**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/2600 |
| Core Size | 16-Bit |
| Speed | 20MHz |
| Connectivity | CANbus, SCI, SmartCard |
| Peripherals | Motor Control PWM, POR, PWM, WDT |
| Number of I/O | 72 |
| Program Memory Size | 128KB (128K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 4.5V ~ 5.5V |
| Data Converters | A/D 12x10b; D/A 2x8b |
| Oscillator Type | Internal |
| Operating Temperature | -20°C ~ 75°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 128-BFQFP |
| Supplier Device Package | 128-QFP (14x20) |
| Purchase URL | https://www.e-xfl.com/product-detail/renesas-electronics-america/df2636f20v |

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

| Type | Symbol | I/O | Name and Function |
|---|---|---|---|
| D/A converter | DA1, DA0*5 | Output | Analog output: Analog output pins for D/A converter. |
| A/D converter, D/A converter | AVCC | Input | Analog power supply: A/D converter and D/A converter power supply pin.<br>When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+5 V). |
| | AVSS | Input | Analog ground: Ground pin for A/D converter and D/A converter.<br>Connect to system power supply (0 V). |
| | Vref | Input | Analog reference power supply: A/D converter and D/A converter reference voltage input pin.<br>When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+5 V). |
| I/O ports | P17 to P10 | I/O | Port 1: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 1 data direction register (P1DDR). |
| | P35 to P30 | I/O | Port 3: A 6-bit I/O port. Input or output can be designated for each bit by means of the port 3 data direction register (P3DDR). |
| | P47 to P40 | Input | Port 4: An 8-bit input port. |
| | P93 to P90 | Input | Port 9: A 4-bit input port. |
| | PA3 to PA0 | I/O | Port A: A 4-bit I/O port. Input or output can be designated for each bit by means of the port A data direction register (PADDR). |
| | PB7 to PB0 | I/O | Port B: An 8-bit I/O port. Input or output can be designated for each bit by means of the port B data direction register (PBDDR). |
| | PC7 to PC0 | I/O | Port C: An 8-bit I/O port. Input or output can be designated for each bit by means of the port C data direction register (PCDDR). |
| | PD7 to PD0 | I/O | Port D: An 8-bit I/O port. Input or output can be designated for each bit by means of the port D data direction register (PDDDR). |
| | PE7 to PE0 | I/O | Port E: An 8-bit I/O port. Input or output can be designated for each bit by means of the port E data direction register (PEDDR). |

| Function | Instruction | Addressing Modes | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | #xx | Rn | @ERn | @(d:16,ERn) | @(d:32,ERn) | @–ERn/@ERn+ | @aa:8 | @aa:16 | @aa:24 | @aa:32 | @(d:8,PC) | @(d:16,PC) | @@aa:8 | — |
| Logic operations | AND, OR, XOR | BWL | BWL | — | — | — | — | — | — | — | — | — | — | — | — |
| | NOT | — | BWL | — | — | — | — | — | — | — | — | — | — | — | — |
| Shift | | — | BWL | — | — | — | — | — | — | — | — | — | — | — | — |
| Bit manipulation | | — | B | B | — | — | — | B | B | — | B | — | — | — | — |
| Branch | Bcc, BSR | — | — | — | — | — | — | — | — | — | — | ○ | ○ | — | — |
| | JMP, JSR | — | — | — | — | — | — | — | — | ○ | — | — | — | ○ | — |
| | RTS | — | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| System control | TRAPA | — | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| | RTE | — | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| | SLEEP | — | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| | LDC | B | B | W | W | W | W | — | W | — | W | — | — | — | — |
| | STC | — | B | W | W | W | W | — | W | — | W | — | — | — | — |
| | ANDC, ORC, XORC | B | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | NOP | — | — | — | — | — | — | — | — | — | — | — | — | — | ○ |
| Block data transfer | | — | — | — | — | — | — | — | — | — | — | — | — | — | BW |

Legend:
B: Byte
W: Word
L: Longword

Notes: 1. Not available in the chip.
2. Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
3. Only registers ER0 to ER6 should be used when using the STM/LDM instruction.

### 6.3.7 Additional Notes

(1) When a PC break is set for an instruction fetch at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction:

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction is fetched, it is not executed, and so a PC break interrupt is not generated by the instruction fetch at the next address.

(2) When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interrupt becomes valid two states after the end of the executing instruction. If a PC break interrupt is set for the instruction following one of these instructions, since interrupts, including NMI, are disabled for a 3-state period in the case of LDC, ANDC, ORC, and XORC, the next instruction is always executed. For details, see section 5, Interrupt Controller.

(3) When a PC break is set for an instruction fetch at the address following a Bcc instruction:

A PC break interrupt is generated if the instruction at the next address is executed in accordance with the branch condition, but is not generated if the instruction at the next address is not executed.

(4) When a PC break is set for an instruction fetch at the branch destination address of a Bcc instruction:

A PC break interrupt is generated if the instruction at the branch destination is executed in accordance with the branch condition, but is not generated if the instruction at the branch destination is not executed.

# Section 10 16-Bit Timer Pulse Unit (TPU)

Note: The H8S/2635 Group is not equipped with a DTC or a PPG.

## 10.1 Overview

The chip has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels.

### 10.1.1 Features

- Maximum 16-pulse input/output
  - A total of 16 timer general registers (TGRs) are provided (four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5), each of which can be set independently as an output compare/input capture register
  - TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - Waveform output at compare match: Selection of 0, 1, or toggle output
  - Input capture function: Selection of rising edge, falling edge, or both edge detection
  - Counter clear operation: Counter clearing possible by compare match or input capture
  - Synchronous operation: Multiple timer counters (TCNT) can be written to simultaneously
  - Simultaneous clearing by compare match and input capture possible
  - Register simultaneous input/output possible by counter synchronous operation
  - PWM mode: Any PWM output duty can be set
  - Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
  - Input capture register double-buffering possible
  - Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel 1 (channel 4) overflow/underflow
- Fast access via internal 16-bit bus
  - Fast access is possible via a 16-bit bus interface

**Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0):** These bits select the compare match that triggers pulse output group 2 (pins PO11 to PO8).

| | | Description | |
|---|---|---|---|
| **Bit 5 G2CMS1** | **Bit 4 G2CMS0** | **Output Trigger for Pulse Output Group 2** | |
| 0 | 0 | Compare match in TPU channel 0 | |
| | 1 | Compare match in TPU channel 1 | |
| 1 | 0 | Compare match in TPU channel 2 | |
| | 1 | Compare match in TPU channel 3 | (Initial value) |

**Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0):** These bits select the compare match that triggers pulse output group 1 (pins PO7 to PO4). However, the chip has no output pins corresponding to pulse output group 1.

| | | Description | |
|---|---|---|---|
| **Bit 3 G1CMS1** | **Bit 2 G1CMS0** | **Output Trigger for Pulse Output Group 1** | |
| 0 | 0 | Compare match in TPU channel 0 | |
| | 1 | Compare match in TPU channel 1 | |
| 1 | 0 | Compare match in TPU channel 2 | |
| | 1 | Compare match in TPU channel 3 | (Initial value) |

**Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0):** These bits select the compare match that triggers pulse output group 0 (pins PO3 to PO0). However, the chip has no output pins corresponding to pulse output group 0.

| | | Description | |
|---|---|---|---|
| **Bit 1 G0CMS1** | **Bit 0 G0CMS0** | **Output Trigger for Pulse Output Group 0** | |
| 0 | 0 | Compare match in TPU channel 0 | |
| | 1 | Compare match in TPU channel 1 | |
| 1 | 0 | Compare match in TPU channel 2 | |
| | 1 | Compare match in TPU channel 3 | (Initial value) |

### 12.5.2 Changing Value of PSS* and CKS2 to CKS0

If bits PSS and CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits PSS* and CKS2 to CKS0.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions, and H8S/2635 Group only.
These functions cannot be used with the other versions.

### 12.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

### 12.5.4 Internal Reset in Watchdog Timer Mode

The chip is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, but TCNT and TSCR of the WDT are reset.

### 12.5.5 OVF Flag Clearing in Interval Timer Mode

If conflict occurs between OVF flag clearing and OVF flag reading in interval timer mode, the flag may not be cleared by writing 0 to OVF even though the OVF = 1 state has been read. When interval timer interrupts are disabled and the OVF flag is polled, for instance, and there is a possibility of conflict between OVF flag setting and reading, the OVF = 1 state should be read at least twice before writing 0 to OVF in order to clear the flag.

### 13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the SCI.

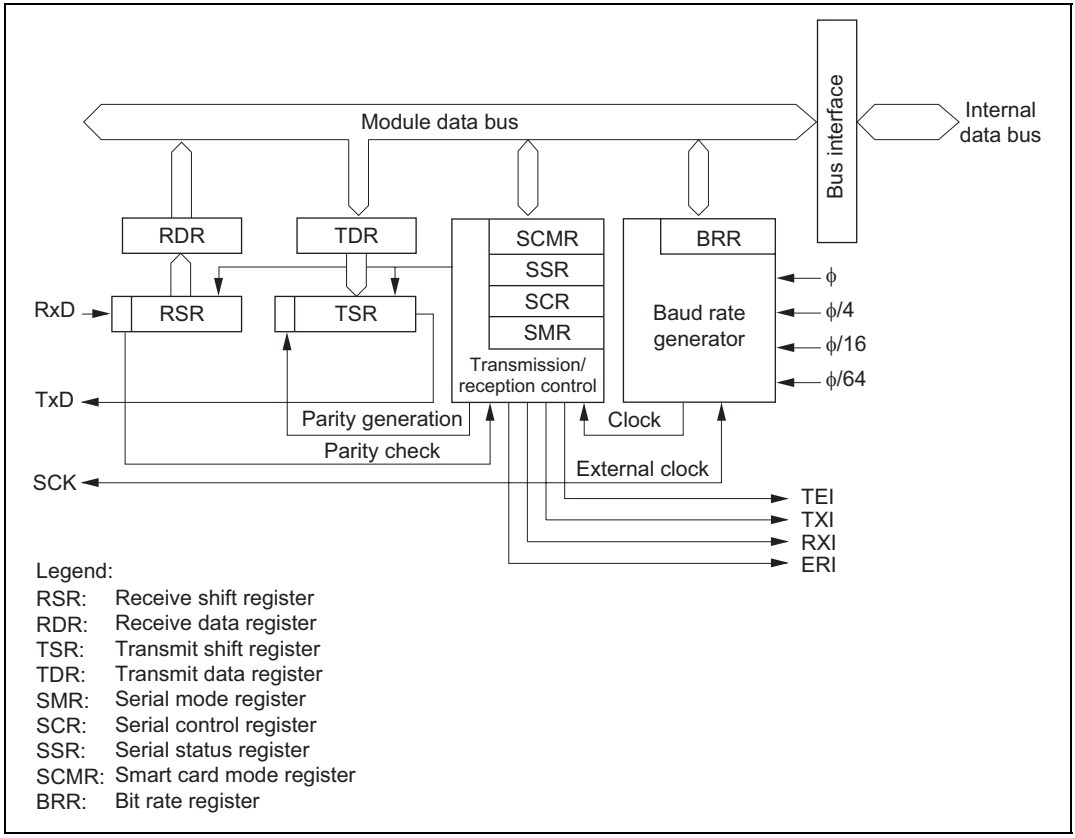



**Figure 13-1 Block Diagram of SCI**

| | φ = 6 MHz | | | φ = 6.144 MHz | | | φ = 7.3728 MHz | | | φ = 8 MHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bit Rate (bit/s)** | **n** | **N** | **Error (%)** | **n** | **N** | **Error (%)** | **n** | **N** | **Error (%)** | **n** | **N** | **Error (%)** |
| 110 | 2 | 106 | –0.44 | 2 | 108 | 0.08 | 2 | 130 | –0.07 | 2 | 141 | 0.03 |
| 150 | 2 | 77 | 0.16 | 2 | 79 | 0.00 | 2 | 95 | 0.00 | 2 | 103 | 0.16 |
| 300 | 1 | 155 | 0.16 | 1 | 159 | 0.00 | 1 | 191 | 0.00 | 1 | 207 | 0.16 |
| 600 | 1 | 77 | 0.16 | 1 | 79 | 0.00 | 1 | 95 | 0.00 | 1 | 103 | 0.16 |
| 1200 | 0 | 155 | 0.16 | 0 | 159 | 0.00 | 0 | 191 | 0.00 | 0 | 207 | 0.16 |
| 2400 | 0 | 77 | 0.16 | 0 | 79 | 0.00 | 0 | 95 | 0.00 | 0 | 103 | 0.16 |
| 4800 | 0 | 38 | 0.16 | 0 | 39 | 0.00 | 0 | 47 | 0.00 | 0 | 51 | 0.16 |
| 9600 | 0 | 19 | –2.34 | 0 | 19 | 0.00 | 0 | 23 | 0.00 | 0 | 25 | 0.16 |
| 19200 | 0 | 9 | –2.34 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 12 | 0.16 |
| 31250 | 0 | 5 | 0.00 | 0 | 5 | 2.40 | — | — | — | 0 | 7 | 0.00 |
| 38400 | 0 | 4 | –2.34 | 0 | 4 | 0.00 | 0 | 5 | 0.00 | — | — | — |

| | φ = 9.8304 MHz | | | φ = 10 MHz | | | φ = 12 MHz | | | φ = 12.288 MHz | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Bit Rate (bit/s)** | **n** | **N** | **Error (%)** | **n** | **N** | **Error (%)** | **n** | **N** | **Error (%)** | **n** | **N** | **Error (%)** |
| 110 | 2 | 174 | –0.26 | 2 | 177 | –0.25 | 2 | 212 | 0.03 | 2 | 217 | 0.08 |
| 150 | 2 | 127 | 0.00 | 2 | 129 | 0.16 | 2 | 155 | 0.16 | 2 | 159 | 0.00 |
| 300 | 1 | 255 | 0.00 | 2 | 64 | 0.16 | 2 | 77 | 0.16 | 2 | 79 | 0.00 |
| 600 | 1 | 127 | 0.00 | 1 | 129 | 0.16 | 1 | 155 | 0.16 | 1 | 159 | 0.00 |
| 1200 | 0 | 255 | 0.00 | 1 | 64 | 0.16 | 1 | 77 | 0.16 | 1 | 79 | 0.00 |
| 2400 | 0 | 127 | 0.00 | 0 | 129 | 0.16 | 0 | 155 | 0.16 | 0 | 159 | 0.00 |
| 4800 | 0 | 63 | 0.00 | 0 | 64 | 0.16 | 0 | 77 | 0.16 | 0 | 79 | 0.00 |
| 9600 | 0 | 31 | 0.00 | 0 | 32 | –1.36 | 0 | 38 | 0.16 | 0 | 39 | 0.00 |
| 19200 | 0 | 15 | 0.00 | 0 | 15 | 1.73 | 0 | 19 | –2.34 | 0 | 19 | 0.00 |
| 31250 | 0 | 9 | –1.70 | 0 | 9 | 0.00 | 0 | 11 | 0.00 | 0 | 11 | 2.40 |
| 38400 | 0 | 7 | 0.00 | 0 | 7 | 1.73 | 0 | 9 | –2.34 | 0 | 9 | 0.00 |

### 15.2.7 Serial Control Register X (SCRX)

| Bit | : | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| | | — | IICX1 | IICX0 | IICE | — | — | — | — |
| Initial value | : | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | : | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W |

SCRX is an 8-bit readable/writable register that controls register access, the $I^2C$ interface operating mode. If a module controlled by SCRX is not used, do not write 1 to the corresponding bit.

SCRX is initialized to H'08 by a reset and in hardware standby mode.

**Bit 7—Reserved:** Do not set 1.

**Bit 6—$I^2C$ Transfer Select 1 (IICX1):** This bit, together with bits CKS2 to CKS0 in ICMR of IIC1, selects the transfer rate in master mode. For details, see section 15.2.4, $I^2C$ Bus Mode Register (ICMR).

**Bit 5—$I^2C$ Transfer Select 0 (IICX0):** This bit, together with bits CKS2 to CKS0 in ICMR of IIC0, selects the transfer rate in master mode. For details, see section 15.2.4, $I^2C$ Bus Mode Register (ICMR).

**Bit 4—$I^2C$ Master Enable (IICE):** Controls CPU access to the $I^2C$ bus interface data and control registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR).

**Bit 4**

| IICE | Description | |
|---|---|---|
| 0 | CPU access to $I^2C$ bus interface data and control registers is disabled | (Initial value) |
| 1 | CPU access to $I^2C$ bus interface data and control registers is enabled | |

**Bit 3— Reserved:** Always returns a value of 1 if it is read.

**Bits 2 to 0—Reserved:** Do not set 1.

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

[1] Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

[2] When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the 8th data bit ($R/\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRF flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

[3] After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 15-18.

[4] When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.

[5] To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps [4] and [5]. To end transmission, write H'FF to ICDR to release SDA on the slave side. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

**Bit 11—Transmit Overload Warning Interrupt Flag (IRR3):** Status flag indicating the error warning state caused by the transmit error counter.

| Bit 11: IRR3 | Description |
|---|---|
| 0 | [Clearing condition]<br>• Writing 1 (Initial value) |
| 1 | Error warning state caused by transmit error<br>[Setting condition]<br>• When TEC ≥ 96 |

**Bit 10—Remote Frame Request Interrupt Flag (IRR2):** Status flag indicating that a remote frame has been received in a mailbox (buffer).

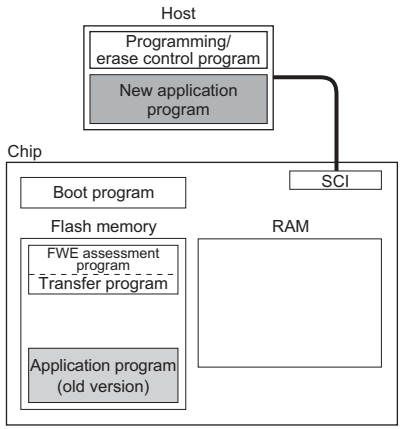
| Bit 10: IRR2 | Description |
|---|---|
| 0 | [Clearing condition]<br>• Clearing of all bits in RFPR (remote request register) of mailbox for which receive interrupt requests are enabled by MBIMR (Initial value) |
| 1 | Remote frame received and stored in mailbox<br>[Setting condition]<br>• When remote frame reception is completed, when corresponding MBIMR = 0 |

**Bit 9—Receive Message Interrupt Flag (IRR1):** Status flag indicating that a mailbox (buffer) receive message has been received normally.

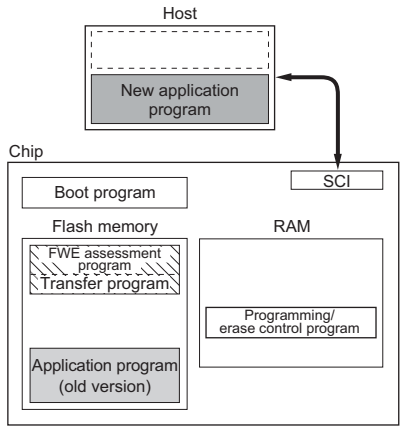
| Bit 9: IRR1 | Description |
|---|---|
| 0 | [Clearing condition]<br>• Clearing of all bits in RXPR (receive complete register) of mailbox for which receive interrupt requests are enabled by MBIMR (Initial value) |
| 1 | Data frame or remote frame received and stored in mailbox<br>[Setting condition]<br>• When data frame or remote frame reception is completed, when corresponding MBIMR = 0 |

## User Program Mode

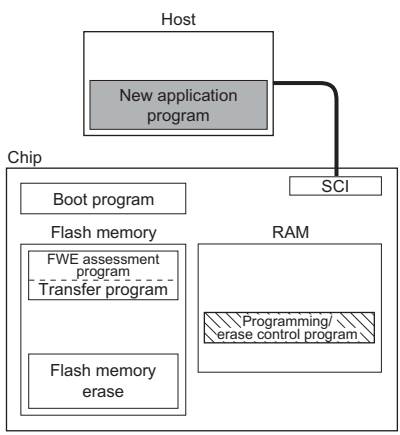
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



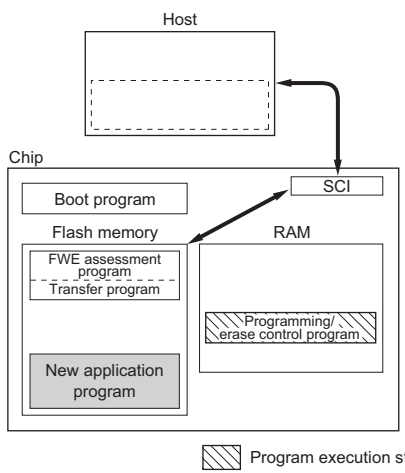

2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.




3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.




4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



**Bits 7 and 6—Reserved:** These bits always read 0.

**Bits 5 and 4—Reserved:** Only 0 may be written to these bits.

**Bit 3—RAM Select (RAMS):** Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

**Bit 3**

| RAMS | Description | |
|---|---|---|
| 0 | Emulation not selected<br>Program/erase-protection of all flash memory blocks is disabled | (Initial value) |
| 1 | Emulation selected<br>Program/erase-protection of all flash memory blocks is enabled | |

**Bits 2 to 0—Flash Memory Area Selection:** These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 21B-8.)

**Table 21B-8 Flash Memory Area Divisions**

| Addresses | Block Name | RAMS | RAM1 | RAM1 | RAM0 |
|---|---|---|---|---|---|
| H'FFD000 to H'FFDFFF | RAM area 4 kbytes | 0 | * | * | * |
| H'000000 to H'000FFF | EB0 (4 kbytes) | 1 | 0 | 0 | 0 |
| H'001000 to H'001FFF | EB1 (4 kbytes) | 1 | 0 | 0 | 1 |
| H'002000 to H'002FFF | EB2 (4 kbytes) | 1 | 0 | 1 | 0 |
| H'003000 to H'003FFF | EB3 (4 kbytes) | 1 | 0 | 1 | 1 |
| H'004000 to H'004FFF | EB4 (4 kbytes) | 1 | 1 | 0 | 0 |
| H'005000 to H'005FFF | EB5 (4 kbytes) | 1 | 1 | 0 | 1 |
| H'006000 to H'006FFF | EB6 (4 kbytes) | 1 | 1 | 1 | 0 |
| H'007000 to H'007FFF | EB7 (4 kbytes) | 1 | 1 | 1 | 1 |

*: Don't care

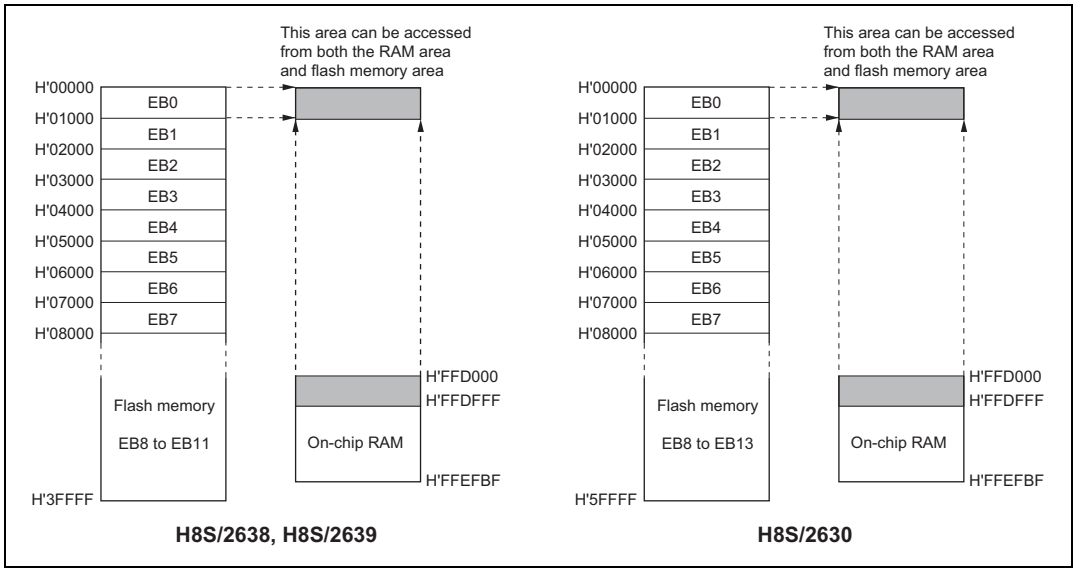



**Figure 21B-16 Example of RAM Overlap Operation**

### Example in which Flash Memory Block Area EB0 is Overlapped

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

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

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

### 21C.8.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI channel to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2635 Group's pins have been set to boot mode, the boot program built into the H8S/2635 Group are started and the programming control program prepared in the host is serially transmitted to the H8S/2635 Group via the SCI. In the H8S/2635 Group, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 21C-7, and the boot mode execution procedure in figure 21C-8.

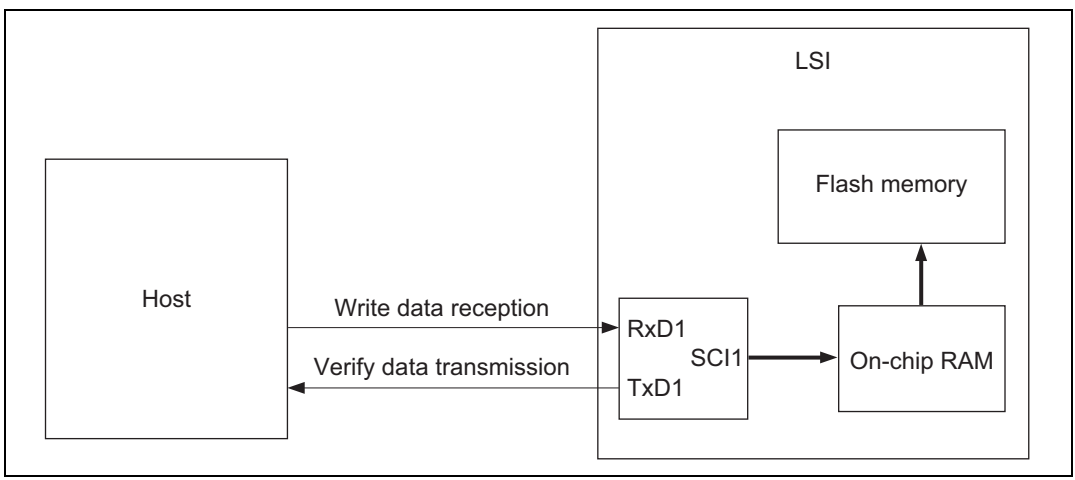



**Figure 21C-7 System Configuration in Boot Mode**

## 23A.2 Register Descriptions

### 23A.2.1 Standby Control Register (SBYCR)

| Bit | : | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| | | SSBY | STS2 | STS1 | STS0 | OPE | — | — | — |
| Initial value | : | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| R/W | : | R/W | R/W | R/W | R/W | R/W | — | — | — |

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'58 by a reset and in hardware standby mode. It is not initialized in software standby mode.

**Bit 7—Software Standby (SSBY):** When making a low power dissipation mode transition by executing the SLEEP instruction, the operating mode is determined in combination with other control bits.

Note that the value of the SSBY bit does not change even when shifting between modes using interrupts.

**Bit 7**

| SSBY | Description |
|---|---|
| 0 | Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. (Initial value) |
| 1 | Shifts to software standby mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. |

**Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0):** These bits select the MCU wait time for clock stabilization when shifting to high-speed mode or medium-speed mode by using a specific interrupt or command to cancel software standby mode. With a quartz oscillator (Table 23A-5), select a wait time of 8ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, select a standby time of 2 ms or more (PLL oscillator settling time), based on the operating frequency.

## A.2 Instruction Codes

Table A-2 shows the instruction codes.

MC7[4]

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

MC7[5]

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | STD_ID2 | STD_ID1 | STD_ID0 | RTR | IDE | — | EXD_ID17 | EXD_ID16 |
| Initial value | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

| | |
|---|---|
| 0 | Standard format |
| 1 | Extended format |

Remote Transmission Request

| | |
|---|---|
| 0 | Data frame |
| 1 | Remote frame |

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC7[6]

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | STD_ID10 | STD_ID9 | STD_ID8 | STD_ID7 | STD_ID6 | STD_ID5 | STD_ID4 | STD_ID3 |
| Initial value | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

# Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

### Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{\text{RES}}$ signal low at least 10 states before the $\overline{\text{STBY}}$ signal goes low, as shown below. $\overline{\text{RES}}$ must remain low until $\overline{\text{STBY}}$ signal goes low (delay from $\overline{\text{STBY}}$ low to $\overline{\text{RES}}$ high: 0 ns or more).

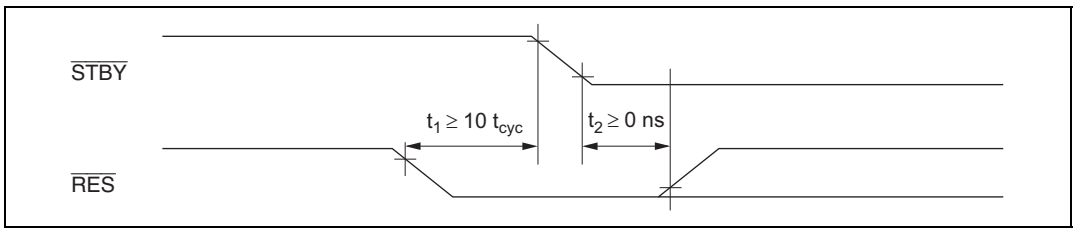



**Figure E-1 Timing of Transition to Hardware Standby Mode**

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained, $\overline{\text{RES}}$ does not have to be driven low as in (1).

### Timing of Recovery from Hardware Standby Mode

Drive the $\overline{\text{RES}}$ signal low and the NMI signal high approximately 100 ns or more before $\overline{\text{STBY}}$ goes high to execute a reset.

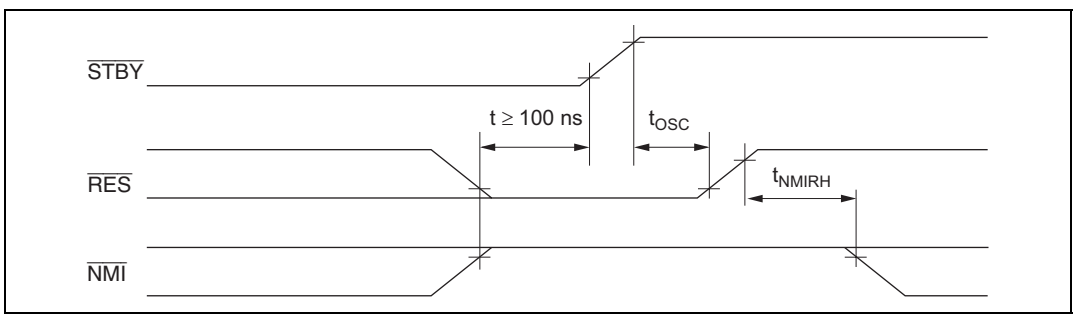



**Figure E-2 Timing of Recovery from Hardware Standby Mode**