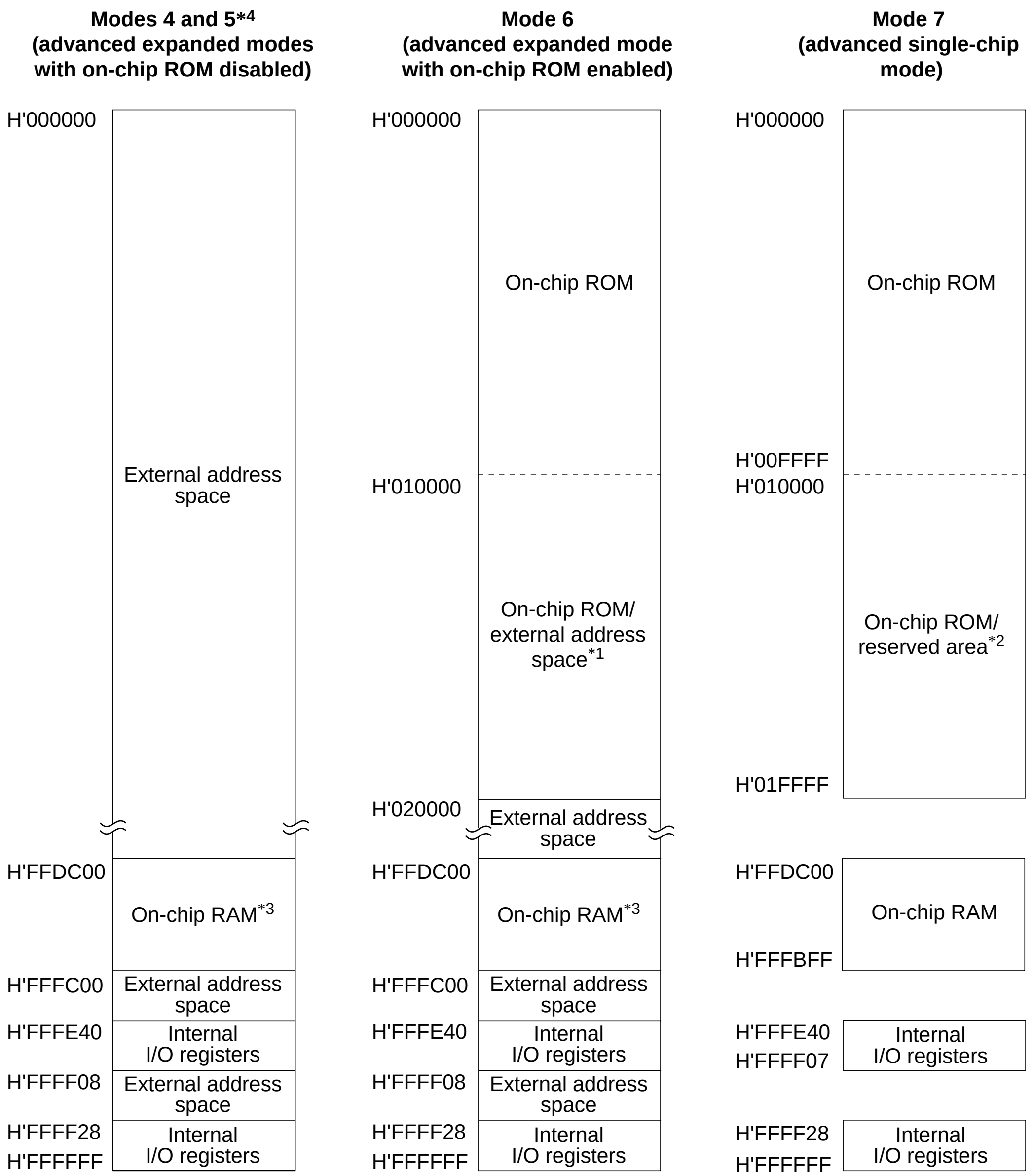
Welcome to E-XFL.COM

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.

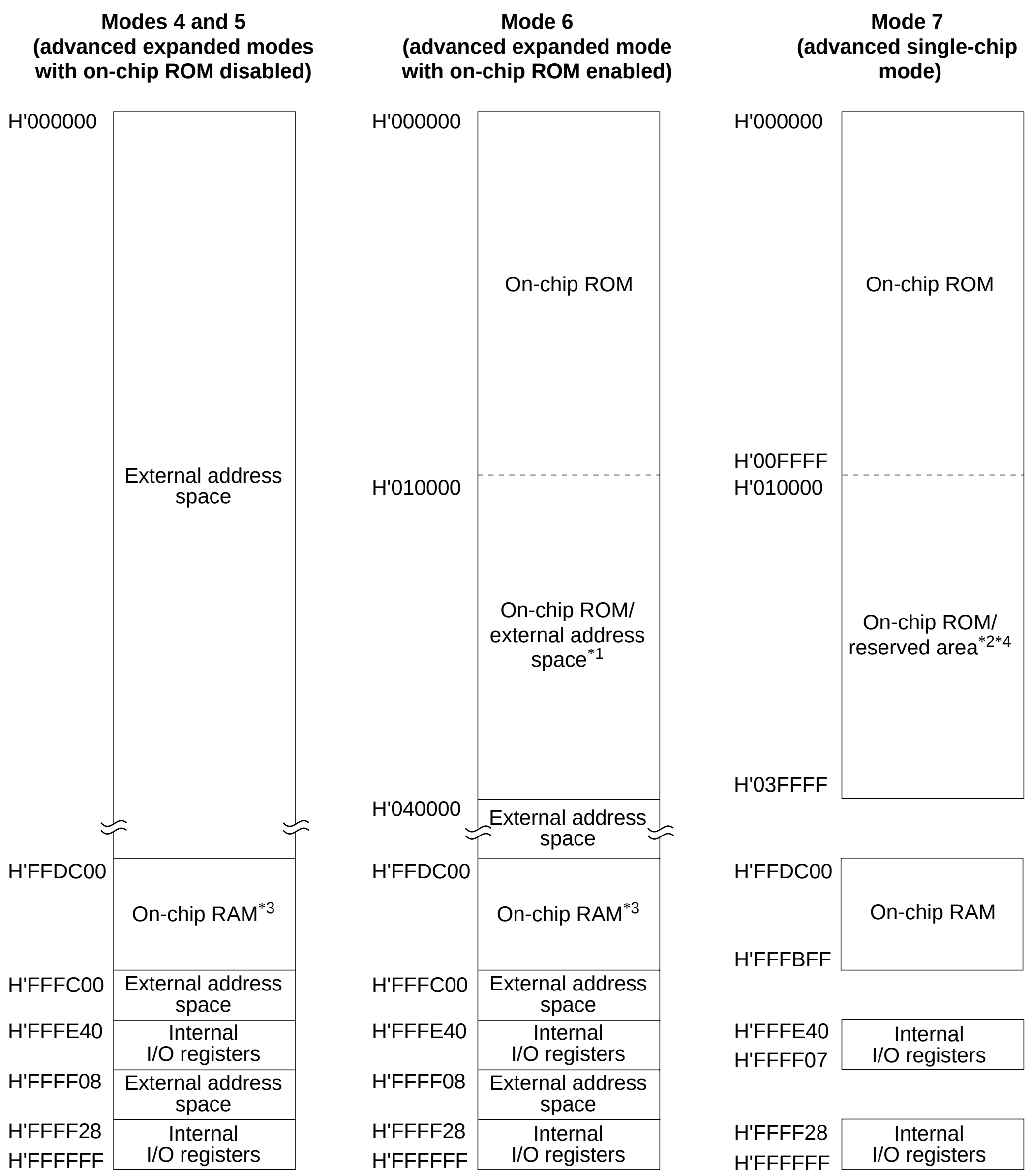
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	H8S/2000
Core Size	16-Bit
Speed	20MHz
Connectivity	SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	87
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	120-TQFP
Supplier Device Package	120-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2357te20v



Notes:
1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
4. Only modes 4 and 5 are provided in the H8S/2352.

Figure 3-1 Memory Map in Each Operating Mode (H8S/2357, H8S/2352) (1)



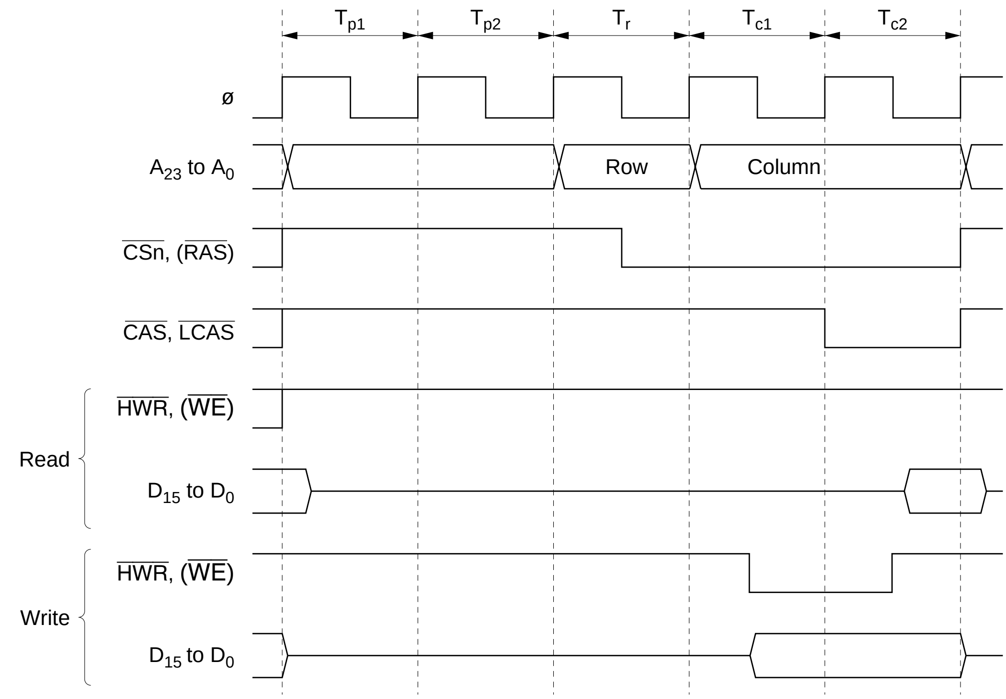
Notes: 1. External addresses when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
2. Reserved area when EAE = 1 in BCRL; on-chip ROM when EAE = 0.
3. External addresses can be accessed by clearing the RAME bit in SYSCR to 0.
4. Access to the reserved area is inhibited.

Figure 3-5 Memory Map in Each Operating Mode (H8S/2398) (2)

6.5.7 Precharge State Control

When DRAM is accessed, RAS precharging time must be secured. With the H8S/2357 Series, one T_p state is always inserted when DRAM space is accessed. This can be changed to two T_p states by setting the TPC bit in MCR to 1. Set the appropriate number of T_p cycles according to the DRAM connected and the operating frequency of the H8S/2357 Group. Figure 6-16 shows the timing when two T_p states are inserted.

When the TPC bit is set to 1, two T_p states are also used for refresh cycles.



Note: n = 2 to 5

Figure 6-16 Timing with Two Precharge States

6.5.8 Wait Control

There are two ways of inserting wait states in a DRAM access cycle: program wait insertion and pin wait insertion using the $\overline{\text{WAIT}}$ pin.

Program Wait Insertion: When the bit in ASTCR corresponding to an area designated as DRAM space is set to 1, from 0 to 3 wait states can be inserted automatically between the T_{c1} state and T_{c2} state, according to the settings of WCRH and WCRL.

Pin Wait Insertion: When the WAITE bit in BCRH is set to 1, wait input by means of the $\overline{\text{WAIT}}$ pin is enabled regardless of the setting of the AST bit in ASTCR. When DRAM space is accessed in this state, a program wait is first inserted. If the $\overline{\text{WAIT}}$ pin is low at the falling edge of ø in the last T_{c1} or T_w state, another T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

(1) Normal Mode

ETCRA

Transfer Counter

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCR																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

In normal mode, ETCRA functions as a 16-bit transfer counter. ETCRA is decremented by 1 each time a transfer is performed, and transfer ends when the count reaches H'0000. ETCRB is not used at this time.

ETCRB

ETCRB is not used in normal mode.

(2) Block Transfer Mode

ETCRA

Holds block size

Bit	15	14	13	12	11	10	9	8
ETCRAH								
Initial value	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Block size counter

Bit	7	6	5	4	3	2	1	0
ETCRAL								
Initial value	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined

ETCRB

Block Transfer Counter

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETCRB																
Initial value	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In block transfer mode, ETCRAL functions as an 8-bit block size counter and ETCRAH holds the block size. ETCRAL is decremented each time a 1-byte or 1-word transfer is performed, and when the count reaches H'00, ETCRAL is loaded with the value in ETCRAH. So by setting the block size in ETCRAH and ETCRAL, it is possible to repeatedly transfer blocks consisting of any desired number of bytes or words.

ETCRB functions in block transfer mode, as a 16-bit block transfer counter. ETCRB is decremented by 1 each time a block is transferred, and transfer ends when the count reaches H'0000.

Bits 3 and 1—Data Transfer Interrupt Enable B (DTIEB): These bits enable or disable an interrupt to the CPU or DTC when transfer is interrupted. If the DTIEB bit is set to 1 when DTME = 0, the DMAC regards this as indicating a break in the transfer, and issues a transfer break interrupt request to the CPU or DTC.

A transfer break interrupt can be canceled either by clearing the DTIEB bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the DTME bit to 1.

Bit 3—Data Transfer Interrupt Enable 1B (DTIE1B): Enables or disables the channel 1 transfer break interrupt.

Bit 3 DTIE1B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bit 1—Data Transfer Interrupt Enable 0B (DTIE0B): Enables or disables the channel 0 transfer break interrupt.

Bit 1 DTIE0B	Description	
0	Transfer break interrupt disabled	(Initial value)
1	Transfer break interrupt enabled	

Bits 2 and 0—Data Transfer End Interrupt Enable A (DTIEA): These bits enable or disable an interrupt to the CPU or DTC when transfer ends. If DTIEA bit is set to 1 when DTE = 0, the DMAC regards this as indicating the end of a transfer, and issues a transfer end interrupt request to the CPU or DTC.

A transfer end interrupt can be canceled either by clearing the DTIEA bit to 0 in the interrupt handling routine, or by performing processing to continue transfer by setting the transfer counter and address register again, and then setting the DTE bit to 1.

Bit 2—Data Transfer Interrupt Enable 1A (DTIE1A): Enables or disables the channel 1 transfer end interrupt.

Bit 2 DTIE1A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

Bit 0—Data Transfer Interrupt Enable 0A (DTIE0A): Enables or disables the channel 0 transfer end interrupt.

Bit 0 DTIE0A	Description	
0	Transfer end interrupt disabled	(Initial value)
1	Transfer end interrupt enabled	

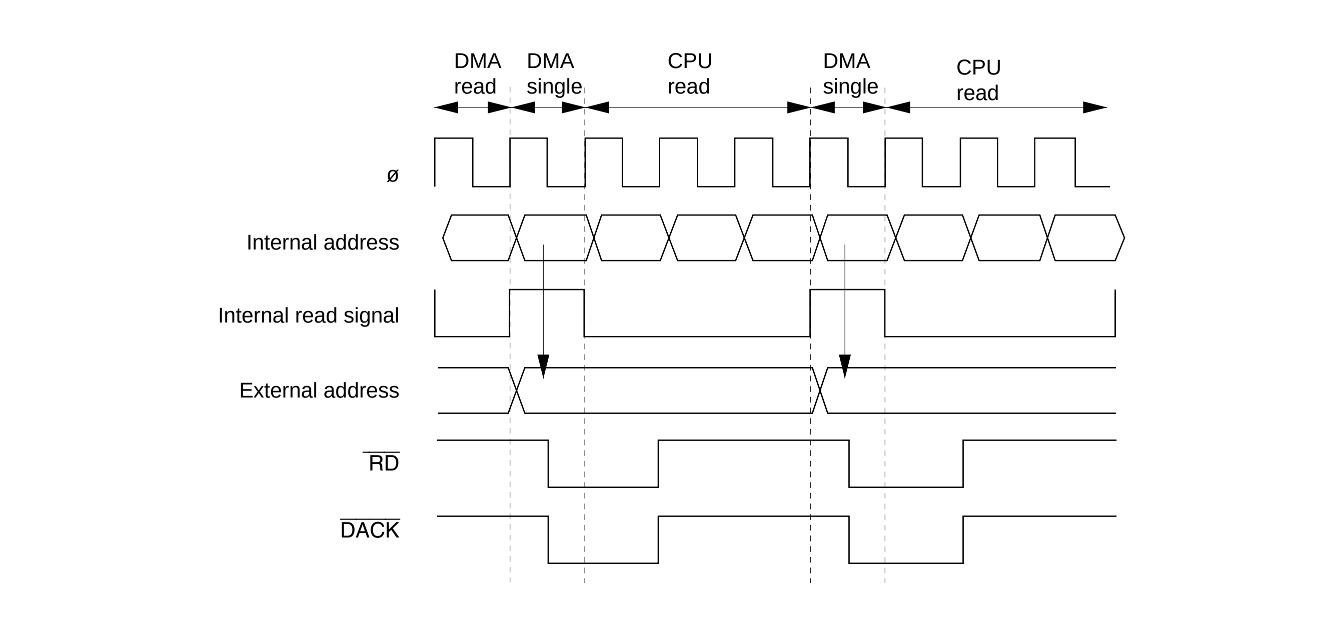


Figure 7-34 Example of Single Address Transfer Using Write Data Buffer Function

When the write data buffer function is activated, the DMAC recognizes that the bus cycle concerned has ended, and starts the next operation. Therefore, $\overline{\text{DREQ}}$ pin sampling is started one state after the start of the DMA write cycle or single address transfer.

7.5.13 DMAC Multi-Channel Operation

The DMAC channel priority order is: channel 0 > channel 1, and channel A > channel B. Table 7-13 summarizes the priority order for DMAC channels.

Table 7-13 DMAC Channel Priority Order

Short Address Mode	Full Address Mode	Priority
Channel 0A	Channel 0	High
Channel 0B		↑
Channel 1A	Channel 1	
Channel 1B		Low

Pin	Selection Method and Pin Functions
$P1_4$/PO12/TIOCA1	The pin function is switched as shown below according to the combination of the TPU channel 1 setting by bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, bits CCLR1 and CCLR0 in TCR1, bit NDER12 in NDERH, and bit P14DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	TIOCA1 output	$P1_4$ input	$P1_4$ output	PO12 output
		TIOCA1 input *1		

Note: 1. TIOCA1 input when MD3 to MD0 = B'0000, B'01××, IOA3 to IOA0 = B'10××.

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01××		B'001×	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1×××	B'0001 to B'0011 B'0101 to B'0111	B'××00	Other than B'××00	Other than B'××00	
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

×: Don't care

Note: 2. TIOCB1 output is disabled.

Port D Register (PORTD) (On-Chip ROM Version Only)

Bit :	7	6	5	4	3	2	1	0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value :	—*	—*	—*	—*	—*	—*	—*	—*
R/W :	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PD_7 to PD_0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port D pins (PD_7 to PD_0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.

After a power-on reset and in hardware standby mode, PORTD contents are determined by the pin states, as PDDDR and PDDR are initialized. PORTD retains its prior state after a manual reset*, and in software standby mode.

Note: * Manual reset is only supported in the H8S/2357 ZTAT.

Port D MOS Pull-Up Control Register (PDPCR) (On-Chip ROM Version Only)

Bit :	7	6	5	4	3	2	1	0
	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
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

Note: Setting is prohibited in the H8S/2352, H8S/2394, H8S/2392, and H8S/2390.

PDPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port D on an individual bit basis.

When a PDDDR bit is cleared to 0 (input port setting) in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PDPCR is initialized to H'00 by a power-on reset, and in hardware standby mode. It retains its prior state after a manual reset*, and in software standby mode.

Note: * Manual reset is only supported in the H8S/2357 ZTAT.

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the PPG.

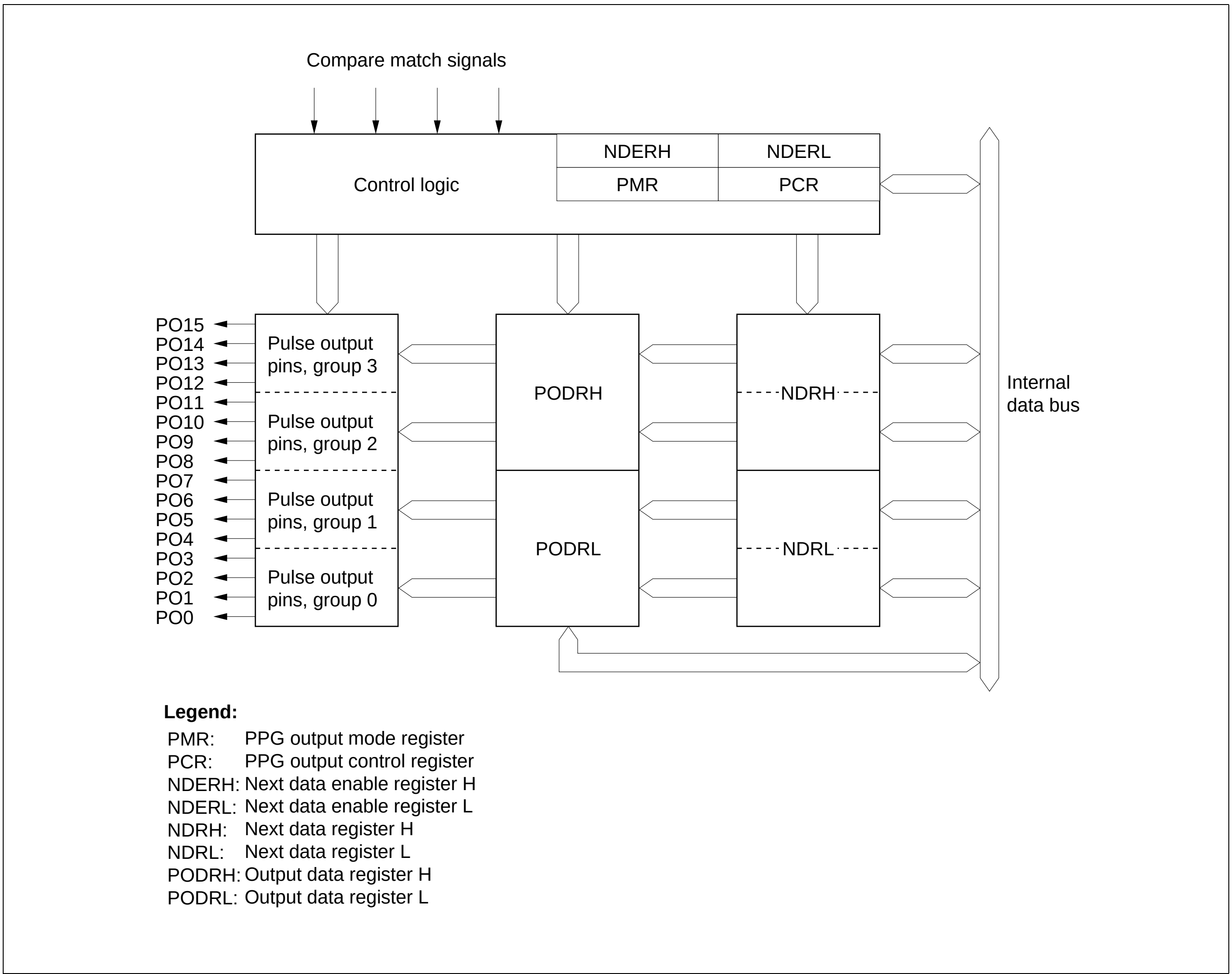


Figure 11-1 Block Diagram of PPG

12.2 Register Descriptions

12.2.1 Timer Counters 0 and 1 (TCNT0, TCNT1)

	TCNT0								TCNT1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	0	0	0	0	0	0	0	0	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	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT0 and TCNT1 are 8-bit readable/writable up-counters that increment on pulses generated from an internal or external clock source. This clock source is selected by clock select bits CKS2 to CKS0 of TCR. The CPU can read or write to TCNT0 and TCNT1 at all times.

TCNT0 and TCNT1 comprise a single 16-bit register, so they can be accessed together by word transfer instruction.

TCNT0 and TCNT1 can be cleared by an external reset input or by a compare match signal. Which signal is to be used for clearing is selected by clock clear bits CCLR1 and CCLR0 of TCR.

When a timer counter overflows from H'FF to H'00, OVF in TCSR is set to 1.

TCNT0 and TCNT1 are each initialized to H'00 by a reset and in hardware standby mode.

12.2.2 Time Constant Registers A0 and A1 (TCORA0, TCORA1)

	TCORA0								TCORA1							
Bit :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value :	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W :	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCORA0 and TCORA1 are 8-bit readable/writable registers. TCORA0 and TCORA1 comprise a single 16-bit register so they can be accessed together by word transfer instruction.

TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding CMFA flag of TCSR is set. Note, however, that comparison is disabled during the T_2 state of a TCOR write cycle.

The timer output can be freely controlled by these compare match signals and the settings of bits OS1 and OS0 of TCSR.

TCORA0 and TCORA1 are each initialized to H'FF by a reset and in hardware standby mode.

- Four interrupt sources
  — Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, and receive error — that can issue requests independently
  — The transmit-data-empty interrupt and receive-data-full interrupt can activate the DMA controller (DMAC) or data transfer controller (DTC) to execute data transfer

- Module stop mode can be set
  — As the initial setting, SCI operation is halted. Register access is enabled by exiting module stop mode.

Note: * Descriptions in this section refer to LSB-first transfer.

For details of the multiprocessor communication function, see section 14.3.3, Multiprocessor Communication Function.

Bit 2 MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ø, ø/4, ø/16, and ø/64, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 14.2.8, Bit Rate Register (BRR).

Bit 1 CKS1	Bit 0 CKS0	Description	
0	0	ø clock	(Initial value)
	1	ø/4 clock	
1	0	ø/16 clock	
	1	ø/64 clock	

14.2.6 Serial Control Register (SCR)

Bit :	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
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

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset, and by putting the device in standby mode or module stop mode. In the H8S/2398, H8S/2394, H8S/2392, and H8S/2390, however, the value in SCR is initialized to H'00 by a reset, or in hardware standby mode, but SCR retains its current state when the device enters software standby mode or module stop mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7 TIE	Description	
0	Transmit data empty interrupt (TXI) requests disabled*	(Initial value)
1	Transmit data empty interrupt (TXI) requests enabled	

Note: * TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3 PER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*2

Notes: 1. The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.
2. If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the last bit of the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2 TEND	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and write data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1 MPB	Description
0	[Clearing condition] (Initial value)* When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the SCAN bit, these bits select the analog input channels.

Only set the input channel while conversion is stopped (ADST = 0).

Group Selection	Channel Selection		Description	
CH2	**CH1**	**CH0**	**Single Mode (SCAN=0)**	**Scan Mode (SCAN=1)**
0	0	0	AN0 (Initial value)	AN0
		1	AN1	AN0, AN1
	1	0	AN2	AN0 to AN2
		1	AN3	AN0 to AN3
1	0	0	AN4	AN4
		1	AN5	AN4, AN5
	1	0	AN6	AN4 to AN6
		1	AN7	AN4 to AN7

16.2.3 A/D Control Register (ADCR)

Bit :	7	6	5	4	3	2	1	0
	TRGS1	TRGS0	—	—	—	—	—	—
Initial value :	0	0	1	1	1	1	1	1
R/W :	R/W	R/W	—	—	—/(R/W)*	—/(R/W)*	—	—

Note: * Applies to the H8S/2398, H8S/2394, H8S/2392, and H8S/2390.

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion operations.

ADCR is initialized to H'3F by a reset, and in standby mode or module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): Select enabling or disabling of the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped (ADST = 0).

Bit 7 TRGS1	Bit 6 TRGS0	Description
0	0	A/D conversion start by external trigger is disabled (Initial value)
	1	A/D conversion start by external trigger (TPU) is enabled
1	0	A/D conversion start by external trigger (8-bit timer) is enabled
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled

(1) For H8S/2357 and H8S/2352

Bits 5 to 0—Reserved: They are always read as 1 and cannot be modified.

(2) For H8S/2398, H8S/2394, H8S/2392, and H8S/2390

Bits 5, 4, 1, and 0—Reserved: They are always read as 1 and cannot be modified.

Bits 3 and 2—Reserved: Should always be written with 1.

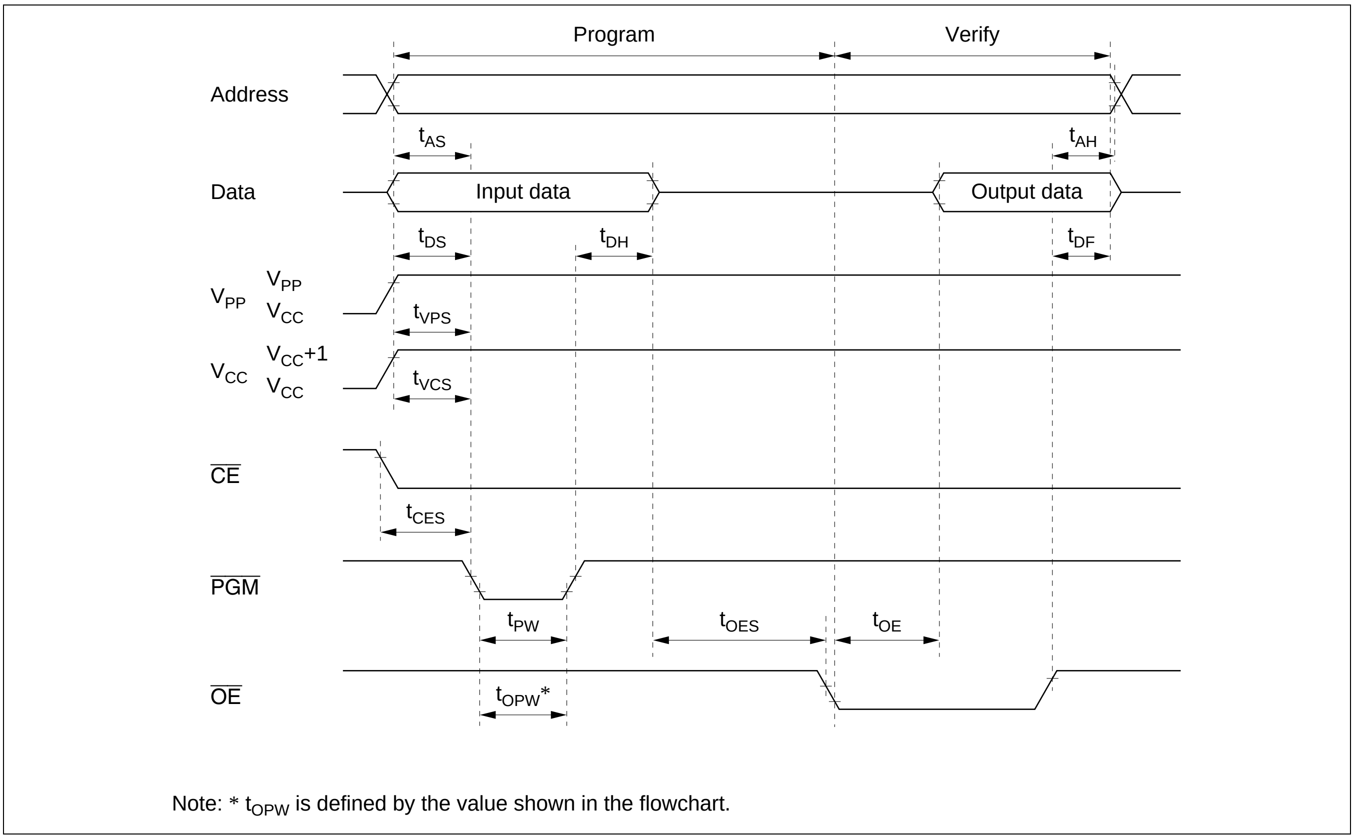


Figure 19-5 PROM Programming/Verification Timing

19.5.3 Programming Precautions

- Program using the specified voltages and timing.
  The programming voltage (V_{PP}) in PROM mode is 12.5 V.
  If the PROM programmer is set to Renesas Technology HN27C101 specifications, V_{PP} will be 12.5 V. Applied voltages in excess of the specified values can permanently destroy the MCU. Be particularly careful about the PROM programmer's overshoot characteristics.

- Before programming, check that the MCU is correctly mounted in the PROM programmer. Overcurrent damage to the MCU can result if the index marks on the PROM programmer, socket adapter, and MCU are not correctly aligned.

- Do not touch the socket adapter or MCU while programming. Touching either of these can cause contact faults and programming errors.

- The MCU cannot be programmed in page programming mode. Select the programming mode carefully.

- The size of the H8S/2357 PROM is 128 kbytes. Always set addresses within the range H'00000 to H'1FFFF. During programming, write H'FF to unused addresses to avoid verification errors.

Automatic SCI Bit Rate Adjustment

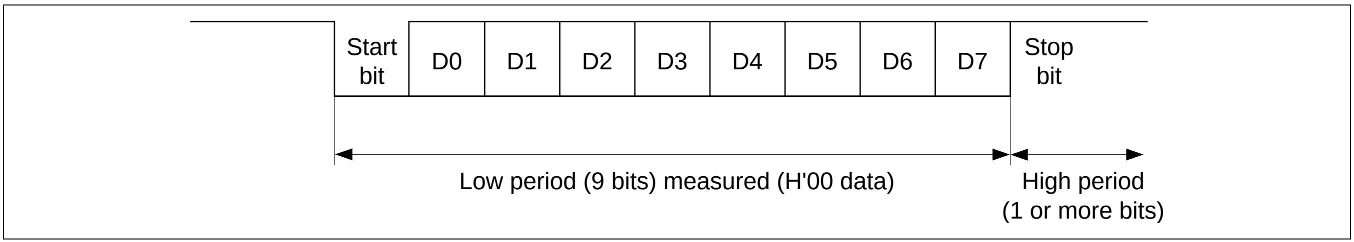


Figure 19-16 Automatic SCI Bit Rate Adjustment

When boot mode is initiated, the H8S/2357 MCU measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The MCU calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the MCU. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host’s transmission bit rate and the MCU’s system clock frequency, there will be a discrepancy between the bit rates of the host and the MCU. To ensure correct SCI operation, the host’s transfer bit rate should be set to (4,800, or 9,600) bps.

Table 19-15 shows typical host transfer bit rates and system clock frequencies for which automatic adjustment of the MCU’s bit rate is possible. The boot program should be executed within this system clock range.

Table 19-15 System Clock Frequencies for which Automatic Adjustment of H8S/2357 Bit Rate is Possible

Host Bit Rate	System Clock Frequency for which Automatic Adjustment of H8S/2357 Bit Rate is Possible
9600 bps	8 to 20 MHz
4800 bps	4 to 20 MHz

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the 2 kbytes area from H'FFDC00 to H'FFE3FF is reserved for use by the boot program, as shown in figure 19-17. The area to which the programming control program is transferred is H'FFE400 to H'FFFB7F. The boot program area can be used when the programming control program transferred into RAM enters the execution state. A stack area should be set up as required.

Section 20 Clock Pulse Generator

20.1 Overview

The H8S/2357 Group has a on-chip clock pulse generator (CPG) that generates the system clock (ø), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator circuit, a duty adjustment circuit, a medium-speed clock divider, and a bus master clock selection circuit.

20.1.1 Block Diagram

Figure 20-1 shows a block diagram of the clock pulse generator.

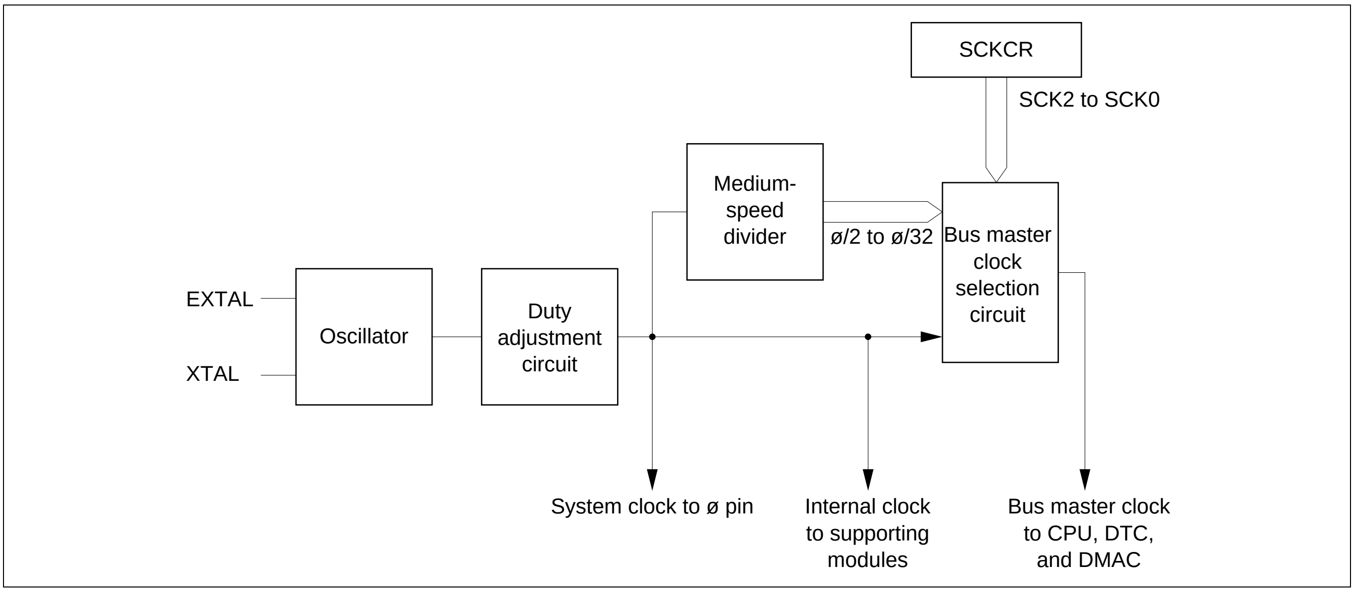


Figure 20-1 Block Diagram of Clock Pulse Generator

20.1.2 Register Configuration

The clock pulse generator is controlled by SCKCR. Table 20-1 shows the register configuration.

Table 20-1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Address*
System clock control register	SCKCR	R/W	H'00	H'FF3A

Note:* Lower 16 bits of the address.

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
JMP	JMP @ERn	2					
	JMP @aa:24	2					1
	JMP @@aa:8 Advanced	2	2				1
JSR	JSR @ERn Advanced	2		2			
	JSR @aa:24 Advanced	2		2			1
	JSR @@aa:8 Advanced	2	2	2			
LDC	LDC #xx:8,CCR	1					
	LDC #xx:8,EXR	2					
	LDC Rs,CCR	1					
	LDC Rs,EXR	1					
	LDC @ERs,CCR	2				1	
	LDC @ERs,EXR	2				1	
	LDC @(d:16,ERs),CCR	3				1	
	LDC @(d:16,ERs),EXR	3				1	
	LDC @(d:32,ERs),CCR	5				1	
	LDC @(d:32,ERs),EXR	5				1	
	LDC @ERs+,CCR	2				1	1
	LDC @ERs+,EXR	2				1	1
	LDC @aa:16,CCR	3				1	
	LDC @aa:16,EXR	3				1	
	LDC @aa:32,CCR	4				1	
	LDC @aa:32,EXR	4				1	
LDM	LDM.L @SP+, (ERn-ERn+1)	2		4			1
	LDM.L @SP+, (ERn-ERn+2)	2		6			1
	LDM.L @SP+, (ERn-ERn+3)	2		8			1
LDMAC	LDMAC ERs,MACH	Cannot be used in the H8S/2357 Group					
	LDMAC ERs,MACL						
MAC	MAC @ERn+,@ERm+	Cannot be used in the H8S/2357 Group					
MOV	MOV.B #xx:8,Rd	1					
	MOV.B Rs,Rd	1					
	MOV.B @ERs,Rd	1			1		
	MOV.B @(d:16,ERs),Rd	2			1		
	MOV.B @(d:32,ERs),Rd	4			1		
	MOV.B @ERs+,Rd	1			1		1
	MOV.B @aa:8,Rd	1			1		
	MOV.B @aa:16,Rd	2			1		
	MOV.B @aa:32,Rd	3			1		
	MOV.B Rs,@ERd	1			1		
	MOV.B Rs,@(d:16,ERd)	2			1		
	MOV.B Rs,@(d:32,ERd)	4			1		

Instruction	H	N	Z	V	C	Definition
DAA	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $Dm \cdot \overline{Rm}$
DIVXS	—	↕	↕	—	—	N = $Sm \cdot \overline{Dm} + \overline{Sm} \cdot Dm$ Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm Z = $\overline{Sm} \cdot \overline{Sm-1} \cdot \cdots\cdots \cdot \overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
INC	—	↕	↕	↕	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm} \cdot Rm$
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC MAC						Cannnot be used in the H8S/2357 Group
MOV	—	↕	↕	0	—	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
MOVFPE MOVTPE						Can not be used in the H8S/2357 Group
MULXS	—	↕	↕	—	—	N = R2m Z = $\overline{R2m} \cdot \overline{R2m-1} \cdot \cdots\cdots \cdot \overline{R0}$
MULXU	—	—	—	—	—	
NEG	↕	↕	↕	↕	↕	H = Dm–4 + Rm–4 N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = Dm · Rm C = Dm + Rm
NOP	—	—	—	—	—	