Renesas Electronics America Inc - R5F113THLFB#V0 Datasheet

Welcome to E-XFL.COM

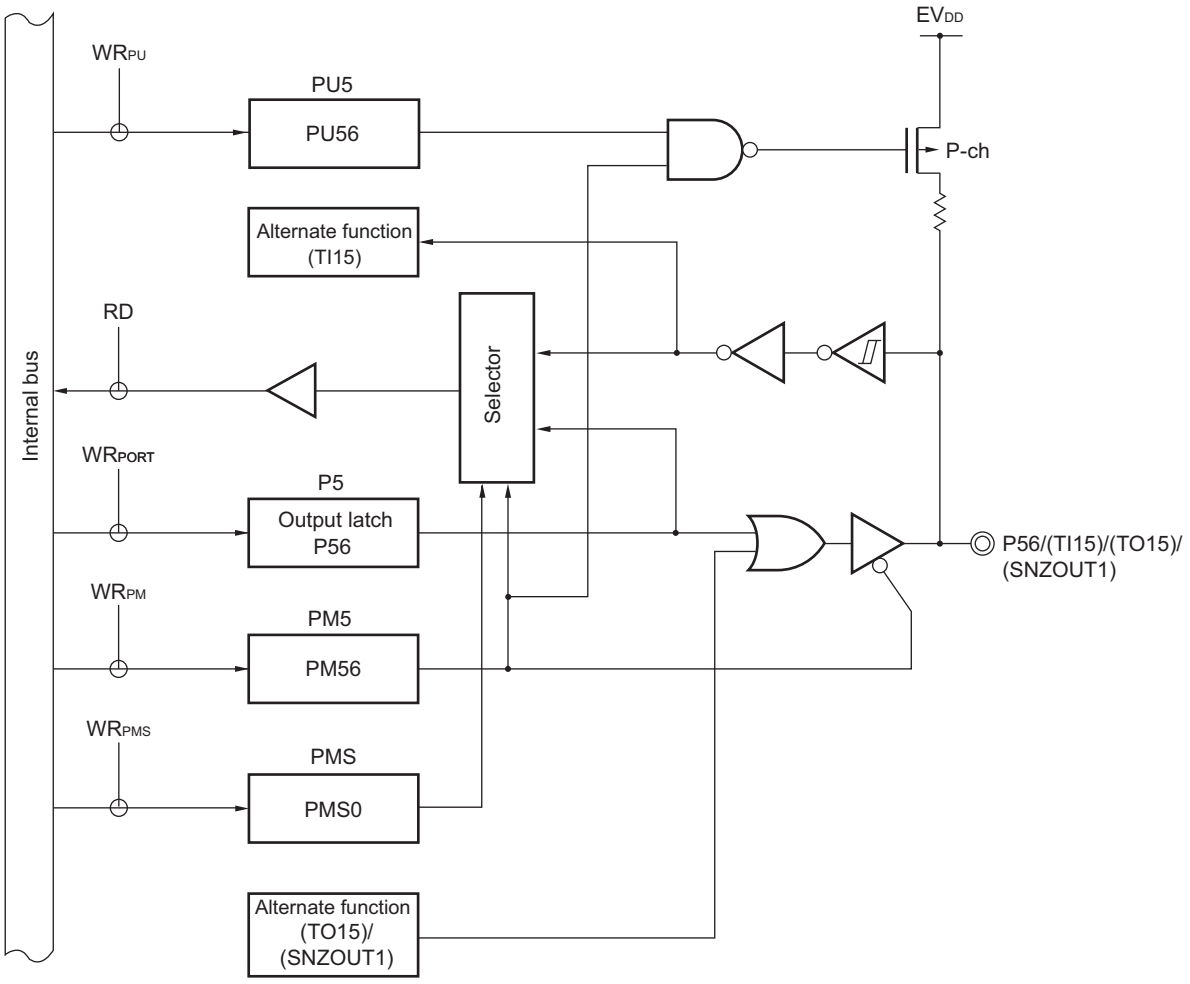
What is "Embedded - Microcontrollers"?

"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	Discontinued at Digi-Key
Core Processor	RL78
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, CSI, I²C, LINbus, SPI, UART/USART
Peripherals	LVD, POR, PWM, WDT
Number of I/O	130
Program Memory Size	192KB (192K x 8)
Program Memory Type	FLASH
EEPROM Size	8K x 8
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 31x10b; D/A 1x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LFQFP (20x20)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r5f113thlfb-v0

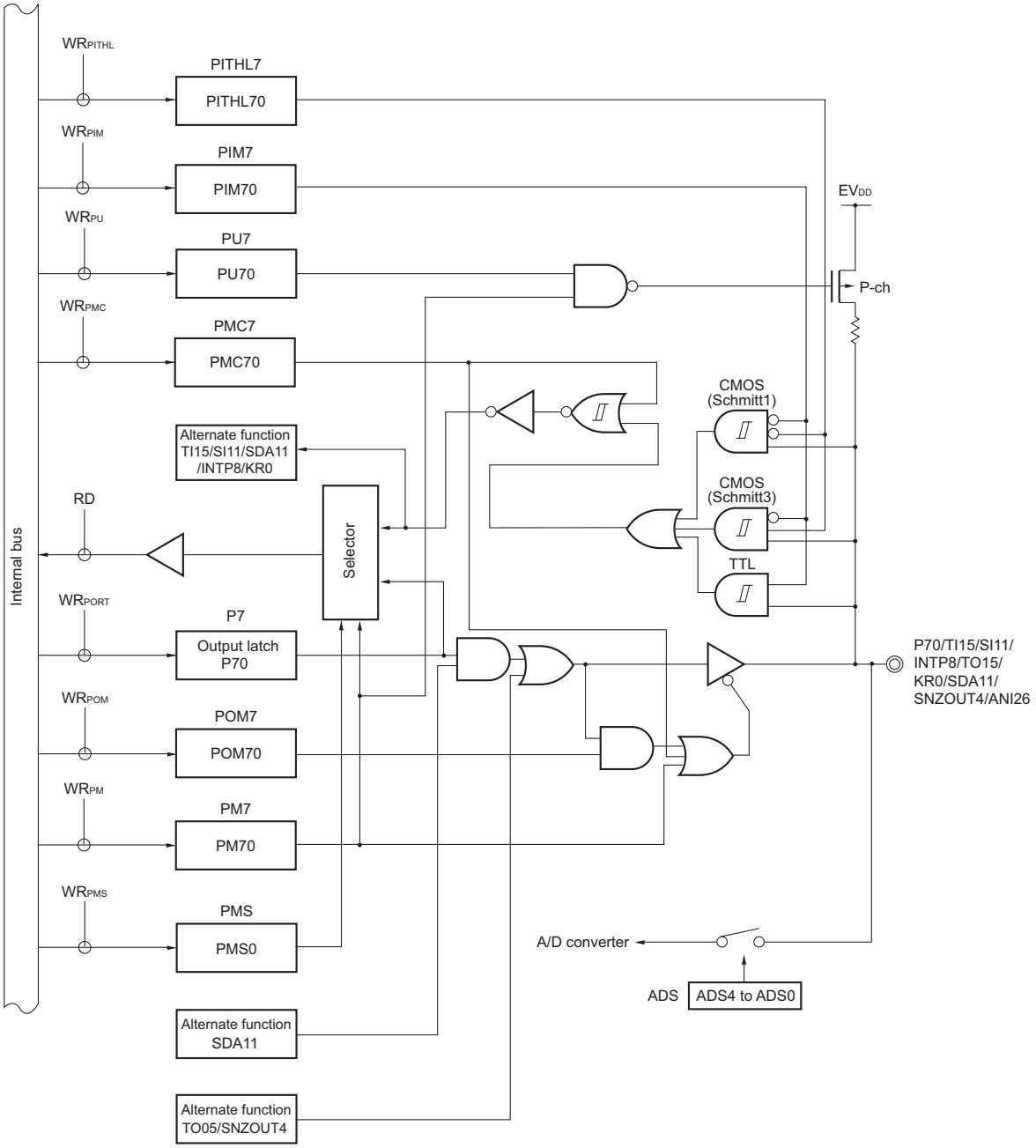
Figure 4-43. Block Diagram of P56



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
PMS: Port mode select register
RD: Read signal
WRxx: Write signal

Figures 4-53 to 4-60 show block diagrams of port 7 for 144-pin products.

Figure 4-53. Block Diagram of P70



P7: Port register 7
PU7: Pull-up resistor option register 7
PM7: Port mode register 7
PIM7: Port input mode register 7
POM7: Port output mode register 7
PMC7: Port mode control register 7
PMS: Port mode select register
PITHL7: Port input threshold control register 7
ADS: Analog input channel specification register
RD: Read signal
WRxx: Write signal

4.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the alternate function output, see **4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function**.

Set PIOR register before the target function is enabled.

No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended for lower power consumption.

Remark 144-pin products m: Unit number (m = 0 to 2), n: Master Channel number (n = 0, 2, 4)
100-pin, 80-pin, 64-pin, and 48-pin products m: Unit number (m = 0 and 1), n: Master Channel number (n = 0, 2, 4)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 7$ (Where p and q are integers greater than n)

7.3.9 Port mode registers 1, 4 (PM1, PM4)

These registers set input/output of ports 1 and 4 in 1-bit units.

When using the ports (such as P41/TRJIO0 and P10/TRJO0) to be shared with the timer output pin for timer output, set the port mode register (PMxx) bit and the port register (Pxx) bit corresponding to each port to 0.

Example: When using P41/TRJIO0 for timer output
Set the PM41 bit of port mode register 4 to 0.
Set the P41 bit of port register 4 to 0.

When using the ports (such as P41/TRJIO0) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P41/TRJIO0 for timer input
Set the PM41 bit of port mode register 4 to 1.
Set the P41 bit of port register 4 to 1.

The PM1 and PM4 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 7-10. Format of Port Mode Registers 1, 4 (PM1, PM4)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF24H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PMmn	Pmn pin I/O mode selection (m = 1, 4 ; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

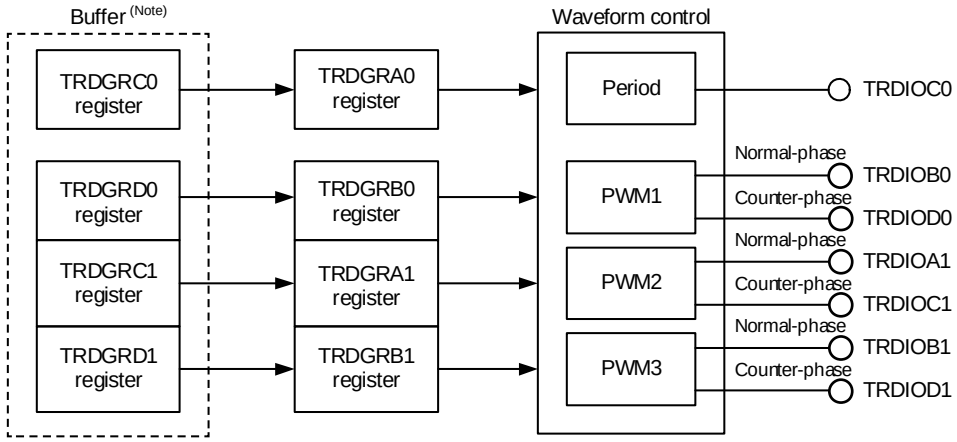
8.3.5 Reset Synchronous PWM Mode

In this mode, three normal-phases and three counter-phases of the PWM waveform are output with the same period (three-phase, sawtooth wave modulation, and no dead time).

Figure 8-55 shows the Block Diagram of Reset Synchronous PWM Mode, Table 8-17 lists the Reset Synchronous PWM Mode Specifications, Figure 8-56 shows an Operation Example in Reset Synchronous PWM Mode.

See **Figure 8-54 Operation Example in PWM Function (Duty Cycle 0%, Duty Cycle 100%)** for an operation example in PWM Mode with duty cycle 0% and duty cycle 100%.

Figure 8-55. Block Diagram of Reset Synchronous PWM Mode



Note When bits TRDBFC0, TRDBFD0, TRDBFC1, and TRDBFD1 in the TRDMR register are set to 1 (buffer register).

9.3.13 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks (f_{RTC}) later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

Set the MONTH register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 01H.

Figure 9-14. Format of Month Count Register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.

9.3.14 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 operating clocks (f_{RTC}) later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

Set the YEAR register by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-15. Format of Year Count Register (YEAR)

Address: FFF98H After reset: 00H R/W

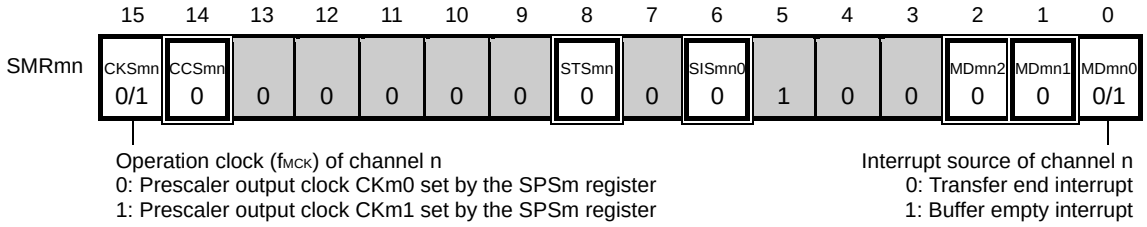
Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), see 9.4.3 Reading/writing real-time clock and follow the described procedures.

(1) Register setting

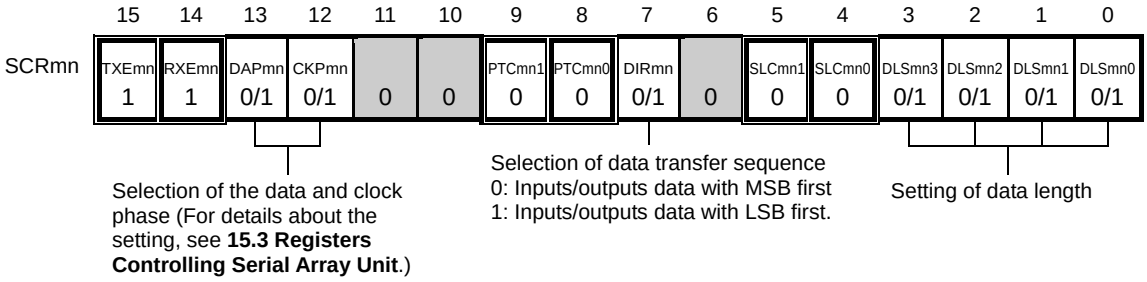
Figure 15-54. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21) (1/2)

(a) Serial mode register mn (SMRmn)

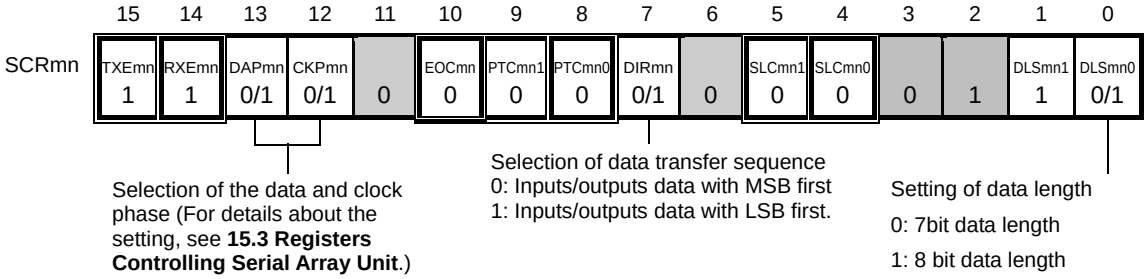


(b) Serial communication operation setting register mn (SCRmn)

(1) CSI00, CSI01, CSI10, CSI11

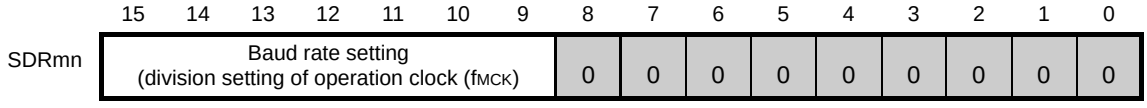


(2) CSI20, CSI21

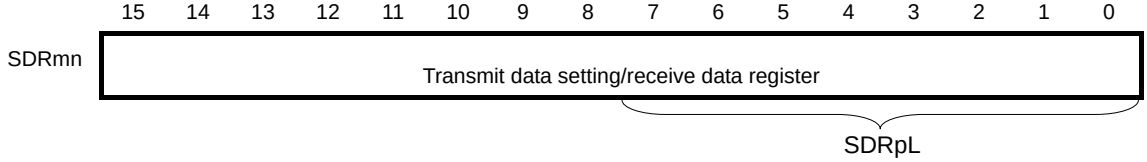


(c) Serial data register mn (SDRmn)

(1) CSI00, CSI01, CSI10, CSI11: When operation is stopped (SEmn = 0)



(2) CSI00, CSI01, CSI10, CSI11: When operation is in progress (SEmn = 1) (Lower 8 bits: SDRpL)



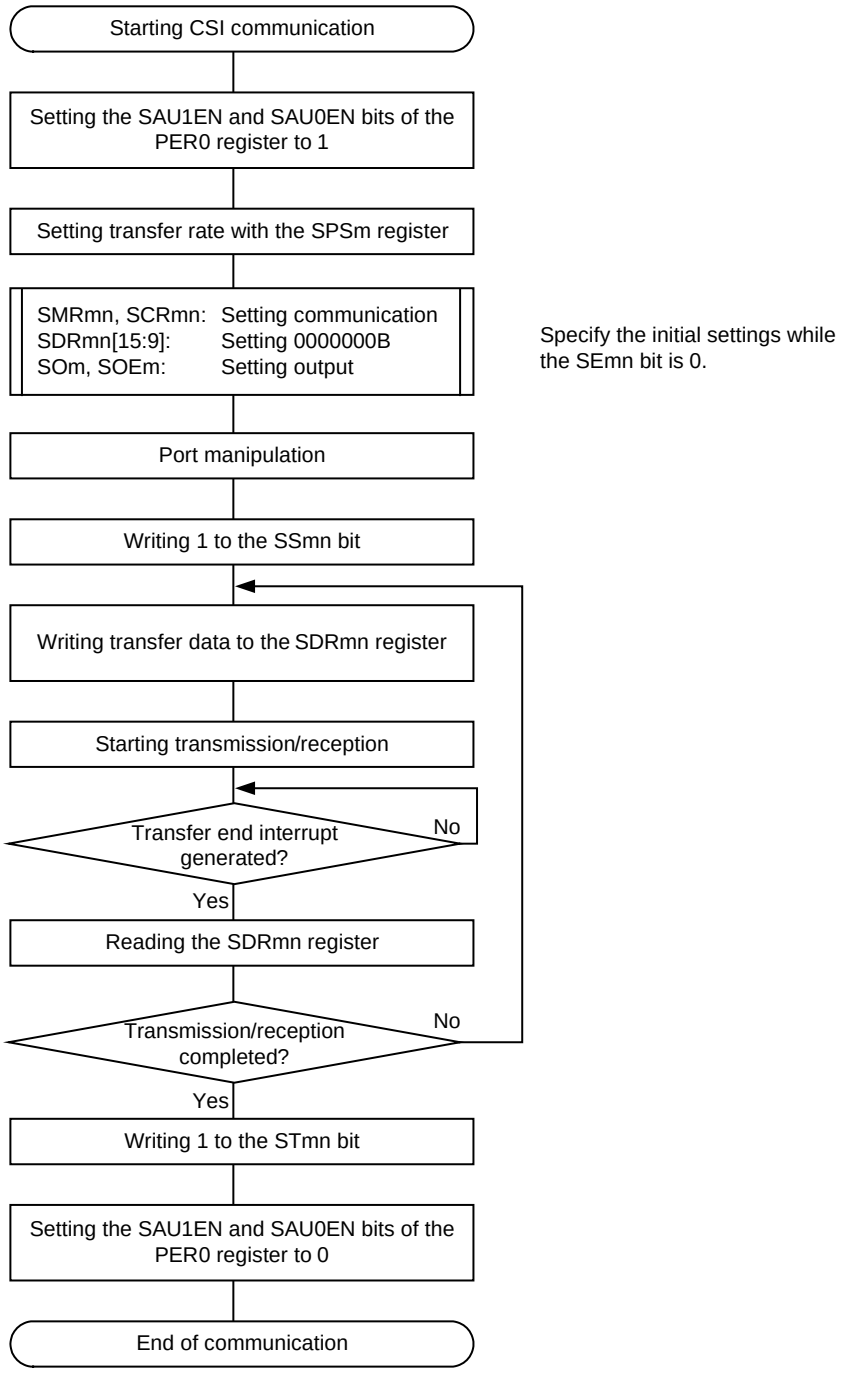
Remarks 1. m: Unit number (m = 0-2), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11, 20, 21), mn = 00, 01, 10, 11, 20, 21

2. ☐: Setting is fixed in the CSI master transmission/reception mode

▒: Setting disabled (set to the initial value)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 15-153. Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



Cautions 1. Be sure to set transmit data to the SDRpL register before the clock from the master is started.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01, 10, 11), mn = 00, 01, 10, 11

Table 18-3. List of CAN Module Registers (12/37)

Address	Special Function Register (SFR) Name	Symbol		R/W	Access Size			After Reset
					1 bit	8 bits	16 bits	
F0444H	CAN receive rule entry register 13CL[Note 1]	GAFLPL13L	GAFLPL13	R/W	—	√	√	0000H
F0445H		GAFLPL13H			—	√		
F0444H	CAN receive buffer register 10BL[Note 2]	RMTS10L	RMTS10	R	—	√	√	0000H
F0445H		RMTS10H			—	√		
F0446H	CAN receive rule entry register 13CH[Note 1]	GAFLPH13L	GAFLPH13	R/W	—	√	√	0000H
F0447H		GAFLPH13H			—	√		
F0446H	CAN receive buffer register 10BH[Note 2]	RMPTR10L	RMPTR10	R	—	√	√	0000H
F0447H		RMPTR10H			—	√		
F0448H	CAN receive rule entry register 14AL[Note 1]	GAFLIDL14L	GAFLIDL14	R/W	—	√	√	0000H
F0449H		GAFLIDL14H			—	√		
F0448H	CAN receive buffer register 10CL[Note 2]	RMDF010L	RMDF010	R	—	√	√	0000H
F0449H		RMDF010H			—	√		
F044AH	CAN receive rule entry register 14AH[Note 1]	GAFLIDH14L	GAFLIDH14	R/W	—	√	√	0000H
F044BH		GAFLIDH14H			—	√		
F044AH	CAN receive buffer register 10CH[Note 2]	RMDF110L	RMDF110	R	—	√	√	0000H
F044BH		RMDF110H			—	√		
F044CH	CAN receive rule entry register 14BL[Note 1]	GAFLML14L	GAFLML14	R/W	—	√	√	0000H
F044DH		GAFLML14H			—	√		
F044CH	CAN receive buffer register 10DL[Note 2]	RMDF210L	RMDF210	R	—	√	√	0000H
F044DH		RMDF210H			—	√		
F044EH	CAN receive rule entry register 14BH[Note 1]	GAFLMH14L	GAFLMH14	R/W	—	√	√	0000H
F044FH		GAFLMH14H			—	√		
F044EH	CAN receive buffer register 10DH[Note 2]	RMDF310L	RMDF310	R	—	√	√	0000H
F044FH		RMDF310H			—	√		
F0450H	CAN receive rule entry register 14CL[Note 1]	GAFLPL14L	GAFLPL14	R/W	—	√	√	0000H
F0451H		GAFLPL14H			—	√		
F0450H	CAN receive buffer register 11AL[Note 2]	RMIDL11L	RMIDL11	R	—	√	√	0000H
F0451H		RMIDL11H			—	√		
F0452H	CAN receive rule entry register 14CH[Note 1]	GAFLPH14L	GAFLPH14	R/W	—	√	√	0000H
F0453H		GAFLPH14H			—	√		
F0452H	CAN receive buffer register 11AH[Note 2]	RMIDH11L	RMIDH11	R	—	√	√	0000H
F0453H		RMIDH11H			—	√		
F0454H	CAN receive rule entry register 15AL[Note 1]	GAFLIDL15L	GAFLIDL15	R/W	—	√	√	0000H
F0455H		GAFLIDL15H			—	√		
F0454H	CAN receive buffer register 11BL[Note 2]	RMTS11L	RMTS11	R	—	√	√	0000H
F0455H		RMTS11H			—	√		
F0456H	CAN receive rule entry register 15AH[Note 1]	GAFLIDH15L	GAFLIDH15	R/W	—	√	√	0000H
F0457H		GAFLIDH15H			—	√		
F0456H	CAN receive buffer register 11BH[Note 2]	RMPTR11L	RMPTR11	R	—	√	√	0000H
F0457H		RMPTR11H			—	√		

Notes 1. These registers are allocated to RAM window 0 for the CAN module (receive rules and CAN RAM test register). When setting these registers, set the RPAGE bit in the GRWCR register to 0.

2. These registers are allocated to RAM window 1 for the CAN module (receive buffer, receive FIFO buffer, transmit/receive FIFO buffer, transmit buffer, and transmit history data). When setting these registers, set the RPAGE bit in the GRWCR register to 1.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode, when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] bits are set to B'001 or more.
The CFMC[5:0] value is incremented to H'01. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer, if the RFE bit in the RFCCm register is set to 1 (receive FIFO buffers are used) and RFDC[2:0] bits in the RFCCm register are set to B'001 or more.
The RFMC[5:0] value in the RFSTSm register is incremented to H'01. When the RFIM bit in the RFCCm register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RFSTSm register is set to 1 (a receive FIFO interrupt request is present).

Table 19-11. Conditions for setting the status transmission request flag (slave)

Various statuses					Value received using the control field				
equa	lockf	eqpa	IEBB0STXE	IEBB0SRXE	0H	3H, 7H	4H, 5H	6H	AH, BH, EH, FH
1	0	0	Any	Any	Set	Not set	Not set	Set	Not set
1	0	1	Any	Any	Set	Not set	Not set	Set	Not set
1	1	0	Any	Any	Set	Not set	Set	Not set	Not set
1	1	1	Any	Any	Set	Not set	Set	Set	Not set

Note equa: Unit match (during individual communication, IEBB0UAR register match, during broadcast communication: group match, FFF match)
lockf: Whether there is a lock
eqpa: Lock master match
IEBB0STXE: Slave transmission enable flag (IEBB0BCR register bit 4)
IEBB0SRXE: Slave reception enable flag (IEBB0BCR register bit 3)

If an IEBus vector interrupt (Status transmission request) occurs in the single mode, read the IEBB0CDR register to check the received control data contents, and then write the required slave status information to the IEBB0DR register.
The received control data and data written to the IEBB0DR register are shown below.

Table 19-12. Slave request conditions (conditions for setting the IEBB0SRQF bit)

Received control data	Function	Data written to the IEBB0DR register
0H, 6H	Slave status transmission	Value read from the IEBB0SSR register
4H	Transmission of the lower 8 bits of the lock address	Lower 8 bits of the IEBB0PAR register
5H	Transmission of the higher 8 bits of the lock address	Higher 8 bits of the IEBB0PAR register

Caution After an IEBus vector interrupt (status transmission request) occurs, be sure to write the appropriate data to the IEBB0DR register before the completion of the message length field. If writing is not in time, do not transmit IEBB0DR register data and cause an underrun error.

- **IEBB0ETRF: Bit 3**

A flag that indicates whether communication ends after the number of bytes set in the message length field have been transferred.

Set/clear condition

Set: When the unit is the communication target (during communication with the master unit or slave unit) and the value of the IEBB0SCR register becomes 0 at the end of the data field acknowledge period.

Clear: The flag is cleared (to 0) by hardware when an IEBus vector interrupt (Start request), IEBus vector interrupt (Status transmission request), IEBus vector interrupt (Frame end with no IEBus vector interrupt (Communication end), IEBus data interrupt (Transmission data write request), or IEBus data interrupt (Reception data read request) occurs.

If the communication completion flag is seto to 1, an IEBus based on the occurrence of the IEBus vector interrupt.

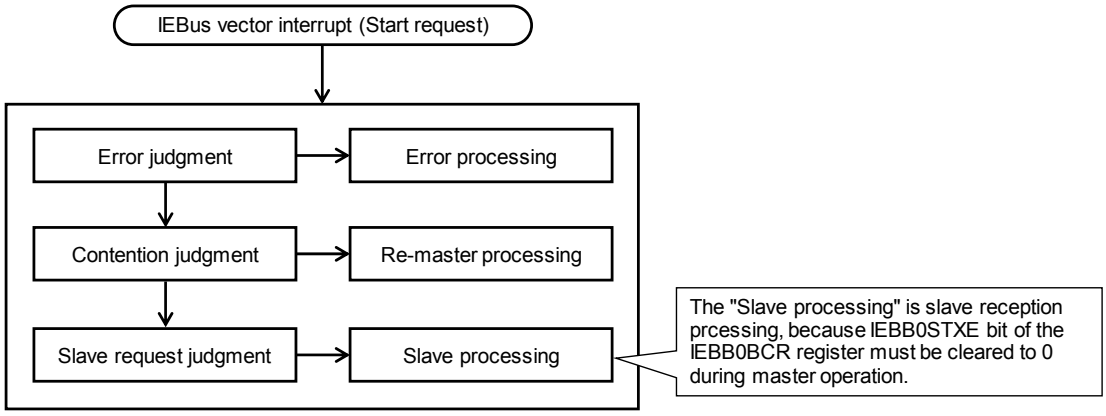
(3) Interrupt servicing examples

The interrupt servicing examples of master reception is shown below.

- IEBus vector interruput (Start request) processing flow example

  The IEBus vector interrupt (Start request) processing flow example is shown below.

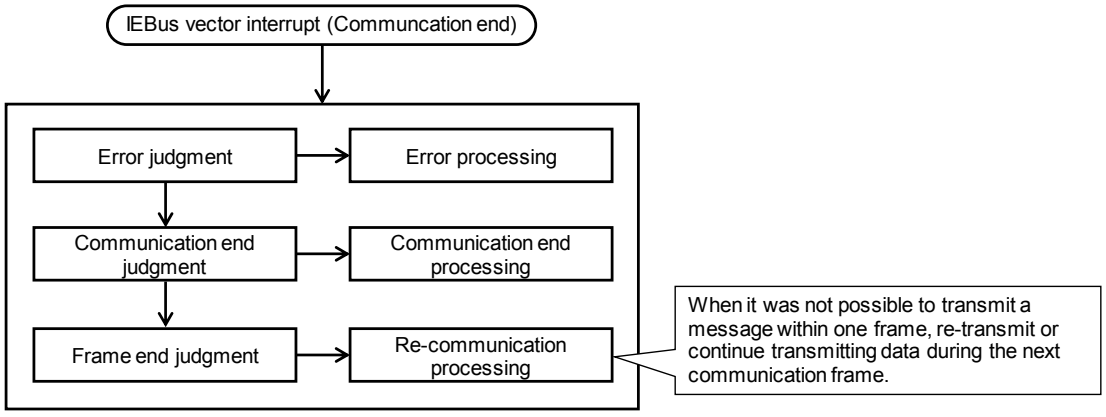
Figure 19-37. IEBus vector interrupt (Start request) processing flow example (Master reception)



- IEBus vector interruput (Communication end) processing flow example

  The IEBus vector interrupt (Communication end) processing flow example is shown below.

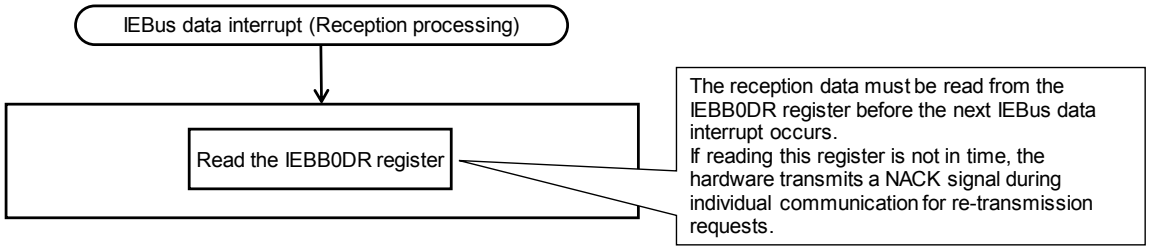
Figure 19-38. IEBus vector interrupt (Communication end) processing flow example (Master reception)



- IEBus data interruput (Reception data read request) processing flow example

  The IEBus data interrupt (Reception data read request) processing flow example is shown below.

Figure 19-39. IEBus data interrupt (Reception data read request) processing flow example (Master reception)



19.5.4 Slave transmission

The unit transfers data and commands to the master unit as a slave.

IEBus data interrupts (Transmission data write request) are used to write transmission data to the IEBB0DR register for each one-byte transfer.

(1) Register settings

After specifying the initial settings in Table 19-17, Initial setup, set up the registers below before starting communication.

Table 19-20. Standard initial processing

Register name	Function	Example
IEBB0DLR	Message length (other than during slave status transmission)	02H
IEBB0DR	Data (1st byte of data)	11H
IEBB0BCR	Communication startup processing	90H

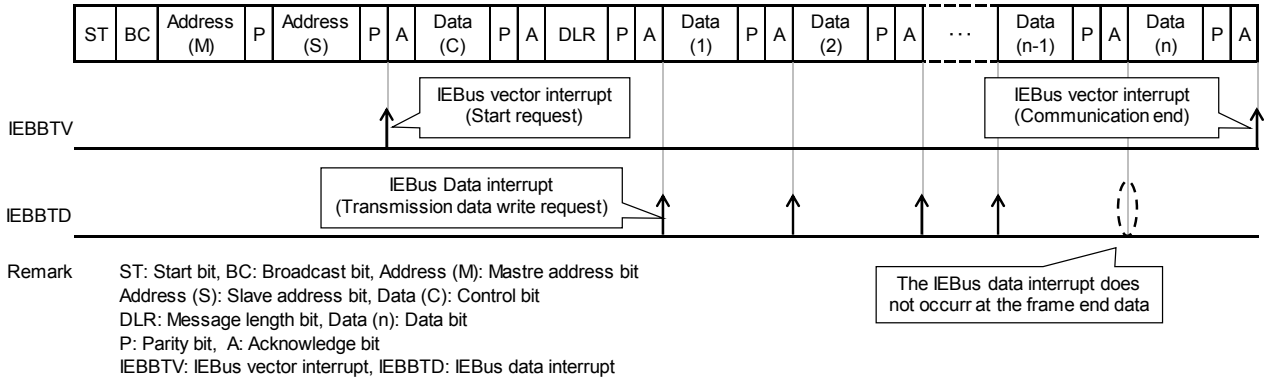
Caution When starting slave transmission, information such as the value to be set to the message length register (IEBB0DLR) and which data is to be returned (the value to be set to the IEBB0DR register) must be assigned in advance by the master, such as during separate communication.

(2) Interrupt occurrence timing

Figure 19-40, Figure 19-41 and Figure 19-42 show the interrupt occurrence timing of slave transmission.

- When the control bit 3H or 7H is received

Figure 19-40. Interrupt occurrence timing (Slave transmission) when the control bit 3H or 7H is received



Remark ST: Start bit, BC: Broadcast bit, Address (M): Mastre address bit
Address (S): Slave address bit, Data (C): Control bit
DLR: Message length bit, Data (n): Data bit
P: Parity bit, A: Acknowledge bit
IEBBTV: IEBus vector interrupt, IEBBTD: IEBus data interrupt

21.3 Operation

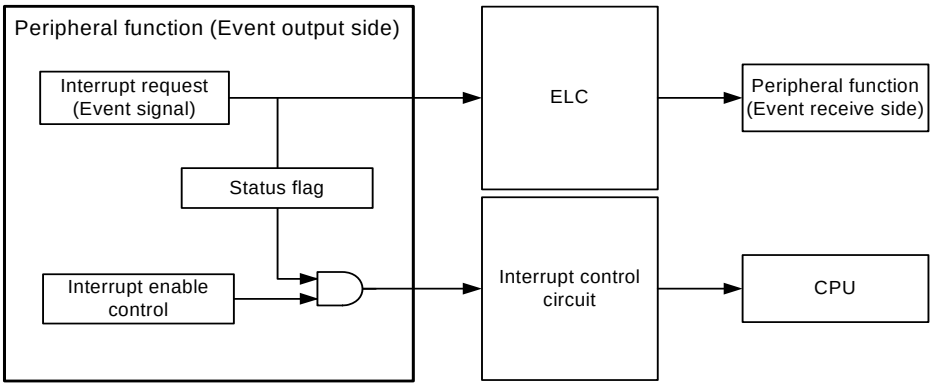
The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

In addition, event link operation can be performed without being influenced by the presence or absence of a CPU clock supply. However, the operating clock of a peripheral function needs to be supplied and be in an operational state.

Figure 21-3 shows the relationship between interrupt handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (see **Table 21-3 Correspondence Between Values Set to ELSELRn (n = 00 to 25) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 21-3. Relationship Between Interrupt Handling and ELC



22.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L and IF3H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, and the IF2L and IF2H and the IF3L and IF3H registers are combined to form 16-bit registers IF0, IF1, IF2 and IF3 they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 22-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5 CMPIF0	PIF4 SPMIF	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	LIN0TRMIF	RAMIF	TRJIIF0	TRDIF1	TRDIF0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	CLMIF PIF13

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03 LIN2STAIF LIN2IF	TMIF02 LIN2RVCIF	TMIF01 LIN2TRMIF	TMIF00	PIF8 RTCIF	IICAIF0	LIN0STAIF LIN0IF	LIN0RVCIF

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04 STIF2 CSIIF20	SRIF1 CSIIF11 IICIF11 IEBBTVIF	STIF1 CSIIF10 IICIF10 IEBBTDIF	PIF10 TMIF03H	PIF9 TMIF01H	PIF7 TMIF13H	PIF6 TMIF11H	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	CAN0CFRIF	CAN0WUPIF	CAN0ERRIF	KRIF	PIF11 LIN0WUPIF	PIF15 TMIF07	TMIF06 SREIF2	TMIF05 SRIF2 CSIIF21

(3) Bit error detection register (ECCER)

Figure 28-10. Format of Bit Error Detection Register (ECCER)

Address: F0203H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ECCER	-	-	-	-	-	-	-	DBERR

DBERR	Bit error detection flag
0	A 1-bit error detected.
1	A 2-bit error detected.

Cautions 1. The DBERR bit is cleared to 0 by writing 0.

2. If setting to 1 due to bit error detection and clearing to 0 by the CPU occur simultaneously, setting to 1 due to bit error detection has a priority.

3. If a bit error detection interrupt request (INTRAM) is not generated, the DBERR value is invalid.

(4) ECC test protect register (ECCTPR)

This register is used to prevent accidentally changing the setting of the ECCTMDR register to trigger entry to the ECC test mode.

Writing a value other than 07H prevents changes to the value of the ECCTMDR register

Figure 28-11. Format of ECC Test Protect Register (ECCTPR)

Address: F0204H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ECCTPR	-	-	-	-	-	TPR2	TPR1	TPR0

TPR2 to TPR0	ECC test protect bits
Other than 00000111	Access to the ECCTMDR register is disabled.
00000111	Access to the ECCTMDR register is enabled.

The voltage range in which to write, erase, or verify data in flash memory programming mode is shown in Table 31-5.

Table 31-5. Voltages at Which Data Can Be Written, Erased, or Verified

Voltages at which data can be written, erased, or verified	Operating frequency
2.7 V ≤ V_{DD} ≤ 5.5 V	1 MHz to 32 MHz

Remark For details about communication commands, see **31.4.4 Communication commands**.

31.4.3 Selecting communication mode

Communications modes of the RL78/F15 are as follows.

Table 31-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed [Note 2]	Frequency	Multiply Rate	
1-line mode (when flash memory programmer or an external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOL0
UART0 (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	–	–	TOOLTXD, TOOLRXD

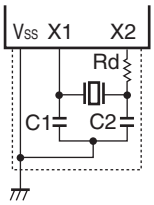
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

35.2 Oscillator Characteristics

35.2.1 Main System Clock Oscillator Characteristics

(T_A = -40 to +105°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator/ Crystal resonator		X1 clock oscillation frequency (fx)	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines.**
- **Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as V_{SS}.**
- **Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

2. Customers are requested to consult the resonator manufacturer to select an appropriate resonator and to determine the proper oscillation constant. Customers are also requested to adequately evaluate the oscillation on their system. Determine the X1 clock oscillation stabilization time using the oscillation stabilization time of the oscillation stabilization time counter status register (OSTC) and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.