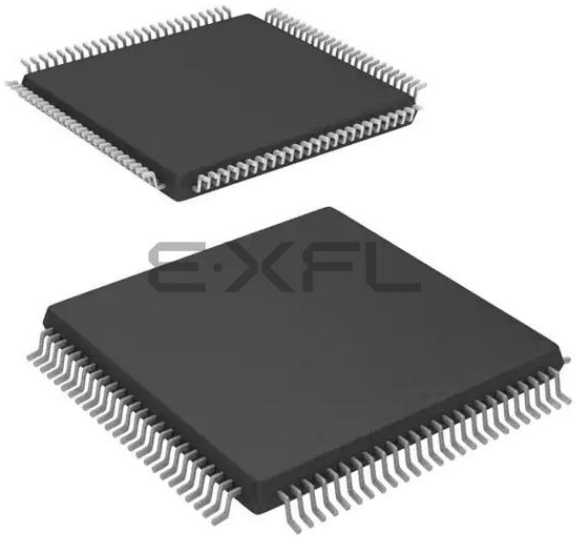

Welcome to E-XFL.COM

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	25MHz
Connectivity	SCI, SmartCard
Peripherals	POR, PWM, WDT
Number of I/O	70
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K 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	100-TQFP
Supplier Device Package	100-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2317vtf25v

Item	Specification
Other features	• Differences between H8S/2319 F-ZTAT and H8S/2319C F-ZTAT — On-chip RAM H8S/2319 F-ZTAT: 8 kbytes (H'FFDC00 to H'FFFBFF) H8S/2319C F-ZTAT: 16 kbytes (H'FFBC00 to H'FFFBFF) — On-chip flash memory The H8S/2319 F-ZTAT and H8S/2319C F-ZTAT both have 512 kbytes of on-chip flash memory. However, the method for controlling the flash memory is different for the two LSIs. When the on-chip flash memory is enabled, the registers (parameters) used to control it are different. For details, see the section about the H8S/2319 F-ZTAT and H8S/2319C F-ZTAT in section 17, ROM. — Address map The address maps of the H8S/2319 F-ZTAT and H8S/2319C F-ZTAT differ in places. For details, see section 3.5, Memory Map in Each Operating Mode.

(6) Immediate—#xx:8, #xx:16, or #xx:32: The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC): This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

(8) Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'000000 to H'0000FF).

In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be all 0 (H'00).

Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

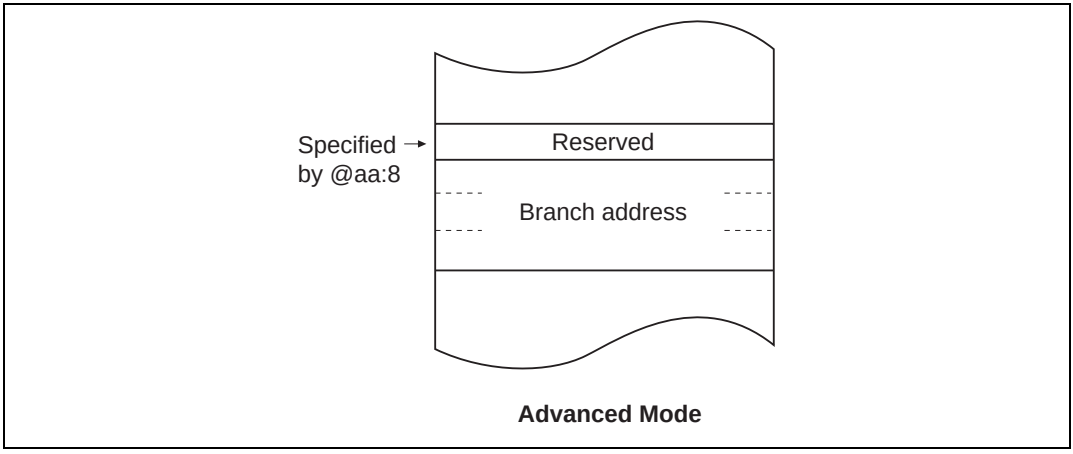

Figure 2.10 Branch Address Specification in Memory Indirect Mode

5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

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
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCR (composed of ISCRH and ISCRL) is a 16-bit readable/writable register that selects rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$.

ISCR is initialized to H'0000 by a reset and in hardware standby mode.

Bits 15 to 0—IRQ7 Sense Control A and B (IRQ7SCA, IRQ7SCB) to IRQ0 Sense Control A and B (IRQ0SCA, IRQ0SCB)

Bits 15 to 0

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input low level (Initial value)
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5 W61	Bit 4 W60	Description
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1.

Bit 3 W51	Bit 2 W50	Description
0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed (Initial value)

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the number of program wait states when area 4 in external space is accessed while the AST4 bit in ASTCR is set to 1.

Bit 1 W41	Bit 0 W40	Description
0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed (Initial value)

16-Bit 2-State Access Space: Figures 6.8 to 6.10 show bus timings for a 16-bit 2-state access space. When a 16-bit access space is accessed, the upper half (D_{15} to D_8) of the data bus is used for the even address, and the lower half (D_7 to D_0) for the odd address.

Wait states cannot be inserted.

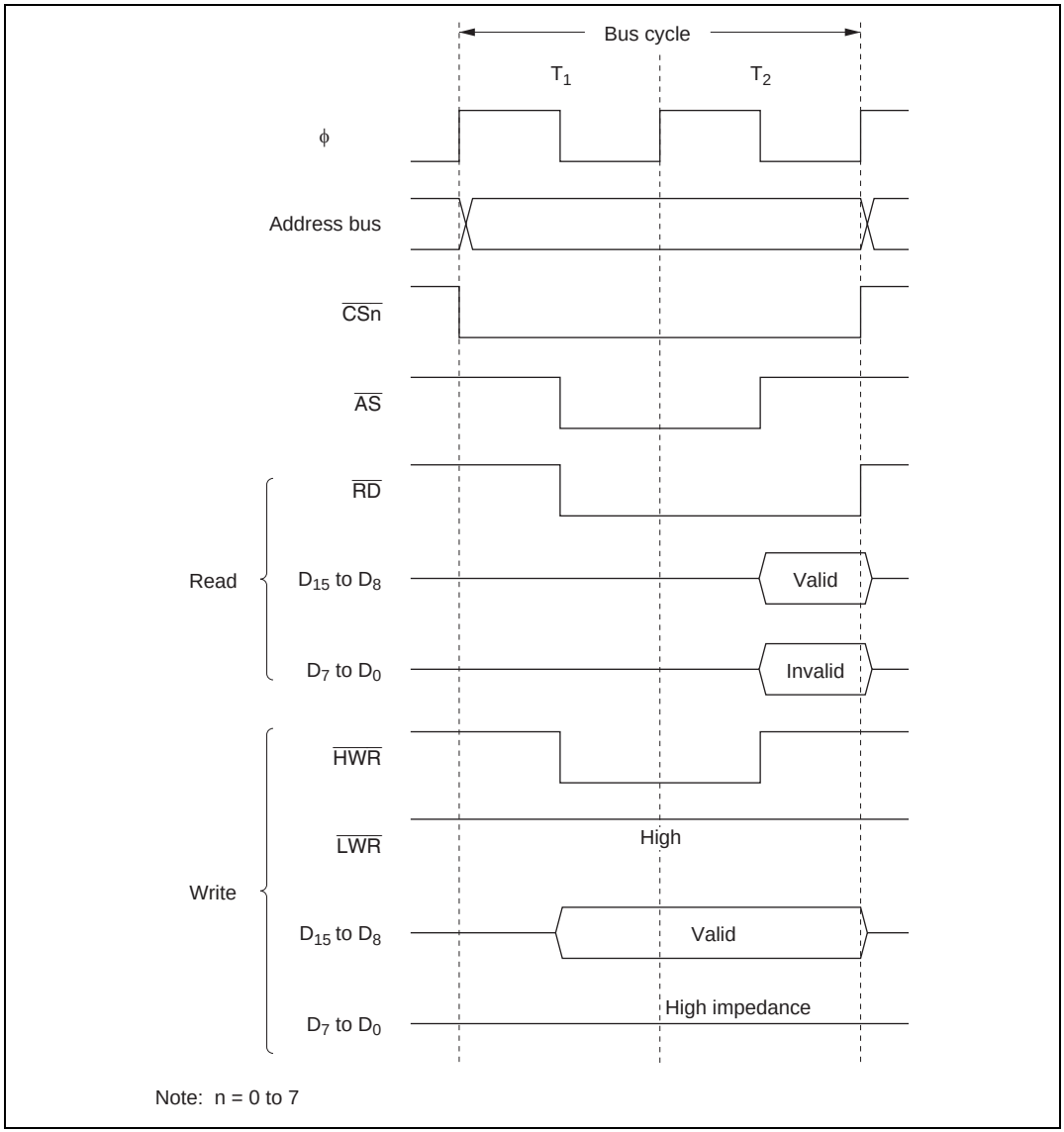

Figure 6.8 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

Port 2 Data Direction Register (P2DDR)

Bit	:	7	6	5	4	3	2	1	0
		P27DDR	P26DDR	P25DDR	P24DDR	P23DDR	P22DDR	P21DDR	P20DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P2DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 2. P2DDR cannot be read; if it is, an undefined value will be read.

Setting P2DDR bits to 1 makes the corresponding port 2 pins output pins, while clearing the bits to 0 makes the pins input pins.

P2DDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 2 Data Register (P2DR)

Bit	:	7	6	5	4	3	2	1	0
		P27DR	P26DR	P25DR	P24DR	P23DR	P22DR	P21DR	P20DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P2DR is an 8-bit readable/writable register that stores output data for the port 2 pins (P27 to P20).

P2DR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

8.9.2 Register Configuration

Table 8.15 shows the port D register configuration.

Table 8.15 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port D data direction register	PDDDR	W	H'00	H'FEBC
Port D data register	PDDR	R/W	H'00	H'FF6C
Port D register	PORTD	R	Undefined	H'FF5C
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'FF73

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

PDDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

- Modes 4 to 6*

  The input/output direction specification by PDDDR is ignored, and port D is automatically designated for data I/O.

- Mode 7*

  Setting PDDDR bits to 1 makes the corresponding port D pins output ports, while clearing the bits to 0 makes the pins input ports.

Note: * Modes 6 and 7 are not available in the ROMless versions.

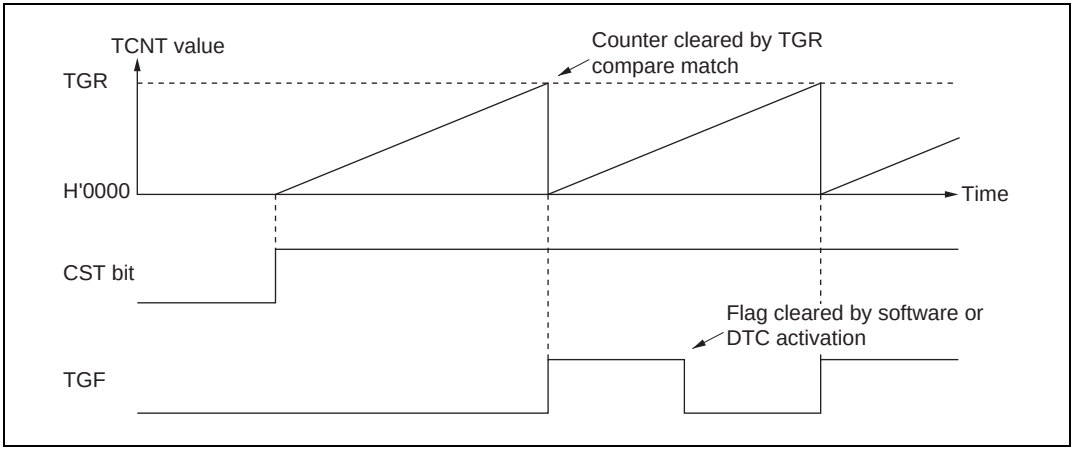

Figure 9.8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match
  Figure 9.9 shows an example of the setting procedure for waveform output by compare match

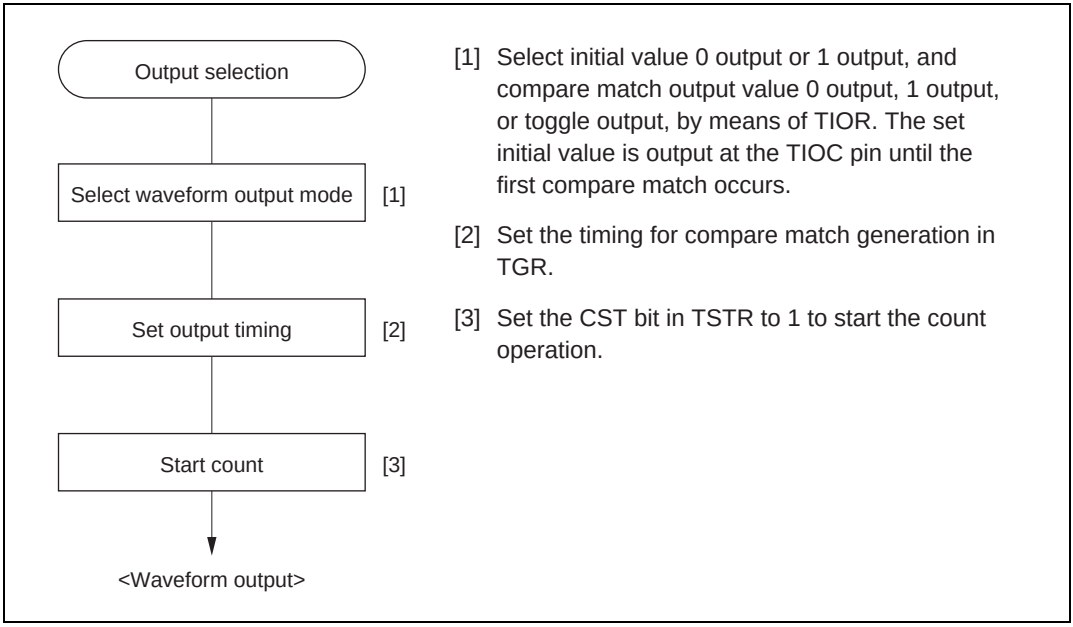

Figure 9.9 Example of Setting Procedure for Waveform Output by Compare Match

In serial reception, the SCI operates as described below.

[1] The SCI performs internal initialization in synchronization with serial clock input or output.

[2] The received data is stored in RSR in LSB-to-MSB order.

After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.

If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 12.11.

Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.

[3] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive-data-full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive-error interrupt (ERI) request is generated.

Figure 12.19 shows an example of SCI operation in reception.

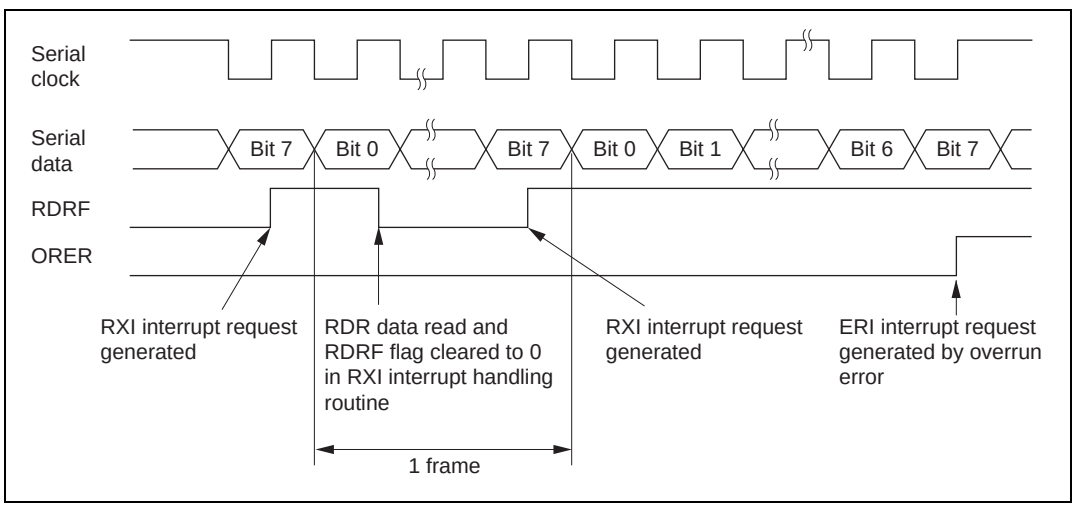

Figure 12.19 Example of SCI Receive Operation

Simultaneous serial data transmission and reception (synchronous mode): Figure 12.20 shows a sample flowchart for simultaneous serial transmit and receive operations.

The following procedure should be used for simultaneous serial data transmit and receive operations.

17.7 Programming/Erasing Flash Memory

In the on-board programming modes, flash memory programming and erasing is performed by software, using the CPU. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transition to these modes can be made for the on-chip ROM area by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program that controls flash memory programming/erasing (the programming control program) should be located and executed in on-chip RAM or external memory. When the program is located in external memory, an instruction for programming the flash memory and the following instruction should be located in on-chip RAM. The DTC should not be activated before or after the instruction for programming the flash memory is executed.

Notes: 1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Perform programming in the erased state. Do not perform additional programming on previously programmed addresses.

17.7.1 Program Mode

Follow the procedure shown in the program/program-verify flowchart in figure 17.15 to write data or programs to flash memory. Performing program operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

For the wait times (x, y, z1, z2, z3, α, ß, γ, ε, η, θ) after bits are set or cleared in flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N), see section 20.3.6, Flash Memory Characteristics.

Following the elapse of (x) μs or more after the SWE bit is set to 1 in flash memory control register 1 (FLMCR1), 128-byte program data is stored in the program data area and reprogram data area, and the 128-byte data in the reprogram data area is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 or H'80. 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

AC Characteristics

Table 17.19 AC Characteristics in Auto-Program Mode

Conditions: V_{CC} = 3.3 V ±0.3 V, V_{SS} = 0 V, T_a = 25°C ±5°C

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{wsts}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Address setup time	t_{as}	0	—	ns
Address hold time	t_{ah}	60	—	ns
Memory write time	t_{write}	1	3000	ms
Write setup time	t_{pns}	100	—	ns
Write end setup time	t_{pnh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

Bits 2 to 0—Flash Memory Area Selection (RAM2 to RAM0): These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM (see table 17.51).

Table 17.51 Flash Memory Area Divisions

RAM Area	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFDC00 to H'FFEBFF	RAM area, 4 kbytes	0	×	×	×
H'000000 to H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000 to H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000 to H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000 to H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000 to H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000 to H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000 to H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000 to H'007FFF	EB7 (4 kbytes)	1	1	1	1

×: Don't care

Download and initialization are performed only once at the beginning.

In this kind of operation, note the following:

- Be careful not to damage on-chip RAM with overlapped settings.
  In addition to the RAM emulation area, erasing program area, and programming program area, areas for the user procedure programs, work area, and stack area are reserved in on-chip RAM. Do not make settings that will overwrite data in these areas.
- Be sure to initialize both the erasing program and programming program.
  Initialization by setting the FPEFEQ parameter must be performed for both the erasing program and the programming program. Initialization must be executed for both entry addresses: (download start address for erasing program) + 32 bytes (H'FFBC20 in this example) and (download start address for programming program) + 32 bytes (H'FFCC20 in this example).

17.24.3 User Boot Mode

This LSI has user boot mode which is initiated with different mode pin settings than those in user program mode or boot mode. User boot mode is a user-arbitrary boot mode, unlike boot mode that uses the on-chip SCI.

Only the user MAT can be programmed/erased in user boot mode. Programming/erasing of the user boot MAT is only enabled in boot mode or programmer mode.

User Boot Mode Initiation: For the mode pin settings to start up user boot mode, see table 17.52.

When the reset start is executed in user boot mode, the built-in check routine runs. The user MAT and user boot MAT states are checked by this check routine.

While the check routine is running, NMI and all other interrupts cannot be accepted.

Next, processing starts from the execution start address of the reset vector in the user boot MAT. At this point, H'AA is set to the flash MAT select register FMATS because the execution MAT is the user boot MAT.

User MAT Programming in User Boot Mode: For programming the user MAT in user boot mode, additional processings made by setting FMATS are required: switching from user-boot-MAT selection state to user-MAT selection state, and switching back to user-boot-MAT selection state after programming completes.

Figure 17.73 shows the procedure for programming the user MAT in user boot mode.

(3) Operating frequency error

Operating frequency is calculated from the received value of the input frequency and the multiplication or division ratio. The input frequency is input to the LSI and the LSI is operated at the operating frequency. The expression is given below.

Operating frequency = Input frequency × Multiplication ratio , or

Operating frequency = Input frequency ÷ Division ratio

The calculated operating frequency should be checked to ensure that it is within the range of minimum to maximum frequencies which are available with the clock modes of the specified device. When it is out of this range, an operating frequency error is generated.

(4) Bit rate

Peripheral operating clock (ϕ), bit rate (B), clock select (CKS) in the serial mode register (SMR). The error as calculated by the method below is checked to ensure that it is less than 4%. When it is 4% or more, a bit-rate selection error is generated.

$$\text{Error (\%)} = \{[\frac{\phi * 10^6}{(N+1) * B * 64 * 2^{(2*n-1)}}] - 1\} * 100$$

When the new bit rate is selectable, the rate will be set in the register after sending ACK in response. The host will send an ACK with the new bit rate for confirmation and the boot program will response with that rate.

Confirmation | H'06 |

- Confirmation, H'06, (1 byte): Confirmation of a new bit rate

Response | H'06 |

- Response, H'06, (1 byte): Response to confirmation of a new bit rate

The sequence of new bit-rate selection is shown in figure 17.85.

Bit 6 STS2	Bit 5 STS1	Bit 4 STS0	Description
0	0	0	Standby time = 8192 states (Initial value)
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states*

Note: * Not available in the F-ZTAT versions.

Bit 3—Output Port Enable (OPE): Specifies whether the output of the address bus and bus control signals ($\overline{CS0}$ to $\overline{CS7}$, $\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$, $\overline{CAS}$) is retained or set to the high-impedance state in software standby mode.

Bit 3 OPE	Description
0	In software standby mode, address bus and bus control signals are high-impedance
1	In software standby mode, address bus and bus control signals retain output state (Initial value)

Bits 2 and 1—Reserved: These bits cannot be modified and are always read as 0.

Bit 0—IRQ37 Software Standby Clear Select (IRQ37S): Specifies whether inputs $\overline{IRQ3}$ to $\overline{IRQ7}$ can be used as software standby mode clearing sources in addition to the usual sources, NMI and $\overline{IRQ0}$ to $\overline{IRQ2}$ inputs.

Bit 0 IRQ37S	Description
0	Inputs $\overline{IRQ3}$ to $\overline{IRQ7}$ cannot be used as software standby mode clearing sources (Initial value)
1	Inputs $\overline{IRQ3}$ to $\overline{IRQ7}$ can be used as software standby mode clearing sources

(2) Control Signal Timing

Table 20.24 Control Signal 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)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 20.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 20.5
NMI hold time	t_{NMIH}	10	—	ns	
NMI pulse width (in recovery from software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (in recovery from software standby mode)	t_{IRQW}	200	—	ns	

Table A.4 Number of States per Cycle

		Access Conditions						
			On-Chip Supporting Module		External Device			
					8-Bit Bus		16-Bit Bus	
Cycle		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	1	4	2	4	6 + 2m	2	3 + m
Branch address read	S_J							
Stack operation	S_K							
Byte data access	S_L		2		2	3 + m		
Word data access	S_M		4		4	6 + 2m		
Internal operation	S_N	1	1	1	1	1	1	1

Legend:

m: Number of wait states inserted into external device access

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
OR	OR.B #xx:8,Rd	1					
	OR.B Rs,Rd	1					
	OR.W #xx:16,Rd	2					
	OR.W Rs,Rd	1					
	OR.L #xx:32,ERd	3					
	OR.L ERs,ERd	2					
ORC	ORC #xx:8,CCR	1					
	ORC #xx:8,EXR	2					
POP	POP.W Rn	1				1	1
	POP.L ERn	2				2	1
PUSH	PUSH.W Rn	1				1	1
	PUSH.L ERn	2				2	1
ROTL	ROTL.B Rd	1					
	ROTL.B #2,Rd	1					
	ROTL.W Rd	1					
	ROTL.W #2,Rd	1					
	ROTL.L ERd	1					
	ROTL.L #2,ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.B #2,Rd	1					
	ROTR.W Rd	1					
	ROTR.W #2,Rd	1					
	ROTR.L ERd	1					
	ROTR.L #2,ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.B #2,Rd	1					
	ROTXL.W Rd	1					
	ROTXL.W #2,Rd	1					
	ROTXL.L ERd	1					
	ROTXL.L #2,ERd	1					

Instruction	1	2	3	4	5	6	7	8	9
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,Rd	R:W NEXT								
BNOT Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BOR #xx:3,Rd	R:W NEXT								
BOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BSET #xx:3,Rd	R:W NEXT								
BSET #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,Rd	R:W NEXT								
BSET Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSR d:8	R:W NEXT	R:W EA	W:W:M stack (H)	W:W:M stack (L)					
BSR d:16	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
BST #xx:3,Rd	R:W NEXT								
BST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BTST #xx:3,Rd	R:W NEXT								
BTST #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						

C.4 Port 4

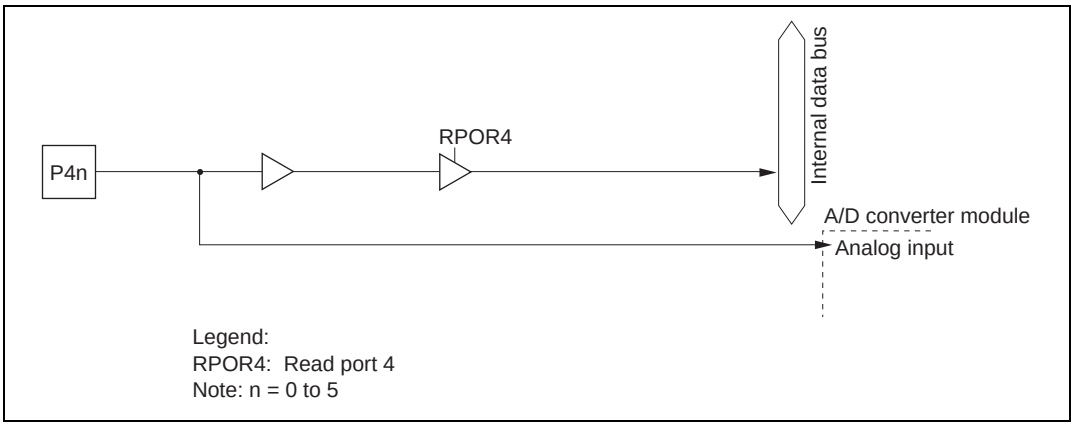

Figure C.4(a) Port 4 Block Diagram (Pins P40 to P45)

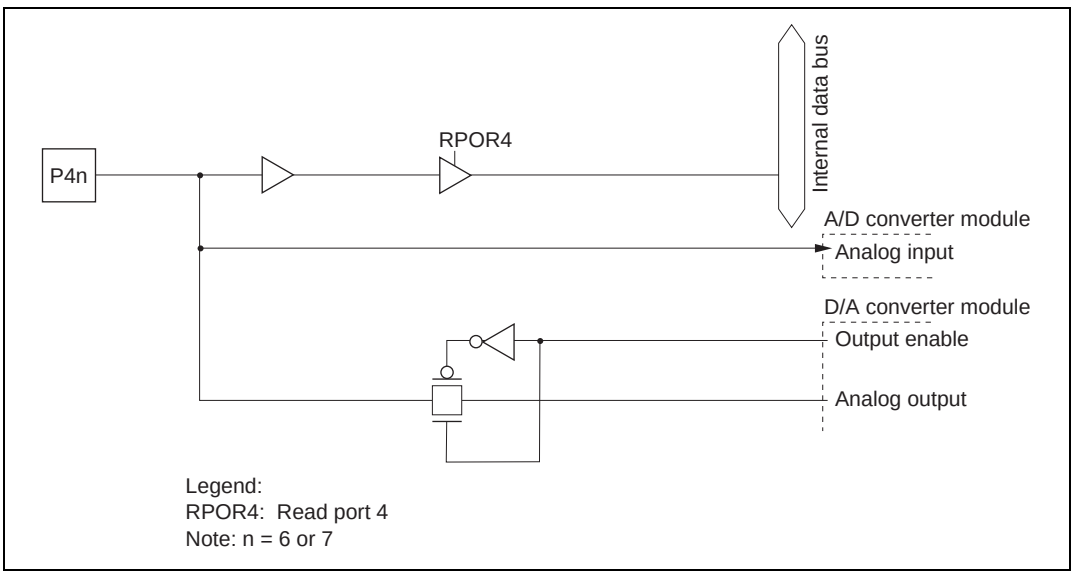

Figure C.4(b) Port 4 Block Diagram (Pins P46 and P47)