Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	H8S/2000
Core Size	16-Bit
Speed	20MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	86
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	128-BFQFP
Supplier Device Package	128-QFP (14x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/d12320vf20v

Type	Symbol	Pin No. TFP-120	Pin No. FP-128B	I/O	Name and Function
16-bit timer pulse unit (TPU)	$TIOCA_4$, $TIOCB_4$	67, 66	75, 74	I/O	**Input capture/output compare match A4 and B4:** The TGR4A and TGR4B input capture input or output compare output, or PWM output pins.
	$TIOCA_5$, $TIOCB_5$	65, 64	73, 72	I/O	**Input capture/output compare match A5 and B5:** The TGR5A and TGR5B input capture input or output compare output, or PWM output pins.
Programmable pulse generator (PPG)	PO_{15} to PO_0	105 to 112, 64 to 71	115 to 122, 72 to 79	Output	**Pulse output 15 to 0:** Pulse output pins.
8-bit timer	TMO_0, TMO_1	65, 64	73, 72	Output	**Compare match output:** The compare match output pins.
	$TMCI_0$, $TMCI_1$	68, 66	76, 74	Input	**Counter external clock input:** Input pins for the external clock input to the counter.
	$TMRI_0$, $TMRI_1$	69, 67	77, 75	Input	**Counter external reset input:** The counter reset input pins.
Watchdog timer (WDT)	$\overline{WDTOVF}$*5	72	80	Output	**Watchdog timer overflow:** The counter overflow signal output pin in watchdog timer mode.
Serial communication interface (SCI)/ smart card interface	TxD_2, TxD_1, TxD_0	89, 54, 53	97, 60, 59	Output	**Transmit data (channel 0, 1, 2):** Data output pins.
	RxD_2, RxD_1, RxD_0	90, 56, 55	98, 62, 61	Input	**Receive data (channel 0, 1, 2):** Data input pins.
	SCK_2, SCK_1, SCK_0	91, 58, 57	101, 64, 63	I/O	**Serial clock (channel 0, 1, 2):** Clock I/O pins.
A/D converter	AN_7 to AN_0	102 to 95	112 to 105	Input	**Analog 7 to 0:** Analog input pins.
	$\overline{ADTRG}$	92	102	Input	**A/D conversion external trigger input:** Pin for input of an external trigger to start A/D conversion.
D/A converter	DA_1, DA_0	102, 101	112, 111	Output	**Analog output:** D/A converter analog output pins.

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

The chip controls interrupts by means of an interrupt controller. The interrupt controller has the following features. This chapter assumes the maximum number of interrupt sources available in these series—nine external interrupts and 52 internal interrupts.

- Two interrupt control modes
  - — Either of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR)
- Priorities settable with IPRs
  - — Interrupt priority registers (IPRs) are provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI
  - — NMI is assigned the highest priority level of 8, and can be accepted at all times
- Independent vector addresses
  - — All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine
- Nine external interrupt pins
  - — NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI
  - — Falling edge, rising edge, or both edge detection, or level sensing, can be selected for IRQ7 to IRQ0
- DTC and DMAC* control
  - — DTC and DMAC* activation is controlled by means of interrupts

Note: * The DMAC is not supported in the H8S/2321.

5.5 Usage Notes

5.5.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction.

In other words, when an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared.

Figure 5.8 shows an example in which the TGIEA bit in the TPU's TIER0 register is cleared to 0.

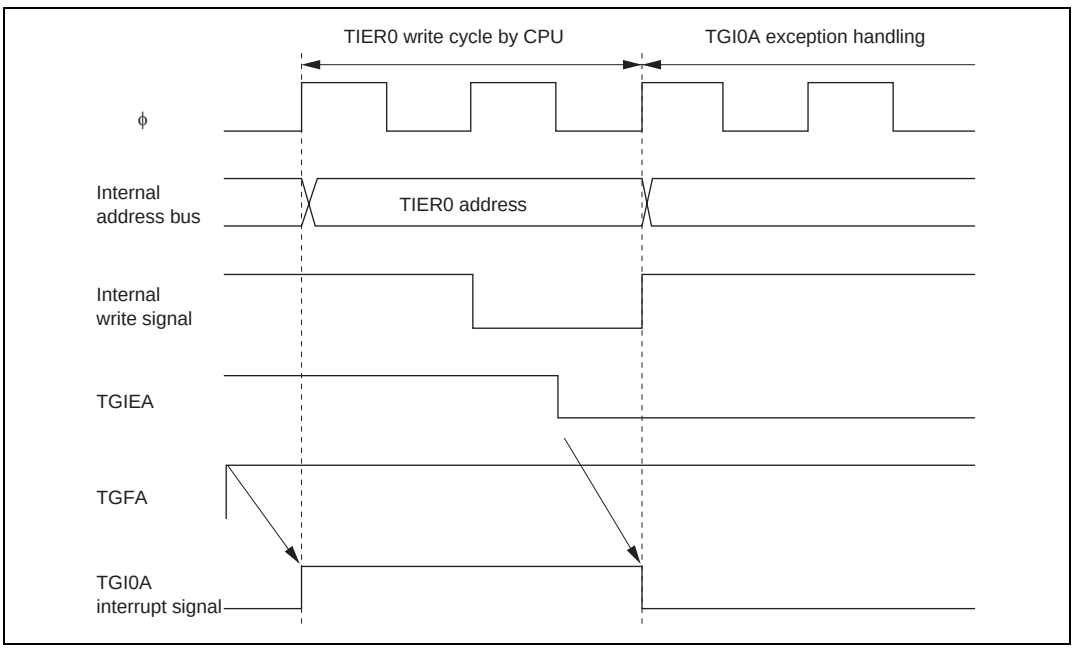

Figure 5.8 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

5.6.3 Operation

The interrupt controller has three main functions in DTC and DMAC* control.

Selection of Interrupt Source: With the DMAC*, the activation source is input directly to each channel. The activation source for each DMAC* channel is selected with bits DTF3 to DTF0 in DMACR. Whether the selected activation source is to be managed by the DMAC* can be selected with the DTA bit of DMABCR. When the DTA bit is set to 1, the interrupt source constituting that DMAC* activation source is not a DTC activation source or CPU interrupt source.

For interrupt sources other than interrupts managed by the DMAC*, it is possible to select DTC activation request or CPU interrupt request with the DTCE bit of DTCERA to DTCERF in the DTC.

After a DTC data transfer, the DTCE bit can be cleared to 0 and an interrupt request sent to the CPU in accordance with the specification of the DISEL bit of MRB in the DTC.

When the DTC has performed the specified number of data transfers and the transfer counter value is zero, the DTCE bit is cleared to 0 and an interrupt request is sent to the CPU after the DTC data transfer.

Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See section 7.6, Interrupts, and section 8.3.3, DTC Vector Table, for the respective priorities.

With the DMAC*, the activation source is input directly to each channel.

Operation Order: If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

If the same interrupt is selected as a DMAC* activation source and a DTC activation source or CPU interrupt source, operations are performed for them independently according to their respective operating statuses and bus mastership priorities.

Table 5.11 summarizes interrupt source selection and interrupt source clearance control according to the settings of the DTA bit of DMABCR in the DMAC*, the DTCE bit of DTCERA to DTCERF in the DTC, and the DISEL bit of MRB in the DTC.

Note: * The DMAC is not supported in the H8S/2321.

6.3.4 Advanced Mode

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode. The bus specifications described here cover basic items only, and the sections on each memory interface (6.4, Basic Bus Interface, 6.5, DRAM Interface (Not supported in the H8S/2321), and 6.7, Burst ROM Interface) should be referred to for further details.

Area 0: Area 0 includes on-chip ROM*1, and in ROM-disabled expansion mode, all of area 0 is external space. In the ROM-enabled expansion mode, the space excluding on-chip ROM*1 is external space.

When area 0 external space is accessed, the $\overline{CS0}$ signal can be output.

Either basic bus interface or burst ROM interface can be selected for area 0.

Areas 1 and 6: In external expansion mode, all of area 1 and area 6 is external space.

When area 1 and 6 external space is accessed, the $\overline{CS1}$ and $\overline{CS6}$ pin signals respectively can be output.

Only the basic bus interface can be used for areas 1 and 6.

Areas 2 to 5: In external expansion mode, all of area 2 to area 5 is external space.

When area 2 to 5 external space is accessed, signals $\overline{CS2}$ to $\overline{CS5}$ can be output.

Basic bus interface or DRAM interface*2 can be selected for areas 2 to 5. With the DRAM interface*2, signals $\overline{CS2}$ to $\overline{CS5}$ are used as $\overline{RAS}$ signals.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In external expansion mode, the space excluding the on-chip RAM and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space .

When area 7 external space is accessed, the $\overline{CS7}$ signal can be output.

Only the basic bus interface can be used for the area 7 memory interface.

Notes: 1. Only applies to versions with ROM.
2. The DRAM interface is not supported in the H8S/2321.

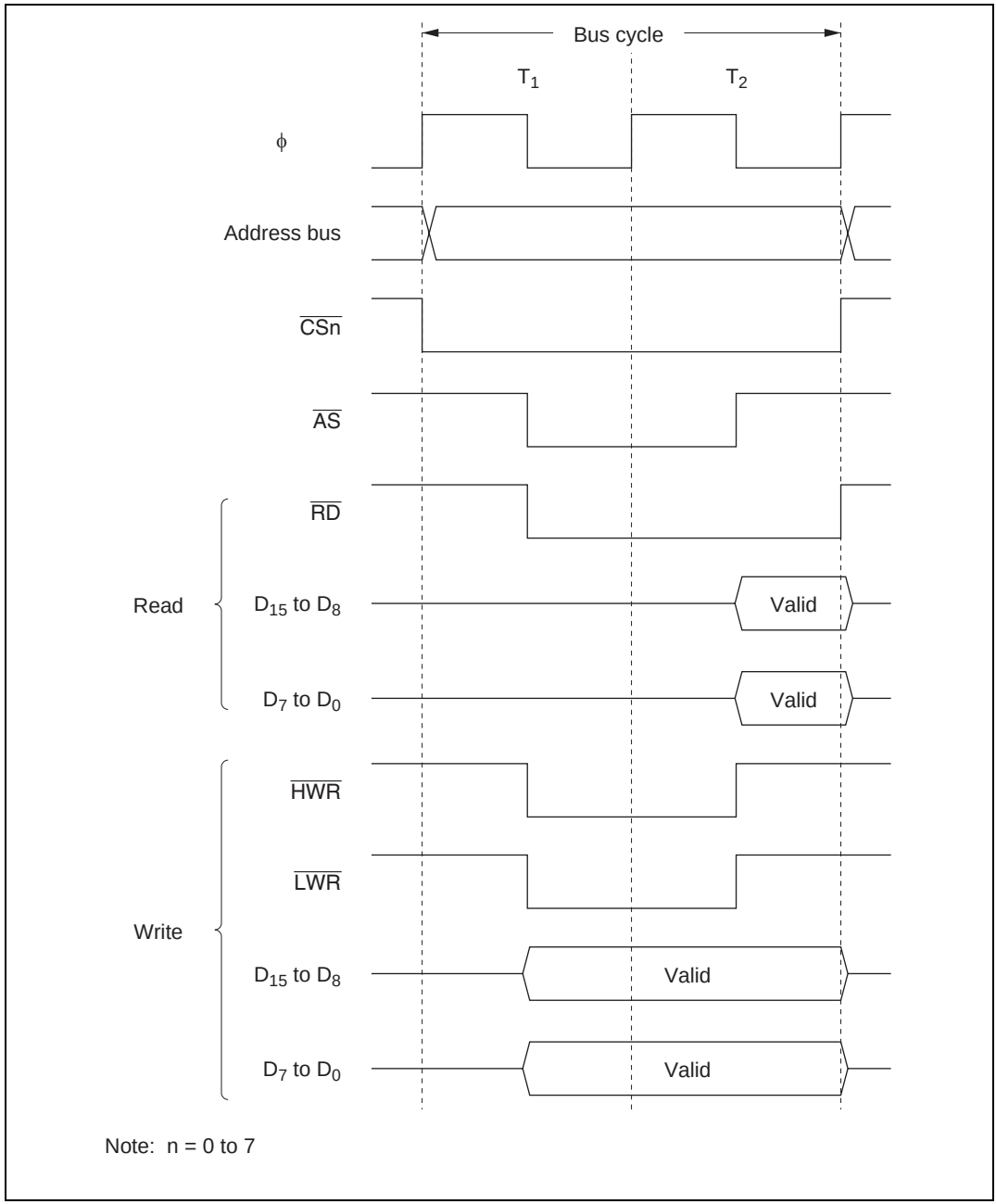

Figure 6.10 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

Figure 7.34 shows an example of single address transfer using the write data buffer function. In this example, the CPU program area is in on-chip memory.

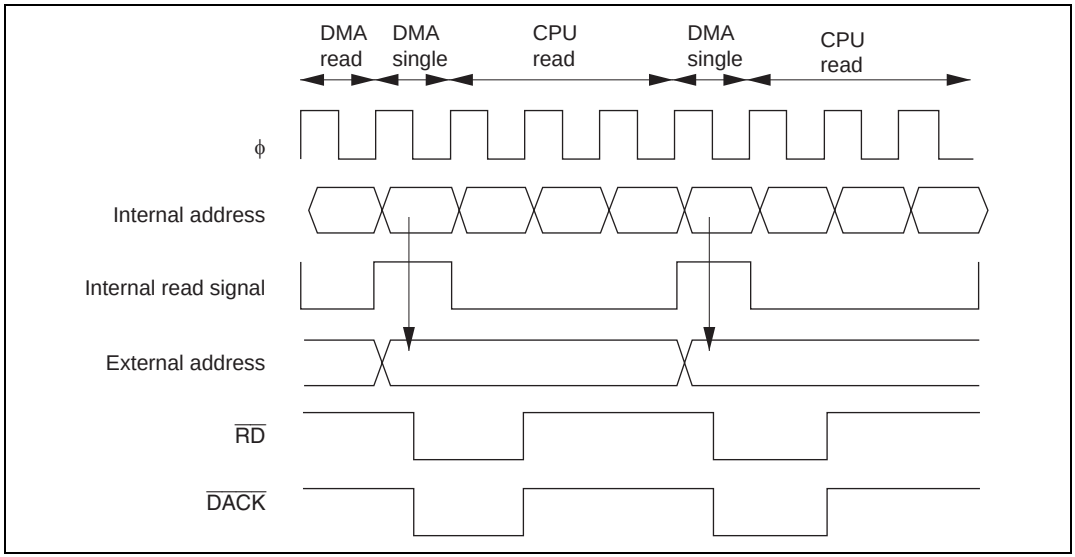

Figure 7.34 Example of Single Address Transfer Using Write Data Buffer Function

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore, $\overline{\text{DREQ}}$ pin sampling is started one state after the start of the DMA write cycle or single address transfer.

7.5.13 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7.13 summarizes the priority order for DMAC channels.

Table 7.13 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		↑
Channel 1A	Channel 1	
Channel 1B		Low

8.2.9 Module Stop Control Register (MSTPCR)

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

MSTPCR is a 16-bit readable/writable register that performs module stop mode control.

When the MSTP14 bit in MSTPCR is set to 1, DTC operation stops at the end of the bus cycle and a transition is made to module stop mode. However, 1 cannot be written in the MSTP14 bit while the DTC is operating. For details, see section 21.5, Module Stop Mode.

MSTPCR is initialized to H'3FFF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 14—Module Stop (MSTP14): Specifies the DTC module stop mode.

Bit 14 MSTP14	Description	
0	DTC module stop mode cleared	(Initial value)
1	DTC module stop mode set	

8.3 Operation

8.3.1 Overview

When activated, the DTC reads register information that is already stored in memory and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation. A setting can also be made to have chain transfer performed only when the transfer counter value is 0. This enables DTC re-setting to be performed by the DTC itself.

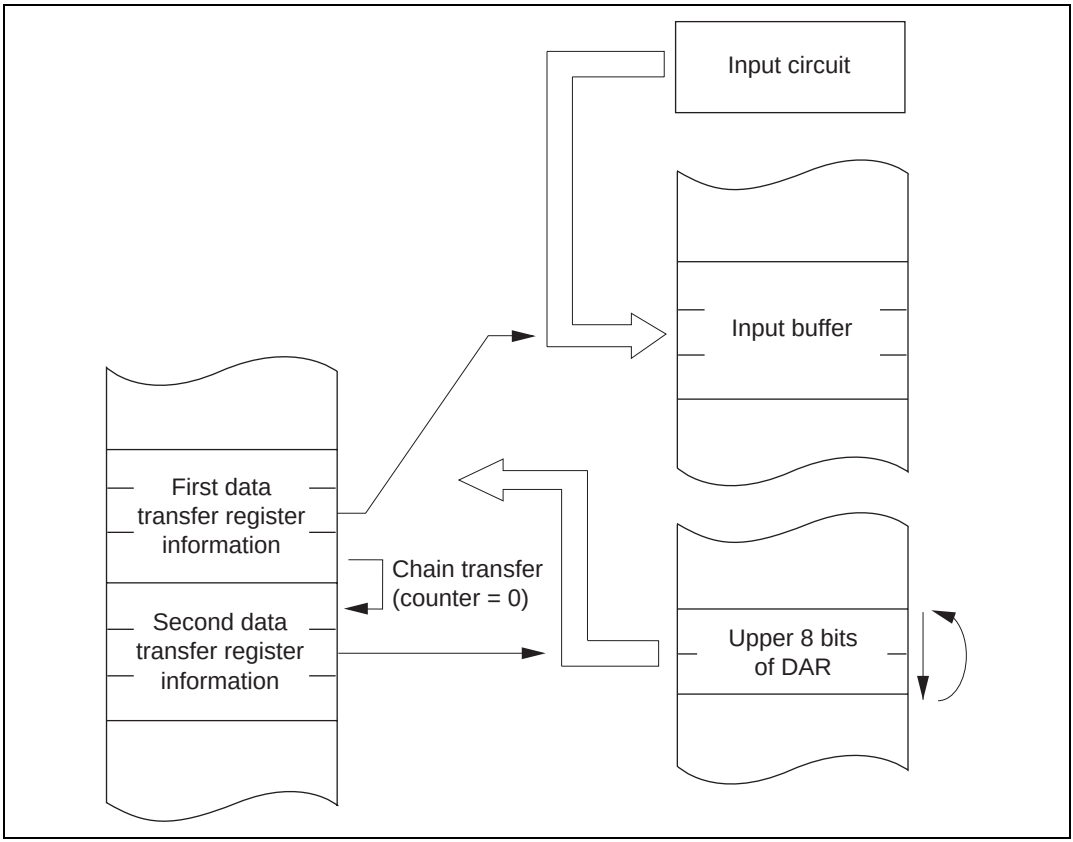

Figure 8.13 Chain Transfer when Counter = 0

Software Activation: An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

[1] Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

[2] Set the start address of the register information at the DTC vector address (H'04C0).

[3] Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.

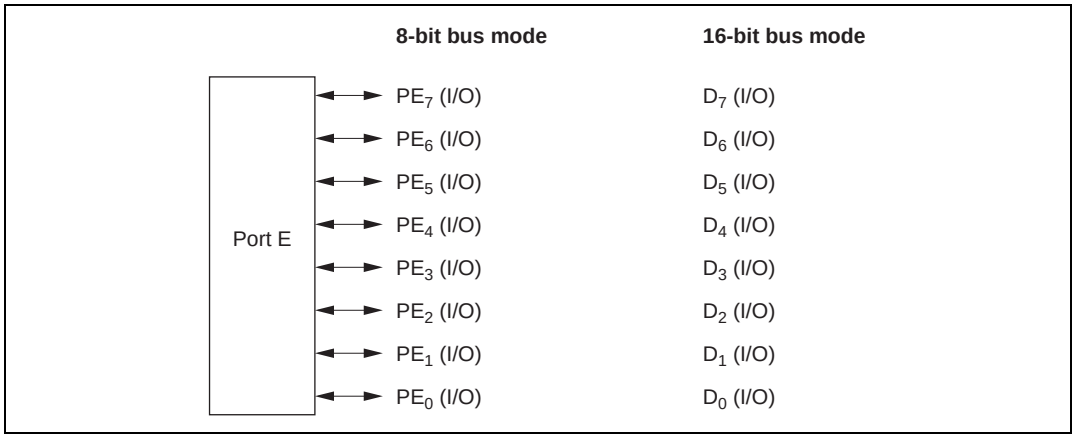

Figure 9.20 Port E Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port E pins function as I/O ports. Input or output can be specified for each pin on a bit-by-bit basis. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Port E pin functions in mode 7 are shown in figure 9.21.

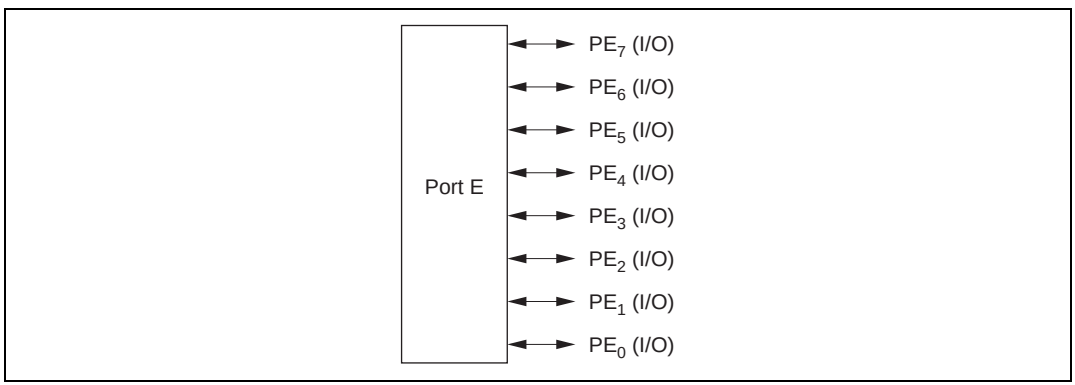

Figure 9.21 Port E Pin Functions (Mode 7)

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
1	0	0	0	0	TGR1B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0C compare match/input capture

*: Don't care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don't care

TCFV Flag/TCFU Flag Setting Timing: Figure 10.44 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing.

Figure 10.45 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

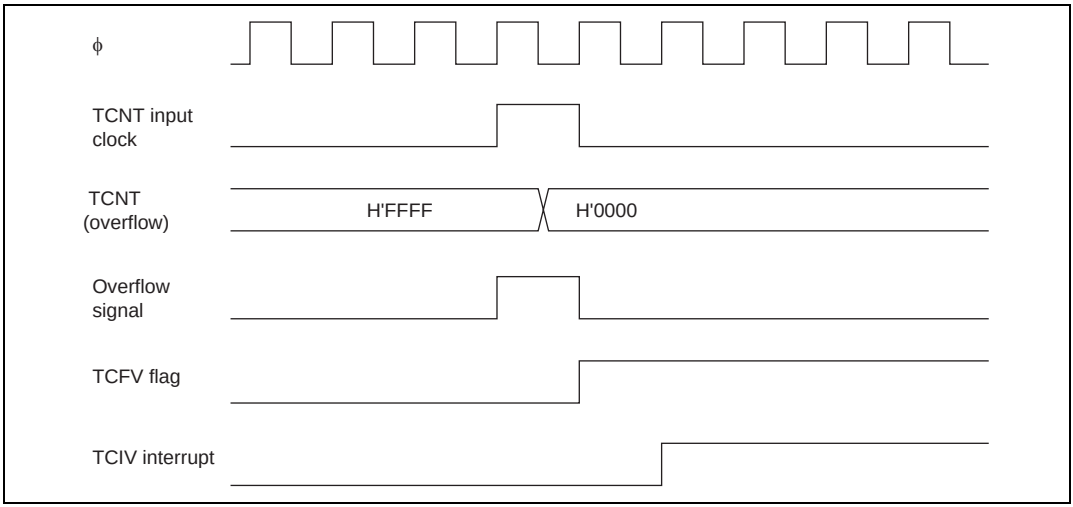

Figure 10.44 TCIV Interrupt Setting Timing

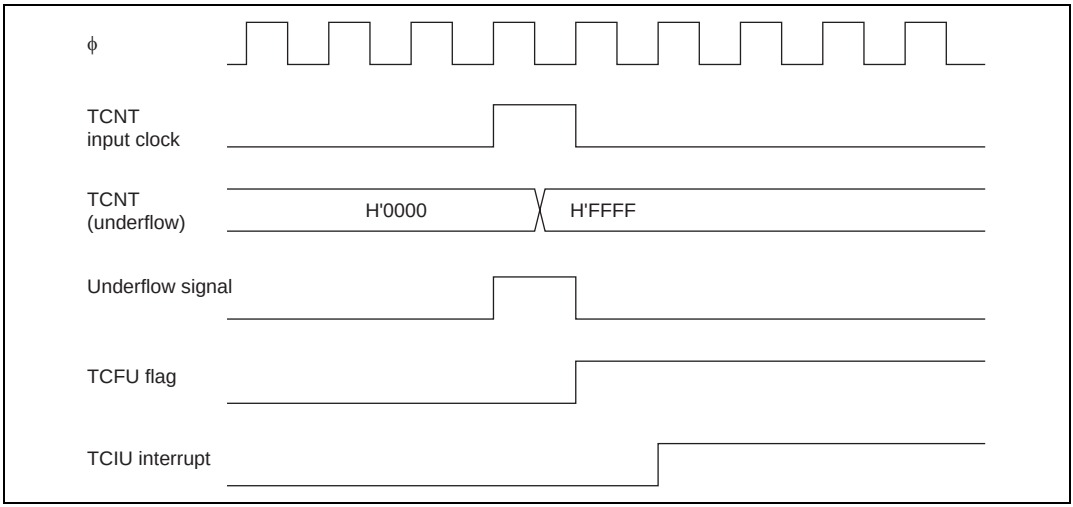

Figure 10.45 TCIU Interrupt Setting Timing

Serial Data Reception (Except Block Transfer Mode): Data reception in smart card mode uses the same processing procedure as for the normal SCI. Figure 15.7 shows an example of the transmission processing flow.

[1] Perform smart card interface mode initialization as described above in Initialization.

[2] Check that the ORER flag and PER flag in SSR are cleared to 0. If either is set, perform the appropriate receive error handling, then clear both the ORER and the PER flag to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the RDRF flag is set to 1.

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2].

[6] To end reception, clear the RE bit to 0.

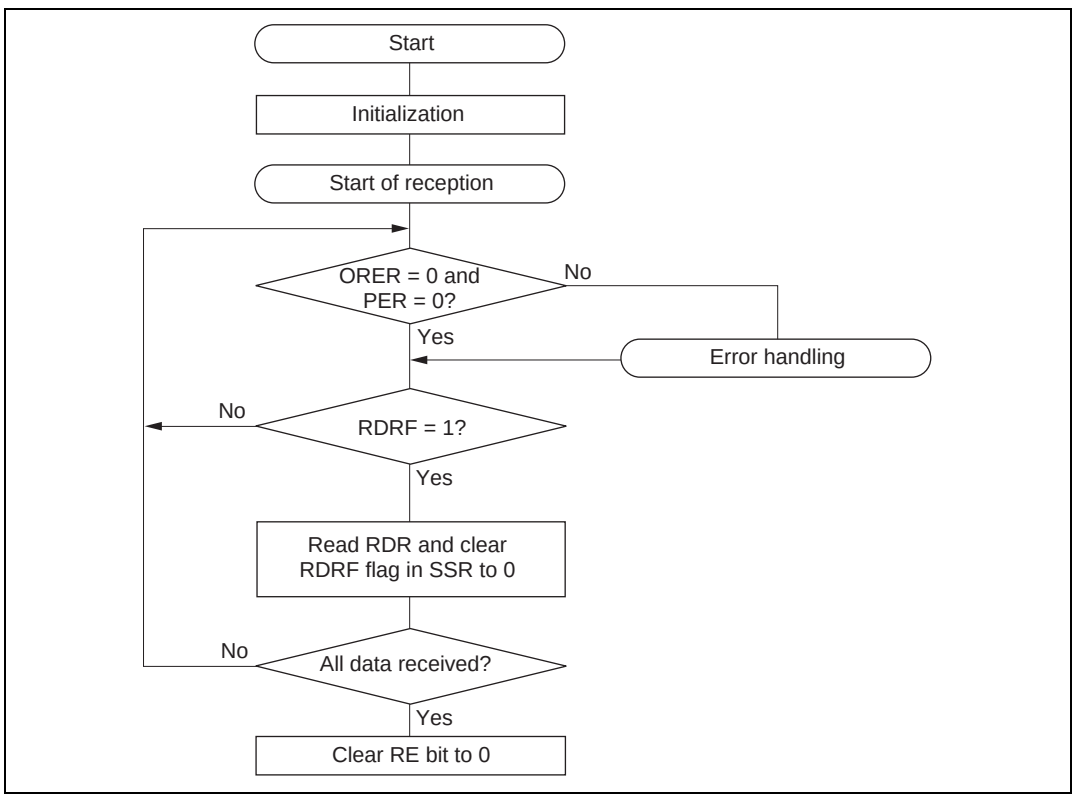

Figure 15.7 Sample Reception Flowchart

Powering On: To secure the clock duty from power-on, the following switching procedure should be followed.

[1] The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.

[2] Fix the SCK pin to the specified output level with the CKE1 bit in SCR.

[3] Set SMR and SCMR, and switch to smart card mode operation.

[4] Set the CKE0 bit in SCR to 1 to start clock output.

15.3.8 Operation in Block Transfer Mode

Operation in block transfer mode is the same as in SCI asynchronous mode, except for the following points. For details, see section 14.3.2, Operation in Asynchronous Mode.

Data Format: The data format is 8 bits with parity. There is no stop bit, but there is a guard time of 2 or more bits (1 or more bits in reception).

Also, except during transmission (with start bit, data bits, and parity bit), the transmission pins go to the high-impedance state, so the signal lines must be fixed high with a pull-up resistor.

Transmit/Receive Clock: Only an internal clock generated by the built-in baud rate generator can be used as the transmit/receive clock. The number of basic clock periods in a 1-bit transfer interval can be set to 32, 64, 372, or 256 with bits BCP1 and BCP0. For details, see section 15.3.5, Clock.

ERS (FER) Flag: As with the normal smart card interface, the ERS flag indicates the error signal status, but since error signal transmission and reception is not performed, this flag is always cleared to 0.

Table 19.12 Software Protection

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Clearing the SWE bit to 0 in FLMCR1 sets the program/erase-protected state for all blocks. • (Execute in on-chip RAM or external memory.)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block registers 1 and 2 (EBR1, EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

19.8.3 Error Protection

In error protection, an error is detected when MCU runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the MCU malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

- When flash memory is read during programming/erasing (including a vector read or instruction fetch)
- Immediately after exception handling (excluding a reset) during programming/erasing
- When a SLEEP instruction (including software standby) is executed during programming/erasing
- When a bus master other than the CPU (the DMAC or DTC) has control of the bus during programming/erasing

19.27.2 RAM Overlap

An example in which flash memory block area EB1 is overlapped is shown below.

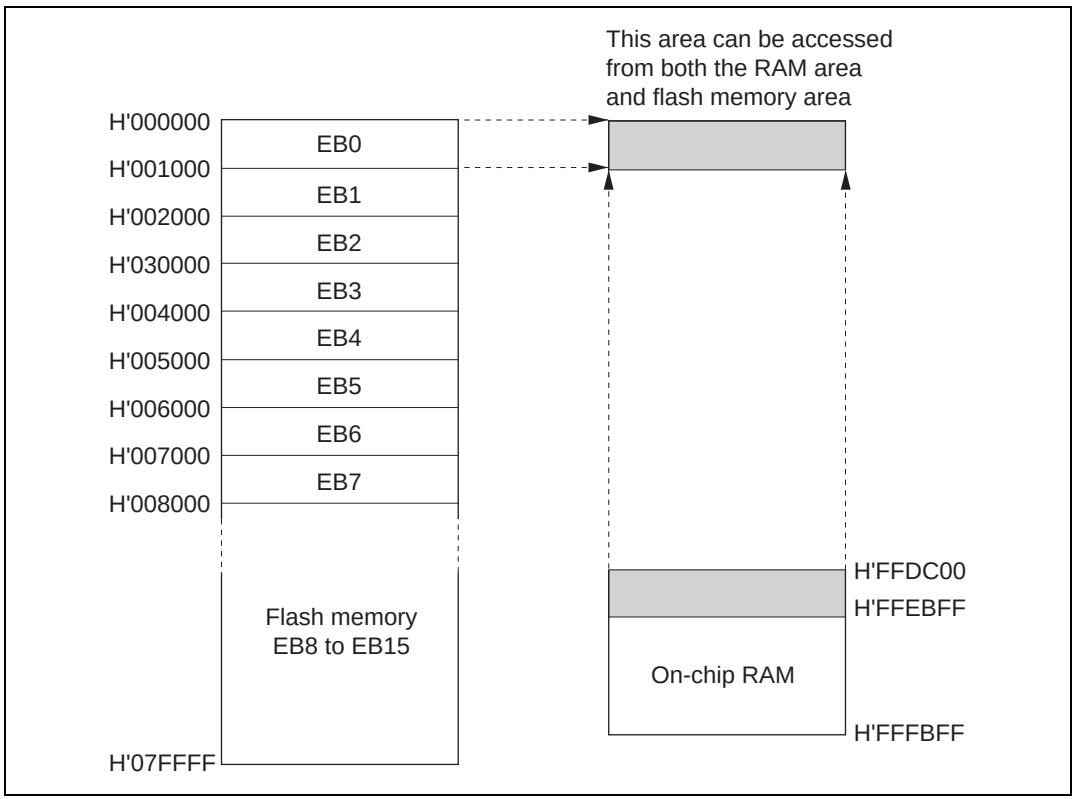

Figure 19.75 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB1 is Overlapped

1. Set bits RAMS, RAM2, RAM1, and RAM0 in RAMER to 1, 0, 0, 1, to overlap part of RAM onto the area (EB1) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB1).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2, RAM1, and RAM0 (emulation protection). In this state, setting the P1 or E1 bit in flash memory control register 1 (FLMCR1), or setting

(3) Bus Timing

Table 22.17 Bus Timing

Condition B: V_{CC} = 3.0 V to 3.6 V, AV_{CC} = 3.0 V to 3.6 V, V_{ref} = 3.0 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 25 MHz, T_a = –20°C to 75°C (regular specifications), T_a = –40°C to 85°C (wide-range specifications)

		Condition B			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Address delay time	t_{AD}	—	20	ns	Figures 22.6 to 22.13
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 15$	—	ns	
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 8$	—	ns	
Precharge time	t_{PCH}	$1.5 \times t_{cyc} - 15$	—	ns	
$\overline{CS}$ delay time 1	t_{CSD1}	—	15	ns	
$\overline{CS}$ delay time 2	t_{CSD2}	—	15	ns	
$\overline{CS}$ delay time 3	t_{CSD3}	—	20	ns	
$\overline{AS}$ delay time	t_{ASD}	—	15	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	15	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	15	ns	
$\overline{CAS}$ delay time	t_{CASD}	—	15	ns	
Read data setup time	t_{RDS}	15	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time 1	t_{ACC1}	—	$1.0 \times t_{cyc} - 20$	ns	
Read data access time 2	t_{ACC2}	—	$1.5 \times t_{cyc} - 20$	ns	
Read data access time 3	t_{ACC3}	—	$2.0 \times t_{cyc} - 20$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 20$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 20$	ns	
Read data access time 6	t_{ACC6}	—	$1.0 \times t_{cyc} - 20$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	15	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	15	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 15$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 15$	—	ns	
Write data delay time	t_{WDD}	—	20	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 15$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 8$	—	ns	
$\overline{WR}$ setup time	t_{WCS}	$0.5 \times t_{cyc} - 10$	—	ns	
$\overline{WR}$ hold time	t_{WCH}	$0.5 \times t_{cyc} - 10$	—	ns	

TIOR3H—Timer I/O Control Register 3H H'FE82 TPU3

Bit	7	6	5	4	3	2	1	0
Initial value	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3A I/O Control

0	0	0	0	TGR3A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3A is input capture register	Capture input source is $TIOCA_3$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down

* : Don't care

TGR3B I/O Control

0	0	0	0	TGR3B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3B is input capture register	Capture input source is $TIOCB_3$ pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

* : Don't care

Note: 1. When bits TPSC2 to TPSC0 in TCR4 are set to B'000, and φ/1 is used as the TCNT4 count clock, this setting is invalid and input capture does not occur.

ISCRH — IRQ Sense Control Register H **H'FF2C** **Interrupt Controller**
ISCRL — IRQ Sense Control Register L **H'FF2D** **Interrupt Controller**

ISCRH

Bit	:	15	14	13	12	11	10	9	8
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ7 to IRQ4 Sense Control

ISCRL

Bit	:	7	6	5	4	3	2	1	0
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	:	0	0	0	0	0	0	0	0
Read/Write	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ3 to IRQ0 Sense Control

IRQnSCB	IRQnSCA	Interrupt Request Generation
0	0	$\overline{IRQ}_n$ input low level
	1	Falling edge of $\overline{IRQ}_n$ input
1	0	Rising edge of $\overline{IRQ}_n$ input
	1	Both falling and rising edges of $\overline{IRQ}_n$ input

(n = 7 to 0)

RDR1—Receive Data Register 1 **H'FF85 SCI1, Smart Card Interface 1**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Stores received serial data

SCMR1—Smart Card Mode Register 1 **H'FF86 SCI1, Smart Card Interface 1**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Smart card interface function is disabled
1	Smart card interface function is enabled

Smart Card Data Invert

0	TDR contents are transmitted as they are Receive data is stored in RDR as it is
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Data Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first