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/2600
Core Size	16-Bit
Speed	20MHz
Connectivity	FIFO, I²C, LPC, SCI, SmartCard
Peripherals	POR, PWM, WDT
Number of I/O	112
Program Memory Size	160KB (160K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	144-TQFP
Supplier Device Package	144-TQFP (16x16)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2117rvpt20hv

Type	Symbol	Pin No. TFP-144V	BP-176V	TLP-145V	I/O	Name and Function
Interrupts	NMI	11	F4	E3	Input	Nonmaskable interrupt request input pin
	$\overline{\text{IRQ15}}$ to $\overline{\text{IRQ0}}$	17, 19 to 21, 47 to 50, 85, 84, 135 to 133, 24 to 22	G2, H2, J4, J3, N6, R6, P6, M7, J13, J12, B6, A6, C6, K4, J2, J1	F1, G4, H4, G1, L5, M6, N5, K5, H12, J11, C6, B5, A6, H2, G3, J4	Input	These pins request a maskable interrupt. To which pin an IRQ interrupt is input can be selected from the $\overline{\text{IRQn}}$ or $\overline{\text{ExIQRn}}$ pin. (n = 15 to 6)
	$\overline{\text{ExIRQ15}}$ to $\overline{\text{ExIRQ6}}$	51 to 58, 12, 10	R7, P7, M8, R8, P8, N9, R9, P9, F3, E1	L6, M7, N6, K6, K7, K8, N7, M8, F2, E2	Input	
H-UDI	$\overline{\text{ETRST}}$*[2]	27	L1	H3	Input	Interface pins for emulator
	ETMS	28	L2	K4	Input	Reset by holding the $\overline{\text{ETRST}}$ pin to low level regardless of the H-UDI activation. At this time, the $\overline{\text{ETRST}}$ pin should be held low level for 20 clocks of ETCK. Then, to activate the H-UDI, the $\overline{\text{ETRST}}$ pin should be set to high level and the pins ETCK, ETMS, and ETDI should be set appropriately. In the normal operation without activating the H-UDI, pins ETCK, ETMS, ETDI, and ETDO should be pulled up to high level. The $\overline{\text{ETRST}}$ pin is pulled up inside the chip.
	ETDO	29	L4	J1	Output	
	ETDI	30	M1	K2	Input	
	ETCK	31	M2	J3	Input	
8-bit timer (TMR_0, TMR_1, TMR_X, TMR_Y)	TMO0 TMO1 TMOX TMOY	137 3 47 48	B5 B1 N6 R6	A5 C2 L5 M6	Output	Waveform output pins with output compare function
	TMI0 TMI1 TMIX TMIY	136 2 58 57	A5 C3 P9 R9	D4 A1 M8 N7	Input	Counter event input and count reset input pins

Bit	Bit Name	Initial Value	R/W	Description
1	KINWUE	0	R/W	Keyboard Control Register Access Enable When the RELOCATE bit is cleared to 0, this bit enables or disables CPU access for the keyboard matrix interrupt registers (KMIMRA and KMIMR), pull-up MOS control register (KMPCR), and registers (TCR_X/TCR_Y, TCSR_X/TCSR_Y, TICRR/TCORA_Y, TICRF/TCORB_Y, TCNT_X/TCNT_Y, TCORC, TCORA_X, TCORB_X, TCONRI, and TCONRS) of 8-bit timers (TMR_X and TMR_Y) 0: Enables CPU access for registers of TMR_X and TMR_Y in areas from H'(FF)FFF0 to H'(FF)FFF7 and from H'(FF)FFFC to H'(FF)FFFF 1: Enables CPU access for the keyboard matrix interrupt registers and input pull-up MOS control register in areas from H'(FF)FFF0 to H'(FF)FFF7 and from H'(FF)FFFC to H'(FF)FFFF When the RELOCATE bit is set to 1, this bit is disabled. For details, see section 3.2.4, System Control Register 3 (SYSCR3) and section 27, List of Registers.
0	RAME	1	R/W	RAM Enable Enables or disables on-chip RAM. 0: On-chip RAM is disabled 1: On-chip RAM is enabled

(6) P42/SDA1/TCMCYI1

The pin function is switched as shown below according to the combination of the IIC1AS and IIC1BS bits in PTCNT1, ICE bit in ICCR of IIC_1, and the P42DDR bit. When the TCMIPE bit in TCMIER_1 of TCM_1 is set to 1, TCMCYI1 functions as an input pin.

		Setting	
		IIC_1	I/O Port
Module Name	Pin Function	SDA1_OE	P42DDR
IIC	SDA1 output	1	—
I/O port	P42 output	0	1
	P42 input (initial setting)	0	0

Note: To use this pin as SDA1, clear the IIC1AS and IIC1BS bits in PTCNT1 to 0. The output format for SDA1 is NMOS output only and direct bus drive is possible. When this pin is used as the P42 output pin, the output format is NMOS push-pull.

(7) P41/TMO0/RxD2/TCMCKI0/TCMMCI0

The pin function is switched as shown below according to the combination of the TMR and the P41DDR bit.

		Setting		
		TMR	SCI	I/O Port
Module Name	Pin Function	TMO0_OE	RE	P41DDR
TMR	TMO0 output	1	0	—
SCI	RxD2 input	0	1	—
I/O port	P41 output	0	0	1
	P41 input (initial setting)	0	0	0

Note: To use this pin as TMO0 output, clear the RE bit in SCR of the SCI2 to 0.

(5) Buffer Operation Timing

Figures 10.36 and 10.37 show the timing in buffer operation.

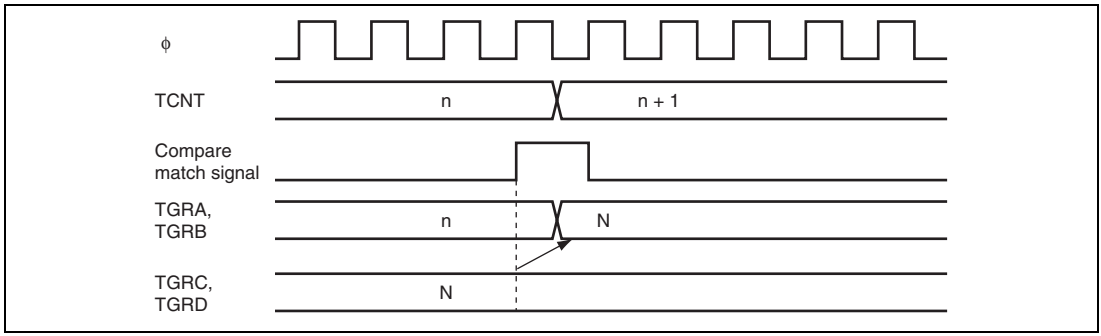

Figure 10.36 Buffer Operation Timing (Compare Match)

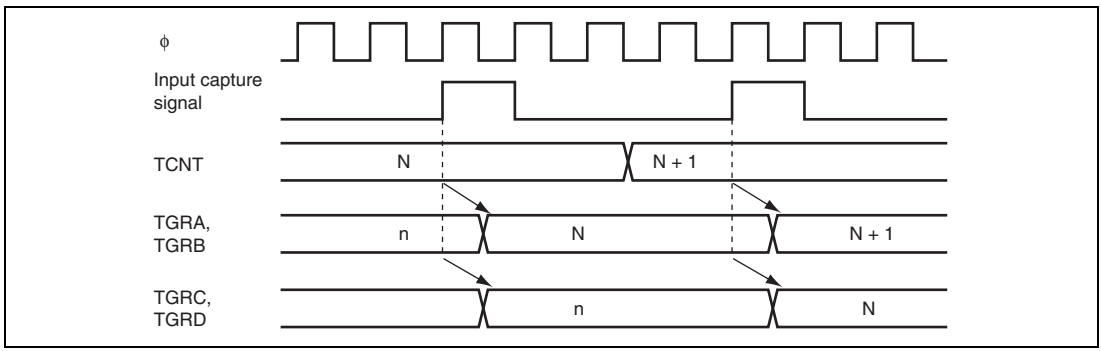

Figure 10.37 Buffer Operation Timing (Input Capture)

(2) Input Capture

The value in TCMCNT is transferred to TCMICR by detecting input edge of TCMCYI pin in timer mode. At this time, the ICPF flag in TCMCSR is set. Detection of rising or falling edges is selectable with the setting of the IEDG bit in TCMCR. Figure 11.4 shows an example of the timing of input capture operations and figure 11.5 shows buffer operation of input capture.

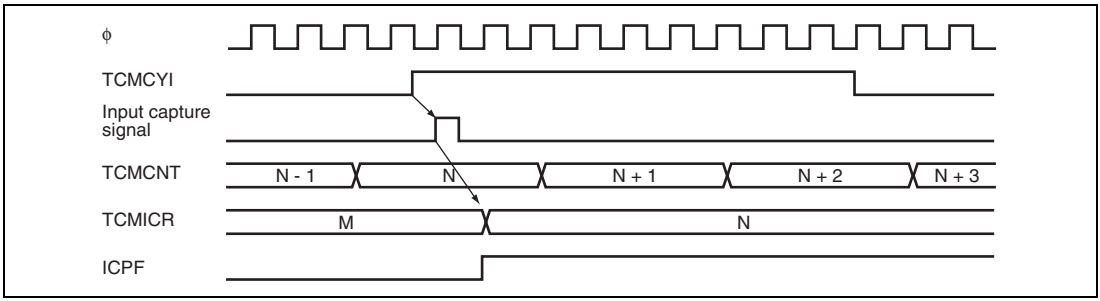

Figure 11.4 Input Capture Operation Timing (Sensing of Rising Edges)

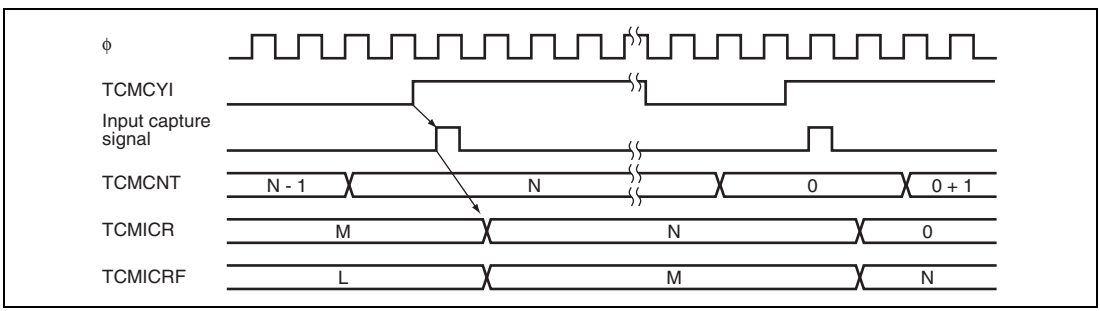

Figure 11.5 Buffer Operation of Input Capture

11.6.5 Conflict between Edge Detection in Cycle Measurement Mode and Clearing of TCMMDS Bit in TCMCR

If the CST bit in TCMCR is set to 1 in cycle measurement mode, and the TCMMDS bit in TCMCR is cleared, but the selected edge from TCMCYI is detected at the same time, detection of the selected edge will cause the timer to continue to operate in cycle measurement mode. The timer will not make the transition to timer mode until the next detection of the selected edge. Thus, ensure that the CST bit is cleared to 0 in cycle measurement mode.

Figure 11.17 shows the timing of this conflict.

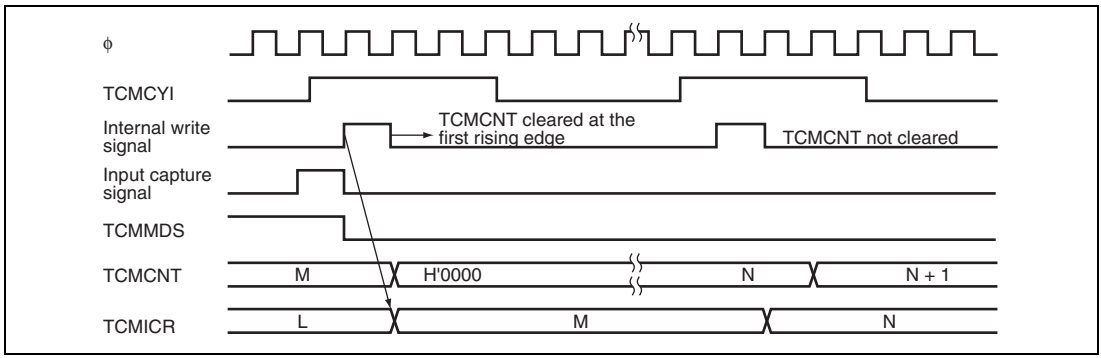

Figure 11.17 Conflict between Edge Detection and Clearing of TCMMDS (to Switch from Cycle Measurement Mode to Timer Mode)

11.6.6 Settings of TCMCKI and TCMMCI

TCMCKI and TCMMCI are multiplexed on the same pin of this LSI. Therefore, the selected external clock and the TCMMCI signal cannot be used at the same time. Do not make the settings CKS2 to CKS0 = B'111 and CMMS = B'1.

11.6.7 Setting for Module Stop Mode

The module-stop control register can be used to select either continuation or stoppage of TCM operation in module-stopped mode. The default setting is for TCM operation to stop. TCM registers become accessible on release from module stop mode. For details, see section 26, Power-Down Modes.

12.4 Operation

The TDP operates in timer mode or cycle measurement mode. After a reset, the TDP is in timer mode.

12.4.1 Timer Mode

When the TDPMDS bit in TDPCR1 is cleared to 0, the TDP operates in timer mode.

(1) Counter Operation

The TDP operates as a free-running counter in timer mode. The TDP starts counting up when the CST bit in TDPCR1 is set to 1. When TDPCNT overflows (H'FFFF changes to H'0000), the OVF bit in TDPCSR is set to 1 and an interrupt request is generated if the OVIE bit in TDPIER is 1. Figure 12.2 shows an example of free-running counter operation. In addition, figure 12.3 shows TDPCNT count timing for external clock operation. Note that the external clock requires a pulse width of at least 1.5 cycles. The counter will not operate correctly if the pulses are narrower than this.

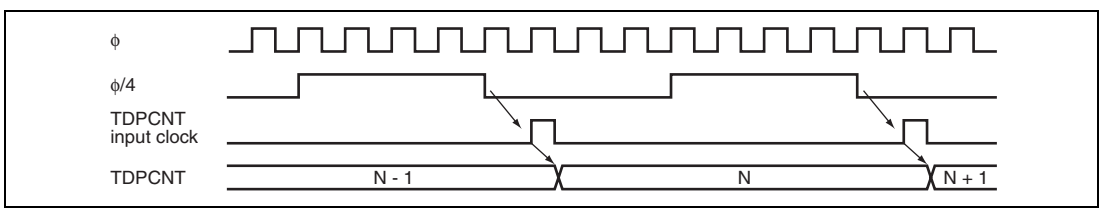

Figure 12.2 Example of Free-Running Counter Operation

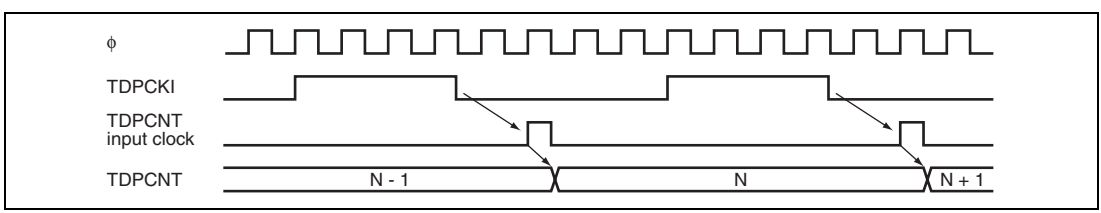

Figure 12.3 Count Timing of External Clock Operation (Falling Edges)

13.8 Interrupt Sources

TMR_0, TMR_1, and TMR_Y can generate three types of interrupts: CMIA, CMIB, and OVI. TMR_X can generate four types of interrupts: CMIA, CMIB, OVI, and ICIX. Table 13.6 shows the interrupt sources and priorities. Each interrupt source can be enabled or disabled independently by interrupt enable bits in TCR or TCSR. Independent signals are sent to the interrupt controller for each interrupt.

Table 13.6 Interrupt Sources of 8-Bit Timers TMR_0, TMR_1, TMR_Y, and TMR_X

Channel	Name	Interrupt Source	Interrupt Flag	Interrupt Priority
TMR_0	CMIA0	TCORA_0 compare-match	CMFA	High
	CMIB0	TCORB_0 compare-match	CMFB	
	OVI0	TCNT_0 overflow	OVF	
TMR_1	CMIA1	TCORA_1 compare-match	CMFA	
	CMIB1	TCORB_1 compare-match	CMFB	
	OVI1	TCNT_1 overflow	OVF	
TMR_Y	CMIAY	TCORA_Y compare-match	CMFA	
	CMIBY	TCORB_Y compare-match	CMFB	
	OVIY	TCNT_Y overflow	OVF	
TMR_X	ICIX	Input capture	ICF	
	CMIAX	TCORA_X compare-match	CMFA	
	CMIBX	TCORB_X compare-match	CMFB	
	OVIX	TCNT_X overflow	OVF	Low

Bit	Bit Name	Initial Value	R/W	Description
2	MP	0	R/W	Multiprocessor Mode (enabled only in asynchronous mode) When this bit is set to 1, the multiprocessor communication function is enabled. The PE bit and O/$\overline{E}$ bit settings are invalid in multiprocessor mode.
1	CKS1	0	R/W	Clock Select 1 and 0 These bits select the clock source for the baud rate generator. 00: ϕ clock (n = 0) 01: ϕ/4 clock (n = 1) 10: ϕ/16 clock (n = 2) 11: ϕ/64 clock (n = 3) For the relation between the bit rate register setting and the baud rate, see section 15.3.9, Bit Rate Register (BRR). n is the decimal display of the value of n in BRR (see section 15.3.9, Bit Rate Register (BRR)).
0	CKS0	0	R/W	

- Bit Functions in Smart Card Interface Mode (when SMIF in SCMR = 1)

Bit	Bit Name	Initial Value	R/W	Description
7	GM	0	R/W	GSM Mode Setting this bit to 1 allows GSM mode operation. In GSM mode, the TEND set timing is put forward to 11.0 etu* from the start and the clock output control function is appended. For details, see section 15.7.8, Clock Output Control.
6	BLK	0	R/W	Setting this bit to 1 allows block transfer mode operation. For details, see section 15.7.3, Block Transfer Mode.
5	PE	0	R/W	Parity Enable (valid only in asynchronous mode) When this bit is set to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. Set this bit to 1 in smart card interface mode.

15.6.3 Serial Data Transmission (Clocked Synchronous Mode)

Figure 15.16 shows an example of SCI operation for transmission in clocked synchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI monitors the TDRE flag in SSR, and if it is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.
2. After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a TXI interrupt request is generated. Because the TXI interrupt routine writes the next transmit data to TDR before transmission of the current transmit data has finished, continuous transmission can be enabled.
3. 8-bit data is sent from the TxD pin synchronized with the output clock when output clock mode has been specified and synchronized with the input clock when use of an external clock has been specified.
4. The SCI checks the TDRE flag at the timing for sending the last bit.
5. If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, and the TxD pin maintains the output state of the last bit. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated. The SCK pin is fixed high.

Figure 15.17 shows a sample flowchart for serial data transmission. Even if the TDRE flag is cleared to 0, transmission will not start while a receive error flag (ORER, FER, or PER) is set to 1. Make sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit to 0 does not clear the receive error flags.

16.5 Noise Canceler Circuit

The CIR incorporates a 4-stage noise canceler. The FLTE, FLT, and FLTCK1 and FLTCK0 bits in HHMAX enable/disable the noise canceler circuit, select the number of stages of the noise canceler circuit, and select the division ratio for generating the noise canceler circuit clock, respectively. Figure 16.6 shows a block diagram of the noise canceler circuit.

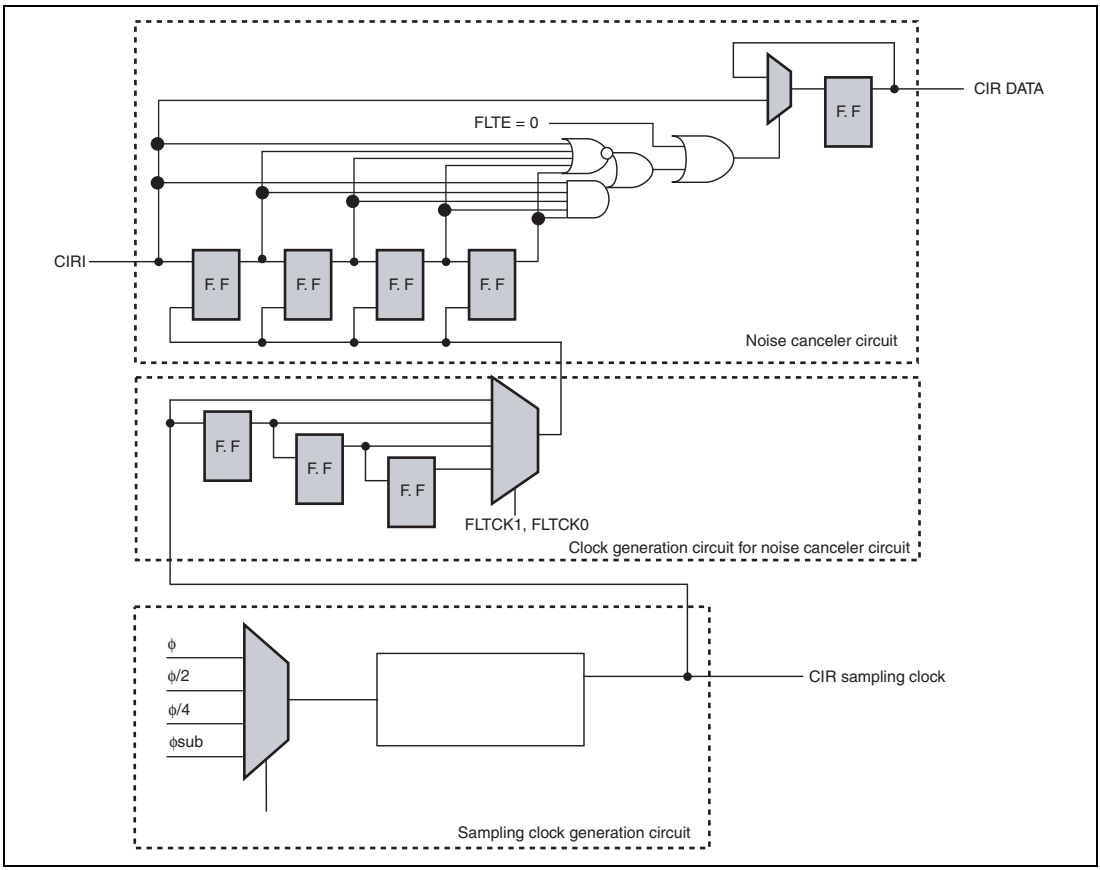

Figure 16.8 Noise Canceler Circuit

18.4.4 Master Receive Operation

In I²C bus format master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data.

The master device transmits data containing the slave address and R/$\overline{W}$ (1: read) in the first frame following the start condition issuance in master transmit mode, selects the slave device, and then switches the mode for receive operation.

Figure 18.10 shows the sample flowchart for the operations in master receive mode.

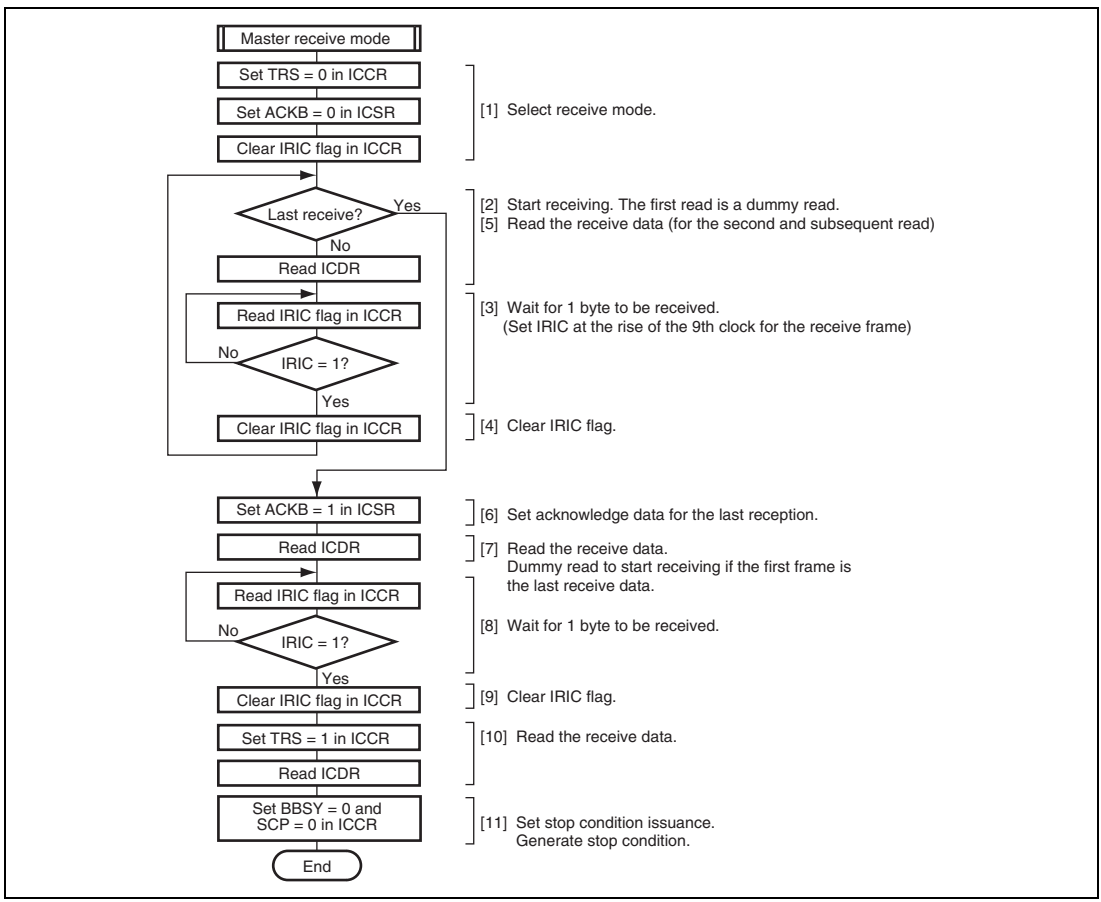

Figure 18.10 Sample Flowchart for Operations in Master Receive Mode

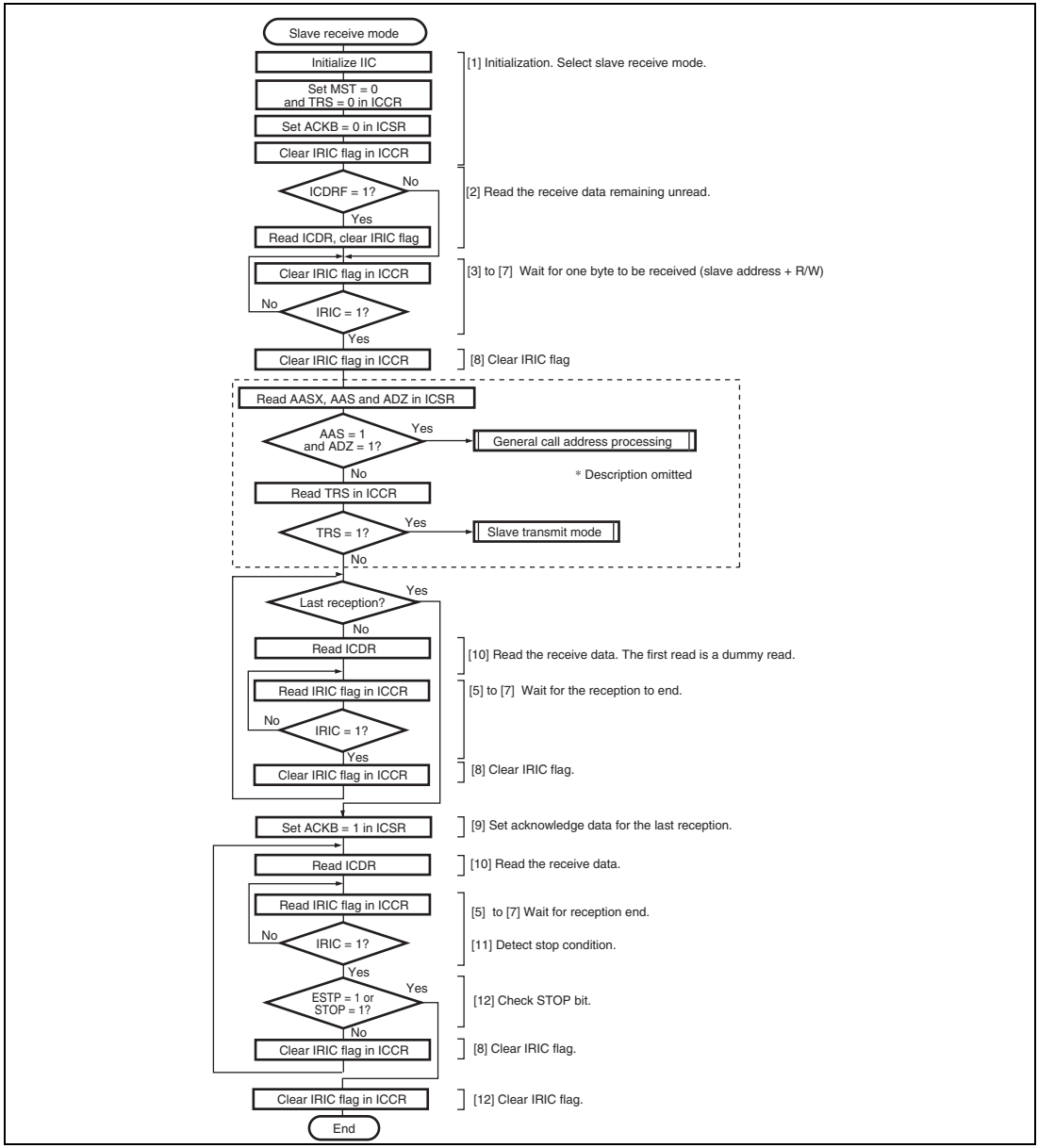

Figure 18.13 Sample Flowchart for Operations in Slave Receive Mode

Bit	Bit Name	Initial Value	R/W Slave	R/W Host	Description
5	IRQ11E4	0	R/W	—	Host IRQ11 Interrupt Enable 4 Enables or disables an HIRQ11 interrupt request when OBF4 is set by an ODR4 write. 0: HIRQ11 interrupt request by OBF4 and IRQE11E4 is disabled [Clearing conditions] • Writing 0 to IRQ11E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ11 interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] HIRQ11 interrupt is requested [Setting condition] Writing 1 after reading IRQ11E4 = 0
4	IRQ10E4	0	R/W	—	Host IRQ10 Interrupt Enable 4 Enables or disables an HIRQ10 interrupt request when OBF4 is set by an ODR4 write. 0: HIRQ10 interrupt request by OBF4 and IRQE10E4 is disabled [Clearing conditions] • Writing 0 to IRQ10E4 • LPC hardware reset, LPC software reset • Clearing OBF4 to 0 (when IEDIR4 = 0) 1: [When IEDIR4 = 0] HIRQ10 interrupt request by setting OBF4 to 1 is enabled [When IEDIR4 = 1] HIRQ10 interrupt is requested [Setting condition] Writing 1 after reading IRQ10E4 = 0

20.4.4 LPC Interface Shutdown Function (LPCPD)

The LPC interface can be placed in the shutdown state according to the state of the $\overline{\text{LPCPD}}$ pin. There are two kinds of LPC interface shutdown state: LPC hardware shutdown and LPC software shutdown. The LPC hardware shutdown state is controlled by the $\overline{\text{LPCPD}}$ pin, while the LPC software shutdown state is controlled by the SDWNB bit. In both states, the LPC interface enters the reset state by itself, and is no longer affected by external signals other than the $\overline{\text{LRESET}}$ and $\overline{\text{LPCPD}}$ signals.

Placing the slave in sleep mode or software standby mode is effective in reducing current dissipation in the shutdown state. If software standby mode is set, some means must be provided for exiting software standby mode before clearing the shutdown state with the $\overline{\text{LPCPD}}$ signal.

If the SDWNE bit has been set to 1 beforehand, the LPC hardware shutdown state is entered at the same time as the $\overline{\text{LPCPD}}$ signal falls, and prior preparation is not possible. If the LPC software shutdown state is set by means of the SDWNB bit, on the other hand, the LPC software shutdown state cannot be cleared at the same time as the rising edge of the $\overline{\text{LPCPD}}$ signal. Taking these points into consideration, the following operating procedure uses a combination of LPC software shutdown and LPC hardware shutdown.

1. Clear the SDWNE bit to 0.
2. Set the ERRIE bit to 1 and wait for an interrupt by the SDWN flag.
3. When an ERRI interrupt is generated by the SDWN flag, check the LPC interface internal status flags and perform any necessary processing.
4. Set the SDWNB bit to 1 to set LPC software standby mode.
5. Set the SDWNE bit to 1 and make a transition to LPC hardware standby mode. The SDWNB bit is cleared automatically.
6. Check the state of the $\overline{\text{LPCPD}}$ signal to make sure that the $\overline{\text{LPCPD}}$ signal has not risen during steps 3 to 5. If the signal has risen, clear SDWNE to 0 to return to the state in step 1.
7. If software standby mode has been set, exit software standby mode by some means independent of the LPC.
8. When a rising edge is detected in the $\overline{\text{LPCPD}}$ signal, the SDWNE bit is automatically cleared to 0. If the slave has been placed in sleep mode, the mode is exited by means of $\overline{\text{LRESET}}$ signal input, on completion of the LPC transfer cycle, or by some other means.

21.4.3 Flash Memory Instructions

Table 21.6 lists the flash memory instructions (INS).

Table 21.6 List of Instructions (INS)

Instruction Name	Description
WREN	Sets write-enable
WRDI	Resets write-enable
RDSR	Reads status register
WRSR	Writes status register
READ	Reads SPI flash memory
Fast-Read	Fast-reads SPI flash memory
Byte-Program	Byte-programs SPI flash memory
Page-Program	Page-programs SPI flash memory
AAI-Program	Address auto increment program
Sector-Erase	Sector erasure
Block-Erase	Block erasure
Chip/Bulk-Erase	Chip/bulk erasure
RDID	Reads manufacturing ID and product ID
EWSR	Enables status register write
DP (DEEP POWER DOWN)	Deep power-down
RES	Releases deep power-down

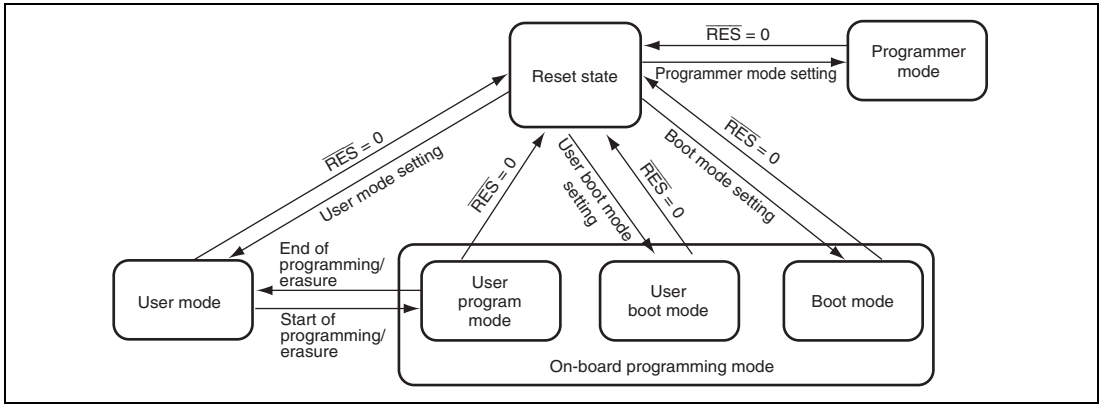

Figure 24.2 Mode Transition of Flash Memory

Table 24.1 Differences between Boot Mode, User Program Mode, and Programmer Mode

Item	Boot Mode	User Program Mode	User Boot Mode	Programmer Mode
Programming/ erasing environment	On-board programming	On-board programming	On-board programming	PROM programmer
Programming/ erasing enable MAT	• User MAT • User boot MAT	• User MAT	• User MAT	• User MAT • User boot MAT
All erasure	O (Automatic)	O	O	O (Automatic)
Block division erasure	O*[1]	O	O	×
Program data transfer	From host via SCI	Via any device	Via any device	Via programmer
Reset initiation MAT	Embedded program storage area	User MAT	User boot MAT*[2]	—
Transition to user mode	Changing mode and reset	Changing FLSHE bit setting	Changing mode and reset	—

Notes: 1. All-erasure is performed. After that, the specified block can be erased.
2. First, the reset vector is fetched from the embedded program storage MAT. After the flash memory related registers are checked, the reset vector is fetched from the user boot MAT.

(f) Operating Clock Frequency Inquiry

The boot program will return the number of operating clock frequencies, and the maximum and minimum values.

Command

H'23

- Command, H'23, (one byte): Inquiry regarding operating clock frequencies

Response

H'33	Size	Number of operating clock frequencies
Minimum value of operating clock frequency		Maximum value of operating clock frequency
…		
SUM		

- Response, H'33, (one byte): Response to operating clock frequency inquiry
- Size (one byte): The number of bytes that represents the minimum values, maximum values, and the number of frequencies.
- Number of operating clock frequencies (one byte): The number of supported operating clock frequency types
  (e.g. when there are two operating clock frequency types, which are the main and peripheral clocks, the number of types will be H'02.)
- Minimum value of operating clock frequency (two bytes): The minimum value of the divided clock frequency.

  The minimum and maximum values of the operating clock frequency represent the values in MHz, valid to the hundredths place of MHz, and multiplied by 100. (e.g. when the value is 20.00 MHz, it will be 2000, which is H'07D0.)
- Maximum value (two bytes): Maximum value among the divided clock frequencies.

  There are as many pairs of minimum and maximum values as there are operating clock frequencies.
- SUM (one byte): Checksum

27.3 Register States in Each Operating Mode

Register Abbreviation	Reset	High-Speed/Medium speed	Watch	Sleep	Module Stop	Software Standby	Module
PIDDR	Initialized	—	—	—	—	—	PORT
PJDDR	Initialized	—	—	—	—	—	
PIODR	Initialized	—	—	—	—	—	
PJODR	Initialized	—	—	—	—	—	
PIPIN	—	—	—	—	—	—	
PJPIN	—	—	—	—	—	—	
PJPCR	Initialized	—	—	—	—	—	
PINOCR	Initialized	—	—	—	—	—	
PJNOCR	Initialized	—	—	—	—	—	
CCR1	Initialized	—	—	—	—	—	CIR
CCR2	Initialized	—	—	—	—	—	
CSTR	Initialized	—	—	—	—	—	
CEIR	Initialized	—	—	—	—	—	
BRR	Initialized	—	—	—	—	—	
CIRRDR0 to 7	Initialized	—	—	—	—	—	
HHMIN	Initialized	—	—	—	—	—	
HHMAX	Initialized	—	—	—	—	—	
HLMIN	Initialized	—	—	—	—	—	
HLMAX	Initialized	—	—	—	—	—	
DT1MIN	Initialized	—	—	—	—	—	
DT1MAX	Initialized	—	—	—	—	—	
DT0MIN	Initialized	—	—	—	—	—	
DT0MAX	Initialized	—	—	—	—	—	
RMIN	Initialized	—	—	—	—	—	
RMAX	Initialized	—	—	—	—	—	

D. Treatment of Unused Pins

The treatments of unused pins are listed in table D.1.

Table D.1 Treatment of Unused Pins

Pin Name	Example of Pin Treatment
$\overline{\text{RES}}$	(Always used as a reset pin)
$\overline{\text{ETRST}}$	(Always used as a reset pin)
MD2, MD1	(Always used as mode pins)
NMI	• Connect to V_{CC} via a pull-up resistor
EXTAL	(Always used as a clock pin)
XTAL	(Always used as a clock pin)
Port 1 Port 2 Port 3 Port 4 Port 5 Port 6 Port 8 Port 9 Port A Port B Port C Port D Port F Port G Port H Port I Port J	• Connect each pin to V_{CC} via a pull-up resistor or to V_{SS} via a pull-down resistor
Port 7	• Connect each pin to AV_{CC} via a pull-up resistor or to AV_{SS} via a pull-down resistor
Port E	• Connect each pin to V_{CC} via a pull-up resistor