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	Obsolete
Core Processor	H8/300H
Core Size	16-Bit
Speed	25MHz
Connectivity	SCI, SmartCard
Peripherals	PWM, WDT
Number of I/O	70
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 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-BFQFP
Supplier Device Package	100-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df3024fbl25v

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4.1 indicates, exception handling may be caused by a reset, interrupt, or trap instruction. Exception handling is prioritized as shown in table 4.1. If two or more exceptions occur simultaneously, they are accepted and processed in priority order. Trap instruction exceptions are accepted at all times in the program execution state.

Table 4.1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin
▲	Interrupt	Interrupt requests are handled when execution of the current instruction or handling of the current exception is completed
Low	Trap instruction (TRAPA)	Started by execution of a trap instruction (TRAPA)

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows.

1. The program counter (PC) and condition code register (CCR) are pushed onto the stack.
2. The CCR interrupt mask bit is set to 1.
3. A vector address corresponding to the exception source is generated, and program execution starts from that address.

Note: For a reset exception, steps 2 and 3 above are carried out.

Section 7 I/O Ports

7.1 Overview

The H8/3024 Group has 10 input/output ports (ports 1, 2, 3, 4, 5, 6, 8, 9, A, and B) and one input-only port (port 7). Table 7.1 summarizes the port functions. The pins in each port are multiplexed as shown in table 7.1.

Each port has a data direction register (DDR) for selecting input or output, and a data register (DR) for storing output data. In addition to these registers, ports 2, 4, and 5 have an input pull-up control register (PCR) for switching input pull-up transistors on and off.

Ports 1 to 6 and port 8 can drive one TTL load and a 90-pF capacitive load. Ports 9, A, and B can drive one TTL load and a 30-pF capacitive load. Ports 1 to 6 and 8 to B can drive a darlington pair. Ports 1, 2, and 5 can drive LEDs (with 10-mA current sink). Pins $P8_2$ to $P8_0$, PA_7 to PA_0 have Schmitt-trigger input circuits.

For block diagrams of the ports see appendix C, I/O Port Block Diagrams.

Table 7.1 Port Functions

Port	Description	Pins	Expanded Modes					Single-Chip Modes	
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port 1	• 8-bit I/O port • Can drive LEDs	$P1_7$ to $P1_0$/ A_7 to A_0	Address output pins (A_7 to A_0)				Address output (A_7 to A_0) and generic input DDR = 0: generic input DDR = 1: address output	Generic input/ output	
Port 2	• 8-bit I/O port • Built-in input pull-up transistors • Can drive LEDs	$P2_7$ to $P2_0$/ A_{15} to A_8	Address output pins (A_{15} to A_8)				Address output (A_{15} to A_8) and generic input DDR = 0: generic input DDR = 1: address output	Generic input/ output	
Port 3	• 8-bit I/O port	$P3_7$ to $P3_0$/ D_{15} to D_8	Data input/output (D_{15} to D_8)					Generic input/ output	

Port	Description	Pins	Expanded Modes					Single-Chip Modes	
			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6	Mode 7
Port 4	• 8-bit I/O port • Built-in input pull-up transistors	$P4_7$ to $P4_0$/ D_7 to D_0	Data input/output (D_7 to D_0) and 8-bit generic input/output 8-bit bus mode: generic input/output 16-bit bus mode: data input/output					Generic input/ output	
Port 5	• 4-bit I/O port • Built-in input pull-up transistors • Can drive LEDs	$P5_3$ to $P5_0$/ A_{19} to A_{16}	Address output (A_{19} to A_{16})				Address output (A_{19} to A_{16}) and 4-bit generic input DDR = 0: generic input DDR = 1: address output	Generic input/ output	
Port 6	• 8-bit I/O port	$P6_7$/ϕ	Clock output (ϕ) and generic input						
		$P6_6$/$\overline{LWR}$ $P6_5$/$\overline{HWR}$ $P6_4$/$\overline{RD}$ $P6_3$/$\overline{AS}$	Bus control signal output ($\overline{LWR}$, $\overline{HWR}$, $\overline{RD}$, $\overline{AS}$)					Generic input/ output	
		$P6_2$/$\overline{BACK}$ $P6_1$/$\overline{BREQ}$ $P6_0$/$\overline{WAIT}$	Bus control signal input/output ($\overline{BACK}$, $\overline{BREQ}$, $\overline{WAIT}$) and 3-bit generic input/output					Generic input/ output	
Port 7	• 8-bit I/O port	$P7_7$/AN_7/ DA_1 $P7_6$/AN_6/ DA_0	Analog input (AN_7, AN_6) to A/D converter, analog output (DA_1, DA_0) from D/A converter, and generic input						
		$P7_5$ to $P7_0$/ AN_5 to AN_0	Analog input (AN_5 to AN_0) to A/D converter, and generic input						
Port 8	• 5-bit I/O port • $P8_2$ to $P8_0$ have Schmitt inputs	$P8_4$/$\overline{CS_0}$	DDR = 0: generic input DDR = 1 (reset value): $\overline{CS_0}$ output				DDR = 0 (after reset): generic input DDR = 1: $\overline{CS_0}$ output	Generic input/ output	
		$P8_3$/$\overline{IRQ_3}$/ $\overline{CS_1}$/$\overline{ADTRG}$	$\overline{IRQ_3}$ input, $\overline{CS_1}$ output, external trigger input ($\overline{ADTRG}$) to A/D converter, and generic input DDR = 0 (after reset): generic input DDR = 1: $\overline{CS_1}$ output					$\overline{IRQ_3}$ input, external trigger input ($\overline{ADTRG}$) to A/D converter, and generic input/output	

7.3.2 Register Descriptions

Table 7.3 summarizes the registers of port 2.

Table 7.3 Port 2 Registers

Address *	Name	Abbreviation	R/W	Initial Value	
				Modes 1 to 4	Modes 5 to 7
H'EE001	Port 2 data direction register	P2DDR	W	H'FF	H'00
H'FFFD1	Port 2 data register	P2DR	R/W	H'00	H'00
H'EE03C	Port 2 input pull-up MOS control register	P2PCR	R/W	H'00	H'00

Note: * Lower 20 bits of the address in advanced mode.

Port 2 Data Direction Register (P2DDR):P2DDR is an 8-bit write-only register that can select input or output for each pin in port 2.

Bit		7	6	5	4	3	2	1	0
		$P2_7DDR$	$P2_6DDR$	$P2_5DDR$	$P2_4DDR$	$P2_3DDR$	$P2_2DDR$	$P2_1DDR$	$P2_0DDR$
Modes 1 to 4	Initial value	1	1	1	1	1	1	1	1
	Read/Write	—	—	—	—	—	—	—	—
Modes 5 to 7	Initial value	0	0	0	0	0	0	0	0
	Read/Write	W	W	W	W	W	W	W	W

Port 2 data direction 7 to 0
These bits select input or output for port 2 pins

Modes 1 to 4 (Expanded Modes with On-Chip ROM Disabled)

P2DDR values are fixed at 1. Port 2 functions as an address bus.

Mode 5 (Expanded Modes with On-Chip ROM Enabled)

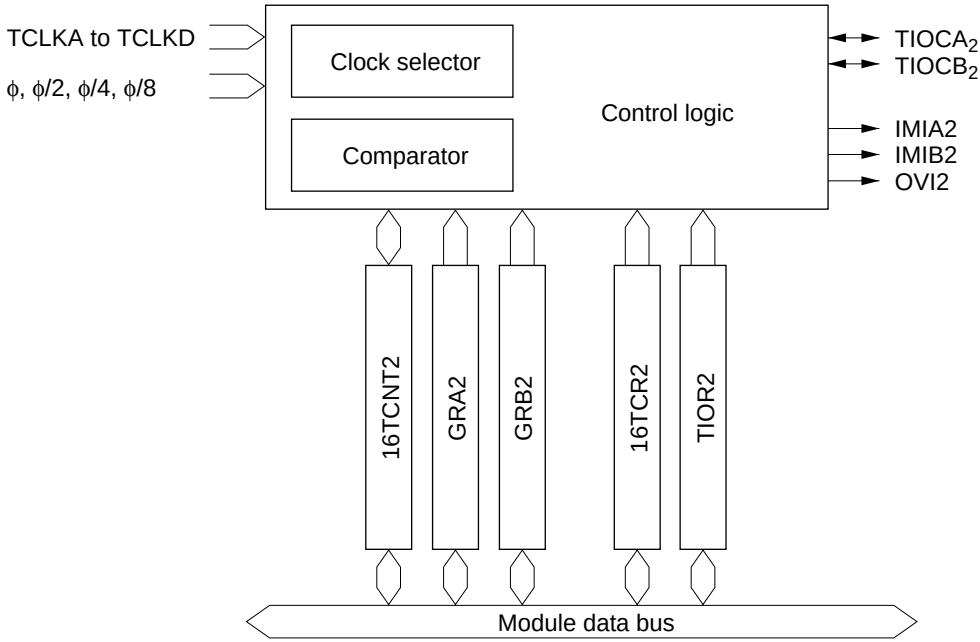
Following a reset, port 2 is an input port. A pin in port 2 becomes an address output pin if the corresponding P2DDR bit is set to 1, and a generic input port if this bit is cleared to 0.

Modes 6 and 7 (Single-Chip Mode)

Port 2 functions as an input/output port. A pin in port 2 becomes an output port if the corresponding P2DDR bit is set to 1, and an input port if this bit is cleared to 0.

In modes 1 to 4, P2DDR bits are always read as 1, and cannot be modified.

Block Diagram of Channel 2: Figure 8.3 is a block diagram of channel 2



Legend:

16TCNT2:	Timer counter 2 (16 bits)
GRA2, GRB2:	General registers A2 and B2 (input capture/output compare registers) (16 bits × 2)
TCR2:	Timer control register 2 (8 bits)
TIOR2:	Timer I/O control register 2 (8 bits)

Figure 8.3 Block Diagram of Channel 2

Bit 1—Input Capture/Compare Match Flag B1 (IMFB1): This status flag indicates GRB1 compare match or input capture events.

Bit 1 IMFB1	Description
0	[Clearing condition] (Initial value) Read IMFB1 flag when IMFB1 =1, then write 0 in IMFB1 flag
1	[Setting conditions] • 16TCNT1 = GRB1 when GRB1 functions as an output compare register • 16TCNT1 value is transferred to GRB1 by an input capture signal when GRB1 functions as an input capture register

Bit 0—Input Capture/Compare Match Flag B0 (IMFB0): This status flag indicates GRB0 compare match or input capture events.

Bit 0 IMFB0	Description
0	[Clearing condition] (Initial value) Read IMFB0 flag when IMFB0 =1, then write 0 in IMFB0 flag
1	[Setting conditions] • 16TCNT0 = GRB0 when GRB0 functions as an output compare register • 16TCNT0 value is transferred to GRB0 by an input capture signal when GRB0 functions as an input capture register

8.4.6 16-Bit Timer Output Timing

The initial value of 16-bit timer output when a timer count operation begins can be specified arbitrarily by making a setting in TOLR.

Figure 8.32 shows the timing for setting the initial value with TOLR.

Only write to TOLR when the corresponding bit in TSTR is cleared to 0.

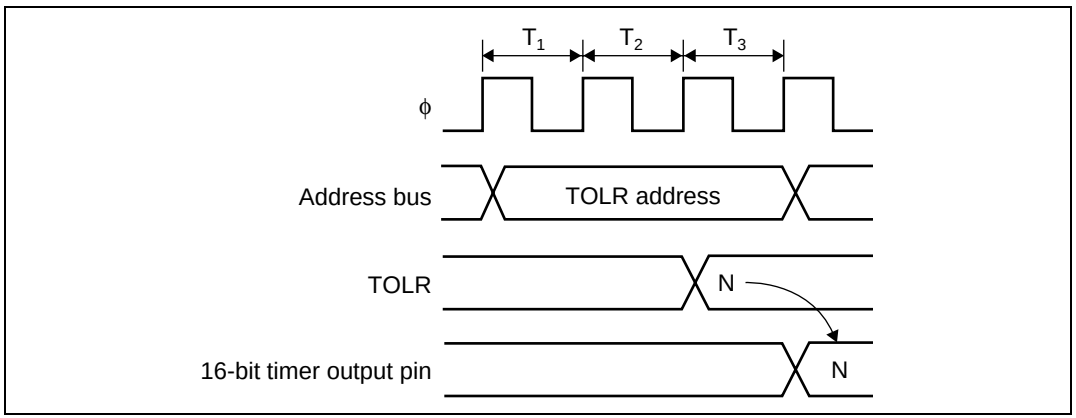
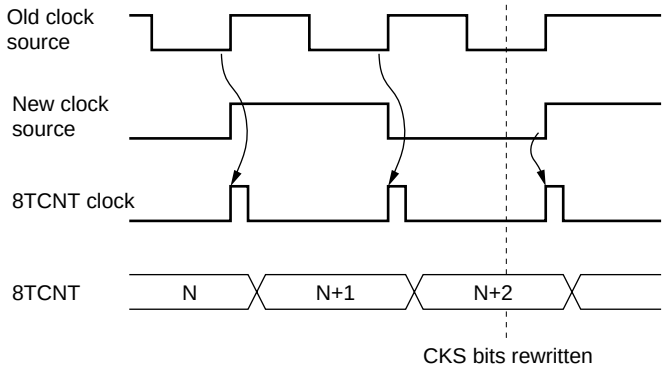


Figure 8.32 Timing for Setting 16-Bit Timer Output Level by Writing to TOLR

No.	CKS1 and CKS0 Write Timing	8TCNT Operation
4	Low → low switchover*4	

Notes: 1. Including switchovers from the high level to the halted state, and from the halted state to the high level.
2. Including switchover from the halted state to the low level.
3. Including switchover from the low level to the halted state.
4. The switchover is regarded as a rising edge, causing 8TCNT to increment.

12.2.3 Transmit Shift Register (TSR)

TSR is the register that transmits serial data.

Bit	7	6	5	4	3	2	1	0
Read/Write	—	—	—	—	—	—	—	—

The SCI loads transmit data from TDR to TSR, then transmits the data serially from the TxD pin, LSB (bit 0) first. After transmitting one data byte, the SCI automatically loads the next transmit data from TDR into TSR and starts transmitting it. If the TDRE flag is set to 1 in SSR, however, the SCI does not load the TDR contents into TSR. The CPU cannot read or write RSR directly.

12.2.4 Transmit Data Register (TDR)

TDR is an 8-bit register that stores data for serial transmission.

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

When the SCI detects that TSR is empty, it moves transmit data written in TDR from TDR into TSR and starts serial transmission. Continuous serial transmission is possible by writing the next transmit data in TDR during serial transmission from TSR.

The CPU can always read and write TDR. TDR is initialized to H'FF by a reset and in standby mode.

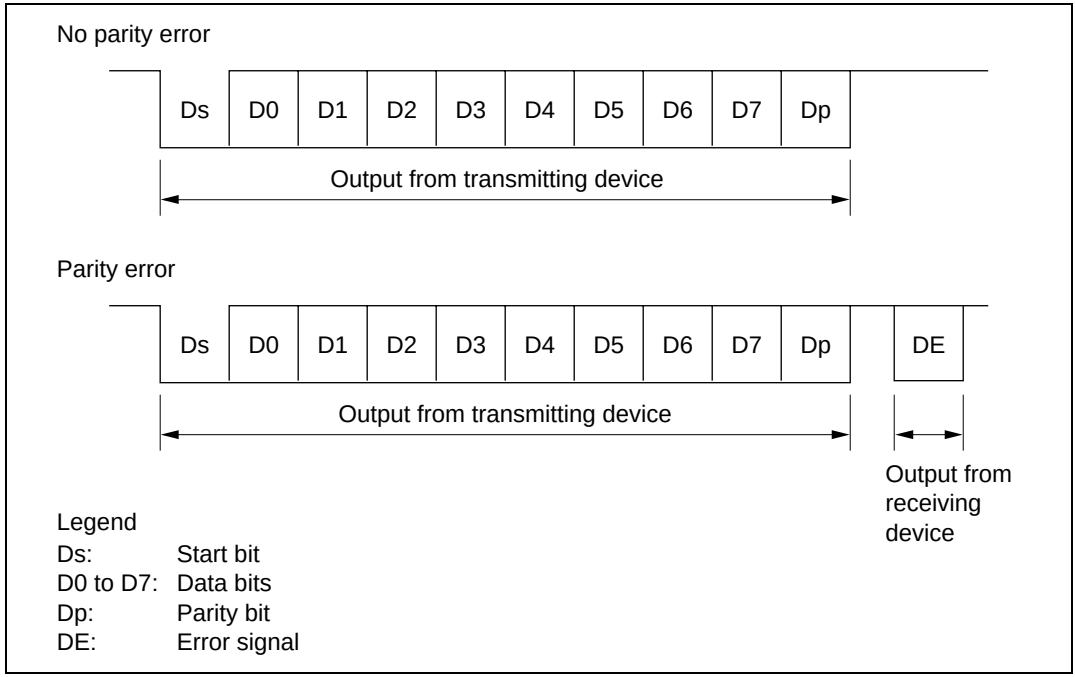


Figure 13.3 Smart Card Interface Data Format

The operating sequence is as follows.

1. When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.
2. The transmitting device starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).
3. With the smart card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.
4. The receiving device carries out a parity check. If there is no parity error and the data is received normally, the receiving device waits for reception of the next data. If a parity error occurs, however, the receiving device outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving device places the signal line in the high-impedance state again. The signal line is pulled high again by a pull-up resistor.
5. If the transmitting device does not receive an error signal, it proceeds to transmit the next data frame. If it receives an error signal, however, it returns to step 2 and transmits the same data again.

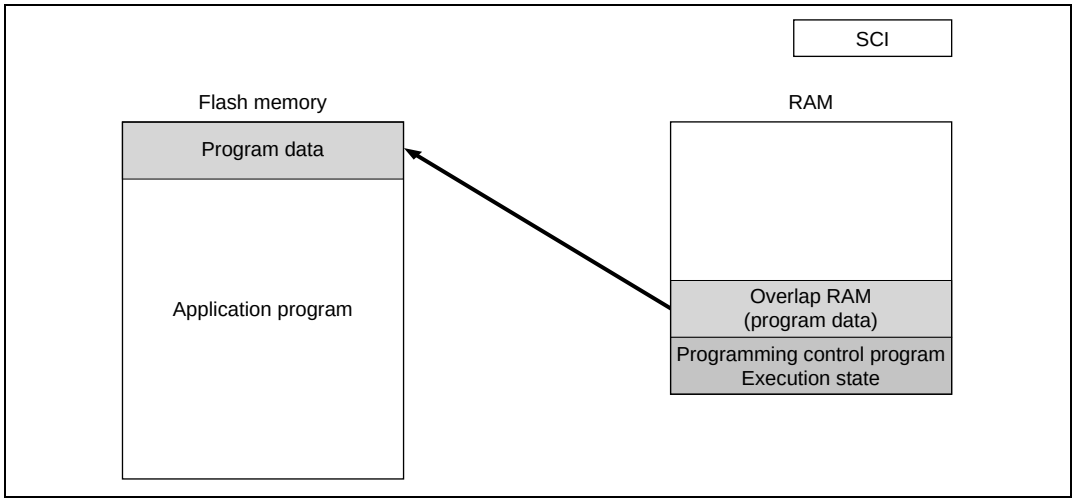
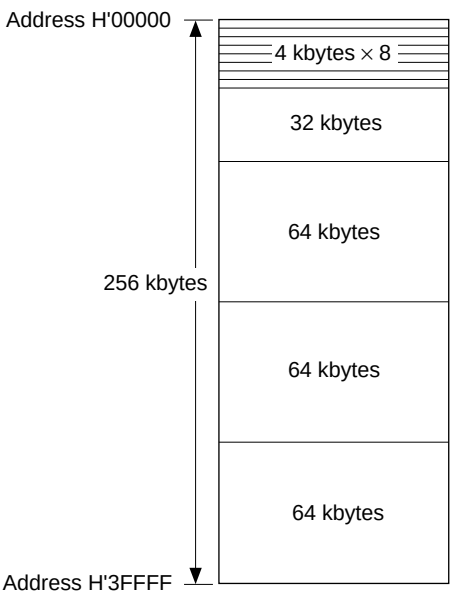


Figure 17.4 Writing Overlap RAM Data in User Program Mode

17.4.4 Block Configuration

The flash memory in the H8/3026F-ZTAT version is divided into three 64-kbyte blocks, one 32-kbyte block, and eight 4-kbyte blocks. Erasing can be carried out in block units.



Example of Flash Memory Block Area EB0 Overlapping

1. Set bits RAMS and RAM2 to RAM0 in RAMCR to 1,0, 0, 0, to overlap part of RAM onto the area (EB0) for which realtime programming is required.
2. Realtime programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P or E bit in FLMCR1 will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.
4. As in on-board programming mode, care is required when applying and releasing FWE to prevent erroneous programming or erasing. To prevent erroneous programming and erasing due to program runaway during FWE application, in particular, the watchdog timer should be set when the PSU, P, ESU, or E bit is set to 1 in FLMCR1, even while the emulation function is being used.
5. When the emulation function is used, NMI input is prohibited when the P bit or E bit is set to 1 in FLMCR1, in the same way as with normal programming and erasing.

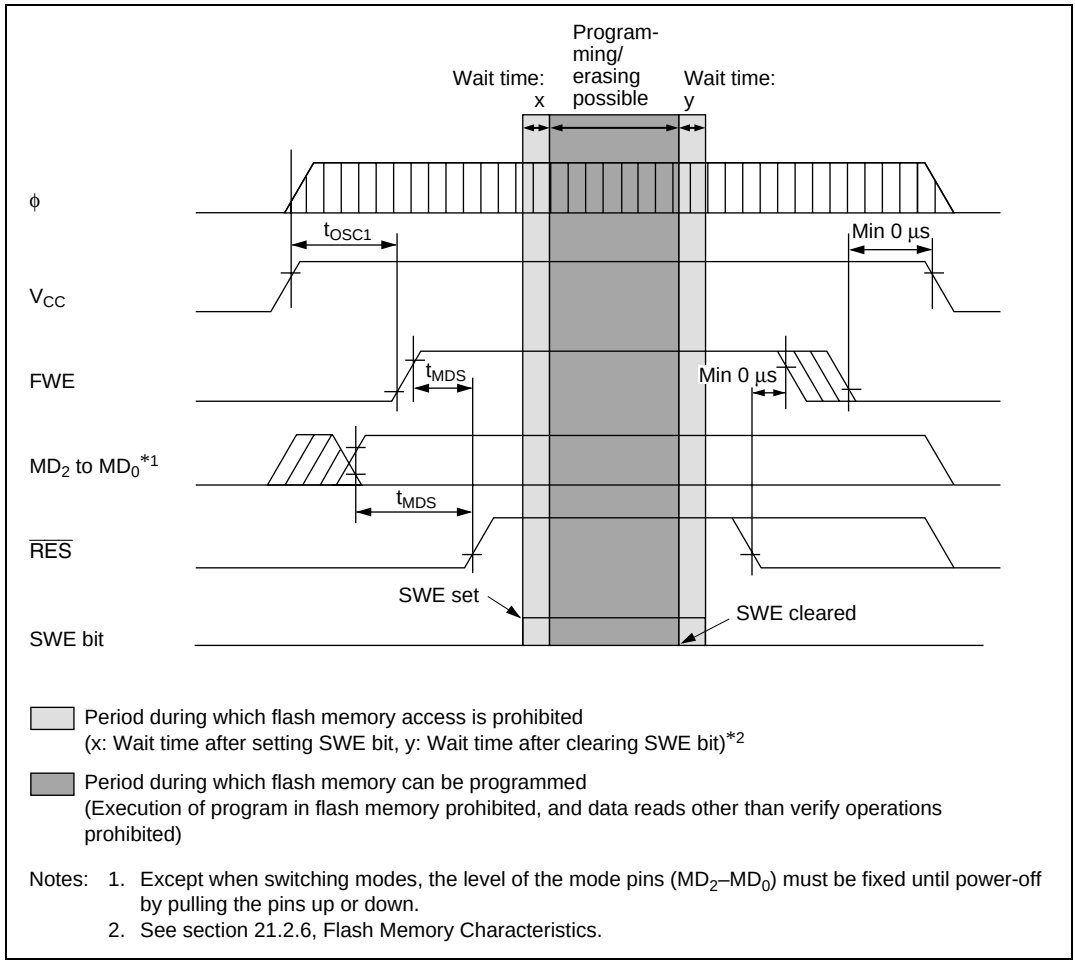
   The P and E bits are cleared by a reset (including a watchdog timer reset), in standby mode, when a high level is not being input to the FWE pin, or when the SWE bit in FLMCR1 is 0 while a high level is being input to the FWE pin.

10. Do not touch the socket adapter or chip during programming.

Touching either of these can cause contact faults and write errors.

11. A wait time of 100 µs or more is necessary when performing a read after a transition to normal mode from program, erase, or verify mode.

12. Use byte access on the registers that control the flash memory (FLMCR1, FLMCR2, EBR1, EBR2, and RAMCR).



Notes: 1. Except when switching modes, the level of the mode pins (MD_2–MD_0) must be fixed until power-off by pulling the pins up or down.
2. See section 21.2.6, Flash Memory Characteristics.

Figure 17.16 Power-On/Off Timing (Boot Mode)

17.14 Notes when Converting the F-ZTAT Application Software to the Mask ROM Versions

Please note the following when converting the F-ZTAT application software to the mask ROM versions.

The values read from the internal registers for the flash ROM or the mask ROM version and F-ZTAT version differ as follows.

			Status	
Register	**Bit**	**Value**	**F-ZTAT Version**	**Mask ROM Version**
FLMCR	FWE	0	Application software running	— (Is not read out)
		1	Programming	Application software running (Always read as 1)

Note: This difference applies to all the F-ZTAT versions and all the mask ROM versions that have different ROM size.

Notes on Use of Emulation in RAM:

1. Flash write enable (FWE) application and releasing

   As in on-board program mode, care is required when applying and releasing FWE to prevent erroneous programming or erasing. To prevent erroneous programming and erasing due to program runaway during FWE application, in particular, the watchdog timer should be set when the PSU, P, ESU, or E bit is set to 1 in FLMCR1, even while the emulation function is being used. For details, see section 18.11, Flash Memory Programming and Erasing Precautions.
2. NMI input disabling conditions

   When the emulation function is used, NMI input is disabled when the P bit or E bit is set to 1 in FLMCR1, in the same way as with normal programming and erasing.

   The P and E bits are cleared by a reset (including a watchdog timer reset), in standby mode, when a high level is not being input to the FWE pin, or when the SWE bit in FLMCR1 is 0 while a high level is being input to the FWE pin.
3. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P or E bit in FLMCR1 will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.
4. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
5. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

18.9 NMI Input Disabling Conditions

All interrupts, including NMI input, should be disabled while flash memory is being programmed or erased (while the P bit or E bit is set in FLMCR1), and while the boot program is executing in boot mode*1, to give priority to the program or erase operation. There are three reasons for this:

1. NMI input during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the NMI exception handling sequence during programming or erasing, the vector would not be read correctly*2, possibly resulting in MCU runaway.
3. If NMI input occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

Table 21.15 Bus Timing

Conditions: V_{CC} = 3.0 to 3.6 V, AV_{CC} = 3.0 to 3.6 V, V_{REF} = 3.0 to AV_{CC}, V_{SS} = AV_{SS} = 0 V, 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
Address delay time	t_{AD}	—	25	ns	Figure 21.11, figure 21.12
Address hold time	t_{AH}	0.5 t_{cyc} – 20	—	ns	
Read strobe delay time	t_{RSD}	—	25	ns	
Address strobe delay time	t_{ASD}	—	25	ns	
Write strobe delay time	t_{WSD}	—	25	ns	
Strobe delay time	t_{SD}	—	25	ns	
Write strobe pulse width 1	t_{WSW1}	1.0 t_{cyc} – 25	—	ns	
Write strobe pulse width 2	t_{WSW2}	1.5 t_{cyc} – 25	—	ns	
Address setup time 1	t_{AS1}	0.5 t_{cyc} – 20	—	ns	
Address setup time 2	t_{AS2}	1.0 t_{cyc} – 20	—	ns	
Read data setup time	t_{RDS}	40	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Write data delay time	t_{WDD}	—	35	ns	
Write data setup time 1	t_{WDS1}	1.0 t_{cyc} – 30	—	ns	
Write data setup time 2	t_{WDS2}	2.0 t_{cyc} – 30	—	ns	
Write data hold time	t_{WDH}	0.5 t_{cyc} – 15	—	ns	
Read data access time 1	t_{ACC1}	—	2.0 t_{cyc} – 45	ns	
Read data access time 2	t_{ACC2}	—	3.0 t_{cyc} – 45	ns	
Read data access time 3	t_{ACC3}	—	1.5 t_{cyc} – 45	ns	
Read data access time 4	t_{ACC4}	—	2.5 t_{cyc} – 45	ns	
Precharge time 1	t_{PCH1}	1.0 t_{cyc} – 20	—	ns	
Precharge time 2	t_{PCH2}	0.5 t_{cyc} – 20	—	ns	
Wait setup time	t_{WTS}	25	—	ns	Figure 21.13
Wait hold time	t_{WTH}	5	—	ns	
Bus request setup time	t_{BRQS}	25	—	ns	Figure 21.14
Bus acknowledge delay time 1	t_{BACD1}	—	30	ns	
Bus acknowledge delay time 2	t_{BACD2}	—	30	ns	
Bus-floating time	t_{BZD}	—	30	ns	

Note: In order to secure the address hold time relative to the rise of the $\overline{RD}$ strobe, address update mode 2 should be used. For details see section 6.3.5, Address Output Method.

Mnemonic	Operand Size	Addressing Mode and Instruction Length (bytes)									Operation	Condition Code						No. of States*1	
		#xx	Rn	@ERn	@(d, ERn)	@–ERn/@ERn+	@aa	@(d, PC)	@@aa	|		I	H	N	Z	V	C	Normal	Advanced
DIVXU. W Rs, ERd	W		2								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	(6)	(7)	—	—	22	
DIVXS. B Rs, Rd	B		4								Rd16 ÷ Rs8 → Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	(8)	(7)	—	—	16	
DIVXS. W Rs, ERd	W		4								ERd32 ÷ Rs16 → ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	(8)	(7)	—	—	24	
CMP.B #xx:8, Rd	B	2									Rd8–#xx:8	—	↕	↕	↕	↕	↕	2	
CMP.B Rs, Rd	B		2								Rd8–Rs8	—	↕	↕	↕	↕	↕	2	
CMP.W #xx:16, Rd	W	4									Rd16–#xx:16	—	(1)	↕	↕	↕	↕	4	
CMP.W Rs, Rd	W		2								Rd16–Rs16	—	(1)	↕	↕	↕	↕	2	
CMP.L #xx:32, ERd	L	6									ERd32–#xx:32	—	(2)	↕	↕	↕	↕	6	
CMP.L ERs, ERd	L		2								ERd32–ERs32	—	(2)	↕	↕	↕	↕	2	
NEG.B Rd	B		2								0–Rd8 → Rd8	—	↕	↕	↕	↕	↕	2	
NEG.W Rd	W		2								0–Rd16 → Rd16	—	↕	↕	↕	↕	↕	2	
NEG.L ERd	L		2								0–ERd32 → ERd32	—	↕	↕	↕	↕	↕	2	
EXTU.W Rd	W		2								0 → (<bits 15 to 8> of Rd16)	—	—	0	↕	0	—	2	
EXTU.L ERd	L		2								0 → (<bits 31 to 16> of ERd32)	—	—	0	↕	0	—	2	
EXTS.W Rd	W		2								(<bit 7> of Rd16) → (<bits 15 to 8> of Rd16)	—	—	↕	↕	0	—	2	
EXTS.L ERd	L		2								(<bit 15> of ERd32) → (<bits 31 to 16> of ERd32)	—	—	↕	↕	0	—	2	

Table A.4 Number of Cycles per Instruction

Instruction	Mnemonic	Instruction Fetch I	Branch Addr. Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8, Rd	1					
	ADD.B Rs, Rd	1					
	ADD.W #xx:16, Rd	2					
	ADD.W Rs, Rd	1					
	ADD.L #xx:32, ERd	3					
	ADD.L ERs, ERd	1					
ADDS	ADDS #1/2/4, ERd	1					
ADDX	ADDX #xx:8, Rd	1					
	ADDX Rs, Rd	1					
AND	AND.B #xx:8, Rd	1					
	AND.B Rs, Rd	1					
	AND.W #xx:16, Rd	2					
	AND.W Rs, Rd	1					
	AND.L #xx:32, ERd	3					
	AND.L ERs, ERd	2					
ANDC	ANDC #xx:8, CCR	1					
BAND	BAND #xx:3, Rd	1					
	BAND #xx:3, @ERd	2			1		
	BAND #xx:3, @aa:8	2			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					
	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					

TCSR—Timer Control/Status Register H'FFF8C WDT

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Clock select 2 to 0

CKS2	CKS1	CKS0	Description
0	0	0	φ/2
		1	φ/32
	1	0	φ/64
		1	φ/128
1	0	0	φ/256
		1	φ/512
	1	0	φ/2048
		1	φ/4096

Timer enable

0	Timer disabled • TCNT is initialized to H'00 and halted
1	Timer enabled • TCNT starts counting up

Timer mode select

0	Interval timer: requests interval timer interrupts
1	Watchdog timer: generates a reset signal

Overflow flag

0	[Clearing condition] Read OVF when OVF = 1, then write 0 in OVF
1	[Setting condition] TCNT changes from H'FF to H'00

Note: * Only 0 can be written to clear the flag.

DADR1—D/A Data Register 1 H'FFF9D D/A

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

D/A conversion data

DACR—D/A Control Register H'FFF9E D/A

Bit	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	0	0	0	1	1	1	1	1
Read/Write	R/W	R/W	R/W	—	—	—	—	—

D/A enable

Bit 7	Bit 6	Bit 5	Description
DAOE1	DAOE0	DAE	
0	0	—	D/A conversion is disabled in channels 0 and 1
0	1	0	D/A conversion is enabled in channel 0 D/A conversion is disabled in channel 1
0	1	1	D/A conversion is enabled in channels 0 and 1
1	0	0	D/A conversion is disabled in channel 0 D/A conversion is enabled in channel 1
1	0	1	D/A conversion is enabled in channels 0 and 1
1	1	—	D/A conversion is enabled in channels 0 and 1

D/A output enable 0

0	DA_0 analog output is disabled
1	Channel-0 D/A conversion and DA_0 analog output are enabled

D/A output enable 1

0	DA_1 analog output is disabled
1	Channel-1 D/A conversion and DA_1 analog output are enabled