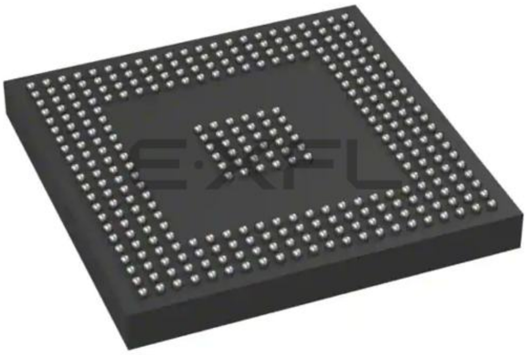
E·XFL Renesas Electronics America Inc - R7S721001VCBG#AC0 Datasheet



Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

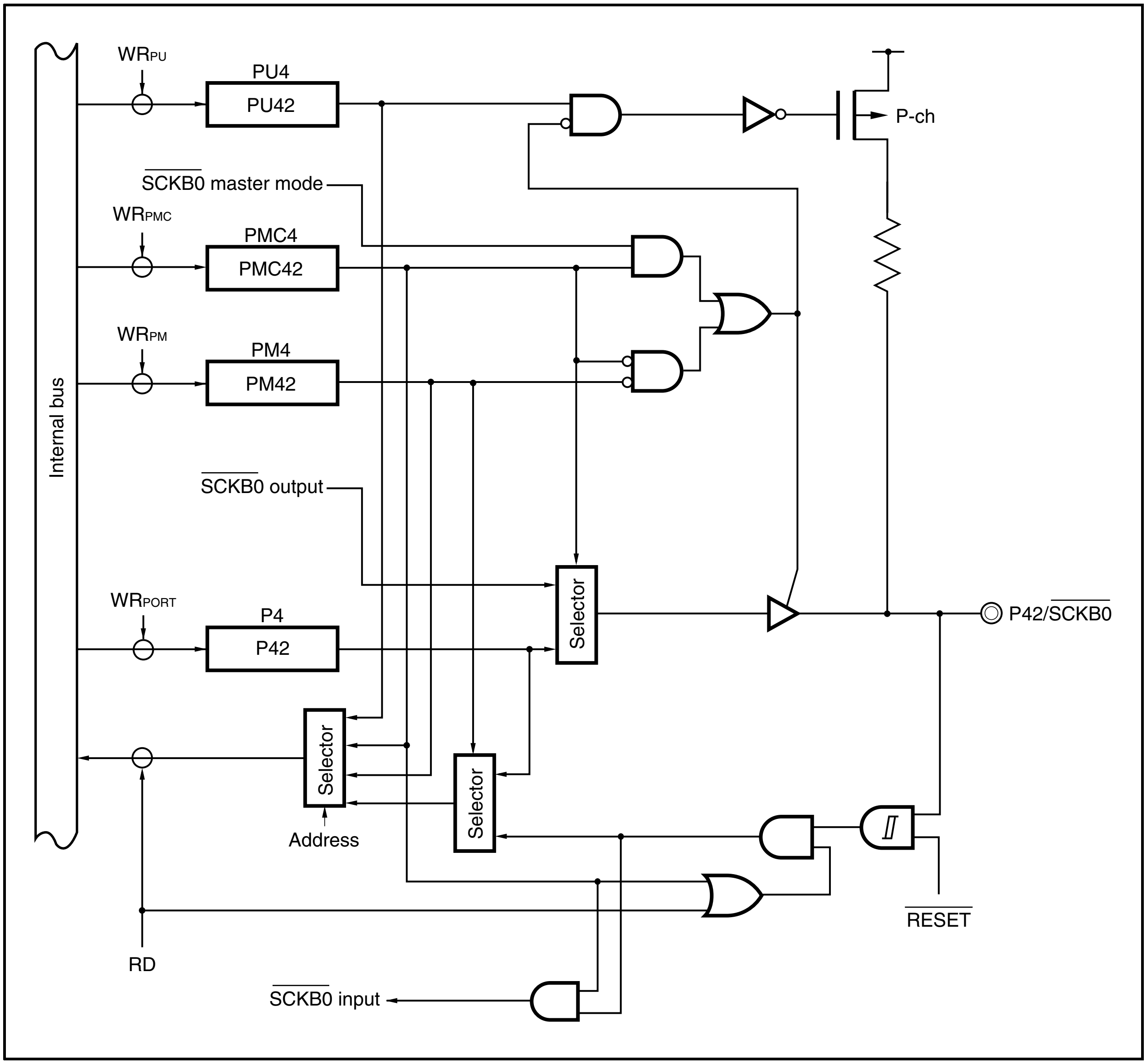
Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Active
Core Processor	ARM® Cortex®-A9
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	400MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	SDRAM, SRAM
Graphics Acceleration	Yes
Display & Interface Controllers	DVD, VDC
Ethernet	10/100Mbps (1), 100Mbps (1)
SATA	-
USB	USB 2.0 (2)
Voltage - I/O	1.2V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	324-FBGA
Supplier Device Package	324-FBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/r7s721001vcbg-ac0

(3/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF310H	External interrupt noise elimination control register	INTPNRC	R/W	√	√		00H
FFFFF400H	Port 0 register	P0		√	√		Undefined
FFFFF402H	Port 1 register	P1		√	√		Undefined
FFFFF404H	Port 2 register	P2		√	√		Undefined
FFFFF406H	Port 3 register	P3		√	√		Undefined
FFFFF408H	Port 4 register	P4		√	√		Undefined
FFFFF420H	Port 0 mode register	PM0		√	√		FFH
FFFFF422H	Port 1 mode register	PM1		√	√		FFH
FFFFF424H	Port 2 mode register	PM2		√	√		FFH
FFFFF426H	Port 3 mode register	PM3		√	√		FFH
FFFFF428H	Port 4 mode register	PM4		√	√		FFH
FFFFF440H	Port 0 mode control register	PMC0		√	√		00H
FFFFF442H	Port 1 mode control register	PMC1		√	√		00H
FFFFF444H	Port 2 mode control register	PMC2		√	√		00H
FFFFF446H	Port 3 mode control register	PMC3		√	√		00H
FFFFF448H	Port 4 mode control register	PMC4		√	√		00H
FFFFF462H	Port 1 function control register	PFC1		√	√		00H
FFFFF466H	Port 3 function control register	PFC3		√	√		00H
FFFFF468H	Port 4 function control register	PFC4		√	√		00H
FFFFF540H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF544H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF5C0H	TMQ0 control register 0	TQ0CTL0		√	√		00H
FFFFF5C1H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF5C2H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF5C3H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H
FFFFF5C4H	TMQ0 I/O control register 2	TQ0IOC2		√	√		00H
FFFFF5C5H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF5C6H	TMQ0 capture/compare register 0	TQ0CCR0				√	0000H
FFFFF5C8H	TMQ0 capture/compare register 1	TQ0CCR1				√	0000H
FFFFF5CAH	TMQ0 capture/compare register 2	TQ0CCR2				√	0000H
FFFFF5CCH	TMQ0 capture/compare register 3	TQ0CCR3				√	0000H
FFFFF5CEH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF5F0H	High-impedance output control register 00	HZA0CTL0	R/W	√	√		00H
FFFFF5F1H	High-impedance output control register 01	HZA0CTL1		√	√		00H
FFFFF600H	TMQ1 control register 0	TQ1CTL0		√	√		00H
FFFFF601H	TMQ1 control register 1	TQ1CTL1		√	√		00H
FFFFF602H	TMQ1 I/O control register 0	TQ1IOC0		√	√		00H
FFFFF605H	TMQ1 option register 0	TQ1OPT0		√	√		00H
FFFFF606H	TMQ1 capture/compare register 0	TQ1CCR0				√	0000H
FFFFF608H	TMQ1 capture/compare register 1	TQ1CCR1				√	0000H
FFFFF60AH	TMQ1 capture/compare register 2	TQ1CCR2				√	0000H

Figure 4-19. Block Diagram of P42 Pin



5.4 PLL Function

5.4.1 Overview

The CPU and the operating clock of the peripheral macro can be switched between output of the oscillation frequency multiplied by 8, and clock-through mode.

When PLL function is used: Input clock (f_X) = 2.5 MHz, output clock (f_{XX}) = 20 MHz
Clock-through mode: Input clock (f_X) = 2.5 MHz, output clock (f_{XX}) = 2.5 MHz

5.4.2 PLL mode

In the PLL mode, the oscillation frequency (f_X) is multiplied by 8 with the PLL to generate a system clock (f_{XX}).

In the PLL mode, the clock is input from the oscillator to the PLL. A clock at a stable frequency must be supplied to the internal circuit after the lapse of the lockup time (frequency stabilization time) during which the phase is locked at a specific frequency and oscillation is stabilized. In the V850ES/IE2, the lockup time after release of reset is secured automatically.

5.4.3 Clock-through mode

In the clock-through mode, a system clock (f_{XX}) of the same frequency as the oscillation frequency (f_X) is generated.

(7) TMPn capture/compare register 0 (TPnCCR0)

The TP0CCR0 and TP2CCR0 registers are 16-bit registers that can be used as capture registers or compare registers depending on the mode. The TP1CCR0 and TP3CCR0 registers are 16-bit registers that can only be used as compare registers.

The TP0CCR0 and TP2CCR0 registers can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TP0OPT0.TP0CCS0 or TP2OPT0.TP2CCS0 bit. In the pulse width measurement mode, the TPnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR0 register can be read or written during operation.

This register can be read or written in 16-bit units.

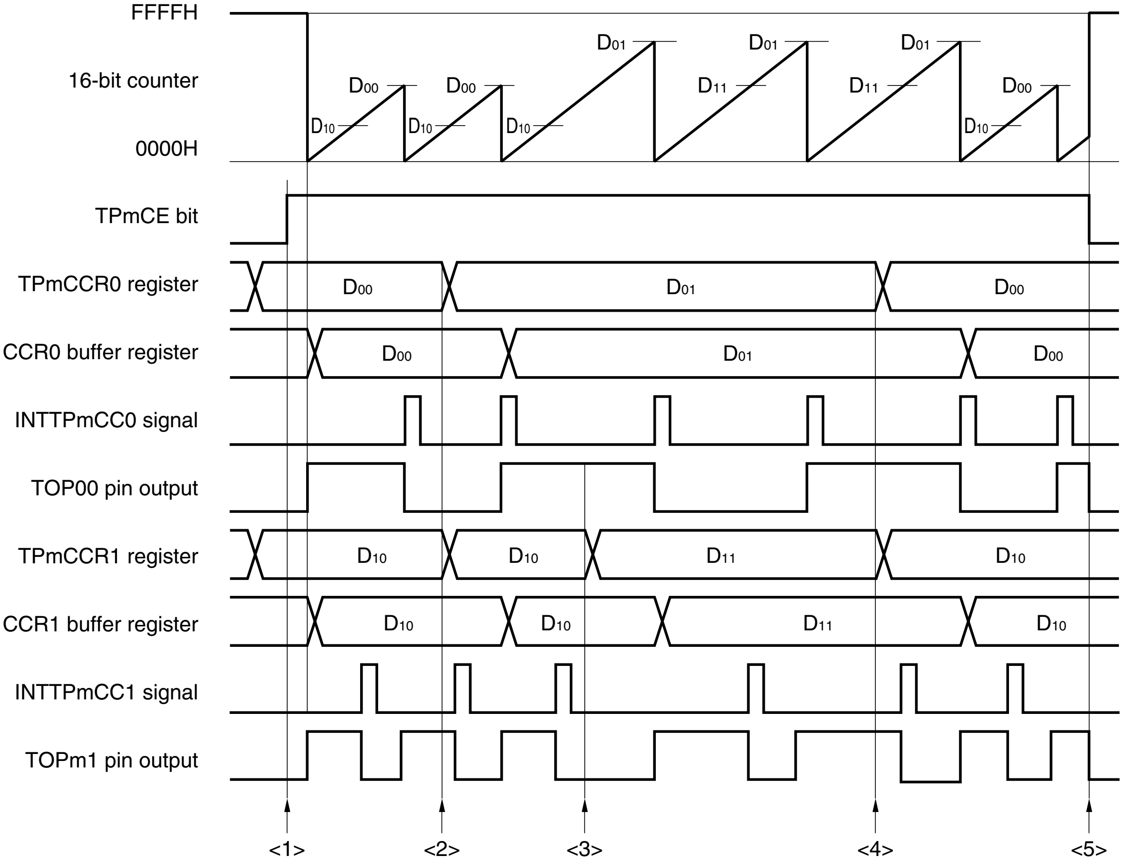
Reset sets this register to 0000H.

After reset: 0000H R/W Address: TP0CCR0 FFFFF646H, TP1CCR0 FFFFF666H,
TP2CCR0 FFFFF686H, TP3CCR0 FFFFF6A6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0 (n = 0 to 3)																

(1) Operation flow in PWM output mode

Figure 6-34. Software Processing Flow in PWM Output Mode (1/2)



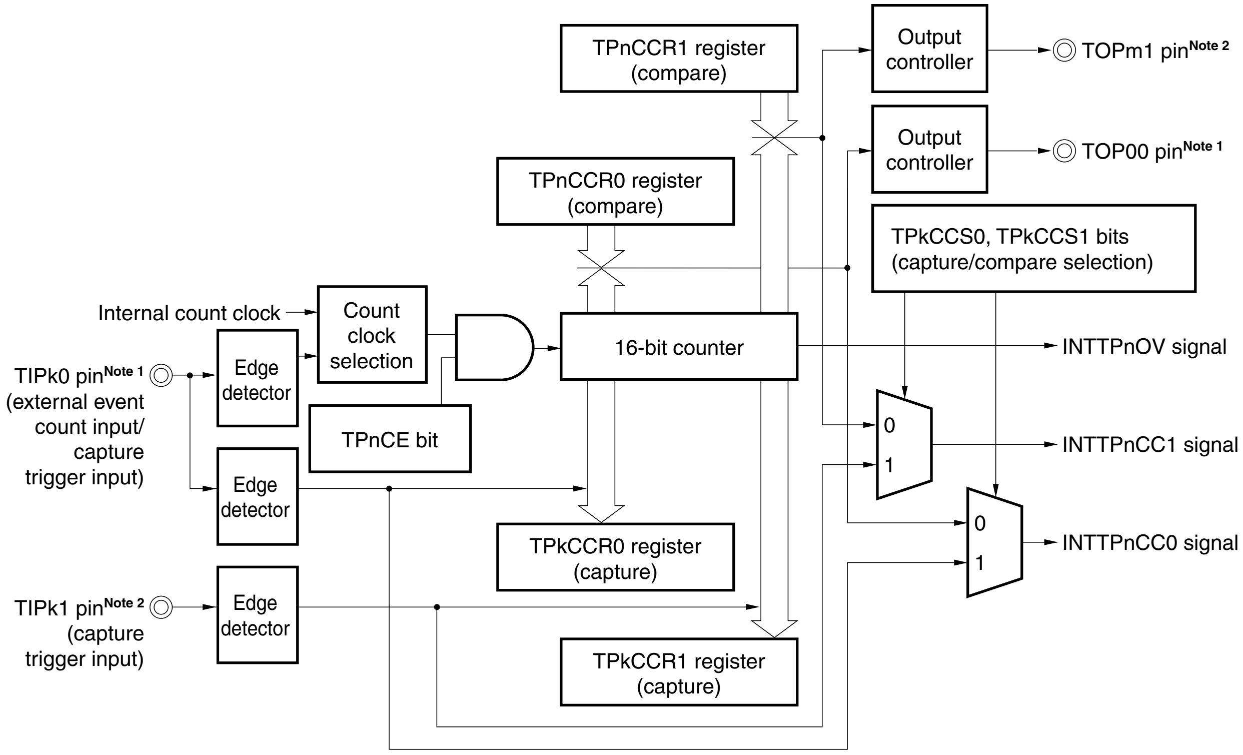
Remark m = 0, 2, 3
k = 0, 2

6.6.6 Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)

The compare function is valid for all of TMP0 to TMP3. The capture function is valid only for TMP0 and TMP2.

In the free-running timer mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. At this time, the TPkCCR0 and TPkCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TPkOPT0.TPkCCS0 and TPkOPT0.TPkCCS1 bits.

Figure 6-35. Configuration in Free-Running Timer Mode



Notes 1. Because the external event count input pin (TIPk0), capture trigger input pin (TIPk0), and timer output pin (TOP00) share the same alternate-function pin, two or more functions cannot be used at the same time.

2. Because the capture trigger input pin (TIPk1) and timer output pin (TOPk1) share the same alternate-function pin, two functions cannot be used at the same time.

Remark n = 0 to 3
m = 0, 2, 3
k = 0, 2

(a) Function as compare register

The TQnCCR0 register can be rewritten even when the TQnCTL0.TQnCE bit = 1.

The set value of the TQnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTQnCC0) is generated. If TOQn0 pin output is enabled at this time, the output of the TOQn0 pin is inverted.

When the TQnCCR0 register is used as a cycle register in the interval timer mode, external event count mode[Note], external trigger pulse output mode[Note], one-shot pulse output mode[Note], or PWM output mode[Note], the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by the TQnCTL0.TQnCE bit = 0.

Note These modes can be set only in TMQ0. They cannot be set in TMQ1.

(b) Function as capture register (TQ0CCR0 register only)

When the TQ0CCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TQ0CCR0 register if the valid edge of the capture trigger input pin (TIQ00 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ00 pin) is detected.

Even if the capture operation and reading the TQ0CCR0 register conflict, the correct value of the TQ0CCR0 register can be read.

The capture register is cleared by the TQ0CTL0.TQ0CE bit = 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Notes 1, 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Notes 1, 2]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMQ0 only

2. This mode can be set only with the software trigger. No external trigger input pin is available.

3. Writing to the TQ0CCR1 register is the trigger.

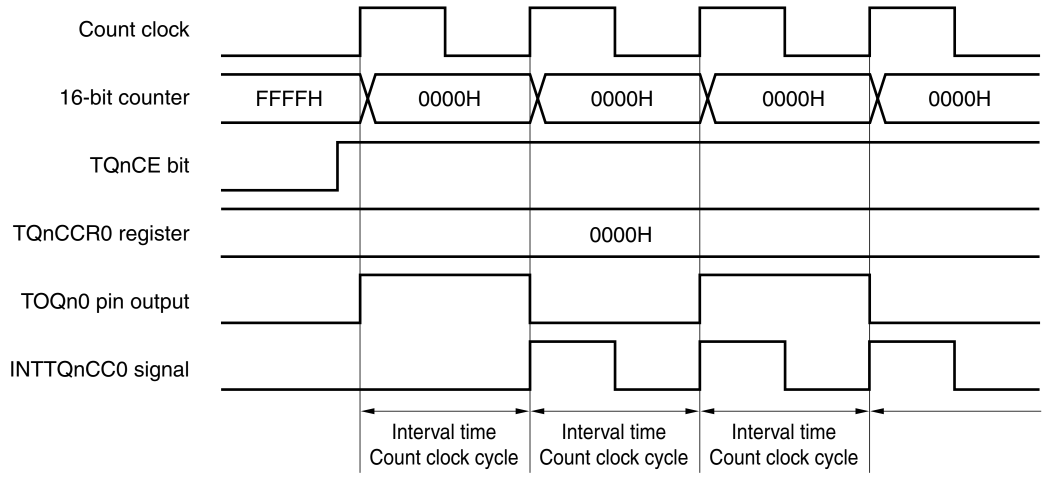
Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(2) Interval timer mode operation timing

(a) Operation if TQnCCR0 register is set to 0000H

If the TQnCCR0 register is set to 0000H, the INTTQnCC0 signal is generated at each count clock, and the output of the TOQn0 pin is inverted.

The value of the 16-bit counter is always 0000H.



Remark n = 0, 1

When the TQ0CE bit is set to 1, 16-bit timer/event counter Q waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOQ0b pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

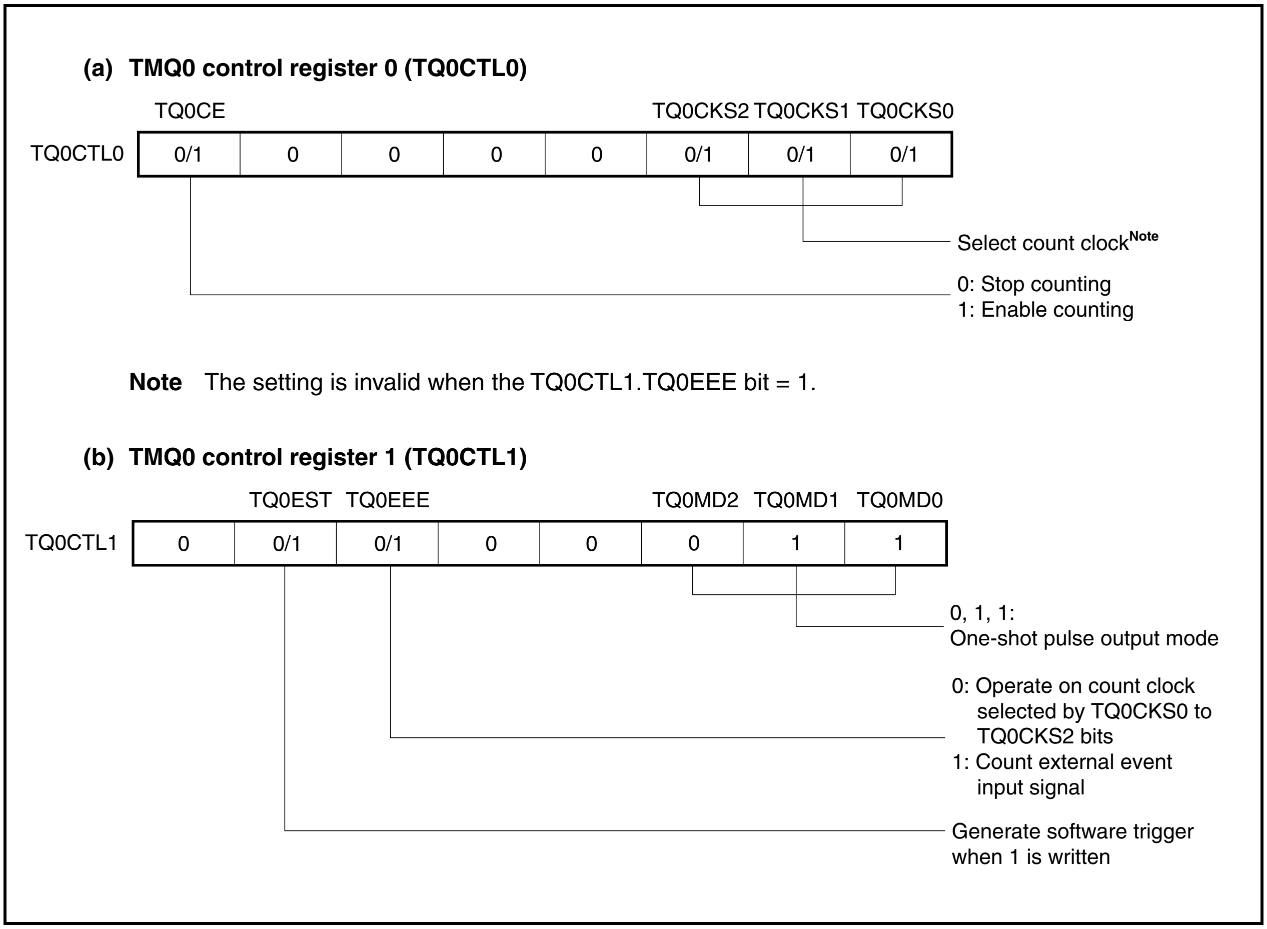
Output delay period = (Set value of TQ0CCRb register) × Count clock cycle
Active level width = (Set value of TQ0CCR0 register − Set value of TQ0CCRb register + 1) × Count clock cycle

The compare match interrupt request signal INTTQ0CC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTQ0CCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

Only setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark b = 1 to 3

Figure 7-28. Register Setting in One-Shot Pulse Output Mode (1/3)



(a) TMQ0 control register 0 (TQ0CTL0)

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0/1	0/1	0/1

TQ0CKS2 to TQ0CKS0: Select count clock[Note]

TQ0CE: 0: Stop counting
1: Enable counting

Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0/1	0/1	0	0	0	1	1

TQ0MD2 to TQ0MD0: 0, 1, 1:
One-shot pulse output mode

TQ0EEE: 0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
1: Count external event input signal

TQ0EST: Generate software trigger when 1 is written

(1) TMQ1 dead-time compare register (TQ1DTC)

The TQ1DTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TQ1CTL0.TQ1CE bit = 1.

This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

<R> **Caution To generate the dead-time period, set the TQ1DTC register to 1 or more.**

When the operation is stopped (TQ1CTL0.TQ1CE bit = 0), the dead-time period is not generated and the output level of the TOQ1T1 to TOQ1T3 pins and TOQ1B1 to TOQ1B3 pins will be in the initial status. For the system protection, therefore, before operation is being stopped, set the TOQ1T1 to TOQ1T3 and TOQ1B1 to TOQ1B3 pins to the high impedance state, or set the output level of pins and switch them to the port mode.

If a dead time period is not needed, set the TQ1DTC register to 0.

After reset: 0000H R/W Address: FFFFF624H

	15 10	9 0
TQ1DTC	000000	TQ1DTC9 to TQ1DTC0

(2) Dead-time counters 1 to 3

The dead-time counters are 10-bit counters that count dead time.

These counters are cleared or count up at the rising or falling edge of the TOQ1m output signal by TMQ1, and are cleared or stopped when their count value matches the value of the TQ1DTC register. The count clock of these counters is the same as that set by the TQ1CTL0.TQ1CKS2 to TQ1CTL0.TQ1CKS0 bits of TMQ1.

Remarks 1. The operation differs when the TQ1OPT2.TQ1DTM bit = 1. For details, see **9.4.2 (4) Automatic dead-time width narrowing function (TQ1OPT2.TQ1DTM bit = 1)**.

2. m = 1 to 3

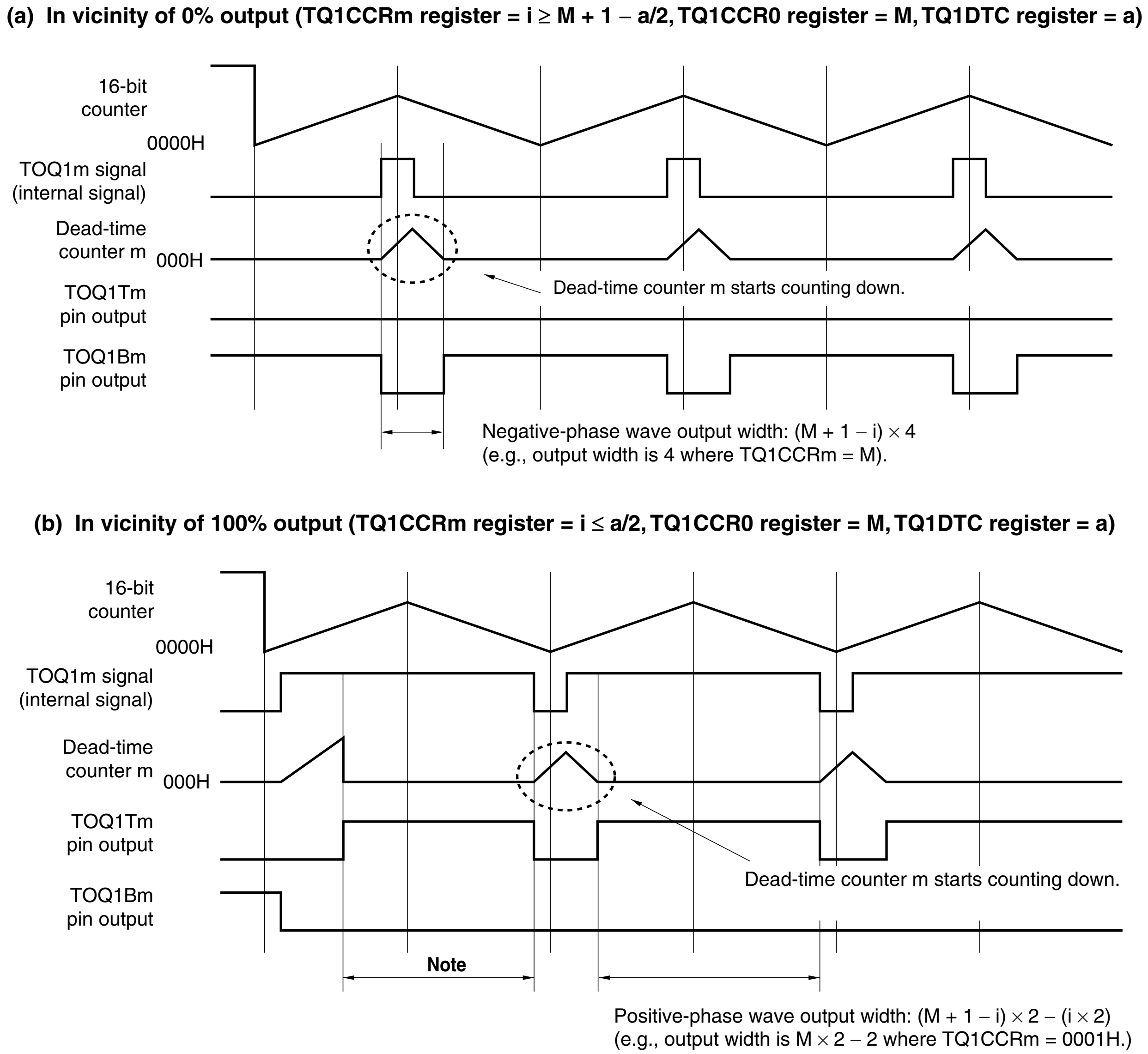
(4) Automatic dead-time width narrowing function (TQ1OPT2.TQ1DTM bit = 1)

The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TQ1OPT2.TQ1DTM bit to 1.

By setting the TQ1DTM bit to 1, the dead-time counter is not cleared, but starts counting down if the TOQ1m (internal signal) output of timer Q changes during dead-time counting.

The following timing chart shows the operation of the dead-time counter when the TQ1DTM bit is set to 1.

Figure 9-13. Operation of Dead-Time Counter m (1)



(a) In vicinity of 0% output (TQ1CCRm register = i ≥ M + 1 − a/2, TQ1CCR0 register = M, TQ1DTC register = a)

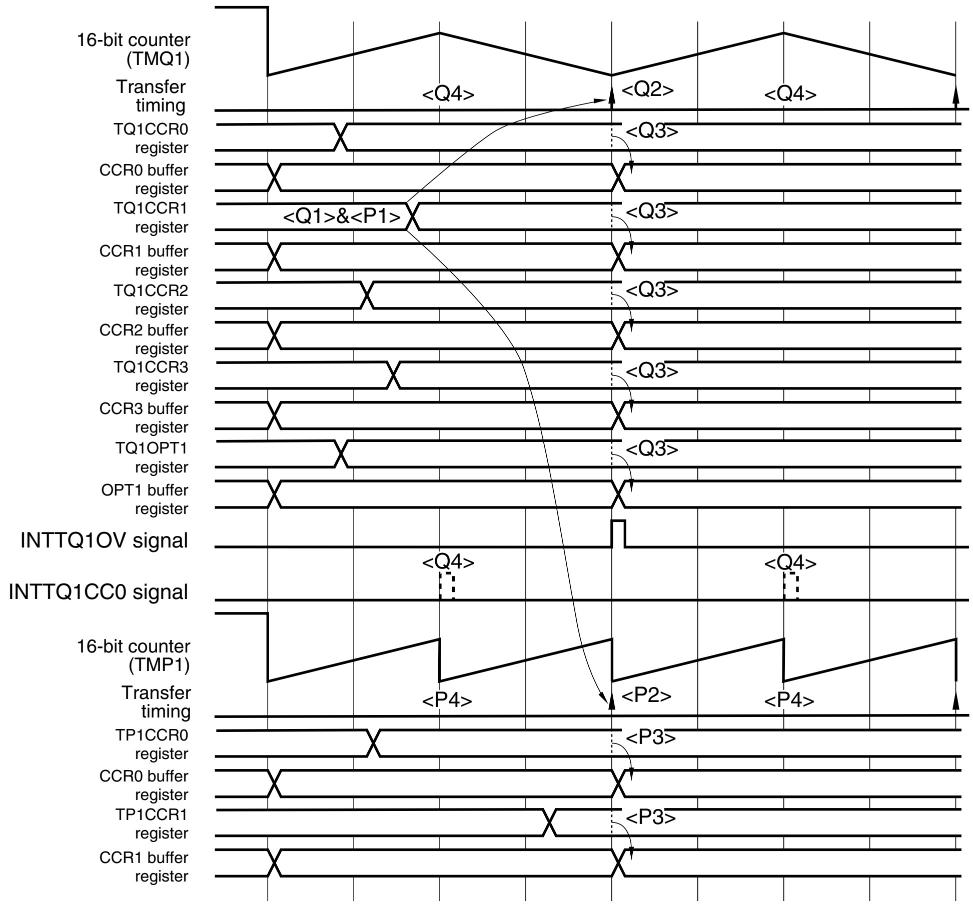
(b) In vicinity of 100% output (TQ1CCRm register = i ≤ a/2, TQ1CCR0 register = M, TQ1DTC register = a)

16-bit counter
0000H
TOQ1m signal (internal signal)
Dead-time counter m
000H
Dead-time counter m starts counting down.
TOQ1Tm pin output
TOQ1Bm pin output
Negative-phase wave output width: (M + 1 − i) × 4
(e.g., output width is 4 where TQ1CCRm = M).
16-bit counter
0000H
TOQ1m signal (internal signal)
Dead-time counter m
000H
TOQ1Tm pin output
Dead-time counter m starts counting down.
TOQ1Bm pin output
Note
Positive-phase wave output width: (M + 1 − i) × 2 − (i × 2)
(e.g., output width is M × 2 − 2 where TQ1CCRm = 0001H.)

Note The output width of the first wave differs from that of the second and subsequent waves immediately after the TQ1CTL0.TQ1CE bit has been set. The first wave is shorter than the second wave because the dead time is fully counted.

Remark m = 1 to 3

Figure 9-31. Basic Operation in Intermittent Batch Rewrite Mode



[TMQ1 operation]

<Q1> Write the TQ1CCR1 register.

<Q2> Rewrite the register at the transfer timing that is generated after the TQ1CCR1 register has been rewritten.

<Q3> The registers are transferred all at once at the transfer timing.

<Q4> The transfer timing is also culled as the interrupts are culled.

[TMP1 operation]

<P1> Write the TQ1CCR1 register.

<P2> Rewrite the register at the transfer timing that is generated after the TQ1CCR1 register has been rewritten.

<P3> The registers are transferred all at once at the transfer timing.

<P4> The transfer timing is also culled as the interrupts are culled.

Remark This is an example of the operation when the TQ1OPT1.TQ1ICE bit = 1, TQ1OPT1.TQ1IOE bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits = 00001.

(2) Operation mode

There are four operation modes in which the ANIn0 to ANIn3 pins are set: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode. The continuous select mode and one-shot select mode have sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the ADAnM0 and ADAnM2 registers.

Remark n = 0, 1

(a) Continuous select mode

In this mode, the analog input pin (ANInm) specified by the ADAnS register is A/D converted continuously. The conversion results are stored in the A/Dn conversion result register (ADAnCRm) corresponding to the ANInm pin. In this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results.

- **1-buffer mode**

  In this mode, the voltage of the analog input pin (ANInm) specified by the ADAnS register is A/D converted. The conversion results are stored in the ADAnCRm register corresponding to the ANInm pin. The ANInm pin and the ADAnCRm register correspond one to one, and an A/Dn conversion end interrupt request signal (INTADn) is generated each time one A/D conversion ends.
  After the end of A/D conversion, the conversion is repeated again unless the ADAnM0.ADAnCE bit is cleared to 0.

  Remark n = 0, 1
  m= 0 to 3

11.9.8 Restrictions on setting one-shot mode and software trigger mode

If the A/D converters 0 and 1 are set in the one-shot select mode and software trigger mode (ADAnM0 register = 1010XX0XB) or one-shot scan mode and software trigger mode (ADAnM0 register = 1011XX0XB), a re-conversion operation should be performed in a new condition when data is written to any of the ADAnM0, ADAnM2, and ADAnS registers upon completion of an A/D conversion operation. However, the re-conversion operation is not performed but the conversion operation is enabled (ADAnM0.ADAnCE bit = 1) and stopped (ADAnM0.ADAnEF bit = 0). The A/Dn conversion end interrupt request signal (INTADn) is not generated, nor is the last A/D conversion result stored. However, the data is correctly written to any of the ADAnM0, ADAnM2, and ADAnS registers.

If this happens, normal operation can be restored by setting the ADAnM0.ADAnCE bit to 1.

For example, if the ANIn0 and ANIn1 pins are set in the scan mode (ADAnS register = 00000001B) and data is written to the ADAnM0 register upon completion of an A/D conversion operation in the one-shot scan mode and software trigger mode (ADAnM0 register = 1011XX0XB), the signal of the ANIn0 pin is correctly converted and the conversion result is correctly stored in the ADAnCR0 register. However, the result of converting the signal of the ANIn1 pin which has been performed immediately before the completion of the A/D conversion is not stored in the ADAnCR1 register, nor is the INTADn interrupt request signal generated.

[Countermeasure]

The above restriction can be avoided by performing any of steps <1> to <3>, below.

<1> Before writing to any of the ADAnM0, ADAnM2, and ADAnS registers, confirm that A/D conversion is stopped (ADAnM0.ADAnEF bit = 0).

<2> After disabling the interrupt (PSW.ID bit = 1), execute an instruction that writes data to any of the ADAnM0, ADAnM2, and ADAnS registers and an instruction that sets the ADAnM0.ADAnCE bit to 1 consecutively, and then enable the interrupt (PSW.ID bit = 0).
This action is to avoid coincidence between the completion of the A/D conversion operation and writing to the ADAnM0, ADAnM2, or ADAnS register. If, for example, executing a write instruction and the completion of the A/D conversion operation coincide and thus the A/D conversion is stopped, the A/D conversion can be started by setting of the ADAnCE bit to 1. If the ADAnM0.ADAnCE bit = 1, the ADAnCE bit is set to 1 again consecutively.

<3> Disable the A/D conversion operation by clearing the ADAnCE bit to 0, write data to any of the ADAnM0, ADAnM2, and ADAnS registers, enable the A/D conversion operation by setting the ADAnCE bit to 1, and start the A/D conversion.

(a) How to use CB0SCE bit

(i) In single reception mode

<1> When the reception of the last data is completed with INTCB0R interrupt servicing, clear the CB0SCE bit to 0, and then read the CB0RX register.

<2> When the reception is disabled after the reception of the last data has been completed, check that the CB0STR.CB0TSF bit is 0, and then clear the CB0PWR and CB0RXE bits to 0. To continue reception, set the CB0SCE bit to 1 and start the next receive operation by performing a dummy read of the CB0RX register.

(ii) In continuous reception mode

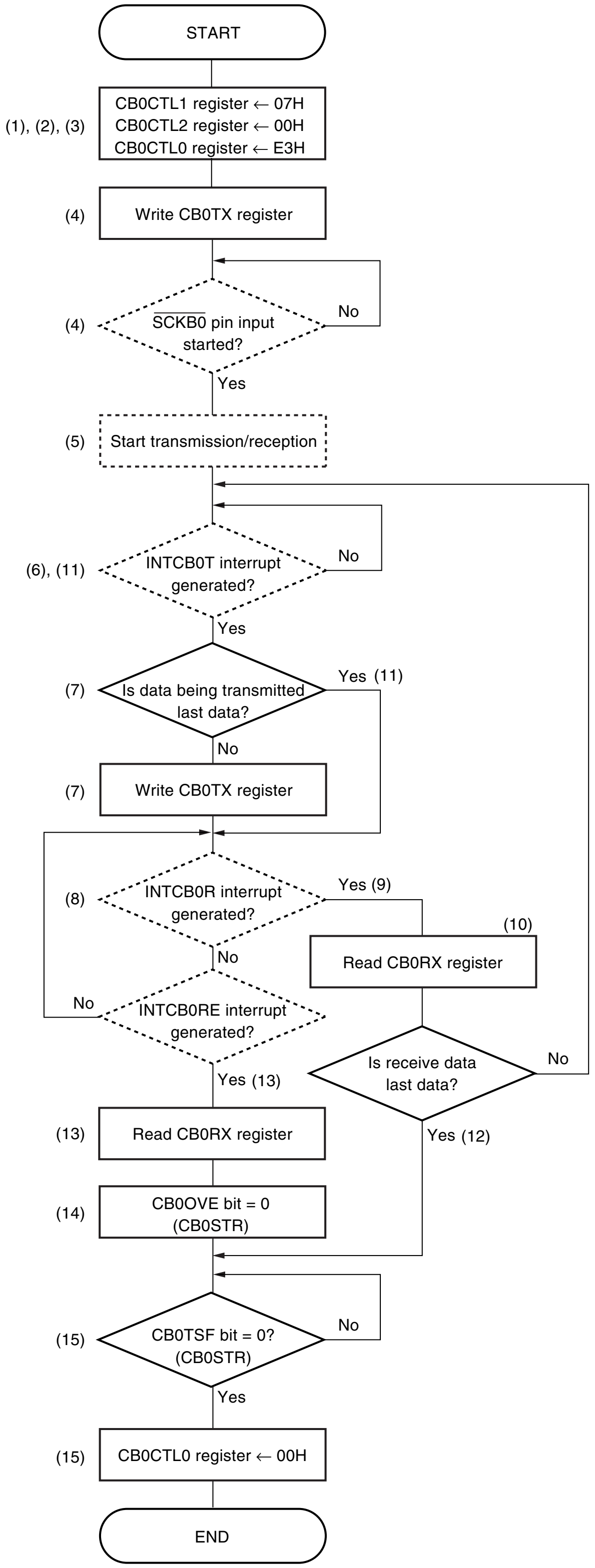
<1> Clear the CB0SCE bit to 0 during reception of the last data with INTCB0R interrupt servicing by the reception before the last reception, and then read the CB0RX register.

<2> After receiving the INTCB0R signal of the last reception, read the last data from the CB0RX register.

<3> When the reception is disabled after the reception of the last data has been completed, check that the CB0STR.CB0TSF bit is 0, and then clear the CB0PWR and CB0RXE bits to 0. To continue reception, set the CB0SCE bit to 1 and start the next receive operation by performing a dummy read of the CB0RX register.

Caution In continuous reception mode, the serial clock is not stopped until the reception executed when the CB0SCE bit is cleared to 0 is completed after the reception is started by a dummy read.

(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

16.3.3 Low-voltage detector (LVI)

(1) Functions

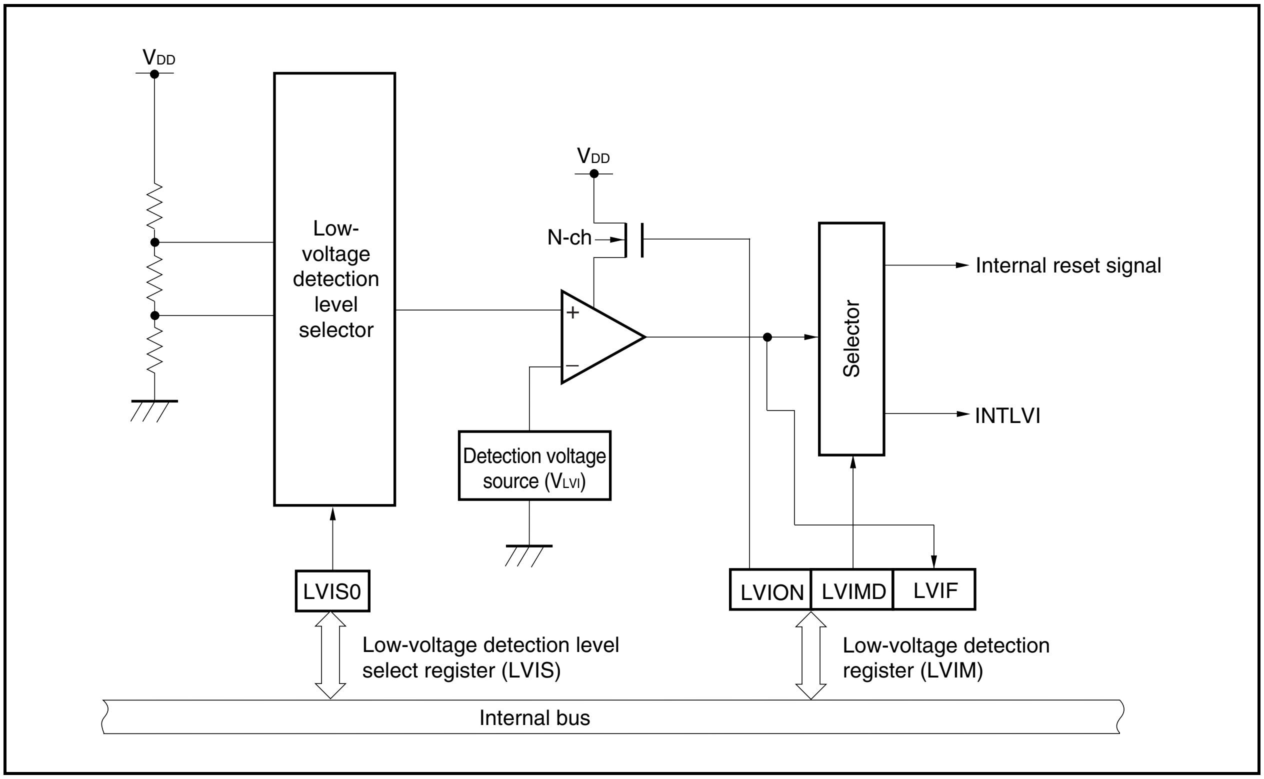
The low-voltage detector (LVI) has the following functions.

- Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}) and generates an interrupt request signal or internal reset signal when $V_{DD} < V_{LVI}$.
- The level of the supply voltage to be detected can be changed in two steps.
- An interrupt request signal or internal reset signal can be selected.
- Can operate in HALT/IDLE/STOP mode.
- Operation can be stopped by software.

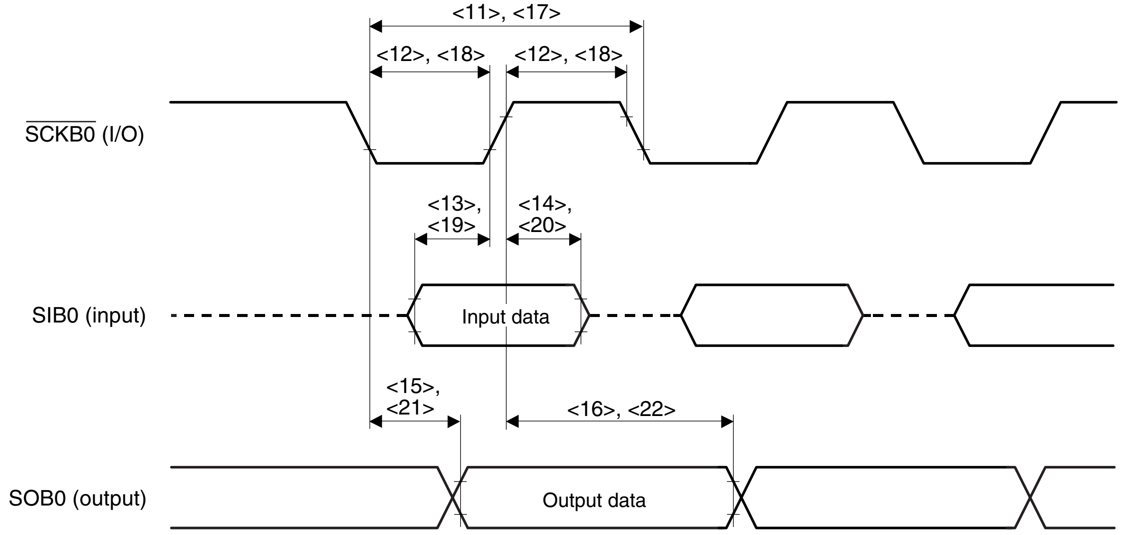
(2) Configuration

The block diagram is shown below.

Figure 16-4. Block Diagram of Low-Voltage Detector

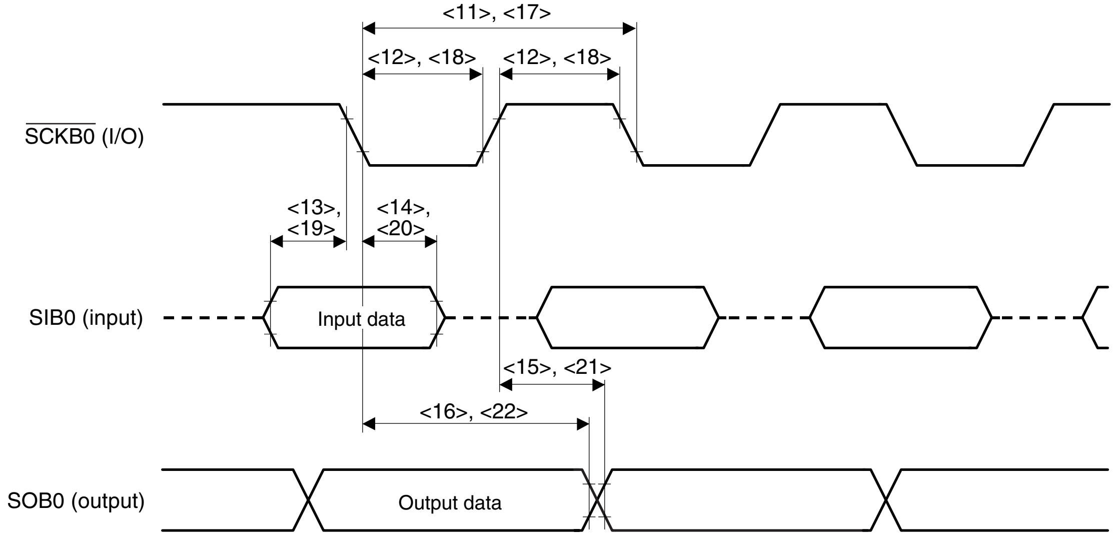

VDD
Low-voltage detection level selector
VDD
N-ch
+
−
Detection voltage source (VLVI)
Selector
Internal reset signal
INTLVI
LVIS0
LVION
LVIMD
LVIF
Low-voltage detection level select register (LVIS)
Low-voltage detection register (LVIM)
Internal bus

CSIB Timing: CB0CTL1.CB0CKP, CB0CTL1.CB0DAP Bits = 00



Remark The broken lines indicate high impedance.

CSIB Timing: CB0CTL1.CB0CKP, CB0CTL1.CB0DAP Bits = 01



Remark The broken lines indicate high impedance.

Characteristics of A/D Converters 0, 1

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 4.5 to 5.5 V, AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 1]					±4.0	LSB
Conversion time	t_{CONV}		2		10	μs
Zero-scale error[Note 1]					±4.0	LSB
Full-scale error[Note 1]					±4.0	LSB
Integral linearity error[Note 1]					±4.0	LSB
Differential linearity error[Note 1]					–1 to +2	LSB
Analog reference voltage	AV_{REF}	AV_{REF0} = AV_{REF1} = AV_{DD0} = AV_{DD1}	4.5		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD0}, AV_{DD1} supply current[Note 2]	AI_{DD}	During operation		6	10	mA
	AI_{DDS}	In STOP mode[Note 3]		0.5	25	μA

<R>

Notes 1. Excluding quantization error (±0.5 LSB).

2. This value is one cycle of A/D converter 0 or A/D converter 1.

3. Stop the operation of A/D converters 0 and 1 (ADAnM0.ADAnCE bit = 0) before setting to the STOP mode.

Remarks 1. LSB: Least Significant Bit

2. n = 0, 1

APPENDIX A CAUTIONS

A.1 Restriction on Conflict Between sld Instruction and Interrupt Request

A.1.1 Description

If a conflict occurs between the decode operation of an instruction in <2> immediately before the sld instruction following an instruction in <1> and an interrupt request before the instruction in <1> is complete, the execution result of the instruction in <1> may not be stored in a register.

Instruction <1>

- ld instruction: ld.b, ld.h, ld.w, ld.bu, ld.hu
- sld instruction: sld.b, sld.h, sld.w, sld.bu, sld.hu
- Multiplication instruction: mul, mulh, mulhi, mulu

Instruction <2>

mov reg1, reg2	not reg1, reg2	satsubr reg1, reg2	satsub reg1, reg2
satadd reg1, reg2	satadd imm5, reg2	or reg1, reg2	xor reg1, reg2
and reg1, reg2	tst reg1, reg2	subr reg1, reg2	sub reg1, reg2
add reg1, reg2	add imm5, reg2	cmp reg1, reg2	cmp imm5, reg2
mulh reg1, reg2	shr imm5, reg2	sar imm5, reg2	shl imm5, reg2

<Example>

```
<i>    ld.w   [r11], r10
              •
              •
              •
<ii>   mov   r10, r28
<iii>  sld.w 0x28, r10
```

If the decode operation of the mov instruction <ii> immediately before the sld instruction <iii> and an interrupt request conflict before execution of the ld instruction <i> is complete, the execution result of instruction <i> may not be stored in a register.

A.1.2 Countermeasure

(1) When compiler (CA850) is used

Use CA850 Ver. 2.61 or later because generation of the corresponding instruction sequence can be automatically suppressed.

(2) For assembler

When executing the sld instruction immediately after instruction <ii>, avoid the above operation using either of the following methods.

- Insert a nop instruction immediately before the sld instruction.
- Do not use the same register as the sld instruction destination register in the above instruction <ii> executed immediately before the sld instruction.