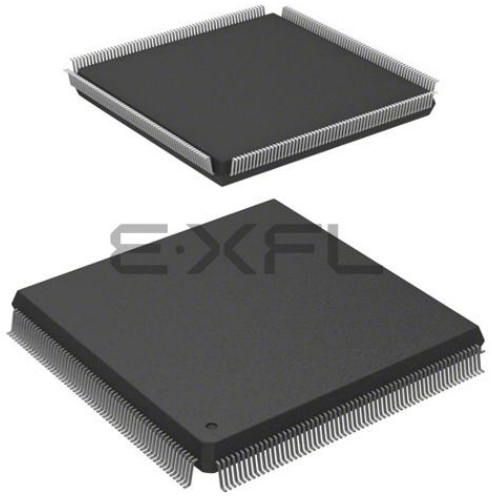


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

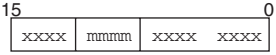
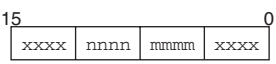
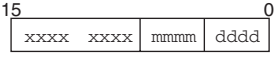
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Last Time Buy
Core Processor	SH2A-FPU
Core Size	32-Bit Single-Core
Speed	200MHz
Connectivity	CANbus, I²C, SCI, Memory Card, SSU, USB
Peripherals	DMA, LCD, POR, WDT
Number of I/O	82
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	80K x 8
Voltage - Supply (Vcc/Vdd)	1.1V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	240-BFQFP
Supplier Device Package	240-QFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5s72631p200fp

Pin	Function 1		Function 2		Function 3	
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O
13	PC4	I/O	$\overline{WE0}$/DQMLL	O	—	—
14	PC3	I/O	$\overline{CS3}$	O	—	—
15	PC2	I/O	$\overline{CS2}$	O	—	—
16	Vcc					
17	PC0	I/O	A0	O	$\overline{CS7}$	O
18	Vss					
19	PVss					
20	PC1	I/O	A1	O	—	—
21	PVcc					
22	A2	O	—	—	—	—
23	A3	O	—	—	—	—
24	A4	O	—	—	—	—
25	A5	O	—	—	—	—
26	A6	O	—	—	—	—

Pin	Function 4		Function 5		Function 6		Weak	Pull-	Simplified
No.	Symbol	I/O	Symbol	I/O	Symbol	I/O	keeper	up	circuit Diagram
13	—	—	—	—	—	—	Yes		Figure 1.3 (9)
14	—	—	—	—	—	—	Yes		Figure 1.3 (9)
15	—	—	—	—	—	—	Yes		Figure 1.3 (9)
16									
17	$\overline{AUDSYNC}$	O	—	—	—	—	Yes		Figure 1.3 (9)
18									
19									
20	—	—	—	—	—	—	Yes		Figure 1.3 (9)
21									
22	—	—	—	—	—	—	Yes		Figure 1.3 (7)
23	—	—	—	—	—	—	Yes		Figure 1.3 (7)
24	—	—	—	—	—	—	Yes		Figure 1.3 (7)
25	—	—	—	—	—	—	Yes		Figure 1.3 (7)
26	—	—	—	—	—	—	Yes		Figure 1.3 (7)

Instruction Formats	Source Operand	Destination Operand	Example
m format 	mmmm: Register direct	Control register or system register	LDC Rm,SR
	mmmm: Register indirect with post-increment	Control register or system register	LDC.L @Rm+,SR
	mmmm: Register indirect	—	JMP @Rm
	mmmm: Register indirect with pre-decrement	R0 (Register direct)	MOV.L @-Rm,R0
	mmmm: PC relative using Rm	—	BRAF Rm
nm format 	mmmm: Register direct	nnnn: Register direct	ADD Rm,Rn
	mmmm: Register direct	nnnn: Register indirect	MOV.L Rm,@Rn
	mmmm: Register indirect with post-increment (multiply-and-accumulate) nnnn*: Register indirect with post-increment (multiply-and-accumulate)	MACH, MACL	MAC.W @Rm+,@Rn+
	mmmm: Register indirect with post-increment	nnnn: Register direct	MOV.L @Rm+,Rn
	mmmm: Register direct	nnnn: Register indirect with pre-decrement	MOV.L Rm,@-Rn
	mmmm: Register direct	nnnn: Indexed register indirect	MOV.L Rm,@(R0,Rn)
md format 	mmmmdddd: Register indirect with displacement	R0 (Register direct)	MOV.B @(disp,Rm),R0

5.6.6 FPU Exceptions

FPU exception handling takes place when the V, Z, O, U, or I bit in the FPU enable field (Enable) of the floating point status/control register (FPSCR) is set to 1. This indicates the occurrence of an invalid operation exception defined by the IEEE 754 standard, a division-by-zero exception, an overflow (in the case of an instruction for which this is possible), an underflow (in the case of an instruction for which this is possible), or an inexact exception (in the case of an instruction for which this is possible).

The instructions that may trigger FPU exception handling are FADD, FSUB, FMUL, FDIV, FMAC, FCMP/EQ, FCMP/GT, FLOAT, FTRC, FCNVDS, FCNVSD, and FSQRT.

FPU exception handling occurs only when the corresponding FPU exception enable bit (Enable) is set to 1. When an exception source triggered by a floating-point operation is detected, FPU operation is halted and the occurrence of FPU exception handling is reported to the CPU. When exception handling starts, the CPU operates as follows:

1. The start address of the exception service routine which corresponds to the FPU exception handling that occurred is fetched from the exception handling vector table.
2. The status register (SR) is saved on the stack.
3. The program counter (PC) is saved on the stack. The PC value saved is the start address of the instruction to be executed after the last executed instruction.
4. After jumping to the address fetched from the exception handling vector table, program execution starts. This jump is not a delayed branch.

The FPU exception flag field (Flag) of FPSCR is always updated regardless of whether or not FPU exception handling has been accepted, and remains set until explicitly cleared by the user through an instruction. The FPU exception source field (Cause) of FPSCR changes each time a floating-point instruction is executed.

When the V bit in the FPU exception enable field (Enable) of FPSCR and the QIS bit in FPSCR are both set to 1, FPU exception handling occurs when qNAN or ±∞ is input to a floating-point operation instruction source.

- CS2WCR, CS3WCR

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	BAS	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	WR[3:0]				WM	-	-	-	-	-	-
Initial value:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	BAS	0	R/W	SRAM with Byte Selection Byte Access Select Specifies the $\overline{WEn}$ and RD/$\overline{WR}$ signal timing when the SRAM interface with byte selection is used. 0: Asserts the $\overline{WEn}$ signal at the read timing and asserts the RD/$\overline{WR}$ signal during the write access cycle. 1: Asserts the $\overline{WEn}$ signal during the read access cycle and asserts the RD/$\overline{WR}$ signal at the write timing.
19 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

10.4.3 Channel Priority

When the DMAC receives simultaneous transfer requests on two or more channels, it selects a channel according to a predetermined priority order. Three modes (fixed mode 1, fixed mode 2, and round-robin mode) are selected using the PR1 and PR0 bits in DMAOR.

(1) Fixed Mode

In fixed modes, the priority levels among the channels remain fixed. There are two kinds of fixed modes as follows:

Fixed mode 1: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7

Fixed mode 2: CH0 > CH4 > CH1 > CH5 > CH2 > CH6 > CH3 > CH7

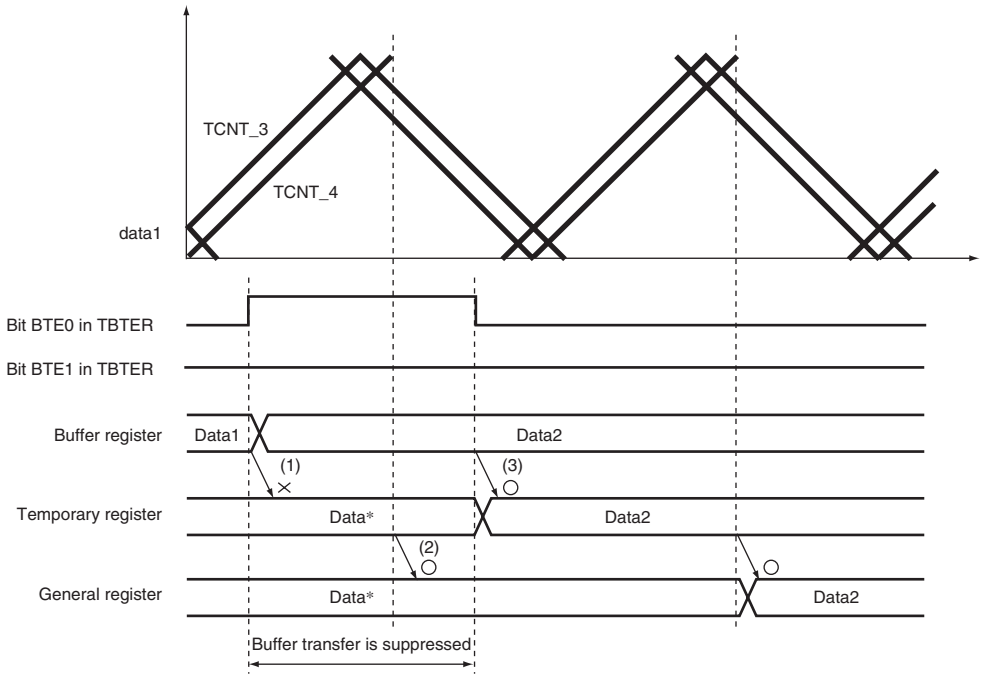
These are selected by the PR1 and PR0 bits in the DMA operation register (DMAOR).

(2) Round-Robin Mode

Each time one unit of word, byte, longword, or 16 bytes is transferred on one channel, the priority order is rotated. The channel on which the transfer was just finished is rotated to the lowest of the priority order among the four round-robin channels (channels 0 to 4). The priority of the channels other than the round-robin channels (channels 0 to 4) does not change even in round-robin mode. The round-robin mode operation is shown in figure 10.3. The priority in round-robin mode is CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 immediately after a reset.

When the round-robin mode has been specified, do not concurrently specify cycle steal mode and burst mode as the bus modes of any two or more channels.

Bit	Bit Name	Initial Value	R/W	Description
4	TCIEV	0	R/W	Overflow Interrupt Enable Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1. 0: Interrupt requests (TCIV) by TCFV disabled 1: Interrupt requests (TCIV) by TCFV enabled
3	TGIED	0	R/W	TGR Interrupt Enable D Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 3 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGID) by TGFD bit disabled 1: Interrupt requests (TGID) by TGFD bit enabled
2	TGIEC	0	R/W	TGR Interrupt Enable C Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0, 3, and 4. In channels 1 and 2, bit 2 is reserved. It is always read as 0 and the write value should always be 0. 0: Interrupt requests (TGIC) by TGFC bit disabled 1: Interrupt requests (TGIC) by TGFC bit enabled
1	TGIEB	0	R/W	TGR Interrupt Enable B Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1. 0: Interrupt requests (TGIB) by TGFB bit disabled 1: Interrupt requests (TGIB) by TGFB bit enabled
0	TGIEA	0	R/W	TGR Interrupt Enable A Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1. 0: Interrupt requests (TGIA) by TGFA bit disabled 1: Interrupt requests (TGIA) by TGFA bit enabled



[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE1 and BTE0 in TBTER are set to 0 and 1, respectively).
(2) Data is transferred from the temporary register to the general register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note: * When buffer transfer at the crest is selected.

Figure 11.70 Example of Operation when Buffer Transfer is Suppressed (BTE1 = 0 and BTE0 = 1)

11.8.3 Operation in Case of Re-Setting Due to Error During Operation, etc.

If an error occurs during MTU2 operation, MTU2 output should be cut by the system. Cutoff is performed by switching the pin output to port output with the PFC and outputting the inverse of the active level. The pin initialization procedures for re-setting due to an error during operation, etc., and the procedures for restarting in a different mode after re-setting, are shown below.

The MTU2 has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Possible mode transition combinations are shown in table 11.57.

Table 11.57 Mode Transition Combinations

	After					
Before	**Normal**	**PWM1**	**PWM2**	**PCM**	**CPWM**	**RPWM**
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	None	None
PCM	(17)	(18)	(19)	(20)	None	None
CPWM	(21)	(22)	None	None	(23) (24)	(25)
RPWM	(26)	(27)	None	None	(28)	(29)

[Legend]
Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 4
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

(15) Operation when Error Occurs during PWM Mode 2 Operation, and Operation is Restarted in PWM Mode 2

Figure 11.129 shows an explanatory diagram of the case where an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

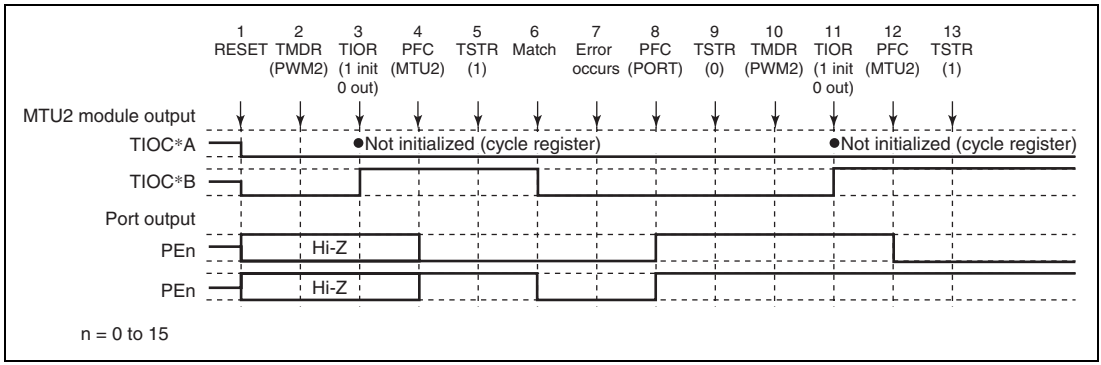


Figure 11.129 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

1 to 9 are the same as in figure 11.127.

10. Not necessary when restarting in PWM mode 2.
11. Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized.)
12. Set MTU2 output with the PFC.
13. Operation is restarted by TSTR.

14.3.9 Second Alarm Register (RSECAR)

RSECAR is an alarm register corresponding to the BCD coded second counter RSECCNT of the RTC. When the ENB bit is set to 1, a comparison with the RSECCNT value is performed. From among RSECAR/RMINAR/RHRAR/RWKAR/RDAYAR/RMONAR/RCR3, the counter and alarm register comparison is performed only on those with ENB bits set to 1, and if each of those coincides, an alarm flag of RCR1 is set to 1.

The assignable range is from 00 through 59 + ENB bits (practically in BCD), otherwise operation errors occur.

Bit:	7	6	5	4	3	2	1	0
	ENB	10 seconds			1 second			
Initial value:	0	-	-	-	-	-	-	-
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
7	ENB	0	R/W	When this bit is set to 1, a comparison with the RSECCNT value is performed.
6 to 4	10 seconds	Undefined	R/W	Ten's position of seconds setting value
3 to 0	1 second	Undefined	R/W	One's position of seconds setting value

15.3.6 Serial Control Register (SCSCR)

SCSCR operates the SCIF transmitter/receiver, enables/disables interrupt requests, and selects the transmit/receive clock source. The CPU can always read and write to SCSCR.

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TIE	RIE	TE	RE	REIE	-	CKE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	TIE	0	R/W	Transmit Interrupt Enable Enables or disables the transmit-FIFO-data-empty interrupt (TXI) requested when the serial transmit data is transferred from the transmit FIFO data register (SCFTDR) to the transmit shift register (SCTSR), when the quantity of data in the transmit FIFO register becomes less than the specified number of transmission triggers, and when the TDFE flag in the serial status register (SCFSR) is set to1. 0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled 1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled* Note: * The TXI interrupt request can be cleared by writing a greater quantity of transmit data than the specified transmission trigger number to SCFTDR and by clearing TDFE to 0 after reading 1 from TDFE, or can be cleared by clearing TIE to 0.

Figure 15.12 shows a sample flowchart for initializing the SCIF.

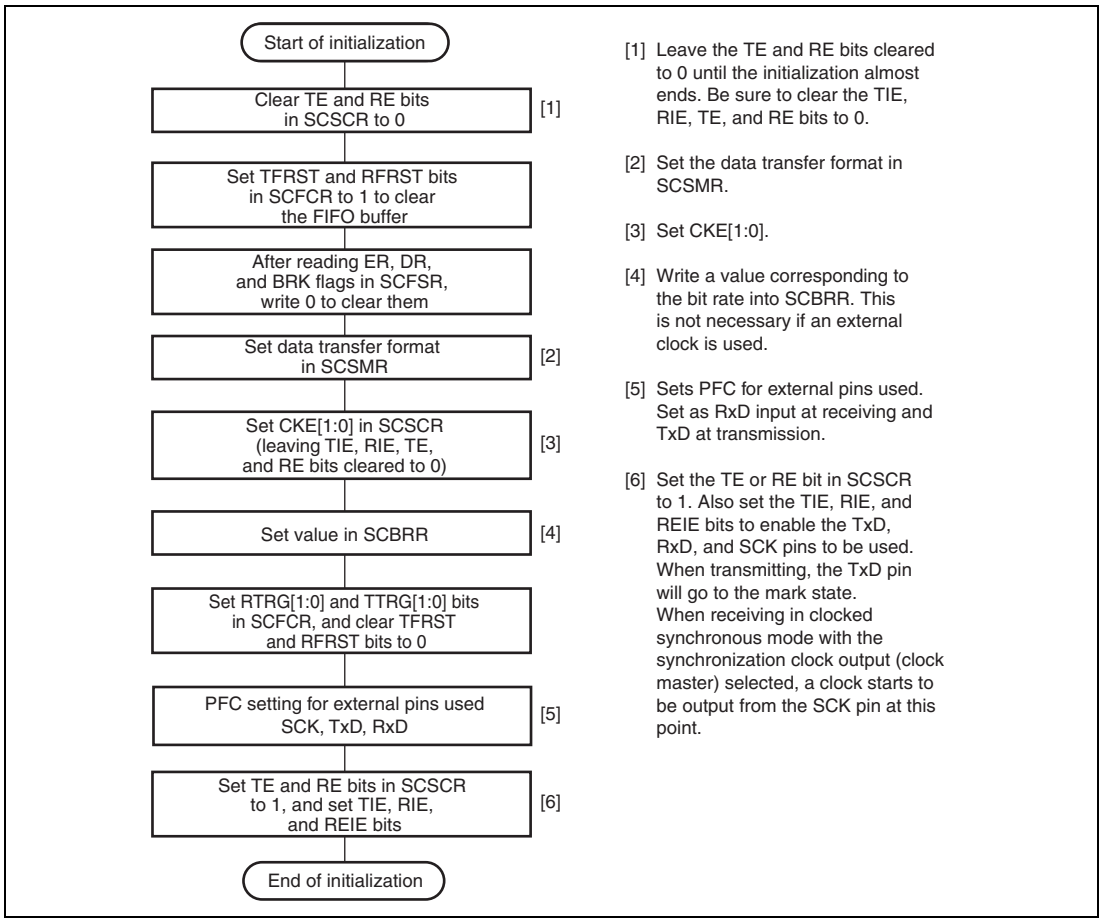


Figure 15.12 Sample Flowchart for SCIF Initialization

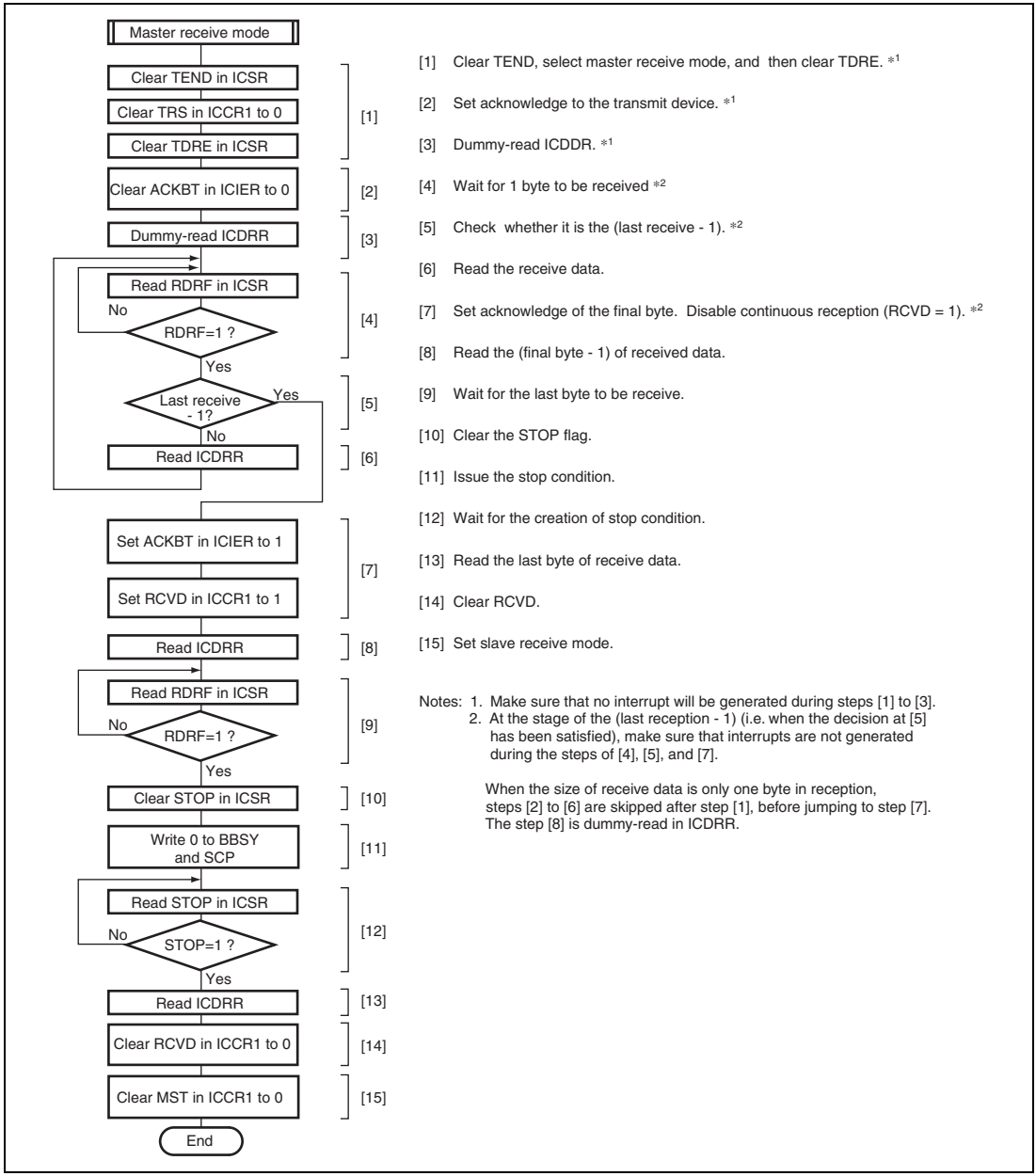


Figure 17.19 Sample Flowchart for Master Receive Mode

18.4.7 Serial Bit Clock Control

This function is used to control and select which clock is used for the serial bus interface.

If the serial clock direction is set to input (SCKD = 0), the SSI module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.

If the serial clock direction is set to output (SCKD = 1), the SSI module is in clock master mode, and the shift register uses the oversampling clock, or the bit clock that is generated by dividing it. The oversampling clock is then divided by the ratio in the serial oversampling clock divide ratio bit (CKDV) in SSICR and used as the bit clock in the shift register.

In either case the module pin, SSISCK, is the same as the bit clock.

Table 22.5 A/D Conversion Time (Multi Mode and Scan Mode)

CKS1	CKS0	Conversion Time (tpcyc)
0	0	128 (constant)
	1	256 (constant)
1	0	512 (constant)

Note: Values in the table are the numbers of t_{pcyc}. t_{pcyc} indicates the peripheral clock (Pϕ) cycle.

22.4.6 External Trigger Input Timing

A/D conversion can also be externally triggered. When the TRGS[3:0] bits in ADCSR are set to B'1001, an external trigger is input to the $\overline{\text{ADTRG}}$ pin. The ADST bit in ADCSR is set to 1 at the falling edge of the $\overline{\text{ADTRG}}$ pin, thus starting A/D conversion. Other operations, regardless of the operating mode, are the same as when the ADST bit has been set to 1 by software. Figure 22.6 shows the timing.

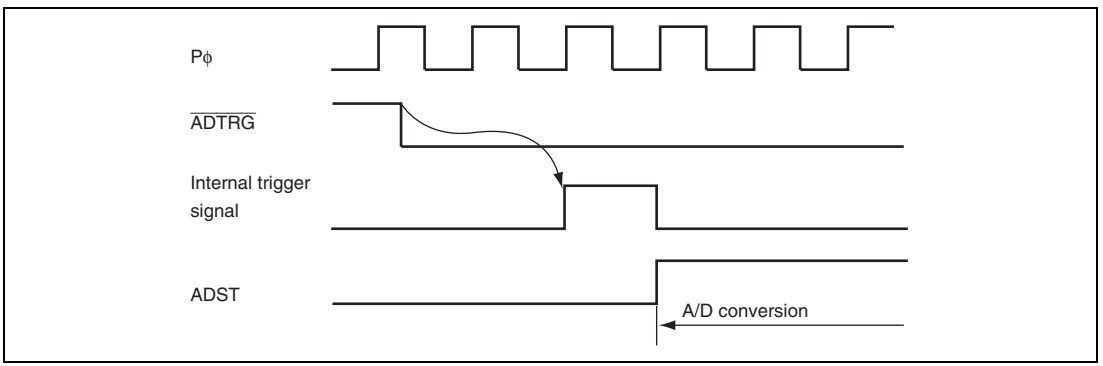


Figure 22.6 External Trigger Input Timing

Section 25 USB 2.0 Host/Function Module (USB)

The USB 2.0 host/function module (USB) provides capabilities as a USB host and USB function and supports high-speed and full-speed transfers defined by USB specification 2.0. This module has a USB transceiver* and supports all of the transfer types defined by the USB specification.

This module has an 8-kbyte buffer memory for data transfer, providing a maximum of eight pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE7, based on the peripheral devices or user system for communication.

Note: * The internal USB transceiver must be set before this module is used. For details, see section 25.5.2, Procedure for Setting the USB Transceiver.

25.1 Features

(1) Host Controller and Function Controller Supporting USB High-Speed Operation

- The USB host controller and USB function controller are incorporated.
- The USB host controller and USB function controller can be switched by register settings.
- Both high-speed transfer (480 Mbps) and full-speed transfer (12 Mbps) are supported.
- High-speed/full-speed USB transceiver (shared by the USB host and USB function) is incorporated.

(2) Reduced Number of External Pins and Space-Saving Installation

- On-chip D+ pull-up resistor (during USB function operation)
- On-chip D+ and D- pull-down resistor (during USB host operation)
- On-chip D+ and D- terminal resistor (during high-speed operation)
- On-chip D+ and D- output resistor (during full-speed operation)

(3) All Types of USB Transfers Supported

- Control transfer
- Bulk transfer
- Interrupt transfer (high bandwidth transfers not supported)
- Isochronous transfer (high bandwidth transfers not supported)

(4) Internal Bus Interfaces

- Two DMA interface channels are incorporated.

26.6 Usage Notes

26.6.1 Procedure for Halting Access to Display Data Storage VRAM (Synchronous DRAM in Area 3)

Follow the procedure below to halt access to VRAM for storing display data (synchronous DRAM in area 3).

- Procedure for Halting Access to Display Data Storage VRAM:

1. Confirm that the LPS1 and LPS0 bits in LDPMMR are currently set to 1.
2. Clear the DON bit in LDCNTR to 0 (display-off mode).
3. Confirm that the LPS1 and LPS0 bits in LDPMMR have changed to 0.
4. Wait for the display time for a single frame to elapse.

This halting procedure is required before selecting self-refreshing for the display data storage VRAM (synchronous DRAM in area 3) or making a transition to standby mode or module standby mode.

Bit	Bit Name	Initial Value	R/W	Description
10 to 8	PD10MD[2:0]	000/001*	R/W	PD10 Mode Select the function of the PD10/D26/PINT2/SD_CMD/TEND0/TIOC3C pin. • Area 0: 32-bit mode 000: Setting prohibited 001: D26 I/O (data) (initial value) 010: Setting prohibited 011: Setting prohibited 100: Setting prohibited 101: Setting prohibited 110: Setting prohibited 111: Setting prohibited • Area 0: 16-bit mode 000: PD10 I/O (port) (initial value) 001: D26 I/O (data) 010: PINT2 input (INTC) 011: SD_CMD I/O (SDHI) 100: TEND0 output (DMAC) 101: TIOC3C I/O (MTU2) 110: Setting prohibited 111: Setting prohibited
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

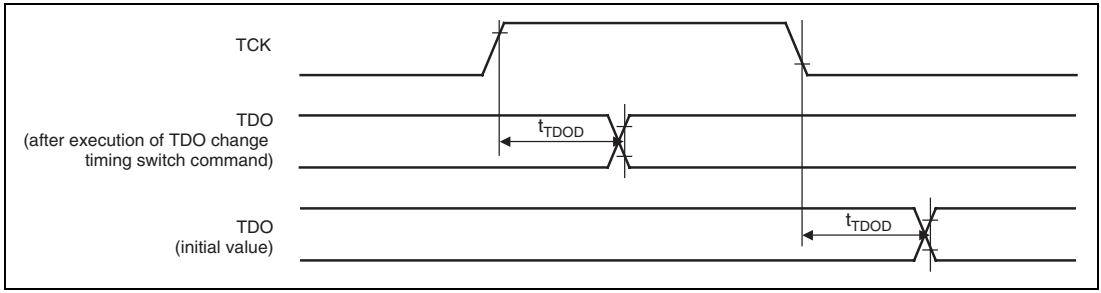


Figure 33.3 H-UDI Data Transfer Timing

33.4.4 H-UDI Reset

An H-UDI reset is executed by setting an H-UDI reset assert command in SDIR. An H-UDI reset is of the same kind as a power-on reset. An H-UDI reset is released by setting an H-UDI reset negate command. The required time between the H-UDI reset assert command and H-UDI reset negate command is the same as time for keeping the $\overline{\text{RES}}$ pin low to apply a power-on reset.

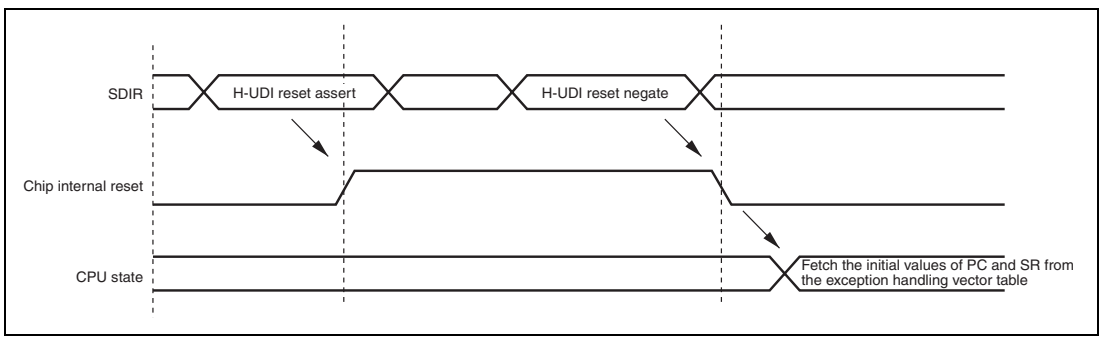


Figure 33.4 H-UDI Reset

33.4.5 H-UDI Interrupt

The H-UDI interrupt function generates an interrupt by setting a command from the H-UDI in SDIR. An H-UDI interrupt is a general exception/interrupt operation, resulting in fetching the exception service routine start address from the exception handling vector table, jumping to that address, and starting program execution from that address. This interrupt request has a fixed priority level of 15.

H-UDI interrupts are accepted in sleep mode, but not in software standby mode.

Module Name	Register Name	Abbreviation	Number of Bits	Address	Access Size
PFC	Port B control register L1	PBCRL1	16	H'FFFE3896	8, 16
	IRQOUT function control register	IFCR	16	H'FFFE38A2	8, 16
	Port C I/O register L	PCIORL	16	H'FFFE3906	8, 16
	Port C control register L4	PCCRL4	16	H'FFFE3910	8, 16, 32
	Port C control register L3	PCCRL3	16	H'FFFE3912	8, 16
	Port C control register L2	PCCRL2	16	H'FFFE3914	8, 16, 32
	Port C control register L1	PCCRL1	16	H'FFFE3916	8, 16
	Port D I/O register L	PDIORL	16	H'FFFE3986	8, 16
	Port D control register L4	PDCRL4	16	H'FFFE3990	8, 16, 32
	Port D control register L3	PDCRL3	16	H'FFFE3992	8, 16
	Port D control register L2	PDCRL2	16	H'FFFE3994	8, 16, 32
	Port D control register L1	PDCRL1	16	H'FFFE3996	8, 16
	Port E I/O register L	PEIORL	16	H'FFFE3A06	8, 16
	Port E control register L4	PECRL4	16	H'FFFE3A10	8, 16, 32
	Port E control register L3	PECRL3	16	H'FFFE3A12	8, 16
	Port E control register L2	PECRL2	16	H'FFFE3A14	8, 16, 32
	Port E control register L1	PECRL1	16	H'FFFE3A16	8, 16
	Port F I/O register H	PFIORH	16	H'FFFE3A84	8, 16, 32
	Port F I/O register L	PFIORL	16	H'FFFE3A86	8, 16
	Port F control register H4	PFCRH4	16	H'FFFE3A88	8, 16, 32
	Port F control register H3	PFCRH3	16	H'FFFE3A8A	8, 16
	Port F control register H2	PFCRH2	16	H'FFFE3A8C	8, 16, 32
	Port F control register H1	PFCRH1	16	H'FFFE3A8E	8, 16
	Port F control register L4	PFCRL4	16	H'FFFE3A90	8, 16, 32
	Port F control register L3	PFCRL3	16	H'FFFE3A92	8, 16
	Port F control register L2	PFCRL2	16	H'FFFE3A94	8, 16, 32
	Port F control register L1	PFCRL1	16	H'FFFE3A96	8, 16
	SSI oversampling clock selection register	SCSR	16	H'FFFE3AA2	8, 16