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	34MHz
Connectivity	I²C, IrDA, SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	96
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K 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/df2370vfq34v

5.3.5 IRQ Status Register (ISR)

ISR is an IRQ15 to IRQ0 interrupt request flag register.

Bit	Bit Name	Initial Value	R/W	Description
15	IRQ15F	0	R/(W)*	[Setting condition] When the interrupt source selected by ISCR occurs [Clearing conditions] • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0 (n = 15 to 0)
14	IRQ14F	0	R/(W)*	
13	IRQ13F	0	R/(W)*	
12	IRQ12F	0	R/(W)*	
11	IRQ11F	0	R/(W)*	
10	IRQ10F	0	R/(W)*	
9	IRQ9F	0	R/(W)*	
8	IRQ8F	0	R/(W)*	
7	IRQ7F	0	R/(W)*	
6	IRQ6F	0	R/(W)*	
5	IRQ5F	0	R/(W)*	
4	IRQ4F	0	R/(W)*	
3	IRQ3F	0	R/(W)*	
2	IRQ2F	0	R/(W)*	
1	IRQ1F	0	R/(W)*	
0	IRQ0F	0	R/(W)*	

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

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*1 Advanced Mode	IPR	Priority	DTC Activation	DMAC Activation
SCI_0	ERI0	88	H'0160	IPRI2 to IPRI0	High	—	—
	RXI0	89	H'0164			○	○
	TXI0	90	H'0168			○	○
	TEI0	91	H'016C			—	—
SCI_1	ERI1	92	H'0170	IPRJ14 to IPRJ12		—	—
	RXI1	93	H'0174			○	○
	TXI1	94	H'0178			○	○
	TEI1	95	H'017C			—	—
SCI_2	ERI2	96	H'0180	IPRJ10 to IPRJ8		—	—
	RXI2	97	H'0184			○	—
	TXI2	98	H'0188			○	—
	TEI2	99	H'018C			—	—
SCI_3	ERI3	100	H'0190	IPRJ6 to IPRJ4		—	—
	RXI3	101	H'0194			○	—
	TXI3	102	H'0198			○	—
	TEI3	103	H'019C			—	—
SCI_4	ERI4	104	H'01A0	IPRJ2 to IPRJ0		—	—
	RXI4	105	H'01A4			○	—
	TXI4	106	H'01A8			○	—
	TEI4	107	H'01AC			—	—
	Reserved for system use	108	H'01B0	IPRK14 to IPRK12		—	—
		109	H'01B4			—	—
		110	H'01B8			—	—
		111	H'01BC			—	—
		112	H'01C0	IPRK10 to IPRK8		—	—
		113	H'01C4			—	—
		114	H'01C8			—	—
		115	H'01CC		Low	—	—

Bit	Bit Name	Initial Value	R/W	Description
2	MXC2	0	R/W	011: 11-bit shift
1	MXC1	0	R/W	• When 8-bit access space is designated: Row address bits A23 to A11 used for comparison When 16-bit access space is designated: Row address bits A23 to A12 used for comparison
0	MXC0	0	R/W	Synchronous DRAM interface 100: 8-bit shift • When 8-bit access space is designated: Row address bits A23 to A8 used for comparison • When 16-bit access space is designated: Row address bits A23 to A9 used for comparison The precharge-sel is A15 to A9 of the column address. 101: 9-bit shift • When 8-bit access space is designated: Row address bits A23 to A9 used for comparison • When 16-bit access space is designated: Row address bits A23 to A10 used for comparison The precharge-sel is A15 to A10 of the column address. 110: 10-bit shift • When 8-bit access space is designated: Row address bits A23 to A10 used for comparison • When 16-bit access space is designated: Row address bits A23 to A11 used for comparison The precharge-sel is A15 to A11 of the column address.

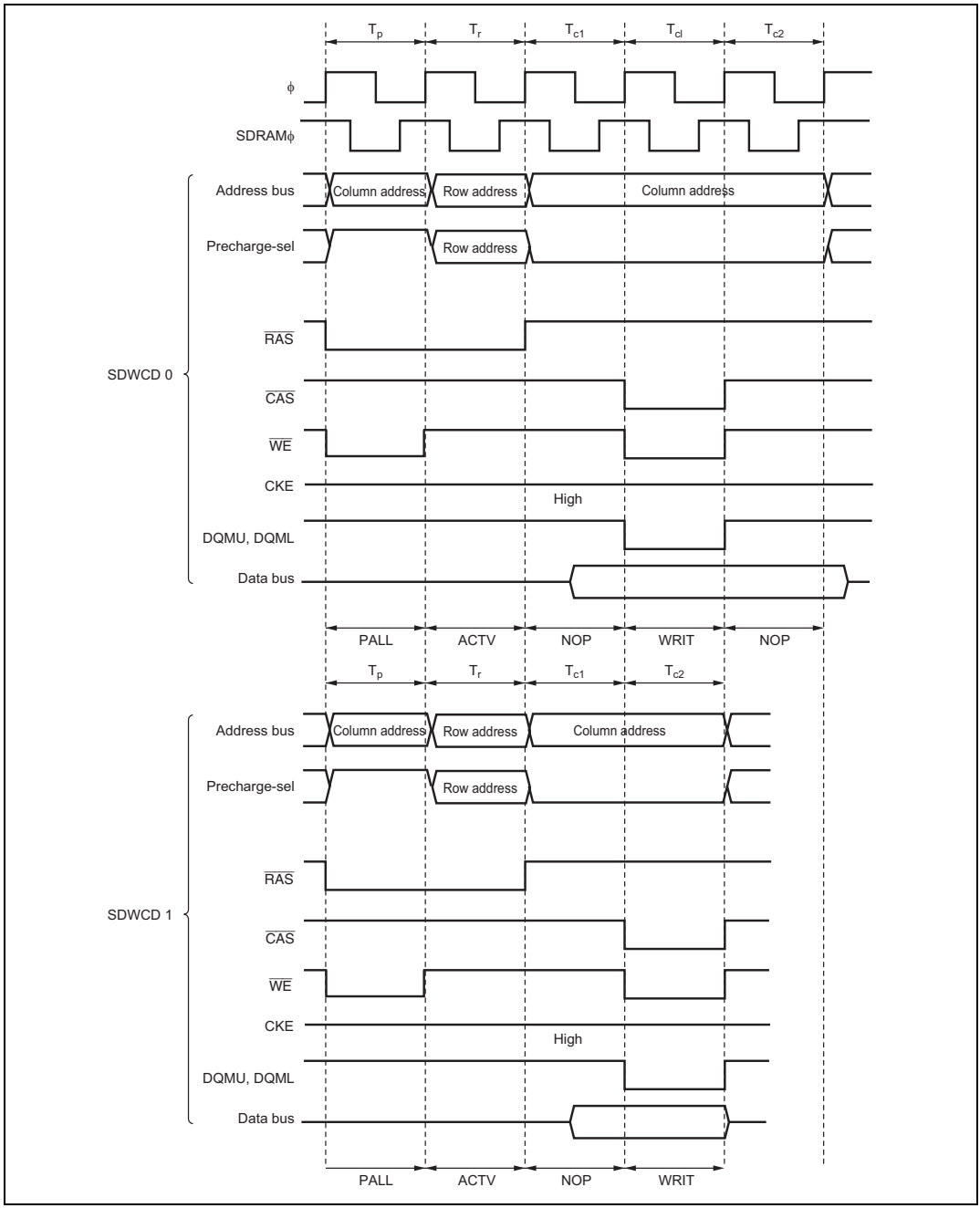


Figure 6.5 CAS Latency Control Cycle Disable Timing during Continuous Synchronous DRAM Space Write Access (for CAS Latency 2)

6.6.12 Refresh Control

This LSI is provided with a DRAM refresh control function. CAS-before-RAS (CBR) refreshing is used. In addition, self-refreshing can be executed when the chip enters the software standby state.

Refresh control is enabled when any area is designated as DRAM space in accordance with the setting of bits RMTS2 to RMTS0 in DRAMCR.

CAS-before-RAS (CBR) Refreshing: To select CBR refreshing, set the RFSHE bit to 1 in REFCR.

With CBR refreshing, RTCNT counts up using the input clock selected by bits RTCK2 to RTCK0 in REFCR, and when the count matches the value set in RTCOR (compare match), refresh control is performed. At the same time, RTCNT is reset and starts counting up again from H'00. Refreshing is thus repeated at fixed intervals determined by RTCOR and bits RTCK2 to RTCK0. Set a value in RTCOR and bits RTCK2 to RTCK0 that will meet the refreshing interval specification for the DRAM used.

When bits RTCK2 to RTCK0 in REFCR are set, RTCNT starts counting up. RTCNT and RTCOR settings should therefore be completed before setting bits RTCK2 to RTCK0. RTCNT operation is shown in figure 6.34, compare match timing in figure 6.35, and CBR refresh timing in figure 6.36.

When the CBRM bit in REFCR is cleared to 0, access to external space other than DRAM space is performed in parallel during the CBR refresh period.

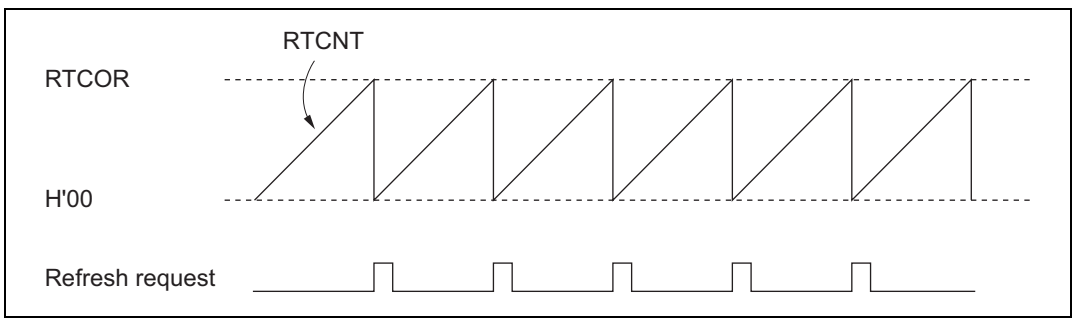


Figure 6.34 RTCNT Operation

7.5.8 Basic Bus Cycles

An example of the basic DMAC bus cycle timing is shown in figure 7.17. In this example, word-size transfer is performed from 16-bit, 2-state access space to 8-bit, 3-state access space. When the bus is transferred from the CPU to the DMAC, a source address read and destination address write are performed. The bus is not released in response to another bus request, etc., between these read and write operations. As like CPU cycles, DMA cycles conform to the bus controller settings.

The address is not output to the external address bus in an access to on-chip memory or an internal I/O register.

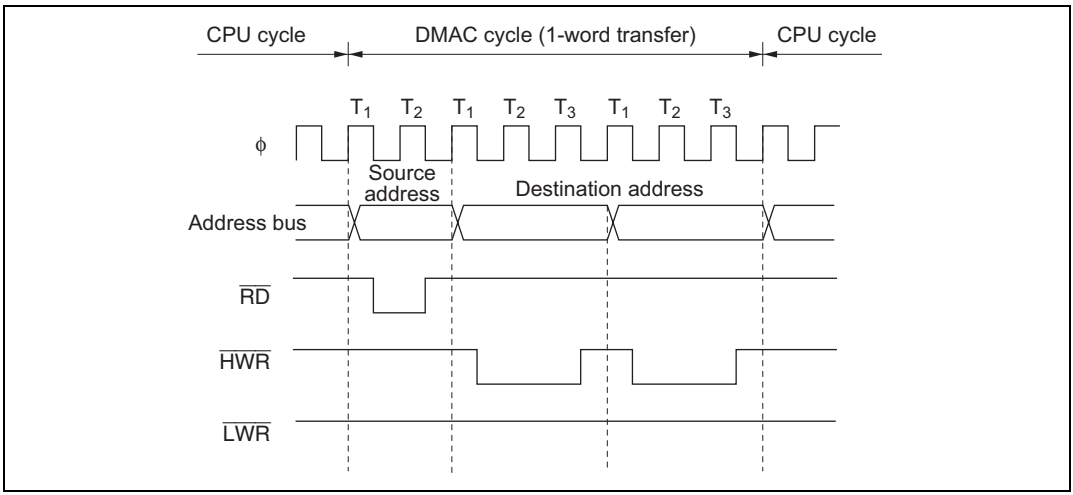


Figure 7.17 Example of DMA Transfer Bus Timing

10.2 Port 2

Port 2 is an 8-bit I/O port that also has other functions. The port 2 has the following registers.

- Port 2 data direction register (P2DDR)
- Port 2 data register (P2DR)
- Port 2 register (PORT2)

10.2.1 Port 2 Data Direction Register (P2DDR)

The individual bits of P2DDR specify input or output for the pins of port 2.

P2DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7	P27DDR	0	W	When a pin function is specified to a general purpose I/O, setting this bit to 1 makes the corresponding port 1 pin an output pin, while clearing this bit to 0 makes the pin an input pin.
6	P26DDR	0	W	
5	P25DDR	0	W	
4	P24DDR	0	W	
3	P23DDR	0	W	
2	P22DDR	0	W	
1	P21DDR	0	W	
0	P20DDR	0	W	

- Timer interrupt enable register_3 (TIER_3)
- Timer status register_3 (TSR_3)
- Timer counter_3 (TCNT_3)
- Timer general register A_3 (TGRA_3)
- Timer general register B_3 (TGRB_3)
- Timer general register C_3 (TGRC_3)
- Timer general register D_3 (TGRD_3)
- Timer control register_4 (TCR_4)
- Timer mode register_4 (TMDR_4)
- Timer I/O control register _4 (TIOR_4)
- Timer interrupt enable register_4 (TIER_4)
- Timer status register_4 (TSR_4)
- Timer counter_4 (TCNT_4)
- Timer general register A_4 (TGRA_4)
- Timer general register B_4 (TGRB_4)
- Timer control register_5 (TCR_5)
- Timer mode register_5 (TMDR_5)
- Timer I/O control register_5 (TIOR_5)
- Timer interrupt enable register_5 (TIER_5)
- Timer status register_5 (TSR_5)
- Timer counter_5 (TCNT_5)
- Timer general register A_5 (TGRA_5)
- Timer general register B_5 (TGRB_5)

Common Registers

- Timer start register (TSTR)
- Timer synchronous register (TSYR)

Table 11.28 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGRA_0	TGRC_0
	TGRB_0	TGRD_0
3	TGRA_3	TGRC_3
	TGRB_3	TGRD_3

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 11.12.

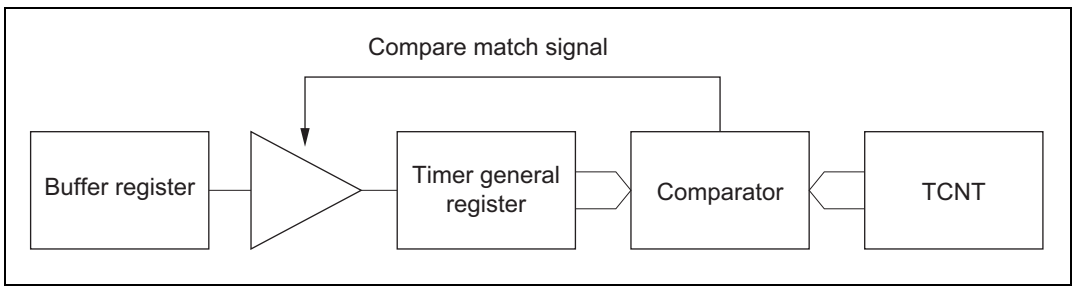


Figure 11.12 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 11.13.

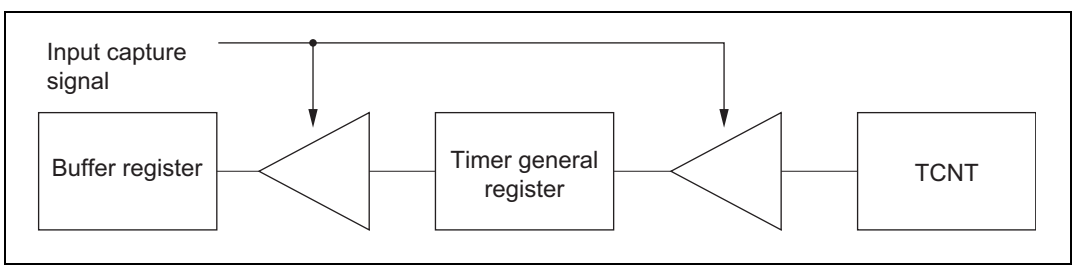


Figure 11.13 Input Capture Buffer Operation

TCFV Flag/TCFU Flag Setting Timing: Figure 11.40 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and the TCIV interrupt request signal timing.

Figure 11.41 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and the TCIU interrupt request signal timing.

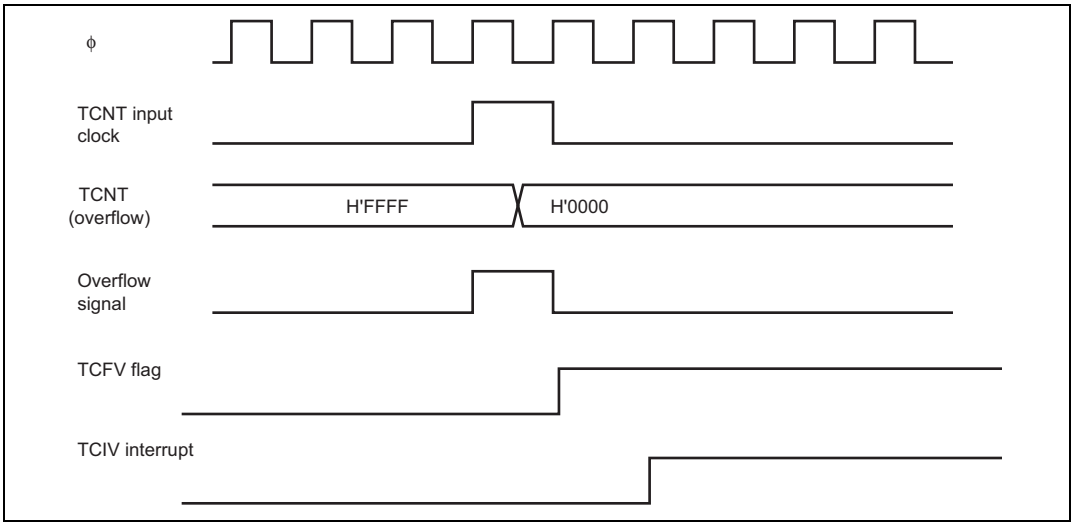


Figure 11.40 TCIV Interrupt Setting Timing

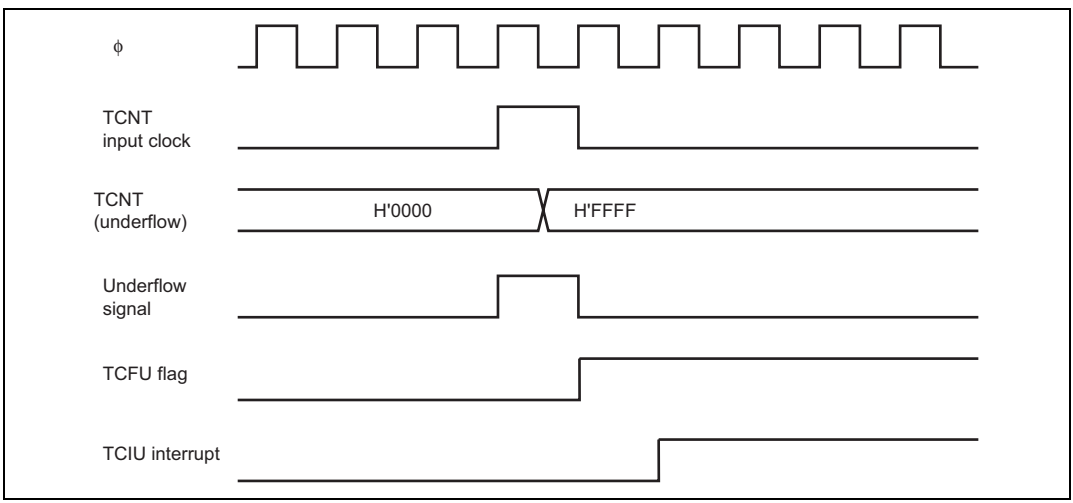


Figure 11.41 TCIU Interrupt Setting Timing

1. Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.

2. Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.

3. The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.

4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts.

   If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

Figure 13.1 shows a block diagram of the 8-bit timer module (TMR_0 and TMR_1).

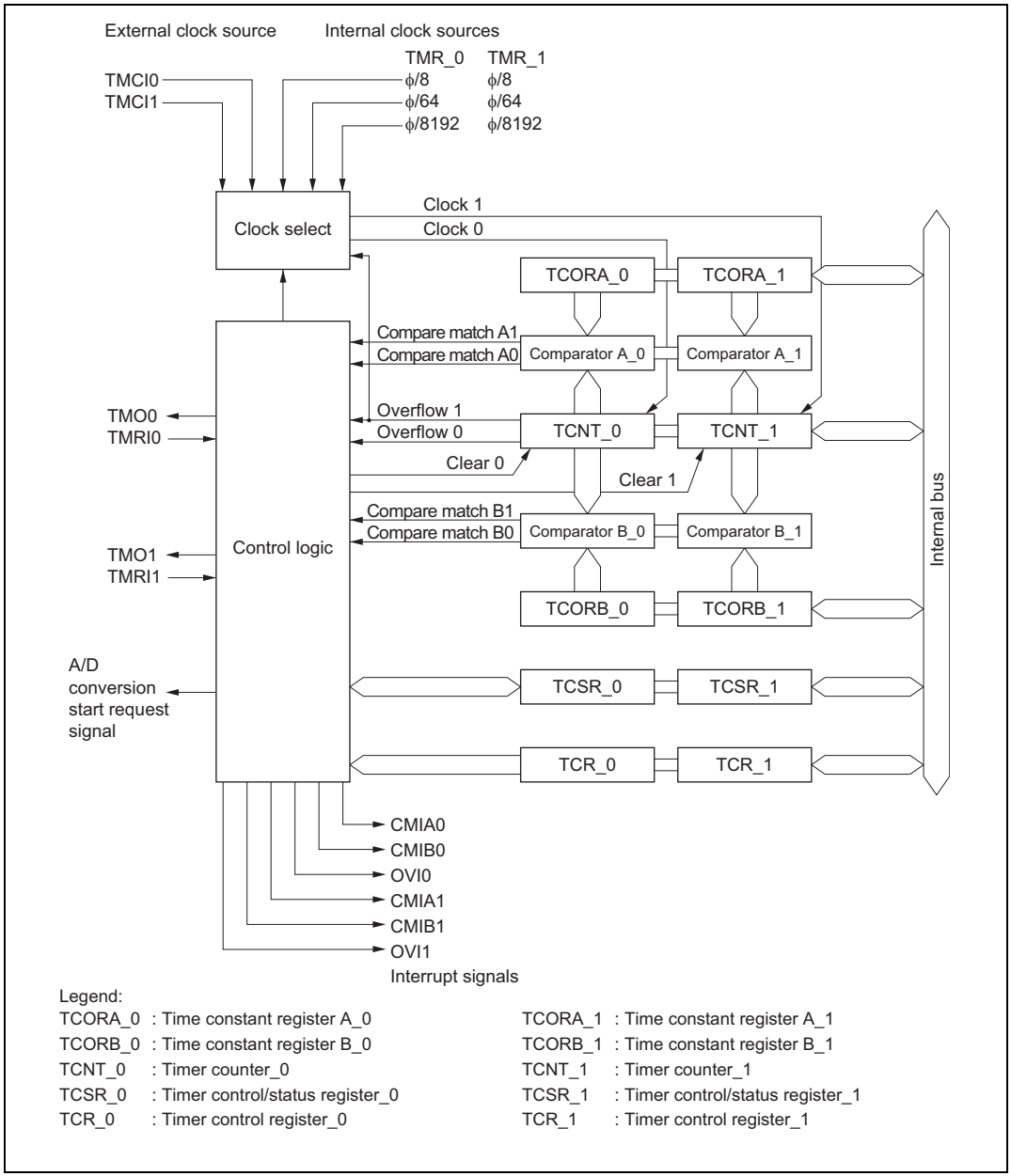


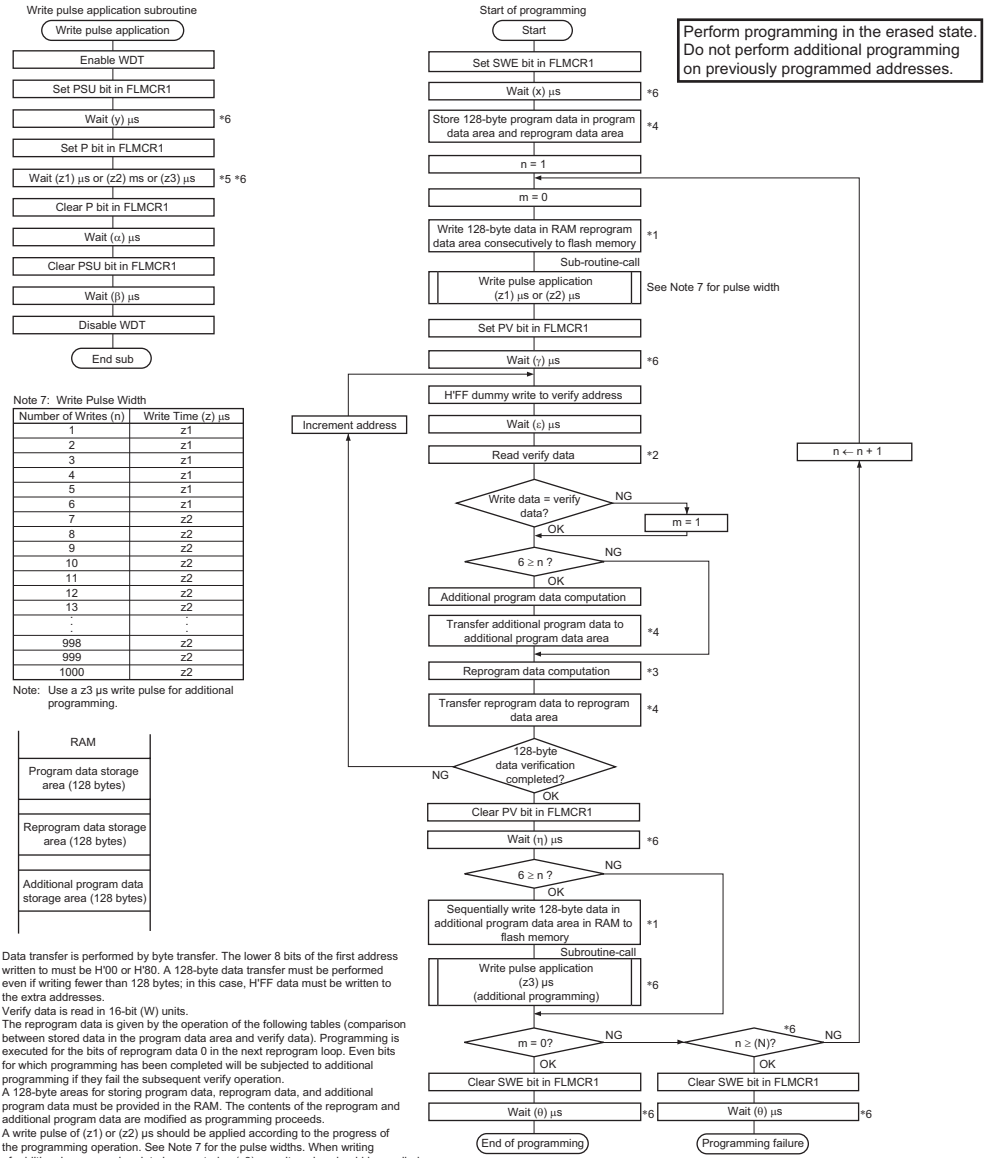
Figure 13.1 Block Diagram of 8-Bit Timer Module

- DACR01 (Available only for the H8S/2377, H8S/2377R, H8S/2378 0.18μm F-ZTAT Group, and H8S/2378R 0.18μm F-ZTAT Group)

Bit	Bit Name	Initial Value	R/W	Description
7	DAOE1	0	R/W	D/A Output Enable 1 Controls D/A conversion and analog output. 0: Analog output (DA1) is disabled 1: Channel 1 D/A conversion is enabled; analog output (DA1) is enabled
6	DAOE0	0	R/W	D/A Output Enable 0 Controls D/A conversion and analog output. 0: Analog output (DA0) is disabled 1: Channel 0 D/A conversion is enabled; analog output (DA0) is enabled
5	DAE	0	R/W	D/A Enable Used together with the DAOE0 and DAOE1 bits to control D/A conversion. When the DAE bit is cleared to 0, channel 0 and 1 D/A conversions are controlled independently. When the DAE bit is set to 1, channel 0 and 1 D/A conversions are controlled together. Output of conversion results is always controlled independently by the DAOE0 and DAOE1 bits. For details, see table 18.2.
4 to 0	—	All 1	—	Reserved These bits are always read as 1 and cannot be modified.

Table 18.2 Control of D/A Conversion

Bit 5 DAE	Bit 7 DAOE1	Bit 6 DAOE0	Description
0	0	0	D/A conversion disabled
		1	Channel 0 D/A conversion enabled, channel1 D/A conversion disabled
	1	0	Channel 1 D/A conversion enabled, channel0 D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
1	0	0	D/A conversion disabled
		1	Channel 0 and 1 D/A conversions enabled
	1	0	
		1	



Note 7: Write Pulse Width

Number of Writes (n)	Write Time (z) μs
1	z1
2	z1
3	z1
4	z1
5	z1
6	z1
7	z2
8	z2
9	z2
10	z2
11	z2
12	z2
13	z2
⋮	⋮
998	z2
999	z2
1000	z2

Note: Use a z3 μs write pulse for additional programming.

Notes:
1. Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80. 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.
2. Verify data is read in 16-bit (W) units.
3. The reprogram data is given by the operation of the following tables (comparison between stored data in the program data area and verify data). Programming is executed for the bits of reprogram data 0 in the next reprogram loop. Even bits for which programming has been completed will be subjected to additional programming if they fail the subsequent verify operation.
4. A 128-byte areas for storing program data, reprogram data, and additional program data must be provided in the RAM. The contents of the reprogram and additional program data are modified as programming proceeds.
5. A write pulse of (z1) or (z2) μs should be applied according to the progress of the programming operation. See Note 7 for the pulse widths. When writing of additional-programming data is executed, a (z3) μs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
6. For the values of x, y, z1, z2, z3, α, β, γ, ε, η, θ, and N, see section 26.1.6, Flash Memory Characteristics.

Program Data Operation Chart

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
	1	0	Programming incomplete; reprogram
1	0	1	
	1		Still in erased state; no action

Additional Program Data Operation Chart

Reprogram Data (X')	Verify Data (V)	Additional Program Data (Y)	Comments
0	0	0	Additional programming executed
	1	1	Additional programming not executed
1	0		Additional programming not executed
	1		Additional programming not executed

Figure 20.7 Program/Program-Verify Flowchart

Item	Storable/Executable Area			Selected MAT		
	On-chip RAM	User Boot MAT	External Space (Expanded Mode)	User MAT	User Boot MAT	Embedded Program Storage Area
Operation for Settings of Program Parameter	○	×	○	○		
Execution of Programming	○	×	×	○		
Determination of Program Result	○	×	○	○		
Operation for Program Error	○	×*2	○	○		
Operation for FKEY Clear	○	×	○	○		
Switching MATs by FMATS	○	×	×		○	

Notes: 1. Transferring the data to the on-chip RAM enables this area to be used.
2. Switching FMATS by a program in the on-chip RAM enables this area to be used.

(d) Clock Mode Selection

The boot program will set the specified clock mode. The program will return the selected clock-mode information after this setting has been made.

The clock-mode selection command should be sent after the device-selection commands.

Command	H'11	Size	Mode	SUM

- Command, H'11, (one byte): Selection of clock mode
- Size (one byte): Amount of data that represents the modes
- Mode (one byte): A clock mode returned in reply to the supported clock mode inquiry.
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to the clock mode selection command
  ACK will be returned when the clock mode matches.

Error Response	H'91	ERROR

- Error response, H'91, (one byte): Error response to the clock mode selection command
- ERROR, (one byte): Error code
  H'11: Checksum error
  H'22: Clock mode error, that is, the clock mode does not match.

Even if the clock mode numbers are H'00 and H'01 by a clock mode inquiry, the clock mode must be selected using these respective values.

Command	H'58	Size	Block number	SUM

- Command, H'58, (one byte): Erasure
- Size, (one byte): The number of bytes that represents the block number
  This is fixed to 1.
- Block number (one byte): H'FF
  Stop code for erasure
- SUM (one byte): Checksum

Response	H'06

- Response, H'06, (one byte): Response to end of erasure (ACK)
  When erasure is to be performed after the block number H'FF has been sent, the procedure should be executed from the erasure selection command.

(11) Memory read

The boot program will return the data in the specified address.

Command	H'52	Size	Area	Read address	
	Read size				SUM

- Command: H'52 (1 byte): Memory read
- Size (1 byte): Amount of data that represents the area, read address, and read size (fixed at 9)
- Area (1 byte)
  H'00: User boot MAT
  H'01: User MAT
  An address error occurs when the area setting is incorrect.
- Read address (4 bytes): Start address to be read from
- Read size (4 bytes): Size of data to be read
- SUM (1 byte): Checksum

Response	H'52	Read size						
	Data	…						
	SUM							

- Response: H'52 (1 byte): Response to memory read
- Read size (4 bytes): Size of data to be read
- Data (n bytes): Data for the read size from the read address
- SUM (1 byte): Checksum

Port Name	MCU Operating Mode		Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
Port E	1, 2, 4	8-bit bus	T	T	keep	keep	I/O port
		16-bit bus	T	T	T	T	D7 to D0
	3, 5*2, 7	8-bit bus	T	T	keep	keep	I/O port
		16-bit bus	T	T	[Data bus]	[Data bus]	[Data bus]
					T	T	D7 to D0
					[Other than the above]	[Other than the above]	[Other than the above]
					keep	keep	I/O port
PF7/ϕ	1, 2, 4		Clock output	T	[Clock output]	[Clock output]	[Clock output]
	3, 5*2, 7		T		H	Clock output	Clock output
					[Other than the above]	[Other than the above]	[Other than the above]
					keep	keep	Input port
PF6/$\overline{AS}$	1, 2, 4		H	T	[OPE = 0, $\overline{AS}$ output]	[$\overline{AS}$ output]	[$\overline{AS}$ output]
					T	T	$\overline{AS}$
	3, 5*2, 7		T		[OPE = 1, $\overline{AS}$ output]	[Other than the above]	[Other than the above]
					H	keep	I/O port
					[Other than the above]		
					keep		

Instruction	1	2	3	4	5	6	7	8	9
OR.L ERs,ERd	R:W 2nd	R:W NEXT							
ORC #xx:8,CCR	R:W NEXT								
ORC #xx:8,EXR	R:W 2nd	R:W NEXT							
POP.W Rn	R:W NEXT	1 State of internal operation	R:W EA						
POP.L ERn	R:W 2nd	R:W NEXT	1 State of internal operation	R:W:M EA	R:W EA+2				
PUSH.W Rn	R:W NEXT	1 State of internal operation	W:W EA						
PUSH.L ERn	R:W 2nd	R:W NEXT	1 State of internal operation	W:W:M EA	W:W EA+2				
ROTL.B Rd	R:W NEXT								
ROTL.B #2,Rd	R:W NEXT								
ROTL.W Rd	R:W NEXT								
ROTL.W #2,Rd	R:W NEXT								
ROTL.L ERd	R:W NEXT								
ROTL.L #2,ERd	R:W NEXT								
ROTR.B Rd	R:W NEXT								
ROTR.B #2,Rd	R:W NEXT								
ROTR.W Rd	R:W NEXT								
ROTR.W #2,Rd	R:W NEXT								
ROTR.L ERd	R:W NEXT								
ROTR.L #2,ERd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
SLEEP	R:W NEXT	Internal operation: M							
STC CCR,Rd	R:W NEXT								
STC EXR,Rd	R:W NEXT								
STC CCR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC EXR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC CCR, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR, @(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC EXR, @(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC CCR, @-ERd	R:W 2nd	R:W NEXT	1 state of internal operation	W:W EA					
STC EXR, @-ERd	R:W 2nd	R:W NEXT	1 state of internal operation	W:W EA					
STC CCR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STC EXR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STM.L (ERn-ERn+1), @-SP *8	R:W 2nd	R:W NEXT	1 state of internal operation	W:W:M Stack (H) *2	W:W Stack (L) *2				
STM.L (ERn-ERn+2), @-SP *8	R:W 2nd	R:W NEXT	1 state of internal operation	W:W:M Stack (H) *2	W:W Stack (L) *2				