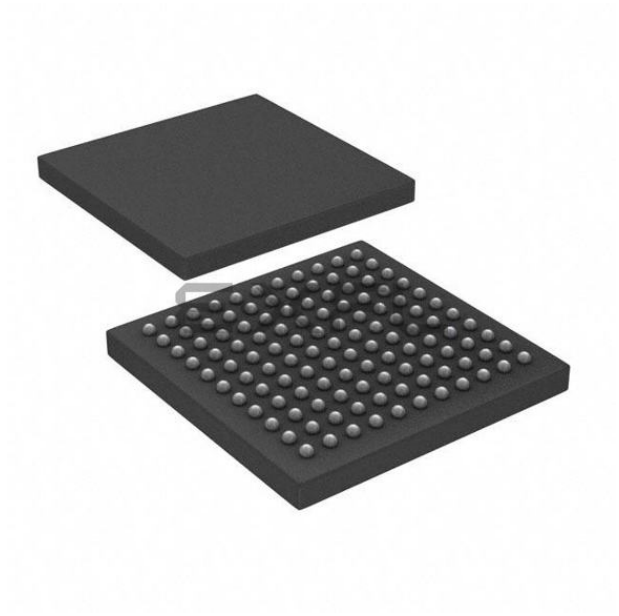


Welcome to E-XFL.COM

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	V850ES
Core Size	32-Bit Single-Core
Speed	20MHz
Connectivity	CSI, EBI/EMI, I²C, UART/USART, USB
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	80
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	40K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 3.6V
Data Converters	A/D 12x10b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	121-LFBGA
Supplier Device Package	121-LFBGA (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3794f1-cah-a

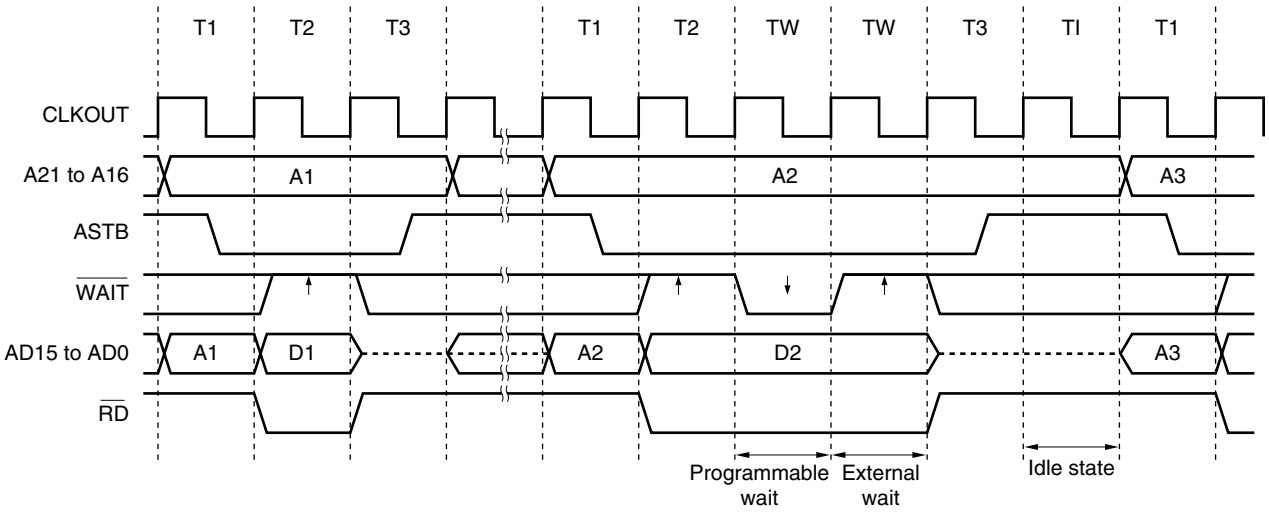
(11/11)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFD26H	CSIB2 transmit data register	CB2TX	R/W			√	0000H
FFFFFD26H	CSIB2 transmit data register L	CB2TXL			√		00H
FFFFFD30H	CSIB3 control register 0	CB3CTL0		√	√		01H
FFFFFD31H	CSIB3 control register 1	CB3CTL1		√	√		00H
FFFFFD32H	CSIB3 control register 2	CB3CTL2			√		00H
FFFFFD33H	CSIB3 status register	CB3STR		√	√		00H
FFFFFD34H	CSIB3 receive data register	CB3RX	R			√	0000H
FFFFFD34H	CSIB3 receive data register L	CB3RXL			√		00H
FFFFFD36H	CSIB3 transmit data register	CB3TX	R/W			√	0000H
FFFFFD36H	CSIB3 transmit data register L	CB3TXL			√		00H
FFFFFD40H	CSIB4 control register 0	CB4CTL0		√	√		01H
FFFFFD41H	CSIB4 control register 1	CB4CTL1	R/W	√	√		00H
FFFFFD42H	CSIB4 control register 2	CB4CTL2			√		00H
FFFFFD43H	CSIB4 status register	CB4STR		√	√		00H
FFFFFD44H	CSIB4 receive data register	CB4RX	R			√	0000H
FFFFFD44H	CSIB4 receive data register L	CB4RXL			√		00H
FFFFFD46H	CSIB4 transmit data register	CB4TX	R/W			√	0000H
FFFFFD46H	CSIB4 transmit data register L	CB4TXL			√		00H
FFFFFD80H	IIC shift register 0	IIC0			√		00H
FFFFFD82H	IIC control register 0	IICC0		√	√		00H
FFFFFD83H	Slave address register 0	SVA0			√		00H
FFFFFD84H	IIC clock select register 0	IICCL0		√	√		00H
FFFFFD85H	IIC function expansion register 0	IICX0		√	√		00H
FFFFFD86H	IIC status register 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC flag register 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC shift register 1	IIC1			√		00H
FFFFFD92H	IIC control register 1	IICC1		√	√		00H
FFFFFD93H	Slave address register 1	SVA1			√		00H
FFFFFD94H	IIC clock select register 1	IICCL1		√	√		00H
FFFFFD95H	IIC function expansion register 1	IICX1		√	√		00H
FFFFFD96H	IIC status register 1	IICS1	R	√	√		00H
FFFFFD9AH	IIC flag register 1	IICF1	R/W	√	√		00H
FFFFFDA0H	IIC shift register 2	IIC2			√		00H
FFFFFDA2H	IIC control register 2	IICC2		√	√		00H
FFFFFDA3H	Slave address register 2	SVA2			√		00H
FFFFFDA4H	IIC clock select register 2	IICCL2		√	√		00H
FFFFFDA5H	IIC function expansion register 2	IICX2		√	√		00H
FFFFFDA6H	IIC status register 2	IICS2	R	√	√		00H
FFFFFDAAH	IIC flag register 2	IICF2	R/W	√	√		00H
FFFFFF40H	USB clock control register	UCKSEL		√	√		00H
FFFFFF41H	USB function control register	UFCKMSK		√	√		03H

5.9 Bus Timing

Typical bus timing diagrams are shown below.

Figure 5-4. Multiplexed Bus Read Timing (Bus Size: 16 Bits, 16-bit Access)



Remarks 1. The validity of data if 8-bit access is executed when 16-bit access has been specified is shown below.

8-bit access	Odd address	Even address
AD15 to AD8	Valid data	Invalid data
AD7 to AD0	Invalid data	Valid data

2. The broken lines indicate high impedance.

7.4.1 Interval timer mode (TPnMD2 to TPnMD0 bits = 000)

In the interval timer mode, setting the TPnCTL0.TPnCE bit to 1 generates an interrupt request signal (INTTPnCC0) at a specified interval. Setting the TPnCE bit to 1 can also start the timer, which then outputs a square wave whose half cycle is equal to the interval from the TOPn0 pin.

Usually, the TPnCCR1 register is not used in the interval timer mode. Mask interrupts from this register by setting the interrupt mask flag (TPnCCMK1).

Remarks 1. For how to set the TOPn0 pin, see **Table 7-2 Pins Used by TMPn** and **Table 4-15 Settings When Pins Are Used for Alternate Functions**.

2. For how to enable the INTTPnCC0 interrupt signal, see **CHAPTER 22 INTERRUPT SERVICING/ EXCEPTION PROCESSING FUNCTION**.

Figure 7-6. Configuration of Interval Timer

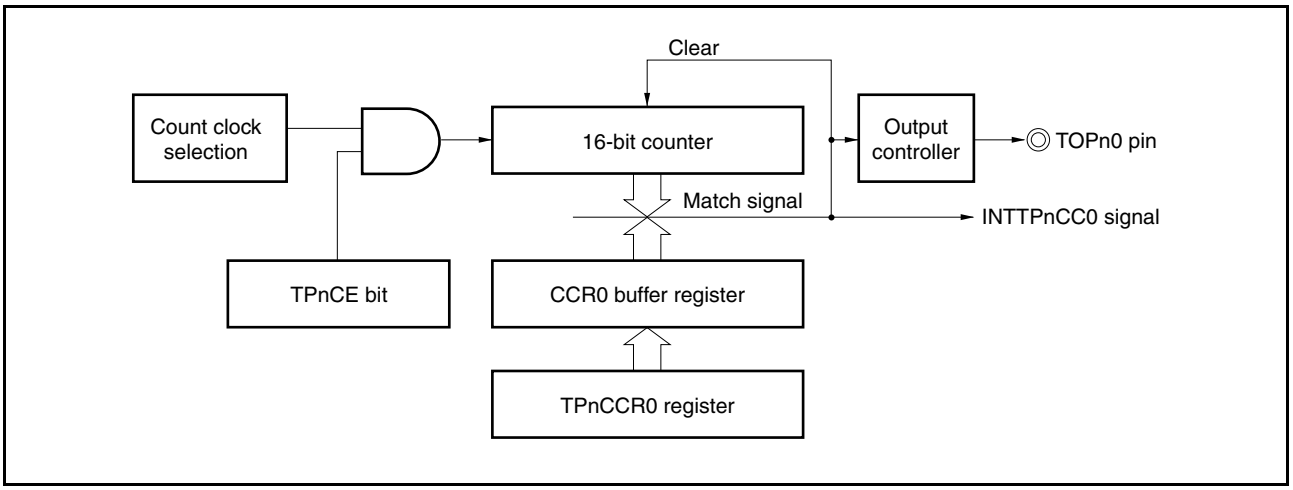


Figure 7-7. Basic Timing of Operation in Interval Timer Mode

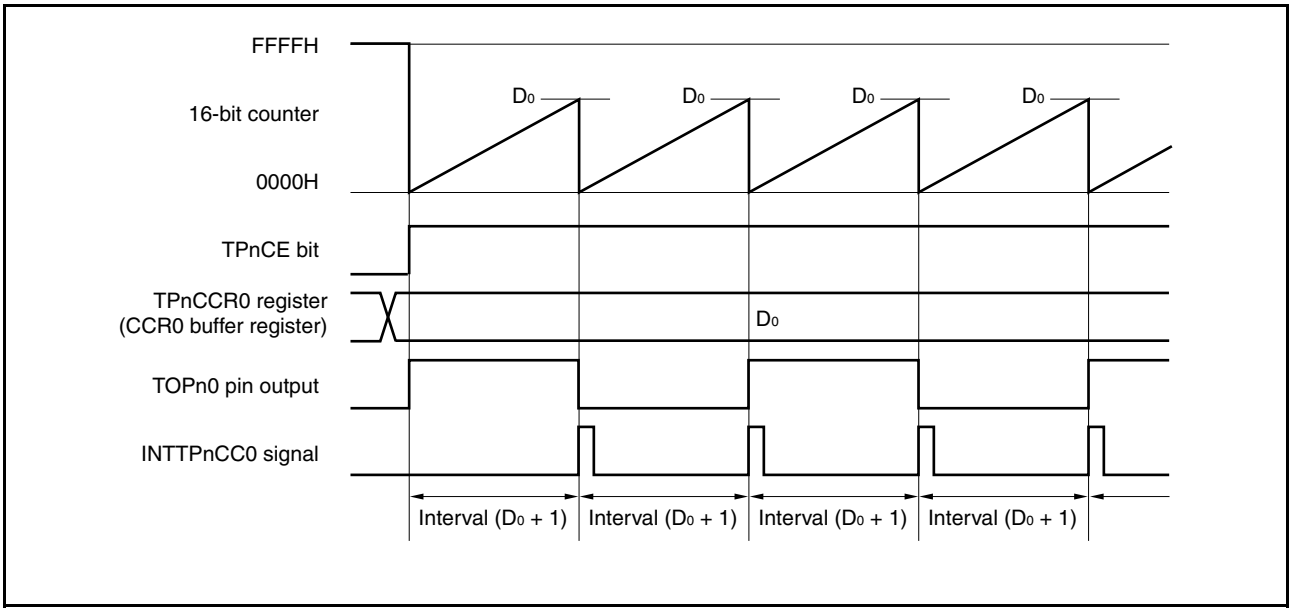


Figure 7-46. Register Settings in PWM Output Mode (2/2)

(d) TMPn I/O control register 2 (TPnIOC2)

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

These bits select the valid edge of the external trigger input.

(e) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading this register.

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If the TPnCCR0 register is set to D_0 and the TPnCCR1 register is set to D_1, the PWM waveform is as follows:

PWM waveform cycle = (D_0 + 1) × Count clock cycle
PWM waveform active level width = D_1 × Count clock cycle

Remark TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the PWM output mode.

(1) Basic counter operation

The basic operation of the 16-bit counter is described below. For more details, see the descriptions of each operating mode.

(a) Starting counting

TMQ0 starts counting from FFFFH in all operating modes, and increments as follows: FFFFH, 0000H, 0001H, 0002H, 0003H….

(b) Clearing TMQ0

TMQ0 is cleared to 0000H when its value matches the value of the compare register or when the value of TMQ0 is captured upon the input of a valid capture trigger signal.

Note that when TMQ0 increments from FFFFH to 0000H after it starts counting and immediately following an overflow, it does not mean that TMQ0 has been cleared. Consequently, the INTTQ0CCm interrupt is not generated in this case (m = 0 to 3).

(c) Overflow

TMQ0 overflows after it increments from FFFFH to 0000H in free-running timer mode and pulse width measurement mode. An overflow sets the TQ0OPT0.TQ0OVF bit to 1 and generates an interrupt request signal (INTTQ0OV). Note that INTTQ0OV will not be generated in the following cases:

- When TMQ0 has just started counting.
- When the compare value at which TMQ0 is cleared is specified as FFFFH.
- In pulse width measurement mode, when TMQ0 increments from FFFFH to 0000H after being cleared when its value of FFFFH was captured.

Caution After the INTTQ0OV overflow interrupt request signal occurs, be sure to confirm that the overflow flag (TQ0OVF) is set to 1.

(d) Reading TMQ0 while it is incrementing

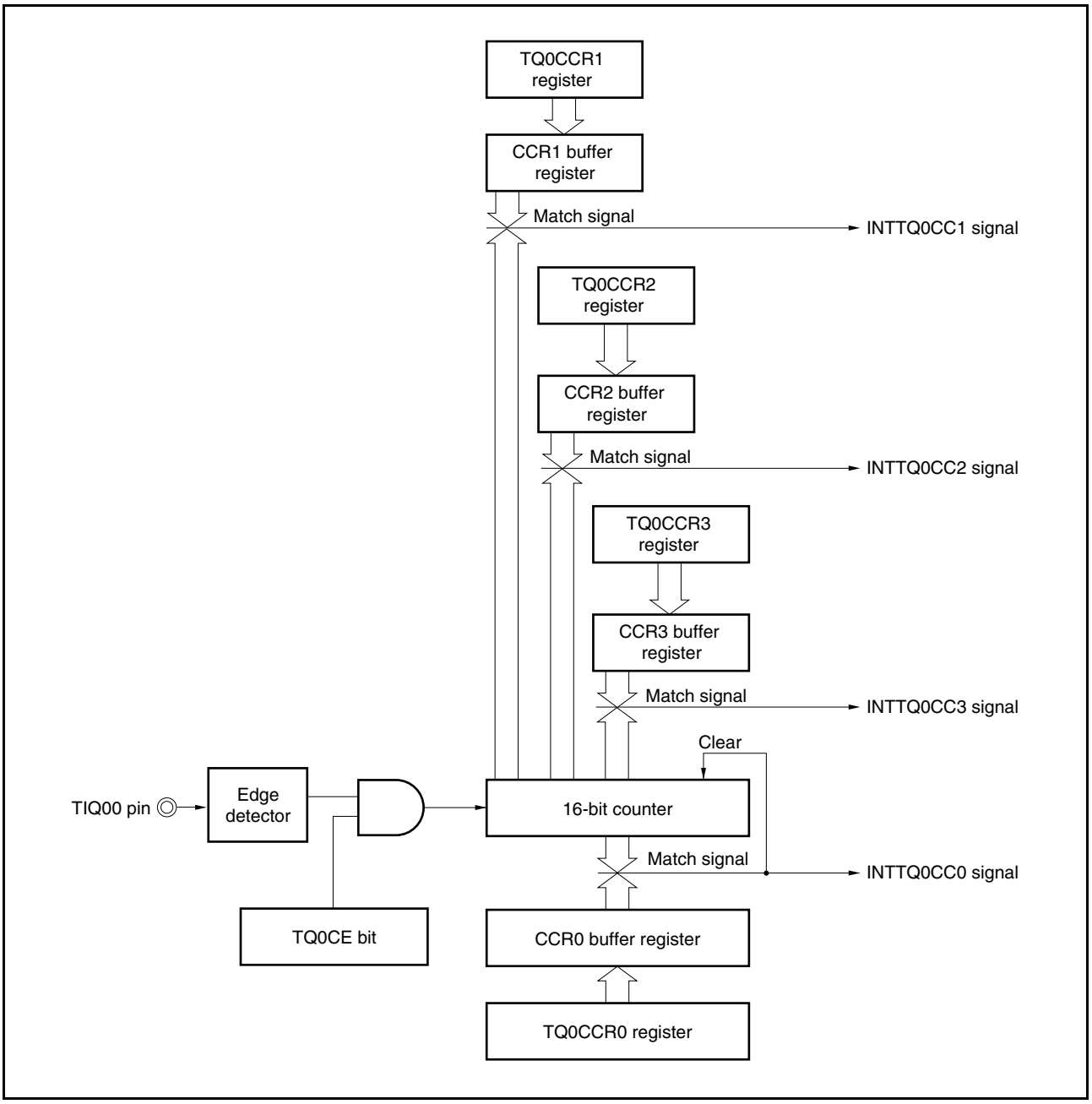
TMQ0 can be read while it is incrementing by using the TQ0CNT register.

Specifically, the value of TMQ0 can be read by reading the TQ0CNT register while the TQ0CLT0.TQ0CE bit is 1. Note, however, that when the TQ0CLT0.TQ0CE bit is 0, the value of TMQ0 is always FFFFH and the value of the TQ0CNT register is always 0000H.

(c) Operation of TQ0CCR1 to TQ0CCR3 registers

The TQ0CCR1 to TQ0CCR3 registers are configured as follows in the external event count mode.

Figure 8-23. Configuration of TQ0CCR1 to TQ0CCR3 Registers



(1) Operations in PWM output mode

Figure 8-47. Timing and Processing of Operations in PWM Output Mode (1/2)

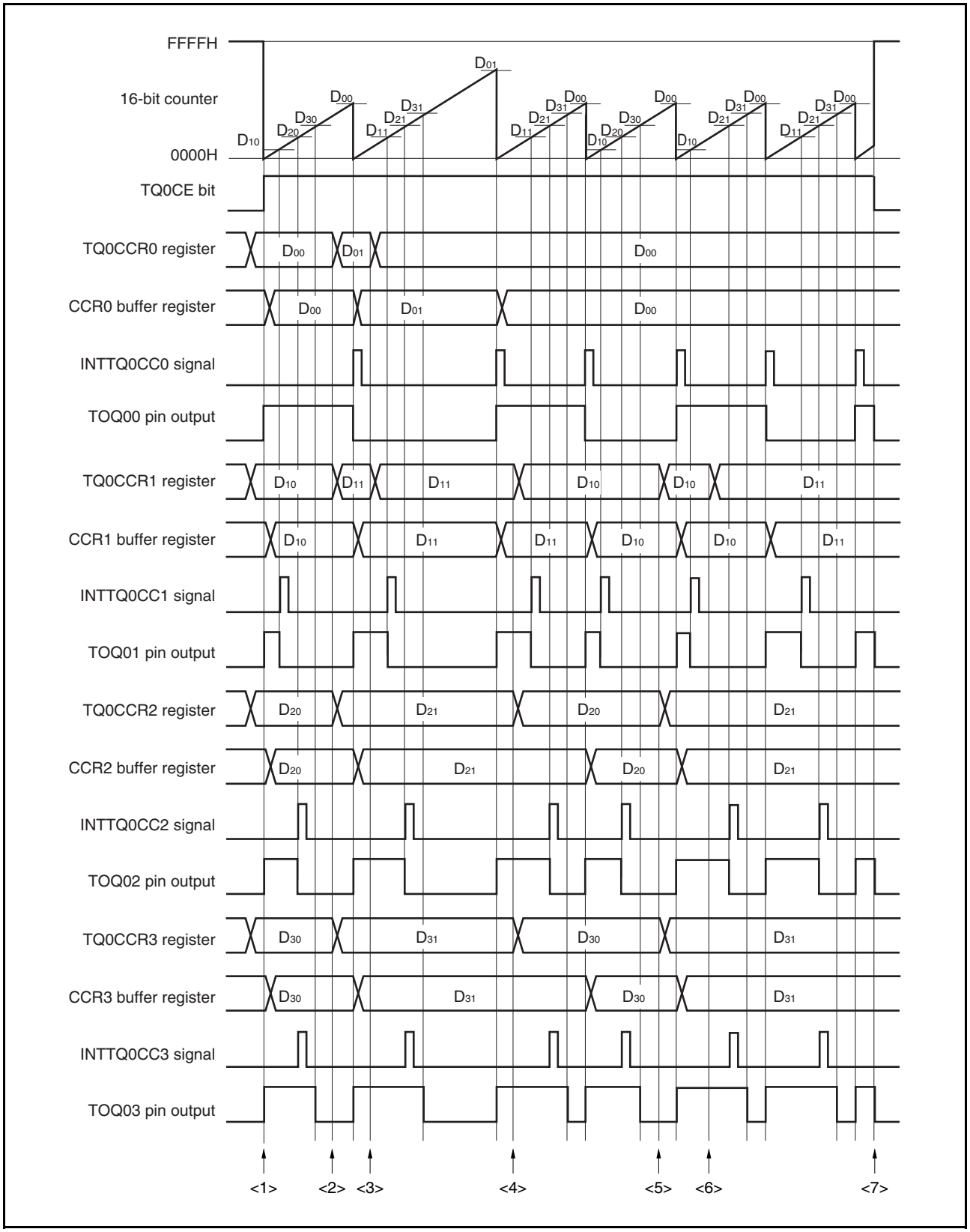
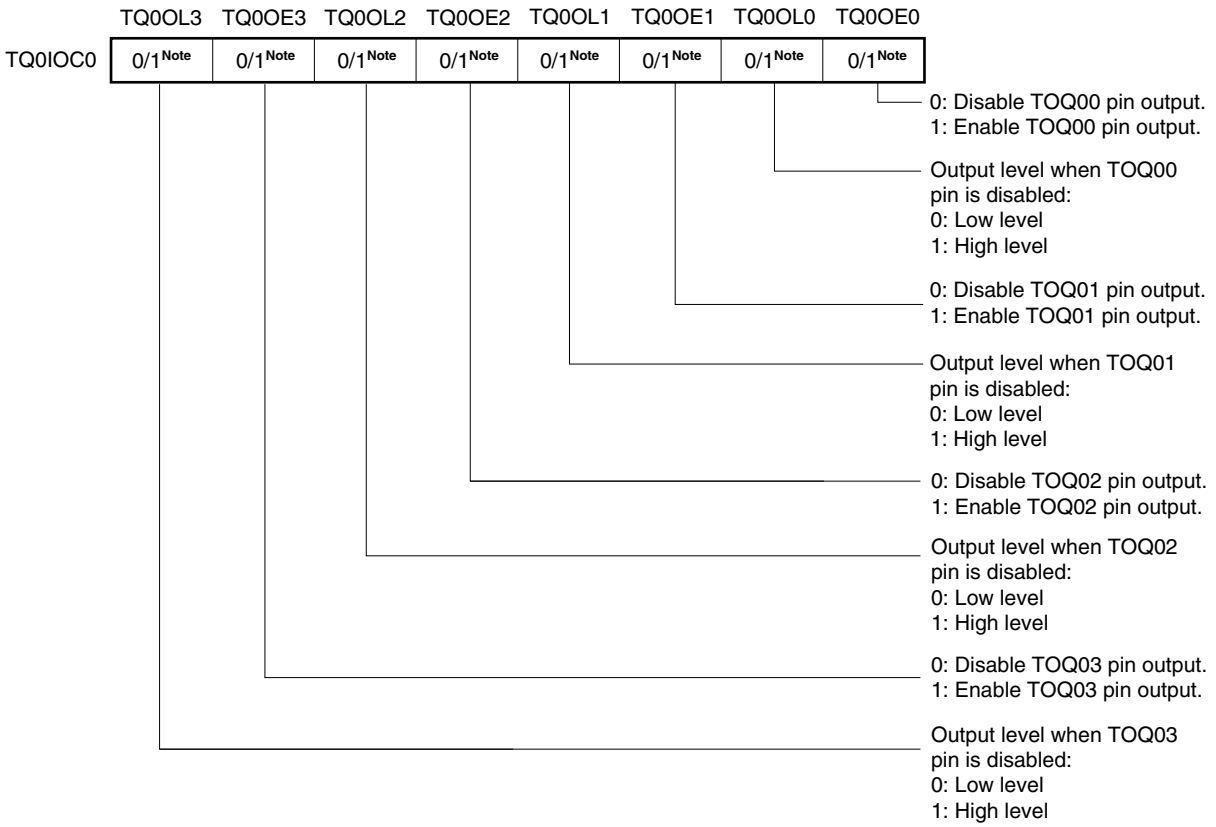


Figure 8-55. Register Settings in Free-Running Timer Mode (2/3)

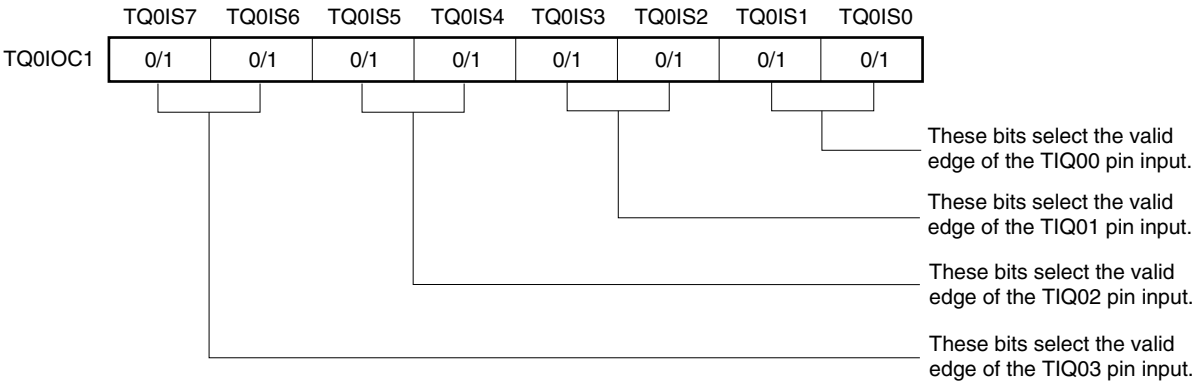
(c) TMQ0 I/O control register 0 (TQ0IOC0)



Note The TOQ0m pin cannot be used when the TIQ0m pin is being used.

Remark m = 0 to 3

(d) TMQ0 I/O control register 1 (TQ0IOC1)



16.4 Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 10H.

(1/2)

After reset: 10H R/W Address: UA0CTL0 FFFFFA00H, UA1CTL0 FFFFFA10H, UA2CTL0 FFFFFA20H, UA3CTL0 FFFFFA30H, UA4CTL0 FFFFFA40H, UA5CTL0 FFFFFA50H

UAnCTL0 (n = 0 to 5)

<7>	<6>	<5>	<4>	3	2	1	0
UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

UAnPWR	UARTAn operation control
0	Disable UARTAn operation (UARTAn reset asynchronously)
1	Enable UARTAn operation

The UARTAn operation is controlled by the UAnPWR bit. The TXDAn pin output is fixed to high level by clearing the UAnPWR bit to 0 (fixed to low level if UAnOPT0.UAnTDL bit = 1).

UAnTXE	Transmission operation enable
0	Disable transmission operation
1	Enable transmission operation

- To start transmission, set the UAnPWR bit to 1 and then set the UAnTXE bit to 1. To stop transmission, clear the UAnTXE bit to 0 and then UAnPWR bit to 0.
- To initialize the transmission unit, clear the UAnTXE bit to 0, wait for two cycles of the base clock, and then set the UAnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **16.7 (1) (a) Base clock**).
- When UARTAn operation is enabled (UAnPWR bit = 1) and the UAnTXE bit is set to 1, transmission is enabled after at least two cycles of the base clock (f_{UCLK}) have elapsed.

UAnRXE	Reception operation enable
0	Disable reception operation
1	Enable reception operation

- To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1. To stop reception, clear the UAnRXE bit to 0 and then UAnPWR bit to 0.
- To initialize the reception unit, clear the UAnRXE bit to 0, wait for two cycles of the base clock, and then set the UAnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **16.7 (1) (a) Base clock**).
- When UARTAn operation is enabled (UAnPWR bit = 1) and the UAnRXE bit is set to 1, reception is enabled after at least two cycles of the base clock (f_{UCLK}) have elapsed. If a start bit is received before reception is enabled, the start bit is ignored.

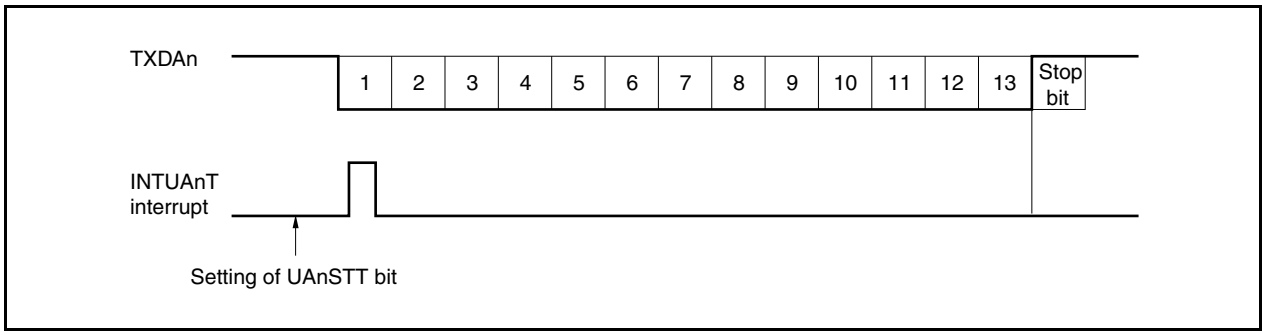
16.6.8 SBF transmission

When the UAnCTL0.UAnPWR bit and UAnCTL0.UAnTXE bit are 1, the transmission enabled status is entered, and SBF transmission is started by setting the SBF transmission trigger (UAnOPT0.UAnSTT bit) to 1.

Thereafter, a low level signal having a length of 13 to 20 bits, as specified by the UAnOPT0.UAnSLS2 to AnOPT0.UAnSLS0 bits, is output. A transmission enable interrupt request signal (INTUAnT) is generated upon the start of SBF transmission. Following the end of SBF transmission, the UAnSTT bit is automatically cleared.

Transmission is suspended until the data to be transmitted next is written to the UAnTX register, or until the SBF transmission trigger (UAnSTT bit) is set.

Figure 16-13. Example of SBF Transmission



(4) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{XX}}{2^{m+1} \times k} \text{ [bps]}$$

Remark f_{UCLK} = Frequency of base clock selected by the UC0CTL1.UC0CKS3 to UC0CTL1.UC0CKS0 bits
f_{XX}: Main clock frequency
m = Value set using the UC0CTL1.UC0CKS3 to UC0CTL1.UC0CKS0 bits (m = 0 to 10)
k = Value set using the UC0CTL2.UC0BRS7 to UC0CTL2.UC0BRS0 bits (k = 4 to 255)

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{f_{XX}}{2^{m+1} \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in (5) Allowable baud rate range during reception.

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

Similarly, obtaining the maximum allowable data frame length yields the following.

$$\frac{10}{11} \times FLmax = 11 \times BL - \frac{k + 2}{2 \times k} \times BL = \frac{21k - 2}{2 \times k} BL$$

$$FLmax = \frac{21k - 2}{20\ k} BL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

Obtaining the allowable baud rate error for UARTC0 and the destination from the above-described equations yields the following.

Table 17-8. Maximum/Minimum Allowable Baud Rate Error (11-Bit Length)

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	−2.43%
8	+3.53%	−3.61%
20	+4.26%	−4.31%
50	+4.56%	−4.58%
100	+4.66%	−4.67%
255	+4.72%	−4.73%

Remarks 1. The reception accuracy depends on the bit count in 1 frame, the base clock frequency (f_{UCLK}), and the division ratio (k). The higher the base clock frequency (f_{UCLK}) and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UC0CTL2.UC0BRS7 to UC0CTL2.UC0BRS0 bits (n = 0 to 2)

18.3.2 CSIB4 and UARTA0 mode switching

In the V850ES/JG3-L, CSIB4 and UARTA0 share pins and therefore cannot be used simultaneously. To use the CSIB4 function, specify the CSIB4 mode in advance by using the PMC3, PFC3, and PFCE3L registers.

Switching the operation mode between CSIB4 and UARTA0 is described below.

Caution Transmission and reception by CSIB4 and UARTA0 are not guaranteed if these operation modes are switched during transmission or reception. Be sure to disable the serial interface that is not being used.

Figure 18-3. Switching CSIB4 and UARTA0 Operation Modes

After reset: 0000H R/W Address: FFFFF446H, FFFFF447H

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38

7	6	5	4	3	2	1	0
0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 0000H R/W Address: FFFFF466H, FFFFF467H

	15	14	13	12	11	10	9	8
PFC3	0	0	0	0	0	0	PFC39	PFC38

7	6	5	4	3	2	1	0
0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

PMC32	PFCE32	PFC32	Operation mode
0	×	×	Port I/O mode
1	0	0	ASCKA0 mode
1	0	1	$\overline{\text{SCKB4}}$ mode

PMC3n	PFC3n	Operation mode
0	×	Port I/O mode
1	0	UARTA0 mode
1	1	CSIB4 mode

Remarks 1. n = 0, 1

2. × = don't care

(3) IIC flag registers 0 to 2 (IICF0 to IICF2)

The IICFn register sets the I²C0n operation mode and indicates the I²C bus status.

These registers can be read or written in 8-bit or 1-bit units. However, the STCFn and IICBSYn bits are read-only.

IICRSVn enables/disables the communication reservation function (see **19.14 Communication Reservation**).

The initial value of the IICBSYn bit is set by using the STCENn bit (see **19.15 Cautions**).

The IICRSVn and STCENn bits can be written only when operation of I²C0n is disabled (IICCn.IICEn bit = 0). After operation is enabled, IICFn can be read (n = 0 to 2).

Reset sets these registers to 00H.

19.6.7 Wait state cancellation method

In the case of I²C0n, a wait state can be canceled normally in the following ways (n = 0 to 2).

- By writing data to the IICn register
- By setting the IICCn.WRELn bit to 1 (wait state cancellation)
- By setting the IICCn.STTn bit to 1 (start condition generation)
- By setting the IICCn.SPTn bit to 1 (stop condition generation)

If any of these wait state cancellation actions is performed, I²C0n will cancel the wait state and restart communication.

When canceling the wait state and sending data (including addresses), write data to the IICn register.

To receive data after canceling the wait state, or to complete data transmission, set the WRELn bit to 1.

To generate a restart condition after canceling the wait state, set the STTn bit to 1.

To generate a stop condition after canceling the wait state, set the SPTn bit to 1.

Cancel each wait state only once.

For example, if data is written to the IICn register following wait state cancellation by setting the WRELn bit to 1, a conflict between the SDA0n line change timing and the IICn register write timing may result in the data output to the SDA0n line being incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICCn.IICEn bit to 0 will stop communication, enabling the wait state to be cancelled.

If the I²C bus deadlocks due to noise, etc., setting the IICCn.LRELn bit to 1 causes the communication to stop, enabling the wait state to be cancelled.

(4) Regulator protection register (REGPR)

The REGPR register is used to protect the regulator output voltage level control register 0 (REGOVL0) so that illegal data is not written to REGOVL0. Data cannot be written to the REGOVL0 register unless enabling data (C9H) is written to the REGPR register. Only two types of data, C9H (enabling data) and 00H (protection data), can be written to the REGPR register. Writing any other value is prohibited. (If a value other than C9H or 00H is written to the REGPR register, the written value is set to prohibit a write access to the REGOVL0 register, but the operation is not guaranteed.)

This register can be read or written only in 8-bit units (accessing it in 1-bit units is prohibited).

Reset sets this register to 00H (protection data status).

After reset: 00H R/W Address: FFFFF331H

	7	6	5	4	3	2	1	0
REGPR	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0

- **Protection data status: REGPR = 00H**

  In this status, the REGOVL0 register is protected from an illegal write access. In the protection data status, a value is not written to the REGOVL0 register even if an attempt is made to write it, and the REGOVL0 register holds the previous value.

  Be sure to set REGPR to 00H, except when changing the value of the REGOVL0 register, in order to avoid unexpected malfunction.

- **Enabling data status: REGPR = C9H**

  In this status, a write access to the REGOVL0 register is enabled.

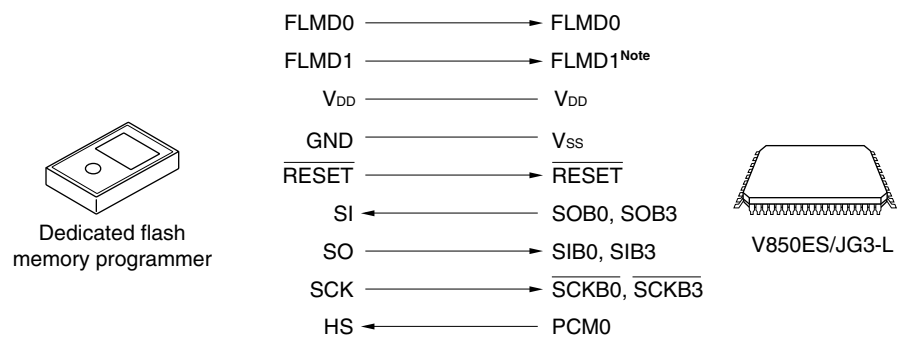
  - Transition from normal mode → low-voltage STOP mode
    See **24.6.1 Setting and operation status.**
  - Transition of subclock operation mode → low-voltage subclock operation mode
    See **24.7.1 Setting and operation status.**
  - Transition of subclock operation mode → low-voltage sub-IDLE mode
    See **24.8.1 Setting and operation status.**

(3) CSIB0 + HS, CSIB3 + HS

Serial clock: 2.4 kHz to 5 MHz (MSB first)

The V850ES/JG3-L operates as a slave.

Figure 31-5. Communication with Dedicated Flash Memory Programmer (CSIB0 + HS, CSIB3 + HS)



Note Connect the FLMD1 pin to the flash memory programmer or connect to GND via a pull-down resistor on the board.

Table 31-8. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode Immediately After Reset Ends

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

(3) Serial interface pin

The following shows the pins used by each serial interface.

Table 31-9. Pins Used by Serial Interfaces

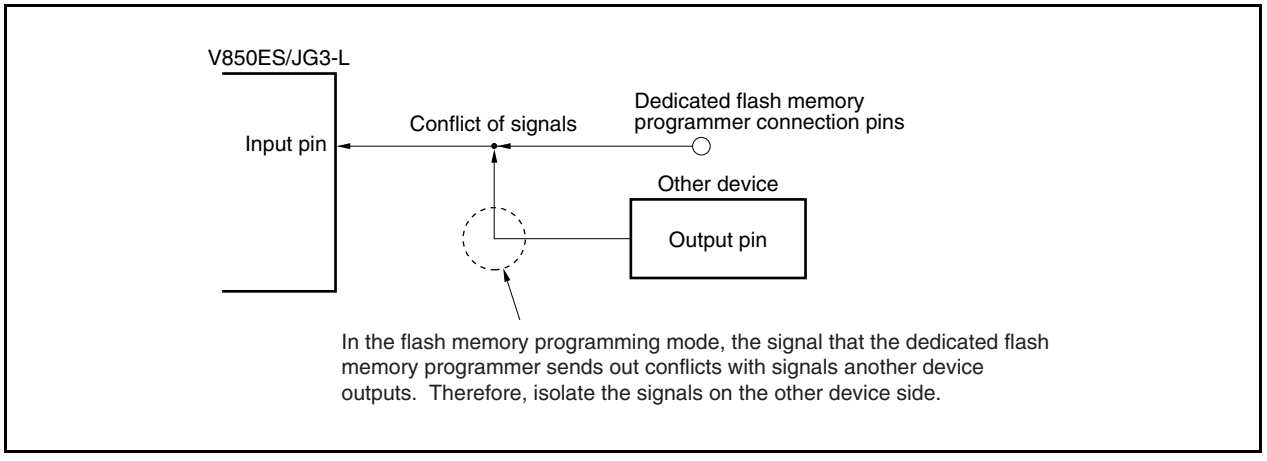
Serial Interface	Pins Used
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB3	SOB3, SIB3, $\overline{\text{SCKB3}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0
CSIB3 + HS	SOB3, SIB3, $\overline{\text{SCKB3}}$, PCM0

When connecting a dedicated flash memory programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(a) Conflict of signals

When the dedicated flash memory programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

Figure 31-12. Conflict of Signals (Serial Interface Input Pin)



32.2.3 Allocation of user resources

The user must prepare the following resources to perform communication between MINICUBE2 and the target device and implement each debug function. These items need to be set in the user program or using the compiler options.

(1) Allocation of memory space

The shaded portions in Figure 32-5 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated to these spaces. These spaces must be secured so as not to be used by the user program.

(2) Security ID setting

The ID code must be embedded in the area between 0000070H and 0000079H in Figure 32-5, to prevent the memory from being read by an unauthorized person. For details, see **32.3 ROM Security Function**.

D.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock i	Execution Clock r	Execution Clock l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R