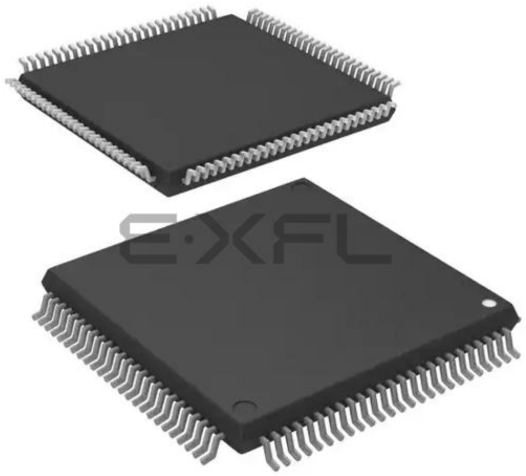


Welcome to E-XFL.COM

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	V850ES
Core Size	32-Bit Single-Core
Speed	64MHz
Connectivity	CSI, EBI/EMI, I²C, UART/USART
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	56
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	3.5V ~ 5.5V
Data Converters	A/D 10x12b, 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3453gc-r-8ea-a

(1) Interrupt status saving registers (EIPC, EIPSW)

There are two interrupt status saving registers, EIPC and EIPSW.

Upon occurrence of a software exception or a maskable interrupt, the contents of the program counter (PC) are saved to EIPC and the contents of the program status word (PSW) are saved to EIPSW (upon occurrence of a non-maskable interrupt (NMI), the contents are saved to the NMI status saving registers (FEPC, FEPSW)).

The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (see **20.9 Periods in Which CPU Does Not Acknowledge Interrupts**).

The current PSW contents are saved to EIPSW.

Since there is only one set of interrupt status saving registers, the contents of these registers must be saved by the program when multiple interrupt servicing is enabled.

Bits 31 to 26 of EIPC and bits 31 to 8 of EIPSW are reserved (fixed to 0) for future function expansion.

When the RETI instruction is executed, the values in EIPC and EIPSW are restored to the PC and PSW, respectively.

Register	31					26	25 ... 0	After reset
EIPC	0	0	0	0	0	0	(PC contents saved)	0xxxxxxxH (x: Undefined)

Register	31 ... 8	7 ... 0	After reset
EIPSW	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	(PSW contents saved)	000000xxH (x: Undefined)

4.3.3 Port 2

Port 2 can be set to the input or output mode in 1-bit units.

Port 2 has an alternate function as the following pins.

Table 4-8. Alternate-Function Pins of Port 2

Pin Name	Pin No.				Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3					
	GC	GC	GF	F1			
P20	23	28	56	P3	TOB1T1/TIB11/TOB11	I/O	Provided
P21	24	29	57	N3	TOB1B1/TIB12/TOB12	I/O	
P22	25	30	58	P4	TOB1T2/TIB13/TOB13	I/O	
P23	26	31	59	N4	TOB1B2/TIB10	I/O	
P24	27	32	60	M4	TOB1T3/EVTB1	I/O	
P25	28	33	61	M5	TOB1B3/TRGB1	I/O	
P26	29	34	62	M6	TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1	I/O	
P27	36	45	73	M11	DMS[Notes 2, 3]	Input	

Notes 1. Software pull-up function

2. V850E/IG3 only

3. The P27 pin also functions as an on-chip debug pin. The on-chip debug function or port function (including the alternate functions) can be selected by using the level of the $\overline{\text{DRST}}$ pin, as shown in the table below.

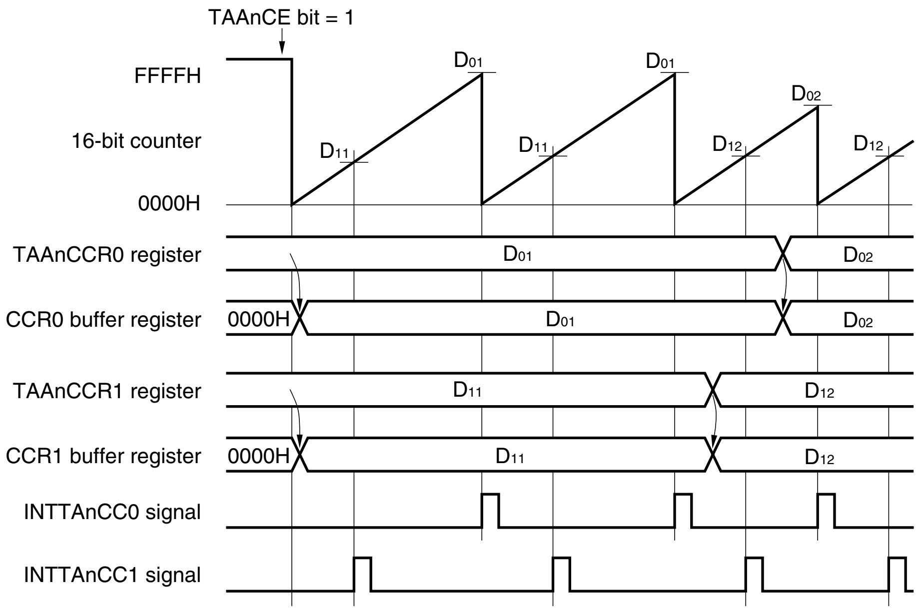
Port 2 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P27	DMS

Caution When P20 to P25 are used as TOB1T1 to TOB1T3 and TOB1B1 to TOB1B3, they go into a high-impedance state by inputting the following active signal.

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)
F1 (V850E/IG3): 161-pin plastic FBGA (10 × 10)

Figure 6-7. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of the TAAnCCR0 register
D_{11}, D_{12}: Set values of the TAAnCCR1 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0 to 4

Figure 6-26. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TAAm I/O control register 2 (TAAmIOC2)

							TAAmEES1	TAAmEES0	TAAmETS1	TAAmETS0
TAAmIOC2	0	0	0	0	0	0	0/1	0/1

Select valid edge of external trigger input (TIAm0 pin)

(e) TAAm counter read buffer register (TAAmCNT)

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

(f) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)

If D_0 is set to the TAAmCCR0 register and D_1 to the TAAmCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle
Active level width = $D_1 \times$ Count clock cycle

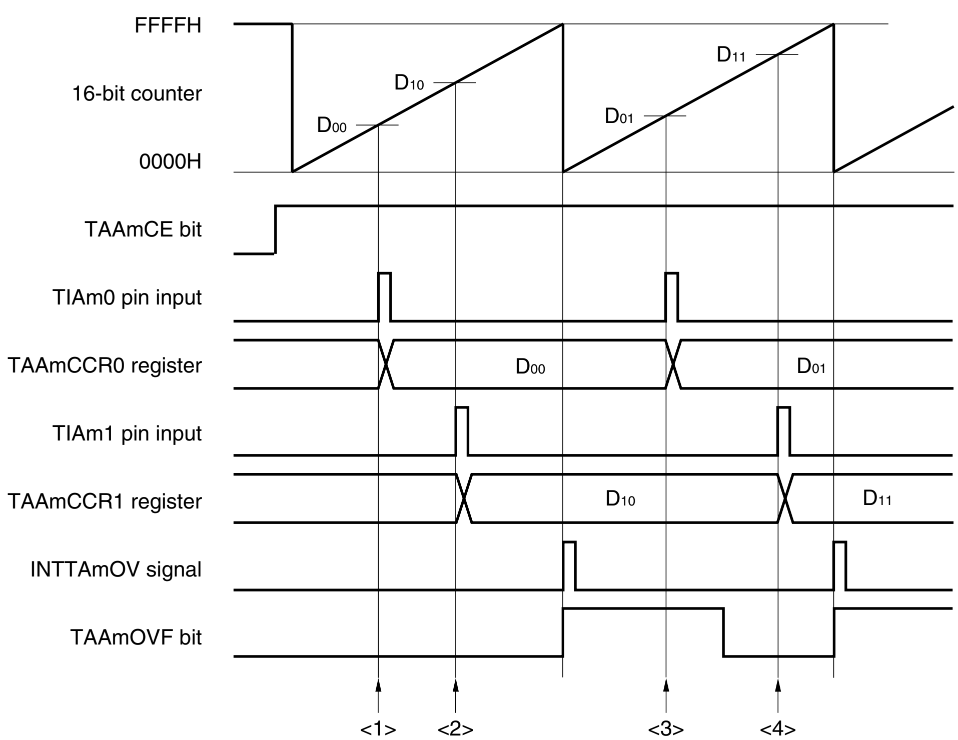
Remarks 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the external trigger pulse output mode.

2. V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TAAmCCR0 register (setting of the default value of the TIAm0 pin input).

<2> Read the TAAmCCR1 register (setting of the default value of the TIAm1 pin input).

<3> Read the TAAmCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

<4> Read the TAAmCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

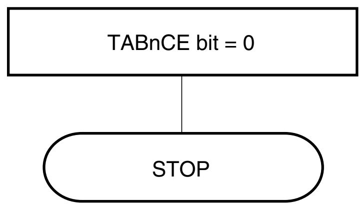
Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.

Use software when using two capture registers. An example of how to use software is shown below.

Figure 7-9. Software Processing Flow in Interval Timer Mode (2/2)

<2> Count operation stop flow

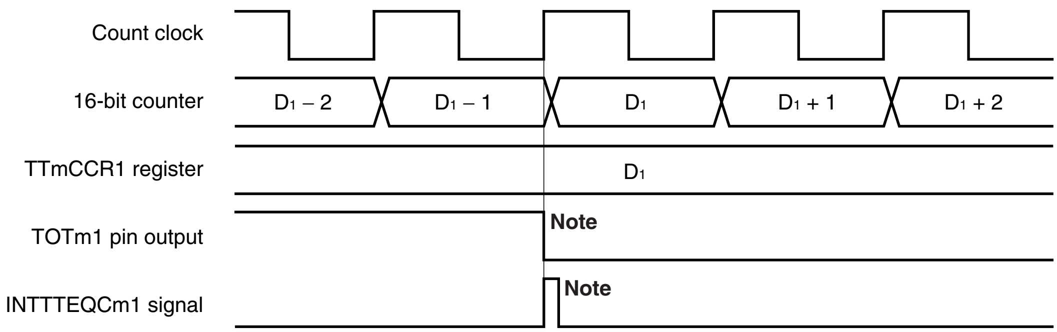


The counter is initialized and counting is stopped by clearing the TABnCE bit to 0. The output level of the TOBn0 pin is as specified by the TABnIOC0 register.

Remark n = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTTEQCm1)

The generation timing of the INTTTEQCm1 signal in the one-shot pulse output mode is different from INTTTEQCm1 signals in other mode; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

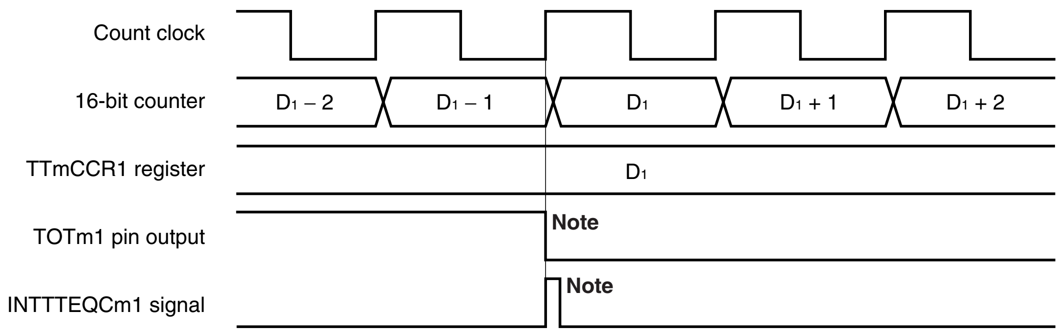
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Usually, the INTTTEQCm1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TTmCCR1 register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOTm1 pin.

(c) Generation timing of compare match interrupt request signal (INTTTEQCm1)

The timing of generation of the INTTTEQCm1 signal in the PWM output mode differs from the timing of INTTTEQCm1 signals in other modes; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Usually, the INTTTEQCm1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TTmCCR1 register.

In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOTm1 pin.

Figure 8-41. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMTm option register 0 (TTmOPT0)

			TTmCCS1	TTmCCS0				TTmOVF
TTmOPT0	0	0	0	0	0	0	0	0/1

TTmOVF: Overflow flag

(f) TMTm counter read buffer register (TTmCNT)

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

(g) TMTm capture/compare registers 0 and 1 (TTmCCR0 and TTmCCR1)

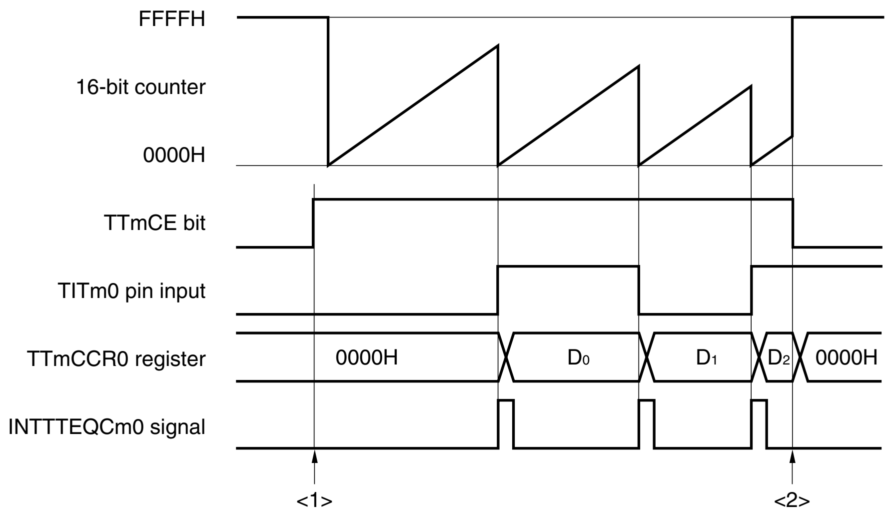
These registers store the count value of the 16-bit counter when the valid edge input to the TITm0 and TITm1 pins is detected.

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 0 (TTmIOC0), TMTm I/O control register 3 (TTmIOC3), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the pulse width measurement mode.

2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(1) Operation flow in pulse width measurement mode

Figure 8-42. Software Processing Flow in Pulse Width Measurement Mode



<1> Count operation start flow

START

Register initial setting
TTmCTL0 register
(TTmCKS0 to TTmCKS2 bits),
TTmCTL1 register,
TTmIOC1 register,
TTmIOC2 register,
TTmOPT0 register

Initial setting of these registers is performed before setting the TTmCE bit to 1.

TTmCE bit = 1

The TTmCKS0 to TTmCKS2 bits can be set at the same time as when counting starts (TTmCE bit = 1).

<2> Count operation stop flow

TTmCE bit = 0

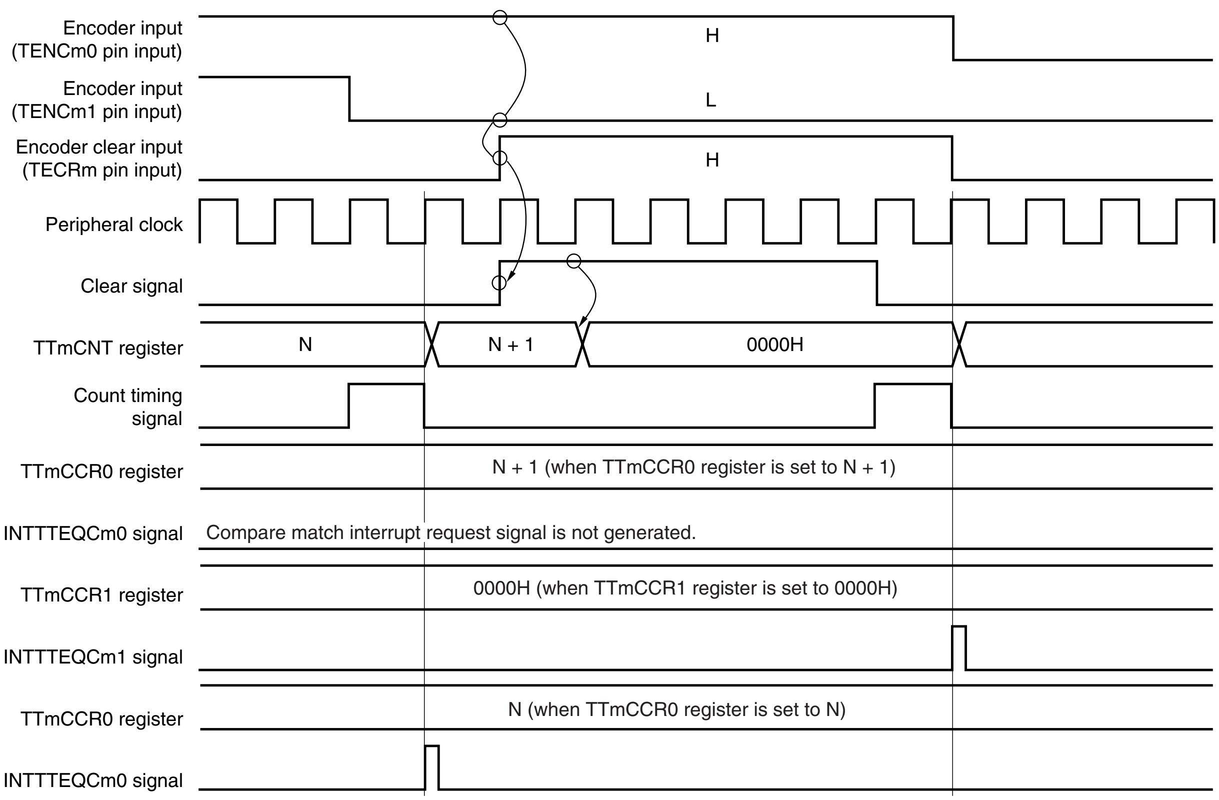
The counter is initialized and counting is stopped by clearing the TTmCE bit to 0.

STOP

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

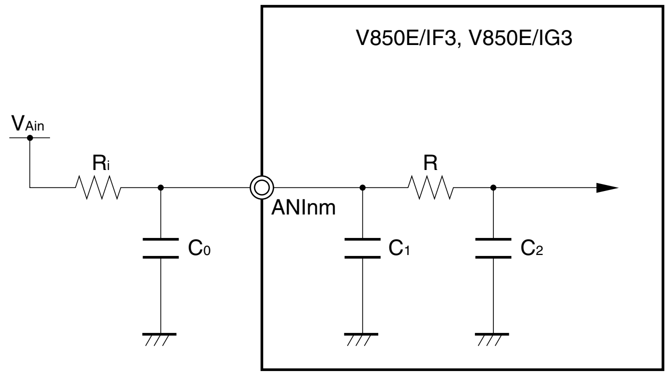
Figure 8-56. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (1/3)

(i) If inputting the high level to the TECRm pin lags behind inputting the low level to the TENCm1 pin while the counter is counting up, the counter is cleared after it counts up.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

An example of calculating an overall error of A/D converters 0 and 1 is shown below.



<R>

f_{XX} (MHz)	A/D conversion time (μs)	Sampling (μs)	R (kΩ)	C_1 (pF)	C_2 (pF)	C_0 (pF)	R_i (kΩ)	Sampling error (LSB)[Note]
64	2.00	0.78	5.1	15	3.9	100	1.0	364.8
	(32/f_{AD01})	(12.5/f_{AD01})				100	0.5	30.4
						100	0.25	0.1 or lower
						100	0.125	0.1 or lower
						50	1.0	62.4
						50	0.5	0.8
						50	0.25	0.1 or lower
						50	0.125	0.1 or lower

Note The error when considering the signal source impedance is “sampling error + overall error”.

Remarks 1. These values are reference values calculated by simulating what happens to C_2 voltage by R_i and C_0 when V_{Ain} is applied from 0 V to 5 V at the same time as sampling start.

2. m = 0 to 5 when n = 0
m = 0 to 7 when n = 1

3. f_{XX}: System clock frequency
f_{AD01}: Basic clock frequency

14.1 Mode Switching Between UARTA and Other Serial Interface

14.1.1 Mode switching between UARTA0 and CSIB0

In the V850E/IF3 and V850E/IG3, UARTA0 and CSIB0 function alternately, and these pins cannot be used at the same time. To switch between UARTA0 and CSIB0, the PMC4, PFC4, and PFCE4 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA0 or CSIB0 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-1. Mode Switch Settings of UARTA0 and CSIB0

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	PFC47	PFC46	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	PFCE40

PMC42	PFCE42	PFC42	Specification of alternate function of P42 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB0}}$ I/O
1	0	1	INTP13 input
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC41	PFCE41	PFC41	Specification of alternate function of P41 pin
0	×	×	I/O port
1	0	0	SOB0 output
1	0	1	TXDA0 output
1	1	0	Setting prohibited
1	1	1	Setting prohibited

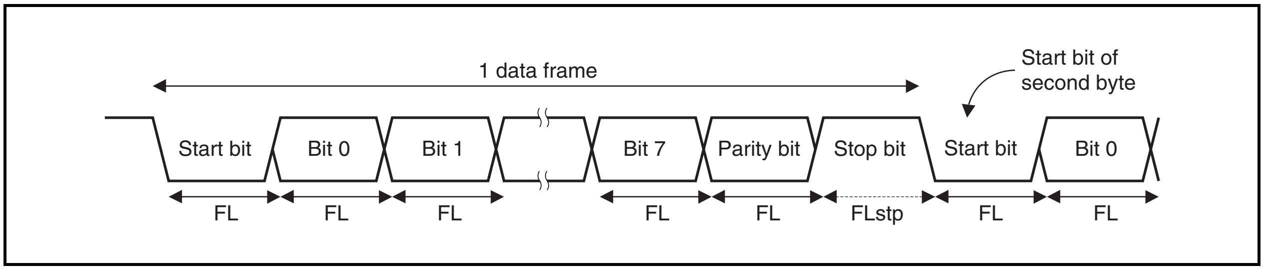
PMC40	PFC40	Specification of alternate function of P40 pin
0	×	I/O port
1	0	SIB0 input
1	1	RXDA0 input

Remark × = don't care

(5) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from a stop bit to the next start bit is extended two clocks longer than normal. However, on the reception side, the transfer result is not affected since the timing is initialized by the detection of the start bit.

Figure 15-11. Transfer Rate During Continuous Transmission



Representing the 1-bit data length by FL, the stop bit length by FLstp, and the base clock frequency by f_{XX} yields the following equation.

$$FLstp = FL + 2/(f_{XX})$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/(f_{XX})$$

(a) How to use CBnSCE bit

(i) In single reception mode

<1> When the reception of the last data is ended with INTCBnR interrupt servicing, clear the CBnSCE bit to 0, and then read the CBnRX register.

<2> When the reception is disabled after the reception of the last data has been ended, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

(ii) In continuous reception mode

<1> Clear the CBnSCE bit to 0 during reception of the last data with INTCBnR interrupt servicing by the reception before the last reception, and then read the CBnRX register.

<2> After receiving the INTCBnR signal of the last reception, read the last data from the CBnRX register.

<3> When the reception is disabled after the reception of the last data has been ended, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

Caution In continuous reception mode, the serial clock is not stopped until the reception executed when the CBnSCE bit is cleared to 0 is ended after the reception is started by a dummy read.

(4/4)

<table>
<tr><th>SPT0</th><th colspan="2">Stop condition trigger</th></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device's transfer).
After the SDA line goes to low level, either set the SCL line to high level or wait until the SCL pin goes to high level. Next, after the rated amount of time has elapsed, the SDA line is changed from low level to high level and a stop condition is generated.</td></tr>
<tr><td colspan="3">Cautions concerning setting timing
For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and during the wait period after slave has been notified of final reception.
For master transmission: A stop condition may not be generated normally during the $\overline{\mathrm{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.
• Cannot be set to 1 at the same time as the STT0 bit.
• The SPT0 bit can be set to 1 only when in master mode[Note].
• When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock.
The WTIM0 bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows output of the ninth clock.
• When the SPT0 bit is set to 1, setting the SPT0 bit to 1 again is disabled until the setting is cleared to 0.</td></tr>
<tr><td colspan="2">Condition for clearing (SPT0 bit = 0)</td><td>Condition for setting (SPT0 bit = 1)</td></tr>
<tr><td colspan="2">• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• When the LREL0 bit = 1 (exit from communications)
• When the IICE0 bit = 0 (operation stop)
• Reset</td><td>• Set by instruction</td></tr>
</table>

Note Set the SPT0 bit to 1 only in master mode. However, the SPT0 bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to operation enable status. For details, see **17.15 Cautions**.

Caution When the IICS0.TRC0 bit is set to 1, the WREL0 bit is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared to 0 and the SDA line is set to high impedance.

Remark The SPT0 bit is 0 if it is read after data setting.

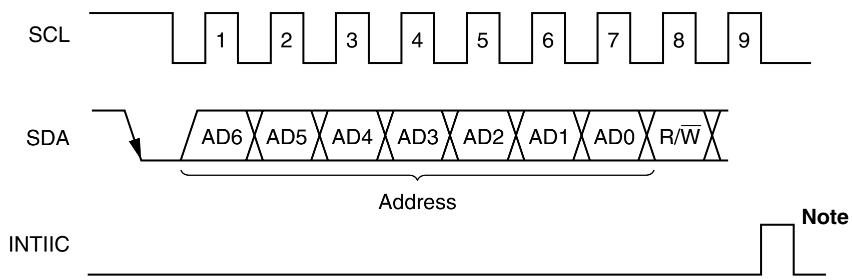
17.6.2 Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the SVA0 register. If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 17-7. Address



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

The slave address and the eighth bit, which specifies the transfer direction as described in **17.6.3 Transfer direction specification** below, are together written to the IIC0 register and are then output. Received addresses are written to the IIC0 register.

The slave address is assigned to the higher 7 bits of the IIC0 register.

17.7.5 Arbitration loss operation (operation as slave after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICC0.WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
				▲1	▲2		▲3	Δ4

▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
				▲1		▲2		▲3 Δ4

▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.12 Arbitration

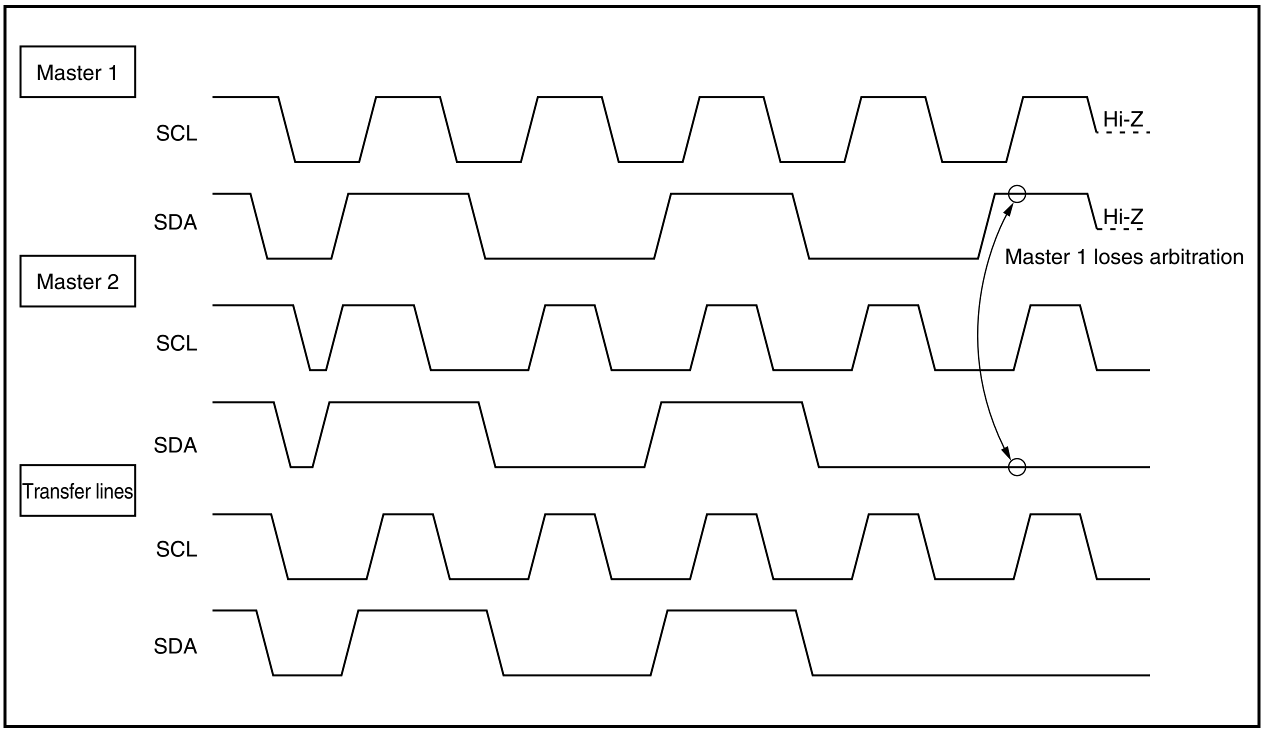
When several master devices simultaneously generate a start condition (when the IICC0.STT0 bit is set to 1 before the IICS0.STD0 bit is set to 1), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICS0.ALD0 bit) is set (1) via the timing by which the arbitration loss occurred, and the SCL and SDA lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request signal (INTIIC) (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 bit = 1 setting that has been made by software.

For details of interrupt request timing, see **17.7 I²C Interrupt Request Signals (INTIIC)**.

Figure 17-12. Arbitration Timing Example



(3/13)

Symbol	Name	Unit	Page
AD1CR4H	A/D1 conversion result register 4H	ADC1	628
AD1CR5	A/D1 conversion result register 5	ADC1	628
AD1CR5H	A/D1 conversion result register 5H	ADC1	628
AD1CR6	A/D1 conversion result register 6	ADC1	628
AD1CR6H	A/D1 conversion result register 6H	ADC1	628
AD1CR7	A/D1 conversion result register 7	ADC1	628
AD1CR7H	A/D1 conversion result register 7H	ADC1	628
AD1CR8	A/D1 conversion result register 8	ADC1	628
AD1CR8H	A/D1 conversion result register 8H	ADC1	628
AD1CR9	A/D1 conversion result register 9	ADC1	628
AD1CR9H	A/D1 conversion result register 9H	ADC1	628
AD1CTC	A/D converter 1 conversion time control register	ADC1	625
AD1CTL0	A/D converter 1 control register	ADC1	632
AD1ECR0	A/D1 conversion result extension register 0	ADC1	638
AD1ECR0H	A/D1 conversion result extension register 0H	ADC1	638
AD1ECR1	A/D1 conversion result extension register 1	ADC1	638
AD1ECR1H	A/D1 conversion result extension register 1H	ADC1	638
AD1ECR2	A/D1 conversion result extension register 2	ADC1	638
AD1ECR2H	A/D1 conversion result extension register 2H	ADC1	638
AD1ECR3	A/D1 conversion result extension register 3	ADC1	638
AD1ECR3H	A/D1 conversion result extension register 3H	ADC1	638
AD1ECR4	A/D1 conversion result extension register 4	ADC1	638
AD1ECR4H	A/D1 conversion result extension register 4H	ADC1	638
AD1FLG	A/D converter 1 flag register	ADC1	640
AD1FLGB	A/D converter 1 flag buffer register	ADC1	641
AD1IC	Interrupt control register	INTC	1000
AD1OCKS	A/D converter 1 clock select register	ADC1	643
AD1SCM	A/D converter 1 scan mode register	ADC1	623
AD1SCMH	A/D converter 1 scan mode register H	ADC1	623
AD1SCML	A/D converter 1 scan mode register L	ADC1	623
AD1TSEL	A/D converter 1 trigger select register	ADC1	633
AD2CR0	A/D2 conversion result register 0	ADC2	695
AD2CR0H	A/D2 conversion result register 0H	ADC2	695
AD2CR1	A/D2 conversion result register 1	ADC2	695
AD2CR1H	A/D2 conversion result register 1H	ADC2	695
AD2CR2	A/D2 conversion result register 2	ADC2	695
AD2CR2H	A/D2 conversion result register 2H	ADC2	695
AD2CR3	A/D2 conversion result register 3	ADC2	695
AD2CR3H	A/D2 conversion result register 3H	ADC2	695
AD2CR4	A/D2 conversion result register 4	ADC2	695
AD2CR4H	A/D2 conversion result register 4H	ADC2	695
AD2CR5	A/D2 conversion result register 5	ADC2	695
AD2CR5H	A/D2 conversion result register 5H	ADC2	695
AD2CR6	A/D2 conversion result register 6	ADC2	695