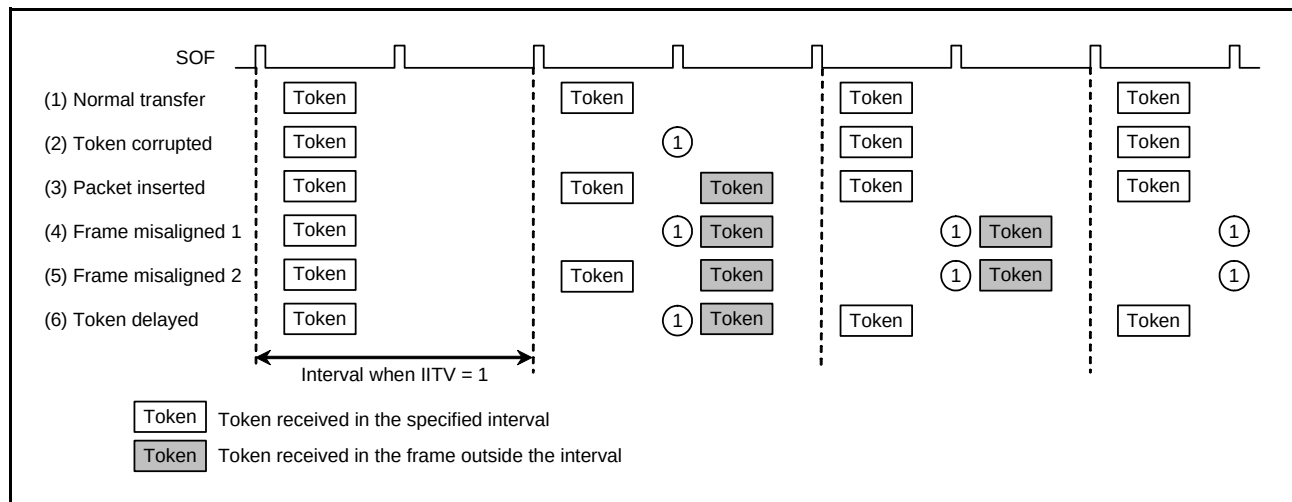


Figure 32.19 Example of Interval Error Occurrence When IITV[2:0] = 001b

32.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing, the FRMNUM.FRNM[10:0] flags are not updated.

32.3.11 Pipe Schedule

32.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 32.25.

Table 32.25 Conditions for Generating a Transaction

Transaction	Conditions for Generation				
	DIR	PID[1:0]	ITV[0]	Buffer State	SUREQ
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

- Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
- Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
- Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

32.3.11.2 Transfer Schedule

This section describes the transfer scheduling within a frame of the USB. After the USB sends an SOF, the transfer is carried out in the sequence described below.

1. Execution of periodic transfers

A pipe is searched in the order of Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.

2. Setup transactions for control transfers

The DCP is checked, and if a setup transaction is possible, it is sent.

3. Execution of bulk transfers, control transfer data stages, and control transfer status stages

A pipe is searched in the order of DCP → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.

When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

32.3.11.3 Enabling USB Communication

Setting the DVSTCTR0.UACT bit to 1 initiates SOF transmission and transaction generation is enabled.

Setting the UACT bit to 0 stops SOF transmission and a suspend state is entered. If the setting of the UACT bit is changed from 1 to 0, processing stops after the next SOF is sent.

32.4 Usage Notes

32.4.1 Setting the Module-Stop Function

Operation of the USB module can be disabled or enabled by setting a bit in the module stop control register B (MSTPCRB). The USB is initially disabled after a reset. Registers in the USB only become accessible after it has been released from the module-stop state. For details, refer to [section 11, Low Power Consumption](#).

32.5 Battery Charging Detection Processing

It is possible to control the processing for data contact detection (D+ line contact check), primary detection (charger detection), and secondary detection (charger verification), which are defined in the battery charging specification.

The following describes required operations for a function device and a host device, individually.

32.5.1 Processing When Function Controller is Selected

The following processing is required when operating the USB module as a portable device for battery charging.

- (1) Detect when the data lines (D+ and D-) have made contact and start the processing for primary detection.
- (2) After primary detection starts, wait 40 ms for masking, and then check the D- voltage level to confirm the primary detection result.
- (3) If the charger is detected during primary detection, also start secondary detection.
- (4) After secondary detection starts, wait 40 ms for masking, and then check the D+ voltage level to confirm the secondary detection result.

For step (1), after VBUS is detected using the VBIT interrupt and the VBSTS flag, wait for 300 to 900 ms by software, and then set the VDPSRCE0 and IDMSINKE0 bits in the USBBCCTRL0 register to 1. Or set the IDPSRCE0 bit to 1, and after a change from high to low on the D+ line is detected using the LNST[1:0] flags, set the IDPSRCE0 bit to 0 and set the VDPSRCE0 and IDMSINKE0 bits to 1. Set the VDPSRCE0 and IDMSINKE0 bits to 0 at the same time.*1

For step (2), set the VDPSRCE0 and IDMSINKE0 bits to 1 and wait 40 ms by software, and then use the CHGDETSTS0 flag to verify the primary detection result.*2

For step (3), if the CHGDETSTS0 flag is set to 1 in step (2), verify that the charger is detected, and then set the VDPSRCE0 and IDMSINKE0 bits to 0 and set the VDMSRCE0 and IDPSINKE0 bits to 1.

For step (4), set the VDMSRCE0 and IDPSINKE0 bits to 1 and wait for 40 ms by software, and then use the PDDETSTS0 flag to verify the secondary detection result.

The following shows the process flow.

Note 1. The battery charging specification describes two implementation methods of the process flow for data contact detection (D+/D- line contact check). One of the methods is to detect a change to logic low due to the pull-down resistor of the host device when the D+ and D- lines have made contact with the target while the D+ line is held at logic high by applying a current of 7 to 13 μ A on the D+ line. The other method is to wait for 300 to 900 ms after VBUS is detected.

Note 2. During primary detection, when the voltage on the D- line is detected to be 0.25 to 0.4 V or above and 0.8 to 2.0 V or below, the target device is recognized as the host device for battery charging (charging downstream port). When using a PHY in which the CHGDETSTS0 flag only indicates that the voltage on the D- line is 0.25 to 0.4 V or above, add the processing to check that the voltage on D- line is 0.8 V to 2.0 V or below using the LNST[1:0] flags, as necessary.

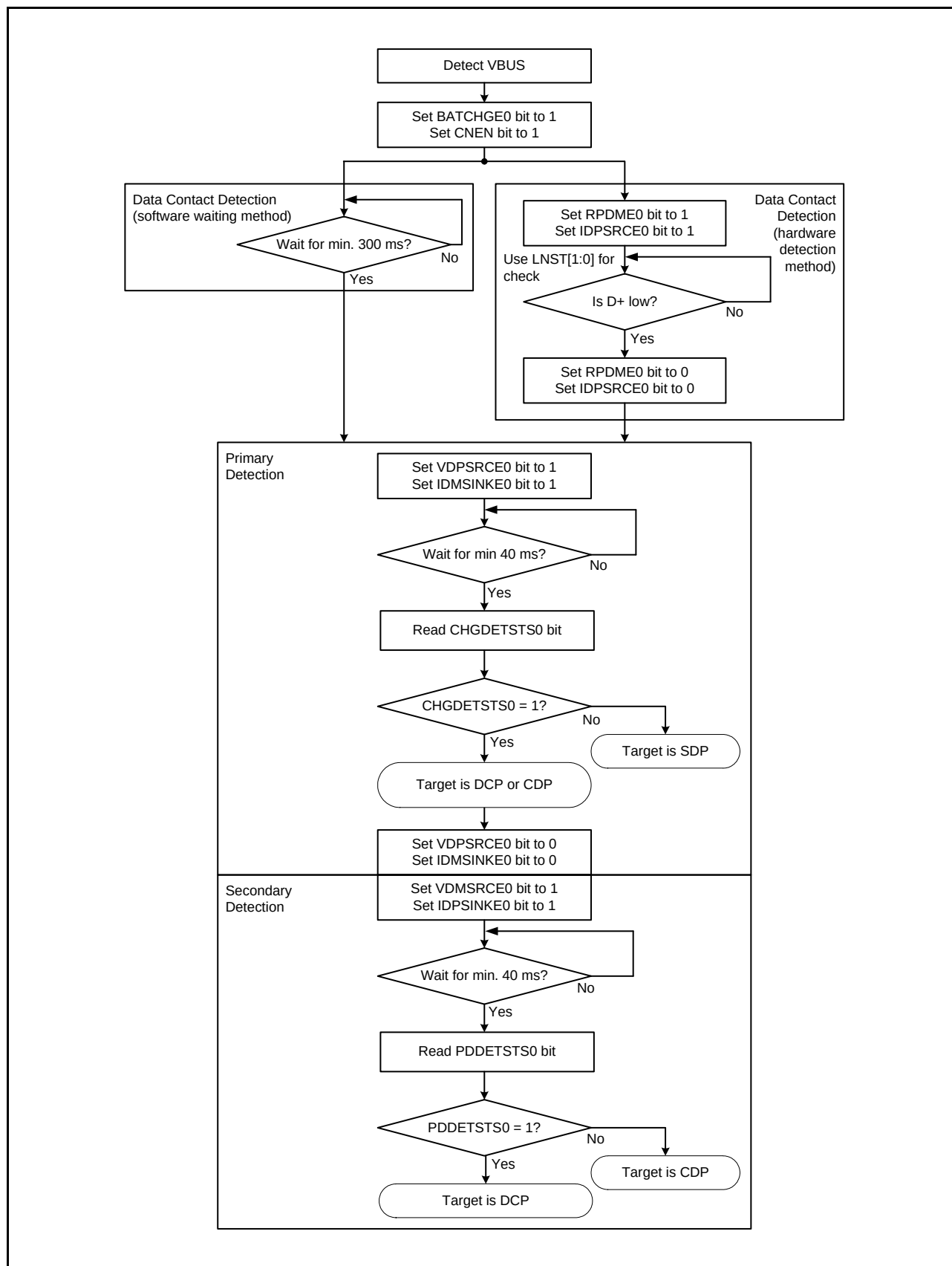


Figure 32.20 Process Flow for Operating as Portable Device

32.5.2 Processing When Host Controller is Selected

The following processing is required when operating the USB module as a charging downstream port for battery charging.

- (1) Start driving the VBUS.
- (2) Enable the portable device detection circuit.
- (3) Monitor the portable device detection signal, and start driving the D⁻ line if the detection signal is high.
- (4) Detect when the portable device detection signal is low level and stop driving the D⁻ line.

Or, the following processing can also be used in accordance with the battery charging specification.

- (A) After disconnection is detected, start driving the D⁻ line within 200 ms.
- (B) After connection is detected, stop driving the D⁻ line within 10 ms.

The D⁻ line must be driven to allow the portable device to detect the primary detection described in section 32.5.1, **Processing When Function Controller is Selected**. The above steps (1) to (4) apply when the portable device detection function is provided by hardware. This method is to drive the D⁻ line when the portable device is detected. Steps (A) and (B) apply when the portable device function is not provided or available by hardware. Regardless of detection of the portable device, the D⁻ line is driven in the disconnected state and the line is not driven in the connected state. In the battery charging specification, either of these methods can be used.

For steps (3) and (4), after a change in the portable device detection signal is detected using the PDDETINT interrupt, the current signal state can be confirmed by reading the PDDETSTS0 flag.

Steps (A) and (B) can be performed only in a software timer.

The following show the process flow for steps (1) to (4) and the process flow for steps (A) to (B), respectively.

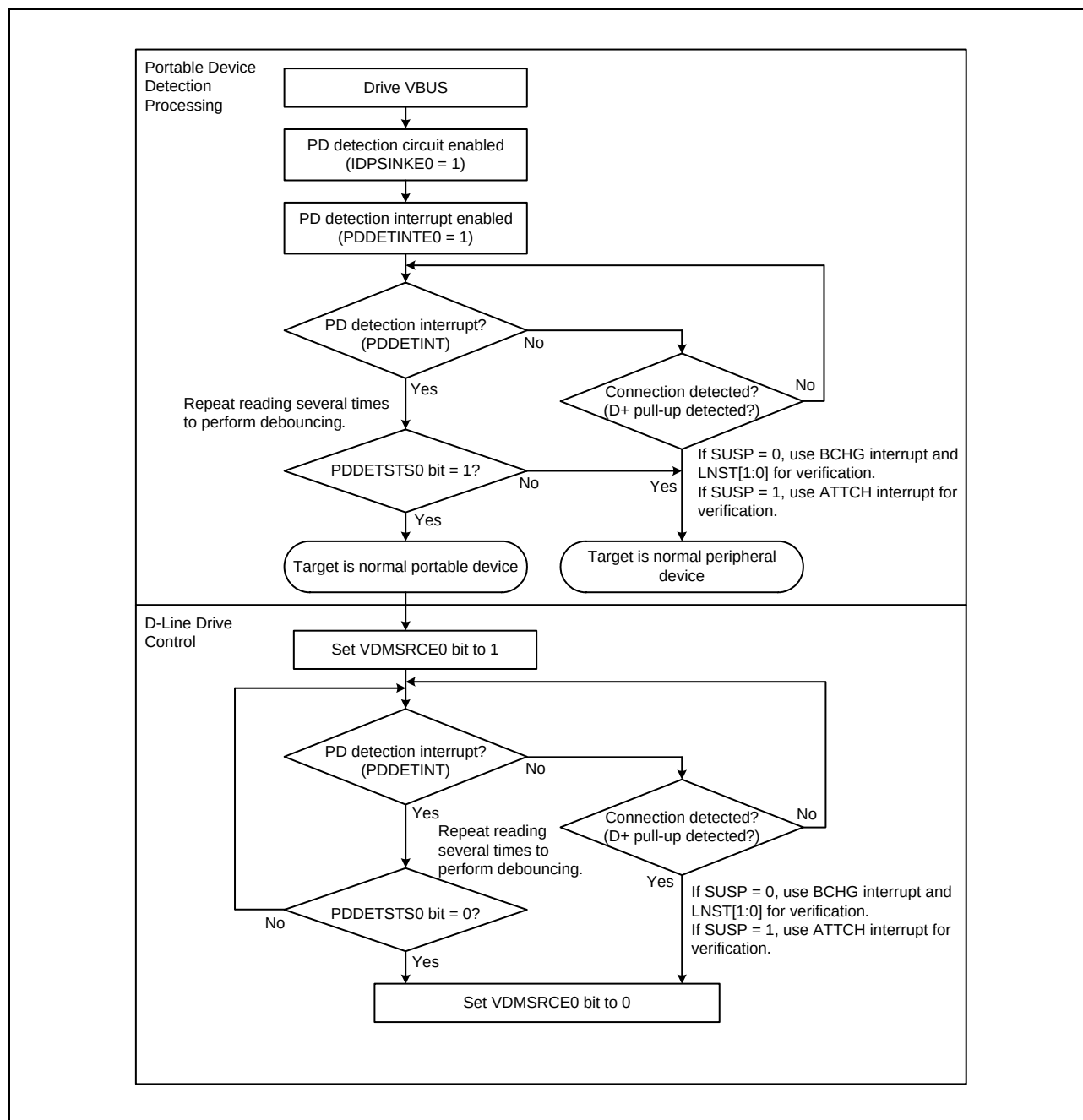


Figure 32.21 Process Flow for Operating as Charging Downstream Port (Steps 1) to (4))

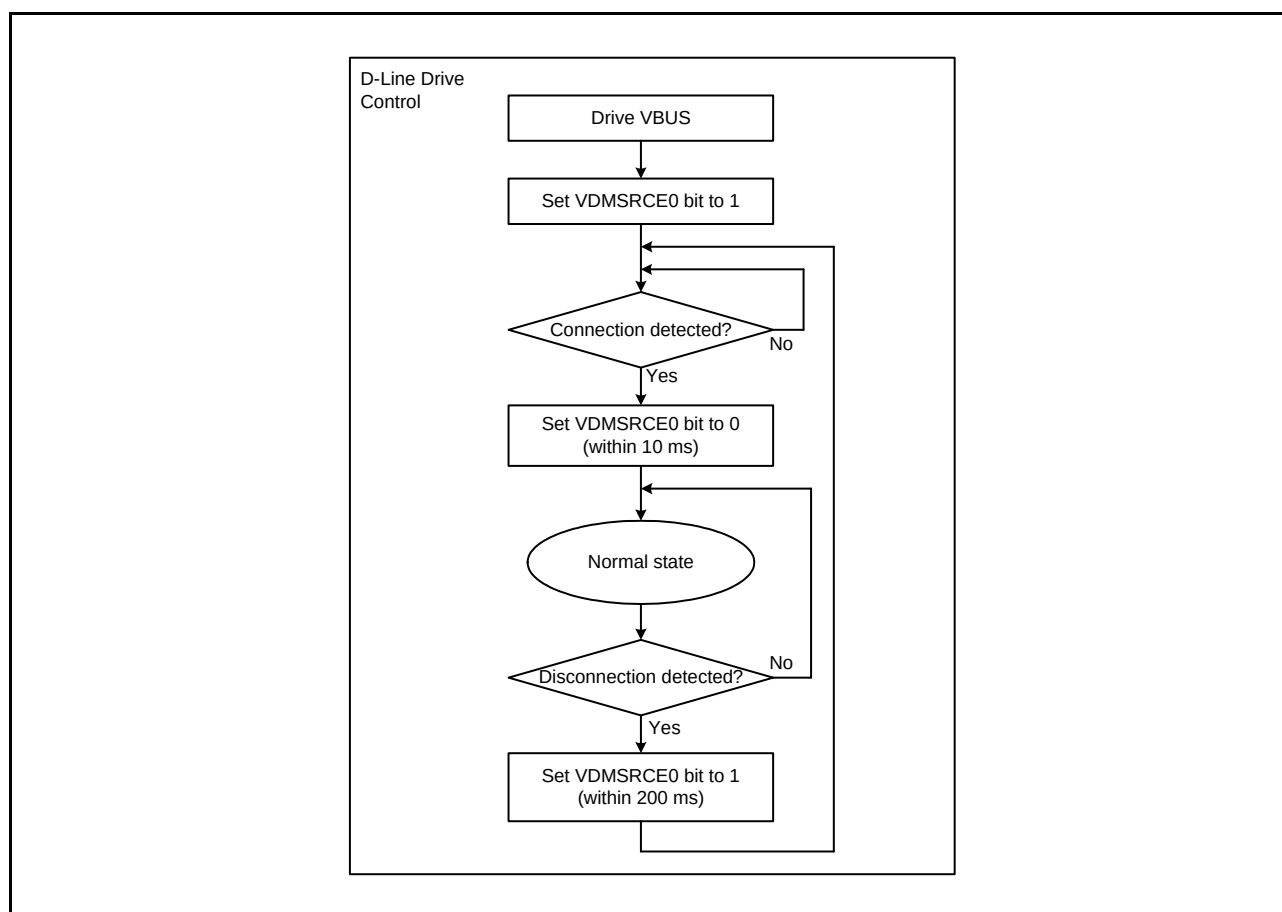


Figure 32.22 Process Flow for Operating as Charging Downstream Port (Steps (A) to (B))

33. Serial Communications Interface (SCIg, SCIH)

This MCU has four independent serial communications interface (SCI) channels. The SCI consists of the SCIg module (SCI1, SCI5, and SCI8) and the SCIH module (SCI12).

The SCIg module (SCI1, SCI5, and SCI8) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI also supports simple SPI interfaces, and simple I²C-bus interfaces when configured for single-master systems.

The SCIH module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.

In this section, "PCLK" is used to refer to PCLKB.

33.1 Overview

Table 33.1 lists the specifications of the SCIg module, Table 33.2 lists the specifications of the SCIH module, and Table 33.3 lists the specifications of the individual SCI channels.

Figure 33.1 shows the block diagram of SCI1 and SCI8, Figure 33.2 shows the block diagram of SCI5, and Figure 33.3 shows the block diagram of SCI12 (SCIH).

Table 33.1 SCIg Specifications (1/2)

Item		Description
Serial communication modes		- Asynchronous - Clock synchronous - Smart card interface - Simple I²C-bus - Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 33.4 to Table 33.6.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I ² C mode)
Low power consumption function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTS _n # and RTS _n # pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXD _n pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI5)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
Noise cancellation		The signal paths from input on the RXD _n pins incorporate digital noise filters.

Table 33.1 SCIg Specifications (2/2)

Item	Description	
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I ² C mode	Transfer format	I ² C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 33.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function (supported by SC15 only)		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Note 1. In simple I²C mode, only MSB first is available.

Table 33.2 SCIH Specifications (1/2)

Item	Description
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I²C-bus • Simple SPI bus
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	Refer to Table 33.4 to Table 33.7.
Data transfer	Selectable as LSB first or MSB first transfer*1
Interrupt sources	Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I ² C mode)
Low power consumption function	Module stop state can be set.

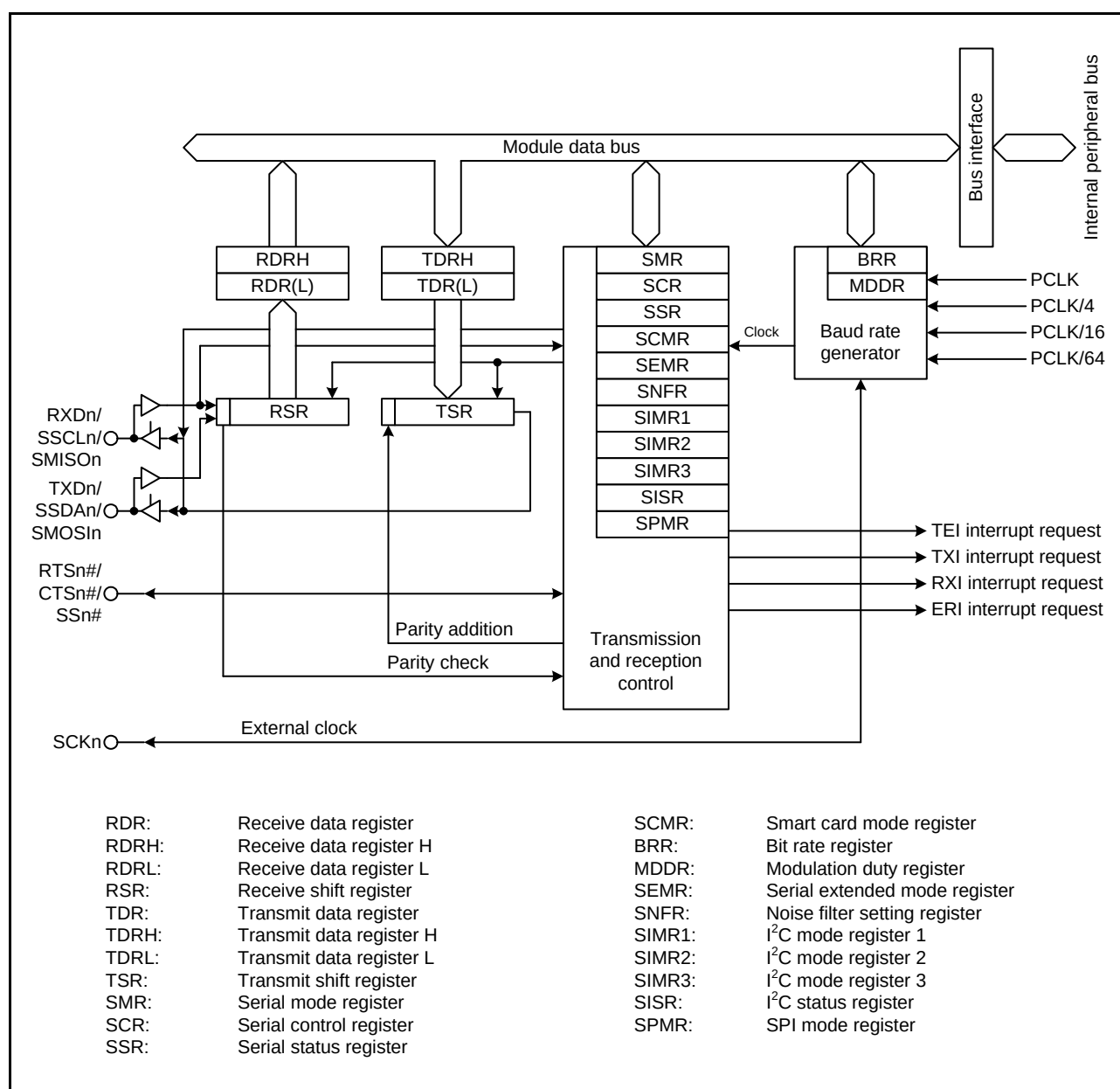
Table 33.2 SCIf Specifications (2/2)

Item	Description	
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I ² C mode	Transfer format	I ² C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 33.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Extended serial mode	Start Frame transmission	- Output of a low level as the Break Field over a specified width and generation of interrupts on completion - Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	- Detection of the Break Field low width and generation of an interrupt on detection - Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match - Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. - A priority interrupt bit can be set in Control Field 1. - Handling of Start Frames that do not include a Break Field - Handling of Start Frames that do not include a Control Field 0 - Function for measuring bit rates
	I/O control function	- Selectable polarity for TXDX12 and RXDX12 signals - Selection of a digital filter for the RXDX12 signal - Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin - Selectable timing for the sampling of data received through RXDX12
	Timer function	Usable as a reload timer
	Bit rate modulation function	Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I²C mode, only MSB first is available.

Table 33.3 Functions of SCI Channels

Item	SCI1, SCI8	SCI5	SCI12
Asynchronous mode	Available	Available	Available
Clock synchronous mode	Available	Available	Available
Smart card interface mode	Available	Available	Available
Simple I ² C mode	Available	Available	Available
Simple SPI mode	Available	Available	Available
Extended serial mode	Not available	Not available	Available
TMR clock input	Not available	Available	Available
Event link function	Not available	Available	Not available

**Figure 33.1 Block Diagram of SCIg (SCI1 and SCI8)**

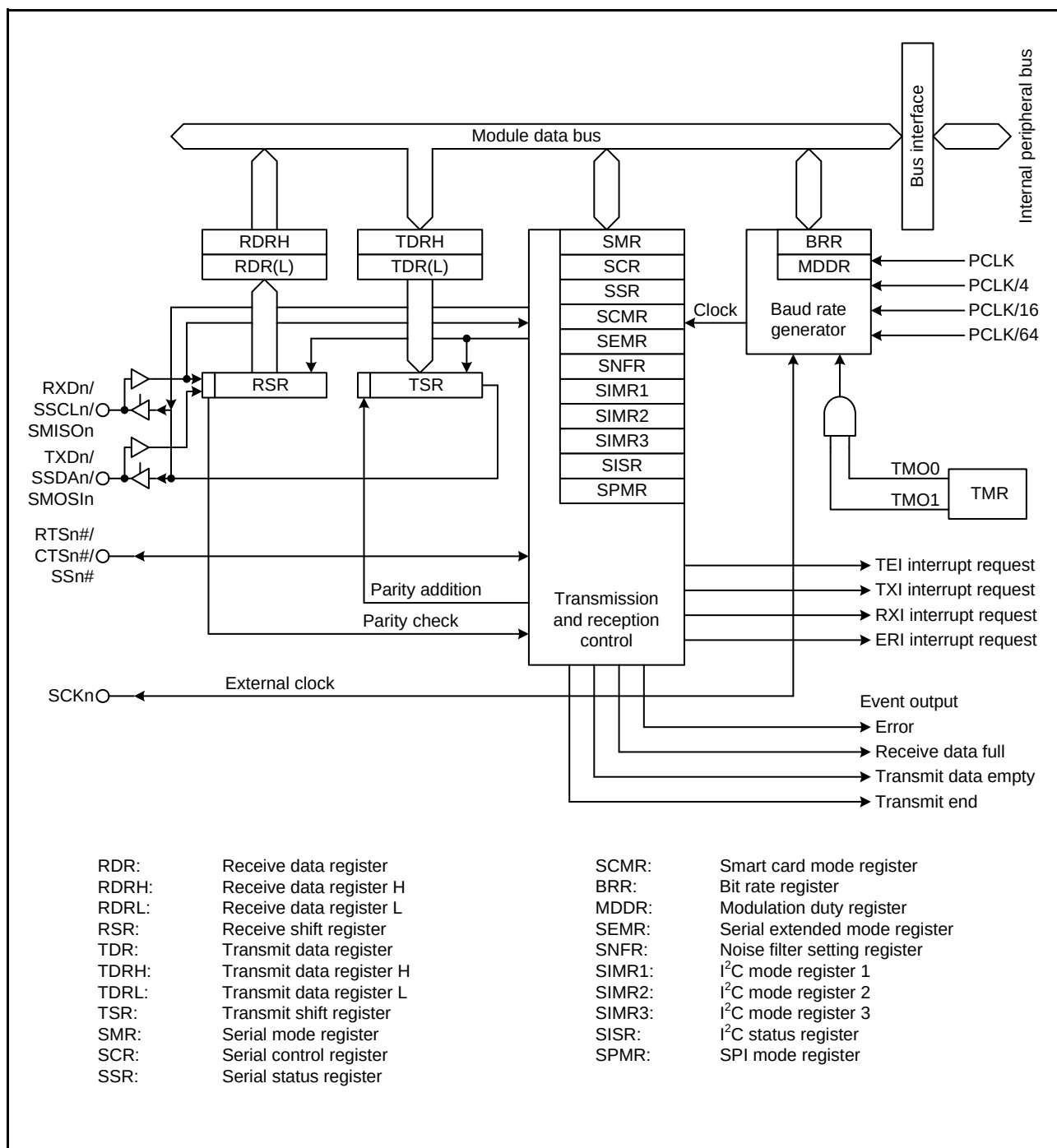


Figure 33.2 Block Diagram of SCIg (SCI5)

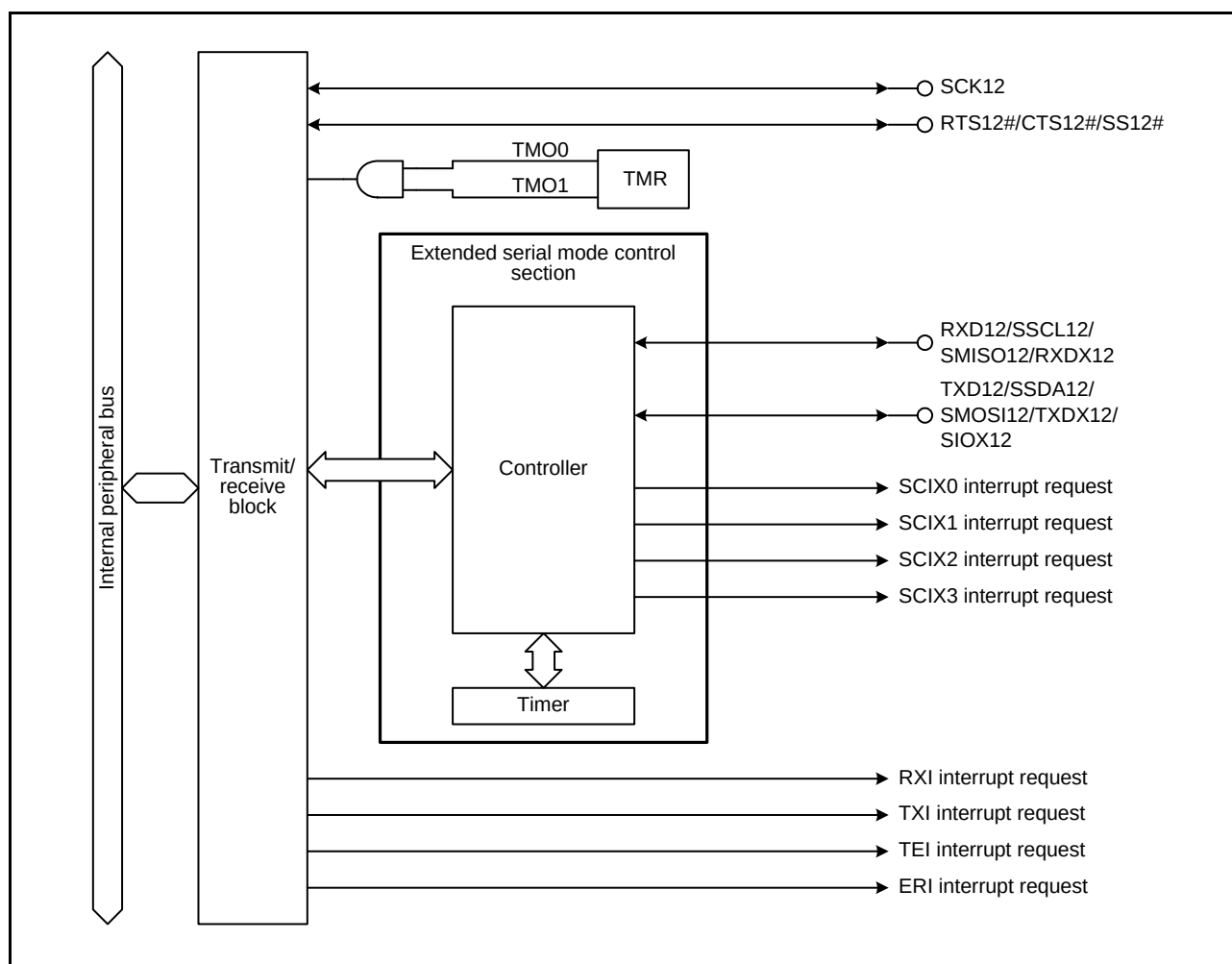


Figure 33.3 Block Diagram of SCIf (SCI12)

Table 33.4 to Table 33.7 list the pin configuration of the SCIs for the individual modes.

Table 33.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI8	SCK8	I/O	SCI8 clock input/output
	RXD8	Input	SCI8 receive data input
	TXD8	Output	SCI8 transmit data output
	CTS8#/RTS8#	I/O	SCI8 transfer start control input/output
SCI12	SCK12	I/O	SCI12 clock input/output
	RXD12	Input	SCI12 receive data input
	TXD12	Output	SCI12 transmit data output
	CTS12#/RTS12#	I/O	SCI12 transfer start control input/output

Table 33.5 SCI Pin Configuration in Simple I²C Mode

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I ² C clock input/output
	SSDA1	I/O	SCI1 I ² C data input/output
SCI5	SSCL5	I/O	SCI5 I ² C clock input/output
	SSDA5	I/O	SCI5 I ² C data input/output
SCI8	SSCL8	I/O	SCI8 I ² C clock input/output
	SSDA8	I/O	SCI8 I ² C data input/output
SCI12	SSCL12	I/O	SCI12 I ² C clock input/output
	SSDA12	I/O	SCI12 I ² C data input/output

Table 33.6 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI5	SCK5	I/O	SCI5 clock input/output
	SMISO5	I/O	SCI5 slave transmit data input/output
	SMOSI5	I/O	SCI5 master transmit data input/output
	SS5#	Input	SCI5 chip select input
SCI8	SCK8	I/O	SCI8 clock input/output
	SMISO8	I/O	SCI8 slave transmit data input/output
	SMOSI8	I/O	SCI8 master transmit data input/output
	SS8#	Input	SCI8 chip select input

Table 33.6 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
SCI12	SCK12	I/O	SCI12 clock input/output
	SMISO12	I/O	SCI12 slave transmit data input/output
	SMOSI12	I/O	SCI12 master transmit data input/output
	SS12#	Input	SCI12 chip select input

Table 33.7 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

33.2 Register Descriptions

33.2.1 Receive Shift Register (RSR)

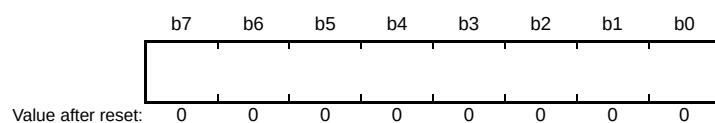
The RSR register is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data.

When one frame of data has been received, it is automatically transferred to the RDR register.

The RSR register cannot be directly accessed by the CPU.

33.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI8.RDR 0008 A105h, SCI12.RDR 0008 B305h



The RDR register is an 8-bit register that stores receive data.

When one frame of serial data has been received, the received serial data is transferred from the RSR register to the RDR register. Then the RSR register can receive the next data.

Since the RSR and RDR registers function as a double buffer in this way, continuous receive operations can be performed.

Read the RDR register only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from the RDR register, an overrun error occurs.

The RDR register cannot be written to by the CPU.

33.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

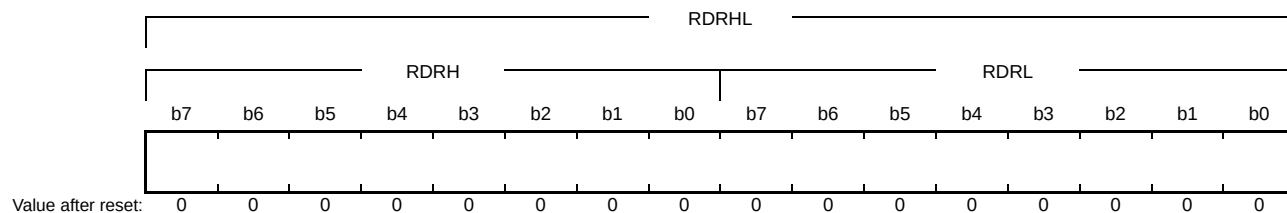
Address(es): SCI1.RDRH 0008 A030h, SCI5.RDRH 0008 A0B0h, SCI8.RDRH 0008 A110h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI1.RDRL 0008 A031h, SCI5.RDRL 0008 A0B1h, SCI8.RDRL 0008 A111h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI1.RDRHL 0008 A030h, SCI5.RDRHL 0008 A0B0h, SCI8.RDRHL 0008 A110h, SCI12.RDRHL 0008 B310h



The RDRH and RDRL registers are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

The RDRL register is the shadow register of the RDR register; i.e. access to the RDRL register is equivalent to access to the RDR register.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

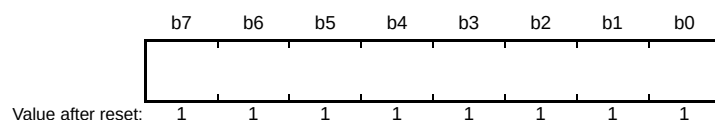
Read the RDRH and RDRL registers should be performed only once in the order from the RDRH register to the RDRL register when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from the RDRL register.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in the RDRH register are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

33.2.4 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI8.TDR 0008 A103h, SCI12.TDR 0008 B303h



The TDR register is an 8-bit register that stores transmit data.

When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

33.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

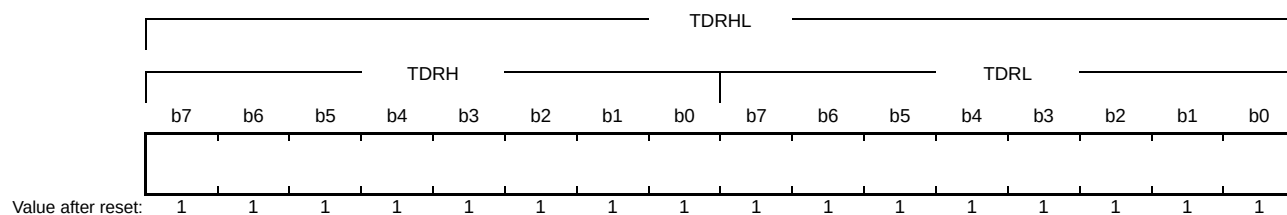
Address(es): SCI1.TDRH 0008 A02Eh, SCI5.TDRH 0008 A0AEh, SCI8.TDRH 0008 A10Eh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI1.TDRL 0008 A02Fh, SCI5.TDRL 0008 A0AFh, SCI8.TDRL 0008 A10Fh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI1.TDRHL 0008 A02Eh, SCI5.TDRHL 0008 A0AEh, SCI8.TDRHL 0008 A10Eh, SCI12.TDRHL 0008 B30Eh



The TDRH and TDRL registers are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.

The TDRL register is the shadow register of the TDR register; i.e. access to the TDRL register is equivalent to access to the TDR register.

When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to the TSR register; i.e., transmitting is started.

The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in the RDRH register are fixed to 1. These bits are read as 1. The write value should be 1.

Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from the TDRH register to the TDRL register when a transmit data empty interrupt (TXI) request is issued.

The TDRHL register can be accessed in 16-bit units.

33.2.6 Transmit Shift Register (TSR)

The TSR register is a shift register that transmits serial data.

To perform serial data transmission, the SCI first automatically transfers transmit data from the TDR register to the TSR register, and then sends the data to the TXDn pin.

The TSR register cannot be directly accessed by the CPU.

33.2.7 Serial Mode Register (SMR)

Some bits in the SMR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI8.SMR 0008 A100h, SCI12.SMR 0008 B300h

b7	b6	b5	b4	b3	b2	b1	b0
CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode or simple I ² C mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).

Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.

Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.

Note 4. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to section 33.2.11, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.

The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.

Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.

Selects in combination with the SCMR.CHR1 bit.

In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SMR 0008 A020h, SMCI5.SMR 0008 A0A0h, SMCI8.SMR 0008 A100h, SMCI12.SMR 0008 B300h

b7	b6	b5	b4	b3	b2	b1	b0
GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 33.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).

Note 2. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 33.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 33.6.4, Receive Data Sampling Timing and Reception Margin.

Table 33.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).

For details on the usage of this bit in smart card interface mode, refer to section 33.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.

The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, refer to section 33.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

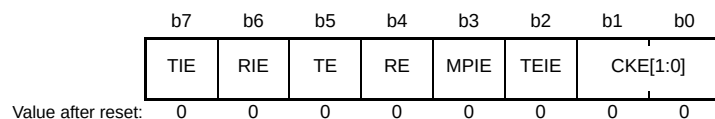
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 33.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 33.6.8, Clock Output Control.

33.2.8 Serial Control Register (SCR)

Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI8.SCR 0008 A102h, SCI12.SCR 0008 B302h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	(Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin becomes high-impedance. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or TMR clock*2 • The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • The SCKn pin becomes high-impedance when the TMR clock*2 is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when the SMR.MP bit is 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*3
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*3
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. TMR clock is selectable for SCI5 and SCI12.

Note 3. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple I²C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI).

In this case, the TEIE bit can be used to enable or disable the STI.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 33.4, Multi-Processor Communications Function.

When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags RDRF, ORER, and FER to 1 is disabled.

When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags RDRF, ORER, and FER to 1 is enabled.

Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMC11.SCR 0008 A022h, SMC15.SCR 0008 A0A2h, SMC18.SCR 0008 A102h, SMC112.SCR 0008 B302h

b7	b6	b5	b4	b3	b2	b1	b0
TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	- When SMR.GM = 0 - b1 b0 0 0: Output disabled The SCKn pin becomes high-impedance. 0 1: Clock output 1 x: Setting prohibited - When SMR.GM = 1 - b1 b0 0 0: Output fixed low - x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 33.12, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 33.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

33.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI8.SSR 0008 A104h, SCI12.SSR 0008 B304h

b7	b6	b5	b4	b3	b2	b1	b0
TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.

Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to the RDR register, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the FER flag after reading FER = 1
When setting the FER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
In the RDR register, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMC11.SSR 0008 A024h, SMC15.SSR 0008 A0A4h, SMC18.SSR 0008 A104h, SMC112.SSR 0008 B304h

b7	b6	b5	b4	b3	b2	b1	b0
TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overflow Error Flag	0: No overflow error occurred 1: An overflow error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.

Note 2. Write 1 when writing is necessary.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0 (serial transmission is disabled)
When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
The set timing is determined by register settings as listed below.
When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to the ERS flag after reading ERS = 1
When setting the ERS flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
In the RDR register, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

33.2.10 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI8.SCMR 0008 A106h, SCI12.SCMR 0008 B306h, SMC11.SCMR 0008 A026h, SMC15.SCMR 0008 A0A6h, SMC18.SCMR 0008 A106h, SMC12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I ² C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert*2, *3	0: Data bits in the TDR register are transferred to the TSR register as they are. Data bits in the RSR register are transferred to the RDR register as they are. 1: Data bits in the TDR register are transferred to the TSR register with inverting. Data bits in the RSR register are transferred to the RDR register with inverting.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction*2, *4	0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1*5	Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*6	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 33.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

Note 2. This bit can be used in the smart card interface mode, asynchronous mode (multi-processor mode), clock synchronous mode, and simple SPI mode.

Note 3. Set this bit to 0 if operation is to be in simple I²C mode.

Note 4. Set this bit to 1 if operation is to be in simple I²C mode.

Note 5. This bit is only valid in asynchronous mode. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.

Note 6. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I²C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

This bit is used to invert the logic level of the data bits when the data is transferred between data register and shift register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SMR.PM bit.

CHR1 Bit (Character Length 1)

Selects the data length of transmit/receive data.

Selects in combination with the SMR.CHR bit.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

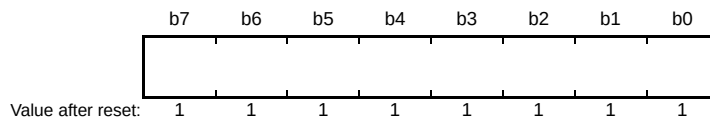
Table 33.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles ($S = 93$)* ¹
0	0	1	128 clock cycles ($S = 128$)* ¹
0	1	0	186 clock cycles ($S = 186$)* ¹
0	1	1	512 clock cycles ($S = 512$)* ¹
1	0	0	32 clock cycles ($S = 32$)* ¹ (Initial Value)
1	0	1	64 clock cycles ($S = 64$)* ¹
1	1	0	372 clock cycles ($S = 372$)* ¹
1	1	1	256 clock cycles ($S = 256$)* ¹

Note 1. S is the value of S in the BRR register (refer to section 33.2.11, Bit Rate Register (BRR)).

33.2.11 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI8.BRR 0008 A101h, SCI12.BRR 0008 B301h



The BRR register is an 8-bit register that adjusts the bit rate.

As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 33.10 shows the relationship between the setting (N) in the BRR register and the bit rate (B) for normal asynchronous mode, multi-processor communication, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I²C mode.

The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 33.10 Relationship between N Setting in the BRR Register and Bit Rate B

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM bit	ABCS bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I ² C*1			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)

N: BRR setting for on-chip baud rate generator ($0 \leq N \leq 255$)

PCLK: Operating frequency (MHz)

n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 33.12 and Table 33.13.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 33.11 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I ² C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 33.12 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 33.13 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 33.14 lists examples of N settings in the BRR register in normal asynchronous mode. Table 33.15 lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 33.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 33.20. Examples of BRR (N) settings in simple I²C mode are listed in Table 33.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 33.6.4, Receive Data Sampling Timing and Reception Margin. Table 33.16 and Table 33.19 list the maximum bit rates with external clock input.

When either the SEMR.ABCS or BGDM bit is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 33.14. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 33.14 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34	0	19	0.00
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	-0.17	3	70	0.03	3	75	0.48	3	79	-0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	-0.93	0	51	0.16	0	55	0.00	0	58	-0.69	0	63	0.00
19200	0	22	-0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	-1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	-2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	20			25			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	-0.25	3	110	-0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	-0.35
300	2	129	0.16	2	162	-0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	-0.35
1200	1	129	0.16	1	162	-0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	-0.35
4800	0	129	0.16	0	162	-0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	-0.35
19200	0	32	-1.36	0	40	-0.76	0	48	-0.35
31250	0	19	0.00	0	24	0.00	0	29	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73

Note: This is an example when the ABCS and BGDM bits in the SEMR register are 0.
 When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
 When both ABCS and BGDM bits in the SEMR register are set to 1, the bit rate increases four times.