Renesas Electronics America Inc - UPD70F3451GC(R)-UBT-A 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	Not For New Designs
Core Processor	V850ES
Core Size	32-Bit Single-Core
Speed	64MHz
Connectivity	CSI, I²C, UART/USART
Peripherals	DMA, LVD, PWM, WDT
Number of I/O	44
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, 4x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	80-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd70f3451gc-r-ubt-a

○ Timer/counter function:
16-bit interval timer M (TMM): 4 channels
16-bit timer/event counter AA (TAA): 5 channels
16-bit timer/event counter AB (TAB): 2 channels
16-bit timer/event counter T (TMT): 2 channels
Motor control function (uses timer TAB: 2 channels (TAB0, TAB1), TAA: 2 channels (TAA0, TAA1))
- 16-bit accuracy 6-phase PWM function with deadtime: 2 channels
- High-impedance output control function
- A/D trigger generation by timer tuning operation function
- Arbitrary cycle setting function
- Arbitrary deadtime setting function

Watchdog timer: 1 channel

○ Serial interfaces:
Asynchronous serial interface A (UARTA)
Asynchronous serial interface B (UARTB)
Clocked serial interface B (CSIB)
I^2C bus interface (I^2C)

UARTA0/CSIB0:	1 channel
UARTA1/I^2C:	1 channel
UARTA2/CSIB1:	1 channel
UARTB/CSIB2:	1 channel

○ A/D converter:
12-bit resolution A/D converters (A/D converters 0 and 1): 5 channels + 5 channels (2 units)
The one A/D converter 0 channel and three A/D converter 1 channels are provided with an operational amplifier for input level amplification and a comparator for overvoltage detection.
10-bit resolution A/D converter (A/D converter 2): 4 channels

○ Clock generator:
4 to 8 MHz resonator connectable (external clock input prohibited)
Multiplication function by PLL clock synthesizer (fixed to multiplication by eight, f_{XX} = 32 to 64 MHz)
CPU clock division function (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$)

○ Power-save function:
HALT/IDLE/STOP mode

○ Power-on-clear function:

○ Low-voltage detection function:

○ Package:
80-pin plastic LQFP (14 × 14)

○ Operation supply voltage:
$V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2}$ = 4.0 to 5.5 V
(when A/D converter 0, 1 or 2 is operating)
$V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2}$ = 3.5 to 5.5 V
(when none of A/D converters 0 to 2 is operating)

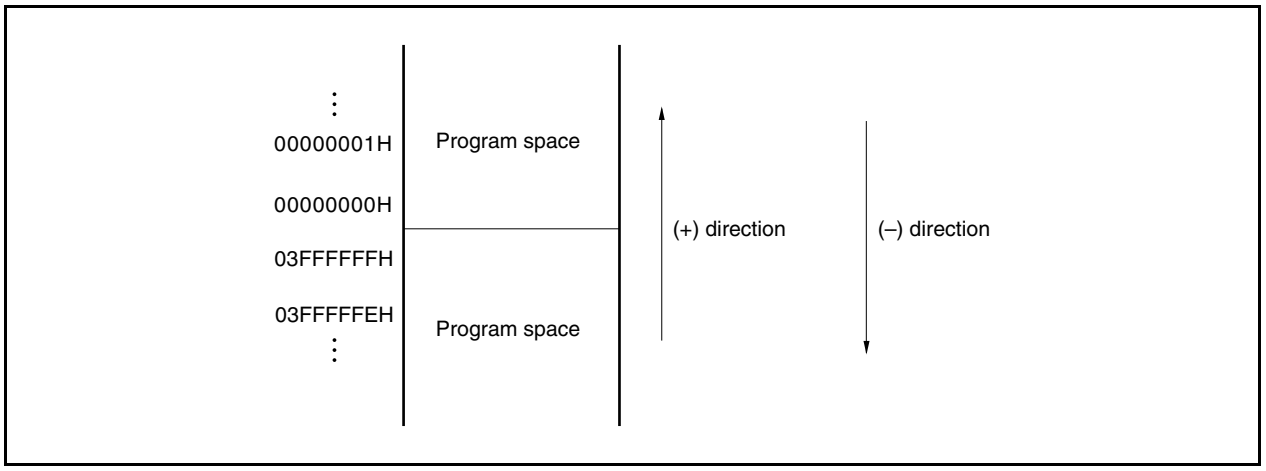
3.4.3 Wraparound of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Even if a carry or borrow occurs from bit 25 to 26 as a result of a branch address calculation, the higher 6 bits ignore the carry or borrow.

Therefore, the upper-limit address of the program space, address 03FFFFFFH, and the lower-limit address 00000000H become contiguous addresses. Wraparound refers to a situation like this whereby the lower-limit address and upper-limit address become contiguous.

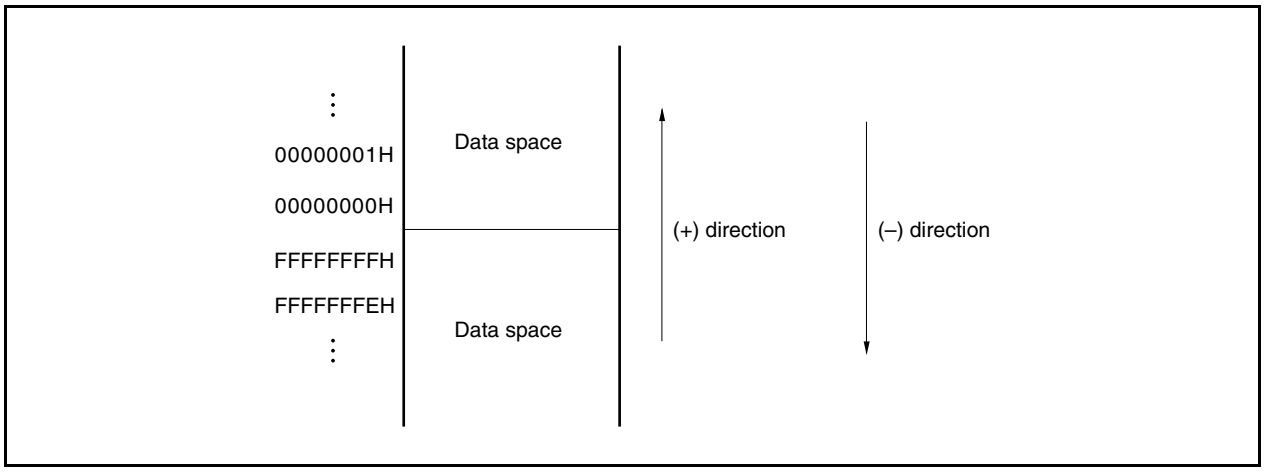
Caution The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. This area is access-prohibited. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

The result of an operand address calculation that exceeds 32 bits is ignored.

Therefore, the upper-limit address of the program space, address FFFFFFFFH, and the lower-limit address 00000000H are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



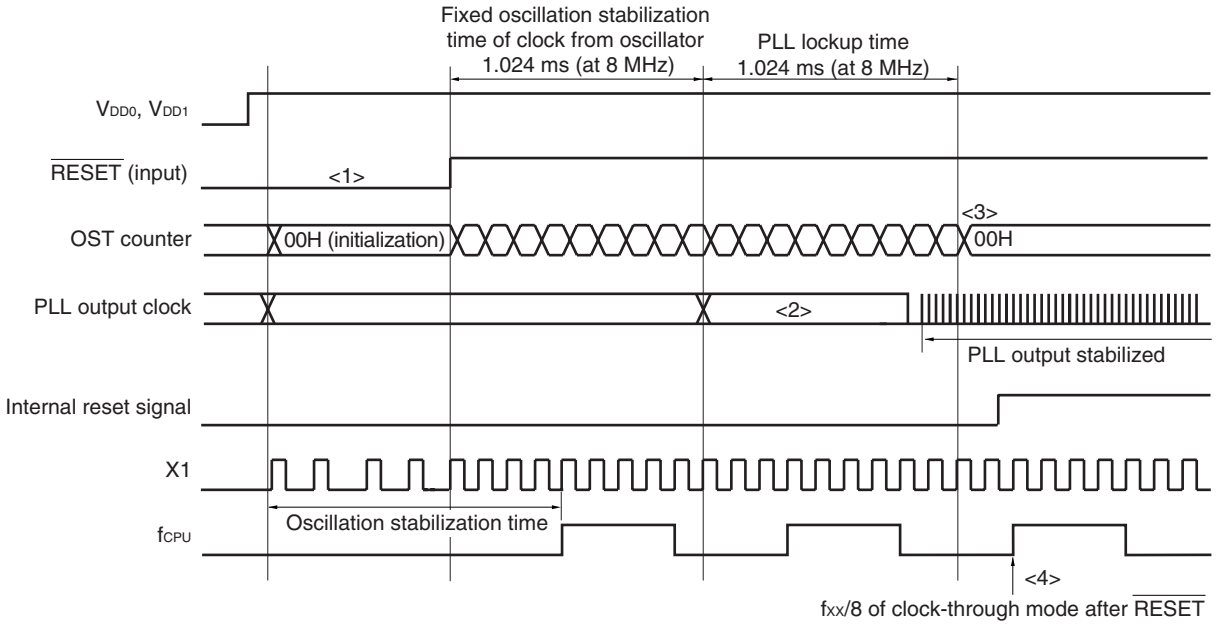
3.4.7 On-chip peripheral I/O registers

(1/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF004H	Port DL register	PDL	R/W			√	Undefined
FFFFF004H	Port DLL register	PDLL		√	√		Undefined
FFFFF005H	Port DLH register	PDLH		√	√		Undefined
FFFFF024H	Port DL mode register	PMDL				√	FFFFH
FFFFF024H	Port DL mode register L	PMDLL		√	√		FFH
FFFFF025H	Port DL mode register H	PMDLH		√	√		FFH
FFFFF044H	Port DL mode control register	PMCDL				√	0000H
FFFFF044H	Port DL mode control register L	PMCDLL		√	√		00H
FFFFF045H	Port DL mode control register H	PMCDLH		√	√		00H
FFFFF066H	Bus size configuration register	BSC				√	5555H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0				√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1				√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2				√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3				√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2				√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3				√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0		√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2		√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3		√	√		00H

5.5.3 Operation timing

(1) Power on (power-on reset)



<1> The oscillator is activated during the $\overline{\text{RESET}}$ period that follows power application.
Make sure that the low-level width of the $\overline{\text{RESET}}$ signal is “Oscillation stabilization time of the used resonator − Fixed oscillation stabilization time” or more, taking the oscillation stabilization time into consideration.
PLL stops during the $\overline{\text{RESET}}$ period and fixed oscillation stabilization time.

<2> When the fixed oscillation stabilization time that elapses after the $\overline{\text{RESET}}$ signal is released expires, PLL stop is released, and counting the lockup time starts.

<3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.

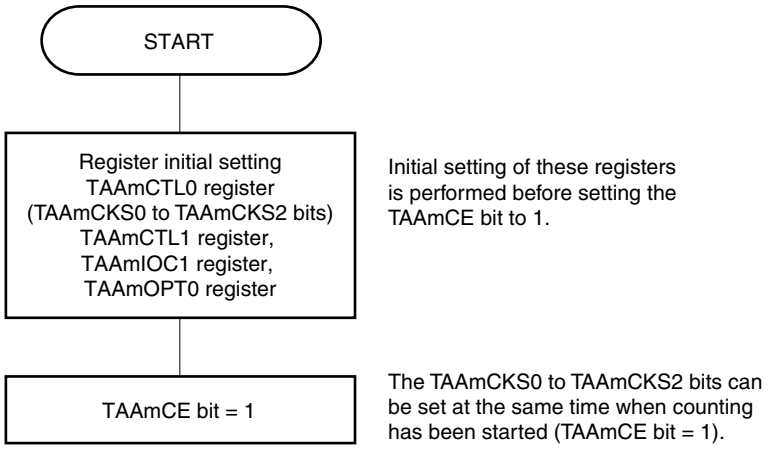
<4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (f_X). The CPU operation clock (f_{CPU}) is f_{XX}/8. The PLL mode can be set by software.

Cautions 1. The clock generated by the oscillator starts oscillating during the $\overline{\text{RESET}}$ period. After the $\overline{\text{RESET}}$ signal is released, a specific wait time (fixed oscillation stabilization time) elapses.

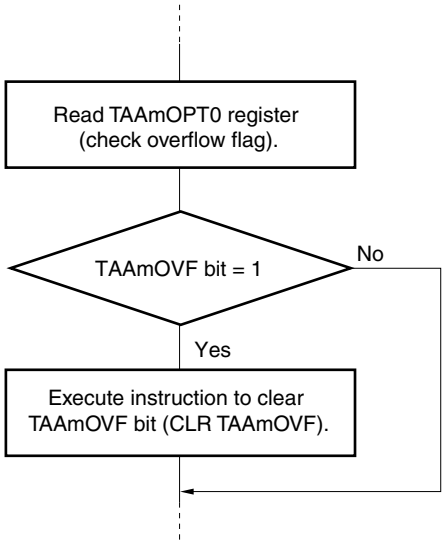
2. To avoid malfunction due to noise, do not change the division ratio of the CPU operation clock (f_{CPU}) by using the PCC register before setting the PLL mode. Before changing the division ratio, be sure to select the PLL mode.

Figure 6-41. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

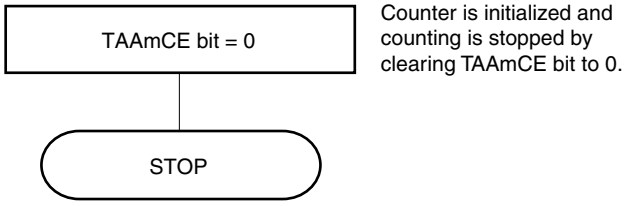
<1> Count operation start flow



<2> Overflow flag clear flow



<3> Count operation stop flow



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

7.5 Timer Output Operations

The following table shows the operations and output levels of the TOBn0 to TOBn3 pins.

Table 7-6. Timer Output Control in Each Mode

Operation Mode	TOBn0 Pin	TOBn1 Pin	TOBn2 Pin	TOBn3 Pin
Interval timer mode	PWM output			
External event count mode	None			
External trigger pulse output mode	PWM output	External trigger pulse output	External trigger pulse output	External trigger pulse output
One-shot pulse output mode		One-shot pulse output	One-shot pulse output	One-shot pulse output
PWM output mode		PWM output	PWM output	PWM output
Free-running timer mode	PWM output (only when compare function is used)			
Pulse width measurement mode	None			

Remark n = 0, 1

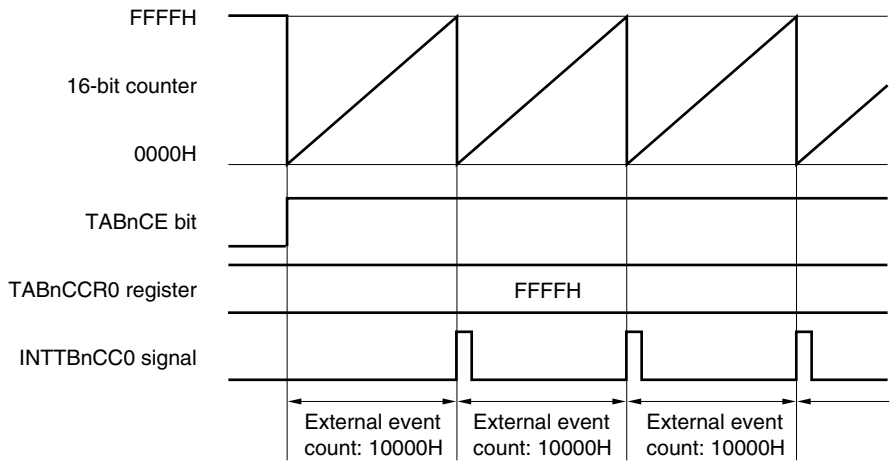
Table 7-7. Truth Table of TOBn0 to TOBn3 Pins Under Control of Timer Output Control Bits

TABnIOC0.TABnOLa Bit	TABnIOC0.TABnOEa Bit	TABnCTL0.TABnCE bit	Level of TOBna Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0, 1
a = 0 to 3

(b) Operation if TABnCCR0 register is set to FFFFH

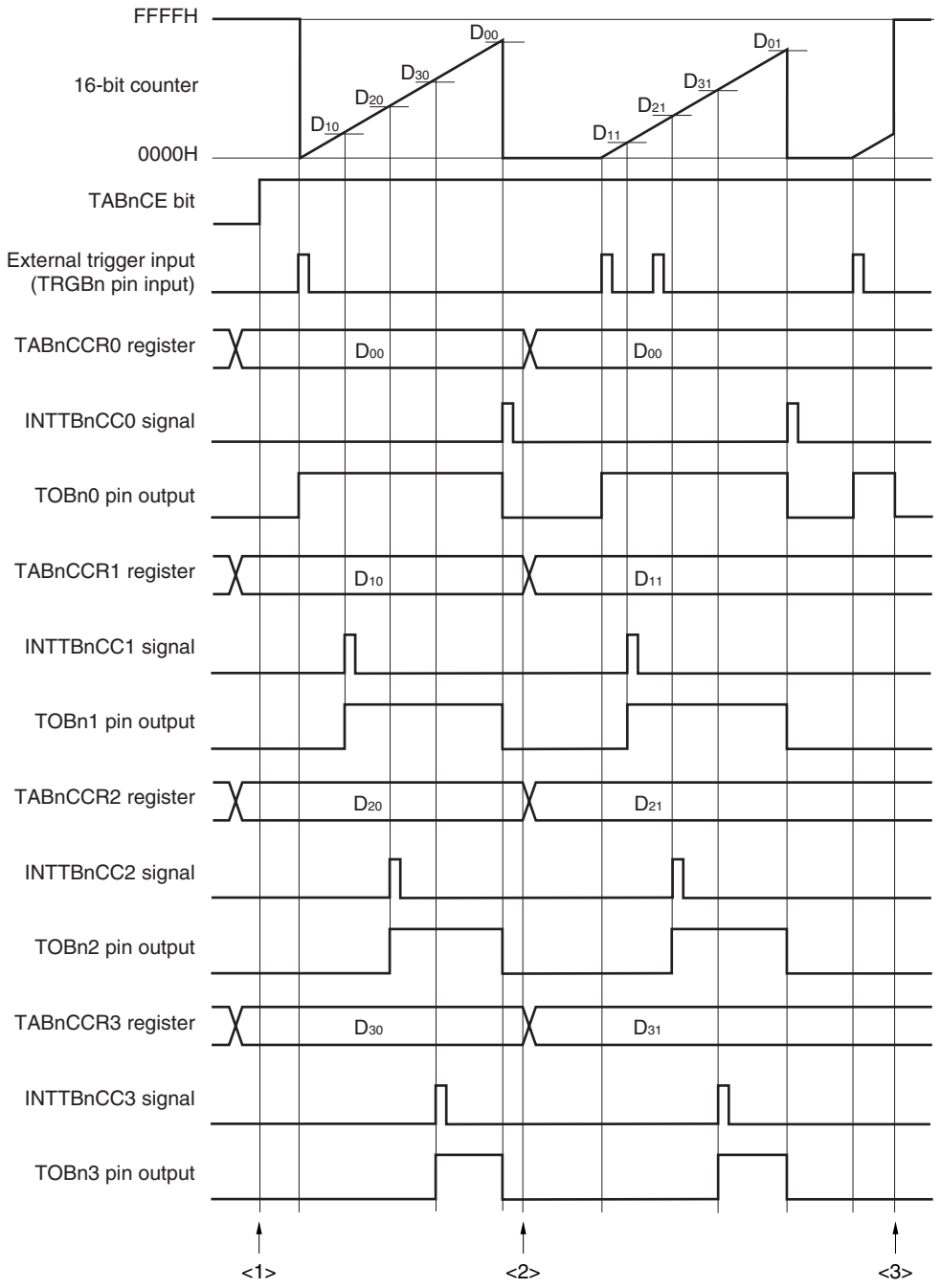
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.



Remark n = 0, 1

(1) Operation flow in one-shot pulse output mode

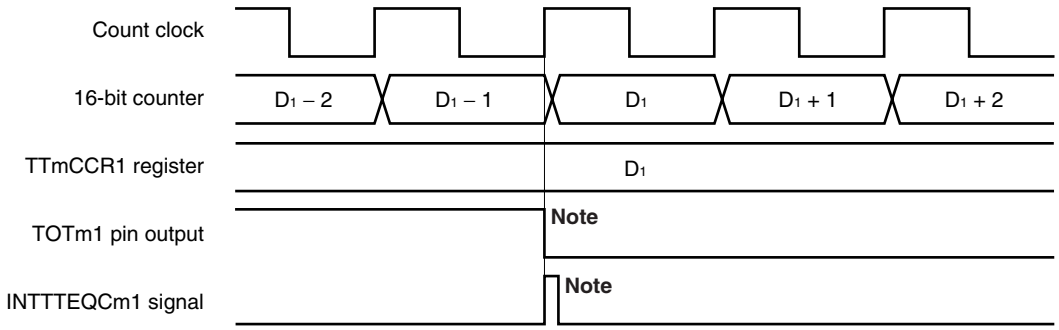
Figure 7-27. Software Processing Flow in One-Shot Pulse Output Mode (1/2)



Remark n = 0, 1

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

The timing of generation of the INTTTEQCm1 signal in the external trigger pulse output mode differs from the timing of 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).

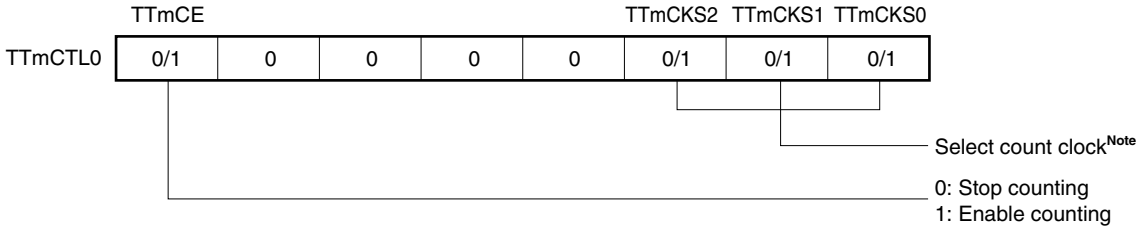
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

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

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOTm1 pin.

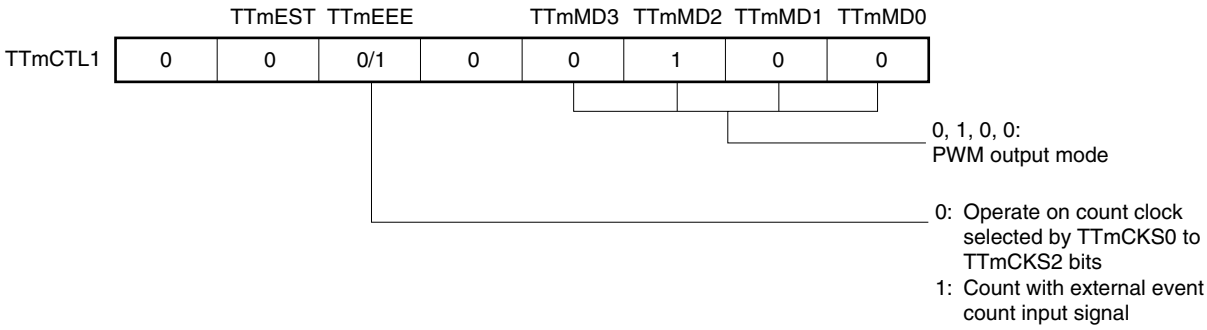
Figure 8-31. Setting of Registers in PWM Output Mode (1/2)

(a) TMTm control register 0 (TTmCTL0)

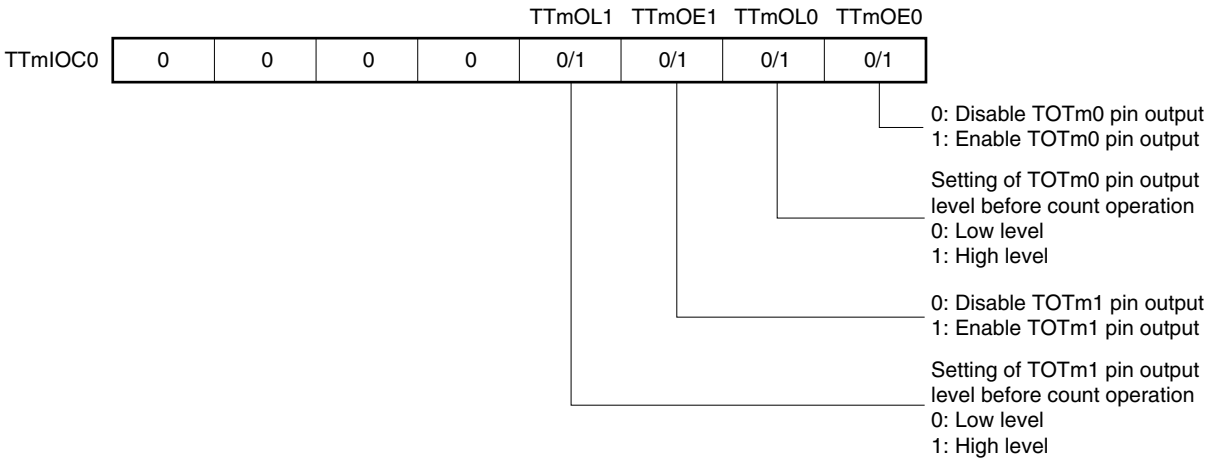


Note The setting is invalid when the TTmCTL1.TTmEEE bit = 1.

(b) TMTm control register 1 (TTmCTL1)



(c) TMTm I/O control register 0 (TTmIOC0)



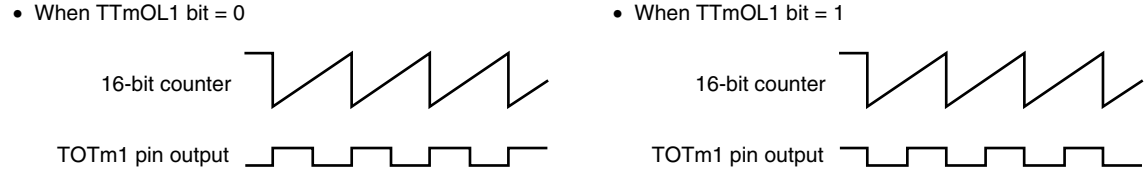
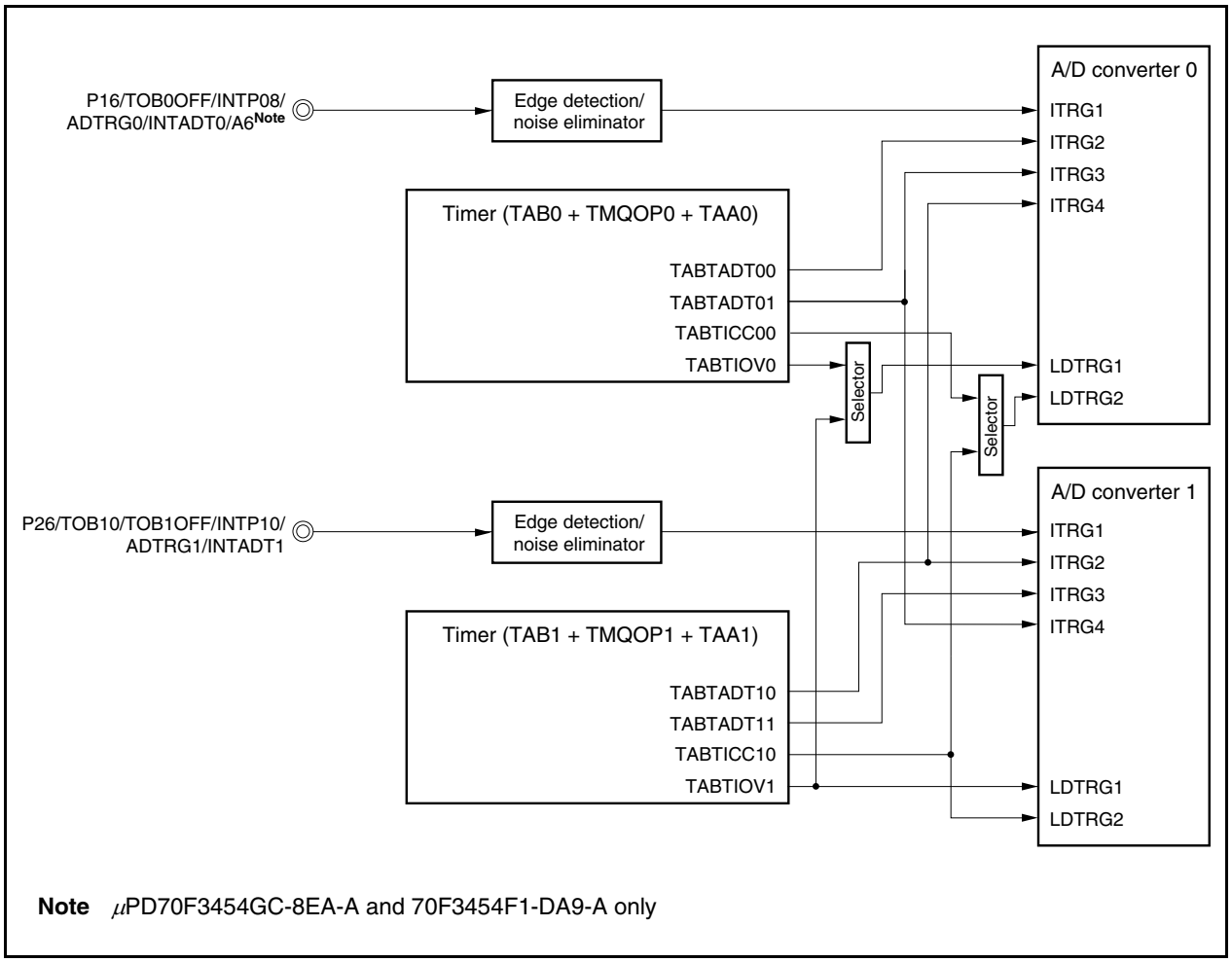


Figure 12-6. Block Diagram of Trigger Source Selector in Hardware Trigger Mode



Note μPD70F3454GC-8EA-A and 70F3454F1-DA9-A only

(6) A/D converter n trigger select register (ADnTSEL)

The ADnTSEL register is a register that specifies trigger in the hardware trigger mode and conversion channel specification mode, and trigger (selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2) in the extension buffer mode.

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

Reset sets this register to 10H.

After reset: 10H R/W Address: AD0TSEL FFFFF231H, AD1TSEL FFFFF2B1H

ADnTSEL (n = 0, 1)

7	6	5	4	3	2	1	0
ADn LDTSEL2 Note	0	ADn TRGSEL21 Note	ADn TRGSEL20 Note	ADn LDTSEL1 Note	0	ADn TRGSEL11	ADn TRGSEL10

ADnLDTSEL2 Note	Specification of selection load trigger 2 for ADnECR3, ADnECR4 registers
0	LDTRG1
1	LDTRG2

ADnTRGSEL21 Note	ADnTRGSEL20 Note	Specification of selection trigger 2 for ADnECR3, ADnECR4 registers
0	0	ITRG1
0	1	ITRG2
1	0	ITRG3
1	1	ITRG4

ADnLDTSEL1 Note	Specification of selection load trigger 1 for ADnECR0 to ADnECR2 registers
0	LDTRG1
1	LDTRG2

ADnTRGSEL11	ADnTRGSEL10	• In hardware trigger mode or conversion channel specification mode: Trigger specification • In expansion buffer mode: Specification of selection trigger 1 for ADnECR0 to ADnECR2 registers
0	0	ITRG1
0	1	ITRG2
1	0	ITRG3
1	1	ITRG4

Note Be sure to set bits 3, 5, and 7 to “0” and set bit 4 to “1” in the hardware trigger mode and conversion channel specification mode.

Caution Set the ADnTSEL register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

(17) Comparator n control register 0 (CMPnCTL0)

The CMPnCTL0 register is a register that controls the operation of the overvoltage detection comparator.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF261H

	7	6	5	4	3	2	1	0
CMP0CTL0	0	0	0	CMP0FEN	0	0	0	CMP0LEN

CMP0FEN	Operation control of comparator 0 (full range) for A/D converter 0
0	Operation disabled (not used)
1	Operation enabled (used)

CMP0LEN	Operation control of comparator 0 (low range) for A/D converter 0
0	Operation disabled (not used)
1	Operation enabled (used)

After reset: 00H R/W Address: FFFFF2E1H

	7	6	5	4	3	2	1	0
CMP1CTL0	0	CMP12FEN	CMP11FEN	CMP10FEN	0	CMP12LEN	CMP11LEN	CMP10LEN

CMP12FEN	Operation control of comparator 2 (full range) for A/D converter 1
0	Operation disabled (not used)
1	Operation enabled (used)

CMP11FEN	Operation control of comparator 1 (full range) for A/D converter 1
0	Operation disabled (not used)
1	Operation enabled (used)

CMP10FEN	Operation control of comparator 0 (full range) for A/D converter 1
0	Operation disabled (not used)
1	Operation enabled (used)

CMP12LEN	Operation control of comparator 2 (low range) for A/D converter 1
0	Operation disabled (not used)
1	Operation enabled (used)

CMP11LEN	Operation control of comparator 1 (low range) for A/D converter 1
0	Operation disabled (not used)
1	Operation enabled (used)

CMP10LEN	Operation control of comparator 0 (low range) for A/D converter 1
0	Operation disabled (not used)
1	Operation enabled (used)

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

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

Similarly, the maximum allowable value can be obtained as follows.

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

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

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

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

The allowable baud rate error of UARTB and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 15-6. Maximum and Minimum Allowable Baud Rate Error

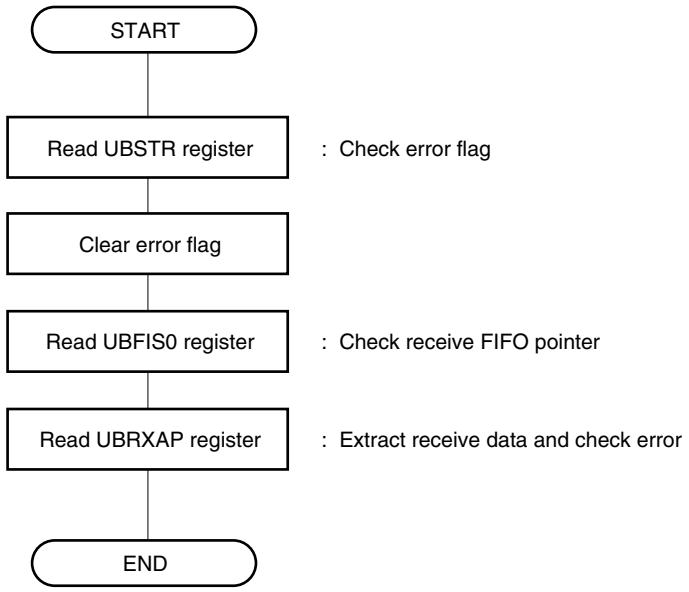
Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.33 %	−2.44
8	+3.53 %	−3.61
16	+4.14 %	−4.19
32	+4.45 %	−4.48
64	+4.61 %	−4.62
128	+4.68 %	−4.69
256	+4.72 %	−4.73
512	+4.74 %	−4.74
1024	+4.75 %	−4.75
2048	+4.76 %	−4.76
4096	+4.76 %	−4.76
8192	+4.76 %	−4.76
16384	+4.76 %	−4.76
32768	+4.76 %	−4.76
65535	+4.76 %	−4.76

Remarks 1. The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: UBCTL2 set value

(11) Example of reception error processing flow in FIFO mode (2)

Figure 15-22. Example of Reception Error Processing Flow in FIFO Mode (2)



Caution Reception can be continued by completing this control flow before reception of the next data is ended. Extract the receive data and check if a reception error has occurred before receive FIFO becomes empty. Note that this control flow is valid only when a parity error or a framing error occurs. If an overflow error occurs, receive FIFO must be cleared (UBFIC0.UBRFC bit = 1).

If the next data is received before this control flow is ended, a reception error interrupt request signal (INTUBTIRE) may occur even if the data has been received correctly.

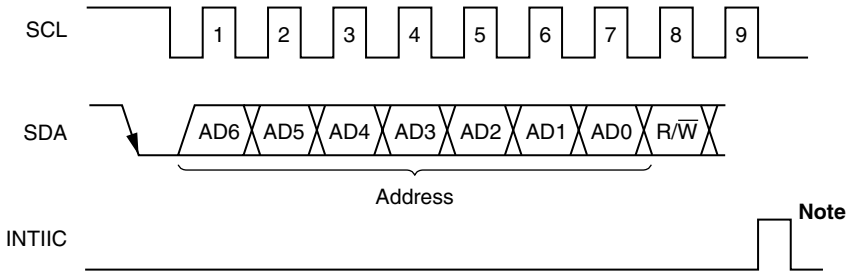
17.6.2 Addresses

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.

The following shows an example of the processing of the slave device by an INTIIC interrupt (it is assumed that no extension codes are used here). During an INTIIC interrupt, the status is confirmed and the following steps are executed.

<1> When a stop condition is detected, communication is terminated.
<2> When a start condition is detected, the address is confirmed. If the address does not match, communication is terminated. If the address matches, the communication mode is set and wait is released, and operation returns from the interrupt (the ready flag is cleared).
<3> For data transmission/reception, when the ready flag is set, operation returns from the interrupt while the I²C bus remains in the wait status.

Remark <1> to <3> in the above correspond to <1> to <3> in **Figure 17-20 Slave Operation Flowchart (2)**.

Figure 17-20. Slave Operation Flowchart (2)

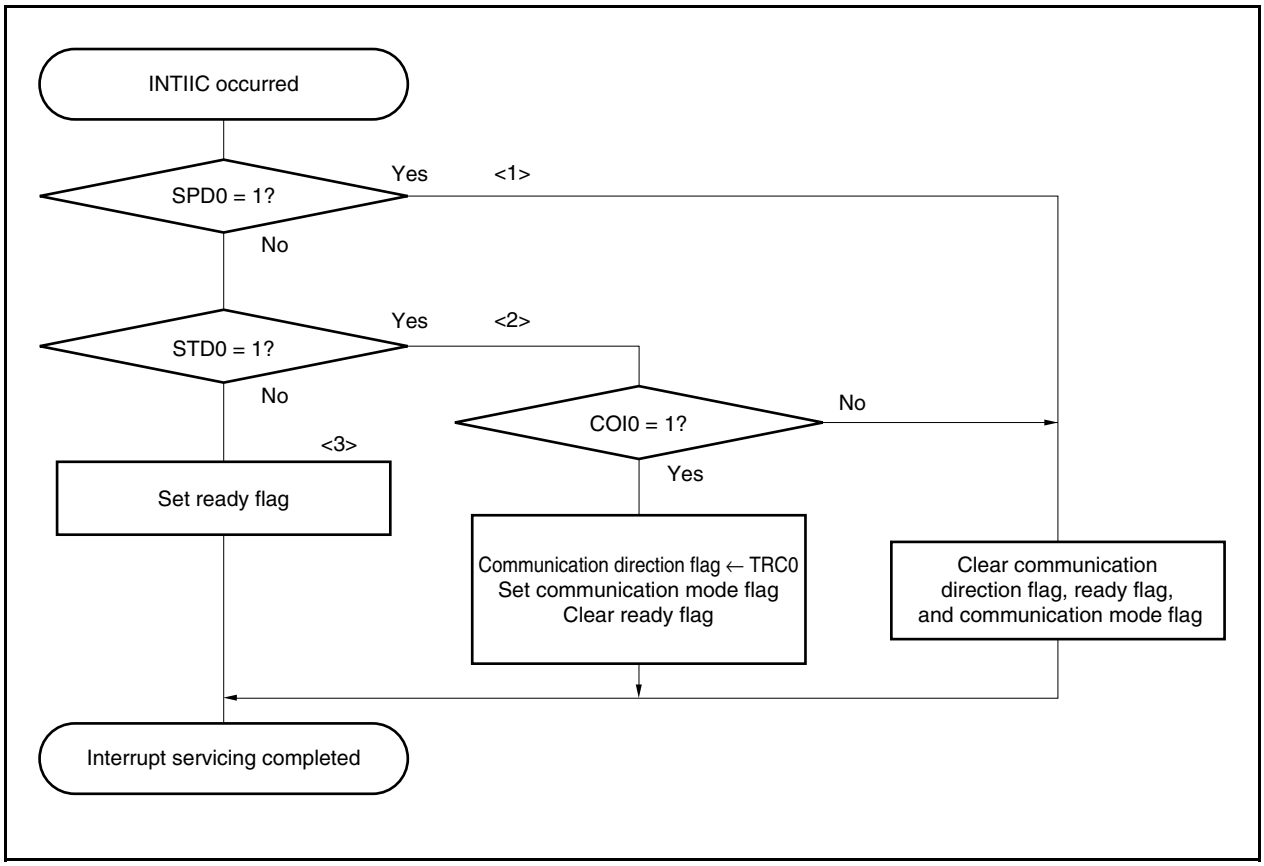
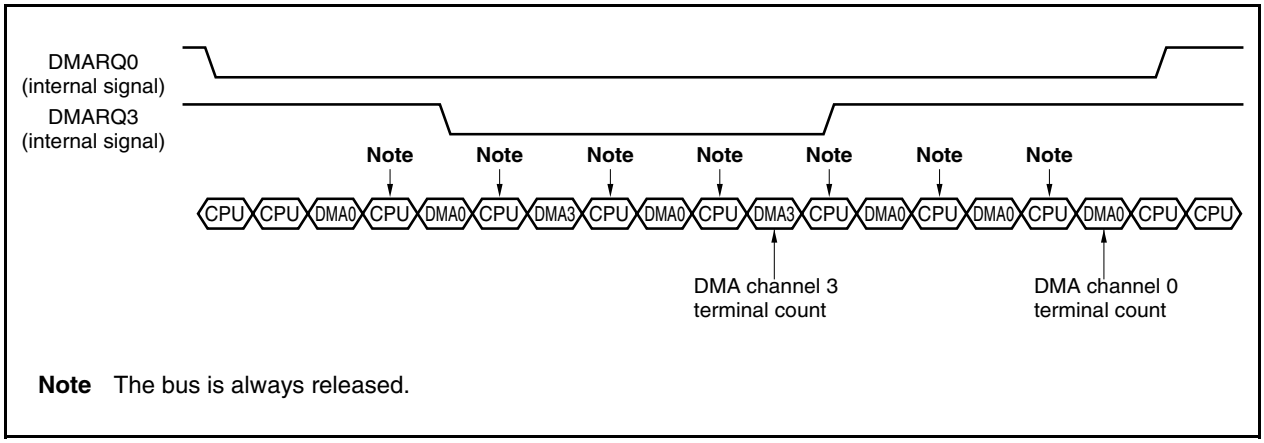


Figure 19-3 is an example of single transfer where a DMA transfer request with a lower priority is issued one clock after single transfer has been completed. DMA channels 0 and 3 are used for single transfer. If two DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed.

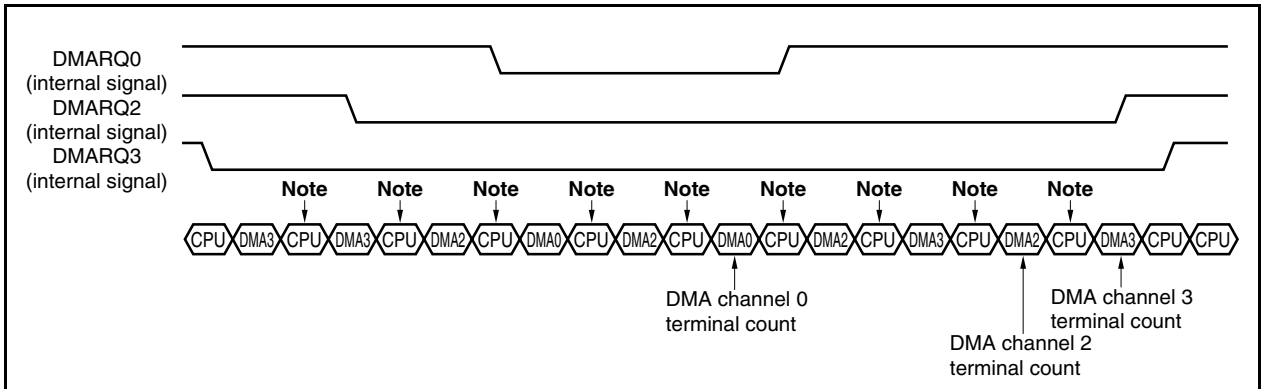
Figure 19-3. Single Transfer Example 3



Note The bus is always released.

Figure 19-4 is an example of single transfer where two or more DMA transfer requests with a lower priority are issued one clock after single transfer has been completed. DMA channels 0, 2, and 3 are used for single transfer. If three or more DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed, starting from the one with the highest priority.

Figure 19-4. Single Transfer Example 4



Note The bus is always released.

CHAPTER 26 ON-CHIP DEBUG FUNCTION

The on-chip debug function of the V850E/IF3 and V850E/IG3 can be realized in the following two ways.

- Debugging using DCU (debug control unit) (using MINICUBE)
  By using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins as debug interface pins, on-chip debugging is realized by the internal DCU of the V850E/IG3[Note].
- Debugging without using DCU (using MINICUBE2)
  On-chip debugging is realized by MINICUBE2 without using the DCU but by using the user resources.

Note The V850E/IF3 does not have an internal DCU.

The following table shows the features of the two on-chip debug functions.

Table 26-1. On-Chip Debug Function Features

		Debugging Using DCU	Debugging Without Using DCU
Target product		V850E/IG3	V850E/IF3, V850E/IG3
Debug interface pins		$\overline{\text{DRST}}$, DCK, DMS, DDI, DDO	• When UARTA0 is used RXDA0, TXDA0 • When CSIB0 is used SIB0, SOB0, $\overline{\text{SCKB0}}$, HS (P43)
Securing of user resources		Not required	Required
Hardware break function		2 points	2 points (V850E/IG3 only)
Software break function	Internal ROM area	4 points	4 points
	RAM area	2000 points	2000 points
Real-time RAM monitor function[Note 1]		Available	Available
Dynamic memory modification (DMM) function[Note 2]		Available	Available
Mask function		Reset, INTWDT, $\overline{\text{WAIT}}$[Note 3]	$\overline{\text{RESET}}$, $\overline{\text{WAIT}}$[Note 3]
ROM security function		10-byte ID code authentication	10-byte ID code authentication
Hardware used		MINICUBE	MINICUBE2
Trace function		Not supported	Not supported
Debug interrupt interface function (DBINT)		Not supported	Not supported

Notes 1. This is a function which reads out memory contents during program execution.
2. This is a function which rewrites RAM contents during program execution.
3. μPD70F3454GC-8EA-A and 70F3454F1-DA9-A only

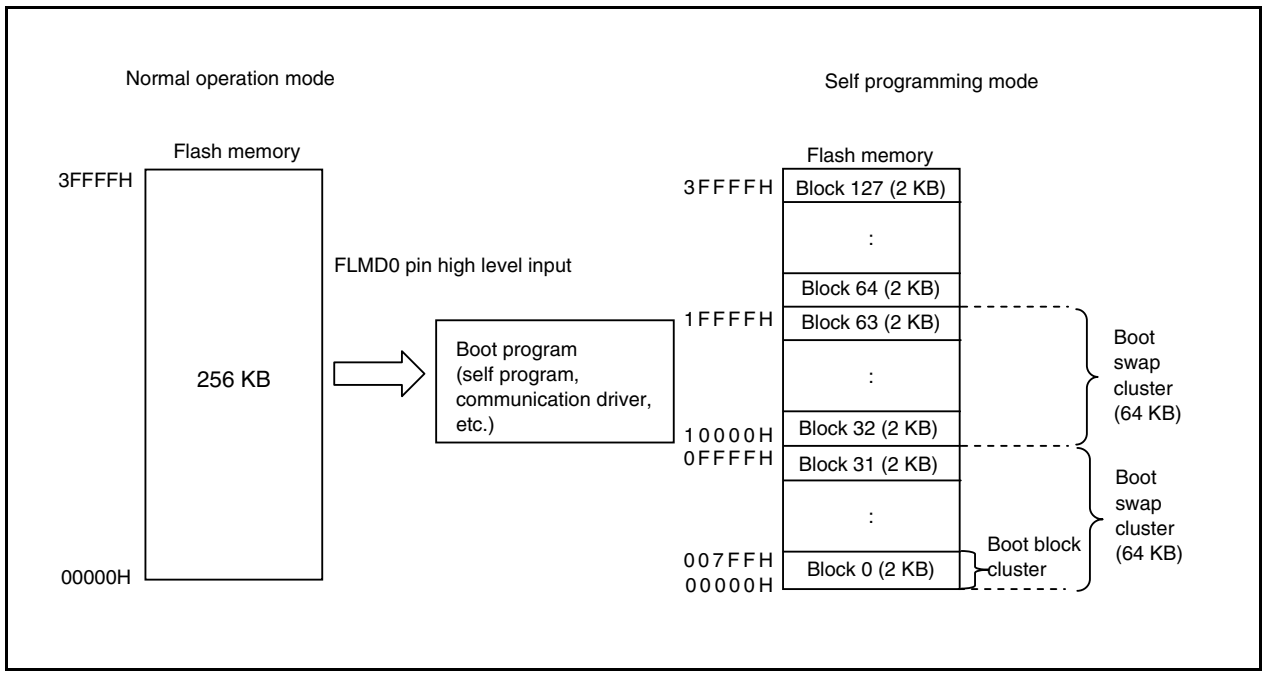
27.9.2 Features

(1) Flash memory self programming

Flash memory self programming is used to erase or write the flash memory by calling the flash function from a program stored in an area other than the flash memory area to be erased or written. To store the program that implements self programming in the area to be erased or written, copy the program to the internal RAM area, execute the program at the copy destination, and call the flash function.

To call the flash function, change the mode from the normal operation mode to the self programming mode by using the flash programming mode control register.

Figure 27-3. Self Programming



(a) Boot swap cluster

The contents of the boot swap cluster of the lower address side (00000H to 0FFFFH) and the boot swap cluster of the higher address side (10000H to 1FFFFH) can be interchanged while flash memory programming is performed.

(b) Boot block cluster

By specifying the boot block cluster from 00000H in 2 KB units, the contents of the boot block cluster can be protected from rewriting.