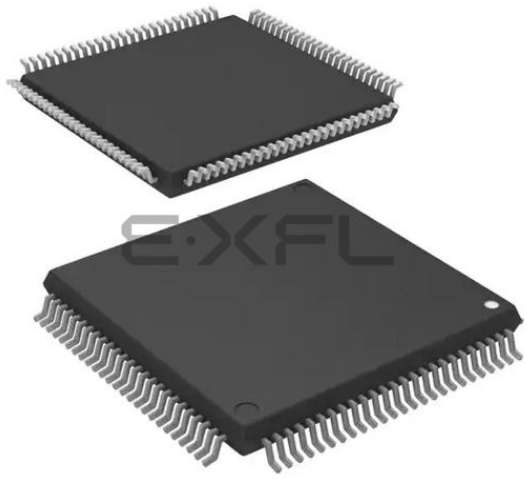
Renesas Electronics America Inc - R5F3640DDFB#30 Datasheet



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	Not For New Designs
Core Processor	M16C/60
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, I²C, SIO, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	85
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 26x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LFQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f3640ddfb-30

10. Clock Generation Circuit

10.1 Type of the Clock Generation Circuit

4 circuits are incorporated to generate the system clock signal:

- Main clock oscillation circuit
- Sub clock oscillation circuit
- 125 kHz on-chip oscillator
- PLL frequency synthesizer

Table 10.1 lists the Clock Generation Circuit Specifications. Figure 10.1 shows the System Clock Generation Circuit.
Figures 10.2 to 10.6 show the clock-related registers.

Table 10.1 Clock Generation Circuit Specifications

Item	Main Clock Oscillation Circuit	Sub Clock Oscillation Circuit	125 kHz On-Chip Oscillator	PLL Frequency Synthesizer
Use of Clock	CPU clock source Peripheral function clock source	CPU clock source Clock source of timer A and B.	CPU clock source Peripheral function clock source CPU and peripheral function clock sources when the main clock stops oscillating	CPU clock source Peripheral function clock source
Clock Frequency	0 to 20 MHz	32.768 kHz	125 kHz	10 to 25 MHz
Usable Oscillator	Ceramic oscillator Crystal oscillator	Crystal oscillator	-	- (1)
Pins to Connect Oscillator	XIN, XOUT	XCIN, XCOUT	-	- (1)
Oscillation Stop, Restart Function	Presence	Presence	Presence	Presence
Oscillator Status After Reset	Oscillating	Stopped	Oscillating	Stopped
Other	Externally derived clock can be input		-	- (1)

Note:
1. The PLL frequency synthesizer uses the main clock oscillation circuit as a reference clock source. The items above are based on those of the main clock oscillation circuit.

10.4.3.3 Exiting Stop Mode

Stop mode is exited by a hardware reset, $\overline{\text{NMI}}$ interrupt, low voltage detection interrupt, or peripheral function interrupt.

When the hardware reset, $\overline{\text{NMI}}$ interrupt, or low voltage detection interrupt is used to exit stop mode, set bits ILVL2 to ILVL0 in the interrupt control registers for the peripheral function interrupt to 000b (interrupt disabled) before setting the CM10 bit to 1.

When the peripheral function interrupt is used to exit stop mode, set the CM10 bit to 1 after the following settings are completed.

(1) Set bits ILVL2 to ILVL0 in the interrupt control registers to decide the peripheral priority level of the peripheral function interrupt.
Set the interrupt priority levels of the interrupts, not being used to exit stop mode, to 0 by setting bits ILVL2 to ILVL0 to 000b (interrupt disabled).

(2) Set the I flag to 1.

(3) Start operation of peripheral function being used to exit stop mode.
When exiting stop mode by the peripheral function interrupt, the interrupt routine is performed after an interrupt request is generated and then the CPU clock is supplied again.

When stop mode is exited by the peripheral function interrupt, low voltage detection interrupt, or $\overline{\text{NMI}}$ interrupt, the CPU clock source is as follows, in accordance with the CPU clock source setting before the microcomputer had entered stop mode.

- When the sub clock is the CPU clock before entering stop mode: sub clock
- When the main clock is the CPU clock source before entering stop mode: main clock divided by 8
- When the 125 kHz on-chip oscillator clock is the CPU clock source before entering stop mode: 125 kHz on-chip oscillator clock divided by 8

12.4.2 Relocatable Vector Tables

The 256 bytes beginning with the start address set in the INTB register comprise a relocatable vector table area. Tables 12.3 and 12.3 list the Relocatable Vector Tables. Setting an even address in the INTB register results in the interrupt sequence being executed faster than setting an odd address.

Table 12.2 Relocatable Vector Table (1)

Interrupt Source	Vector Address (1) Address (L) to Address (H)	Software Interrupt Number	Reference
BRK Instruction (5)	+0 to +3 (0000h to 0003h)	0	M16C/60, M16C/20, M16C/ Tiny Series Software Manual
– (Reserved)		1	
$\overline{\text{INT7}}$	+8 to +11 (0008h to 000Bh)	2	**12.6 “INT Interrupt”**
$\overline{\text{INT6}}$	+12 to +15 (000Ch to 000Fh)	3	
$\overline{\text{INT3}}$	+16 to +19 (0010h to 0013h)	4	
Timer B5	+20 to +23 (0014h to 0017h)	5	**15. “Timers”**
Timer B4, UART1 Bus Collision Detect (4, 6)	+24 to +27 (0018h to 001Bh)	6	**15. “Timers”** **17. “Serial Interface”**
Timer B3, UART0 Bus Collision Detect (4, 6)	+28 to +31 (001Ch to 001Fh)	7	
SI/O4, $\overline{\text{INT5}}$ (2)	+32 to +35 (0020h to 0023h)	8	**12.6 “INT Interrupt”** **17. “Serial Interface”**
SI/O3, $\overline{\text{INT4}}$ (2)	+36 to +39 (0024h to 0027h)	9	
UART2 Bus Collision Detection (6)	+40 to +43 (0028h to 002Bh)	10	**17. “Serial Interface”**
DMA0	+44 to +47 (002Ch to 002Fh)	11	**14. “DMAC”**
DMA1	+48 to +51 (0030h to 0033h)	12	
Key Input Interrupt	+52 to +55 (0034h to 0037h)	13	**12.8 “Key Input Interrupt”**
A/D Converter	+56 to +59 (0038h to 003Bh)	14	**18. “A/D Converter”**
UART2 Transmit, NACK2 (3)	+60 to +63 (003Ch to 003Fh)	15	**17. “Serial Interface”**
UART2 Receive, ACK2 (3)	+64 to +67 (0040h to 0043h)	16	
UART0 Transmit, NACK0 (3)	+68 to +71 (0044h to 0047h)	17	
UART0 Receive, ACK0 (3)	+72 to +75 (0048h to 004Bh)	18	
UART1 Transmit, NACK1 (3)	+76 to +79 (004Ch to 004Fh)	19	
UART1 Receive, ACK1 (3)	+80 to +83 (0050h to 0053h)	20	
Timer A0	+84 to +87 (0054h to 0057h)	21	**15. “Timers”**
Timer A1	+88 to +91 (0058h to 005Bh)	22	
Timer A2	+92 to +95 (005Ch to 005Fh)	23	
Timer A3	+96 to +99 (0060h to 0063h)	24	
Timer A4	+100 to +103 (0064h to 0067h)	25	
Timer B0	+104 to +107 (0068h to 006Bh)	26	
Timer B1	+108 to +111 (006Ch to 006Fh)	27	
Timer B2	+112 to +115 (0070h to 0073h)	28	

Notes:

1. Address relative to address in INTB.
2. Use bits IFSR6 and IFSR7 in the IFSR register to select.
3. During I^2C mode, interrupts NACK and ACK comprise the interrupt source.
4. Use bits IFSR26 and IFSR27 in the IFSR2A register to select.
5. These interrupts cannot be disabled using the I flag.
6. Bus collision detection: During IE mode, this bus collision detection constitutes the interrupt source. During I^2C mode, however, a start condition or a stop condition detection constitutes the interrupt source.

15.1.1 Timer Mode

In timer mode, the timer counts a count source generated internally. Table 15.1 lists the Specifications in Timer Mode. Figure 15.10 shows TAiMR Register in Timer Mode.

Table 15.1 Specifications in Timer Mode

Item	Specification
Count Source	f1TIMAB, f2TIMAB, f8TIMAB, f32TIMAB, f64TIMAB, fOCO-S, fC32
Count Operation	• Decrement • When the timer underflows, it reloads the reload register contents and continues counting
Divide Ratio	1 / (n+1) n: set value of TAi register 0000h to FFFFh
Count Start Condition	Set the TAiS bit in the TABSR register to 1 (start counting)
Count Stop Condition	Set the TAiS bit to 0 (stop counting)
Interrupt Request Generation Timing	Timer underflow
TAiIN Pin Function	I/O port or gate input
TAiOUT Pin Function	I/O port or pulse output
Read from Timer	Count value can be read by reading the TAi register
Write to Timer	• When not counting Value written to the TAi register is written to both reload register and counter • When counting Value written to the TAi register is written to only reload register (Transferred to counter when reloaded next)
Select Function	• Gate function Counting can be started and stopped by an input signal to the TAiIN pin • Pulse output function Whenever the timer underflows, the output polarity of TAiOUT pin is inverted. When the TAiS bit is set to 0 (stop counting), the pin outputs low.(when the POFS1 bit in the TAPOFS register is set to 1, the timer outputs high.) • Waveform output select function The output polarity of the TAiOUT pin is selected.

i = 0 to 4

Timer Ai Mode Register (i = 0 to 4)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	After Reset
		0				0	0	TA0MR to TA4MR	0336h to 033Ah	00h

Bit Symbol	Bit Name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Pulse output function select bit	0 : No pulse output (TAiOUT pin functions as I/O port) 1 : Pulse output (1) (TAiOUT pin functions as a pulse output pin)	RW
MR1	Gate function select bit	b4 b3 0 0 : Gate function not available 0 1 : (TAiIN pin functions as I/O port) 1 0 : Counts while input on the TAiIN pin is low (2) 1 1 : Counts while input on the TAiIN pin is high (2)	RW
MR2			RW
MR3	Set to 0 in timer mode		RW
TCK0	Count source select bit (4)	b1 b0 0 0 : f1TIMAB or f2TIMAB (3) 0 1 : f8TIMAB 1 0 : f32TIMAB 1 1 : fC32	RW
TCK1			

Notes :
1. The TA0OUT pin is N-channel open drain output.
2. Set the port direction bit for the TAiIN pin to 0 (input mode).
3. Selected by the PCLK0 bit in the PCLKR register.
4. Valid when the TCS3 bit or TCS7 bit in registers TACS0 to TACS2 is set to 0 (TCK0, TCK1 enabled).

Figure 15.10 TAiMR Register in Timer Mode

17. Serial Interface

Serial interfaces consist of eight channels: UART0 to UART2, UART5 to UART7, SI/O3, and SI/O4.

17.1 UARTi (i = 0 to 2, 5 to 7)

Each UARTi has an exclusive timer to generate a transfer clock, so it operates independently of each other.
Figures 17.1 to 17.3 show the block diagrams of UARTi. Figure 17.4 shows the UARTi Transmit / Receive Unit.

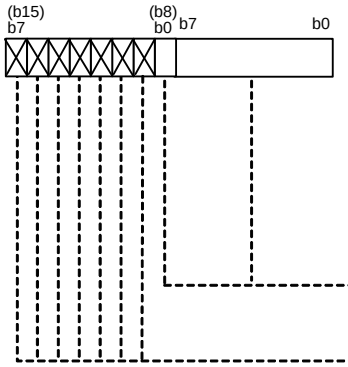
UARTi has the following modes:

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode)
- Special mode 1 (I^2C mode)
- Special mode 2
- Special mode 3 (Bus collision detection function, IE mode)
- Special mode 4 (SIM mode) : UART2

Figures 17.5 to 17.11 show the UARTi-related registers.
Refer to tables for each mode for register setting.

UART6 and UART7 cannot be used in memory expansion mode or microprocessor mode.

UARTi Transmit Buffer Register (i = 0 to 2, 5 to 7) (1, 2)

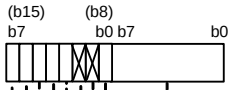


Symbol	Address	After Reset
U0TB	024Bh to 024Ah	Indeterminate
U1TB	025Bh to 025Ah	Indeterminate
U2TB	026Bh to 026Ah	Indeterminate
U5TB	028Bh to 028Ah	Indeterminate
U6TB	029Bh to 029Ah	Indeterminate
U7TB	02ABh to 02AAh	Indeterminate

Function	RW
Transmit data	WO
No register bits. If necessary, set to 0. Read as undefined value	—

Notes :
1. Use MOV instruction to write to this register.
2. When the transfer data is 9 bits long, write in 16-bit unit or write high byte then low byte in 8-bit unit.

UARTi Receive Buffer Register (i = 0 to 2, 5 to 7)



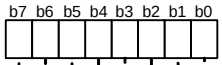
Symbol	Address	After Reset
U0RB	024Fh to 024Eh	Indeterminate
U1RB	025Fh to 025Eh	Indeterminate
U2RB	026Fh to 026Eh	Indeterminate
U5RB	028Fh to 028Eh	Indeterminate
U6RB	029Fh to 029Eh	Indeterminate
U7RB	02AFh to 02AEh	Indeterminate

Bit Symbol	Bit Name	Function	RW
— (b7-b0)	——	Receive data (D7 to D0)	RO
— (b8)	——	Receive data (D8)	RO
— (b10-b9)	No register bits. If necessary, set to 0. Read as undefined value		—
ABT	Arbitration lost detect flag (2)	0 : Not detected 1 : Detected	RW
OER	Overrun error flag (1)	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag (1, 3)	0 : No framing error 1 : Framing error found	RO
PER	Parity error flag (1,3)	0 : No parity error 1 : Parity error found	RO
SUM	Error sum flag (1, 3)	0 : No error 1 : Error found	RO

Notes :
1. When bits SMD2 to SMD0 in the UiMR register = 000b (serial interface disabled) or the RE bit in the UiC1 register = 0 (reception disabled), all of bits SUM, PER, FER, and OER are set to 0 (no error). The SUM bit is set to 0 (no error) when all of bits PER, FER, and OER = 0 (no error). Bits PER and FER are set to 0 by reading the lower byte of the UiRB register.
2. The ABT bit is set to 0 by writing 0 in a program. (Writing a 1 has no effect.)
3. These error flags are disabled when bits SMD2 to SMD0 are set to 001b (clock synchronous serial I/O mode) or to 010b (I^2C mode). Read as undefined values.

Figure 17.5 Registers U0TB to U2TB, U5TB to U7TB, U0RB to U2RB, and U5RB to U7RB

UARTi Transmit / Receive Control Register 0 (i = 0 to 2, 5 to 7)



Symbol	Address	After Reset
U0C0, U1C0, U2C0	024Ch, 025Ch, 026Ch	00001000b
U5C0, U6C0, U7C0	028Ch, 029Ch, 02ACh	00001000b

Bit Symbol	Bit Name	Function	RW
CLK0	UiBRG count source select bit (6)	b1 b0 0 0 : f1SIO or f2SIO is selected (5) 0 1 : f8SIO is selected 1 0 : f32SIO is selected 1 1 : Do not set to this value	RW
CLK1			RW
CRS	$\overline{CTS}$ / $\overline{RTS}$ function select bit (4)	Valid when CRD = 0 0 : $\overline{CTS}$ function selected (1) 1 : $\overline{RTS}$ function selected	RW
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	RO
CRD	$\overline{CTS}$ / $\overline{RTS}$ disable bit	0 : $\overline{CTS}$ / $\overline{RTS}$ function enabled 1 : $\overline{CTS}$ / $\overline{RTS}$ function disabled (P6_0, P6_4, P7_3, P8_1, P1_0, and P4_4 can be used as I/O ports)	RW
NCH	Data output select bit (2)	0 : Pins TXDi / SDAi and SCLi are CMOS output 1 : Pins TXDi / SDAi and SCLi are N-channel open-drain output	RW
CKPOL	CLK polarity select bit	0 : Transmit data is output at the falling edge of transfer clock and receive data is input at the rising edge 1 : Transmit data is output at the rising edge of transfer clock and receive data is input at the falling edge	RW
UFORM	Transfer format select bit (3)	0 : LSB first 1 : MSB first	RW

Notes :
1. Set the corresponding port direction bit for each $\overline{CTSi}$ pin to 0 (input mode).
2. TXD2 / SDA2 and SCL2 are N-channel open-drain output. Cannot be set to the CMOS output. No NCH bit in the U2C0 register is assigned. If necessary, set to 0.
3. The UFORM bit is enabled when bits SMD2 to SMD0 in the UiMR register are set to 001b (clock synchronous serial I/O mode), or 101b (UART mode, 8-bit transfer data).
   Set this bit to 1 when bits SMD2 to SMD0 are set to 010b (I^2C mode), and to 0 when bits SMD2 to SMD0 are set to 100b (UART mode, 7-bit transfer data) or 110b (UART mode, 9-bit transfer data).
4. $\overline{CTS1}$ / $\overline{RTS1}$ can be used when the CLKMD1 bit in the UCON register = 0 (only CLK1 output) and the RCSP bit in the UCON register = 0 ($\overline{CTS0}$ / $\overline{RTS0}$ not separated).
5. Selected by the PCLK1 bit in the PCLKR register.
6. When changing bits CLK1 and CLK0, set the UiBRG register.

Figure 17.7 Registers U0C0 to U2C0 and U5C0 to U7C0

17.1.1.5 Serial Data Logic Switching Function

When the UiLCH bit in the UiC1 register (i = 0 to 2, 5 to 7) = 1 (reverse), the data written to the UiTB register has its logic reversed before being transmitted. Similarly, the received data has its logic reversed when read from the UiRB register. Figure 17.15 shows Serial Data Logic Switching.

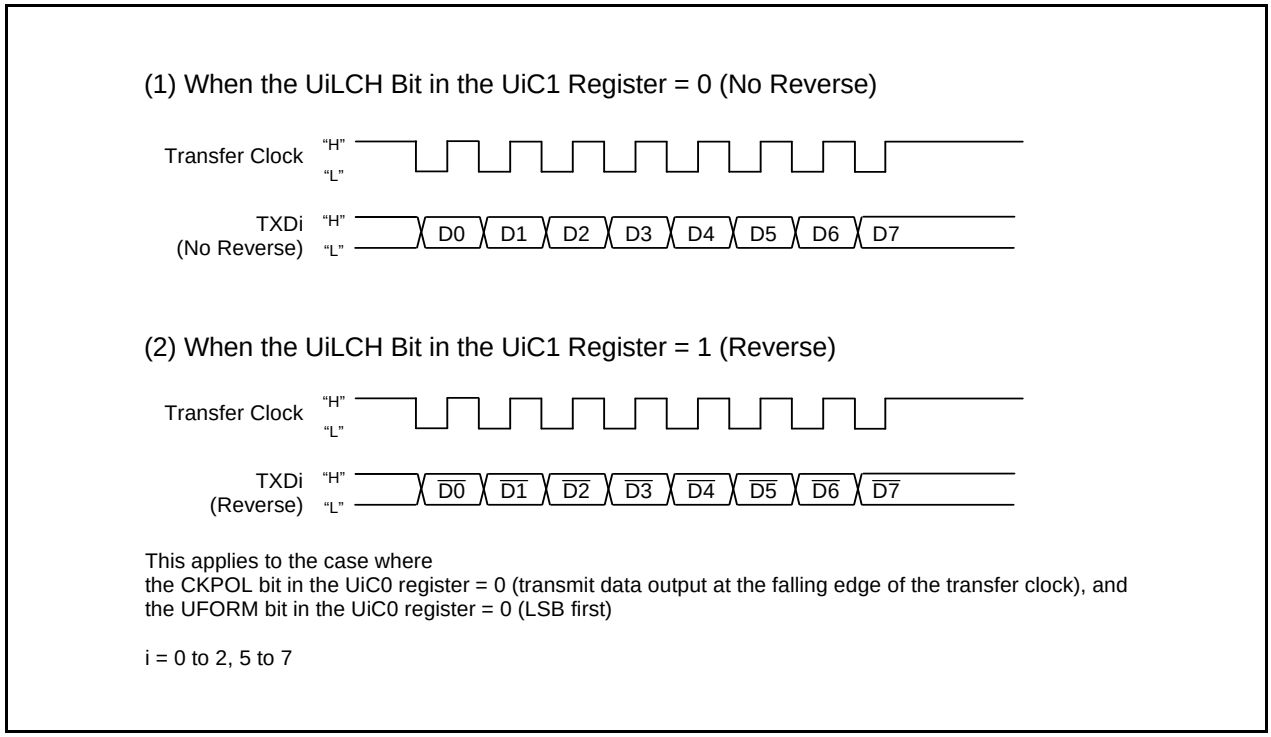


Figure 17.15 Serial Data Logic Switching

17.1.1.6 Transfer Clock Output from Multiple Pins (UART1)

Use bits CLKMD1 to CLKMD0 in the UCON register to select one of the two transfer clock output pins (see **Figure 17.16**). This function can be used when the selected transfer clock for UART1 is an internal clock.

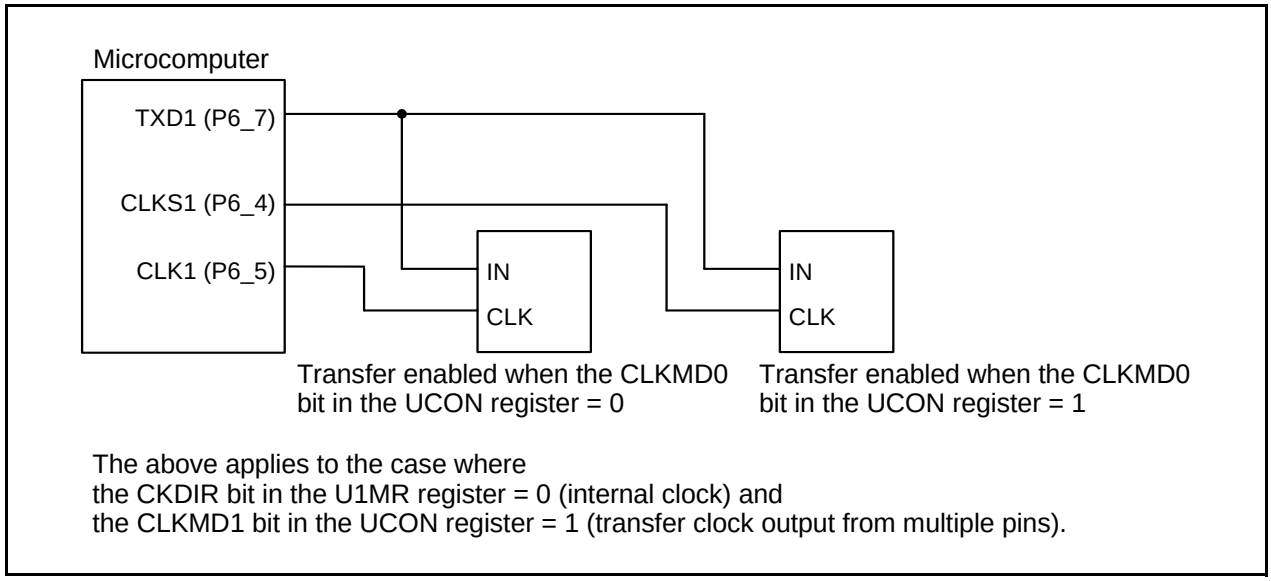


Figure 17.16 Transfer Clock Output from Multiple Pins

17.1.1.7 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ Function

The $\overline{\text{CTS}}$ function is used to start transmit and receive operation when “L” is applied to the $\overline{\text{CTSi}}$ / $\overline{\text{RTSi}}$ (i = 0 to 2, 5 to 7) pin. Transmit and receive operation begins when the $\overline{\text{CTSi}}$ / $\overline{\text{RTSi}}$ pin is held “L”. If the “L” signal is switched to “H” during a transmit or receive operation, the operation stops before the next data.

For the $\overline{\text{RTS}}$ function, the $\overline{\text{CTSi}}$ / $\overline{\text{RTSi}}$ pin outputs “L” when the microcomputer is ready to receive. The output level becomes “H” on the first falling edge of the CLKi pin.

- The CRD bit in the UiC0 register = 1 (disable $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ function) $\overline{\text{CTSi}}$ / $\overline{\text{RTSi}}$ pin is programmable I/O function
- The CRD bit = 0, CRS bit = 0 ($\overline{\text{CTS}}$ function selected) $\overline{\text{CTSi}}$ / $\overline{\text{RTSi}}$ pin is $\overline{\text{CTS}}$ function
- The CRD bit = 0, CRS bit = 1 ($\overline{\text{RTS}}$ function selected) $\overline{\text{CTSi}}$ / $\overline{\text{RTSi}}$ pin is $\overline{\text{RTS}}$ function

17.1.1.8 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ Separate Function (UART0)

This function separates $\overline{\text{CTS0}}$ / $\overline{\text{RTS0}}$, outputs $\overline{\text{RTS0}}$ from the P6_0 pin, and inputs $\overline{\text{CTS0}}$ from the P6_4 pin. To use this function, set the register bits as shown below.

- The CRD bit in the U0C0 register = 0 (enable $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ of UART0)
- The CRS bit in the U0C0 register = 1 (output $\overline{\text{RTS}}$ of UART0)
- The CRD bit in the U1C0 register = 0 (enable $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ of UART1)
- The CRS bit in the U1C0 register = 0 (input $\overline{\text{CTS}}$ of UART1)
- The RCSP bit in the UCON register = 1 (inputs $\overline{\text{CTS0}}$ from the P6_4 pin)
- The CLKMD1 bit in the UCON register = 0 (CLKS1 not used)

Note that when using the $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ separate function, $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ of UART1 function cannot be used.

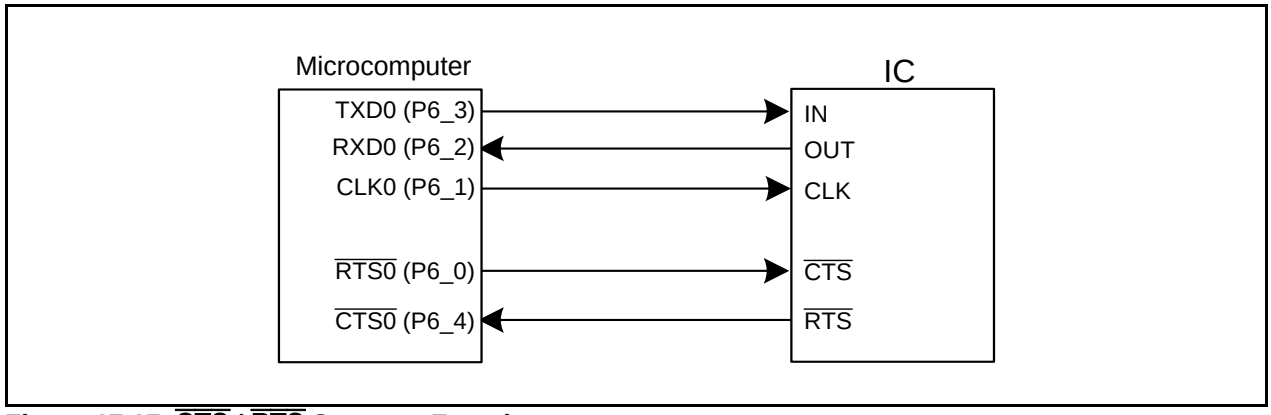


Figure 17.17 $\overline{\text{CTS}}$ / $\overline{\text{RTS}}$ Separate Function

17.1.5 Special Mode 3 (IE mode)

In this mode, one bit of IEBus is approximated with one byte of UART mode waveform.

Table 17.17 lists the Registers to Be Used and Settings in IE Mode. Figure 17.32 shows the Bus Collision Detect Function-Related Bits.

If the TXDi pin (i = 0 to 2, 5 to 7) output level and RXDi pin input level do not match, a UARTi bus collision detect interrupt request is generated.

Use bits IFSR26 and IFSR27 in the IFSR2A register to enable the UART0 / UART1 bus collision detect function.

Table 17.17 Registers to Be Used and Settings in IE Mode

Register	Bit	Function
UiTB	0 to 8	Set transmission data
UiRB (3)	0 to 8	Reception data can be read
	OER, FER, PER, SUM	Error flag
UiBRG	0 to 7	Set a bit rate
UiMR	SMD2 to SMD0	Set to 110b
	CKDIR	Select the internal clock or external clock
	STPS	Set to 0
	PRY	Invalid because PRYE = 0
	PRYE	Set to 0
	IOPOL	Select the TXD and RXD input / output polarity
UiC0	CLK1, CLK0	Select the count source for the UiBRG register
	CRS	Invalid because CRD = 1
	TXEPT	Transmit register empty flag
	CRD	Set to 1
	NCH	Select TXDi pin output format (2)
	CKPOL	Set to 0
	UFORM	Set to 0
UiC1	TE	Set to 1 to enable transmission
	TI	Transmit buffer empty flag
	RE	Set to 1 to enable reception
	RI	Reception complete flag
	UjIRS (1)	Select the source of UARTj transmit interrupt
	UjRRM (1)	Set to 0
	UiLCH	Set to 0
	UiERE	Set to 0
UiSMR	0 to 3, 7	Set to 0
	ABSCS	Select the sampling timing at which to detect a bus collision
	ACSE	Set this bit to 1 to use the auto clear function of transmit enable bit
	SSS	Select the transmit start condition
UiSMR2	0 to 7	Set to 0
UiSMR3	0 to 7	Set to 0
UiSMR4	0 to 7	Set to 0
IFSR2A	IFSR26, IFSR27	Set to 1
UCON	U0IRS	Select the source of UART0 transmit interrupt
	U1IRS	Select the source of UART1 transmit interrupt
	U0RRM	Set to 0
	U1RRM	Set to 0
	CLKMD0	Invalid because CLKMD1 = 0
	CLKMD1, RCSP, 7	Set to 0

i = 0 to 2, 5 to 7
j = 2, 5 to 7
Notes:
1. Set bits 4 and 5 in registers U0C0 and U1C1 to 0. Bits U0IRS, U1IRS, U0RRM, and U1RRM are in the UCON register.
2. The TXD2 pin is N channel open-drain output. No NCH bit in the U2C0 register is assigned. When write, set to 0.
3. Set the bits not listed above to 0 when writing to the registers in IE mode.

17.2 SI/O3 and SI/O4

SI/O3 and SI/O4 are exclusive clock-synchronous serial I/Os.

Figure 17.37 shows the SI/O3 and SI/O4 Block Diagram, and Figure 17.38 shows the Registers S3C, S4C, S3BRG, S4BRG, S3TRR, and S4TRR.

Table 17.20 shows the SI/O3 and SI/O4 Specifications.

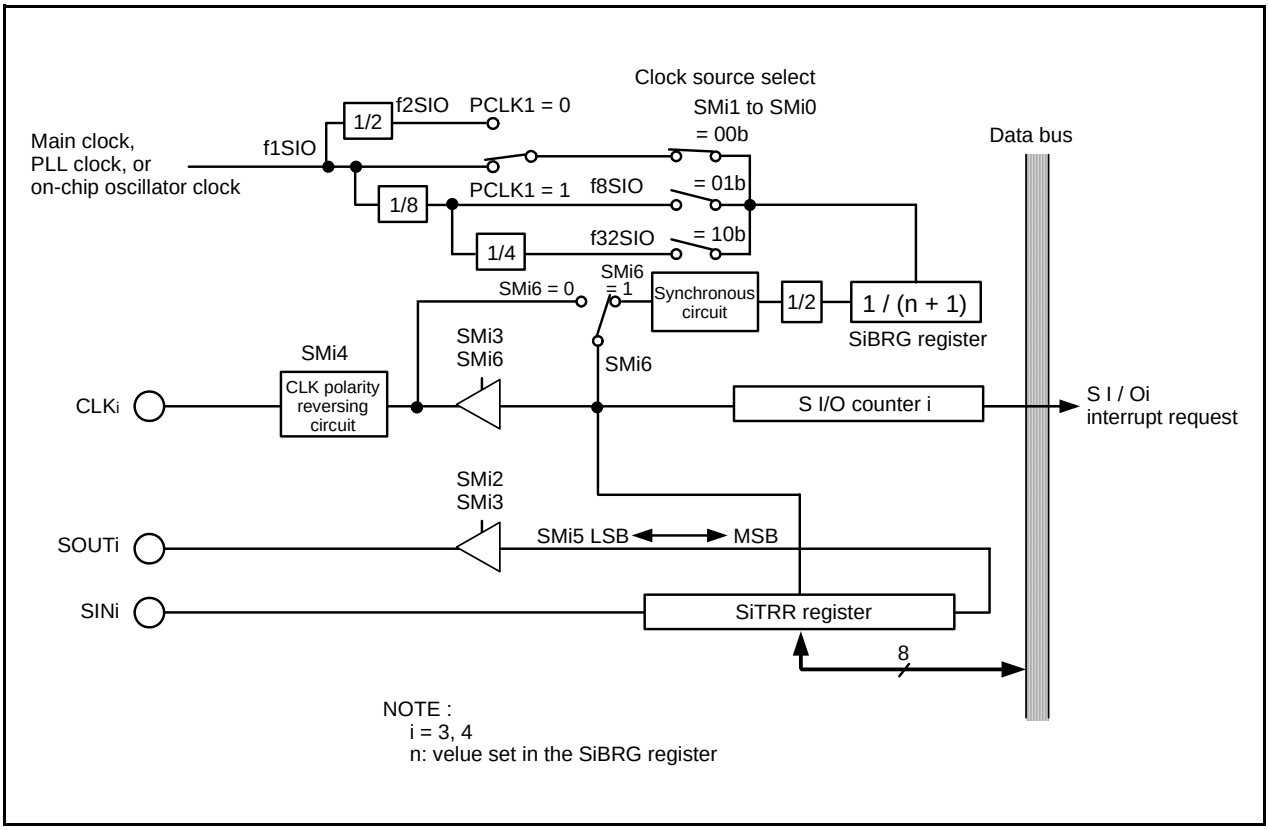


Figure 17.37 SI/O3 and SI/O4 Block Diagram

18.1.5 Repeat Sweep Mode 1

In repeat sweep mode 1, analog voltage selectively applied to all pins is repeatedly converted to a digital code. Table 18.6 shows the Repeat Sweep Mode 1 Specifications. Figure 18.8 shows Registers ADCON0 and ADCON1 (Repeat Sweep Mode 1).

Table 18.6 Repeat Sweep Mode 1 Specifications

Item	Specification
Function	The input voltages on all pins selected by bits ADGSEL1 and ADGSEL0 in the ADCON2 register are A/D converted repeatedly, with priority given to pins selected by bits SCAN1 and SCAN0 in the ADCON1 register and bits ADGSEL1 and ADGSEL0. Example: If AN0 selected, input voltages are A/D converted in order of AN0→AN1→AN0→AN2→AN0→AN3, and so on.
A/D Conversion Start Condition	• When the TRG bit in the ADCON0 register is 0 (software trigger), the ADST bit in the ADCON0 register is set to 1 (A/D conversion start) • When the TRG bit is 1 ($\overline{\text{ADTRG}}$ trigger), input on the $\overline{\text{ADTRG}}$ pin changes state from high to low after the ADST bit is set to 1 (A/D conversion start)
A/D Conversion Stop Condition	Set the ADST bit to 0 (A/D conversion stop)
Interrupt Request Generation timing	No interrupt requests generated
Analog Input Pins to be Given Priority When A/D Converted	Select from AN0 (1 pin), AN0 and AN1 (2 pins), AN0 to AN2 (3 pins), and AN0 to AN3 (4 pins) (1)
Reading of Result of A/D Converter	Read one of the registers AD0 to AD7 that corresponds to the selected pin

Notes:
1. AN0_0 to AN0_7 and AN2_0 to AN2_7 can be used in the same way as AN0 to AN7.

18.5 External Sensor

To perform A/D conversion accurately, charging the internal capacitor C shown in Figure 18.9 must be completed within a specified period of time.

T: Specified period of time (sampling time)
R0: Output impedance of sensor equivalent circuit
R: Internal resistance of the MCU
X: Precision (error) of the A/D converter
Y: Resolution of the A/D converter be Y (Y is 1024)

Generally, $VC = VIN\left\{1 - e^{-\frac{1}{C(R0+R)}t}\right\}$

When t = T, $VC = VIN - \frac{X}{Y}VIN = VIN\left(1 - \frac{X}{Y}\right)$

$$e^{-\frac{1}{C(R0+R)}T} = \frac{X}{Y}$$

$$-\frac{1}{C(R0+R)}T = \ln\frac{X}{Y}$$

Therefore, $R0 = -\frac{T}{C \bullet \ln\frac{X}{Y}} - R$

Figure 18.9 shows Analog Input Pin and External Sensor Equivalent Circuit. Impedance R0 by which voltage VC between pins of the capacitor C changes from 0 to VIN - (0.1/1024)VIN in time T when the difference between VIN and VC is 0.1LSB is obtained. (0.1/1024) means that A/D precision drop due to insufficient capacitor charge is kept to 0.1LSB in A/D conversion. Actual error however is the value of absolute accuracy added to 0.1LSB.
When ϕAD is 20 MHz, T is 0.75 μs. Output impedance R0 for charging capacitor C sufficiently within the time T is obtained as follows.

T = 0.75 μs, R = 10 kΩ, C = 6.0 pF, X = 0.1, and Y = 1024. Therefore,

$$R0 = -\frac{0.75 \times 10^{-6}}{6.0 \times 10^{-12} \bullet \ln\frac{0.1}{1024}} - 10 \times 10^{3} \approx 3.5 \times 10^{3}$$

Thus, the output impedance R0 of the sensor equivalent circuit, making the A/D converter precision (error) 0.1LSB or less, is up to 3.5 kΩ.

22.1 Memory Map

The flash memory contains program ROM 1, program ROM 2, and a data flash. Figure 22.1 shows a Flash Memory Block Diagram.

Program ROM 1 is divided into several blocks, each of which can be protected (locked) from program or erase. Program ROM 1 and program ROM 2 can be rewritten in CPU rewrite, standard serial I/O, and parallel I/O modes.

Program ROM 2 can be used when the PRG2C0 bit in the PRG2C register is set to 0 (program ROM 2 enabled). The user boot code area is in program ROM 2. Data flash can be used when the PM10 bit in the PM1 register is set to 1 (0E000h to 0FFFFh: data flash). Data flash is divided into block A and block B.

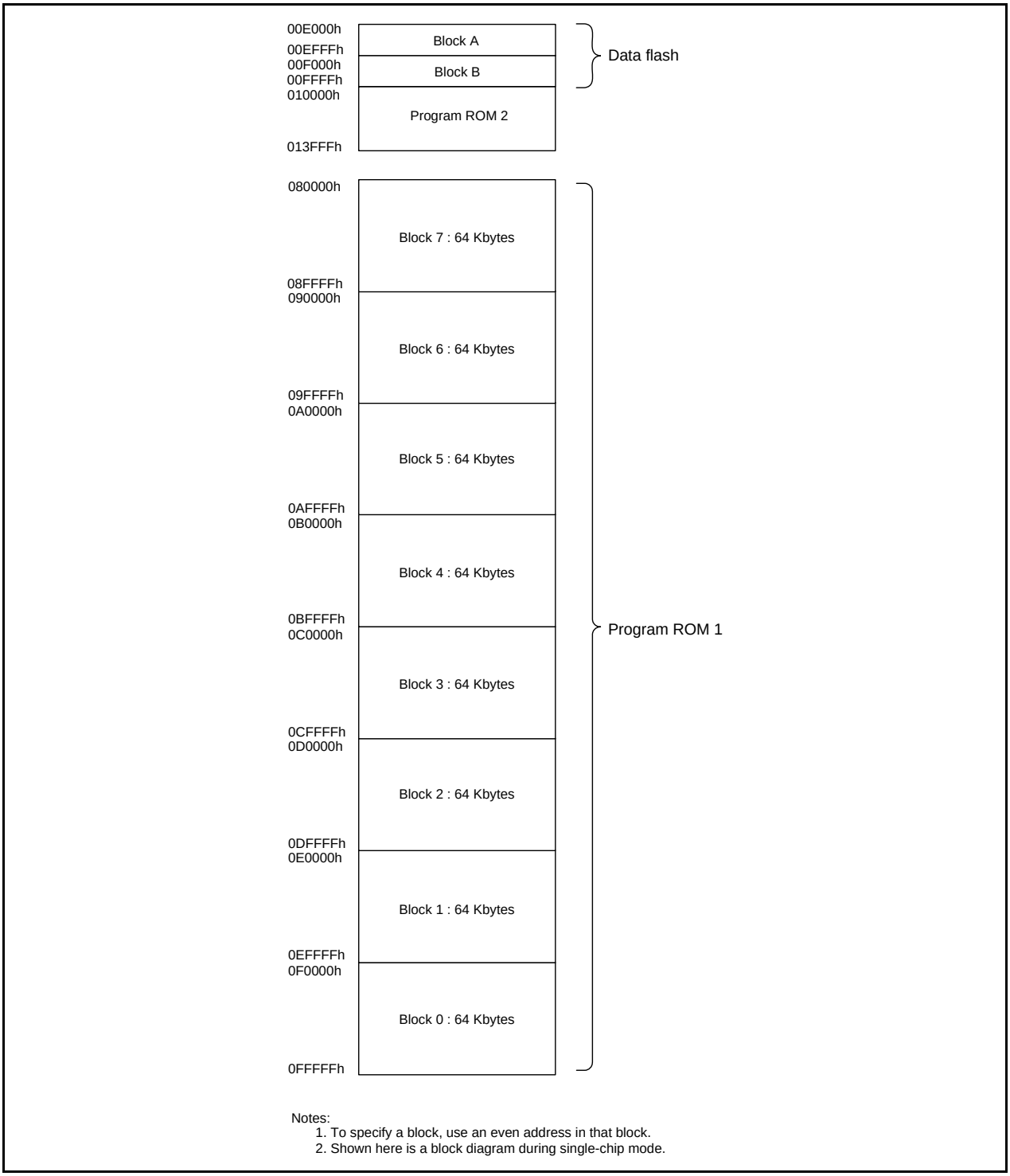


Figure 22.1 Flash Memory Block Diagram

22.3.4 Precautions on CPU Rewrite Mode

22.3.4.1 Operating Speed

Set the CM06 bit in the CM0 register and bits CM17 and CM16 in the CM1 register to a CPU clock frequency of 10 MHz or less before entering CPU rewrite mode (EW0 or EW1 mode). Also, set the PM17 bit in the PM1 register to 1 (wait state).

22.3.4.2 Prohibited Instructions

The following instructions cannot be used in EW0 mode because the CPU tries to read data in the flash memory: the UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction.

22.3.4.3 Interrupts (EW0 mode)

- To use interrupts with vectors in a relocatable vector table, relocate the vectors to the RAM area.
- The $\overline{\text{NMI}}$ and watchdog timer interrupts are available since registers FMR0 and FMR1 are forcibly reset when either interrupt occurs. Allocate the jump addresses for each interrupt routine to the fixed vector table. Flash memory rewrite operation stops when the $\overline{\text{NMI}}$ or watchdog timer interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.
- The address match interrupt is not available since the CPU tries to read data in the flash memory.

22.3.4.4 Interrupts (EW1 mode)

- Do not acknowledge any interrupts with vectors in a relocatable vector table or address match interrupt during the auto program or auto erase period.
- The $\overline{\text{NMI}}$ interrupt is available since registers FMR0 and FMR1 are forcibly reset when the interrupt occurs. Allocate the jump address for the interrupt routine to the fixed vector table. Flash memory rewrite operation stops when the $\overline{\text{NMI}}$ interrupt occurs. Execute the rewrite program again after exiting the interrupt routine.
- The watchdog timer stops counting during auto write and auto erase.

22.3.4.5 How to Access

To set the FMR01 or FMR02 bit to 1, write a 1 after first setting the bit to 0. Make sure that no interrupts or no DMA transfers will occur before writing a 1 after writing a 0.

22.3.4.6 Rewrite (EW0 mode)

If the supply voltage drops while rewriting the block where the rewrite control program is stored, the rewrite control program is not correctly rewritten. This may cause the flash memory not to be rewritten. If this error occurs, use standard serial I/O mode or parallel I/O mode for rewriting.

22.3.4.7 Rewrite (EW1 mode)

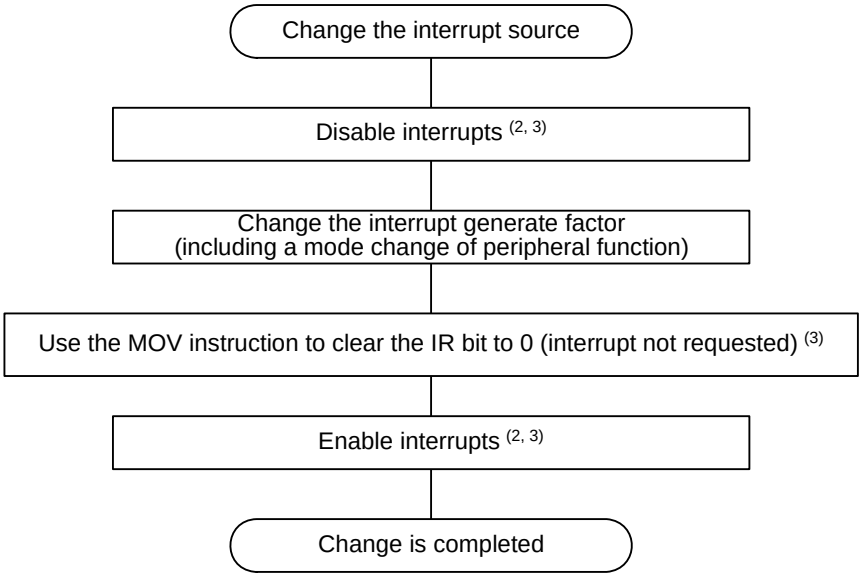
- Do not rewrite any block in which the rewrite control program is stored.
- When a software command is executed for another block on the flash memory, the software command may not be executed as expected. This may occur when RAM is written by the rewriting program which is executed in EW1 mode.

24.3 Bus

- When hardware reset 1 or brown-out reset is performed with “H” input on the CNVSS pin, contents of internal ROM cannot be read.

24.7.4 Changing an Interrupt Generate Factor

If the interrupt generate factor is changed, the IR bit in the interrupt control register for the changed interrupt may inadvertently be set to 1 (interrupt requested). To use an interrupt, change the interrupt generate factor, and then be sure to clear the IR bit for that interrupt to 0 (interrupt not requested).
Changing the interrupt generate factor referred to here means any act of changing the source, polarity or timing of the interrupt assigned to each software interrupt number. Therefore, if a mode change of any peripheral function involves changing the source, polarity or timing of an interrupt, be sure to clear the IR bit for that interrupt to 0 (interrupt not requested) after making such changes. Refer to the description of each peripheral function for details about the interrupts from peripheral functions.
Figure 24.3 shows the Procedure for Changing the Interrupt Generate Factor.



IR bit: A bit in the interrupt control register for the interrupt whose interrupt generate factor is to be changed

Notes :
1. The above settings must be executed individually. Do not execute two or more settings simultaneously (using one instruction).
2. Use the I flag for the $\overline{\text{INTi}}$ interrupt (i = 0 to 5).
For the interrupts from peripheral functions other than the $\overline{\text{INTi}}$ interrupt, turn off the peripheral function that is the source of the interrupt in order not to generate an interrupt request before changing the interrupt generate factor. In this case, if the maskable interrupts can all be disabled without causing a problem, use the I flag. Otherwise, use the corresponding bits ILVL2 to ILVL0 for the interrupt whose interrupt generate factor is to be changed.
3. Refer to **24.7.6 Rewriting the Interrupt Control Register** for details about the instructions to use and the notes to be taken for instruction execution.

Figure 24.3 Procedure for Changing the Interrupt Generate Factor

24.7.5 $\overline{\text{INT}}$ Interrupt

- Either an “L” level of at least tw (INL) width or an “H” level of at least tw (INH) width is necessary for the signal input to pins $\overline{\text{INT0}}$ through $\overline{\text{INT7}}$ regardless of the CPU operation clock.
- If the POL bit in registers INT0IC to INT7IC, bits IFSR7 to IFSR0 in the IFSR register, or bits IFSR31 and IFSR30 in the IFSR3A register are changed, the IR bit may inadvertently set to 1 (interrupt requested). Be sure to clear the IR bit to 0 (interrupt not requested) after changing any of those register bits.

24.8 DMAC

24.8.1 Write to the DMAE Bit in the DMiCON Register (i = 0 to 3)

When both of the conditions below are met, follow the steps below.

Conditions

- The DMAE bit is set to 1 (DMAi is in active state) again while it remains 1.
- A DMA request may occur simultaneously when the DMAE bit is being written.

Steps

(1) Write a 1 to the DMAE bit and DMAS bit in the DMiCON register simultaneously (1).

(2) Make sure that the DMAi is in initial state (2) in a program.
If the DMAi is not in initial state, repeat the above steps.

Notes:

1. The DMAS bit remains unchanged even if a 1 is written. However, if a 0 is written to this bit, it is set to 0 (DMA not requested). In order to prevent the DMAS bit from being modified to 0, 1 should be written to the DMAS bit when 1 is written to the DMAE bit. In this way the state of the DMAS bit immediately before being written can be maintained.
2. Similarly, when writing to the DMAE bit with a read-modify-write instruction, 1 should be written to the DMAS bit in order to maintain a DMA request which is generated during execution.
3. Read the TCRi register to verify whether the DMAi is in initial state. If the read value is equal to a value which was written to the TCRi register before DMA transfer start, the DMAi is in initial state. (In the case a DMA request occurs after writing to the DMAE bit, the read value is a value written to the TCRi register minus one.) If the read value is a value in the middle of transfer, the DMAi is not in initial state.

24.8.2 DMAC Channel Priority

When a same DMA request source is selected for DMAC channels, the DMA transfer may not be executed in the priority order of DMA0>DMA1>DMA2>DMA3.

24.9 Timers

24.9.1 Timer A

24.9.1.1 Timer A (Timer Mode)

The timer is stopped after reset. Set the mode, count source, counter value, and others using registers TAiMR, TAi, TACS0 to TACS2, and TAPOFS before setting the TAiS bit in the TABSR register to 1 (count starts) (i = 0 to 4).
Always make sure registers TAiMR, TACS0 to TACS2, and TAPOFS are modified while the TAiS bit is 0 (count stops) regardless of whether after reset or not.

While counting is in progress, the counter value can be read out at any time by reading the TAi register. However, if the counter is read at the same time it is reloaded, the value FFFFh is read. Also, if the counter is read before it starts counting after a value is set in the TAi register while not counting, the set value is read.

If a low-level signal is applied to the $\overline{SD}$ pin when the IVPCR1 bit in the TB2SC register = 1 (three-phase output forcible cutoff by input on the $\overline{SD}$ pin enabled), pins TA1OUT, TA2OUT, and TA4OUT go to high-impedance state.