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.

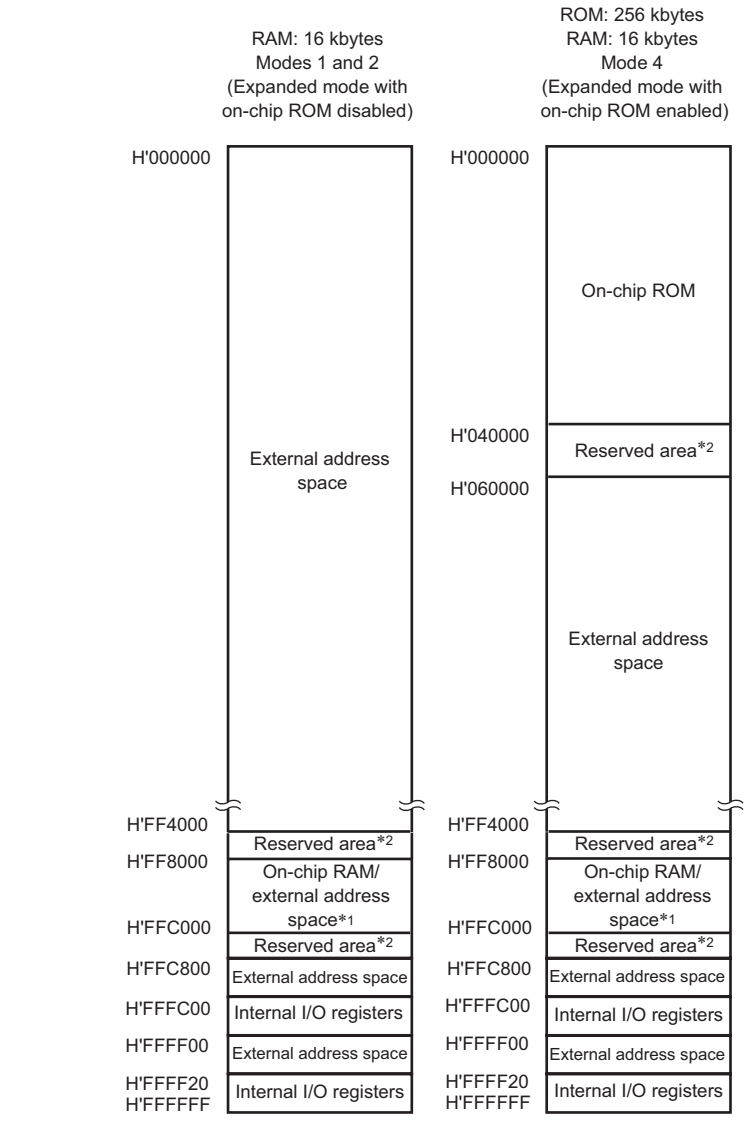
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	H8S/2000
Core Size	16-Bit
Speed	34MHz
Connectivity	I²C, IrDA, SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	96
Program Memory Size	384KB (384K 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/df2374rvfq34v

Table 2.3 Data Transfer Instructions

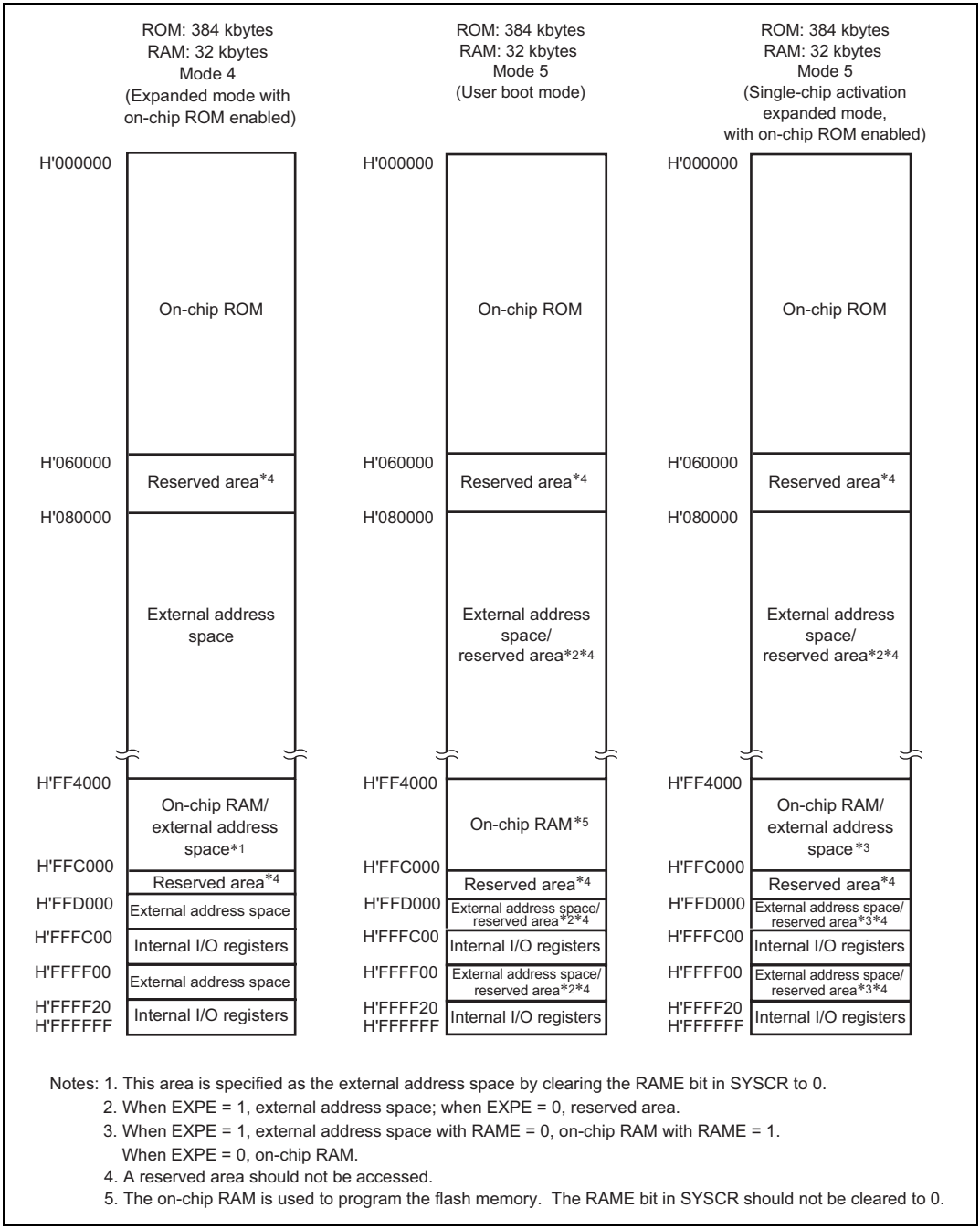
Instruction	Size*	Function
MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
MOVFPE	B	Cannot be used in this LSI.
MOVTPE	B	Cannot be used in this LSI.
POP	W/L	@SP+ → Rn Pops a general register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn
PUSH	W/L	Rn → @-SP Pushes a general register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @-SP. PUSH.L ERn is identical to MOV.L ERn, @-SP.
LDM	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
STM	L	Rn (register list) → @-SP Pushes two or more general registers onto the stack.

Note: * Size refers to the operand size.
B: Byte
W: Word
L: Longword



Notes: 1. This area is specified as external address space by clearing the RAME bit in SYSCR to 0.
2. A reserved area should not be accessed.

Figure 3.5 Memory Map for H8S/2375 and H8S/2375R (1)



Notes: 1. This area is specified as the external address space by clearing the RAME bit in SYSCR to 0.
2. When EXPE = 1, external address space; when EXPE = 0, reserved area.
3. When EXPE = 1, external address space with RAME = 0, on-chip RAM with RAME = 1.
When EXPE = 0, on-chip RAM.
4. A reserved area should not be accessed.
5. The on-chip RAM is used to program the flash memory. The RAME bit in SYSCR should not be cleared to 0.

Figure 3.8 Memory Map for H8S/2374 and H8S/2374R (2)

7.5.16 Clearing Full Address Mode

Figure 7.37 shows the procedure for releasing and initializing a channel designated for full address mode. After full address mode has been cleared, the channel can be set to another transfer mode using the appropriate setting procedure.

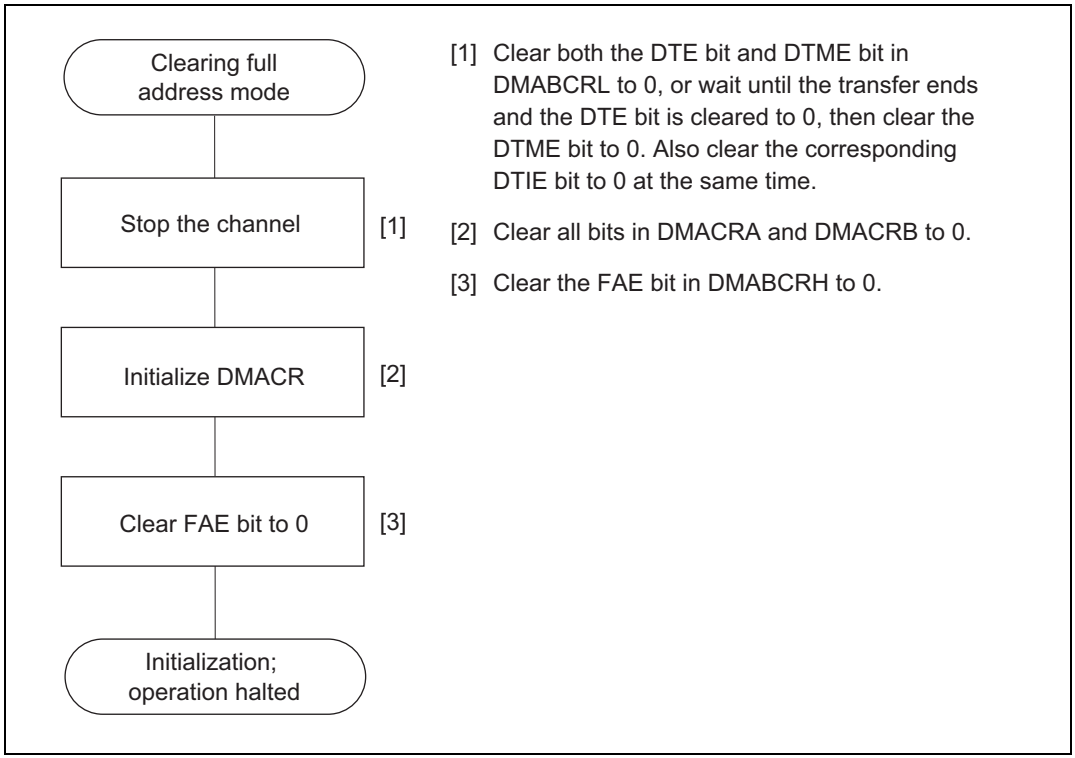


Figure 7.37 Example of Procedure for Clearing Full Address Mode

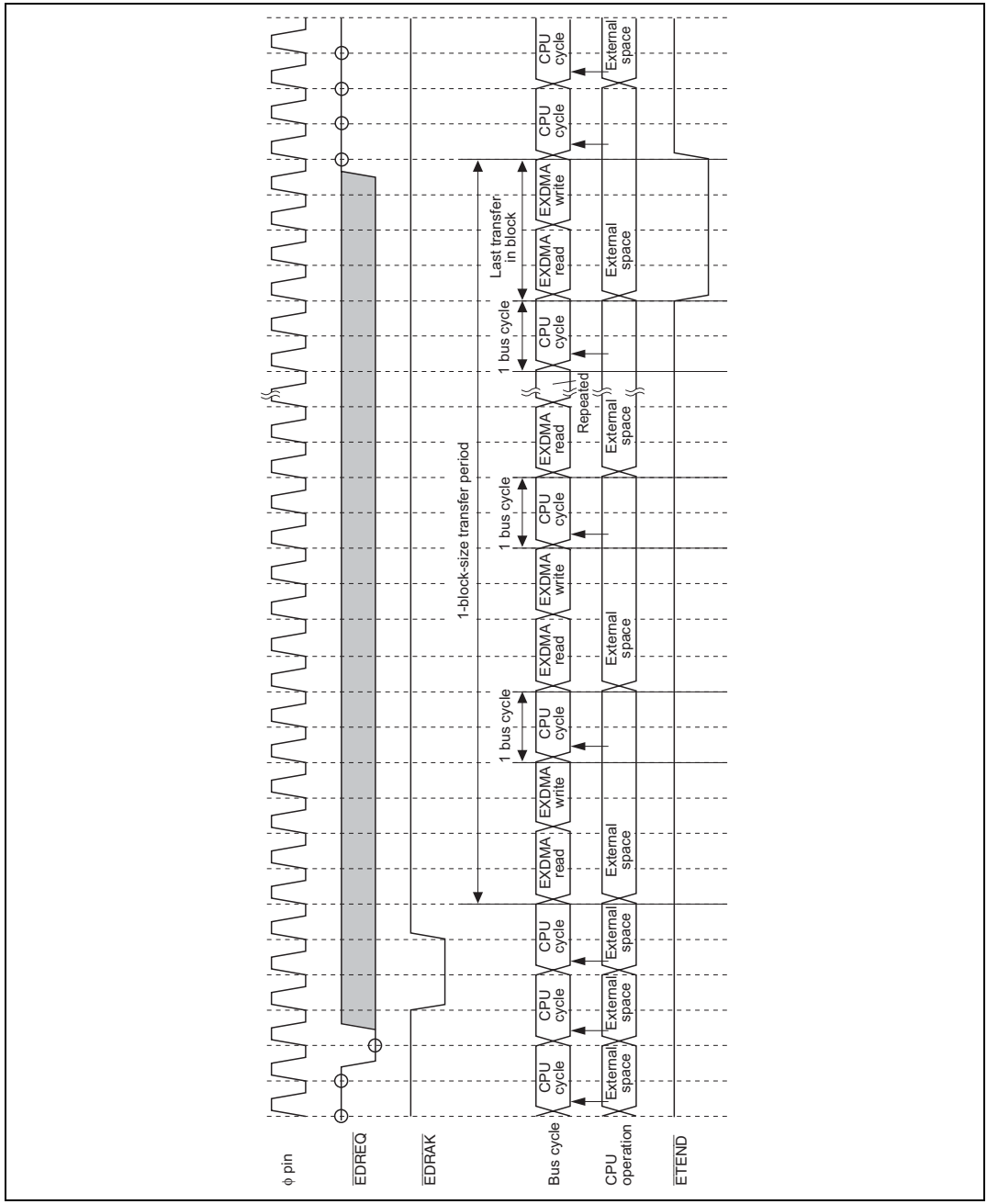


Figure 8.42 External Request/Cycle Steal Mode/Block Transfer Mode (CPU Cycles/Dual Address Mode/Low Level Sensing/BGUP = 1)

9.3 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI_0.

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

Table 9.1 shows a relationship between activation sources and DTCER clear conditions. Figure 9.2 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

Table 9.1 Relationship between Activation Sources and DTCER Clearing

Activation Source	DISEL = 0 and Specified Number of Transfers Has Not Ended	DISEL = 1 or Specified Number of Transfers Has Ended
Activation by software	SWDTE bit is cleared to 0	• SWDTE bit remains set to 1 • Interrupt request to CPU
Activation by an interrupt	• Corresponding DTCER bit remains set to 1. • Activation source flag is cleared to 0.	• Corresponding DTCER bit is cleared to 0. • Activation source flag remains set to 1. • Interrupt that became the activation source is requested to the CPU.

10.7 Port 8

Port 8 is a 6-bit I/O port that also has other functions. The port 8 has the following registers.

- Port 8 data direction register (P8DDR)
- Port 8 data register (P8DR)
- Port 8 register (PORT8)

10.7.1 Port 8 Data Direction Register (P8DDR)

The individual bits of P8DDR specify input or output for the pins of port 8.

P8DDR cannot be read; if it is, an undefined value will be read.

Bit	Bit Name	Initial Value	R/W	Description
7, 6	—	0	—	Reserved These bits are always read as 0 and cannot be modified.
5	P85DDR	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.
4	P84DDR	0	W	
3	P83DDR	0	W	
2	P82DDR	0	W	
1	P81DDR	0	W	
0	P80DDR	0	W	

10.13 Port E

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

- Port E data direction register (PEDDR)
- Port E data register (PEDR)
- Port E register (PORTE)
- Port E pull-up MOS control register (PEPCR)

10.13.1 Port E Data Direction Register (PEDDR)

The individual bits of PEDDR specify input or output for the pins of port E.

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

Bit	Bit Name	Initial Value	R/W	Description
7	PE7DDR	0	W	• Modes 1, 2, and 4 When 8-bit bus mode is selected, port E functions as an I/O port. The pin states can be changed with PEDDR. When 16-bit bus mode is selected, port E is designated for data input/output. For details on 8-bit and 16-bit bus modes, see section 6, Bus Controller (BSC). • Mode 7 (when EXPE = 1) When 8-bit bus mode is selected, port E functions as an I/O port. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port. When 16-bit bus mode is selected, port E is designated for data input/output. • Mode 7 (when EXPE = 0) Port E is an I/O port, and its pin functions can be switched with PEDDR.
6	PE6DDR	0	W	
5	PE5DDR	0	W	
4	PE4DDR	0	W	
3	PE3DDR	0	W	
2	PE2DDR	0	W	
1	PE1DDR	0	W	
0	PE0DDR	0	W	

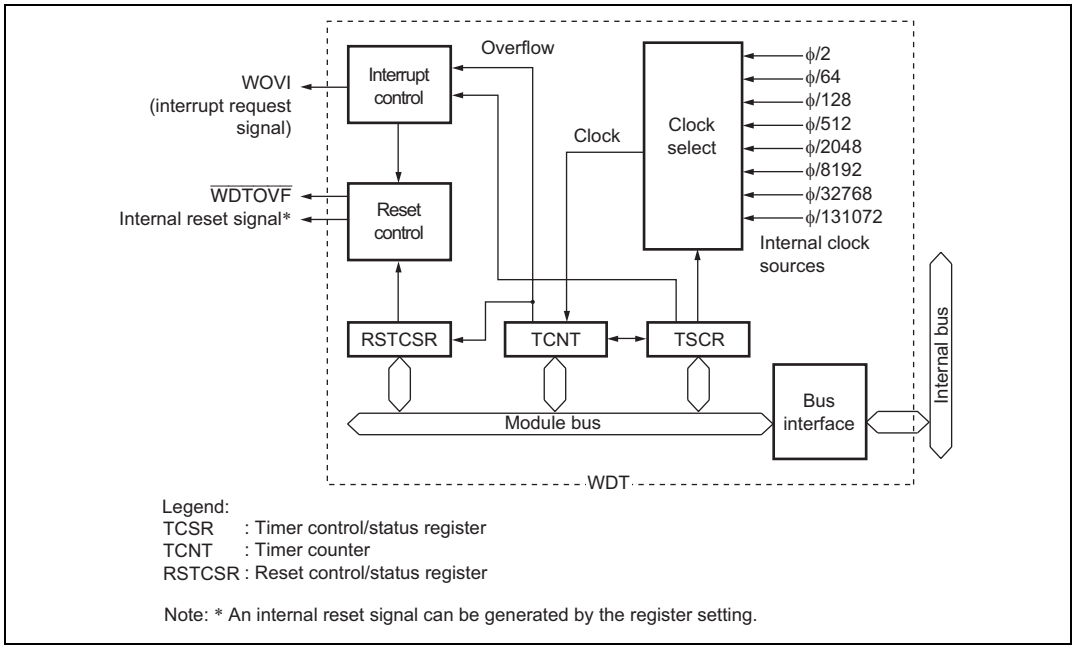


Figure 14.1 Block Diagram of WDT

14.2 Input/Output Pin

Table 14.1 shows the WDT pin configuration.

Table 14.1 Pin Configuration

Name	Symbol	I/O	Function
Watchdog timer overflow	$\overline{\text{WDTOVF}}$	Output	Outputs counter overflow signal in watchdog timer mode

15.3.11 Serial Extension Mode Register (SEMR)

SEMR selects the clock source in asynchronous mode. The basic clock can be automatically set by selecting the average transfer rate.

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	—	Undefined	—	Reserved If these bits are read, an undefined value will be returned and cannot be modified.
3	ABCS	0	R/W	Asynchronous basic clock selection (valid only in asynchronous mode) Selects the basic clock for 1-bit period in asynchronous mode. 0: Operates on a basic clock with a frequency of 16 times the transfer rate. 1: Operates on a basic clock with a frequency of 8 times the transfer rate.

Table 15.11 shows the states of the SSR status flags and receive data handling when a receive error is detected. If a receive error is detected, the RDRF flag retains its state before receiving data. Reception cannot be resumed while a receive error flag is set to 1. Accordingly, clear the ORER, FER, PER, and RDRF bits to 0 before resuming reception. Figure 15.9 shows a sample flowchart for serial data reception.

Table 15.11 SSR Status Flags and Receive Data Handling

SSR Status Flag					
RDRF*	**ORER**	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	1	0	0	Lost	Overrun error
0	0	1	0	Transferred to RDR	Framing error
0	0	0	1	Transferred to RDR	Parity error
1	1	1	0	Lost	Overrun error + framing error
1	1	0	1	Lost	Overrun error + parity error
0	0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	1	Lost	Overrun error + framing error + parity error

Note: * The RDRF flag retains its state before data reception.

18.3 Register Descriptions

The D/A converter has the following registers.

- D/A data register 0 (DADR0)*
- D/A data register 1 (DADR1)*
- D/A data register 2 (DADR2)
- D/A data register 3 (DADR3)
- D/A data register 4 (DADR4)*
- D/A data register 5 (DADR5)*
- D/A control register 01 (DACR01)*
- D/A control register 23 (DACR23)
- D/A control register 45 (DACR45)*

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

18.3.1 D/A Data Registers 0 to 5 (DADR0 to DADR5)

DADR0 to DADR5 are 8-bit readable/writable registers that store data for conversion.

Whenever output is enabled, the values in DADR are converted and output to the analog output pins.

When the H8S/2375, H8S/2375R, H8S/2373, or H8S/2373R is in use, the registers which are not supported must not be accessed.

18.3.2 D/A Control Registers 01, 23, and 45 (DACR01, DACR23, DACR45)

DACR01, DACR23, and DACR45 control the operation of the D/A converter. DACR01, DACR23, and DACR45 control the operation of channels 0 and 1, channels 2 and 3, and channels 4 and 5, respectively.

(2) Programming Procedure in User Program Mode

The procedures for download, initialization, and programming are shown in figure 21.11.

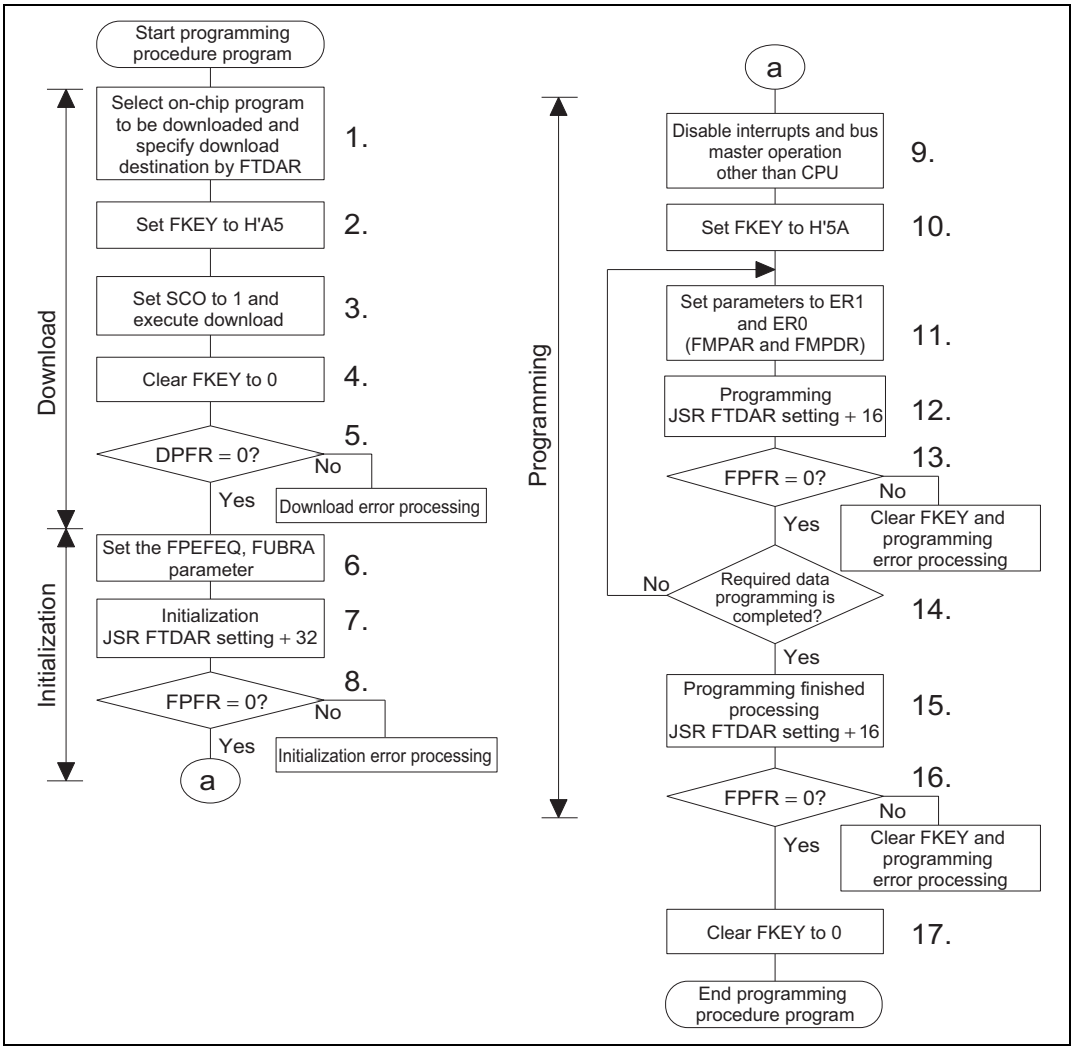


Figure 21.11 Programming Procedure

The procedure program must be executed in an area other than the flash memory to be programmed. Especially the part where the SCO bit in FCCS is set to 1 for downloading must be executed in the on-chip RAM.

The area that can be executed in the steps of the user procedure program (on-chip RAM, user MAT, and external space) is shown in section 21.4.4, Procedure Program and Storable Area for Programming Data.

21.4.4 Procedure Program and Storable Area for Programming Data

In the descriptions in the previous section, the programming/erasing procedure programs and storable areas for program data are assumed to be in the on-chip RAM. However, the program and the data can be stored in and executed from other areas, such as part of flash memory which is not to be programmed or erased, or somewhere in the external address space.

(1) Conditions that Apply to Programming/Erasing

1. The on-chip programming/erasing program is downloaded from the address in the on-chip RAM specified by FTDAR, therefore, this area is not available for use.
2. The on-chip programming/erasing program will use the 128 bytes as a stack. So, make sure that this area is secured.
3. Download by setting the SCO bit to 1 will lead to switching of the MAT. If, therefore, this operation is used, it should be executed from the on-chip RAM.
4. The flash memory is accessible until the start of programming or erasing, that is, until the result of downloading has been determined. When in a mode in which the external address space is not accessible, such as single-chip mode, the required procedure programs, NMI handling vector and NMI handler should be transferred to the on-chip RAM before programming/erasing of the flash memory starts.
5. The flash memory is not accessible during programming/erasing operations, therefore, the operation program is downloaded to the on-chip RAM to be executed. The NMI-handling vector and programs such as that which activate the operation program, and NMI handler should thus be stored in on-chip memory other than flash memory or the external address space.
6. After programming/erasing, the flash memory should be inhibited until FKEY is cleared.

   The reset state ($\overline{RES}$ = 0) must be in place for more than 100 μs when the LSI mode is changed to reset on completion of a programming/erasing operation.

   Transitions to the reset state, and hardware standby mode are inhibited during programming/erasing. When the reset signal is accidentally input to the chip, a longer period in the reset state than usual (100 μs) is needed before the reset signal is released.
7. Switching of the MATs by FMATS should be needed when programming/erasing of the user boot MAT is operated in user-boot mode. The program which switches the MATs should be executed from the on-chip RAM. See section 21.6, Switching between User MAT and User

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.

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
Execution of Erasure	○	×	×	○		
Determination of Erasure Result	○	×	○	○		
Operation for Erasure Error	○	×*	○	○		
Operation for FKEY Clear	○	×	○	○		
Switching MATs by FMATS	○	×	×	○		

Note: * Switching FMATS by a program in the on-chip RAM enables this area to be used.

21.7 Programmer Mode

Along with its on-board programming mode, this LSI also has a PROM mode as a further mode for the writing and erasing of programs and data. In the PROM mode, a general-purpose PROM programmer can freely be used to write programs to the on-chip ROM. Program/erase is possible on the user MAT and user boot MAT. The PROM programmer must support Renesas microcomputers with 512-kbyte flash memory as a device type.

A status-polling system is adopted for operation in automatic program, automatic erase, and status-read modes. In the status-read mode, details of the system's internal signals are output after execution of automatic programming or automatic erasure. In the PROM mode, provide a 12-MHz input-clock signal.

21.8 Serial Communication Interface Specification for Boot Mode

Initiating boot mode enables the boot program to communicate with the host by using the internal SCI. The serial communication interface specification is shown below.

(1) Status

The boot program has three states.

1. Bit-Rate-Adjustment State

   In this state, the boot program adjusts the bit rate to communicate with the host. Initiating boot mode enables starting of the boot program and entry to the bit-rate-adjustment state. The program receives the command from the host to adjust the bit rate. After adjusting the bit rate, the program enters the inquiry/selection state.
2. Inquiry/Selection State

   In this state, the boot program responds to inquiry commands from the host. The device name, clock mode, and bit rate are selected. After selection of these settings, the program is made to enter the programming/erasing state by the command for a transition to the programming/erasing state. The program transfers the libraries required for erasure to the RAM and erases the user MATs and user boot MATs before the transition.
3. Programming/erasing state

   Programming and erasure by the boot program take place in this state. The boot program is made to transfer the programming/erasing programs to the RAM by commands from the host. Sum checks and blank checks are executed by sending these commands from the host.

These boot program states are shown in figure 21.17.

Section 24 Power-Down Modes

In addition to the normal program execution state, this LSI has power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip peripheral modules, and so on.

This LSI's operating modes are high-speed mode and six power down modes:

- Clock division mode
- Sleep mode
- Module stop mode
- All module clock stop mode
- Software standby mode
- Hardware standby mode

Sleep mode is a CPU state, clock division mode is an on-chip peripheral function (including bus masters and the CPU) state, and module stop mode is an on-chip peripheral function (including bus masters other than the CPU) state. A combination of these modes can be set.

After a reset, this LSI is in high-speed mode.

Table 24.1 shows the internal states of this LSI in each mode. Figure 24.1 shows the mode transition diagram.

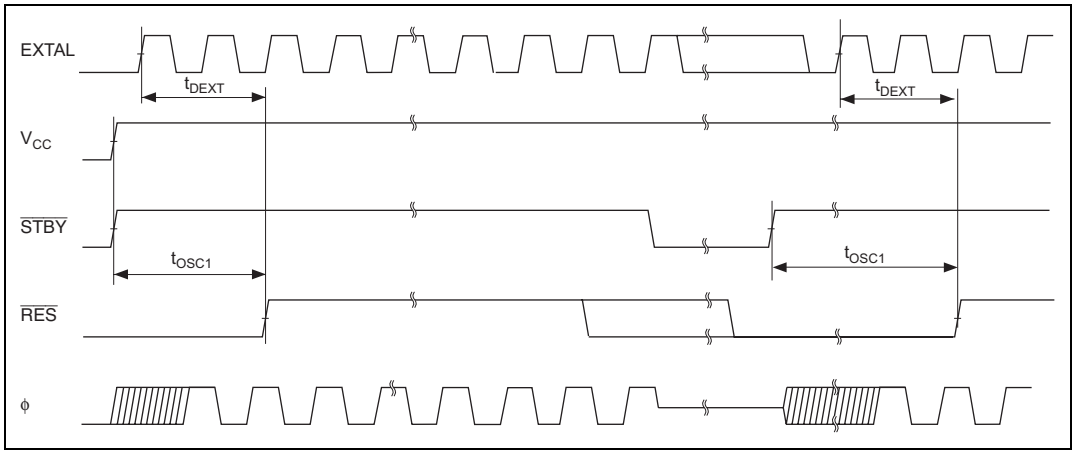


Figure 26.4 (1) Oscillation Settling Timing

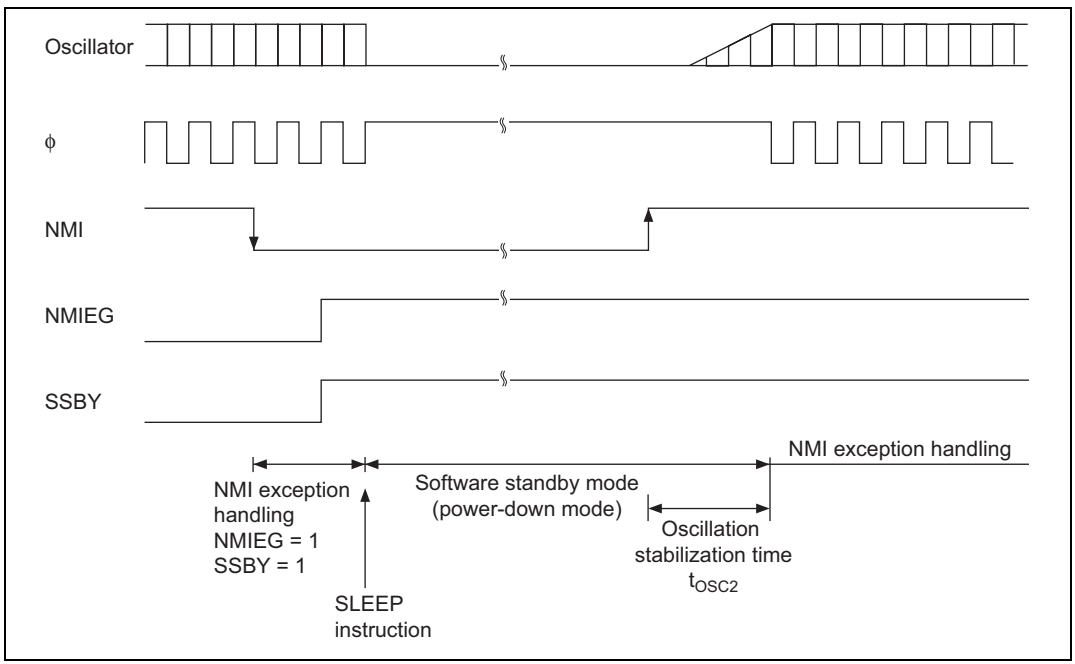
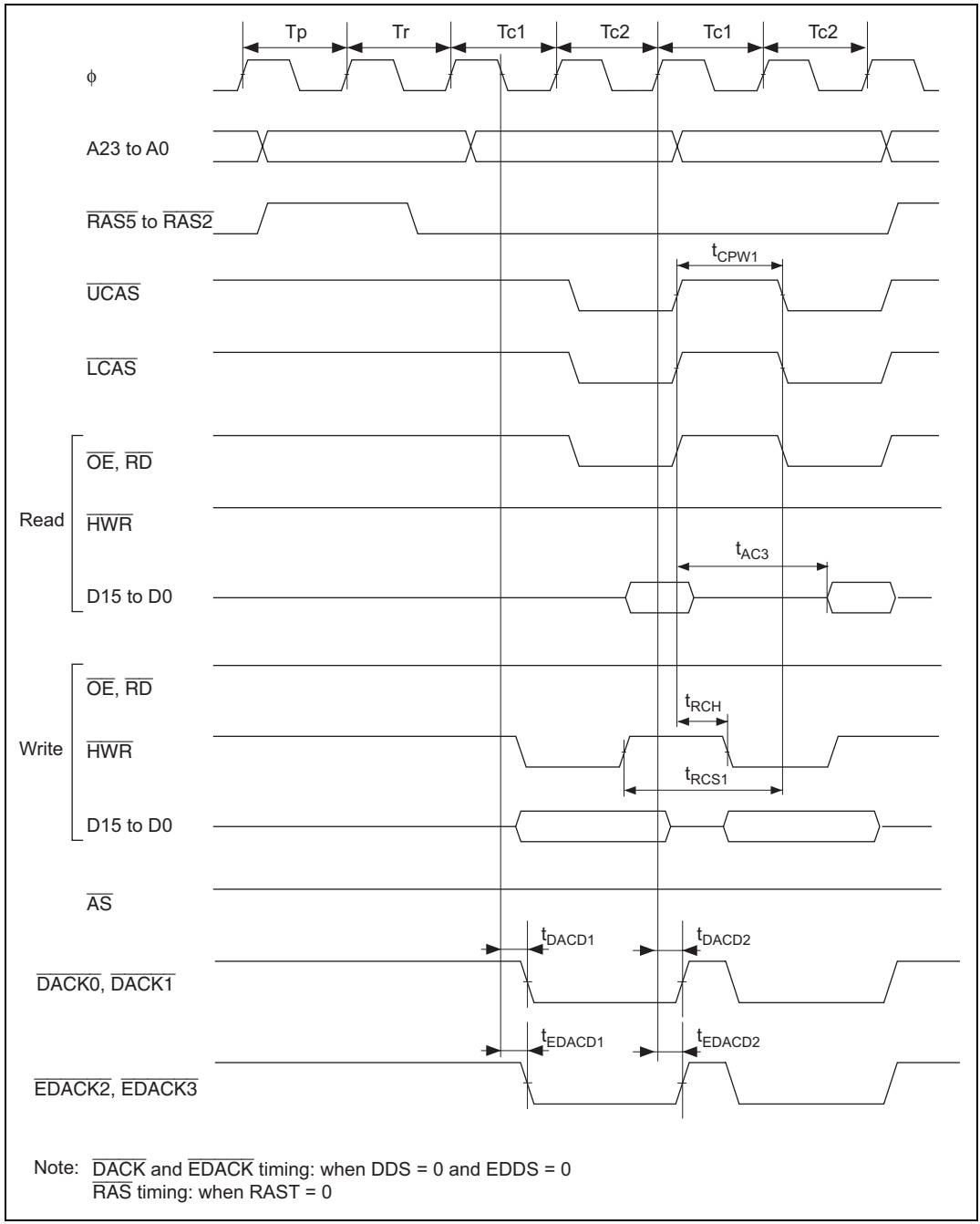


Figure 26.4 (2) Oscillation Settling Timing



Note: DACK and EDACK timing: when DDS = 0 and EDDS = 0
RAS timing: when RAST = 0

Figure 26.16 DRAM Access Timing: Two-State Burst Access