To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

MITSUBISHI 16-BIT SINGLE-CHIP MICROCOMPUTER
M16C FAMILY / M16C/20 SERIES

M30245 Group

User's Manual

Preliminary Specifications
Chapter 3

http://www.infomicom.maec.co.jp/indexe.htm

Before using this material, please visit the above website to confirm that this is the most current document available.

RENESAS
Renesas Technology Corp.

Preliminary Specifications Rev. A
Revision date: Jan. 24, 2003

Keep safety first in your circuit designs!

- Mitsubishi Electric Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.
  The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for further details on these materials or the products contained therein.

REVISION HISTORY	M30245 GROUP USER'S MANUAL

Rev.	Date	Description	
		Page	Summary
A	1/24/2003		First edition

Chapter 3

Examples of Peripheral Functions Applications

Applications

This chapter presents applications in which peripheral functions built in the M30245 are used. They are shown here as examples. In practical use, make suitable changes and perform sufficient evaluation. For basic use, see Chapter 2 Peripheral Functions Usage.

[MEMO]

3.1 Long-Period Timers

Overview In this process, Timer A0 and Timer A1 are connected to make a 16-bit timer with a 16-bit prescaler. Figure 3.1.1 shows the operation timing, Figure 3.1.2 shows the connection diagram, and Figures 3.1.3 and 3.1.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Event counter mode of timer A

Specifications

(1) Set timer A0 to timer mode, and set timer A1 to event counter mode.

(2) Perform a count on count source f_1 using timer A0 to count for 1 ms, and perform a count on timer A0 using timer A1 to count for 1 second.

(3) Connect a 16-MHz oscillator to XIN.

Operation (1) Setting the count start flag to “1” causes the counter to begin counting. The counter of timer A0 performs a down count on count source f_1.

(2) If the counter of timer A0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A0 interrupt request bit goes to “1”. The counter of timer A1 performs a down count on underflows in timer A0.

(3) If the counter of timer A1 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A1 interrupt request bit goes to “1”.

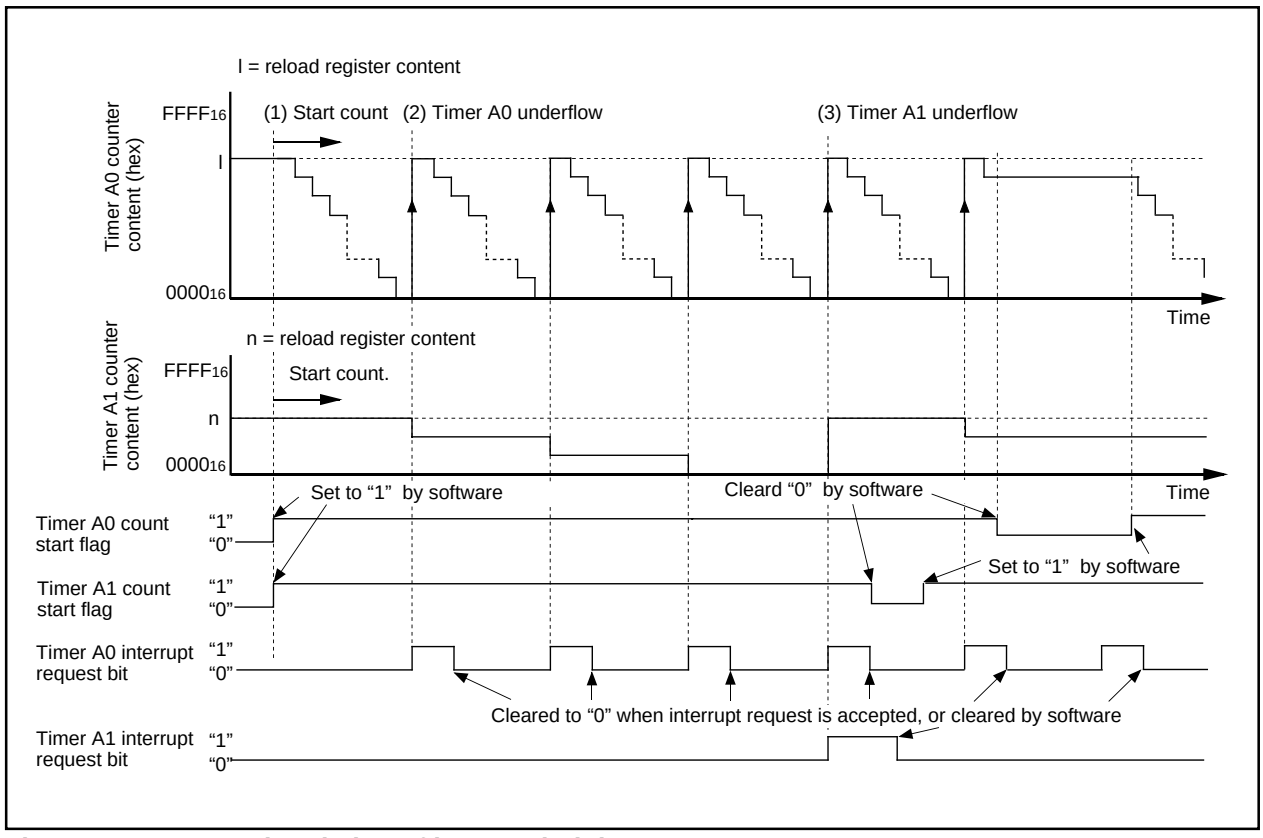


Figure 3.1.1. Operation timing of long-period timers

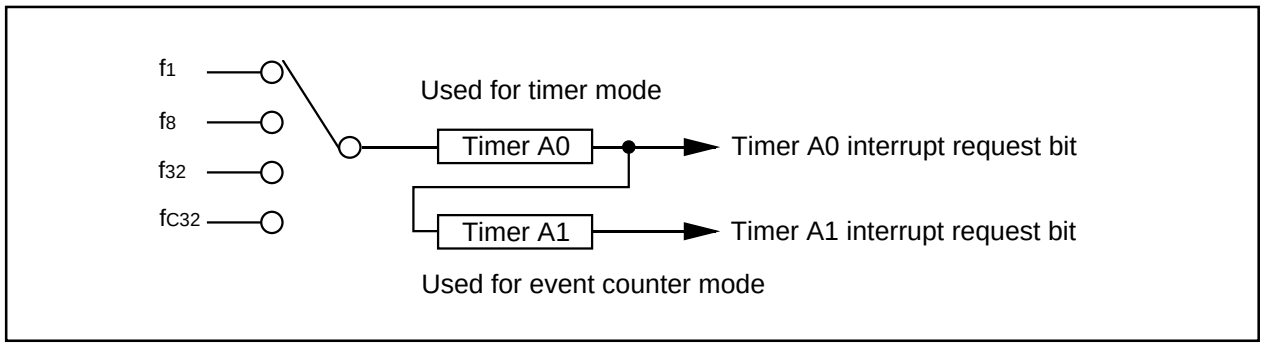
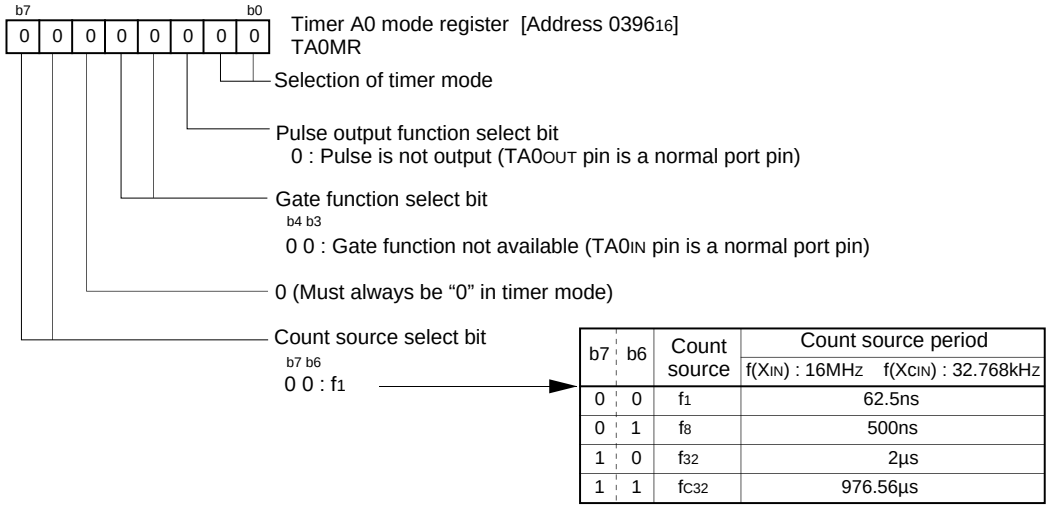


Figure 3.1.2. Connection diagram of long-period timers

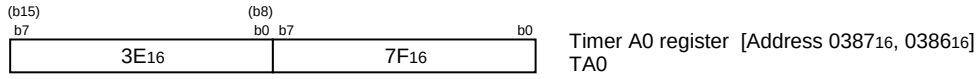
Setting timer A0

Selecting timer mode and functions



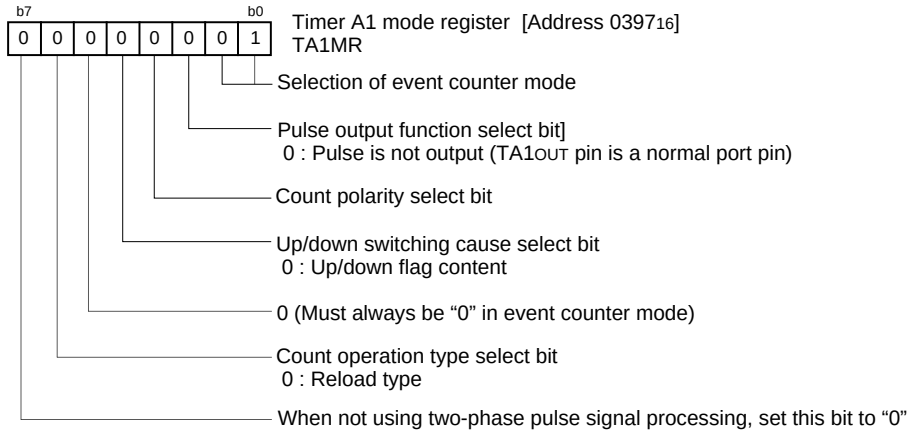
b7	b6	Count source	Count source period f(XIN) : 16MHz f(XCIN) : 32.768kHz
0	0	f1	62.5ns
0	1	f8	500ns
1	0	f32	2µs
1	1	fC32	976.56µs

Setting counter value



Setting timer A1

Selecting event counter mode and each function



Continued to the next page

Figure 3.1.3. Set-up procedure of long-period timers (1)

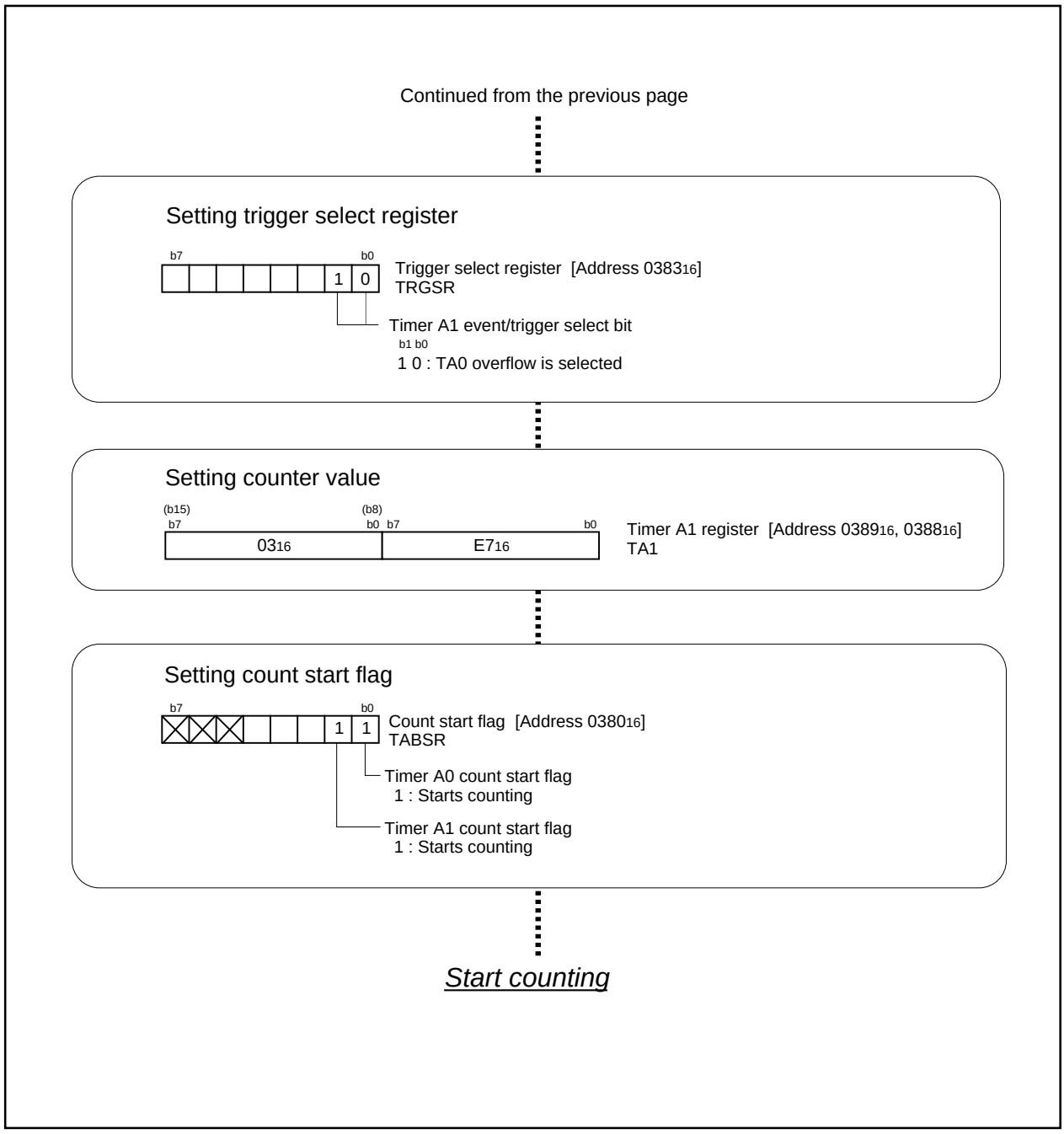


Figure 3.1.4. Set-up procedure of long-period timers (2)

3.2 Variable-Period Variable-Duty PWM Output

Overview In this process, Timer A0 and A1 are used to generate variable-period, variable-duty PWM output. Figure 3.2.1 shows the operation timing, Figure 3.2.2 shows the connection diagram, and Figures 3.2.3 and 3.2.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- One-shot timer mode of timer A

Specifications

(1) Set timer A0 in timer mode, and set timer A1 in one-shot timer mode with pulse-output function.

(2) Set 1 ms, the PWM period, to timer A0. Set 500 µs, the width of PWM "H" pulse, to timer A1. Both timer A0 and timer A1 use f_1 for the count source.

(3) Connect a 16-MHz oscillator to XIN.

Operation (1) Setting the count start flag to "1" causes the counter of timer A0 to begin counting. The counter of timer A0 performs a down count on count source f_1.

(2) If the counter of timer A0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer A0 interrupt request bit goes to "1".

(3) An underflow in timer A0 triggers the counter of timer A1 and causes it to begin counting. When the counter of timer A1 begins counting, the output level of the TA1OUT pin goes to "H".

(4) As soon as the count of the counter of timer A1 becomes "0000_{16}", the output level of TA1OUT pin goes to "L", and the counter reloads the content of the reload register and stops counting. At the same time, the timer A1 interrupt request bit goes to "1".

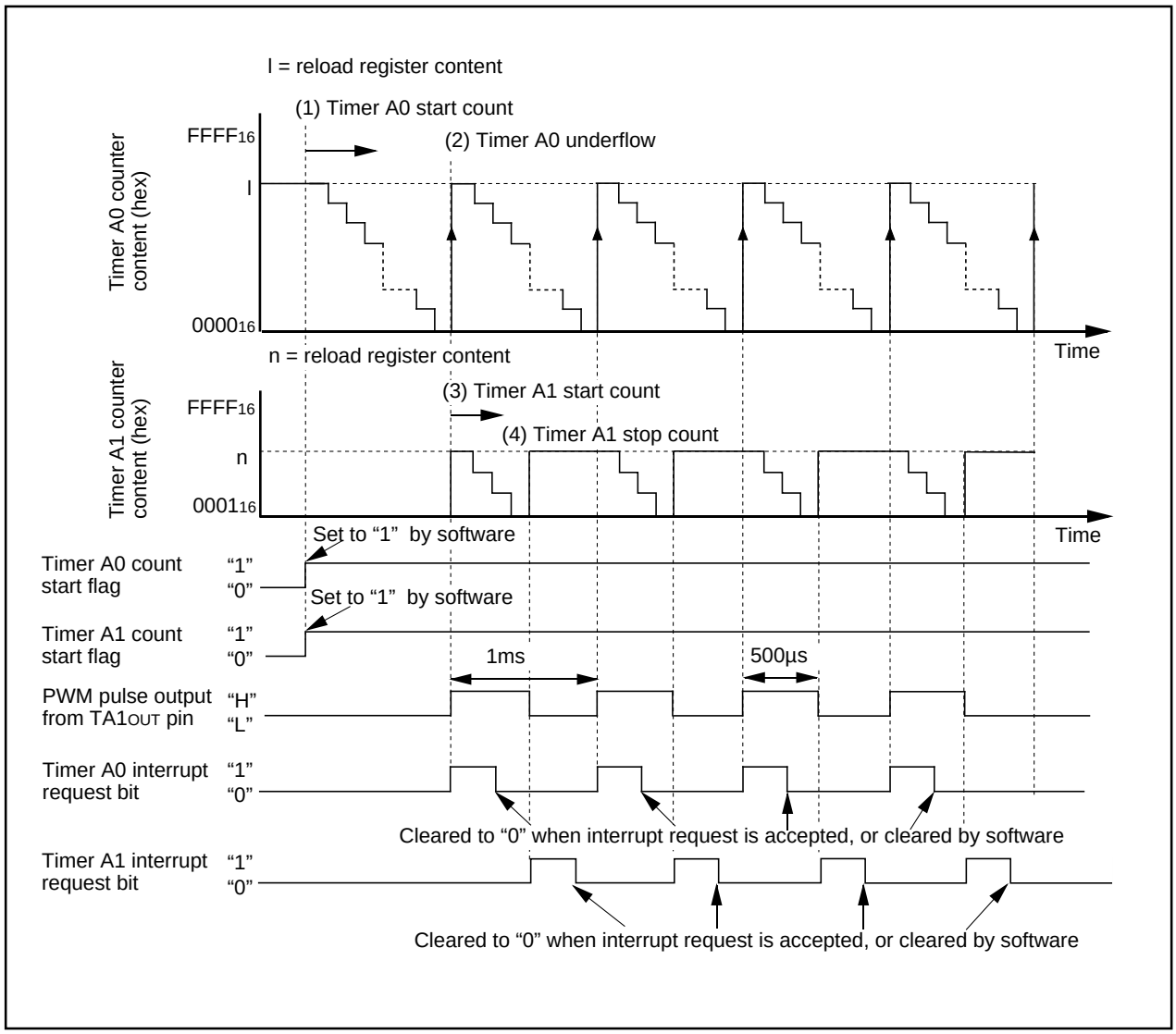


Figure 3.2.1. Operation timing of variable-period variable-duty PWM output

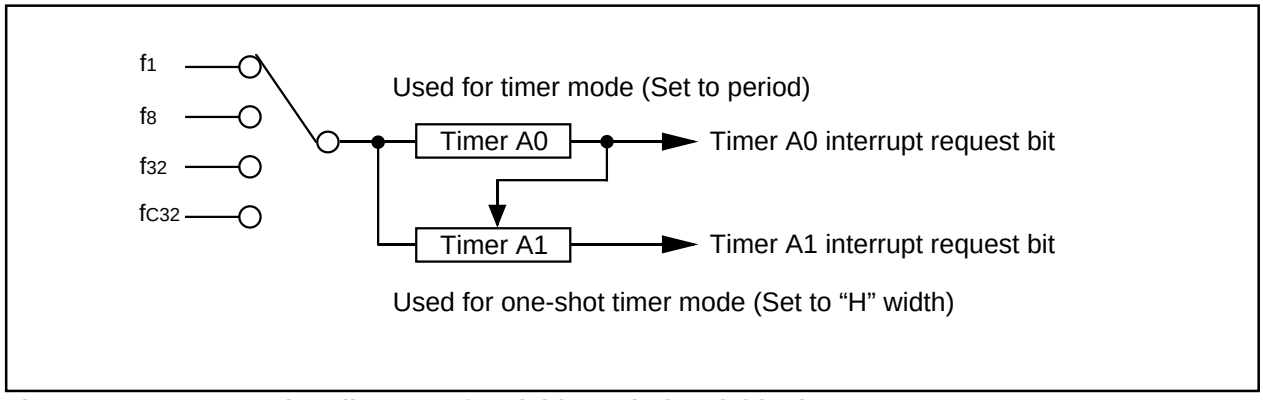


Figure 3.2.2. Connection diagram of variable-period variable-duty PWM output

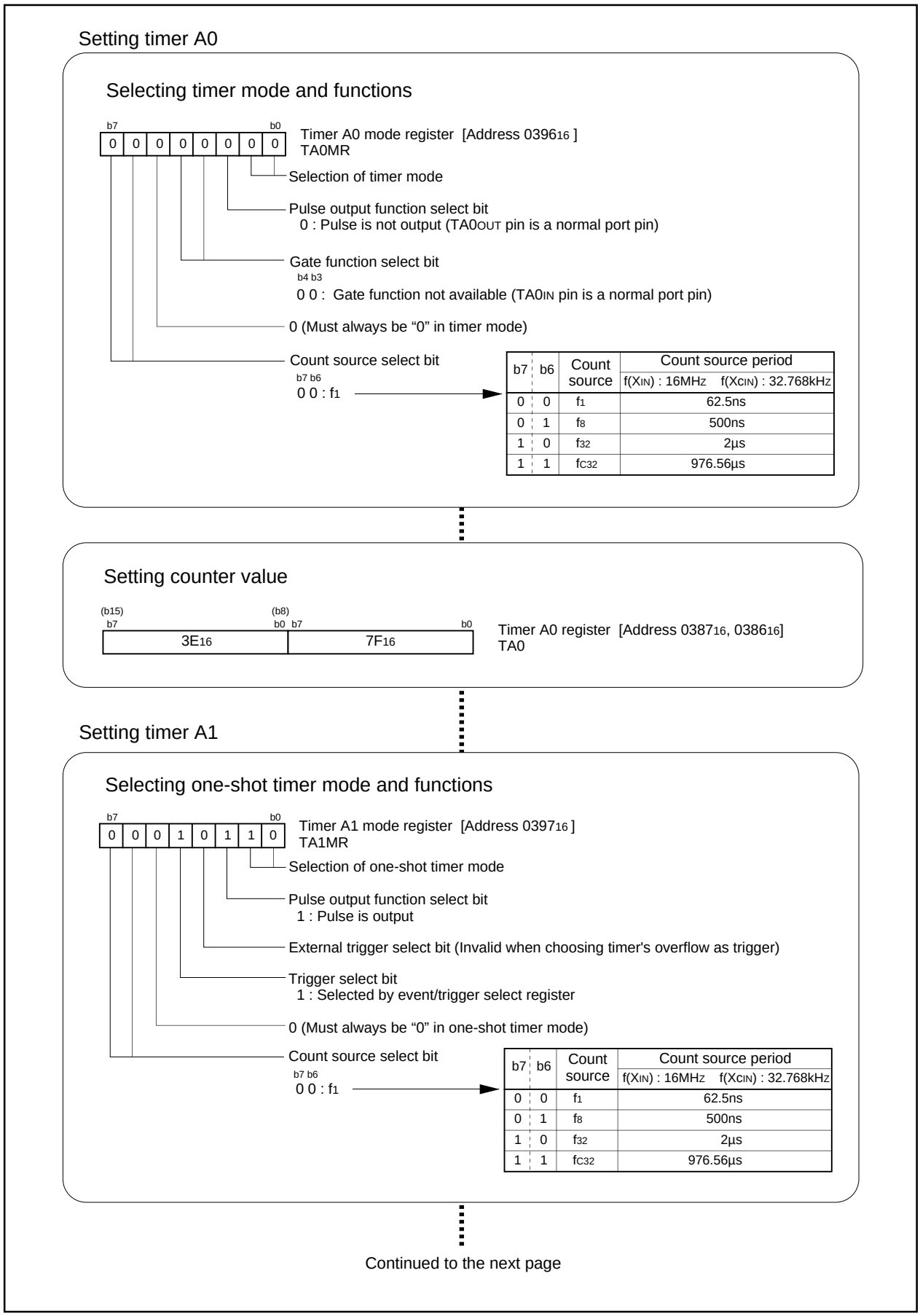


Figure 3.2.3. Set-up procedure of variable-period variable-duty PWM output (1)

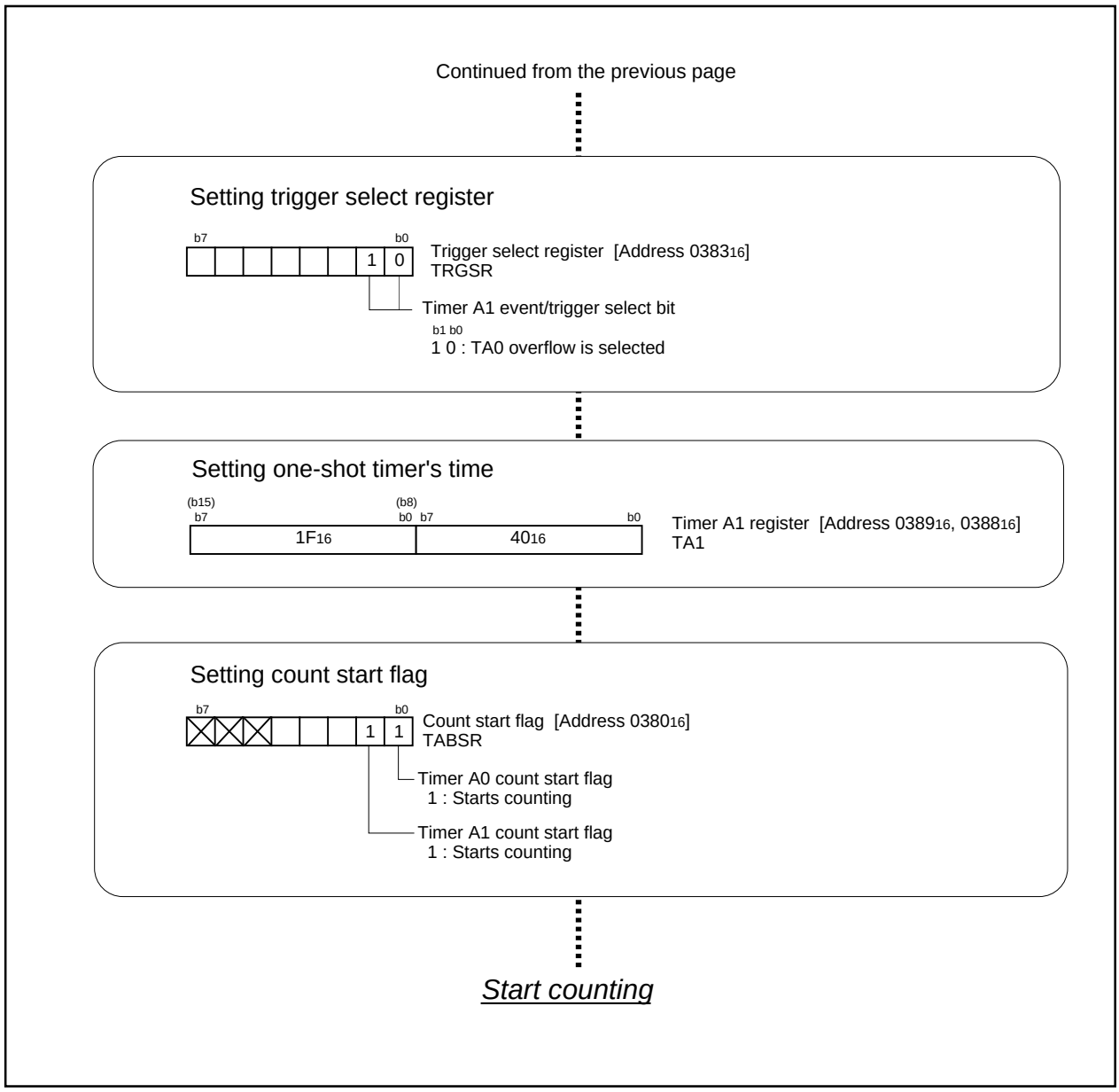


Figure 3.2.4. Set-up procedure of variable-period variable-duty PWM output (2)

3.3 Delayed One-Shot Output

Overview The following are steps of outputting a pulse only once after a specified elapse since an external trigger is input. Figure 3.3.1 shows the operation timing, Figure 3.3.2 shows the connection diagram, and Figures 3.3.3 and 3.3.4 show the set-up procedure.

Use the following peripheral function:

- One-shot timer mode of timer A

Specifications

(1) Set timer A0 in one-shot timer mode, and set timer A1 in one-shot timer mode with pulse-output function.

(2) Set 1 ms, an interval before a pulse is output, in timer A0; and set 50 µs, a pulse width, in timer A1. Both timer A0 and timer A1 use f_1 for the count source.

(3) Connect a 16-MHz oscillator to X_{IN}.

Operation (1) Setting the trigger select bit to “1” and setting the count start flag to “1” enables the counter of timer A0 to count.

(2) If an effective edge, selected by use of the external trigger select bit, is input to the $TA0_{IN}$ pin, the counter begins a down count. The counter of timer A0 performs a down count on count source f_1.

(3) As soon as the counter of timer A0 becomes “0000_{16}”, the counter reloads the content of the reload register and stops counting. At this time, the timer A0 interrupt request bit goes to “1”.

(4) An underflow in timer A0 triggers the counter of timer A1 and causes it to begin counting. When timer A1 begins counting, the output level of the $TA1_{OUT}$ pin goes to “H”.

(5) As soon as the counter of timer A1 becomes “0000_{16}”, the output level of the $TA1_{OUT}$ pin goes to “L”, the counter reloads the content of the reload register, and stops counting. At this time, timer A1 interrupt request bit goes to “1”.

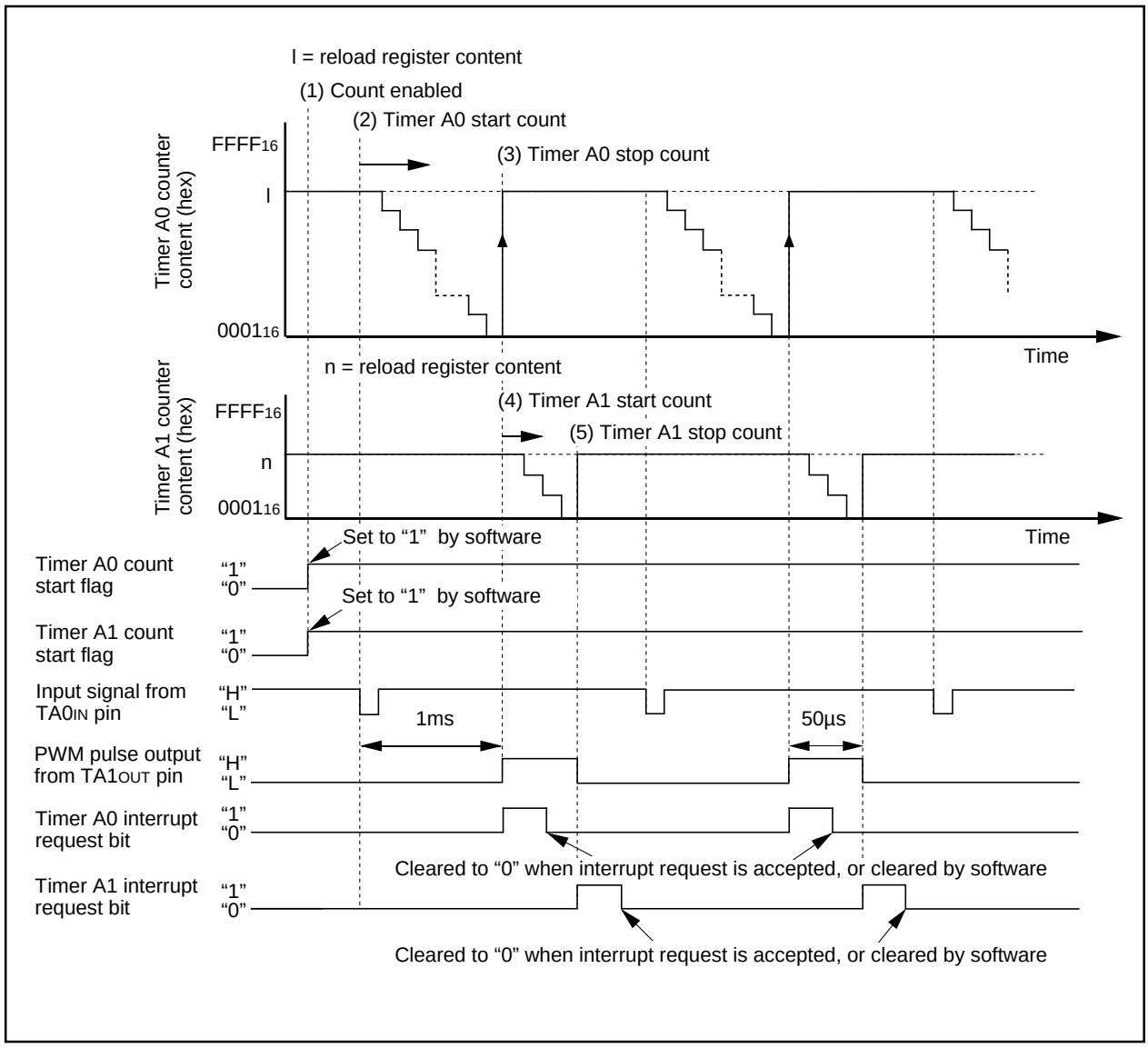


Figure 3.3.1. Operation timing of delayed one-shot output

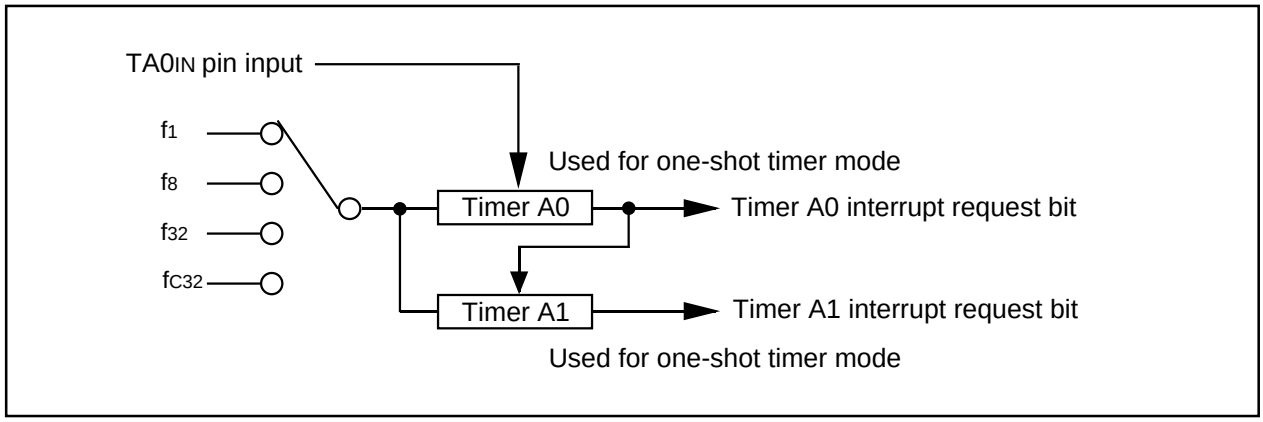


Figure 3.3.2. Connection diagram of delayed one-shot output

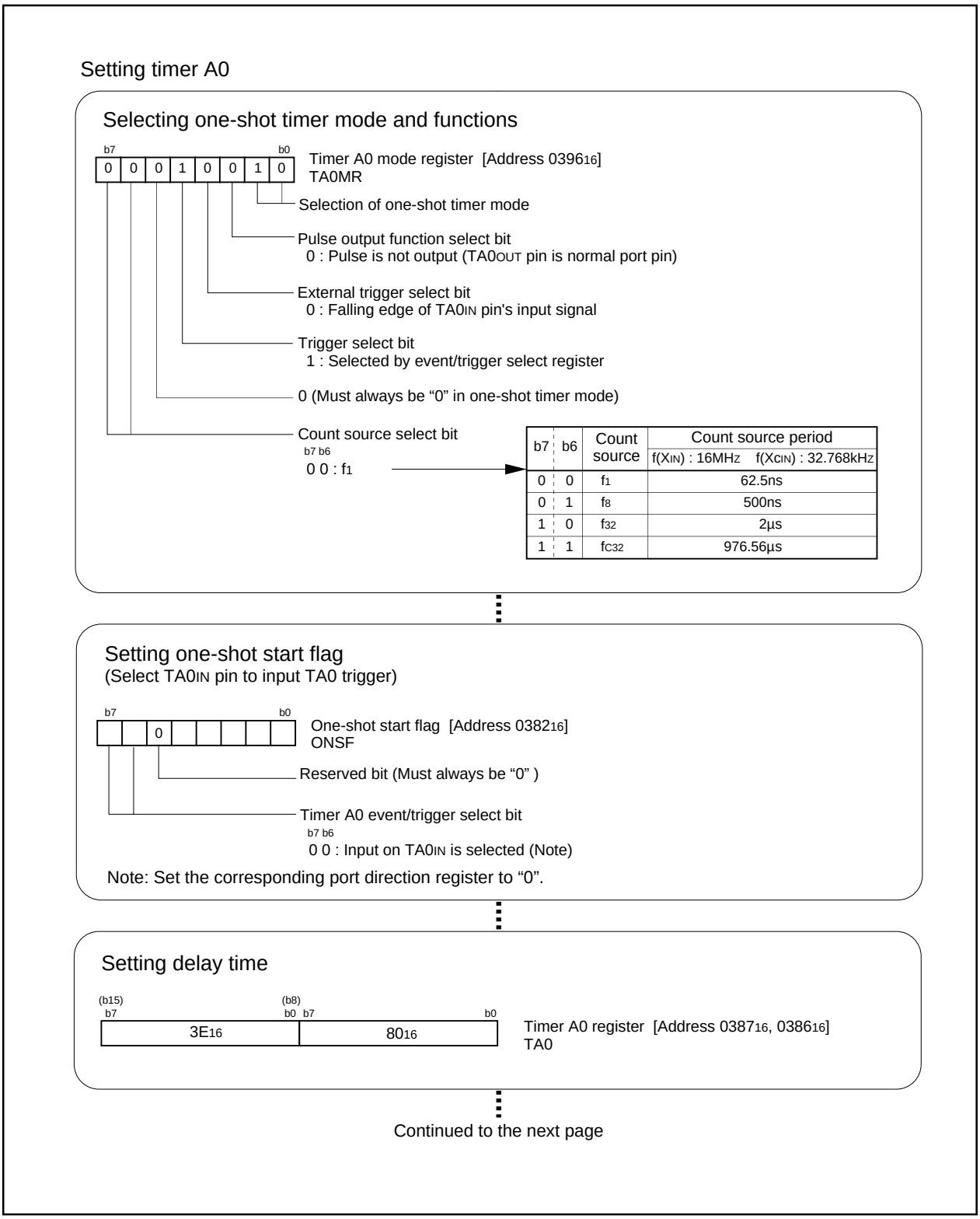


Figure 3.3.3. Set-up procedure of delayed one-shot output (1)

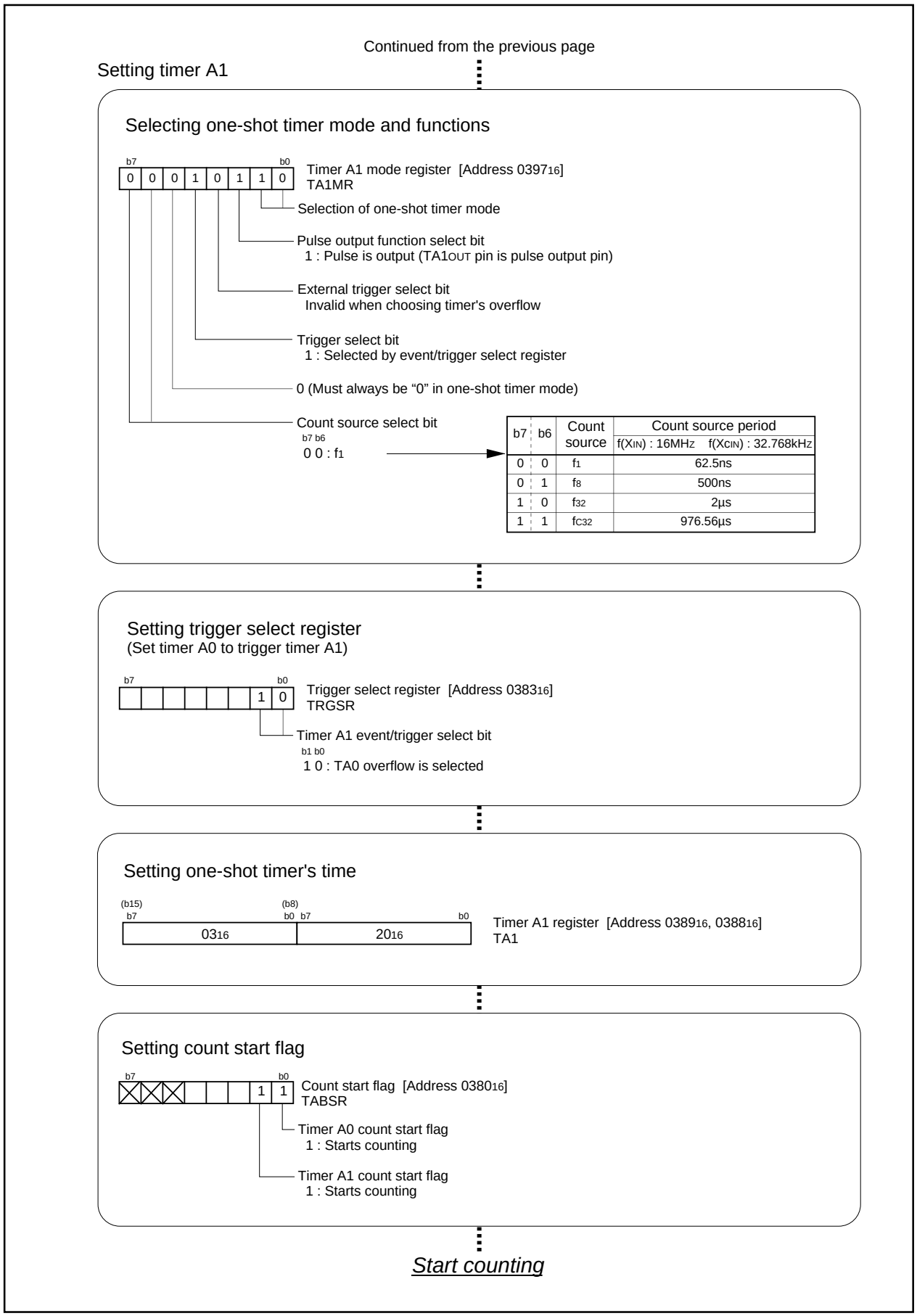


Figure 3.3.4. Set-up procedure of delayed one-shot output (2)

3.4 Buzzer Output

Overview The timer mode is used to make the buzzer ring. Figure 3.4.1 shows the operation timing, and Figure 3.4.2 shows the set-up procedure.

Use the following peripheral function:

• The pulse-outputting function in timer mode of timer A.

Specifications

(1) Sound a 2-kHz buzz beep by use of timer A0.

(2) Effect pull-up in the relevant port by use of a pull-up resistor. When the buzzer is off, set the port high-impedance, and stabilize the potential resulting from pulling up.

(3) Connect a 16-MHz oscillator to XIN.

Operation (1) The microcomputer begins performing a count on timer A0. Timer A0 has disabled interrupts.

(2) The microcomputer begins pulse output by setting the pulse output function select bit to "Pulse output effected". P70 changes into TA0OUT pin and outputs 2-kHz pulses.

(3) The microcomputer stops outputting pulses by setting the pulse output function select bit to "Pulse output not effected". P70 goes to an input pin, and the output from the pin becomes high-impedance.

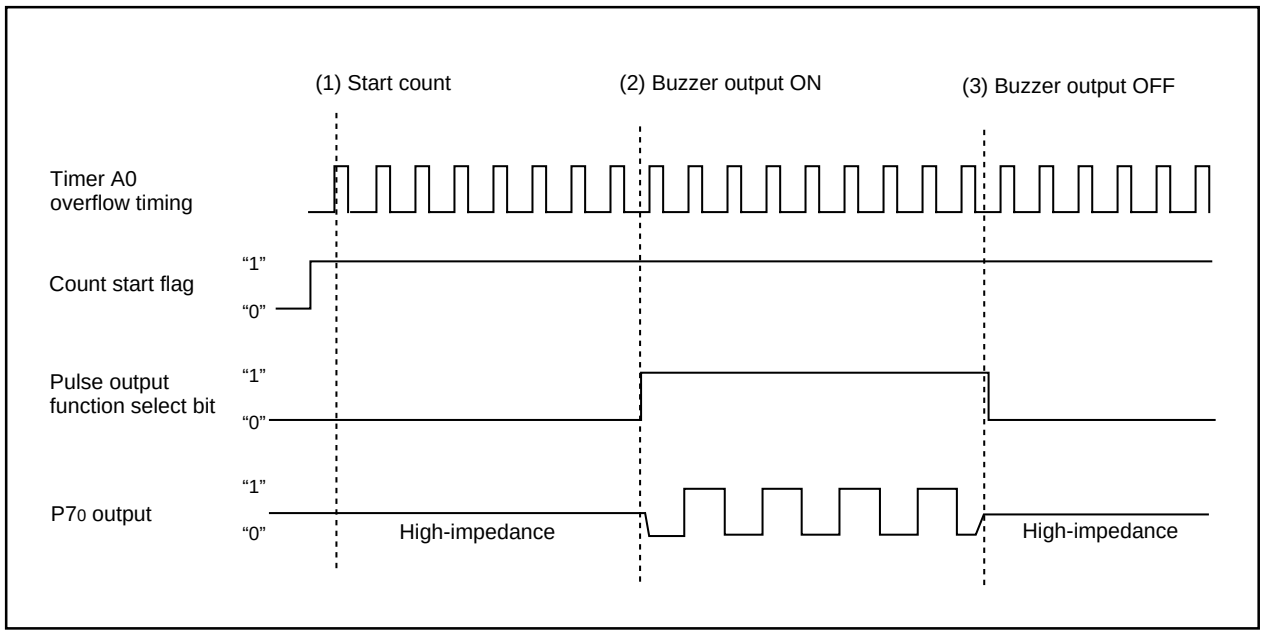


Figure 3.4.1. Operation timing of buzzer output

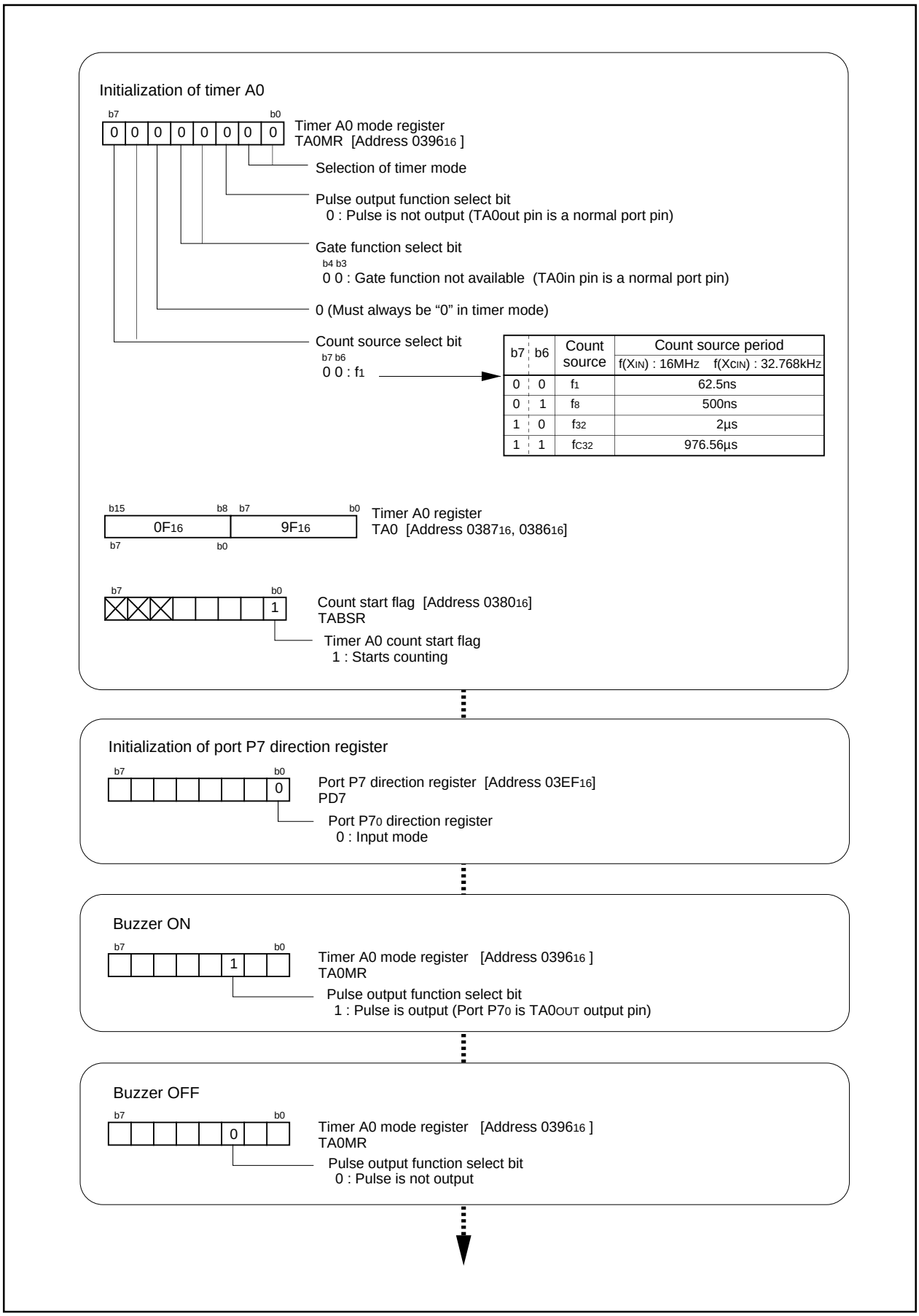


Figure 3.4.2. Set-up procedure of buzzer output

3.5 Solution for External Interrupt Pins Shortage

Overview The following are solution for external interrupt pins shortage. Figure 3.5.1 shows the set-up procedure.

Use the following peripheral function:

• Event counter mode of timer A

Specifications

(1) Inputting a falling edge to the TA0IN pin generates a timer A0 interrupt.

Operation (1) Set timer A0 to event counter mode, set timer to “0”, and set interrupt priority levels in timer A0.

(2) Inputting a falling edge to the TA0IN pin generates a timer A0 interrupt.

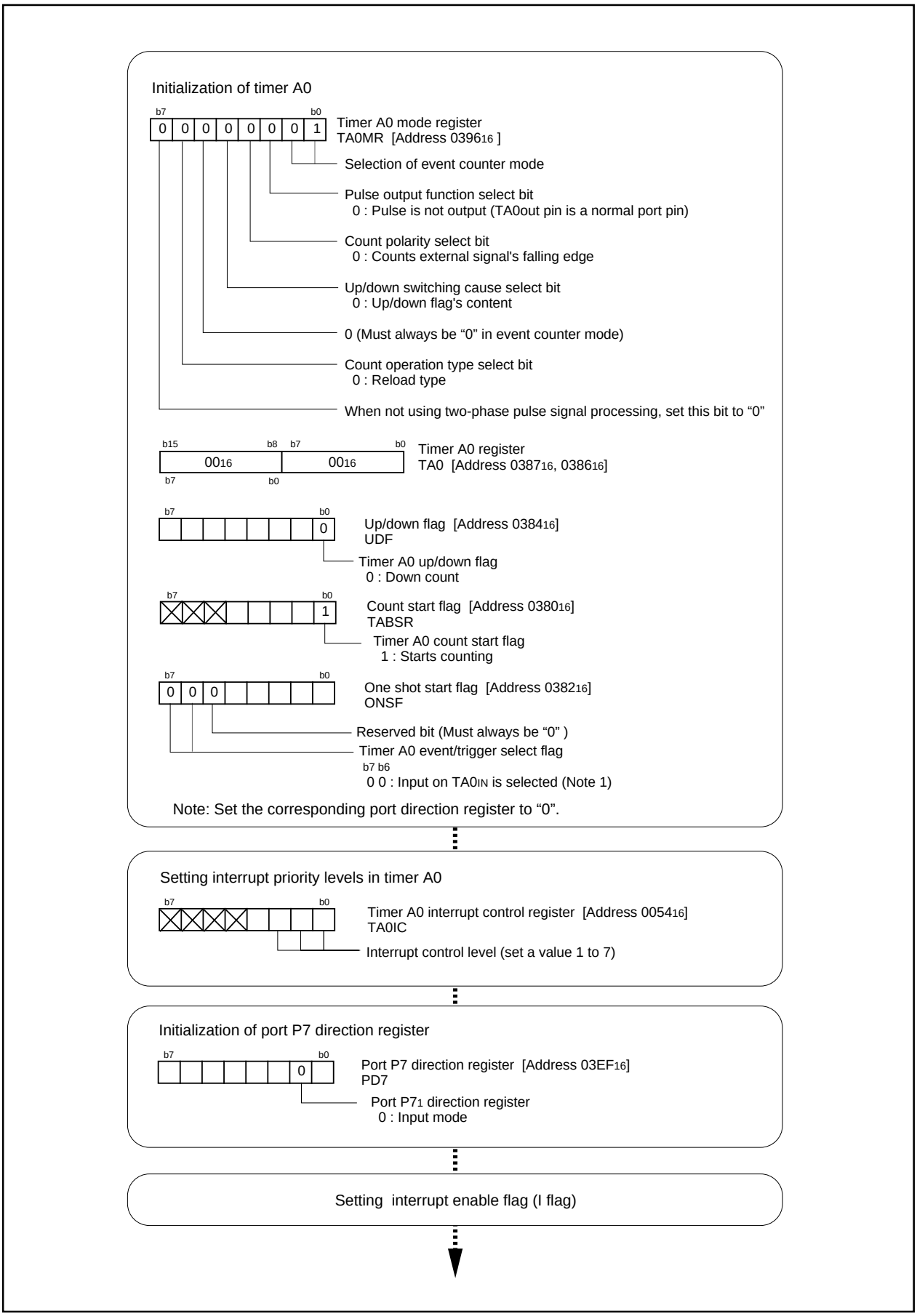


Figure 3.5.1. Set-up procedure of solution for a shortage of external interrupt pins

3.6 Memory to Memory DMA Transfer

Overview The following are steps for changing both source address and destination address to transfer data from memory to another. The DMA transfer utilizes the workings that assign a higher priority to the DMA0 transfer if transfer requests simultaneously occur in two DMA channels. Figure 3.6.1 shows the operation timing, Figure 3.6.2 shows the block diagram, and Figures 3.6.3 and 3.6.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Two DMAC channels
- One-byte temporary RAM (address 0800_{16})

Specifications

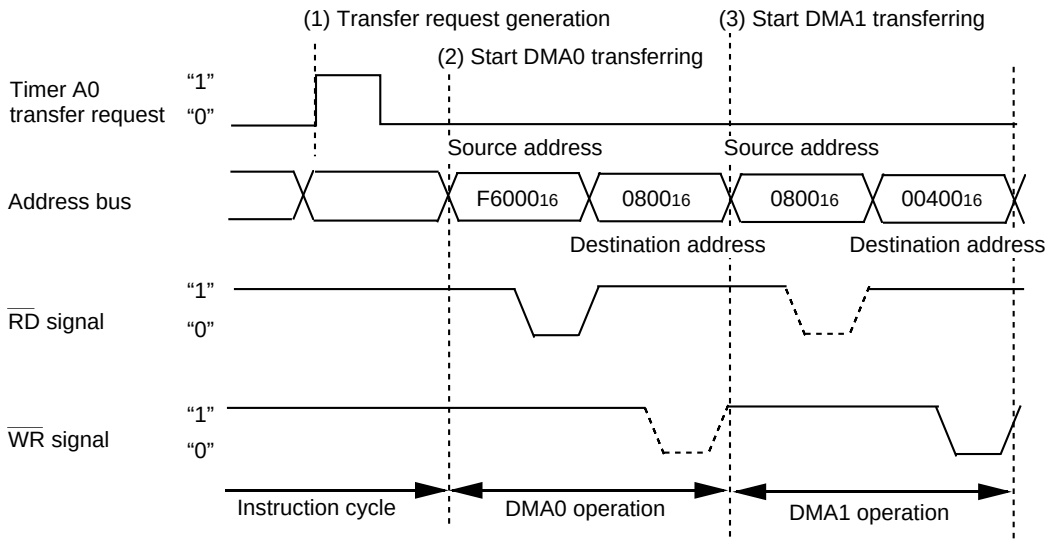
(1) Transfer the content of memory extending over 128 bytes from address $F6000_{16}$ to a 128-byte area starting from address 00400_{16}. Transfer the content every time a timer A0 interrupt request occurs.

(2) Use DMA0 for a transfer from the source to built-in memory, and DMA1 for a transfer from built-in memory to the destination.

Operation (1) A timer A interrupt request occurs. Though both a DMA0 transfer request and a DMA1 transfer request occur simultaneously, the former is executed first.

(2) DMA0 receives a transfer request and transfers data from the source to the built-in memory. At this time, the source address is incremented.

(3) Next, DMA1 receives a transfer request and transfers data involved from built-in memory to the destination. At this time, the destination address is incremented.



Note 1: The DMA0 operation and DMA1 operation are not necessarily executed in succession due to the a cycle steal operation.
Note 2: The instruction cycle varies from instruction to instruction.
Note 3: Since the parts of the $\overline{RD}$ and $\overline{WR}$ signals shown in short-dash lines vary in step with writing to the internal RAM, waveforms are not output to the $\overline{RD}$ and $\overline{WR}$ pins.

Figure 3.6.1. Operation timing of memory to memory DMA transfer

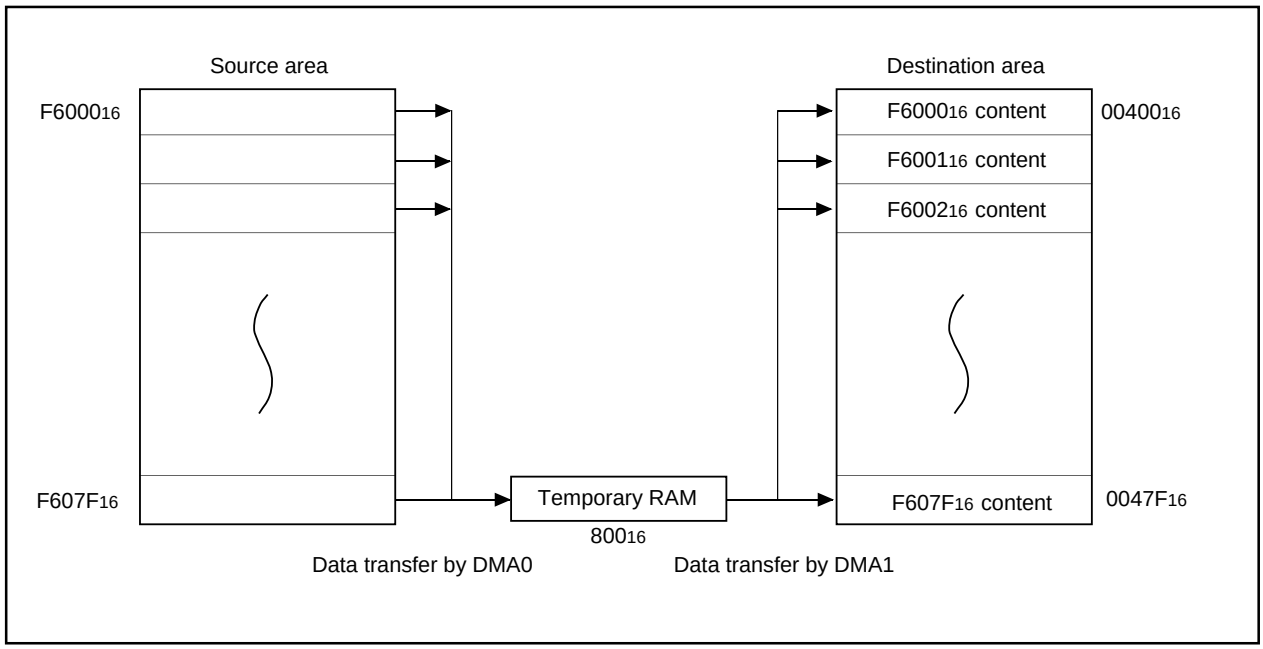


Figure 3.6.2. Block diagram of memory to memory DMA transfer

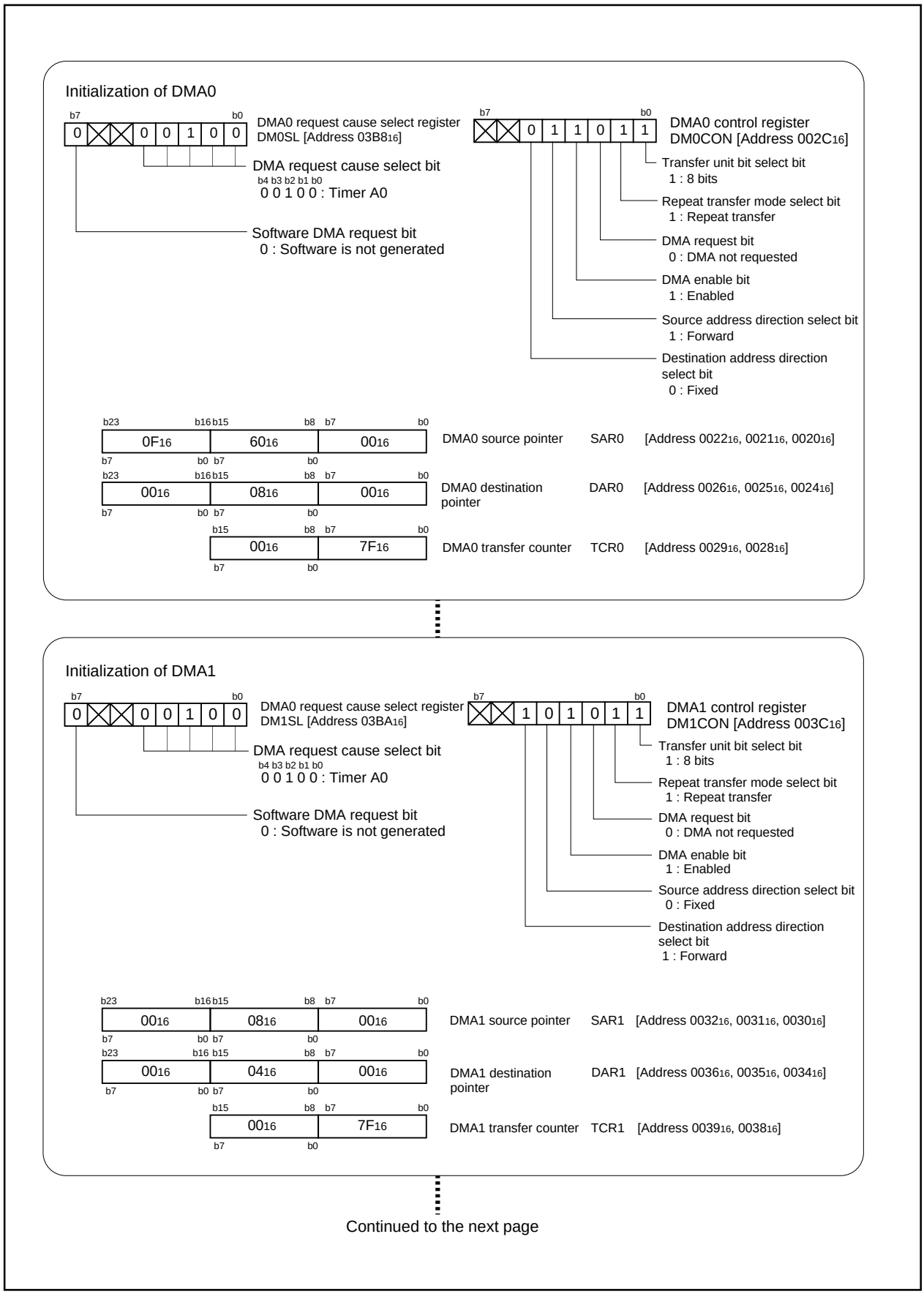


Figure 3.6.3. Set-up procedure of memory to memory DMA transfer (1)

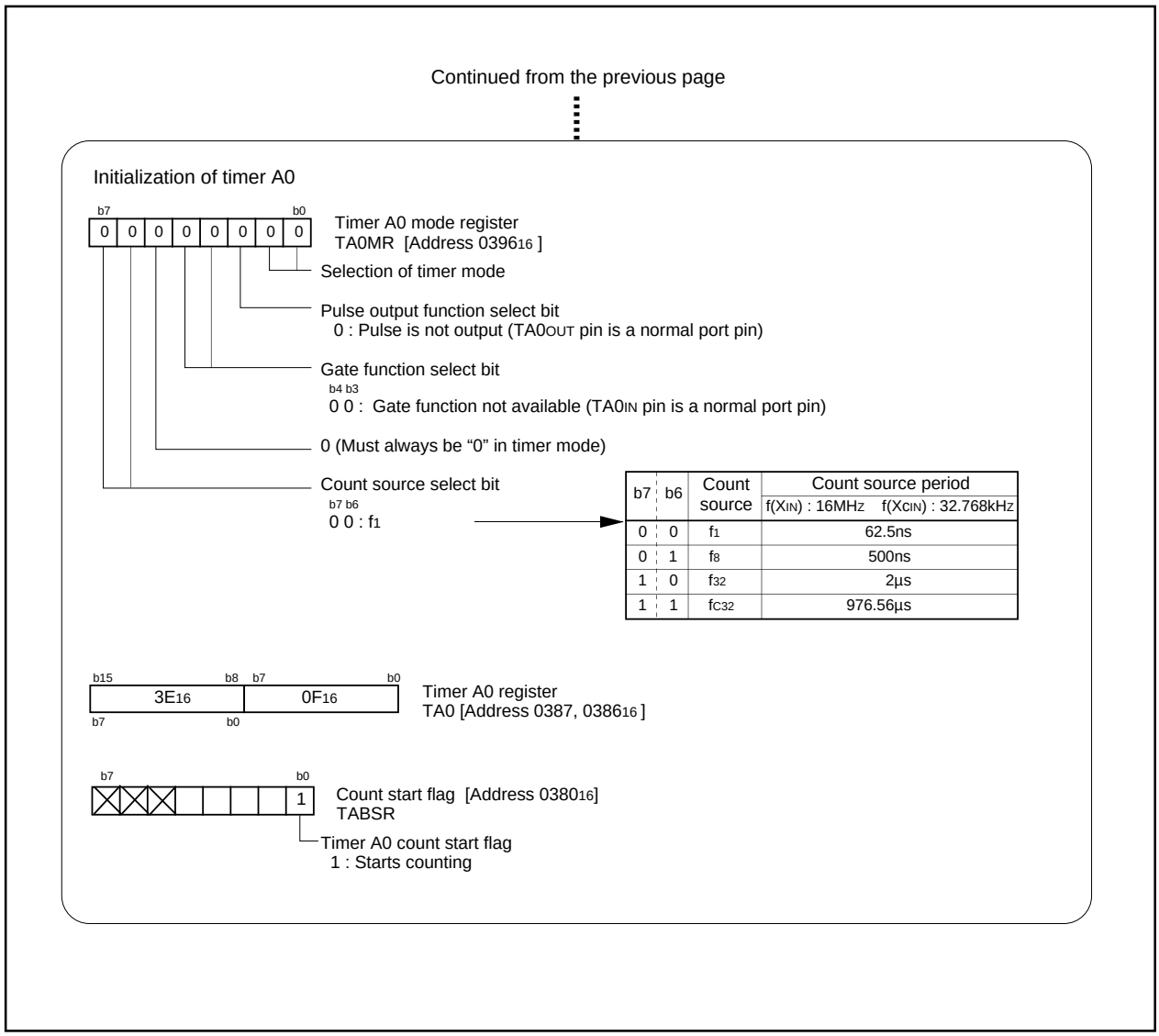


Figure 3.6.4. Set-up procedure of memory to memory DMA transfer (2)

3.7 CRC Calculation SFR Access Snoop Function in Clock Synchronous Serial Data Transmit

Overview The M30245 group, by use of DMAC, transfers data from the internal RAM to the UART1 and the result is transferred to the UART1 by use of SFR access snoop function. The block diagram is shown in Figure 3.7.1 and the setting routine is shown in Figure 3.7.2 to Figure 3.7.4.

The peripheral functions to be used are as follows:

- DMAC 1 Channel
- Internal RAM (address 00400_{16}) 512 bytes
- UART1 (Clock synchronous serial I/O mode)
- CRC calculation circuit
- SFR access snoop function

Specifications

(1) Data transfer is performed starting at address 00400_{16} from the area with 512 bytes to the UART1. Data are transferred from area between the address 00400_{16} and the 512nd byte to the UART1. Transfer is executed every time 1 byte of serial transmit is completed.

(2) Use the DMA0 to transfer data from the internal RAM to the UART1. Select the UART1 transmit to the DMA0 request factor. Select the single transfer mode and set the DMA0 transfer counter to 511 bytes (512-1).

(3) Set the CRC calculation circuit to the CRC-CCITT and set CRC snoop address register to the address of UART1 transmit buffer register (write snoop).

(4) On completing the DMA, 2-byte data of CRC data register (calculation result) are transferred to the UART1 and operation is completed.

Operation

(1) Initialize the UART1 related registers.

(2) Initialize the DMA0 related registers in DMA disable state.

(3) Set the DMA0 transfer counter to the transfer data consisting of 511 bytes (in this case, 8-bit transfer).

(4) Initialize the CRC calculation circuit and the SFR access snoop function.

(5) Set the software DMA request bit of DMA0 to “1”. At this time, 1st byte data are transferred from RAM to the transmit buffer of the UART1. Simultaneously, the transfer source address is incremented and the content of the transfer counter is down-counted. The transferred data are automatically written in CRC input register by the SFR access snoop function.

(6) When the transmit buffer of the UART1 becomes writable state, the DMA transfer request is occurred by the UART1. At this time, the next data are transferred from RAM to the transmit buffer of the UART1. Simultaneously, the transfer source address is incremented and the content of the transfer counter is down-counted. The transferred data are automatically written in CRC input register by the SFR access snoop function.

(7) As a result of repetition of the above (6), when the DMA0 transfer counter underflow, DMA enable bit is set to “0” to complete the DMA0 transfer. Simultaneously, the DMA0 interrupt request occurs. When the DMA0 interrupt request is detected, CRC data register (2 bytes) is read, it is transferred to the UART1 transmit buffer sequentially.

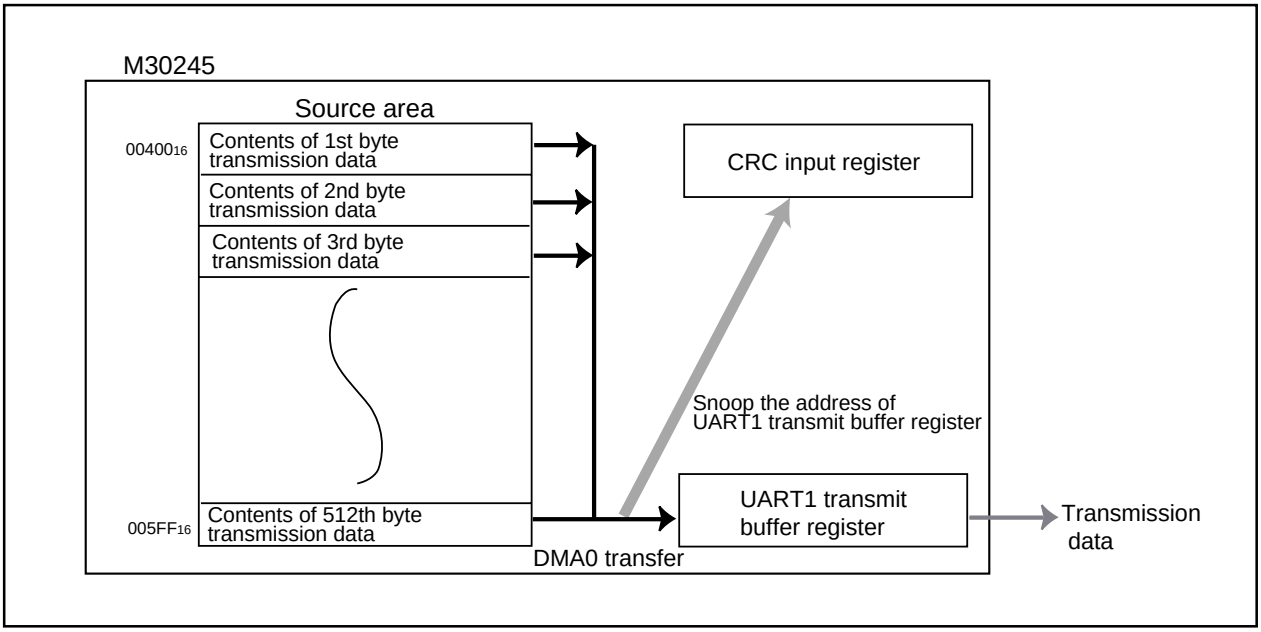


Figure 3.7.1. Block diagram of DMA transfer from RAM to UART and SFR snooping function

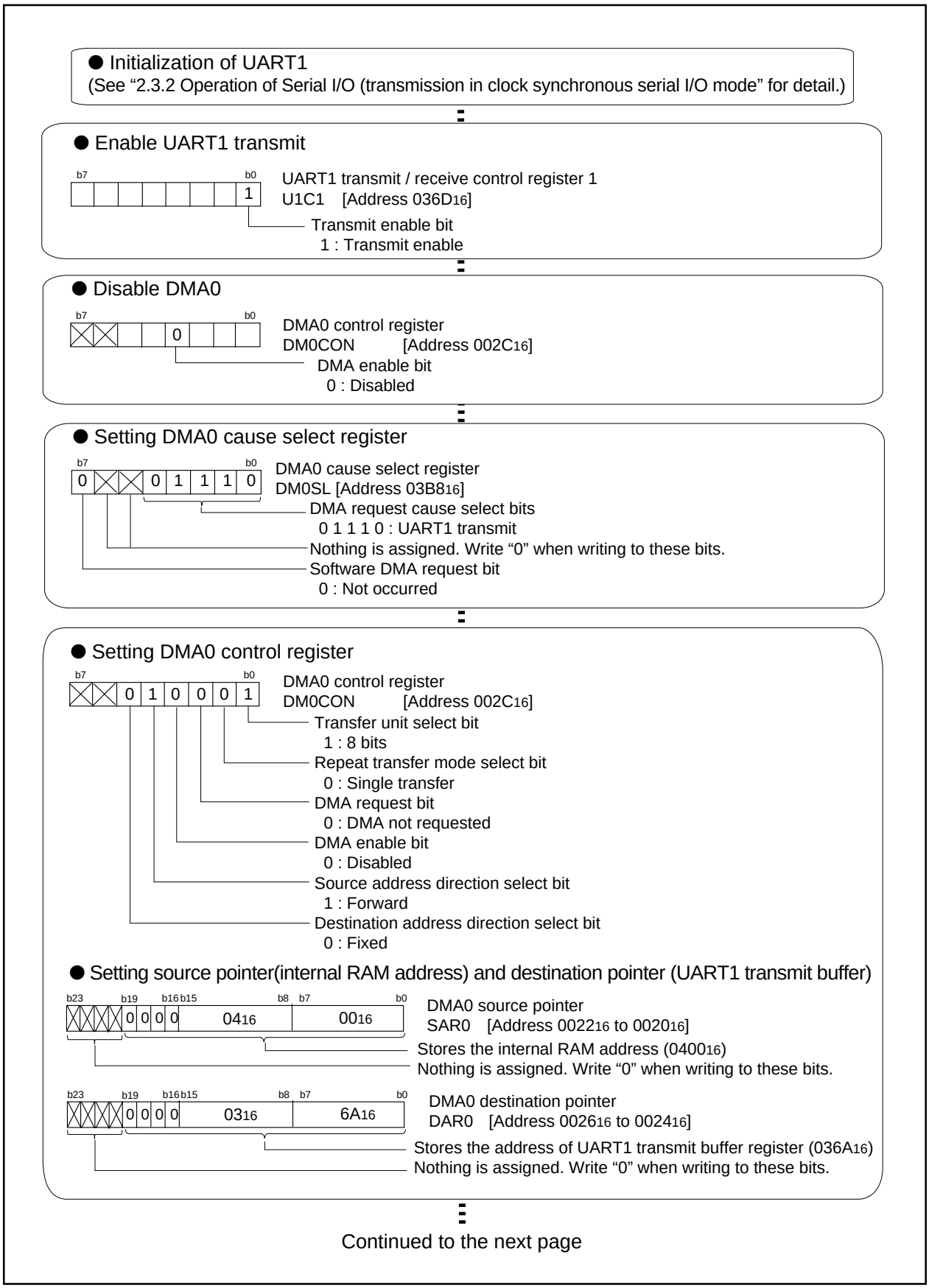


Figure 3.7.2. Setting routine (1) of DMA transfer from RAM to UART using SFR snooping function

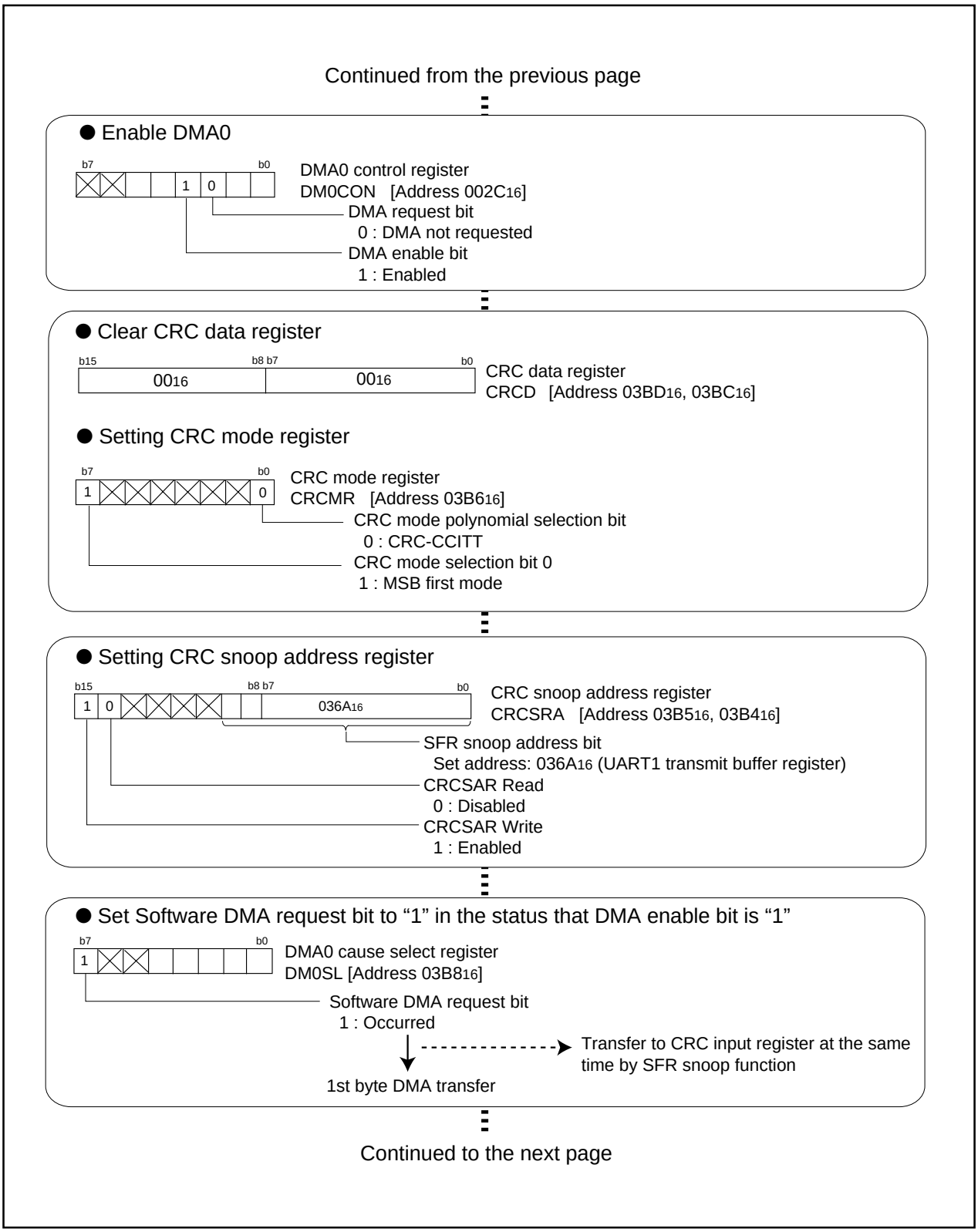


Figure 3.7.3. Setting routine (2) of DMA transfer from RAM to UART using SFR snooping function

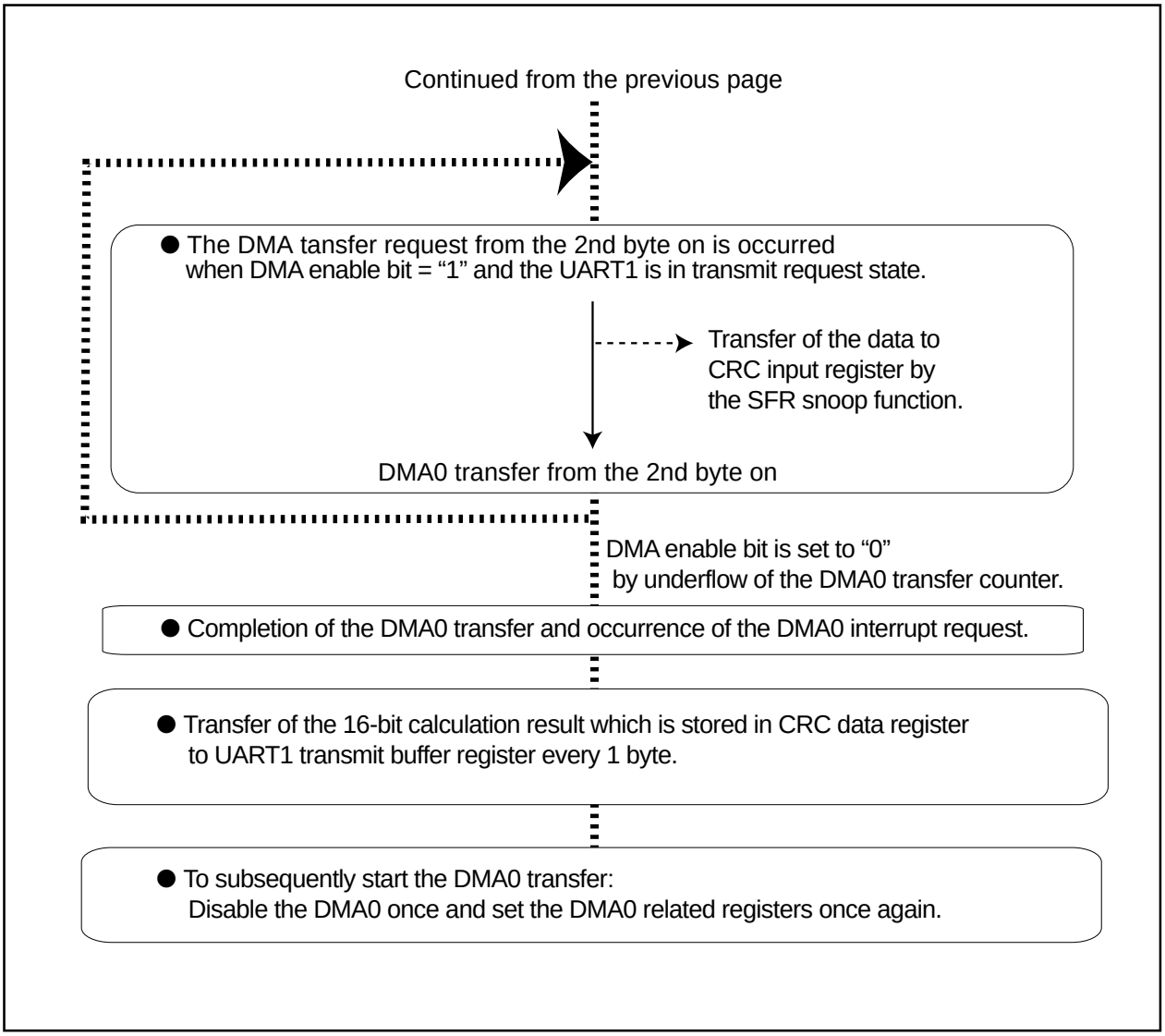


Figure 3.7.4. Setting routine (3) of DMA transfer from RAM to UART using SFR snooping function

3.8 Transfer from USB FIFO to Serial Sound Interface

Overview The M30245 group, by use of DMAC, transfers data from the USB endpoint 1 OUT FIFO to SS interface 1 transmit buffer register and fetches one packet data.

The block diagram is shown in Figure 3.8.1 and the setting routine is shown in Figure 3.8.2 to Figure 3.8.4.

The peripheral functions to be used are as follows:

- DMAC 1 channel
- USB endpoint 1 OUT (Receive)
- Serial sound interface 1

Specifications

(1) Receive packet data of the endpoint 1 OUT FIFO are transferred to SS interface 1 transmit buffer register. Transfer is executed every time the DMA transfer factor of the serial sound interface 1 occurs.

(2) Use the DMA0 to transfer data from the endpoint 1 OUT FIFO to SS interface 1 transmit buffer register. Select the serial sound interface 1 transmit to the DMA0 request factor. Select the single transfer mode and set the DMA0 transfer counter to 1/2 × (the data count of one packet received with endpoint 1 OUT) –1.

(3) Set the endpoint 1 OUT maximum packet size to 288 bytes (when sampling 48KHz/ 24-bit/ stereo) and disable the AUTO_CLR function. The data count of receive packet of endpoint 1 (endpoint 1 OUT write count register) is set to 288 bytes. Endpoint 1 OUT is used in isochronous transfer.

(4) On completing the DMA0 transfer, fetch of one packet data from the endpoint 1 OUT FIFO is completed by setting CLR_OUT_BUF_RDY bit of endpoint 1 to “1”.

Operation

(1) Initialize the DMA0 related registers in the state which DMA is disabled and USB DMA0 request register is not selected (in this case, 16-bit transfer).

(2) When the OUT_BUF_STS1 flag of endpoint 1 is set to “1” and packet data receive has been detected, set the DMA0 transfer counter to the 1/2 × (the data count of receive one packet) –1 (in this application example, 143 value is set).

(3) Set DMA enable bit of DMA0CON to “1” (DMA0 is enabled). Then, the DMA0 transfer request from the serial sound interface occurs.

(4) When the transfer request is received, the DMA0 transfers the 1st word (16-bit) data from the endpoint 1 OUT FIFO to the serial sound interface 1. Simultaneously, the content of the transfer counter is down-counted. Then, the DMA0 transfer request from the serial sound interface occurs .

(5) As a result of repetition of the above (4), when the DMA0 transfer counter underflow, DMA enable bit is set to “0” to complete the DMA0 transfer. Simultaneously, the DMA0 interrupt request occurs. When the DMA0 interrupt request is detected, set CLR_OUT_BUF_RDY bit of endpoint 1 OUT to “1”.

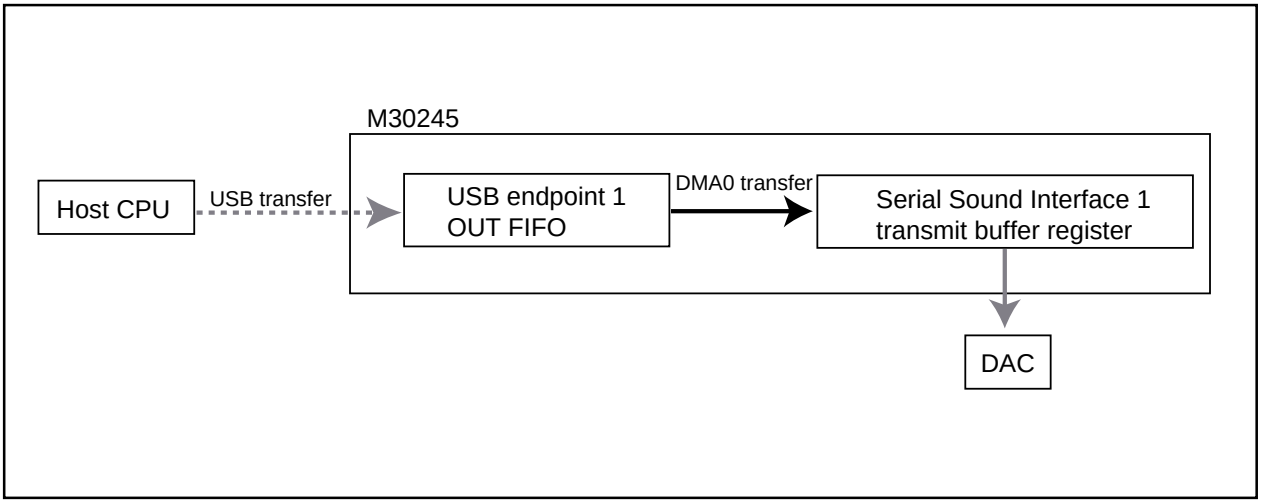


Figure 3.8.1. Block diagram of DMA transfer from USB FIFO to serial sound interface

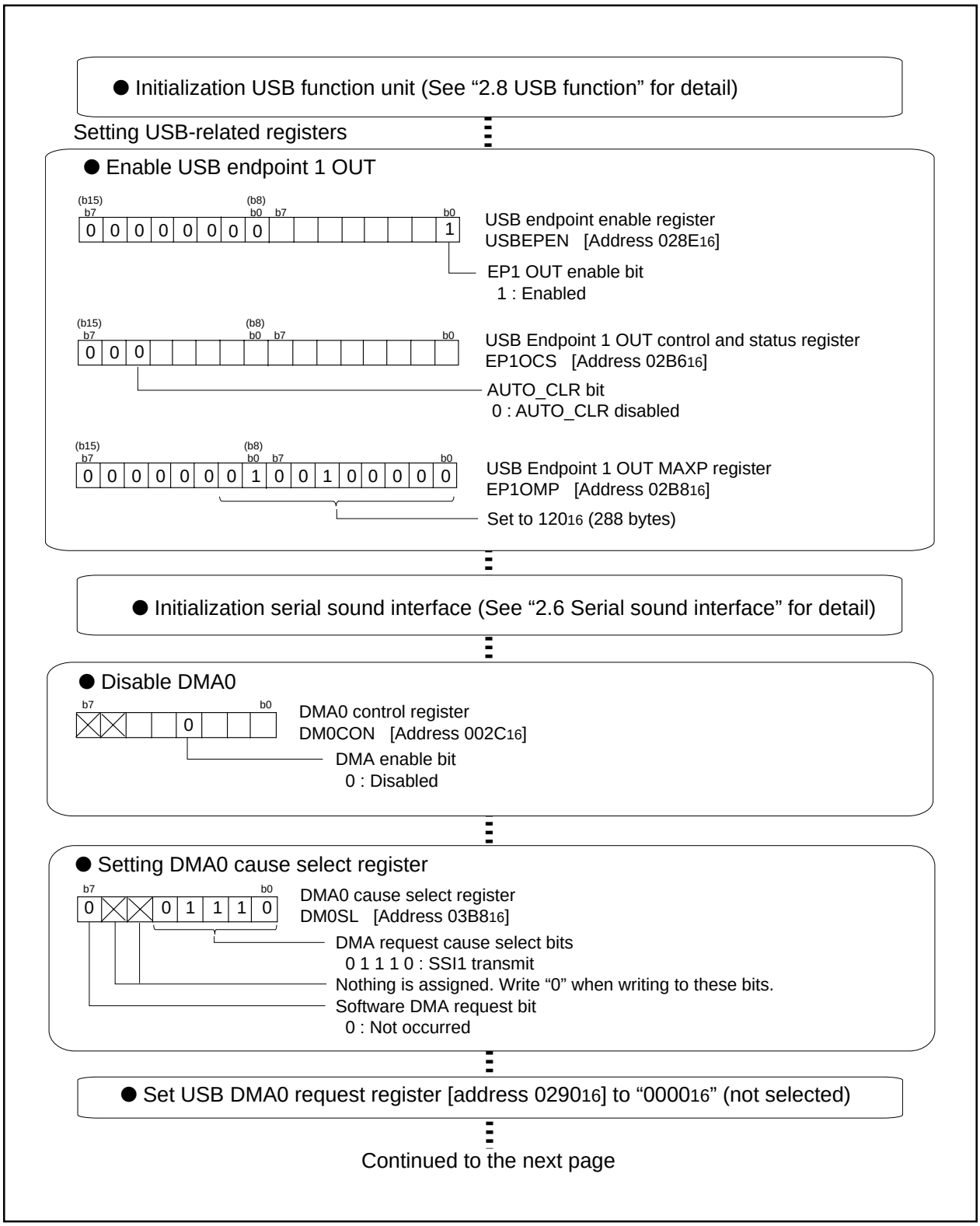
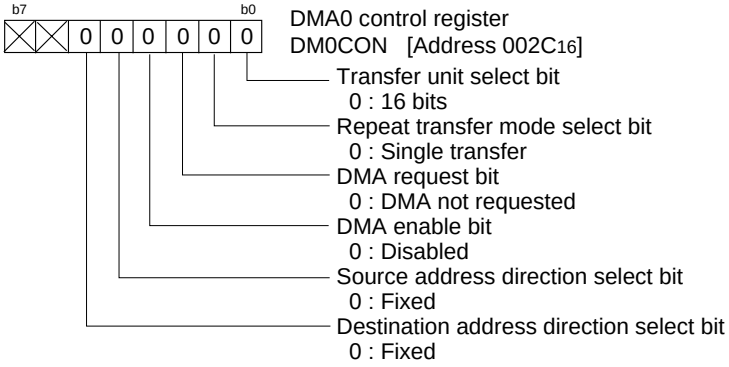


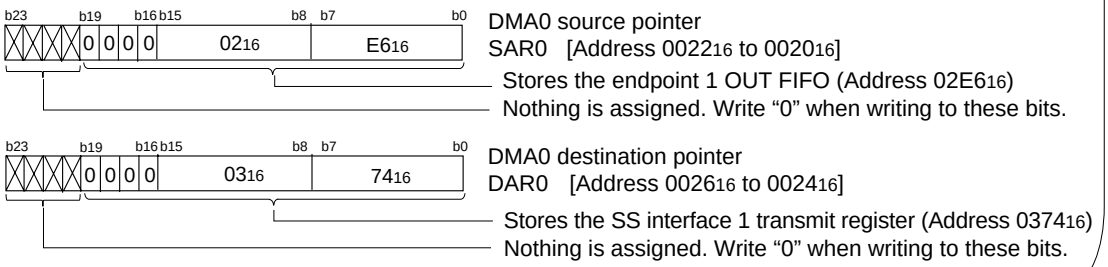
Figure 3.8.2. Setting routine (1/3) of DMA transfer from USB OUT FIFO to serial sound interface

Continued from the previous page

● Setting DMA0 control register



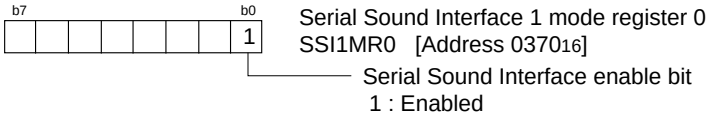
● Setting source pointer(endpoint 1 OUT FIFO data register) and destination pointer (SS interface 1 transmit buffer register)



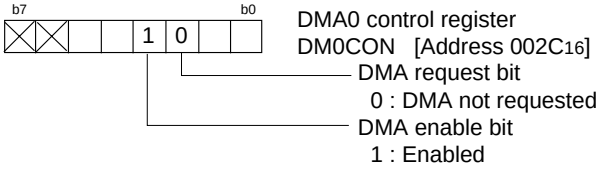
● Checking that OUT_BUF_STS1 flag is “1” and setting the number of the transfer bytes (Note)

Note: Set 1/2 × (the value of Endpoint 1 OUT write count register) –1.

● Enable serial sound interface 1



● Enable DMA0



Continued to the next page

Figure 3.8.3. Setting routine (2/3) of DMA transfer from USB OUT FIFO to serial sound interface

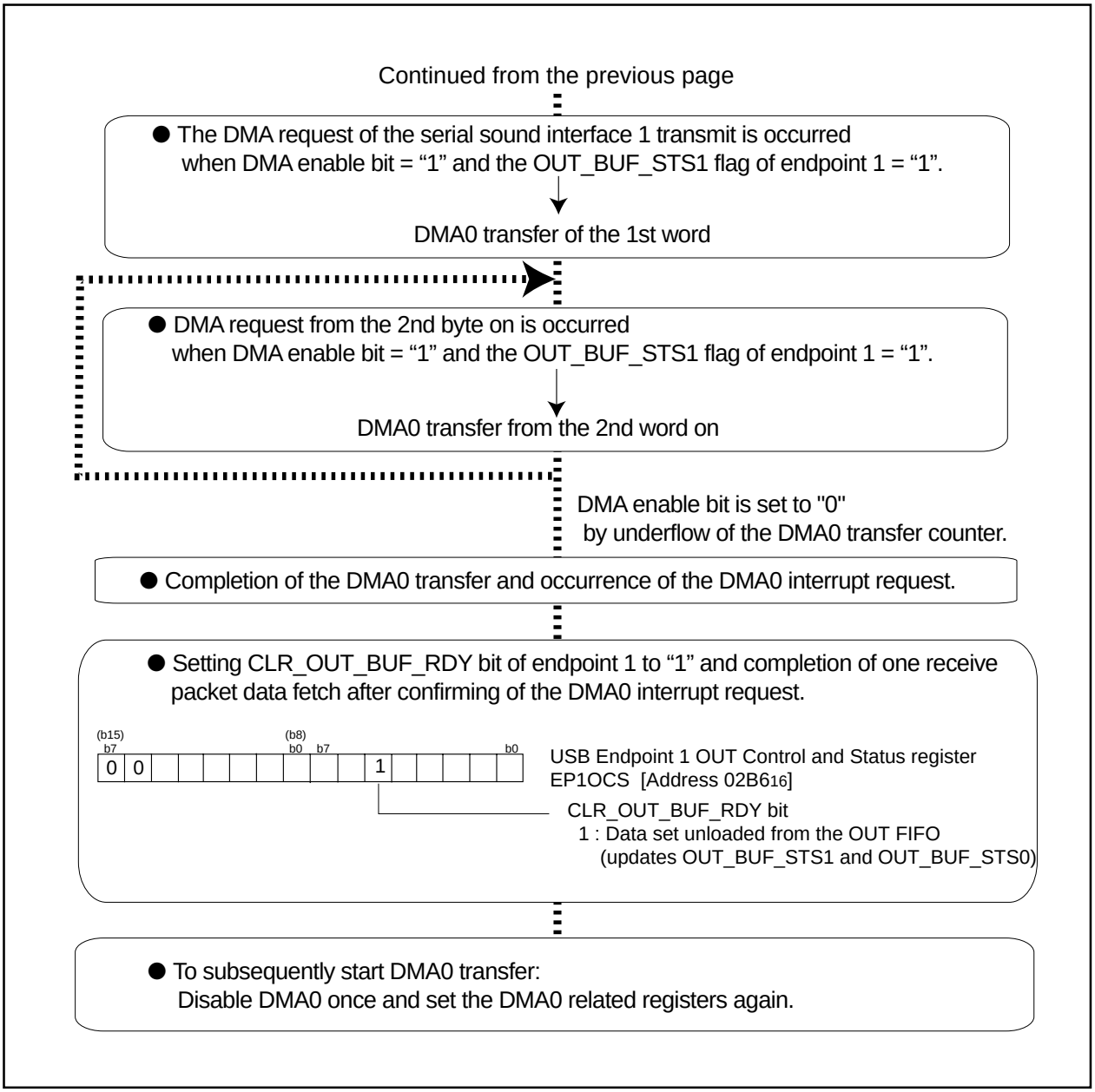


Figure 3.8.4. Setting routine (3/3) of DMA transfer from USB OUT FIFO to serial sound interface

3.9 Controlling Power Using Stop Mode

Overview The following are steps for controlling power using stop mode. Figure 3.9.1 shows the operation timing, Figure 3.9.2 shows an example of circuit, and Figures 3.9.3 and 3.9.4 show the set-up procedure.

Use the following peripheral functions:

- Key-input interrupts
- Stop mode
- Pull-up function

This example is not performed USB power control. Please refer section 2.7.4 for the power control of USB related.

Specifications

(1) Use $P0_0$ through $P0_3$ for the scan output pins of a key matrix. Use the input pins ($\overline{KI_0}$ through $\overline{KI_3}$) of the key-input interrupt function for the key-input reading pins. The pull-up function is also used.

(2) If a key-input interrupt request occurs, clear the stop mode and read a key.

Operation (1) Enable a key-input interrupt and set the pull-up function to pins $\overline{KI_0}$ through $\overline{KI_3}$. Change the output of $P0_0$ through $P0_3$ to "L" and enter stop mode.

(2) If a key is pressed, "L" is input to one of pins $\overline{KI_0}$ through $\overline{KI_3}$ to clear stop mode. A key-input interrupt occurs to execute the key-input interrupt handling routine.

(3) Sequentially set $P0_0$ through $P0_3$ to "L" to determine which key was pressed.

(4) When the process to determine the key pressed is completed, change the output from $P0_0$ through $P0_3$ to "L" again and enter stop mode.

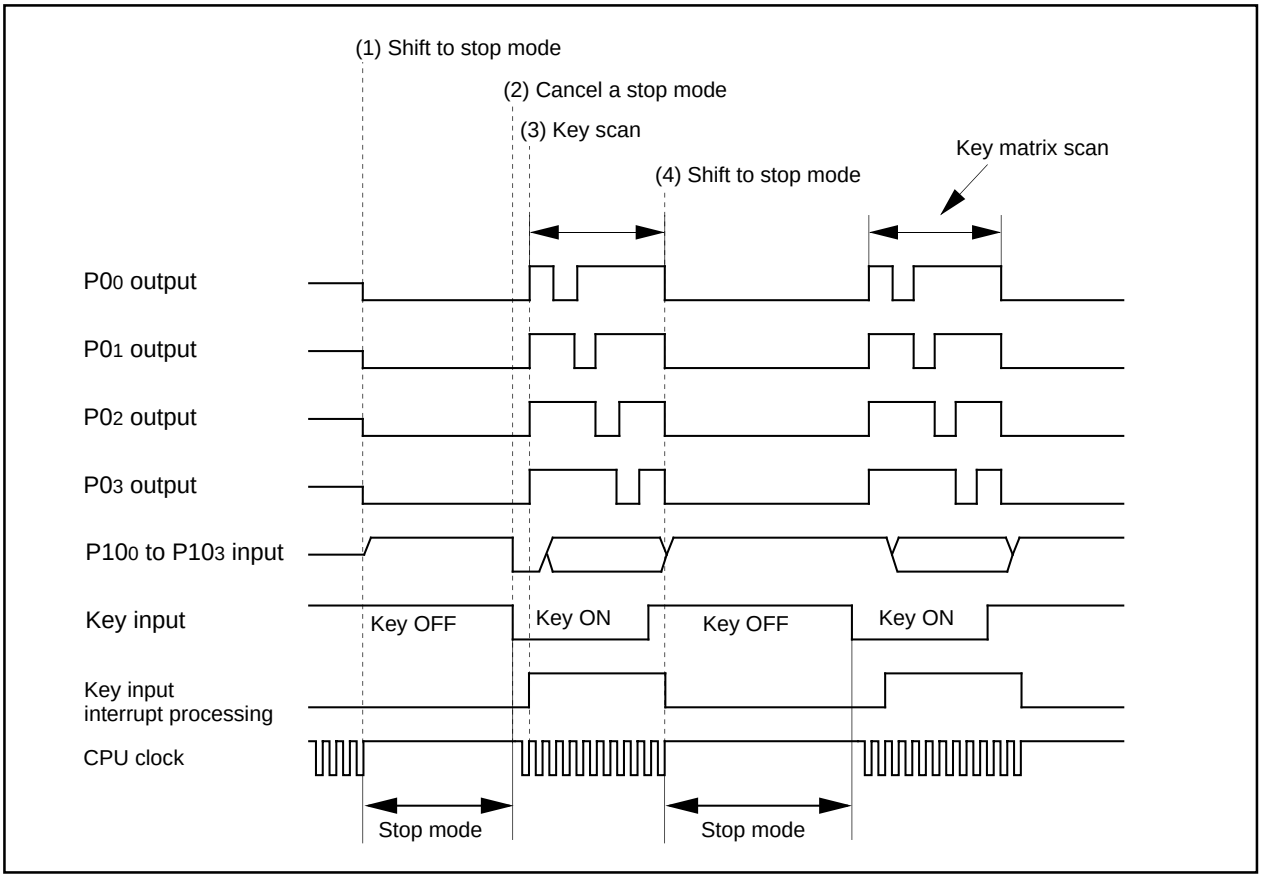


Figure 3.9.1. Operation timing of controlling power using stop mode

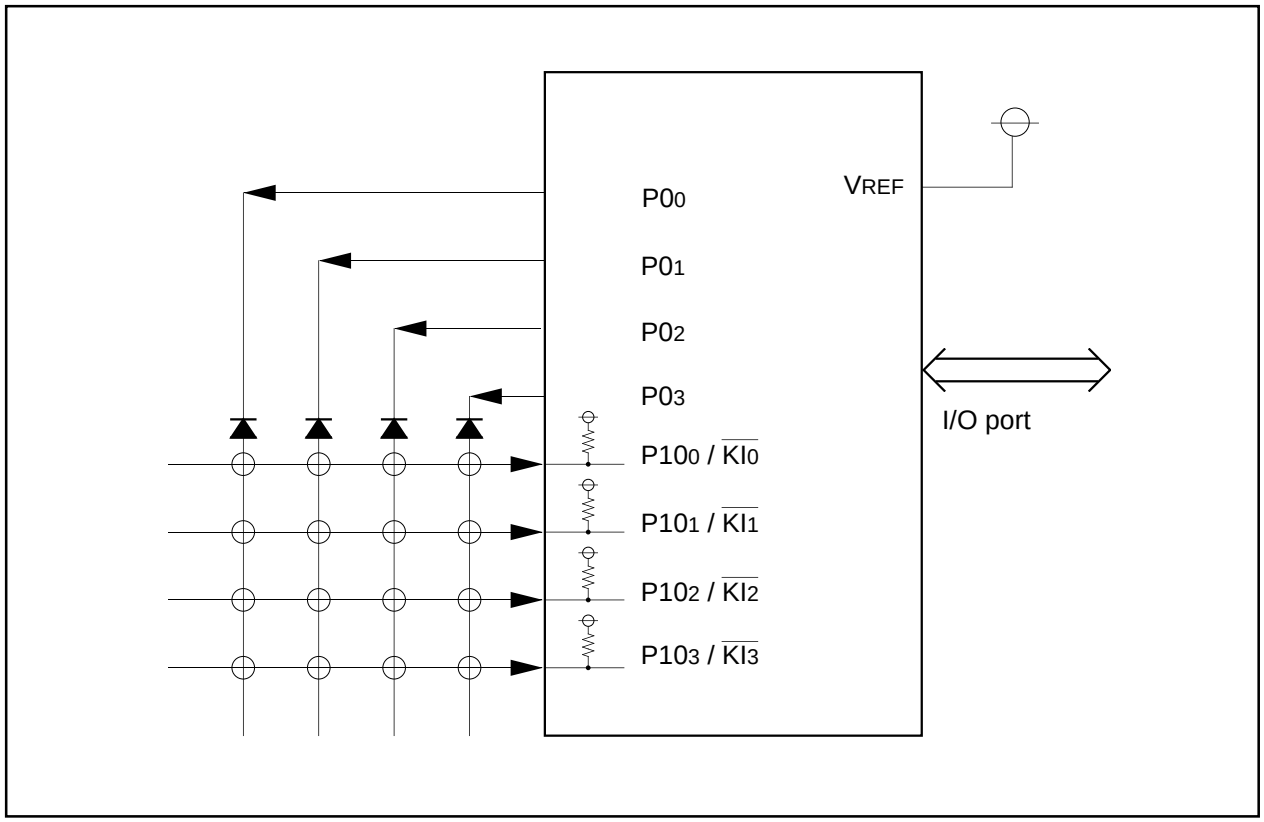
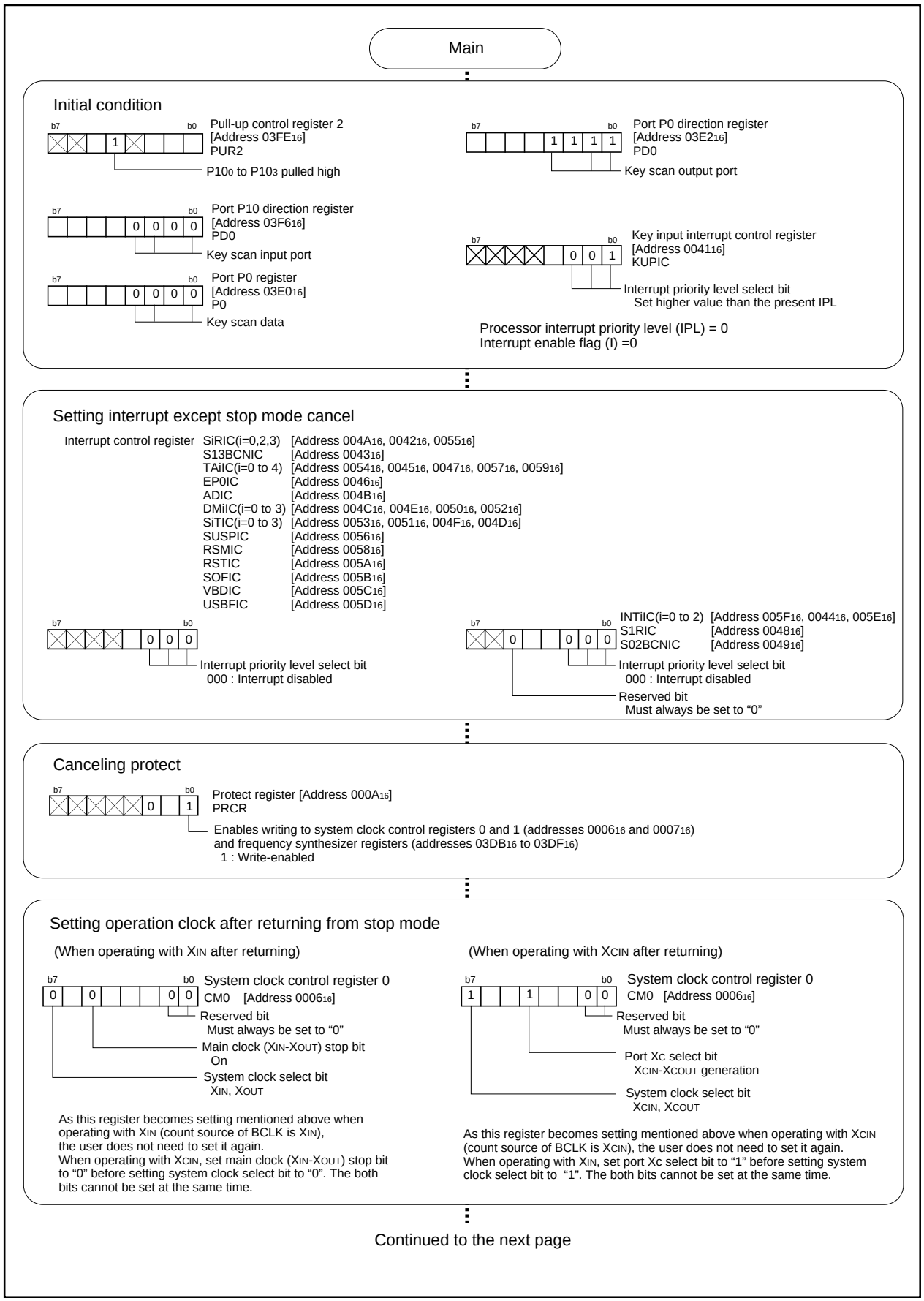


Figure 3.9.2. Example of circuit of controling power using stop mode



Continued to the next page

Figure 3.9.3. Set-up procedure of controlling power using stop mode (1)

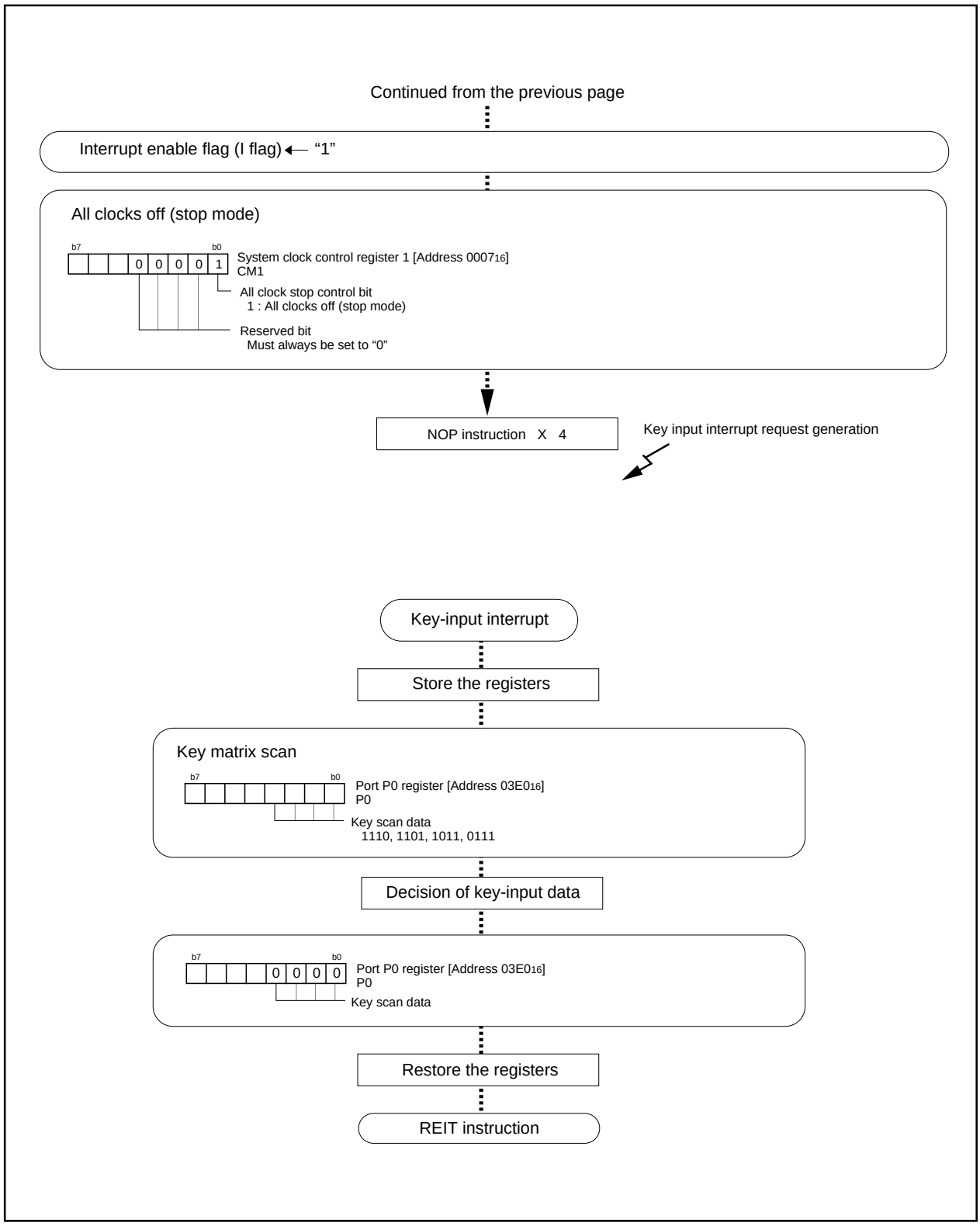


Figure 3.9.4. Set-up procedure of controlling power using stop mode (2)

3.10 Controlling Power Using Wait Mode

Overview The following are steps for controling power using wait mode. Figure 3.10.1 shows the operation timing, and Figures 3.10.2 to 3.10.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer A
- Wait mode

A flag named “F-WIT” is used in the set-up procedure. The purpose of this flag is to decide whether or not to clear wait mode. If F_WIT = “1” in the main program, the wait mode is entered; if F_WIT = “0”, the wait mode is cleared.

Specifications

(1) Connect a 32.768-kHz oscillator to XCIN to serve as the timer count source. As interrupts occur every one second, which is a count the timer reaches, the controller returns from wait mode and count the clock using a program.

(2) Clear wait mode if a $\overline{\text{INT0}}$ interrupt request occurs.

Operation (1) Switch the system clock from XIN to XCIN to get low-speed mode.

(2) Stop XIN and enter wait mode. In this instance, enable the timer A2 interrupt and the $\overline{\text{INT0}}$ interrupt.

(3) When a timer A2 interrupt request occurs (at 1-second intervals), start supplying the BCLK from XCIN. At this time, count the clock within the routine that handles the timer A2 interrupts and enter wait mode again.

(4) If a $\overline{\text{INT0}}$ interrupt occurs, start supplying the BCLK from XCIN. Start the XIN oscillation within the $\overline{\text{INT0}}$ interrupt, and switch the system clock to XIN.

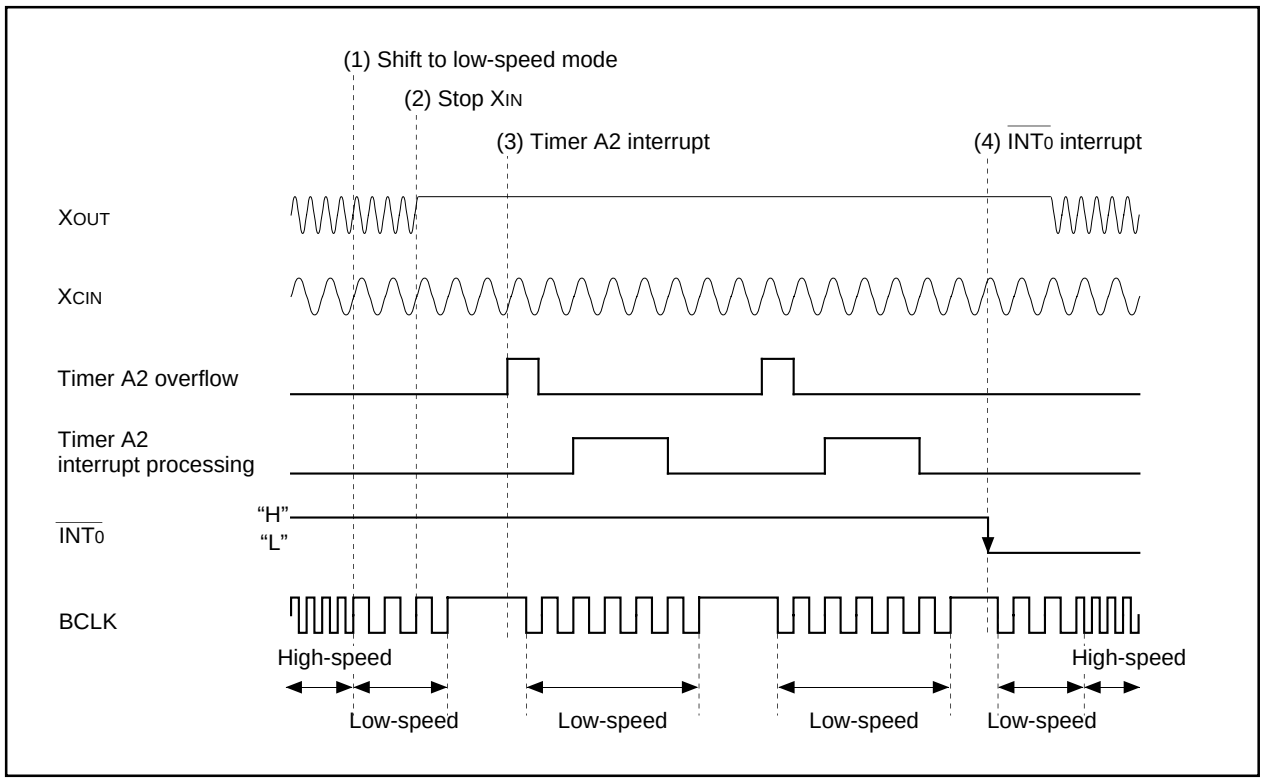


Figure 3.10.1. Operation timing of controling power using wait mode

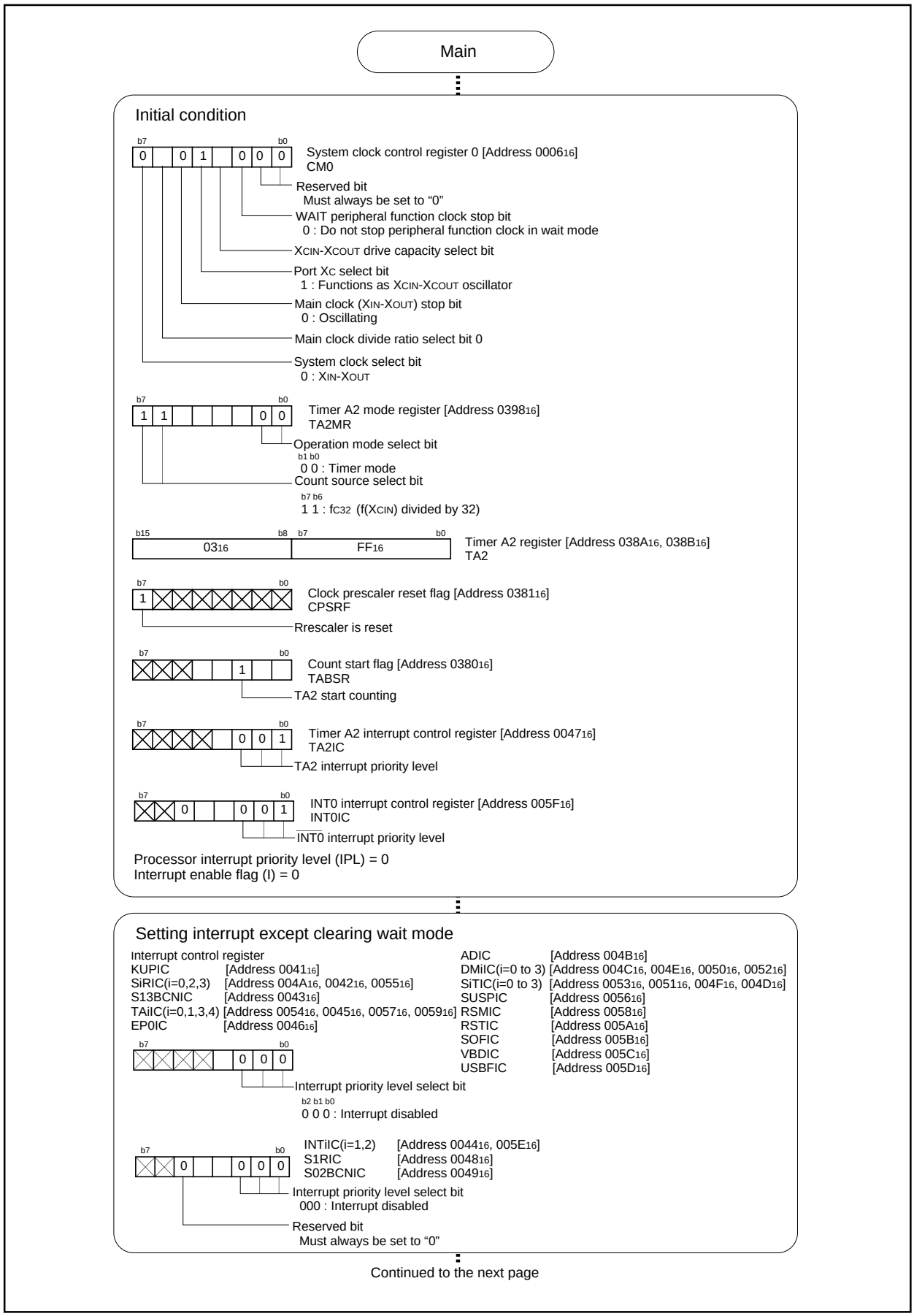


Figure 3.10.2. Set-up procedure of controlling power using wait mode (1)

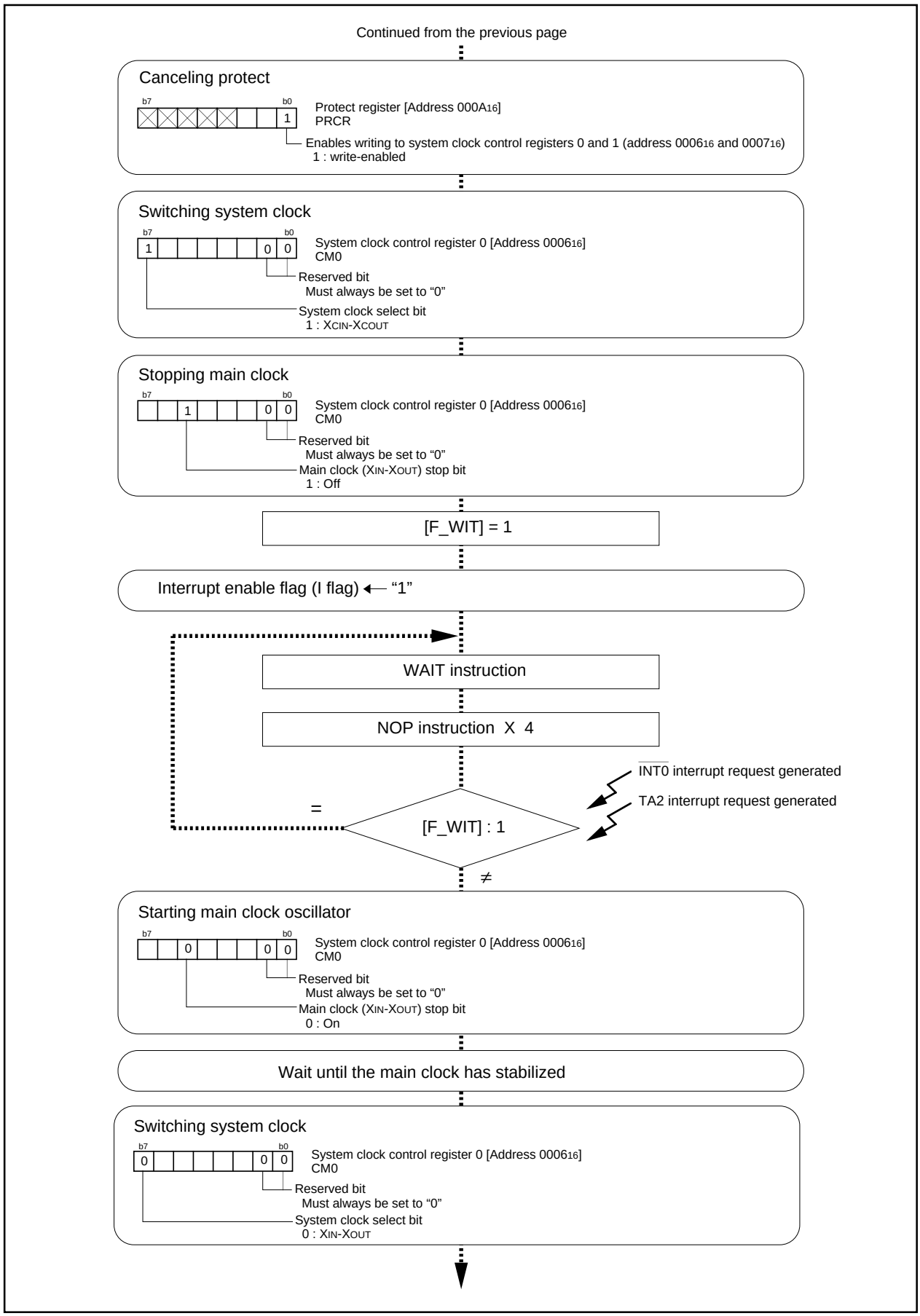


Figure 3.10.3. Set-up procedure of controlling power using wait mode (2)

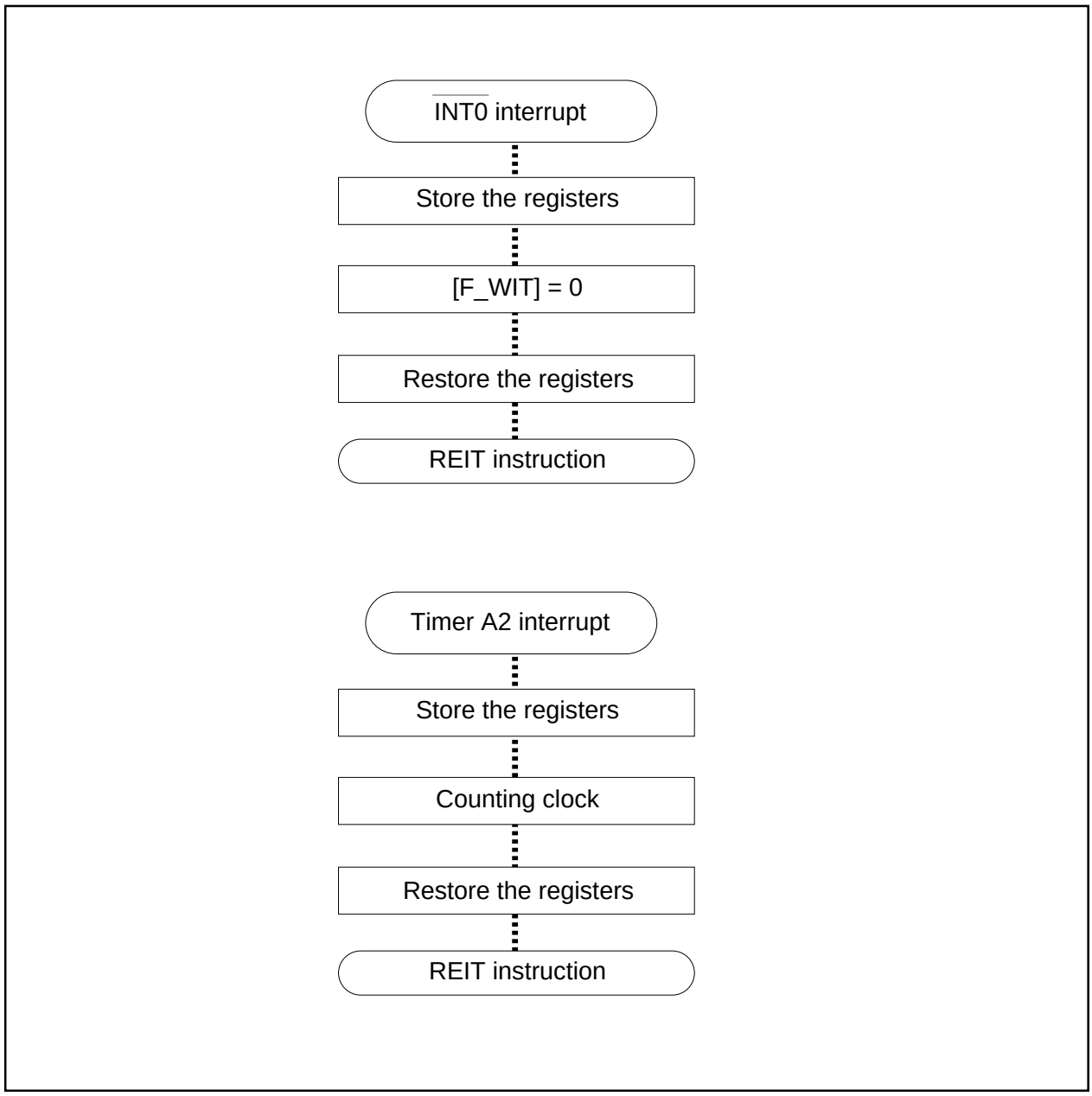


Figure 3.10.4. Set-up procedure of controlling power using wait mode (3)

MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
M30245 Group

Jan. First Edition 2003

Editioned by
Committee of editing of Mitsubishi Semiconductor User's Manual

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

User's Manual
M30245 Group

Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan

New publication, effective Jan. 2003.
Specifications subject to change without notice.