Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Not For New Designs
Core Processor	H8S/2000
Core Size	16-Bit
Speed	13MHz
Connectivity	I²C, SCI, SmartCard
Peripherals	POR, PWM, WDT
Number of I/O	72
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.2V ~ 3.6V
Data Converters	A/D 8x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	100-BFQFP
Supplier Device Package	100-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2238rfa13v

Preface

The H8S/2558 Group, H8S/2239 Group, H8S/2238 Group, H8S/2237 Group, and H8S/2227 Group are high-performance microcomputers made up of the internal 32-bit configuration H8S/2000 CPU as their cores, and the peripheral functions required to configure a system.

A single-power flash memory (F-ZTAT™*) version and masked ROM version are available for these LSIs' ROM. These versions provide flexibility as they can be reprogrammed in no time to cope with all situations from the early stages of mass production to full-scale mass production. This is particularly applicable to application devices of which the specifications frequently changeable.

On-chip peripheral functions of each microcomputer are summarized below.

Note: * F-ZTAT is a trademark of Renesas Technology Corp.

Section 14 IEBus™ Controller (IEB) [H8S/2258 Group]

Section 15 Serial Communication Interface (SCI)

1.2 Internal Block Diagram

Figures 1.1 to 1.5 show the internal block diagrams.

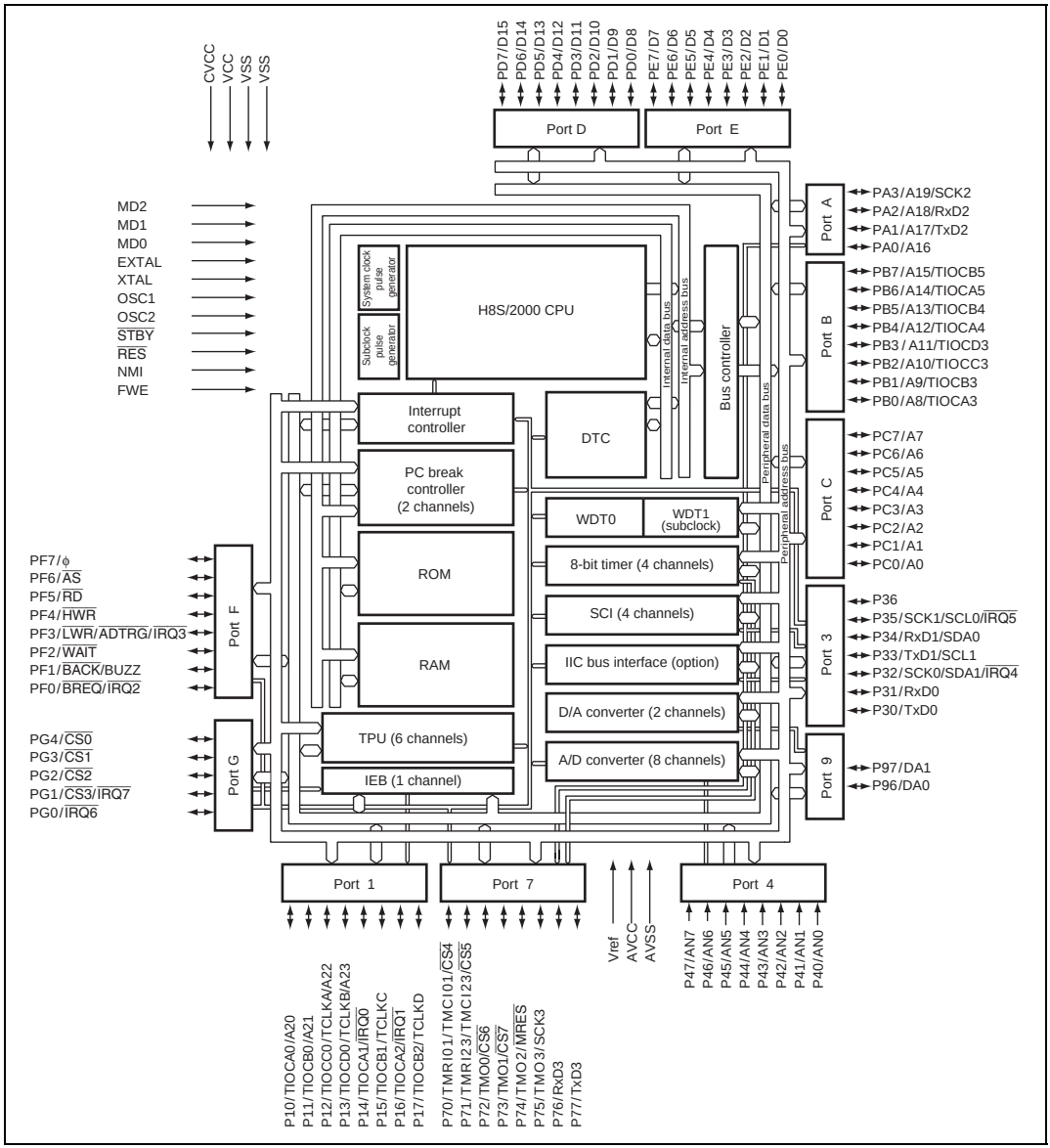

Figure 1.1 Internal Block Diagram of H8S/2258 Group

Type	Symbol	Pin No. TFP-100B TFP-100BV FP-100B FP-100BV	Pin No. FP-100A FP-100AV	I/O	Function
I/O ports	PC7 to PC0	21 to 15, 13	24 to 18, 16	Input/ Output	8-bit I/O pins.
	PD7 to PD0	11 to 4	14 to 7	Input/ Output	8-bit I/O pins.
	PE7 to PE0	100 to 96, 3 to 1	100, 99, 6 to 1	Input/ Output	8-bit I/O pins.
	PF7 to PF0	75 to 68	78 to 71	Input/ Output	8-bit I/O pins.
	PG4 to PG0	95 to 91	98 to 94	Input/ Output	5-bit I/O pins.

Note: * Measures should be taken to deal with noise, which can cause operation errors otherwise.

Table 4.4 Status of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	Trace exception handling cannot be used.			
2	1	—	—	0

Legend:

1: Set to 1
0: Cleared to 0
—: Retains value prior to execution

4.5 Interrupts

Interrupts are controlled by the interrupt controller. The interrupt control has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control. For details, refer to section 5, Interrupt Controller.

Interrupt exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution begins from that address.

4.6 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

Trap instruction exception handling is conducted as follows:

1. The values in the program counter (PC), condition code register (CCR), and extended control register (EXR) are saved to the stack.
2. The interrupt mask bit is updated and the T bit is cleared to 0.
3. A vector address corresponding to the interrupt source is generated, the start address is loaded from the vector table to the PC, and program execution starts from that address.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

(2) Pin Wait Insertion

Setting the WAITE bit in BCRL to 1 enables wait insertion by means of the $\overline{\text{WAIT}}$ pin. When external space is accessed in this state, program wait insertion is first carried out according to the settings in WCRH and WCRL. Then, if the $\overline{\text{WAIT}}$ pin is low at the falling edge of ϕ in the last T_2 or T_w state, a T_w state is inserted. If the $\overline{\text{WAIT}}$ pin is held low, T_w states are inserted until it goes high.

Figure 7.18 shows an example of wait state insertion timing.

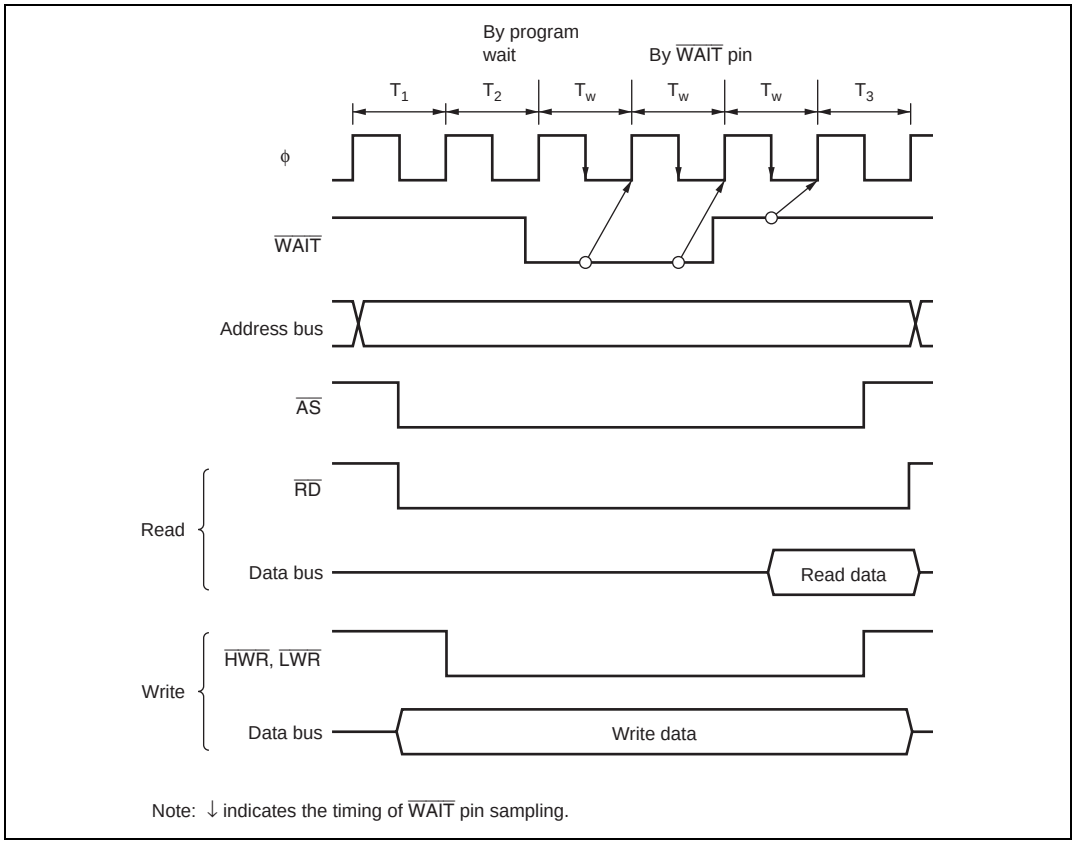

Figure 7.18 Example of Wait State Insertion Timing

- DMABCRL

Bit	Bit Name	Initial Value	R/W	Description
7	DTME1	0	R/W	Data Transfer Master Enable 1 Together with the DTE1 bit, this bit controls enabling or disabling of data transfer on channel 1. When both the DTME1 bit and DTE1 bit are set to 1, transfer is enabled for channel 1. If channel 1 is in the middle of a burst mode transfer when an NMI interrupt is generated, the DTME1 bit is cleared, the transfer is interrupted, and bus mastership passes to the CPU. When the DTME1 bit is subsequently set to 1 again, the interrupted transfer is resumed. In block transfer mode, however, the DTME1 bit is not cleared by an NMI interrupt, and transfer is not interrupted. 0: Data transfer is disabled 1: Data transfer is enabled [Clearing conditions] • When initialization is performed • When NMI is input in burst mode • When 0 is written to the DTME1 bit [Setting condition] When 1 is written to DTME1 after reading DTME1 = 0

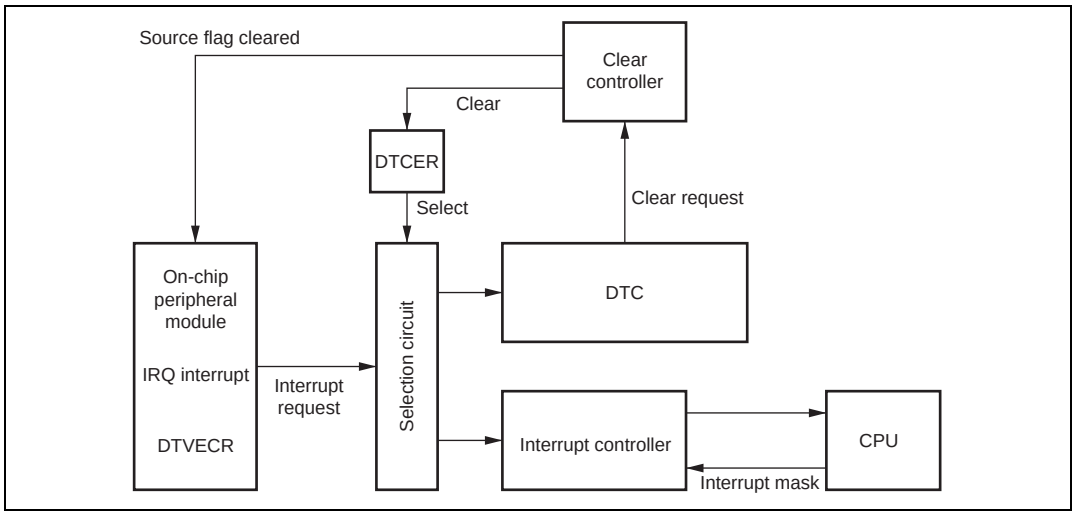

Figure 9.2 Block Diagram of DTC Activation Source Control

9.4 Location of Register Information and DTC Vector Table

Locate the register information in the on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF). Register information should be located at an address that is a multiple of four within the range. Locating the register information in address space is shown in figure 9.3. Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information.

In the case of chain transfer, register information should be located in consecutive areas as shown in figure 9.3, and the register information start address should be located at the vector address corresponding to the interrupt source. Figure 9.4 shows the correspondence between DTC vector address and register information. The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address.

When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] × 2). For example, if DTVECR is H'10, the vector address is H'0420.

The configuration of the vector address is the same in both normal* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the register information start address.

Note: * Normal mode cannot be used in this LSI.

10.1 Port 1

Port 1 is an 8-bit I/O port and has the following registers.

- Port 1 data direction register (P1DDR)
- Port 1 data register (P1DR)
- Port 1 register (PORT1)

10.1.1 Port 1 Data Direction Register (P1DDR)

P1DDR specifies input or output of the port 1 pins using the individual bits. P1DDR cannot be read; if it is, an undefined value will be read. This register is a write-only register, and cannot be written by bit manipulation instruction. For details, see section 2.9.4, Access Methods for Registers with Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DDR	0	W	When a pin is specified as a general purpose I/O port, setting this bit to 1 makes the corresponding port 1 pin an output pin. Clearing this bit to 0 makes the pin an input pin.
6	P16DDR	0	W	
5	P15DDR	0	W	
4	P14DDR	0	W	
3	P13DDR	0	W	
2	P12DDR	0	W	
1	P11DDR	0	W	
0	P10DDR	0	W	

10.1.2 Port 1 Data Register (P1DR)

P1DR stores output data for port 1 pins.

Bit	Bit Name	Initial Value	R/W	Description
7	P17DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose output port.
6	P16DR	0	R/W	
5	P15DR	0	R/W	
4	P14DR	0	R/W	
3	P13DR	0	R/W	
2	P12DR	0	R/W	
1	P11DR	0	R/W	
0	P10DR	0	R/W	

10.1.3 Port 1 Register (PORT1)

PORT1 shows the pin states. This register cannot be modified.

Bit	Bit Name	Initial Value	R/W	Description
7	P17	—*	R	If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.
6	P16	—*	R	
5	P15	—*	R	
4	P14	—*	R	
3	P13	—*	R	
2	P12	—*	R	
1	P11	—*	R	
0	P10	—*	R	

Note: * Determined by the states of pins P17 to P10.

- Port B pull-up MOS control register (PBPCR)

10.7.1 Port B Data Direction Register (PBDDR)

PBDDR specifies input or output the port B pins using the individual bits. PBDDR cannot be read; if it is, an undefined value will be read. This register is a write-only register, and cannot be written by bit manipulation instruction. For details, see section 2.9.4, Access Methods for Registers with Write-Only Bits.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DDR	0	W	When a pin is specified as a general purpose I/O port, setting the bit to 1 makes the corresponding port B pin an output pin. Clearing the bit to 0 makes the pin an input pin.
6	PB6DDR	0	W	
5	PB5DDR	0	W	
4	PB4DDR	0	W	
3	PB3DDR	0	W	
2	PB2DDR	0	W	
1	PB1DDR	0	W	
0	PB0DDR	0	W	

10.7.2 Port B Data Register (PBDR)

PBDR stores output data for port B pins.

Bit	Bit Name	Initial Value	R/W	Description
7	PB7DR	0	R/W	Output data for a pin is stored when the pin is specified as a general purpose output port.
6	PB6DR	0	R/W	
5	PB5DR	0	R/W	
4	PB4DR	0	R/W	
3	PB3DR	0	R/W	
2	PB2DR	0	R/W	
1	PB1DR	0	R/W	
0	PB0DR	0	R/W	

11.3 Register Descriptions

The TPU has the following registers in each channel.

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

When more than one unit starts transfer of communications frame at the same timing, broadcast communications has priority over normal communications, and arbitration occurs.

(2) Master Address Field

The master address field is a field for transmitting the unit address (master address) to other units. The master address field is comprised of master address bits and a parity bit.

The master address has 12 bits and are output MSB first.

When more than one unit starts transfer of the broadcast bit having the same value at the same timing, arbitration is decided by the master address field.

In the master address field, self-output data and data on the bus are compared for every one-bit transfer. If the self-output master address and data on the bus are different, the unit that loses arbitration, stops transfer, and enters the receive state.

Since the IEBus is configured with wired AND, a unit having the smallest master address of the units in arbitration (arbitration master) wins in arbitration.

Finally, only a single unit remains in the transfer state as a master unit after outputting 12-bit master address.

Next, this master unit outputs a parity bit*, defines the master address to other units, and then enters the slave address field output state.

Note: * Since even parity is used, when the number of one bits in the master address is odd, the parity bit is 1.

(3) Slave Address Field

The slave address field is a field to transmit an address (slave address) of a unit (slave unit) to which a master transmit data. The slave address field is comprised of slave address bits, a parity bit, and an acknowledge bit.

The slave address has 12 bits and is output MSB first. The parity bit is output after the 12-bit slave address is transmitted in order to avoid receiving the slave address accidentally. The master unit then detects the acknowledgement from the slave unit in order to confirm that the slave unit exists on the bus. When the acknowledgement is detected, the master unit enters the control field output state. However, the master unit enters the control field output state without detecting the acknowledgement in broadcast communications.

14.3.5 IEBus Master Unit Address Register 2 (IEAR2)

IEAR2 sets the upper 8 bits of the master unit address. In master communications, this register becomes the master address field value. In slave communications, this register is compared with the received slave address field.

Bit	Bit Name	Initial Value	R/W	Description
7	IAR11	0	R/W	Upper 8 Bits of IEBus Master Unit Address
6	IAR10	0	R/W	Set the upper 8 bits of the master unit address.
5	IAR9	0	R/W	
4	IAR8	0	R/W	
3	IAR7	0	R/W	
2	IAR6	0	R/W	
1	IAR5	0	R/W	
0	IAR4	0	R/W	

14.3.6 IEBus Slave Address Setting Register 1 (IESA1)

IESA1 sets the lower 4 bits of the communications destination slave unit address. For slave communications, it is not necessary to set this register.

Bit	Bit Name	Initial Value	R/W	Description
7	ISA3	0	R/W	Lower 4 Bits of IEBus Slave Address
6	ISA2	0	R/W	These bits set the lower 4 bits of the communications destination slave unit address
5	ISA1	0	R/W	
4	ISA0	0	R/W	
3 to 0	—	All 0	—	Reserved These bits are always read as 0 and cannot be modified.

(c) Setting the IEBus Transmit Message Length Register (IETBFL)

Specify the message length bits.

(d) Setting the IEBus Transmit/Runaway Interrupt Enable Register (IEIET)

Enable the TxRDY (IETxI), TxS, and TxE (IETSI) interrupts.

The above registers can be specified in any order. (The register specification order does not affect the IEB operation.)

(2) DTC Initialization

1. Set the start address of the RAM which stores the register information necessary for the DTC transfer in the vector address (H'000004D4) to be accessed a DTC transfer request is generated.
2. Set the following data from the start address of the RAM.
   - Transfer source address (SAR): Start address of the RAM which stores data to be transmitted from the data field.
   - Transfer destination address (DAR): Address (H'FFF808) of the IEBus transmit buffer register (IETBR)
   - Transfer count (CRA): The same value as IETBFL
3. Set bit DTCEG5 in the DTC enabler register G (DTCERG), and enable the TxRDY interrupt (IETxI).

   Because the TxRDY flag is retained after reset, the DTC transfer is executed when the IETxI is enabled and the first data field data is written to IETBR. The DTC negates the TxRDY flag and the DTC transfer of the first byte is completed.

(3) Slave Transmission Flow

Figure 14.12 shows the slave transmission flow. Numbers in the following description correspond to the numbers in Figure 14.12.

1. After the IEB and DTC have been initialized, a slave communications request command is issued from IECMR. During slave reception, the command execution status flag (CMX) in IEFLG is set and the slave communications request will not be issued.
2. The CMX flag is cleared when the slave reception is completed, the slave communications command is executed, and the SRQ flag is set.
3. If data up to the control field has been received correctly and if the contents of the control bits is H'3 or H'7, the transmit start detection flag (TxS) in IETSR register is set to 1. In this case, the TxS flag is cleared in the TxS interrupt handling routine.
4. The slave then transmits the message length field, and the IEB loads the transmit data in the data field from IETBR when the ACK is received. Then the TxRDY flag is set to 1. A DTC

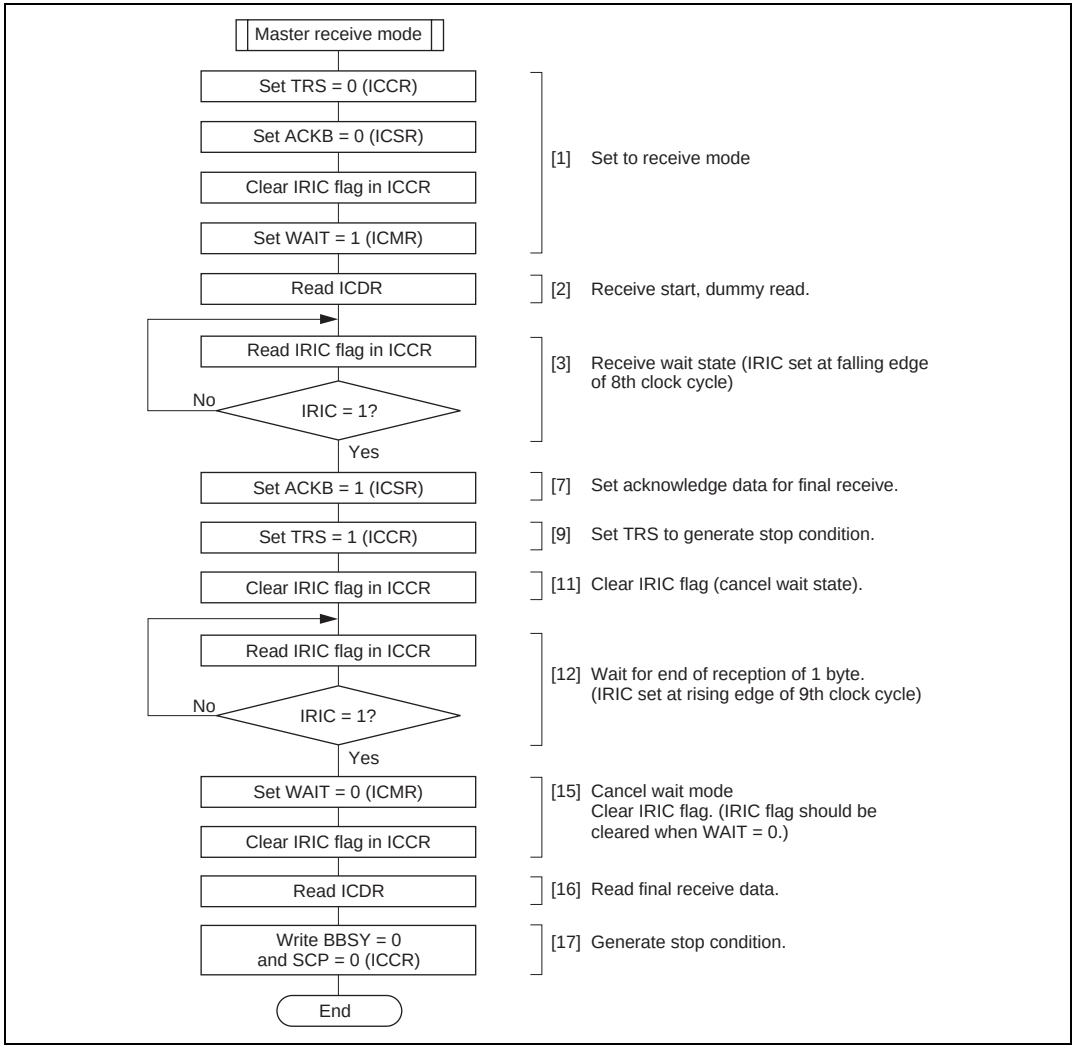

Figure 16.11 Flowchart for Master Receive Mode (Receiving 1 Byte) (WAIT = 1) (Example)

The procedure for receiving data sequentially, using the wait states (WAIT bit) for synchronization with ICDR (ICDRR) read operations, is described below.

The procedure below describes the operation for receiving multiple bytes. Note that some of the steps are omitted when receiving only 1 byte. Refer to figure 16.11 for details.

8. Notes on Start Condition Issuance for Retransmission

Depending on the timing combination with the start condition issuance and the subsequently writing data to ICDR, it may not be possible to issue the retransmission and the data transmission after retransmission condition issuance.

After start condition issuance is done and determined the start condition, write the transmit data to ICDR, as shown below. Figure 16.22 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart.

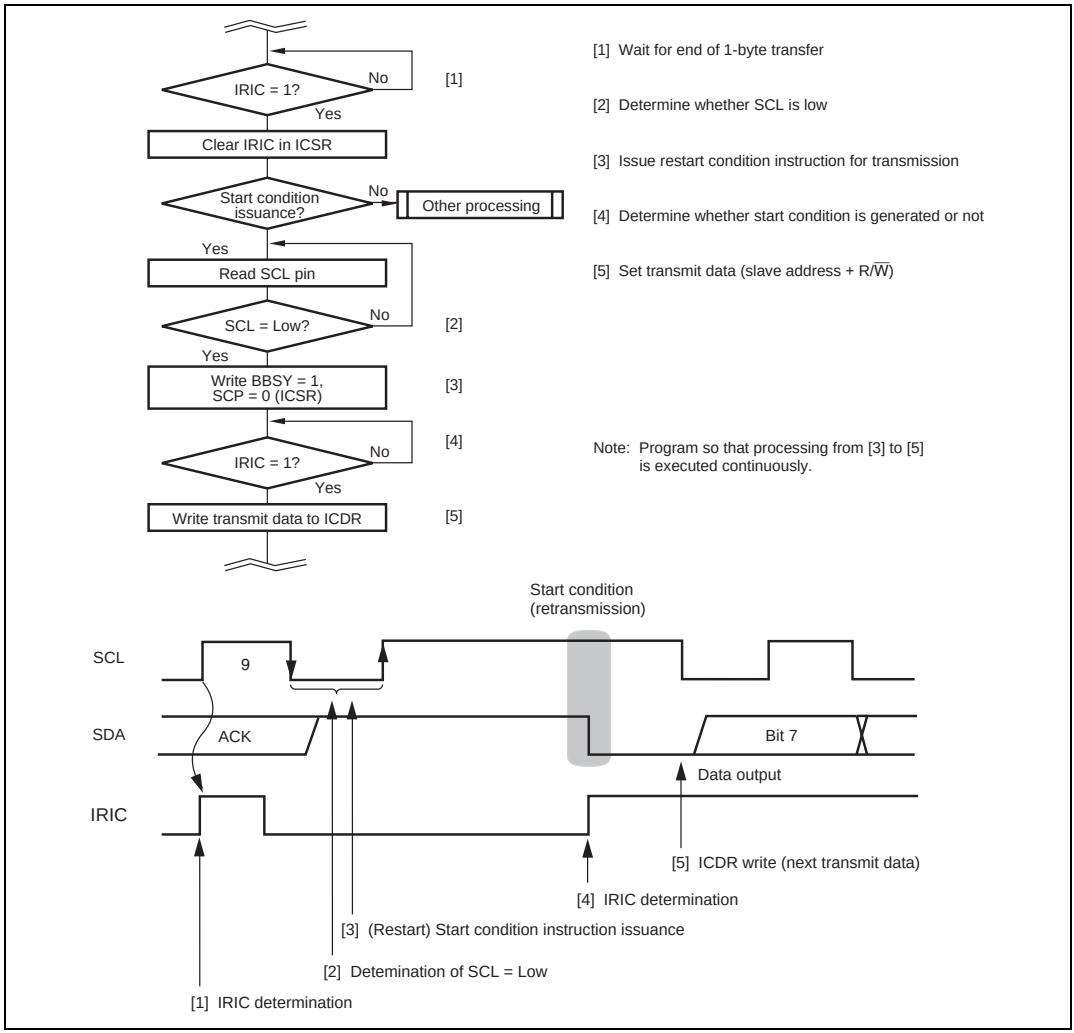

Figure 16.22 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission

Register Name	Reset	Manual Reset	High-speed	Medium-speed	Sleep	Module Stop	Watch	Sub-active	Sub-sleep	Software Standby	Hardware Standby	Module
BRR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_0
ICSR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC_0
SCR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_0
TDR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_0	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICDR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC_0
SARX_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICMR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
SAR_0	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
SMR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_1
ICCR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC_1
BRR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_1
ICSR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC_1
SCR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_1
TDR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_1	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICDR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	IIC_1
SARX_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
ICMR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
SAR_1	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
SMR_2	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	SCI_2
BRR_2	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
SCR_2	Initialized	Initialized	—	—	—	—	—	—	—	—	Initialized	
TDR_2	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SSR_2	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
RDR_2	Initialized	Initialized	—	—	—	Initialized	Initialized	Initialized	Initialized	Initialized	Initialized	
SCMR_2	Initialized	—	—	—	—	—	—	—	—	—	Initialized	

27.4.4 A/D Conversion Characteristics

A/D converter characteristics for the F-ZTAT and masked ROM versions are shown in table 27.35.

Table 27.35 A/D Conversion Characteristics (F-ZTAT and Masked ROM Versions)

Condition: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.6 V to 5.5 V, V_{ref} = 3.6 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 13.5 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	bit
Conversion time	9.6	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±6.0	LSB
Offset error	—	—	±4.0	LSB
Full-scale error	—	—	±4.0	LSB
Quantization	—	—	±0.5	LSB
Absolute accuracy	—	—	±8.0	LSB

27.4.5 D/A Conversion Characteristics

Table 27.36 lists the D/A conversion characteristics.

Table 27.36 D/A Conversion Characteristics (F-ZTAT and Masked ROM Versions)

Condition: V_{CC} = 3.0 V to 5.5 V, AV_{CC} = 3.6 V to 5.5 V, V_{ref} = 3.6 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V, ϕ = 2 MHz to 13.5 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition				
Item	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	bit	
Conversion time	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	±2.0	±3.0	LSB	2-MΩ resistive load
	—	—	±2.0	LSB	4-MΩ resistive load

Table 27.42 Clock Timing

Condition A (F-ZTAT version and masked ROM version):
V_{CC} = 2.7 V to 3.6 V, AV_{CC} = 2.7 V to 3.6 V,
V_{ref} = 2.7 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 32.768 kHz, 2 to 13.5 MHz,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

Condition B (F-ZTAT version): V_{CC} = 2.2 V to 3.6 V, AV_{CC} = 2.2 V to 3.6 V,
V_{ref} = 2.2 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 32.768 kHz, 2 to 6.25 MHz,
T_a = –20°C to +75°C (regular specifications)

Condition C (Masked ROM version): V_{CC} = 2.2 V to 3.6 V, AV_{CC} = 2.2 V to 3.6 V,
V_{ref} = 2.2 V to AV_{CC}, V_{SS} = AV_{SS} = 0 V,
ϕ = 32.768 kHz, 2 to 6.25 MHz,
T_a = –20°C to +75°C (regular specifications),
T_a = –40°C to +85°C (wide-range specifications)

		Condition A			Conditions B, C				Test
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**	**Conditions**
Clock cycle time	t_{cyc}	74	—	500	160	—	500	ns	Figure 27.10
Clock high pulse width	t_{CH}	25	—	—	50	—	—	ns	
Clock low pulse width	t_{CL}	25	—	—	50	—	—	ns	
Clock rise time	t_{Cr}	—	—	10	—	—	25	ns	
Clock fall time	t_{Cf}	—	—	10	—	—	25	ns	
Oscillation stabilization time at reset (crystal)	t_{OSC1}	20	—	—	40	—	—	ms	Figure 27.11
Oscillation stabilization time in software standby (crystal)	t_{OSC2}	8	—	—	16	—	—	ms	
External clock output stabilization delay time	t_{DEXT}	500	—	—	1000	—	—	µs	Figure 27.11
Subclock oscillation stabilization time	t_{OSC3}	—	—	2	—	—	4	s	
Subclock oscillator frequency	f_{SUB}	—	32.768	—	—	32.768	—	kHz	
Subclock (ϕ_{SUB}) cycle time	t_{SUB}	—	30.5	—	—	30.5	—	µs	