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"

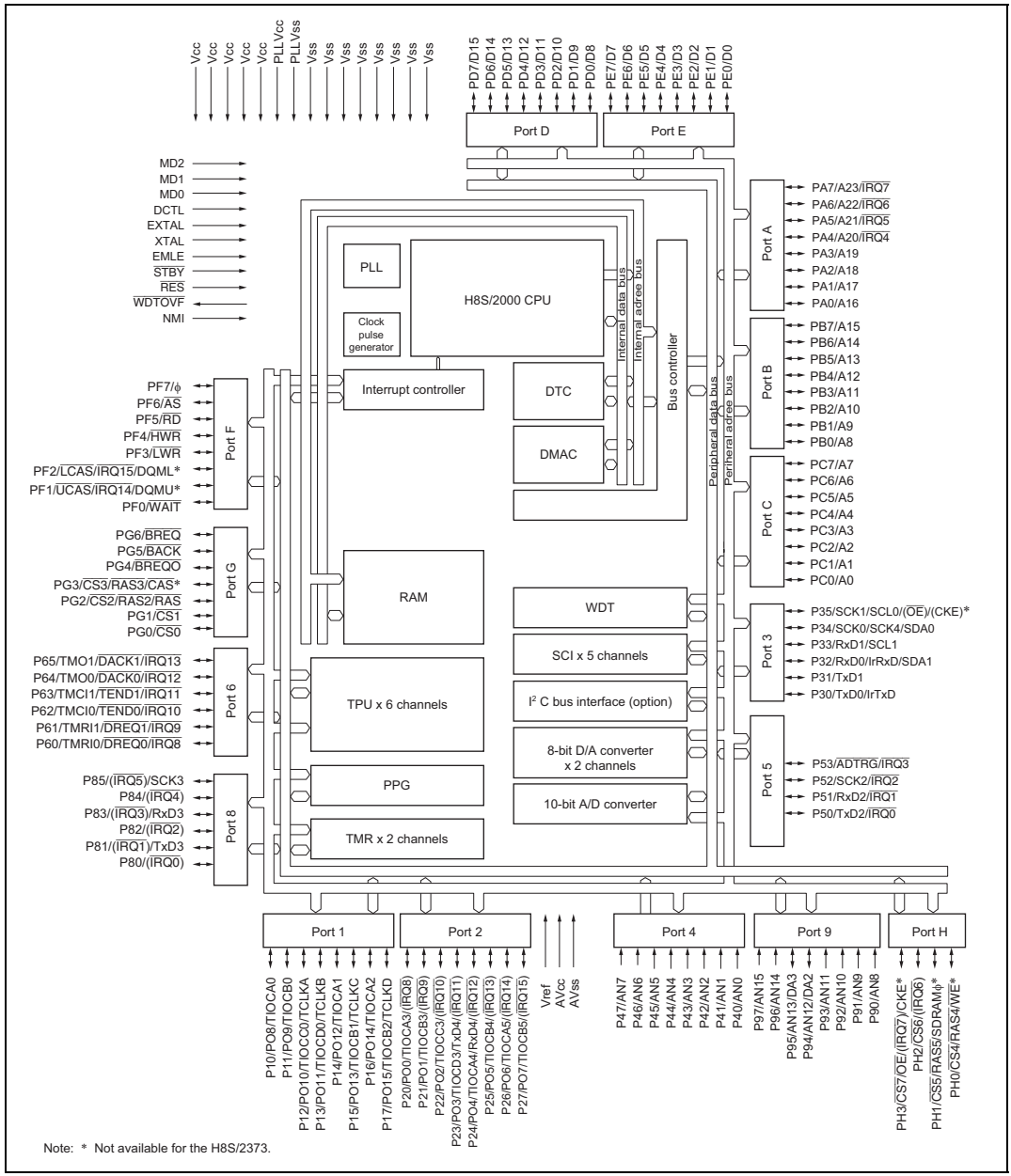
Details	
Product Status	Not For New Designs
Core Processor	H8S/2000
Core Size	16-Bit
Speed	34MHz
Connectivity	I²C, IrDA, SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	96
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 16x10b; D/A 6x8b
Oscillator Type	External
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2378rvfq34v

Figures

Section 1 Overview 1
Figure 1.1 Internal Block Diagram for H8S/2378 0.18µm F-ZTAT Group and H8S/2378R 0.18µm F-ZTAT Group 3
Figure 1.2 Internal Block Diagram for H8S/2377 and H8S/2377R 4
Figure 1.3 Internal Block Diagram for H8S/2375 and H8S/2375R 5
Figure 1.4 Internal Block Diagram for H8S/2373 and H8S/2373R 6
Figure 1.5 Pin Arrangement for H8S/2378 0.18µm F-ZTAT Group and H8S/2378R 0.18µm F-ZTAT Group 7
Figure 1.6 Pin Arrangement for H8S/2377 and H8S/2377R 8
Figure 1.7 Pin Arrangement for H8S/2375 and H8S/2375R 9
Figure 1.8 Pin Arrangement for H8S/2373 and H8S/2373R 10
Figure 1.9 Pin Arrangement (TLP-145V: Top View) 11

Section 2 CPU 35
Figure 2.1 Exception Vector Table (Normal Mode) 39
Figure 2.2 Stack Structure in Normal Mode 39
Figure 2.3 Exception Vector Table (Advanced Mode) 40
Figure 2.4 Stack Structure in Advanced Mode 41
Figure 2.5 Memory Map 42
Figure 2.6 CPU Internal Registers 43
Figure 2.7 Usage of General Registers 44
Figure 2.8 Stack 45
Figure 2.9 General Register Data Formats (1) 48
Figure 2.9 General Register Data Formats (2) 49
Figure 2.10 Memory Data Formats 50
Figure 2.11 Instruction Formats (Examples) 62
Figure 2.12 Branch Address Specification in Memory Indirect Addressing Mode 65
Figure 2.13 State Transitions 69

Section 3 MCU Operating Modes 71
Figure 3.1 Memory Map for H8S/2378 and H8S/2378R (1) 78
Figure 3.2 Memory Map for H8S/2378 and H8S/2378R (2) 79
Figure 3.3 Memory Map for H8S/2377 and H8S/2377R (1) 80
Figure 3.4 Memory Map for H8S/2377 and H8S/2377R (2) 81
Figure 3.5 Memory Map for H8S/2375 and H8S/2375R (1) 82
Figure 3.6 Memory Map for H8S/2375 and H8S/2375R (2) 83
Figure 3.7 Memory Map for H8S/2374 and H8S/2374R (1) 84
Figure 3.8 Memory Map for H8S/2374 and H8S/2374R (2) 85



Note: * Not available for the H8S/2373.

Figure 1.4 Internal Block Diagram for H8S/2373 and H8S/2373R

Table 5.2 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
External pin	NMI	7	H'001C	—	High	—	—
	IRQ0	16	H'0040	IPRA14 to IPRA12		○	—
	IRQ1	17	H'0044	IPRA10 to IPRA8		○	—
	IRQ2	18	H'0048	IPRA6 to IPRA4		○	—
	IRQ3	19	H'004C	IPRA2 to IPRA0		○	—
	IRQ4	20	H'0050	IPRB14 to IPRB12		○	—
	IRQ5	21	H'0054	IPRB10 to IPRB8		○	—
	IRQ6	22	H'0058	IPRB6 to IPRB4		○	—
	IRQ7	23	H'005C	IPRB2 to IPRB0		○	—
	IRQ8	24	H'0060	IPRC14 to IPRC12		○	—
	IRQ9	25	H'0064	IPRC10 to IPRC8		○	—
	IRQ10	26	H'0068	IPRC6 to IPRC4		○	—
	IRQ11	27	H'006C	IPRC2 to IPRC0		○	—
	IRQ12	28	H'0070	IPRD14 to IPRD12		○	—
	IRQ13	29	H'0074	IPRD10 to IPRD8		○	—
	IRQ14	30	H'0078	IPRD6 to IPRD4		○	—
	IRQ15	31	H'007C	IPRD2 to IPRD0		○	—
DTC	SWDTEND	32	H'0080	IPRE14 to IPRE12		○	—
WDT	WOVI	33	H'0084	IPRE10 to IPRE8		—	—
—	Reserved for system use	34	H'0088	IPRE6 to IPRE4		—	—
Refresh controller	CMI	35	H'008C	IPRE2 to IPRE0		—	—
—	Reserved for system use	36	H'0090	IPRF14 to IPRF12		—	—
		37	H'0094			—	—
A/D	ADI	38	H'0098	IPRF10 to IPRF8		○	○
	Reserved for system use	39	H'009C		Low	—	—

5.7 Usage Notes

5.7.1 Conflict between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to mask interrupts, the masking becomes effective after execution of the instruction.

When an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored. The same also applies when an interrupt source flag is cleared to 0. Figure 5.6 shows an example in which the TCIEV bit in the TPU's TIER_0 register is cleared to 0. The above conflict will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

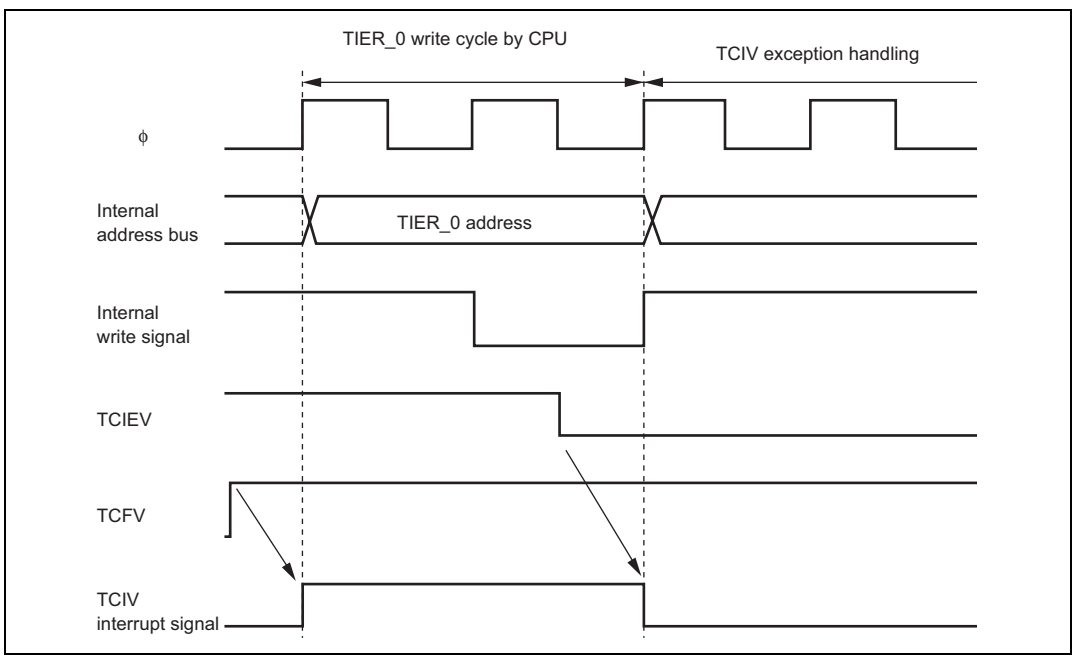


Figure 5.6 Conflict between Interrupt Generation and Disabling

- WTCRAL

Bit	Bit Name	Initial Value	R/W	Description
7	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
6 5 4	W52 W51 W50	1 1 1	R/W R/W R/W	Area 5 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 5 while AST5 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted
3	—	0	R	Reserved This bit is always read as 0 and cannot be modified.
2 1 0	W42 W41 W40	1 1 1	R/W R/W R/W	Area 4 Wait Control 2 to 0 These bits select the number of program wait states when accessing area 4 while AST4 bit in ASTCR = 1. 000: Program wait not inserted 001: 1 program wait state inserted 010: 2 program wait states inserted 011: 3 program wait states inserted 100: 4 program wait states inserted 101: 5 program wait states inserted 110: 6 program wait states inserted 111: 7 program wait states inserted

Figure 6.20 shows an example of the timing when the $\overline{CS}$ assertion period is extended in basic bus 3-state access space.

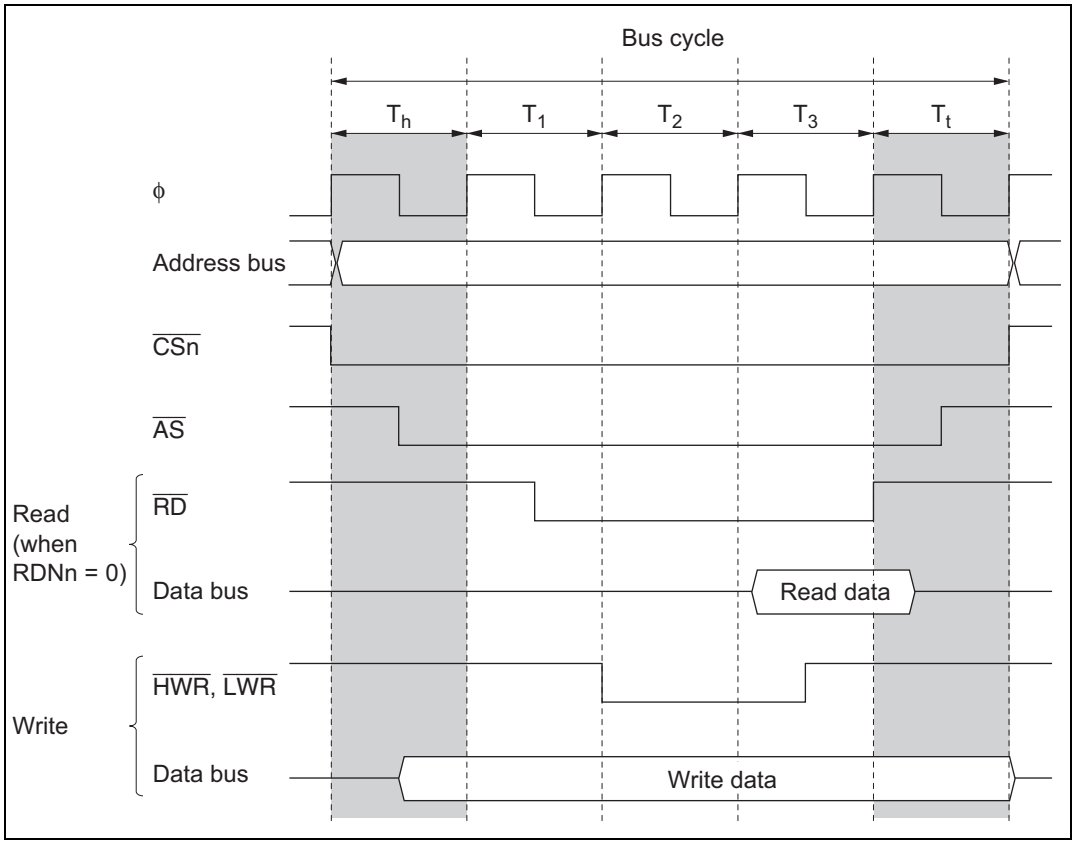


Figure 6.20 Example of Timing when Chip Select Assertion Period Is Extended

Both extension state T_h inserted before the basic bus cycle and extension state T_t inserted after the basic bus cycle, or only one of these, can be specified for individual areas. Insertion or non-insertion can be specified for the T_h state with the upper 8 bits (CSXH7 to CSXH0) in the CSACR register, and for the T_t state with the lower 8 bits (CSXT7 to CSXT0).

space to be output from a dedicated $\overline{OE}$ pin. In this case, the $\overline{OE}$ signal for DRAM space is output from both the $\overline{RD}$ pin and the ($\overline{OE}$) pin, but in external read cycles for other than DRAM space, the signal is output only from the $\overline{RD}$ pin.

6.6.6 Column Address Output Cycle Control

The column address output cycle can be changed from 2 states to 3 states by setting the CAST bit to 1 in DRAMCR. Use the setting that gives the optimum specification values ($\overline{CAS}$ pulse width, etc.) according to the DRAM connected and the operating frequency of this LSI. Figure 6.22 shows an example of the timing when a 3-state column address output cycle is selected.

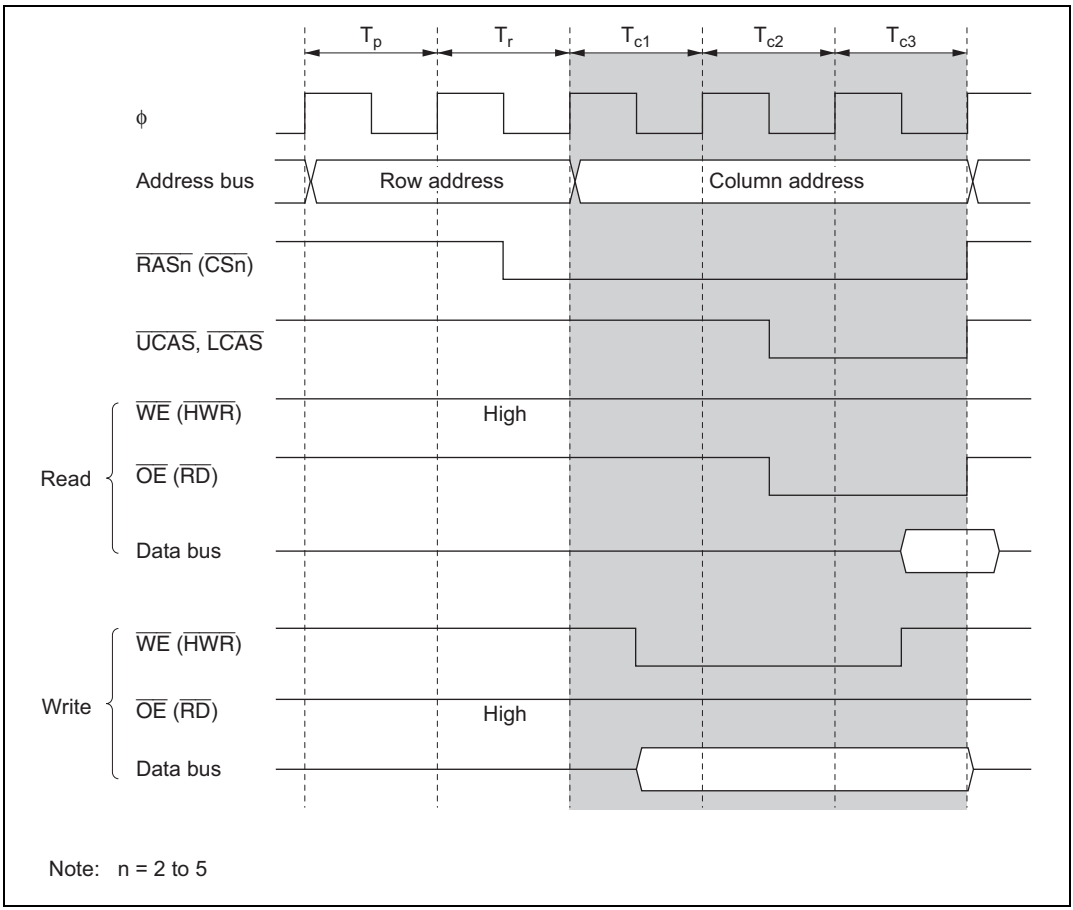


Figure 6.22 Example of Access Timing with 3-State Column Address Output Cycle (RAST = 0)

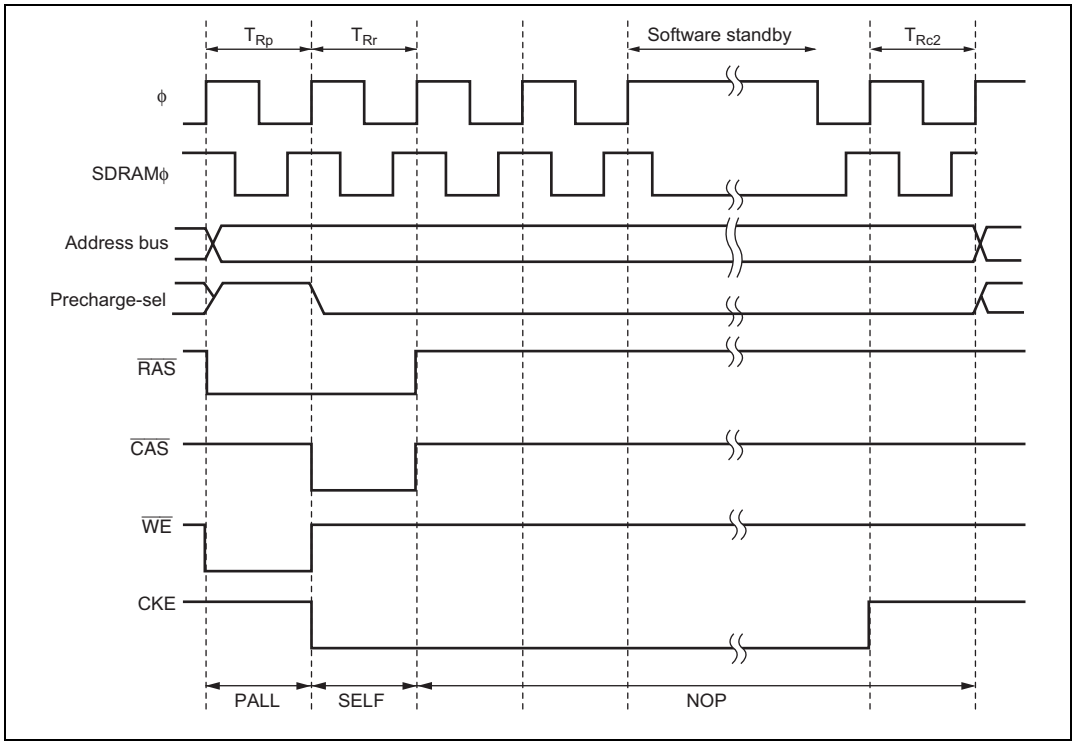


Figure 6.57 Self-Refresh Timing
(TPC1 = 1, TPC0 = 0, RCW1 = 0, RCW0 = 0, RLW1 = 0, RLW0 = 0)

In some synchronous DRAMs provided with a self-refresh mode, the interval between clearing self-refreshing and the next command is specified. A setting can be made in bits TPCS2 to TPCS0 in REFCR to make the precharge time after self-refreshing from 1 to 7 states longer than the normal precharge time. In this case, too, normal precharging is performed according to the setting of bits TPC1 and TPC0 in DRACCR, and therefore a setting should be made to give the optimum post-self-refresh precharge time, including this time. Figure 6.58 shows an example of the timing when the precharge time after self-refreshing is extended by 2 states.

6.12.2 Bus Transfer Timing

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific timings at which each bus master can relinquish the bus.

CPU: The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, DMAC, or EXDMAC*, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the component operations.
- With bit manipulation instructions such as BSET and BCLR, the sequence of operations is: data read (read), relevant bit manipulation operation (modify), write-back (write). The bus is not transferred during this read-modify-write cycle, which is executed as a series of bus cycles.
- If the CPU is in sleep mode, the bus is transferred immediately.

Note: * Not supported by the H8S/2375, H8S/2375R, H8S/2373, and H8S/2373R.

DTC: The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).

DMAC: The DMAC sends the bus arbiter a request for the bus when an activation request is generated.

In the case of an external request in short address mode or normal mode, and in cycle steal mode, the DMAC releases the bus after a single transfer.

In block transfer mode, it releases the bus after transfer of one block, and in burst mode, after completion of the transfer. However, in the event of an EXDMAC or external bus release request, which have a higher priority than the DMAC, the bus may be transferred to the bus master even if block or burst transfer is in progress.

7.4.2 Activation by External Request

If an external request ($\overline{\text{DREQ}}$ pin) is specified as a DMAC activation source, the relevant port should be set to input mode in advance*. Level sensing or edge sensing can be used for external requests.

External request operation in normal mode of short address mode or full address mode is described below.

When edge sensing is selected, a byte or word is transferred each time a high-to-low transition is detected on the $\overline{\text{DREQ}}$ pin. The next data transfer may not be performed if the next edge is input before data transfer is completed.

When level sensing is selected, the DMAC stands by for a transfer request while the $\overline{\text{DREQ}}$ pin is held high. While the $\overline{\text{DREQ}}$ pin is held low, transfers continue in succession, with the bus being released each time a byte or word is transferred. If the $\overline{\text{DREQ}}$ pin goes high in the middle of a transfer, the transfer is interrupted and the DMAC stands by for a transfer request.

Note: * If the relevant port is set as an output pin for another function, DMA transfers using the channel in question cannot be guaranteed.

7.4.3 Activation by Auto-Request

Auto-request is activated by register setting only, and transfer continues to the end. With auto-request activation, cycle steal mode or burst mode can be selected.

In cycle steal mode, the DMAC releases the bus to another bus master each time a byte or word is transferred. DMA and CPU cycles are usually repeated alternately. In burst mode, the DMAC keeps possession of the bus until the end of the transfer so that transfer is performed continuously.

7.5 Operation

7.5.1 Transfer Modes

Table 7.4 lists the DMAC transfer modes.

Port	Description	Mode 1*3	Mode 2*3	Mode 4	Mode 7		Input/ Output Type
					EXPE = 1	EXPE = 0	
Port A	General I/O port also functioning as address outputs	PA7/A23/$\overline{\text{IRQ7}}$		PA7/A23/$\overline{\text{IRQ7}}$		PA7/$\overline{\text{IRQ7}}$	Only PA4 to PA7 are Schmitt-triggered input when used as IRQ input. Built-in input pull-up MOS Open-drain output capability
		PA6/A22/$\overline{\text{IRQ6}}$		PA6/A22/$\overline{\text{IRQ6}}$		PA6/$\overline{\text{IRQ6}}$	
		PA5/A21/$\overline{\text{IRQ5}}$		PA5/A21/$\overline{\text{IRQ5}}$		PA5/$\overline{\text{IRQ5}}$	
		A20/$\overline{\text{IRQ4}}$		PA4/A20/$\overline{\text{IRQ4}}$		PA4/$\overline{\text{IRQ4}}$	
		A19		PA3/A19		PA3	
		A18		PA2/A18		PA2	
		A17		PA1/A17		PA1	
		A16		PA0/A16		PA0	
Port B	General I/O port also functioning as address outputs	A15		PB7/A15		PB7	Built-in input pull-up MOS
		A14		PB6/A14		PB6	
		A13		PB5/A13		PB5	
		A12		PB4/A12		PB4	
		A11		PB3/A11		PB3	
		A10		PB2/A10		PB2	
		A9		PB1/A9		PB1	
		A8		PB0/A8		PB0	
Port C	General I/O port also functioning as address outputs	A7		PC7/A7		PC7	Built-in input pull-up MOS
		A6		PC6/A6		PC6	
		A5		PC5/A5		PC5	
		A4		PC4/A4		PC4	
		A3		PC3/A3		PC3	
		A2		PC2/A2		PC2	
		A1		PC1/A1		PC1	
		A0		PC0/A0		PC0	

11.3 Register Descriptions

The TPU has the following registers in each channel.

- Timer control register_0 (TCR_0)
- Timer mode register_0 (TMDR_0)
- Timer I/O control register H_0 (TIORH_0)
- Timer I/O control register L_0 (TIORL_0)
- Timer interrupt enable register_0 (TIER_0)
- Timer status register_0 (TSR_0)
- Timer counter_0 (TCNT_0)
- Timer general register A_0 (TGRA_0)
- Timer general register B_0 (TGRB_0)
- Timer general register C_0 (TGRC_0)
- Timer general register D_0 (TGRD_0)
- Timer control register_1 (TCR_1)
- Timer mode register_1 (TMDR_1)
- Timer I/O control register _1 (TIOR_1)
- Timer interrupt enable register_1 (TIER_1)
- Timer status register_1 (TSR_1)
- Timer counter_1 (TCNT_1)
- Timer general register A_1 (TGRA_1)
- Timer general register B_1 (TGRB_1)
- Timer control register_2 (TCR_2)
- Timer mode register_2 (TMDR_2)
- Timer I/O control register_2 (TIOR_2)
- Timer interrupt enable register_2 (TIER_2)
- Timer status register_2 (TSR_2)
- Timer counter_2 (TCNT_2)
- Timer general register A_2 (TGRA_2)
- Timer general register B_2 (TGRB_2)
- Timer control register_3 (TCR_3)
- Timer mode register_3 (TMDR_3)
- Timer I/O control register H_3 (TIORH_3)
- Timer I/O control register L_3 (TIORL_3)

Table 11.15 TIOR_2

				Description	
Bit 7 IOB3	**Bit 6 IOB2**	**Bit 5 IOB1**	**Bit 4 IOB0**	**TGRB_2 Function**	**TIOCB2 Pin Function**
0	0	0	0	Output compare register	Output disabled
			1		Initial output is 0 output 0 output at compare match
		1	0		Initial output is 0 output 1 output at compare match
			1		Initial output is 0 output Toggle output at compare match
	1	0	0		Output disabled
			1		Initial output is 1 output 0 output at compare match
		1	0		Initial output is 1 output 1 output at compare match
			1		Initial output is 1 output Toggle output at compare match
1	×	0	0	Input capture register	Capture input source is TIOCB2 pin Input capture at rising edge
			1		Capture input source is TIOCB2 pin Input capture at falling edge
		1	×		Capture input source is TIOCB2 pin Input capture at both edges

Legend: ×: Don't care

Table 11.36 TPU Interrupts

Channel	Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
0	TGI0A	TGRA_0 input capture/compare match	TGFA_0	Possible	Possible
	TGI0B	TGRB_0 input capture/compare match	TGFB_0	Possible	Not possible
	TGI0C	TGRC_0 input capture/compare match	TGFC_0	Possible	Not possible
	TGI0D	TGRD_0 input capture/compare match	TGFD_0	Possible	Not possible
	TGI0E	TCNT_0 overflow	TCFV_0	Not possible	Not possible
1	TGI1A	TGRA_1 input capture/compare match	TGFA_1	Possible	Possible
	TGI1B	TGRB_1 input capture/compare match	TGFB_1	Possible	Not possible
	TCI1V	TCNT_1 overflow	TCFV_1	Not possible	Not possible
	TCI1U	TCNT_1 underflow	TCFU_1	Not possible	Not possible
2	TGI2A	TGRA_2 input capture/compare match	TGFA_2	Possible	Possible
	TGI2B	TGRB_2 input capture/compare match	TGFB_2	Possible	Not possible
	TCI2V	TCNT_2 overflow	TCFV_2	Not possible	Not possible
	TCI2U	TCNT_2 underflow	TCFU_2	Not possible	Not possible
3	TGI3A	TGRA_3 input capture/compare match	TGFA_3	Possible	Possible
	TGI3B	TGRB_3 input capture/compare match	TGFB_3	Possible	Not possible
	TGI3C	TGRC_3 input capture/compare match	TGFC_3	Possible	Not possible
	TGI3D	TGRD_3 input capture/compare match	TGFD_3	Possible	Not possible
	TCI3V	TCNT_3 overflow	TCFV_3	Not possible	Not possible
4	TGI4A	TGRA_4 input capture/compare match	TGFA_4	Possible	Possible
	TGI4B	TGRB_4 input capture/compare match	TGFB_4	Possible	Not possible
	TCI4V	TCNT_4 overflow	TCFV_4	Not possible	Not possible
	TCI4U	TCNT_4 underflow	TCFU_4	Not possible	Not possible
5	TGI5A	TGRA_5 input capture/compare match	TGFA_5	Possible	Possible
	TGI5B	TGRB_5 input capture/compare match	TGFB_5	Possible	Not possible
	TCI5V	TCNT_5 overflow	TCFV_5	Not possible	Not possible
	TCI5U	TCNT_5 underflow	TCFU_5	Not possible	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

12.4.1 Output Timing

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 12.3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

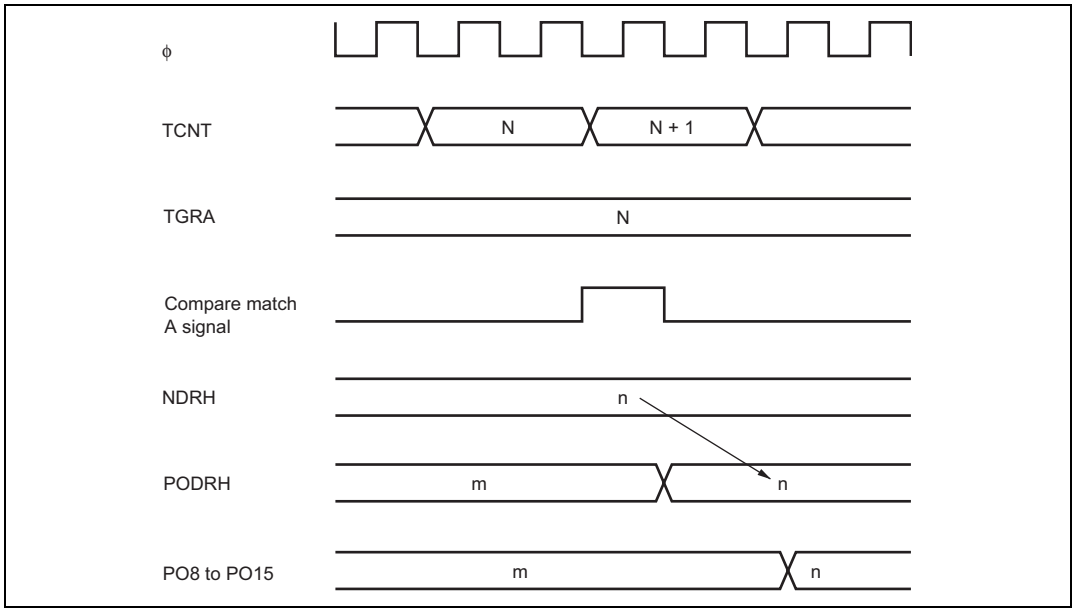


Figure 12.3 Timing of Transfer and Output of NDR Contents (Example)

Bit	Bit Name	Initial Value	R/W	Description
2	TEIE	0	R/W	Transmit End Interrupt Enable When this bit is set to 1, TEI interrupt request is enabled. TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or by clearing the TEIE bit to 0.
1	CKE1	0	R/W	Clock Enable 1 and 0
0	CKE0	0	R/W	Selects the clock source and SCK pin function. Asynchronous mode 00: On-chip baud rate generator SCK pin functions as I/O port 01: On-chip baud rate generator (Outputs a clock of the same frequency as the bit rate from the SCK pin.) 1×: External clock (Inputs a clock with a frequency 16 times the bit rate from the SCK pin.) Clocked synchronous mode 0×: Internal clock (SCK pin functions as clock output) 1×: External clock (SCK pin functions as clock input)

Legend: ×: Don't care

Table 15.9 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

φ (MHz)	Maximum Bit Rate (bit/s)	n	N	φ (MHz)	Maximum Bit Rate (bit/s)	n	N
10.00	13441	0	0	20.00	26882	0	0
10.7136	14400	0	0	25.00	33602	0	0
13.00	17473	0	0	30.00	40323	0	0
14.2848	19200	0	0	33.00	44355	0	0
16.00	21505	0	0	34.00*1	45699	0	0
18.00	24194	0	0	35.00*2	47043	0	0

Notes: 1. Supported on the H8S/2378 0.18μm F-ZTAT Group and H8S/2378R 0.18μm F-ZTAT Group only.

2. Supported on the H8S/2378 only.

3. When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. Prepare a programming control program in accordance with the description in section 20.7, Flash Memory Programming/Erasing.
4. Before branching to the programming control program, the chip terminates transfer operations by the SCI_1 (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. Therefore, the programming control program can still use it for transfer of program data or verify data with the host. The TxD pin is high. The contents of the CPU general registers are undefined immediately after branching to the programming control program. These registers must be initialized at the beginning of the programming control program, since the stack pointer (SP), in particular, is used implicitly in subroutine calls, etc.
5. In boot mode, if flash memory contains data (all data is not 1), all blocks of flash memory are erased. Boot mode is used for the initial programming in the on-board state or for a forcible return when a program that is to be initiated in user program mode was accidentally erased and could not be executed in user program mode.

Notes: 1. In boot mode, a part of the on-chip RAM area (H'FF8000 to H'FF87FF) is used by the boot program. Addresses H'FF8800 to H'FFBFFF is the area to which the programming control program is transferred from the host. The boot program area cannot be used until the execution state in boot mode switches to the programming control program.
2. Boot mode can be cleared by a reset. Release the reset by setting the MD pins, after waiting at least 20 states since driving the reset pin low. Boot mode is also cleared when the WDT overflow reset occurs.
3. Do not change the MD pin input levels in boot mode.
4. All interrupts are disabled during programming or erasing of the flash memory.

23.2 Oscillator

Clock pulses can be supplied by connecting a crystal resonator, or by input of an external clock.

23.2.1 Connecting a Crystal Resonator

A crystal resonator can be connected as shown in the example in figure 23.2. Select the damping resistance R_d according to table 23.1. An AT-cut parallel-resonance type should be used.

Figure 23.3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 23.2. When a crystal resonator is used, the range of its frequencies is from 8 to 25 MHz.

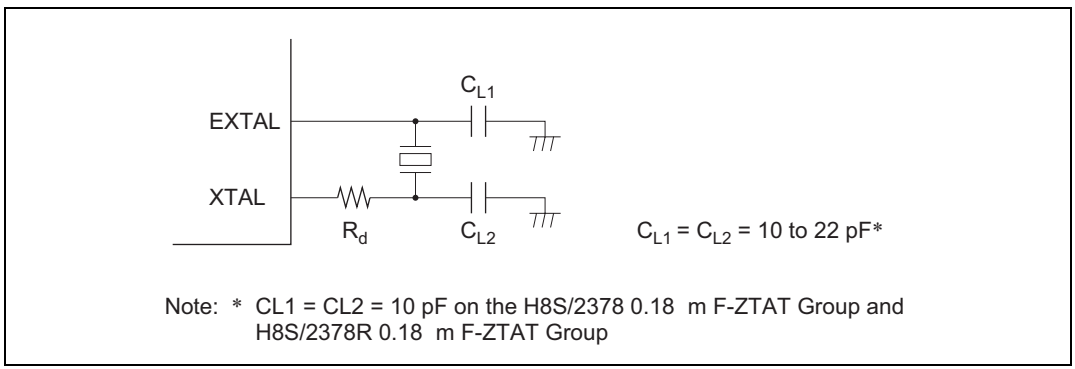


Figure 23.2 Connection of Crystal Resonator (Example)

Table 23.1 Damping Resistance Value

Frequency (MHz)	8	12	16	20	25
R_d (Ω)	200	0	0	0	0

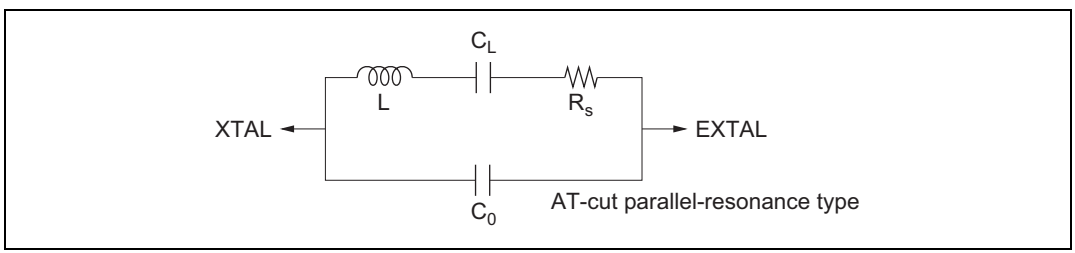


Figure 23.3 Crystal Resonator Equivalent Circuit

Appendix

A. I/O Port States in Each Pin State

Port Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port 1	1, 2, 4, 7	T	T	keep	keep	I/O port
Port 2	1, 2, 4, 7	T	T	keep	keep	I/O port
P34 to P30	1, 2, 4, 7	T	T	keep	keep	I/O port
P35/$\overline{OE}$/ CKE*1	1, 2, 4, 7	T	T	[OPE = 0, $\overline{OE}$, CKE output]	[OPE = 0, $\overline{OE}$, CKE output]	[OPE = 0, $\overline{OE}$, CKE output]
				T	T	$\overline{OE}$, CKE
				[OPE = 1, $\overline{OE}$ output]	[Other than the above]	[Other than the above]
				H	keep	I/O port
				[OPE = 1, CKE output]		
				L		
				[Other than the above]		
				keep		
P47/DA1	1, 2, 4, 7	T	T	[DAOE1 = 1]	keep	Input port
				keep		
				[DAOE1 = 0]		
				T		
P46/DA0	1, 2, 4, 7	T	T	[DAOE0 = 1]	keep	Input port
				keep		
				[DAOE0 = 0]		
				T		
P45 to P40	1, 2, 4, 7	T	T	T	T	Input port
P53 to P50	1, 2, 4, 7	T	T	keep	keep	I/O port
Port 6	1, 2, 4, 7	T	T	keep	keep	I/O port
Port 8	1, 2, 4, 7	T	T	keep	keep	I/O port