Mitsubishi single-chip microcomputer

M30201 Group

User's manual (tentative)

Specifications written in this user's manual are believed to be accurate, but are not guaranteed to be entirely free of error.
Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

Mitsubishi Electric Corporation Kitaitami Works
Mitsubishi Electric Semiconductor System Corporation
REV.A

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 non-flammable 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 or circuit application examples contained in these materials.

- All information contained in these materials, including product data, diagrams and charts, represent 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.

- 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.

Preface

This user's manual describes the function and features of the Mitsubishi M30201 CMOS 16-bit microcomputer. The software features are explained to help designers take full advantage of the M16C functions.
For details about the software, please refer to the “M16C/60, M16C/20 series software manual”, and for the development support tools, please refer to the related instruction manual.

How to Use This Manual

This user's manual is written for the M30201 group.

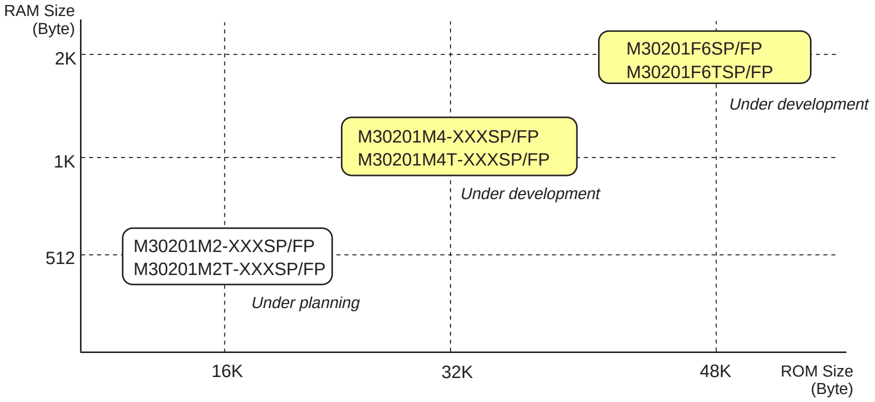
The reader of this manual is expected to have the basic knowledge of electric and logic circuits and microcomputers.

This manual is for the use of the models below.

- M30201M2-XXXSP/FP
- M30201M2T-XXXSP/FP
- M30201M4-XXXSP/FP
- M30201M4T-XXXSP/FP
- M30201F6SP/FP
- M30201F6TSP/FP

These products have similar features except for the memories, which differ from one product to another. This manual gives descriptions of M30201M4-XXXSP. Memories built-in are as shown below. Be careful when writing a program, as the memories have different capacities.

The figure of each register configuration describes its functions, contents at reset, and attributes as follows :



This manual comprises of eight chapters. Use the suggested chapters as a reference for the following topics:

- Bit attribute

R.....Read

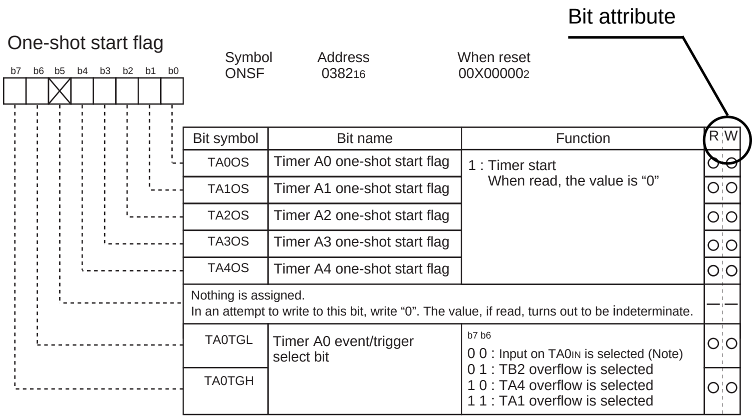
O.....Possible to read

X.....Impossible to read

W.....Write

O.....Possible to write

X.....Impossible to write



One-shot start flag

Symbol ONSF

Address 0382$_{16}$

When reset 00X00000$_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
TA3OS	Timer A3 one-shot start flag		○	○
TA4OS	Timer A4 one-shot start flag		○	○
Nothing is assigned. In an attempt to write to this bit, write “0”. The value, if read, turns out to be indeterminate.			—	—
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on TA0$_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

Note: Set the corresponding port direction register to “0”.

* To understand hardware specifications .... Chapter 1 Hardware

* To understand the basic way of using peripheral
features and the operation timing................................Chapter 2 Peripheral Functions Usage

* To observe applications of
peripheral features ....Chapter 3 Examples of Peripheral Functions Applications

* To understand interrupt timing in detail ....Chapter 4 Interrupts

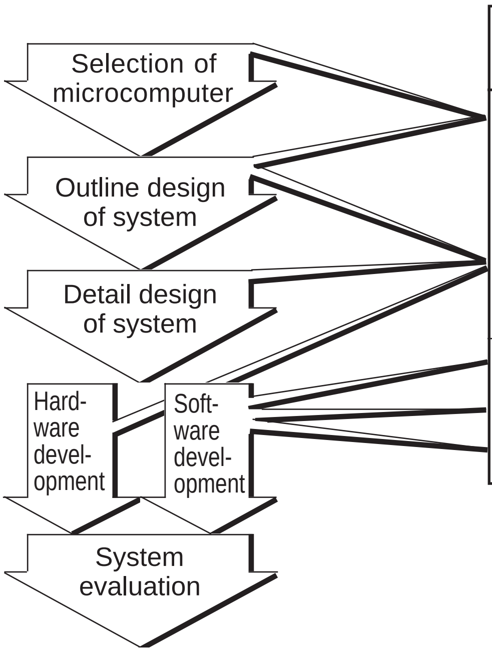
* To understand standard data..Chapter 5 Standard Characteristics

This manual includes a quick reference immediately following the Table of Contents, indicate the page of the topic to be pursued.

* To find a page describing a specific register
by the register address............................... Quick Reference to Pages Classified by Address

M16C Family-related document list

Usages
(Microcomputer development flow)



Type of document		Contents
Hardware	Data sheet and data book	Hardware specifications (pin assignment, memory map, specifications of peripheral functions, electrical characteristics, timing charts)
	User's manual	Detailed description about hardware specifications, operation, and application examples (connection with peripherals, relationship with software)
Software	Programming manual	Method for creating programs using assembly and C languages
	Software manual	Detailed description about operation of each instruction (assembly language)

M16C Family Line-up

M16C Family	M16C/80 Series	M16C/80 Group
	M16C/60 Series	M16C/60 Group M16C/61 Group M16C/62 Group
	M16C/20 Series	M16C/20 Group M16C/21 Group

Table of Contents

Chapter 1 Hardware

Chapter 2 Peripheral Functions Usage

Chapter 3 Examples of Peripheral functions Applications

Chapter 4 Interrupt

Chapter 5 Standard Characteristics

Quick Reference to Pages Classified by Address

Address	Register	Page
0000_{16}		
0001_{16}		
0002_{16}		
0003_{16}		
0004_{16}	Processor mode register 0 (PM0)	17
0005_{16}	Processor mode register 1(PM1)	
0006_{16}	System clock control register 0 (CM0)	2 2
0007_{16}	System clock control register 1 (CM1)	
0008_{16}		
0009_{16}	Address match interrupt enable register (AIER)	45
$000A_{16}$	Protect register (PRCR)	28
$000B_{16}$		
$000C_{16}$		
$000D_{16}$		
$000E_{16}$	Watchdog timer start register (WDTS)	48
$000F_{16}$	Watchdog timer control register (WDC)	
0010_{16}	Address match interrupt register 0 (RMAD0)	45
0011_{16}		
0012_{16}		
0013_{16}		
0014_{16}	Address match interrupt register 1 (RMAD1)	45
0015_{16}		
0016_{16}		
0017_{16}		
0018_{16}		
0019_{16}		
$001A_{16}$		
$001B_{16}$		
$001C_{16}$		
$001D_{16}$		
$001E_{16}$		
$001F_{16}$		
0020_{16}		
0021_{16}		
0022_{16}		
0023_{16}		
0024_{16}		
0025_{16}		
0026_{16}		
0027_{16}		
0028_{16}		
0029_{16}		
$002A_{16}$		
$002B_{16}$		
$002C_{16}$		
$002D_{16}$		
$002E_{16}$		
$002F_{16}$		
0030_{16}		
0031_{16}		
0032_{16}		
0033_{16}		
0034_{16}		
0035_{16}		
0036_{16}		
0037_{16}		
0038_{16}		
0039_{16}		
$003A_{16}$		
$003B_{16}$		
$003C_{16}$		
$003D_{16}$		
$003E_{16}$		

Address	Register	Page
0040_{16}		
0041_{16}		
0042_{16}		
0043_{16}		
0044_{16}		
0045_{16}		
0046_{16}		
0047_{16}		
0048_{16}		
0049_{16}		
$004A_{16}$		
$004B_{16}$		
$004C_{16}$		
$004D_{16}$	Key input interrupt control register (KUPIC)	35
$004E_{16}$	A-D conversion interrupt control register (ADIC)	
$004F_{16}$		
0050_{16}		
0051_{16}	UART0 transmit interrupt control register (S0TIC)	35
0052_{16}	UART0 receive interrupt control register (S0RIC)	
0053_{16}	UART1 transmit interrupt control register (S1TIC)	
0054_{16}	UART1 receive interrupt control register (S1RIC)	
0055_{16}	Timer A0 interrupt control register (TA0IC)	
0056_{16}	Timer X0 interrupt control register (TX0IC)	
0057_{16}	Timer X1 interrupt control register (TX1IC)	
0058_{16}	Timer X2 interrupt control register (TX2IC)	
0059_{16}		
$005A_{16}$	Timer B0 interrupt control register (TB0IC)	35
$005B_{16}$	Timer B1 interrupt control register (TB1IC)	
$005C_{16}$		
$005D_{16}$	INT0 interrupt control register (INT0IC)	35
$005E_{16}$	INT1 interrupt control register (INT1IC)	
$005F_{16}$		

Quick Reference to Pages Classified by Address

Address	Register	Page
0380_{16}	Count start flag (TABSR)	51
0381_{16}	Clock prescaler reset flag (CPSRF)	52
0382_{16}	One-shot start flag (ONSF)	
0383_{16}	Trigger select register (TRGSR)	
0384_{16}	Up-down flag (UDF)	51
0385_{16}		
0386_{16}	Timer A0 (TA0)	51
0387_{16}		
0388_{16}	Timer X0 (TX0)	67
0389_{16}		
$038A_{16}$	Timer X1 (TX1)	
$038B_{16}$		
$038C_{16}$	Timer X2 (TX2)	
$038D_{16}$		
$038E_{16}$	Clock divided counter (CDC)	
$038F_{16}$		
0390_{16}	Timer B0 (TB0)	61
0391_{16}		
0392_{16}	Timer B1 (TB1)	
0393_{16}		
0394_{16}		
0395_{16}		
0396_{16}	Timer A0 mode register (TA0MR)	50
0397_{16}	Timer X0 mode register (TX0MR)	66
0398_{16}	Timer X1 mode register (TX1MR)	
0399_{16}	Timer X2 mode register (TX2MR)	
$039A_{16}$		
$039B_{16}$	Timer B0 mode register (TB0MR)	60
$039C_{16}$	Timer B1 mode register (TB1MR)	
$039D_{16}$		
$039E_{16}$		
$039F_{16}$		
$03A0_{16}$	UART0 transmit/receive mode register (U0MR)	79
$03A1_{16}$	UART0 bit rate generator (U0BRG)	78
$03A2_{16}$	UART0 transmit buffer register (U0TB)	
$03A3_{16}$		
$03A4_{16}$	UART0 transmit/receive control register 0 (U0C0)	79
$03A5_{16}$	UART0 transmit/receive control register 1 (U0C1)	80
$03A6_{16}$	UART0 receive buffer register (U0RB)	78
$03A7_{16}$		
$03A8_{16}$	UART1 transmit/receive mode register (U1MR)	79
$03A9_{16}$	UART1 bit rate generator (U1BRG)	78
$03AA_{16}$	UART1 transmit buffer register (U1TB)	
$03AB_{16}$		
$03AC_{16}$	UART1 transmit/receive control register 0 (U1C0)	79
$03AD_{16}$	UART1 transmit/receive control register 1 (U1C1)	80
$03AE_{16}$	UART1 receive buffer register (U1RB)	78
$03AF_{16}$		
$03B0_{16}$	UART transmit/receive control register 2 (UCON)	80
$03B1_{16}$		
$03B2_{16}$		
$03B3_{16}$		
$03B4_{16}$	Flash memory control register 0 (FCON0) (Note)	128
$03B5_{16}$	Flash memory control register 1 (FCON1) (Note)	
$03B6_{16}$	Flash command register (FCMD) (Note)	
$03B7_{16}$		
$03B8_{16}$		
$03B9_{16}$		
$03BA_{16}$		
$03BB_{16}$		
$03BC_{16}$		
$03BD_{16}$		
$03BE_{16}$		
$03BF_{16}$		

Note: This register is only exist in flash memory version.

Address	Register	Page
$03C0_{16}$	A-D register 0 (AD0)	93
$03C1_{16}$		
$03C2_{16}$	A-D register 1 (AD1)	
$03C3_{16}$		
$03C4_{16}$	A-D register 2 (AD2)	
$03C5_{16}$		
$03C6_{16}$	A-D register 3 (AD3)	
$03C7_{16}$		
$03C8_{16}$	A-D register 4 (AD4)	
$03C9_{16}$		
$03CA_{16}$	A-D register 5 (AD5)	
$03CB_{16}$		
$03CC_{16}$	A-D register 6 (AD6)	
$03CD_{16}$		
$03CE_{16}$	A-D register 7 (AD7)	
$03CF_{16}$		
$03D0_{16}$		
$03D1_{16}$		
$03D2_{16}$		
$03D3_{16}$		
$03D4_{16}$	A-D control register 2 (ADCON2)	93
$03D5_{16}$		
$03D6_{16}$	A-D control register 0 (ADCON0)	92
$03D7_{16}$	A-D control register 1 (ADCON1)	
$03D8_{16}$		
$03D9_{16}$		
$03DA_{16}$		
$03DB_{16}$		
$03DC_{16}$		
$03DD_{16}$		
$03DE_{16}$		
$03DF_{16}$		
$03E0_{16}$	Port P0 (P0)	105
$03E1_{16}$	Port P1 (P1)	
$03E2_{16}$	Port P0 direction register (PD0)	104
$03E3_{16}$	Port P1 direction register (PD1)	
$03E4_{16}$	Port P2 (P2)	105
$03E5_{16}$	Port P3 (P3)	
$03E6_{16}$	Port P2 direction register (PD2)	104
$03E7_{16}$	Port P3 direction register (PD3)	
$03E8_{16}$	Port P4 (P4)	105
$03E9_{16}$	Port P5 (P5)	
$03EA_{16}$	Port P4 direction register (PD4)	104
$03EB_{16}$	Port P5 direction register (PD5)	
$03EC_{16}$	Port P6 (P6)	105
$03ED_{16}$	Port P7 (P7)	
$03EE_{16}$	Port P6 direction register (PD6)	104
$03EF_{16}$	Port P7 direction register (PD7)	
$03F0_{16}$		
$03F1_{16}$		
$03F2_{16}$		
$03F3_{16}$		
$03F4_{16}$		
$03F5_{16}$		
$03F6_{16}$		
$03F7_{16}$		
$03F8_{16}$		
$03F9_{16}$		
$03FA_{16}$		
$03FB_{16}$		
$03FC_{16}$	Pull-up control register 0 (PUR0)	106
$03FD_{16}$	Pull-up control register 1 (PUR1)	
$03FE_{16}$	Pull-up control register 2 (PUR2)	
$03FF_{16}$		

Chapter 1

Hardware

Description

The M30201 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/60 Series CPU core. M30201 group is packaged in a 52-pin plastic molded SDIP, or 56-pin plastic molded QFP. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 1M bytes of address space, they are capable of executing instructions at high speed.
The M30201 group includes a wide range of products with different internal memory types and sizes and various package types.

Features

• Basic machine instructions	Compatible with the M16C/60 series
• Memory capacity	ROM/RAM (See figure 1.4. ROM expansion.)
• Shortest instruction execution time	100ns (f(X_{IN})=10MHz)
• Supply voltage	4.0 to 5.5V (f(X_{IN})=10MHz) :mask ROM version 2.7 to 5.5V (f(X_{IN})=7MHz with software one-wait):mask ROM version 4.0 to 5.5V (f(X_{IN})=10MHz) :flash memory version
• Interrupts	9 internal and 3 external interrupt sources, 4 software (including key input interrupt)
• Multifunction 16-bit timer	Timer A x 1, timer B x 2, timer X x 3
• Clock output	
• Serial I/O	1 channel for UART or clock synchronous, 1 for UART
• A-D converter	10 bits X 8 channels (Expandable up to 13 channels)
• Watchdog timer	1 line
• Programmable I/O	43 lines
• LED drive ports	8 ports
• Clock generating circuit	2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)

Applications

Home appliances, Audio, office equipment, Automobiles

Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error.
Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

------Table of Contents------

Pin Configuration

Figures 1.1 to 1.2 show the pin configurations (top view).

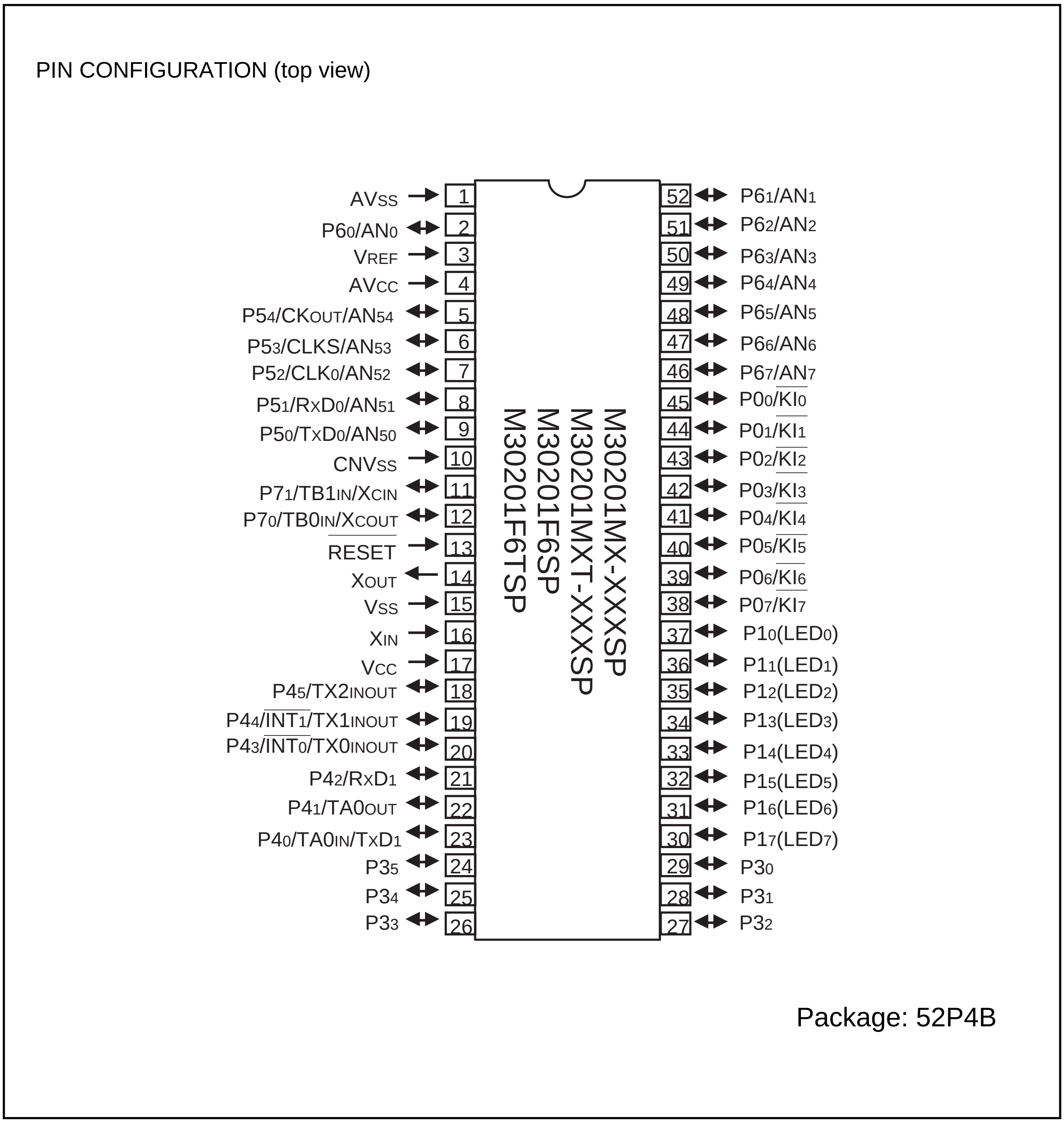


Figure 1.1. Pin configuration for the M30201 group (shrink DIP product) (top view)

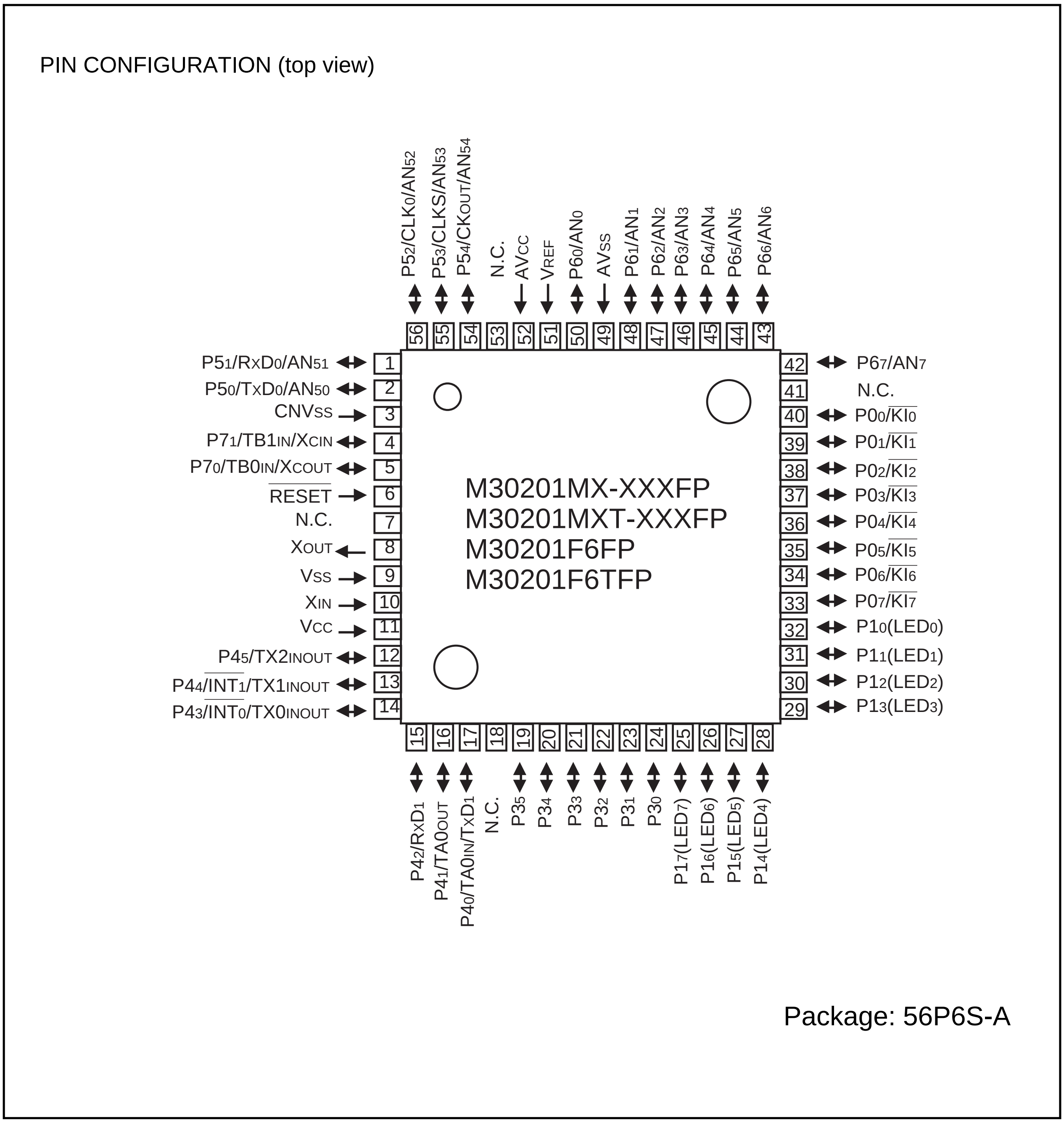
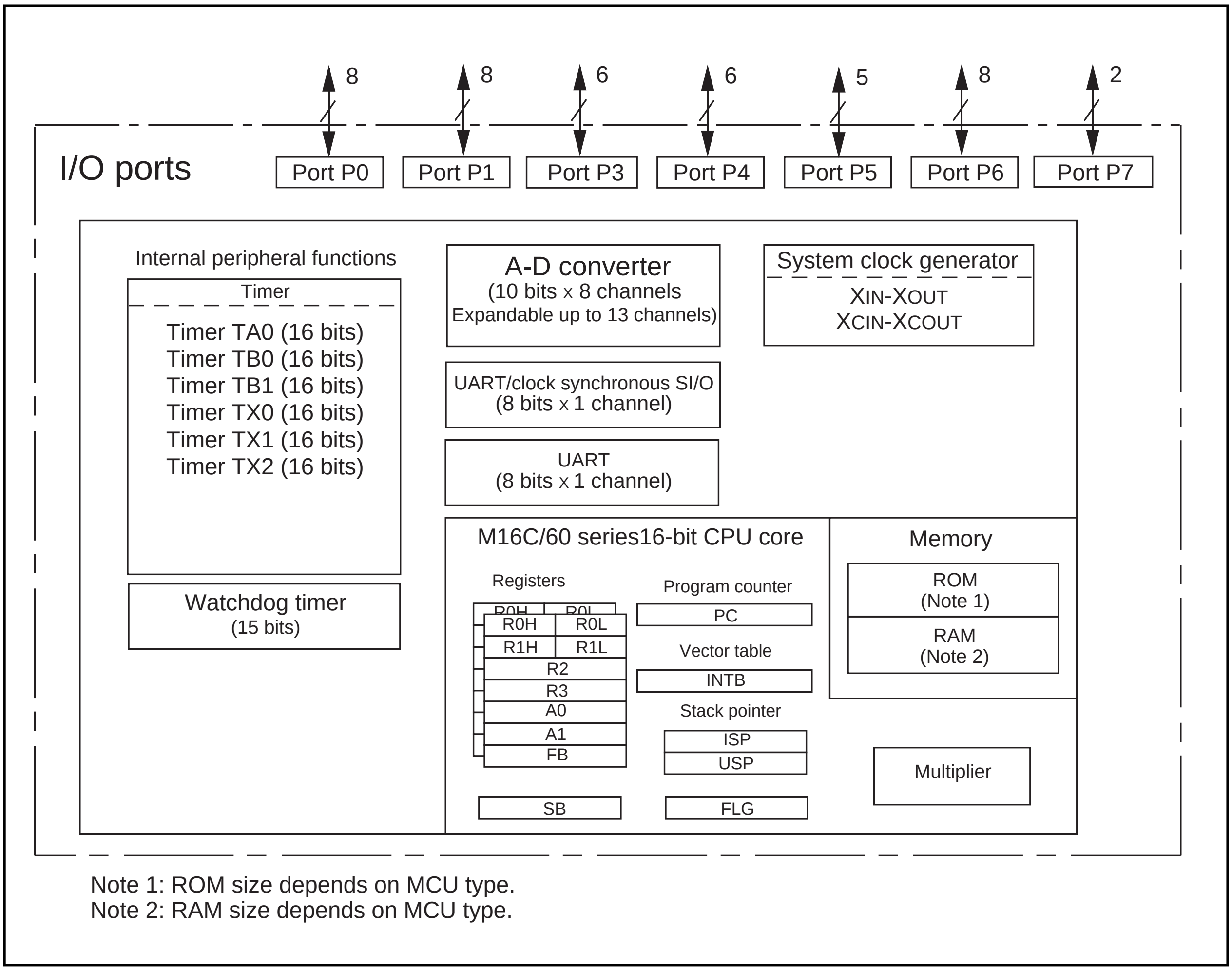


Figure 1.2. Pin configuration for the M30201 group (QFP product) (top view)

Block Diagram

Figure 1.3 is a block diagram of the M30201 group.



Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.

Figure 1.3. Block diagram for the M30201 group

Performance Outline

Table 1.1 is performance outline of M30201 group.

Table 1.1. Performance outline of M30201 group

Item		Performance
Number of basic instructions		91 instructions
Shortest instruction execution time		100ns (f(XIN)=10MHz
Memory capacity	ROM	(See figure 4. ROM expansion.)
	RAM	(See figure 4. ROM expansion.)
I/O port	P0 to P7	43 lines
Multifunction timer	TA0	16 bits x 1
	TB0, TB1	16 bits x 2
	TX0, TX1, TX2	16 bits x 3
Serial I/O	UART0	(UART or clock synchronous) x 1
	UART1	UART x 1
A-D converter		10 bits x 8 channels (Expandable up to 13 channels)
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		9 internal and 3 external sources, 4 software sources
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistor, and external ceramic or quartz oscillator)
Supply voltage		4.0 to 5.5V (f(XIN)=10MHz) :mask ROM version 2.7 to 5.5V (f(XIN)=7MHz with software one-wait) :mask ROM version 4.0 to 5.5V (f(XIN)=10MHz) :flash memory version
Power consumption		18mW (f(XIN)=7MHz with software one-wait, Vcc=3V) :mask ROM version 95mW (f(XIN)=10MHz no wait, Vcc=5V) :flash memory version
I/O characteristics	I/O withstand voltage	5V
	Output current	5mA (15mA:LED drive port)
Device configuration		CMOS silicon gate
Package		52-pin plastic mold SDIP 56-pin plastic mold QFP

Mitsubishi plans to release the following products in the M30201 group:

(1) Support for mask ROM version and flash memory version

(2) ROM capacity

(3) Package

52P4B	: Plastic molded SDIP (mask ROM version and flash memory version)
56P6S-A	: Plastic molded QFP (mask ROM version and flash memory version)

July 1998

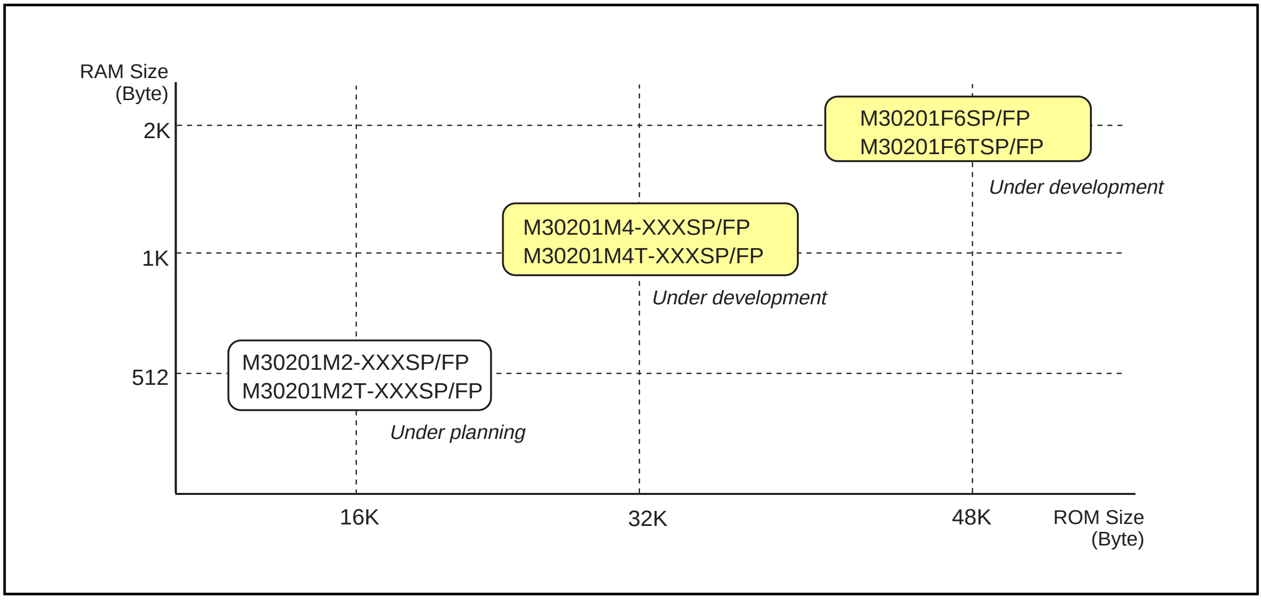


Figure 1.4. ROM expansion

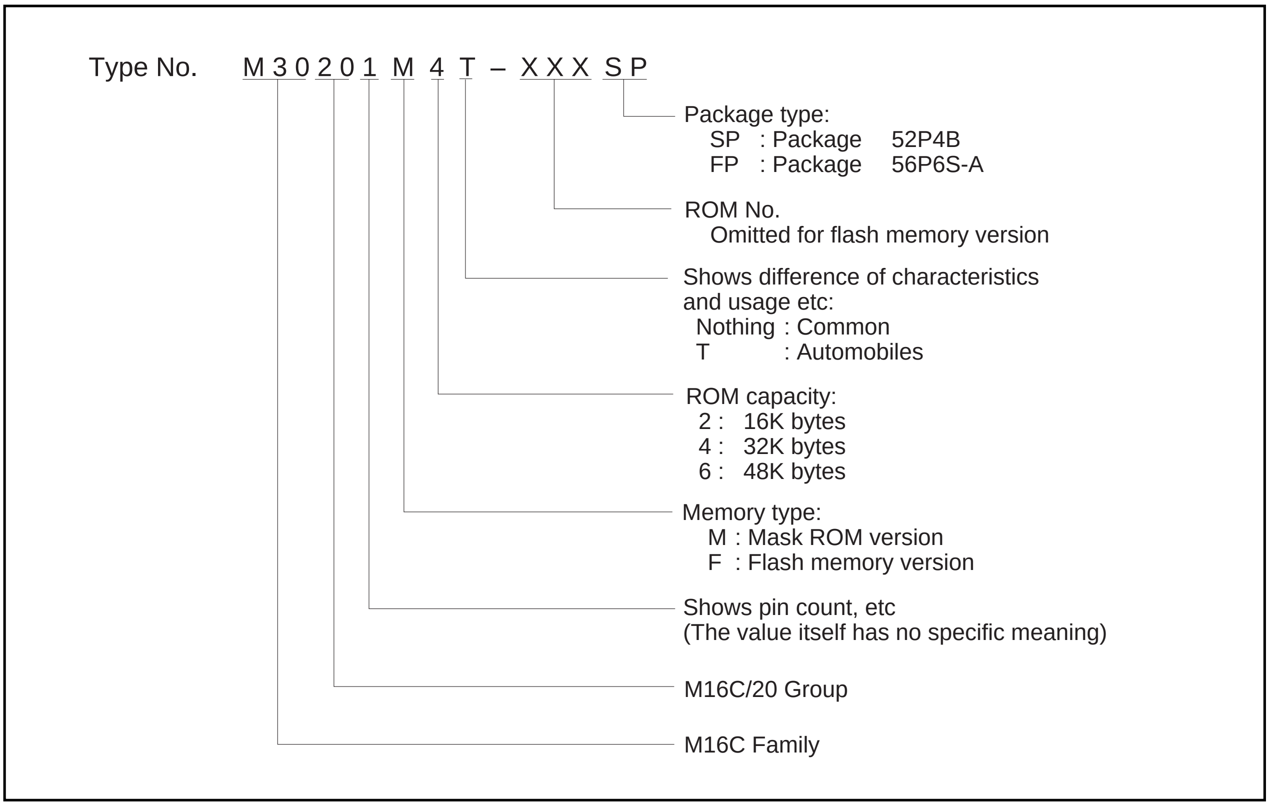


Figure 1.5. Type No., memory size, and package

Under development

Pin Description

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 2.7 to 5.5 V to the V_{CC} pin. Supply 0 V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	Input	Connect it to the V_{SS} pin.
$\overline{\text{RESET}}$	Reset input	Input	A “L” on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	Input Output	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
AV_{CC}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect it to V_{CC}.
AV_{SS}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect it to V_{SS}.
V_{REF}	Reference voltage input	Input	This pin is a reference voltage input for the A-D converter.
$P0_0$ to $P0_7$	I/O port P0	Input/output	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistor.
$P1_0$ to $P1_7$	I/O port P1	Input/output	This is an 8-bit I/O port equivalent to P0.
$P3_0$ to $P3_5$	I/O port P3	Input/output	This is a 6-bit I/O port equivalent to P0.
$P4_0$ to $P4_5$	I/O port P4	Input/output	This is a 6-bit I/O port equivalent to P0. The $P4_0$ pin is shared with timer A0 input and serial I/O output TxD1. The $P4_1$ pin is shared with timer A0 output. The $P4_2$ pin is shared with serial I/O input RxD1. The $P4_3$ pin is shared with external interrupt INT0 and timer X0 input/output $TX0_{INOUT}$. The $P4_4$ pin is shared with external interrupt INT1 and timer X1 input/output $TX1_{INOUT}$. The $P4_5$ pin is shared with timer X2 input/output $TX2_{INOUT}$.
$P5_0$ to $P5_4$	I/O port P5	Input/output	This is a 5-bit I/O port equivalent to P0. The $P5_0$, $P5_1$, $P5_2$, and $P5_3$ pins are shared with serial I/O pins TxD_0, RxD_0, CLK_0, and CLKS. The $P5_4$ pin is shared with clock output CLK_{OUT}. Also, these pins are shared with analog input pins AN_{50} through AN_{54}.
$P6_0$ to $P6_7$	I/O port P6	Input/output	This is an 8-bit I/O port equivalent to P0. These pins are shared with analog input pins AN_0 through AN_7.
$P7_0$ to $P7_1$	I/O port P7	Input/output	This is a 2-bit I/O port equivalent to P0 . These pins are used for input/output to and from the oscillator circuit for the clock. Connect a crystal oscillator between the X_{CIN} and the X_{COUT} pins.

Under development

Operation of Functional Blocks

The M30201 accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, A-D converter, and I/O ports.

The following explains each unit.

Memory

Figure 1.6 is a memory map of the M30201. The address space extends the 1M bytes from address 00000_{16} to $FFFFF_{16}$. From $FFFFF_{16}$ down is ROM. For example, in the M30201M4-XXXFP, there is 32K bytes of internal ROM from $F8000_{16}$ to $FFFFF_{16}$. The vector table for fixed interrupts such as the reset are mapped to $FFFDC_{16}$ to $FFFFF_{16}$. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 00400_{16} up is RAM. For example, in the M30201M4-XXXFP, there is 1K byte of internal RAM from 00400_{16} to $007FF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped to 00000_{16} to $003FF_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is mapped to $FFE00_{16}$ to $FFFDB_{16}$. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

Type No.	Address $XXXXX_{16}$	Address $YYYYY_{16}$
M30201M4	$F8000_{16}$	$007FF_{16}$
M30201M2	$FC000_{16}$	$005FF_{16}$
M30201F6	$F4000_{16}$	$00BFF_{16}$

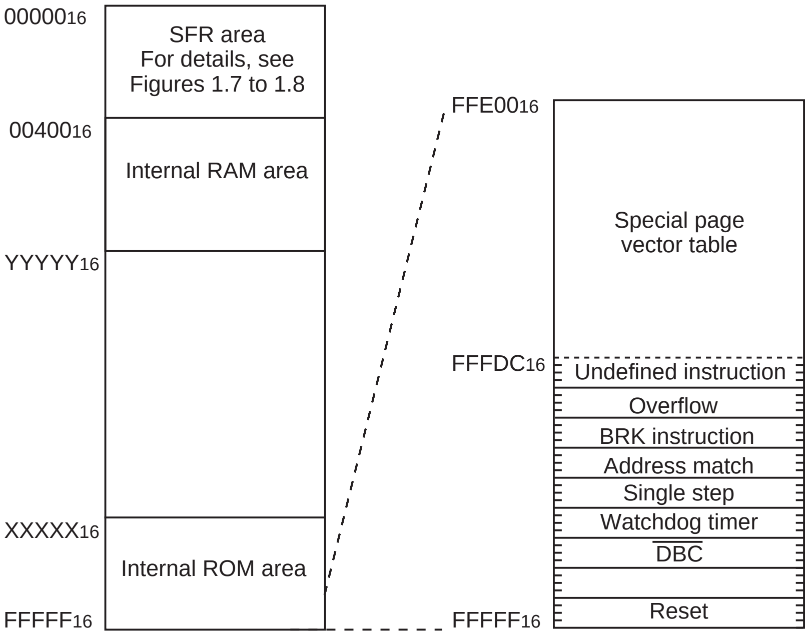


Figure 1.6. Memory map

Under development

0000_{16}	
0001_{16}	
0002_{16}	
0003_{16}	
0004_{16}	Processor mode register 0 (PM0)
0005_{16}	Processor mode register 1(PM1)
0006_{16}	System clock control register 0 (CM0)
0007_{16}	System clock control register 1 (CM1)
0008_{16}	
0009_{16}	Address match interrupt enable register (AIER)
$000A_{16}$	Protect register (PRCR)
$000B_{16}$	
$000C_{16}$	
$000D_{16}$	
$000E_{16}$	Watchdog timer start register (WDTS)
$000F_{16}$	Watchdog timer control register (WDC)
0010_{16}	Address match interrupt register 0 (RMAD0)
0011_{16}	
0012_{16}	
0013_{16}	
0014_{16}	Address match interrupt register 1 (RMAD1)
0015_{16}	
0016_{16}	
0017_{16}	
0018_{16}	
0019_{16}	
$001A_{16}$	
$001B_{16}$	
$001C_{16}$	
$001D_{16}$	
$001E_{16}$	
$001F_{16}$	
0020_{16}	
0021_{16}	
0022_{16}	
0023_{16}	
0024_{16}	
0025_{16}	
0026_{16}	
0027_{16}	
0028_{16}	
0029_{16}	
$002A_{16}$	
$002B_{16}$	
$002C_{16}$	
$002D_{16}$	
$002E_{16}$	
$002F_{16}$	
0030_{16}	
0031_{16}	
0032_{16}	
0033_{16}	
0034_{16}	
0035_{16}	
0036_{16}	
0037_{16}	
0038_{16}	
0039_{16}	
$003A_{16}$	
$003B_{16}$	
$003C_{16}$	
$003D_{16}$	
$003E_{16}$	
$003F_{16}$	
0040_{16}	
0041_{16}	
0042_{16}	
0043_{16}	
0044_{16}	
0045_{16}	
0046_{16}	
0047_{16}	
0048_{16}	
0049_{16}	
$004A_{16}$	
$004B_{16}$	
$004C_{16}$	
$004D_{16}$	Key input interrupt control register (KUPIC)
$004E_{16}$	A-D conversion interrupt control register (ADIC)
$004F_{16}$	
0050_{16}	
0051_{16}	UART0 transmit interrupt control register (S0TIC)
0052_{16}	UART0 receive interrupt control register (S0RIC)
0053_{16}	UART1 transmit interrupt control register (S1TIC)
0054_{16}	UART1 receive interrupt control register (S1RIC)
0055_{16}	Timer A0 interrupt control register (TA0IC)
0056_{16}	Timer X0 interrupt control register (TX0IC)
0057_{16}	Timer X1 interrupt control register (TX1IC)
0058_{16}	Timer X2 interrupt control register (TX2IC)
0059_{16}	
$005A_{16}$	Timer B0 interrupt control register (TB0IC)
$005B_{16}$	Timer B1 interrupt control register (TB1IC)
$005C_{16}$	
$005D_{16}$	INT0 interrupt control register (INT0IC)
$005E_{16}$	INT1 interrupt control register (INT1IC)
$005F_{16}$	

Figure 1.7. Location of peripheral unit control registers (1)

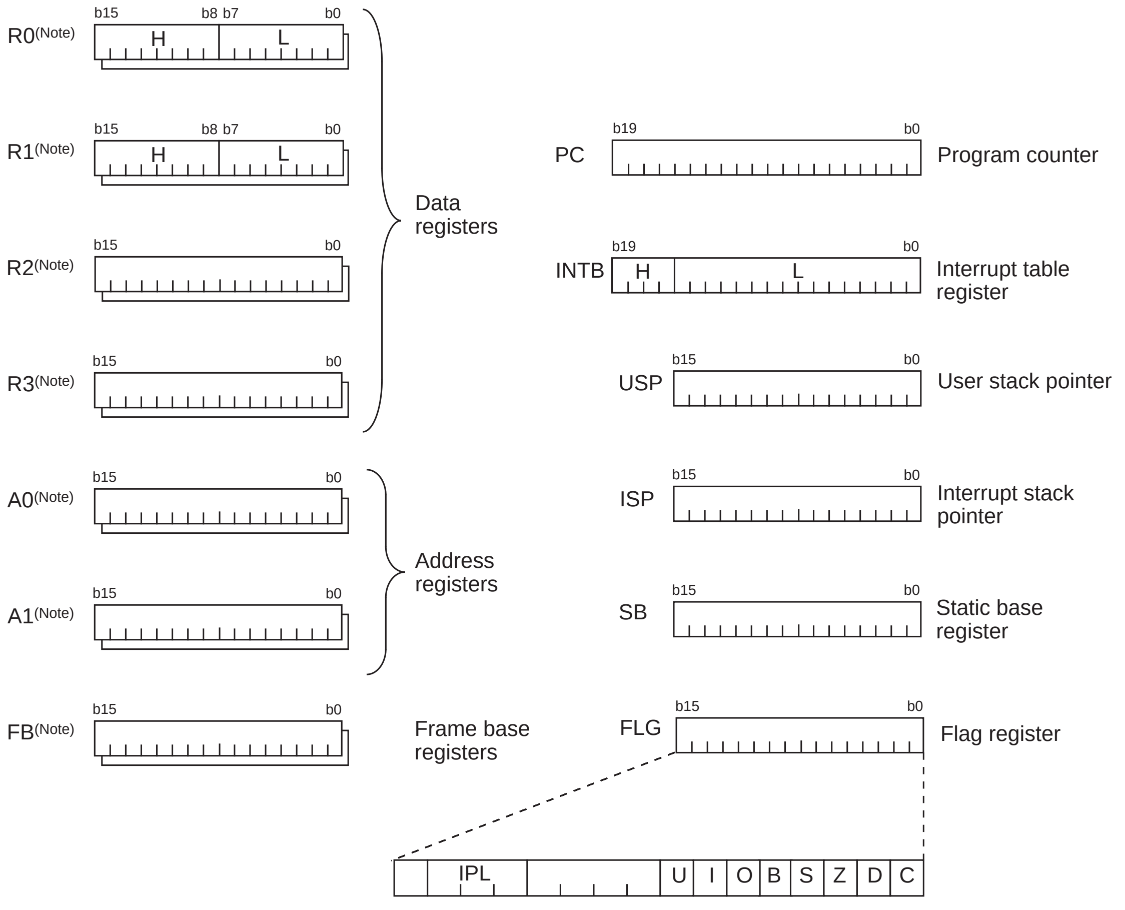
0380$_{16}$	Count start flag (TABSR)	03C0$_{16}$	A-D register 0 (AD0)
0381$_{16}$	Clock prescaler reset flag (CPSRF)	03C1$_{16}$	
0382$_{16}$	One-shot start flag (ONSF)	03C2$_{16}$	A-D register 1 (AD1)
0383$_{16}$	Trigger select register (TRGSR)	03C3$_{16}$	
0384$_{16}$	Up-down flag (UDF)	03C4$_{16}$	A-D register 2 (AD2)
0385$_{16}$		03C5$_{16}$	
0386$_{16}$	Timer A0 (TA0)	03C6$_{16}$	A-D register 3 (AD3)
0387$_{16}$		03C7$_{16}$	
0388$_{16}$	Timer X0 (TX0)	03C8$_{16}$	A-D register 4 (AD4)
0389$_{16}$		03C9$_{16}$	
038A$_{16}$	Timer X1 (TX1)	03CA$_{16}$	A-D register 5 (AD5)
038B$_{16}$		03CB$_{16}$	
038C$_{16}$	Timer X2 (TX2)	03CC$_{16}$	A-D register 6 (AD6)
038D$_{16}$		03CD$_{16}$	
038E$_{16}$	Clock divided counter (CDC)	03CE$_{16}$	A-D register 7 (AD7)
038F$_{16}$		03CF$_{16}$	
0390$_{16}$	Timer B0 (TB0)	03D0$_{16}$	
0391$_{16}$		03D1$_{16}$	
0392$_{16}$	Timer B1 (TB1)	03D2$_{16}$	
0393$_{16}$		03D3$_{16}$	
0394$_{16}$		03D4$_{16}$	A-D control register 2 (ADCON2)
0395$_{16}$		03D5$_{16}$	
0396$_{16}$	Timer A0 mode register (TA0MR)	03D6$_{16}$	A-D control register 0 (ADCON0)
0397$_{16}$	Timer X0 mode register (TX0MR)	03D7$_{16}$	A-D control register 1 (ADCON1)
0398$_{16}$	Timer X1 mode register (TX1MR)	03D8$_{16}$	
0399$_{16}$	Timer X2 mode register (TX2MR)	03D9$_{16}$	
039A$_{16}$		03DA$_{16}$	
039B$_{16}$	Timer B0 mode register (TB0MR)	03DB$_{16}$	
039C$_{16}$	Timer B1 mode register (TB1MR)	03DC$_{16}$	
039D$_{16}$		03DD$_{16}$	
039E$_{16}$		03DE$_{16}$	
039F$_{16}$		03DF$_{16}$	
03A0$_{16}$	UART0 transmit/receive mode register (U0MR)	03E0$_{16}$	Port P0 (P0)
03A1$_{16}$	UART0 bit rate generator (U0BRG)	03E1$_{16}$	Port P1 (P1)
03A2$_{16}$	UART0 transmit buffer register (U0TB)	03E2$_{16}$	Port P0 direction register (PD0)
03A3$_{16}$		03E3$_{16}$	Port P1 direction register (PD1)
03A4$_{16}$	UART0 transmit/receive control register 0 (U0C0)	03E4$_{16}$	Port P2 (P2) (Reserved)
03A5$_{16}$	UART0 transmit/receive control register 1 (U0C1)	03E5$_{16}$	Port P3 (P3)
03A6$_{16}$	UART0 receive buffer register (U0RB)	03E6$_{16}$	Port P2 direction register (PD2) (Reserved)
03A7$_{16}$		03E7$_{16}$	Port P3 direction register (PD3)
03A8$_{16}$	UART1 transmit/receive mode register (U1MR)	03E8$_{16}$	Port P4 (P4)
03A9$_{16}$	UART1 bit rate generator (U1BRG)	03E9$_{16}$	Port P5 (P5)
03AA$_{16}$	UART1 transmit buffer register (U1TB)	03EA$_{16}$	Port P4 direction register (PD4)
03AB$_{16}$		03EB$_{16}$	Port P5 direction register (PD5)
03AC$_{16}$	UART1 transmit/receive control register 0 (U1C0)	03EC$_{16}$	Port P6 (P6)
03AD$_{16}$	UART1 transmit/receive control register 1 (U1C1)	03ED$_{16}$	Port P7 (P7)
03AE$_{16}$	UART1 receive buffer register (U1RB)	03EE$_{16}$	Port P6 direction register (PD6)
03AF$_{16}$		03EF$_{16}$	Port P7 direction register (PD7)
03B0$_{16}$	UART transmit/receive control register 2 (UCON)	03F0$_{16}$	
03B1$_{16}$		03F1$_{16}$	
03B2$_{16}$		03F2$_{16}$	
03B3$_{16}$		03F3$_{16}$	
03B4$_{16}$	Flash memory control register 0 (FCON0) (Note)	03F4$_{16}$	
03B5$_{16}$	Flash memory control register 1 (FCON1) (Note)	03F5$_{16}$	
03B6$_{16}$	Flash command register (FCMD) (Note)	03F6$_{16}$	
03B7$_{16}$		03F7$_{16}$	
03B8$_{16}$		03F8$_{16}$	
03B9$_{16}$		03F9$_{16}$	
03BA$_{16}$		03FA$_{16}$	
03BB$_{16}$		03FB$_{16}$	
03BC$_{16}$		03FC$_{16}$	Pull-up control register 0 (PUR0)
03BD$_{16}$		03FD$_{16}$	Pull-up control register 1 (PUR1)
03BE$_{16}$		03FE$_{16}$	Port P1 drive control register (DRR)
03BF$_{16}$		03FF$_{16}$	

Note: This register is only exist in flash memory version.

Figure 1.8. Location of peripheral unit control registers (2)

Central Processing Unit (CPU)

The CPU has a total of 13 registers shown in Figure 1.9. Seven of these registers (R0, R1, R2, R3, A0, A1, and FB) come in two sets; therefore, these have two register banks.



Note: These registers consist of two register banks.

Figure 1.9. Central processing unit register

(1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, and R3)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H, R1H), and low-order bits as (R0L, R1L). In some instructions, registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0, R3R1).

(2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 16 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

In some instructions, registers A1 and A0 can be combined for use as a 32-bit address register (A1A0).

(3) Frame base register (FB)

Frame base register (FB) is configured with 16 bits, and is used for FB relative addressing.

(4) Program counter (PC)

Program counter (PC) is configured with 20 bits, indicating the address of an instruction to be executed.

(5) Interrupt table register (INTB)

Interrupt table register (INTB) is configured with 20 bits, indicating the start address of an interrupt vector table.

(6) Stack pointer (USP/ISP)

Stack pointer comes in two types: user stack pointer (USP) and interrupt stack pointer (ISP), each configured with 16 bits.

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag). This flag is located at the position of bit 7 in the flag register (FLG).

(7) Static base register (SB)

Static base register (SB) is configured with 16 bits, and is used for SB relative addressing.

(8) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 1.10 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**

  This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.
- **Bit 1: Debug flag (D flag)**

  This flag enables a single-step interrupt.

  When this flag is “1”, a single-step interrupt is generated after instruction execution. This flag is cleared to “0” when the interrupt is acknowledged.
- **Bit 2: Zero flag (Z flag)**

  This flag is set to “1” when an arithmetic operation resulted in 0; otherwise, cleared to “0”.
- **Bit 3: Sign flag (S flag)**

  This flag is set to “1” when an arithmetic operation resulted in a negative value; otherwise, cleared to “0”.
- **Bit 4: Register bank select flag (B flag)**

  This flag chooses a register bank. Register bank 0 is selected when this flag is “0” ; register bank 1 is selected when this flag is “1”.
- **Bit 5: Overflow flag (O flag)**

  This flag is set to “1” when an arithmetic operation resulted in overflow; otherwise, cleared to “0”.
- **Bit 6: Interrupt enable flag (I flag)**

  This flag enables a maskable interrupt.

  An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 7: Stack pointer select flag (U flag)**
  Interrupt stack pointer (ISP) is selected when this flag is "0" ; user stack pointer (USP) is selected when this flag is "1".
  This flag is cleared to "0" when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.
- **Bits 8 to 11: Reserved area**
- **Bits 12 to 14: Processor interrupt priority level (IPL)**
  Processor interrupt priority level (IPL) is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.
  If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.
- **Bit 15: Reserved area**

The C, Z, S, and O flags are changed when instructions are executed. See the software manual for details.

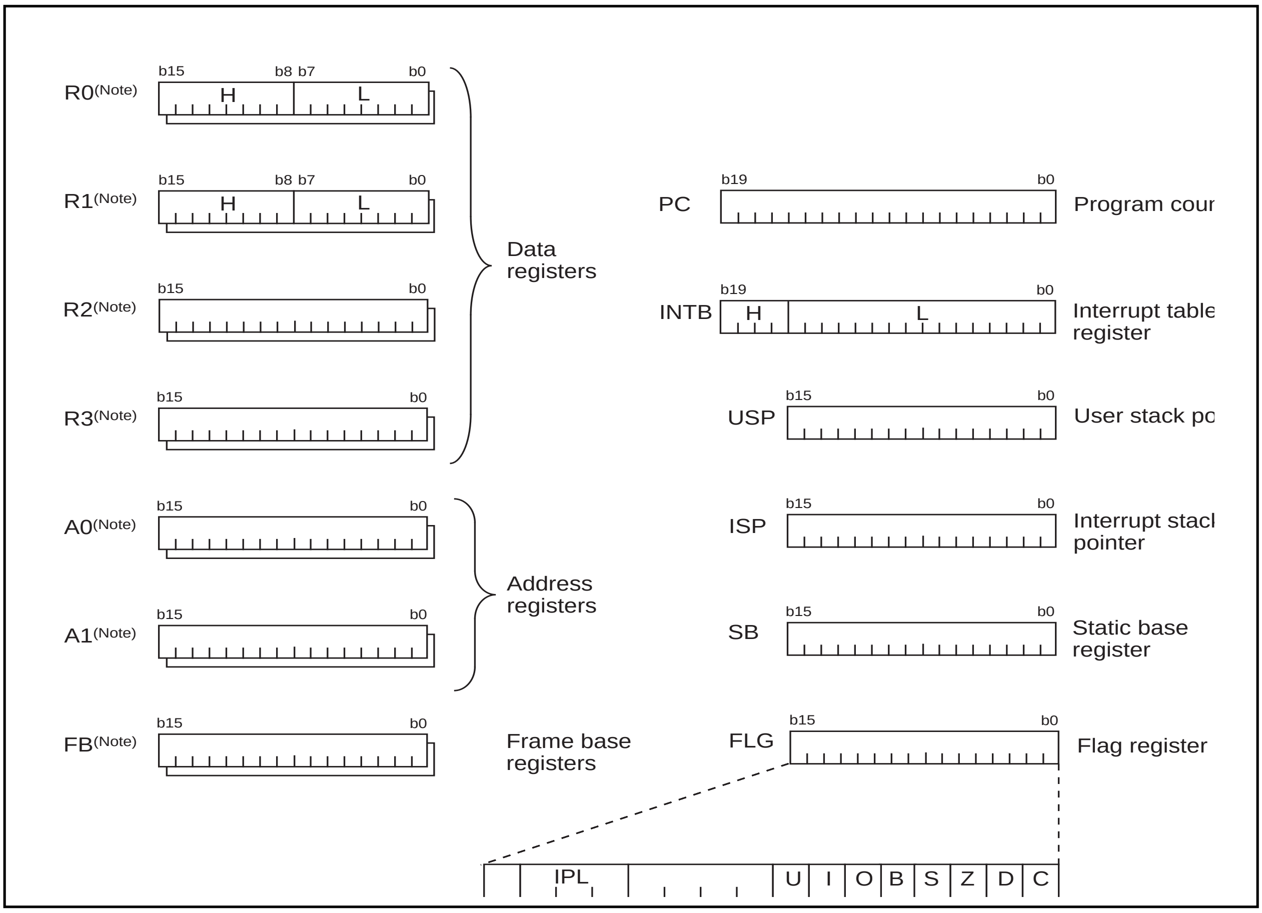


Figure 1.10. Flag register (FLG)

Reset

There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See “Software Reset” for details of software resets.) This section explains on hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level “L” (0.2VCC max.) for at least 20 cycles. When the reset pin level is then returned to the “H” level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure 1.11 shows the example reset circuit. Figure 1.12 shows the reset sequence.

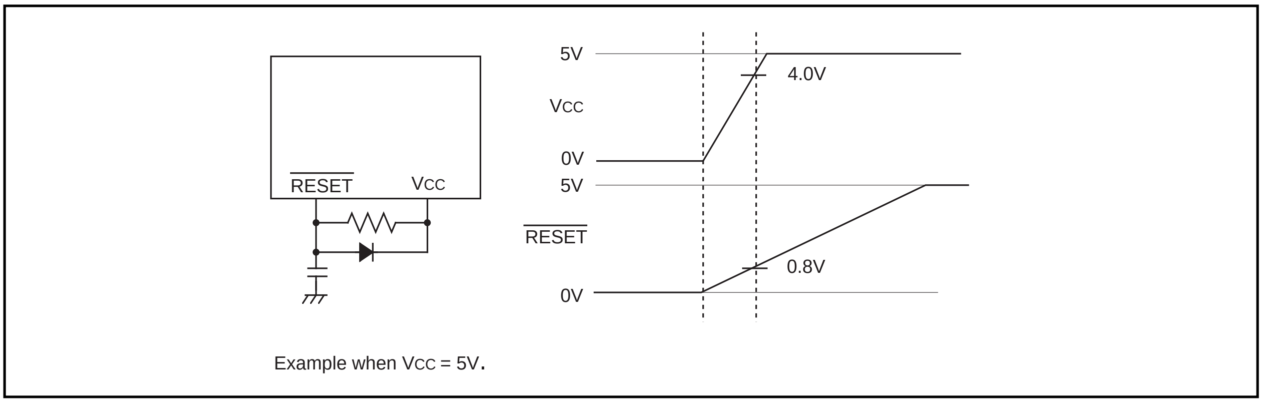


Figure 1.11. Example reset circuit

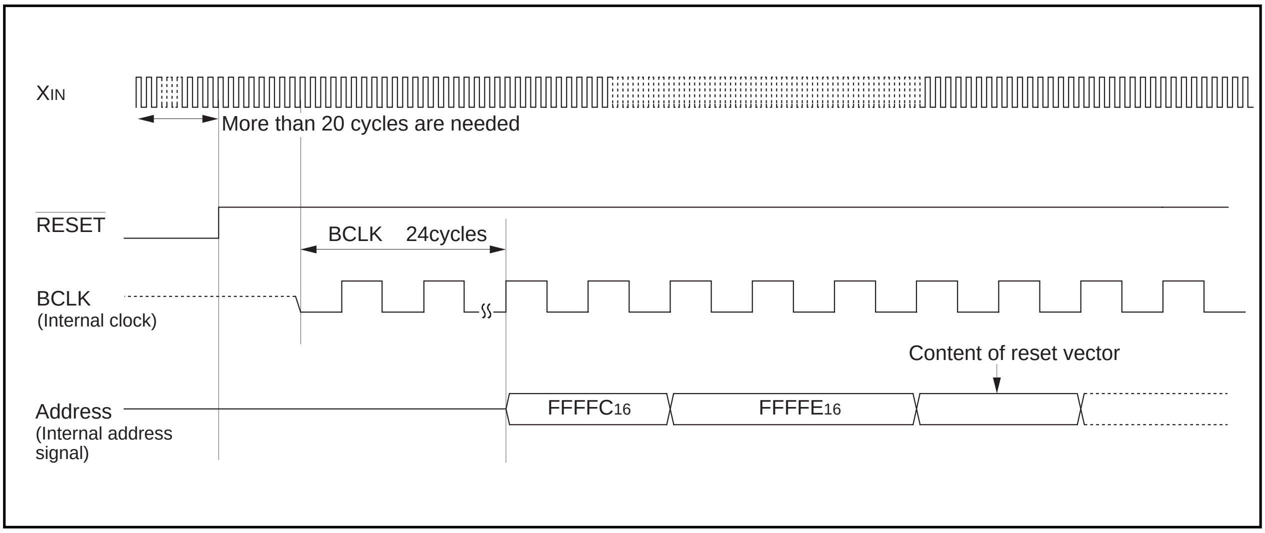


Figure 1.12. Reset sequence

No.	Register	Address	Value after reset
(1)	Processor mode register 0	(0004$_{16}$)···	x x x x 0 0 0 0
(2)	Processor mode register 1	(0005$_{16}$)···	0 x x x x x 0 0
(3)	System clock control register 0	(0006$_{16}$)···	0 1 0 0 1 0 0 0
(4)	System clock control register 1	(0007$_{16}$)···	0 0 1 0 0 0 0 0
(5)	Address match interrupt enable register	(0009$_{16}$)···	x x x x x x 0 0
(6)	Protect register	(000A$_{16}$)···	x x x x x 0 0 0
(7)	Watchdog timer control register	(000F$_{16}$)···	0 0 0 ? ? ? ? ?
(8)	Address match interrupt register 0	(0010$_{16}$)···	00$_{16}$
		(0011$_{16}$)···	00$_{16}$
		(0012$_{16}$)···	x x x x 0 0 0 0
(9)	Address match interrupt register 1	(0014$_{16}$)···	00$_{16}$
		(0015$_{16}$)···	00$_{16}$
		(0016$_{16}$)···	x x x x 0 0 0 0
(10)	Key input interrupt control register	(004D$_{16}$)···	x x x x ? 0 0 0
(11)	A-D conversion interrupt control register	(004E$_{16}$)···	x x x x ? 0 0 0
(12)	UART0 transmit interrupt control register	(0051$_{16}$)···	x x x x ? 0 0 0
(13)	UART0 receive interrupt control register	(0052$_{16}$)···	x x x x ? 0 0 0
(14)	UART1 transmit interrupt control register	(0053$_{16}$)···	x x x x ? 0 0 0
(15)	UART1 receive interrupt control register	(0054$_{16}$)···	x x x x ? 0 0 0
(16)	Timer A0 interrupt control register	(0055$_{16}$)···	x x x x ? 0 0 0
(17)	Timer X0 interrupt control register	(0056$_{16}$)···	x x x x ? 0 0 0
(18)	Timer X1 interrupt control register	(0057$_{16}$)···	x x x x ? 0 0 0
(19)	Timer X2 interrupt control register	(0058$_{16}$)···	x x x x ? 0 0 0
(20)	Timer B0 interrupt control register	(005A$_{16}$)···	x x x x ? 0 0 0
(21)	Timer B1 interrupt control register	(005B$_{16}$)···	x x x x ? 0 0 0
(22)	INT0 interrupt control register	(005D$_{16}$)···	x x 0 0 ? 0 0 0
(23)	INT1 interrupt control register	(005E$_{16}$)···	x x 0 0 ? 0 0 0
(24)	Count start flag	(0380$_{16}$)···	0 0 0 x 0 0 0 0
(25)	Clock prescaler reset flag	(0381$_{16}$)···	0 x x x x x x x
(26)	One-shot start flag	(0382$_{16}$)···	x x x x 0 0 0 0
(27)	Trigger select flag	(0383$_{16}$)···	00$_{16}$
(28)	Up-down flag	(0384$_{16}$)···	x x x 0 x x x 0
(29)	Timer A0 mode register	(0396$_{16}$)···	00$_{16}$
(30)	Timer X0 mode register	(0397$_{16}$)···	00$_{16}$
(31)	Timer X1 mode register	(0398$_{16}$)···	00$_{16}$
(32)	Timer X2 mode register	(0399$_{16}$)···	00$_{16}$
(33)	Timer B0 mode register	(039B$_{16}$)··	0 0 ? x 0 0 0 0
(34)	Timer B1 mode register	(039C$_{16}$)··	0 0 ? x 0 0 0 0
(35)	UART0 transmit/receive mode register	(03A0$_{16}$)··	00$_{16}$
(36)	UART0 transmit/receive control register 0	(03A4$_{16}$)··	0 0 0 0 1 0 0 0
(37)	UART0 transmit/receive control register 1	(03A5$_{16}$)··	0 0 0 0 0 0 1 0
(38)	UART1 transmit/receive mode register	(03A8$_{16}$)··	00$_{16}$
(39)	UART1 transmit/receive control register 0	(03AC$_{16}$)··	0 0 0 0 1 0 0 0
(40)	UART1 transmit/receive control register 1	(03AD$_{16}$)··	0 0 0 0 0 0 1 0
(41)	UART transmit/receive control register 2	(03B0$_{16}$)··	x 0 0 0 0 0 0 0
(42)	Flash memory control register 0 (Note)	(03B4$_{16}$)··	0 0 1 0 0 0 0 0
(43)	Flash memory control register 1 (Note)	(03B5$_{16}$)··	x x x x x x 0 0
(44)	Flash command register	(03B6$_{16}$)··	00$_{16}$
(45)	A-D control register 2	(03D4$_{16}$)··	x x x x 0 0 0 0
(46)	A-D control register 0	(03D6$_{16}$)··	0 0 0 0 0 ? ? ?
(47)	A-D control register 1	(03D7$_{16}$)··	00$_{16}$
(48)	Port P0 direction register	(03E2$_{16}$)··	00$_{16}$
(49)	Port P1 direction register	(03E3$_{16}$)··	00$_{16}$
(50)	Port P2 direction register	(03E6$_{16}$)···	x 0 0 0 0 0 0 0
(51)	Port P3 direction register	(03E7$_{16}$)···	x x 0 0 0 0 0 0
(52)	Port P4 direction register	(03EA$_{16}$)···	x x 0 0 0 0 0 0
(53)	Port P5 direction register	(03EB$_{16}$)···	x x x 0 0 0 0 0
(54)	Port P6 direction register	(03EE$_{16}$)··	00$_{16}$
(55)	Port P7 direction register	(03EF$_{16}$)···	x x x x x x 0 0
(56)	Pull-up control register 0	(03FC$_{16}$)··	00$_{16}$
(57)	Pull-up control register 1	(03FD$_{16}$)··	00$_{16}$
(58)	Port P1 drive capacity control register	(03FE$_{16}$)··	00$_{16}$
(59)	Data registers (R0/R1/R2/R3)		0000$_{16}$
(60)	Address registers (A0/A1)		0000$_{16}$
(61)	Frame base register (FB)		0000$_{16}$
(62)	Interrupt table register (INTB)		00000$_{16}$
(63)	User stack pointer (USP)		0000$_{16}$
(64)	Interrupt stack pointer (ISP)		0000$_{16}$
(65)	Static base register (SB)		0000$_{16}$
(66)	Flag register (FLG)		0000$_{16}$

x : Nothing is mapped to this bit
? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Note: This register is only exist in flash memory version.

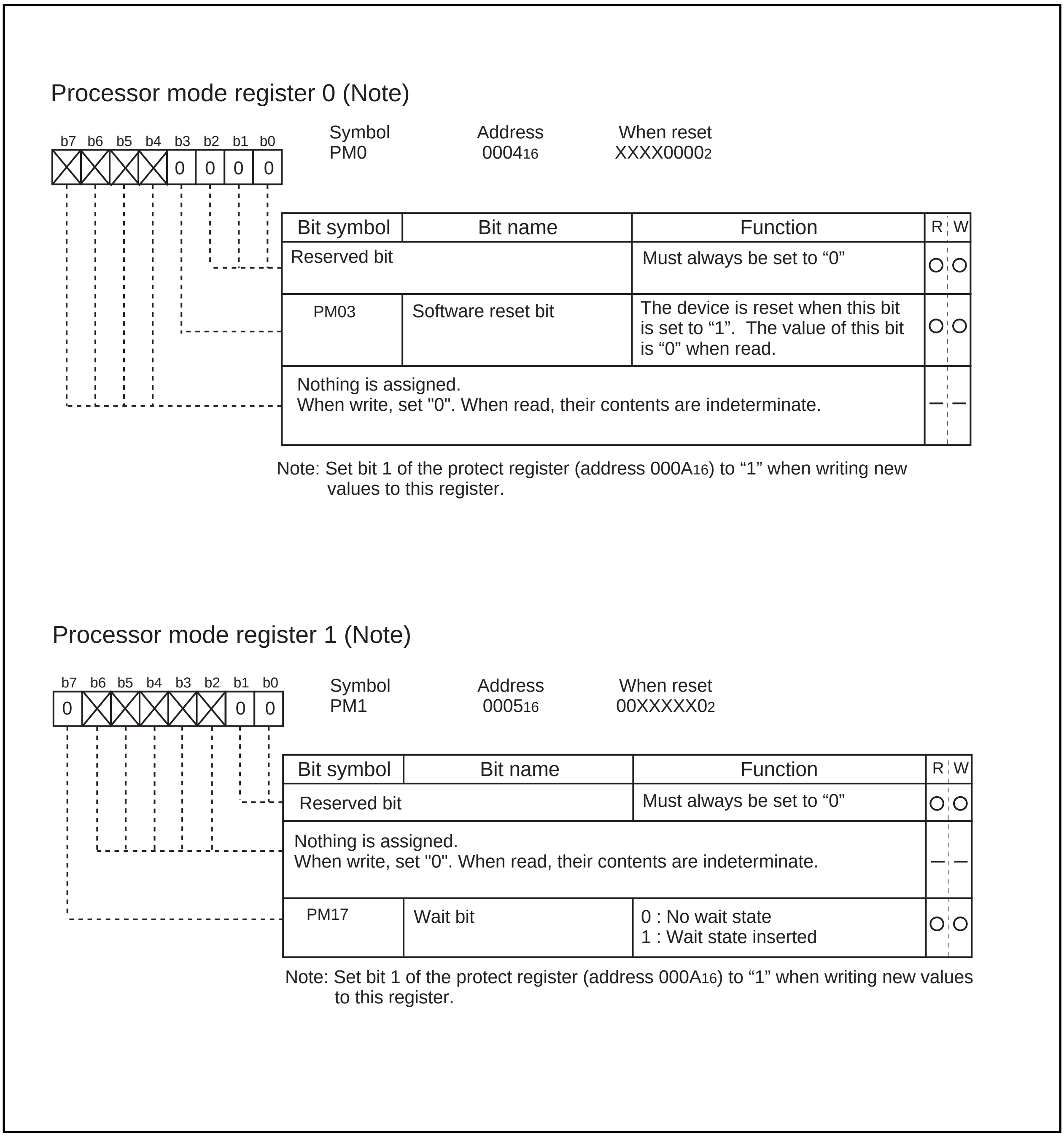
Figure 1.13. Device's internal status after a reset is cleared

Under development

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004_{16}) applies a (software) reset to the microcomputer. A software reset has almost the same effect as a hardware reset. The contents of internal RAM are preserved.

Figure 1.14 shows the processor mode register 0 and 1.



Processor mode register 0 (Note)

Symbol	Address	When reset
PM0	0004_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
PM03	Software reset bit	The device is reset when this bit is set to “1”. The value of this bit is “0” when read.	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Note: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Processor mode register 1 (Note)

Symbol	Address	When reset
PM1	0005_{16}	$00XXXXX0_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
PM17	Wait bit	0 : No wait state 1 : Wait state inserted	○	○

Note: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.

Figure 1.14. Processor mode register 0 and 1.

Software wait

The wait bit (bit 7) of the processor mode register 1 (address 0005_{16})(note) allows you to insert software wait states for the internal ROM/RAM areas. If this bit is 0, the bus cycle is executed in one BCLK (internal clock) period; if the bit is 1, the bus cycle is executed in two BCLK periods. This bit is cleared to 0 after a reset.

The SFR area is unaffected by this control bit; it is always accessed in two BCLK periods.

Table 1.2 shows the relationship between software wait states and bus cycles.

Note: Before attempting to change the contents of the processor mode register 1, set bit 1 of the protect register (address $000A_{16}$) to “1”.

Table 1.2. Software waits and bus cycles

Area	Wait bit	Bus cycle
SFR	Invalid	2 BCLK cycles
Internal ROM/RAM	0	1 BCLK cycle
	1	2 BCLK cycles

Clock Generating Circuit

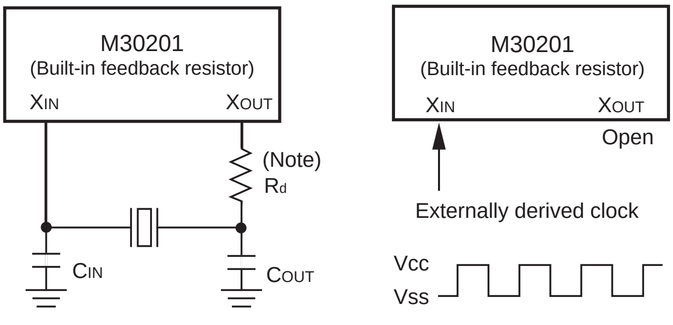
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

Table 1.3. Main clock and sub-clock generating circuits

	Main clock generating circuit	Sub clock generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer A/B/X's count clock source
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

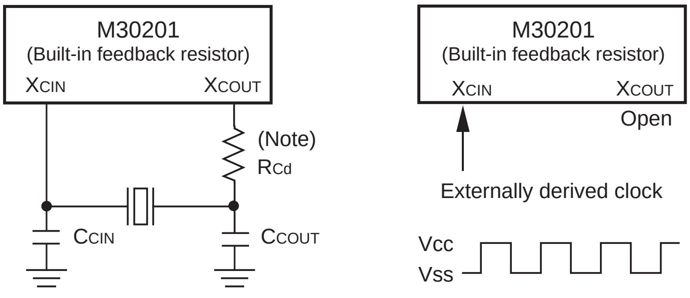
Example of oscillator circuit

Figure 1.15 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure 1.16 shows some examples of sub-clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures 15 and 16 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XIN and XOUT following the instruction.

Figure 1.15. Examples of main clock



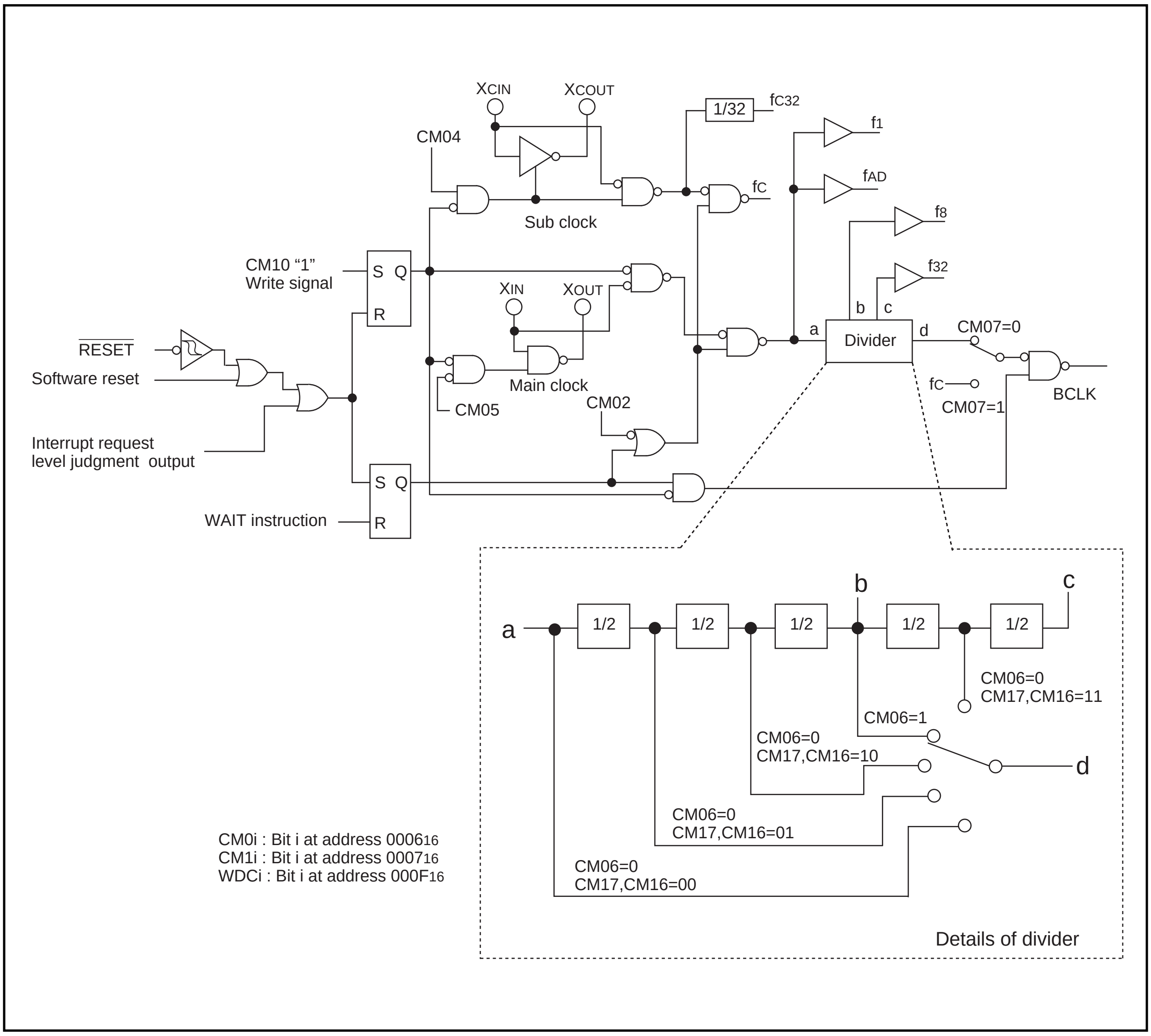
Note: Insert a damping resistor if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable. Also, if the oscillator manufacturer's data sheet specifies that a feedback resistor be added external to the chip, insert a feedback resistor between XCIN and XCOUT following the instruction.

Figure 1.16. Examples of sub-clock

Under development

Clock Control

Figure 1.17 shows the block diagram of the clock generating circuit.



CM0i : Bit i at address 0006_{16}
CM1i : Bit i at address 0007_{16}
WDCi : Bit i at address $000F_{16}$

Figure 1.17. Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Stopping the clock, after switching the operating clock source of CPU to the sub-clock, reduces the power dissipation. After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the X_{IN}-X_{OUT} drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

(2) Sub clock

The sub-clock is generated by the sub-clock oscillation circuit. No sub-clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006_{16}), the sub-clock can be selected as BCLK by using the system clock select bit (bit 7 at address 0006_{16}). However, be sure that the sub-clock oscillation has fully stabilized before switching.

After the oscillation of the sub-clock oscillation circuit has stabilized, the drive capacity of the sub-clock oscillation circuit can be reduced using the X_{CIN}-X_{COUT} drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub-clock oscillation circuit reduces the power dissipation. This bit changes to “1” when shifting to stop mode and at a reset.

(3) BCLK

The BCLK is the clock that drives the CPU, and is fc or the clock is derived by dividing the main clock by 1, 2, 4, 8, or 16. The BCLK is derived by dividing the main clock by 8 after a reset.

The main clock division select bit 0(bit 6 at address 0006_{16}) changes to “1” when shifting from high-speed/medium-speed to stop mode and at reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

(4) Peripheral function clock (f_1, f_8, f_{32}, f_{AD})

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006_{16}) to “1” and then executing a WAIT instruction.

(5) f_{C32}

This clock is derived by dividing the sub-clock by 32. It is used for the timer A, timer B and timer X counts.

(6) f_C

This clock has the same frequency as the sub-clock. It is used for BCLK and for the watchdog timer.

Under development

Figure 1.18 shows the system clock control registers 0 and 1.

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
CM0	0006_{16}	48_{16}

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port $P5_4$ 0 1 : f_C output 1 0 : f_8 output 1 1 : Clock divide counter output	○	○
CM01			○	○
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	○	○
CM03	X_{CIN}-X_{COUT} drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM04	Port X_C select bit	0 : I/O port 1 : X_{CIN}-X_{COUT} generation	○	○
CM05	Main clock (X_{IN}-X_{OUT}) stop bit (Note 3,4,5)	0 : On 1 : Off	○	○
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	○	○
CM07	System clock select bit (Note 6)	0 : X_{IN}, X_{OUT} 1 : X_{CIN}, X_{COUT}	○	○

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shifting to stop mode and at a reset.
Note 3: This bit is used to stop the main clock when placing the device in a low-power mode. If you want to operate with X_{IN} after exiting from the stop mode, set this bit to “0”. When operating with a self-excited oscillator, set the system clock select bit (CM07) to “1” before setting this bit to “1”.
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to “1”, X_{OUT} turns “H”. The built-in feedback resistor remains being connected, so X_{IN} turns pulled up to X_{OUT} (“H”) via the feedback resistor.
Note 6: Set port Xc select bit (CM04) to “1” and stabilize the sub-clock oscillating before setting to this bit from “0” to “1”. Do not write to both bits at the same time. And also, set the main clock stop bit (CM05) to “0” and stabilize the main clock oscillating before setting this bit from “1” to “0”.
Note 7: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 8: f_{C32} is not included.

System clock control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
			0	0	0	0	

Symbol	Address	When reset
CM1	0007_{16}	20_{16}

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note 4)	0 : Clock on 1 : All clocks off (stop mode)	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
CM15	X_{IN}-X_{OUT} drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	○	○
CM17				

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 0006_{16}) is “0”. If “1”, division mode is fixed at 8.
Note 4: If this bit is set to “1”, X_{OUT} turns “H”, and the built-in feedback resistor is cut off. X_{CIN} and X_{COUT} turn high-impedance state.

Figure 1.18. Clock control registers 0 and 1

Clock Output

The clock output function select bit allows you to choose the clock from f_8, fc, or a divide-by-n clock that is output from the $P5_4/CK_{OUT}$ pin. The clock divide counter is an 8-bit counter whose count source is f_{32}, and its divide ratio can be set in the range of 00_{16} to FF_{16}. Figure 1.19 shows a block diagram of clock output.

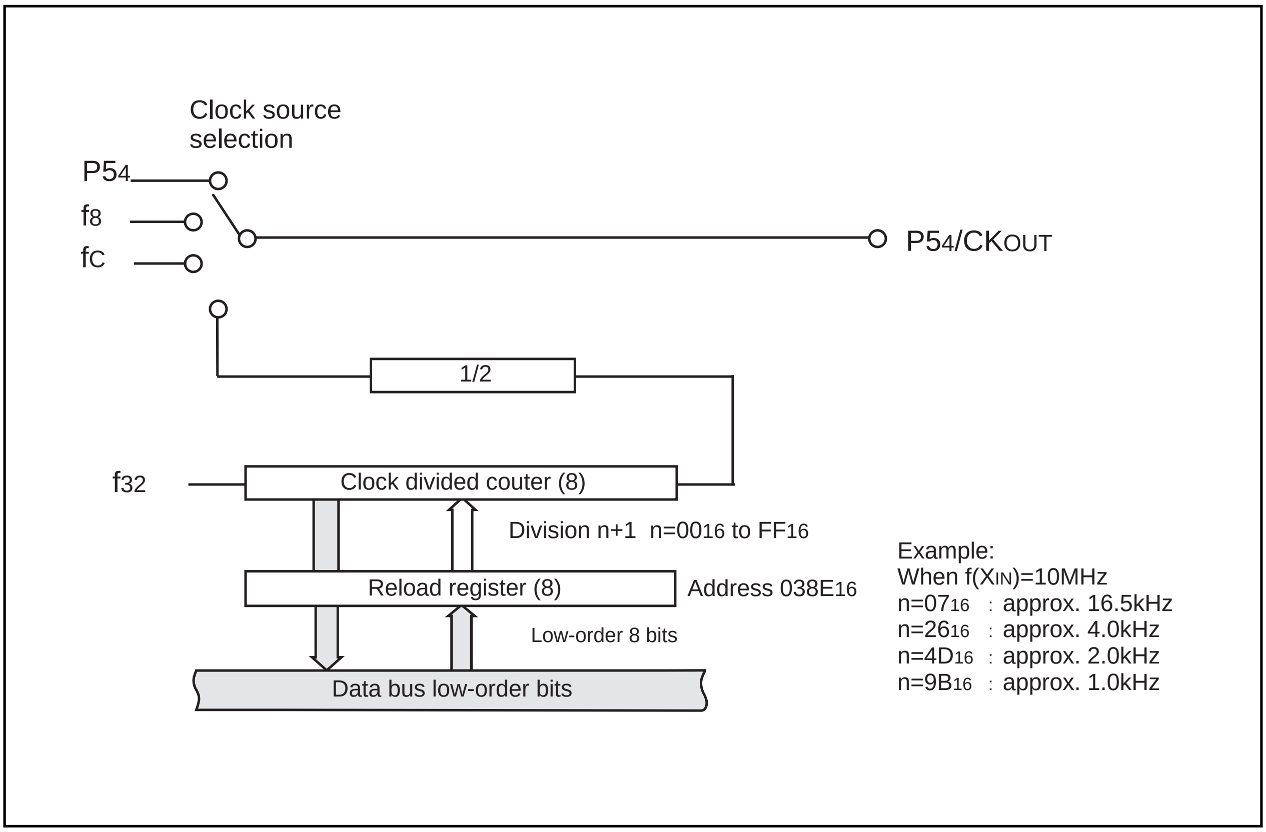


Figure 1.19. Block diagram of clock output

Stop Mode

Writing “1” to the all-clock stop control bit (bit 0 at address 0007_{16}) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V_{CC} remains above 2V.

Because the oscillation of BCLK, f_1 to f_{32}, fc, fc_{32}, and f_{AD} stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A, timer B and timer X operate provided that the event counter mode is set to an external pulse, and UART0 functions provided an external clock is selected. Table 1.4 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or an interrupt. If an interrupt is to be used to cancel stop mode, that interrupt must first have been enabled. If returning by an interrupt, that interrupt routine is executed.

When shifting from high-speed/medium-speed mode to stop mode and at a reset, the main clock division select bit 0 (bit 6 at address 0006_{16}) is set to “1”. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

Table 1.4. Port status during stop mode

Pin		States
Port		Retains status before stop mode
CLK_{OUT}	When f_C selected	“H”
	When f_8, clock devided counter output selected	Retains status before stop mode

Wait Mode

When a WAIT instruction is executed, BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table 1.5 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts from the interrupt routine using as BCLK, the clock that had been selected when the WAIT instruction was executed.

Table 1.5. Port status during wait mode

Pin		States
Port		Retains status before wait mode
CLK_{OUT}	When f_C selected	Does not stop
	When f_8, clock devided counter output selected	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”,the status immediately prior to entering wait mode is maintained.

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 1.6 shows the operating modes corresponding to the settings of system clock control registers 0 and 1.

When reset, the device starts in division by 8 mode. The main clock division select bit 0(bit 6 at address 0006_{16}) changes to “1” when shifting from high-speed/medium-speed to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.

The following shows the operational modes of BCLK.

(1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

(2) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

(3) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. When reset, the device starts operating from this mode. Before the user can go from this mode to no division mode, division by 2 mode, or division by 4 mode, the main clock must be oscillating stably. When going to low-speed or lower power consumption mode, make sure the sub-clock is oscillating stably.

(4) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

(5) No-division mode

The main clock is divided by 1 to obtain the BCLK.

(6) Low-speed mode

f_C is used as BCLK. Note that oscillation of both the main and sub-clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub-clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

(7) Low power dissipation mode

f_C is the BCLK and the main clock is stopped.

Note : Before the count source for BCLK can be changed from X_{IN} to X_{CIN} or vice versa, the clock to which the count source is going to be switched must be oscillating stably. Allow a wait time in software for the oscillation to stabilize before switching over the clock.

Table 1.6. Operating modes dictated by settings of system clock control registers 0 and 1

CM17	CM16	CM07	CM06	CM05	CM04	Operating mode of BCLK
0	1	0	0	0	Invalid	Division by 2 mode
1	0	0	0	0	Invalid	Division by 4 mode
Invalid	Invalid	0	1	0	Invalid	Division by 8 mode
1	1	0	0	0	Invalid	Division by 16 mode
0	0	0	0	0	Invalid	No-division mode
Invalid	Invalid	1	Invalid	0	1	Low-speed mode
Invalid	Invalid	1	Invalid	1	1	Low power dissipation mode

Power Saving

There are three power save modes.

(1) Normal operating mode

- **High-speed mode**

  In this mode, one main clock cycle forms BCLK. The CPU operates on the BCLK. The peripheral functions operate on the clocks specified for each respective function.

- **Medium-speed mode**

  In this mode, the main clock is divided into 2, 4, 8, or 16 to form BCLK. The CPU operates on the BCLK. The peripheral functions operated on the clocks specified for each respective function.

- **Low-speed mode**

  In this mode, fc forms BCLK. The CPU operates on the fc clock. fc is the clock supplied by the subclock. The peripheral functions operate on the clocks specified for each respective function.

- **Low power-dissipation mode**

  This mode is selected when the main clock is stopped from low-speed mode. The CPU operates on the fc clock. fc is the clock supplied by the subclock. Only the peripheral functions for which the subclock was selected as the count source continue to run.

(2) Wait mode

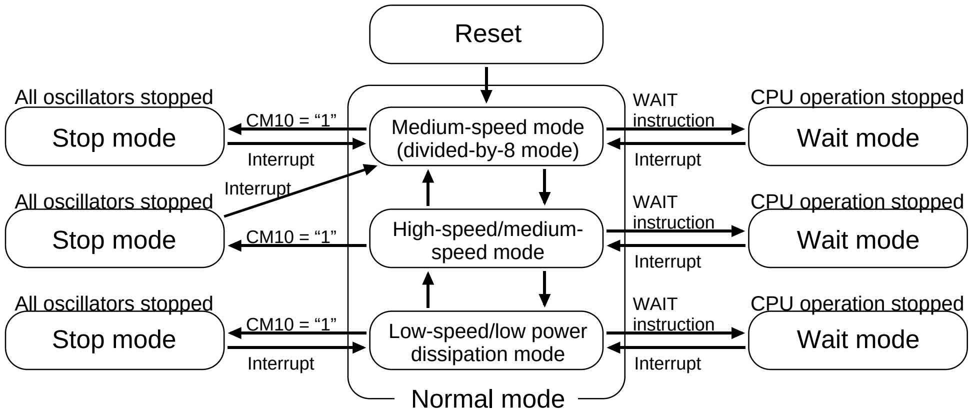
CPU operation is halted in this mode. The oscillator continues to run.

(3) Stop mode

All oscillators stop in this mode. The CPU and internal peripheral functions all stop. Of all 3 power saving modes, power savings are greatest in this mode.

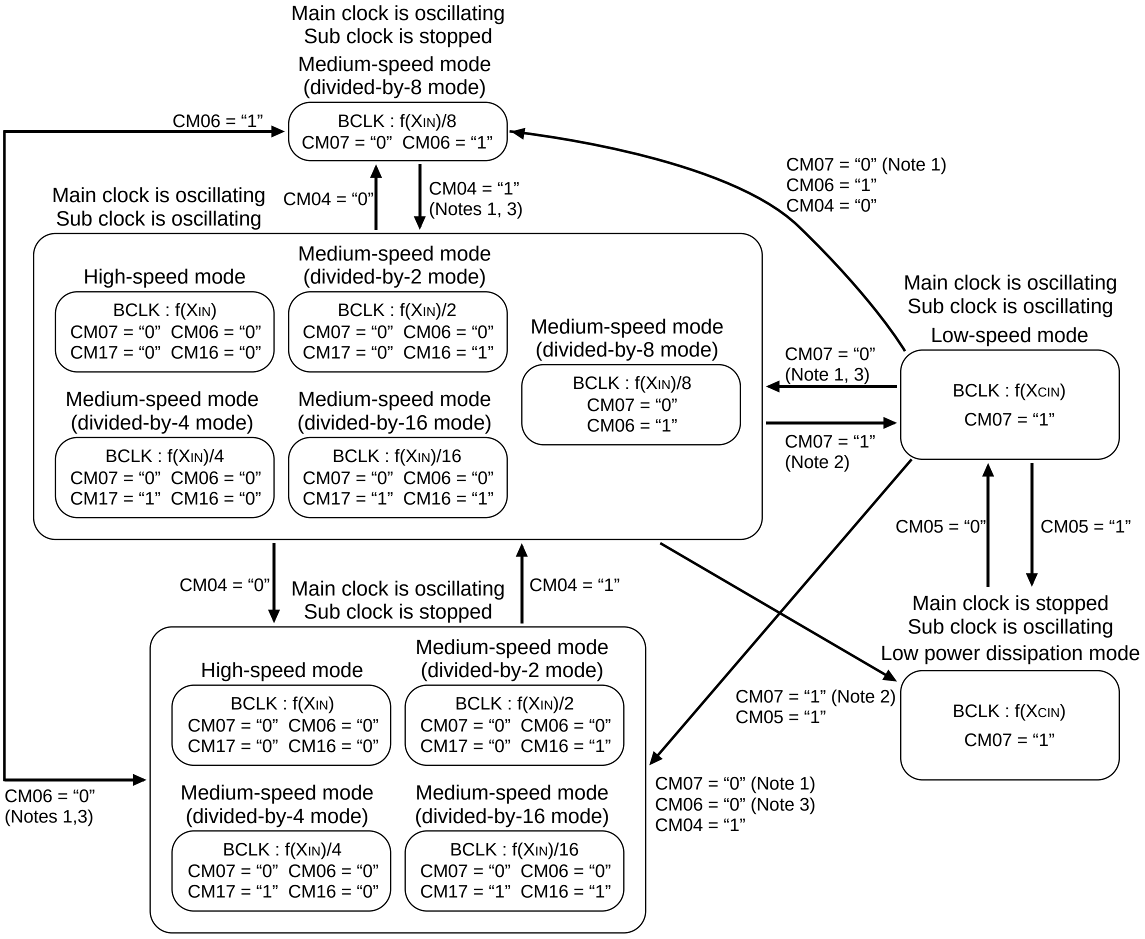
Figure 1.20 shows the transition between each of the three modes, (1), (2), and (3).

Transition of stop mode, wait mode



(Refer to the following for the transition of normal mode.)

Transition of normal mode



Note 1: Switch clock after oscillation of main clock is sufficiently stable.
Note 2: Switch clock after oscillation of sub clock is sufficiently stable.
Note 3: Change CM06 after changing CM17 and CM16.
Note 4: Transit in accordance with arrow.

Figure 1.20. Clock transition

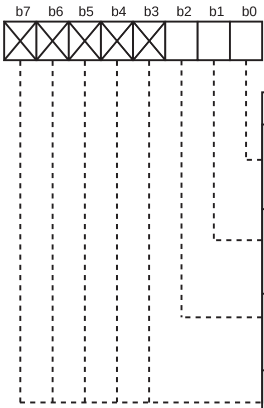
Under development

Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure 1.21 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), system clock control register 1 (address 0007_{16}) and port P4 direction register (address $03EA_{16}$) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P4.

If, after “1” (write-enabled) has been written to the port P4 direction register write-enable bit (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to “0” (write-inhibited). However, the system clock control registers 0 and 1 write-enable bit (bit 0 at $000A_{16}$) and processor mode register 0 and 1 write-enable bit (bit 1 at $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



Symbol	Address	When reset
PRCR	$000A_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P4 direction register (address $03EA_{16}$) (Note)	0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

Note: Writing a value to an address after “1” is written to this bit returns the bit to “0” . Other bits do not automatically return to “0” and they must therefore be reset by the program.

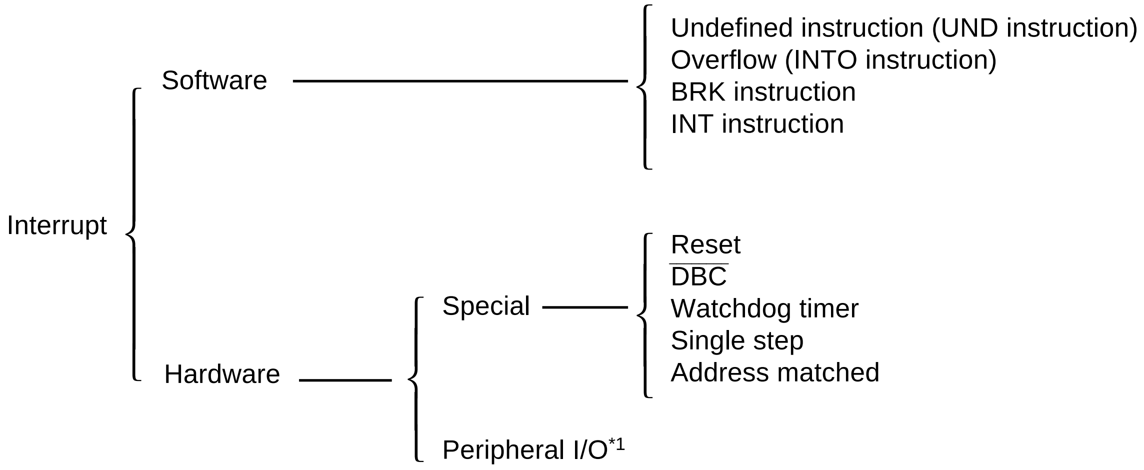
Figure 1.21. Protect register

Under development

Overview of Interrupt

Type of Interrupts

Figure 1.22 lists the types of interrupts.



*1 Peripheral I/O interrupts are generated by the peripheral functions built into the microcomputer system.

Figure 1.22. Classification of interrupts

- Maskable interrupt : An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority can be changed by priority level.
- Non-maskable interrupt : An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority cannot be changed by priority level.

Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

• Undefined instruction interrupt

An undefined instruction interrupt occurs when executing the UND instruction.

• Overflow interrupt

An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to “1”. The following are instructions whose O flag changes by arithmetic:
ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB

• BRK interrupt

A BRK interrupt occurs when executing the BRK instruction.

• INT interrupt

An INT interrupt occurs when assigning one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.

The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.

So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to “0” and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

Under development

Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

(1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**
  Reset occurs if an “L” is input to the RESET pin.
- **DBC interrupt**
  This interrupt is exclusively for the debugger, do not use it in other circumstances.
- **Watchdog timer interrupt**
  Generated by the watchdog timer.
- **Single-step interrupt**
  This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1”, a single-step interrupt occurs after one instruction is executed.
- **Address match interrupt**
  An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”. If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs.

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Key-input interrupt**
  A key-input interrupt occurs if an “L” is input to the $\overline{\text{KI}}$ pin.
- **A-D conversion interrupt**
  This is an interrupt that the A-D converter generates.
- **UART0 and UART1 transmission interrupt**
  These are interrupts that the serial I/O transmission generates.
- **UART0 and UART1 reception interrupt**
  These are interrupts that the serial I/O reception generates.
- **Timer A0 interrupt**
  This is an interrupts that timer A0 generates.
- **Timer B0 and timer B2 interrupt**
  These are interrupts that timer B generates.
- **Timer X0 to timer X2 interrupt**
  These are interrupts that timer X generates.
- **$\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ interrupt**
  An $\overline{\text{INT}}$ interrupt occurs if either a rising edge or a falling edge is input to the $\overline{\text{INT}}$ pin.

Under development

Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure 1.23 shows format for specifying interrupt vector addresses.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

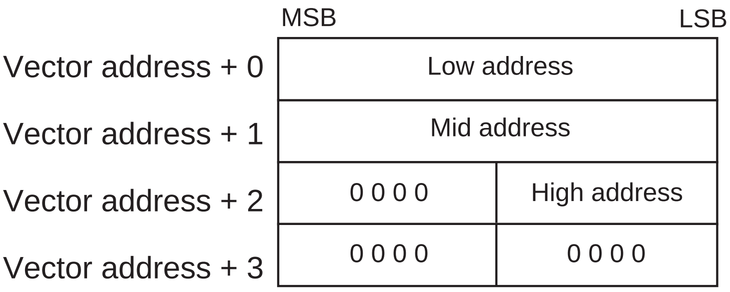


Figure 1.23. Format for specifying interrupt vector addresses

• Fixed vector tables

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFDC$_{16}$ to FFFFF$_{16}$. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 1.7 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 1.7. Interrupt and fixed vector address

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFDC$_{16}$ to FFFDF$_{16}$	Interrupt on UND instruction
Overflow	FFFE0$_{16}$ to FFFE3$_{16}$	Interrupt on INTO instruction
BRK instruction	FFFE4$_{16}$ to FFFE7$_{16}$	If the vector is filled with FF$_{16}$, program execution starts from the address shown by the vector in the variable vector table
Address match	FFFE8$_{16}$ to FFFEB$_{16}$	There is an address-matching interrupt enable bit
Single step (Note)	FFFEC$_{16}$ to FFFEF$_{16}$	Do not use
Watchdog timer	FFFF0$_{16}$ to FFFF3$_{16}$	
$\overline{\text{DBC}}$ (Note)	FFFF4$_{16}$ to FFFF7$_{16}$	Do not use
-	FFFF8$_{16}$ to FFFFB$_{16}$	-
Reset	FFFFC$_{16}$ to FFFFF$_{16}$	

Note: Interrupts used for debugging purposes only.

Under development

• Variable vector tables

The addresses in the variable vector table can be modified, according to the user’s settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 1.8 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 1.8. Interrupt causes (variable interrupt vector addresses)

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note)	BRK instruction	Cannot be masked by I flag
——		——	
Software interrupt number 11	+44 to +47 (Note)	——	
Software interrupt number 12	+48 to +51 (Note)	——	
Software interrupt number 13	+52 to +55 (Note)	Key input interrupt	
Software interrupt number 14	+56 to +59 (Note)	A-D	
——		——	
Software interrupt number 17	+68 to +71 (Note)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note)	Timer A0	
Software interrupt number 22	+88 to +91 (Note)	Timer X0	
Software interrupt number 23	+92 to +95 (Note)	Timer X1	
Software interrupt number 24	+96 to +99 (Note)	Timer X2	
Software interrupt number 25	+100 to +103 (Note)	——	
Software interrupt number 26	+104 to +107 (Note)	Timer B0	
Software interrupt number 27	+108 to +111 (Note)	Timer B1	
Software interrupt number 28	+112 to +115 (Note)	——	
Software interrupt number 29	+116 to +119 (Note)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note)	——	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked by I flag

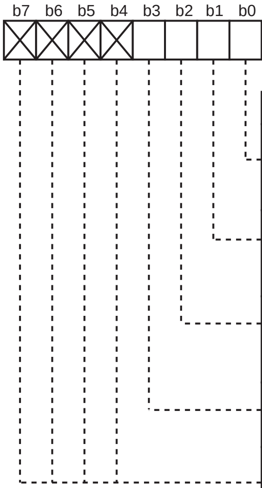
Note : Address relative to address in interrupt table register (INTB).

Interrupt Control

Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.
Enable or disable a maskable interrupt using the interrupt enable flag (I flag), interrupt priority level select bit, and processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).
Figure 1.24 shows the interrupt control registers.

Interrupt control register

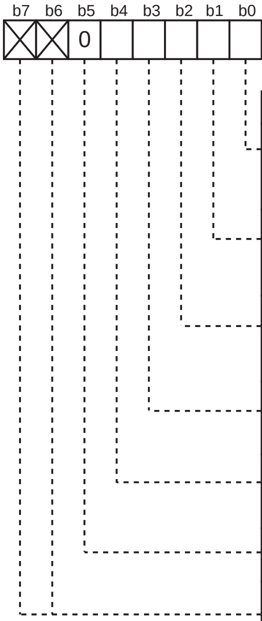
Symbol	Address	When reset
KUPIC	004D$_{16}$	XXXXX000$_2$
ADIC	004E$_{16}$	XXXXX000$_2$
SiTIC(i=0, 1)	0051$_{16}$, 0053$_{16}$	XXXXX000$_2$
SiRIC(i=0, 1)	0052$_{16}$, 0054$_{16}$	XXXXX000$_2$
TAiIC(i=0)	0055$_{16}$	XXXXX000$_2$
TXiIC(i=0 to 2)	0056$_{16}$ to 0058$_{16}$	XXXXX000$_2$
TBiIC(i=0, 1)	005A$_{16}$, 005B$_{16}$	XXXXX000$_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note)
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Symbol	Address	When reset
INTiIC(i=0, 1)	005D$_{16}$, 005E$_{16}$	XX00X000$_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Figure 1.24. Interrupt control register

Interrupt Enable Flag

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to “1” enables all maskable interrupts; setting it to “0” disables all maskable interrupts. This flag is set to “0” after reset.

Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.
Table 1.9 shows the settings of interrupt priority levels and Table 1.10 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:

- interrupt enable flag (I flag) = 1
- interrupt request bit = 1
- interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

Table 1.9. Settings of interrupt priority levels

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	——
0 0 1	Level 1	Low
0 1 0	Level 2	
0 1 1	Level 3	
1 0 0	Level 4	
1 0 1	Level 5	
1 1 0	Level 6	
1 1 1	Level 7	High

Table 1.10. Interrupt levels enabled according to the contents of the IPL

IPL	Enabled interrupt priority levels
IPL_2 IPL_1 IPL_0 0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

Under development

Changing the Interrupt Control Register

< Program examples >

The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    NOP                     ; Four NOP instructions are required when using HOLD function.
    NOP
    FSET    I               ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    MOV.W   MEM, R0         ; Dummy read.
    FSET    I               ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
    PUSHC   FLG             ; Push Flag register onto stack
    FCLR    I               ; Disable interrupts.
    AND.B   #00h, 0055h     ; Clear TA0IC int. priority level and int. request bit.
    POPC    FLG             ; Enable interrupts.
```

The reason why two NOP instructions or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

If changing the interrupt control register using an instruction other than the instructions listed hear, and if an interrupt occurs associated with this register during execution of the instruction, there can be instances in which the interrupt request bit is not set. To avoid this problem, use one of the instructions given below to change the register.
Following instructions: AND, OR, BCLR or BSET

Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

In the interrupt sequence, the processor carries out the following in sequence given:

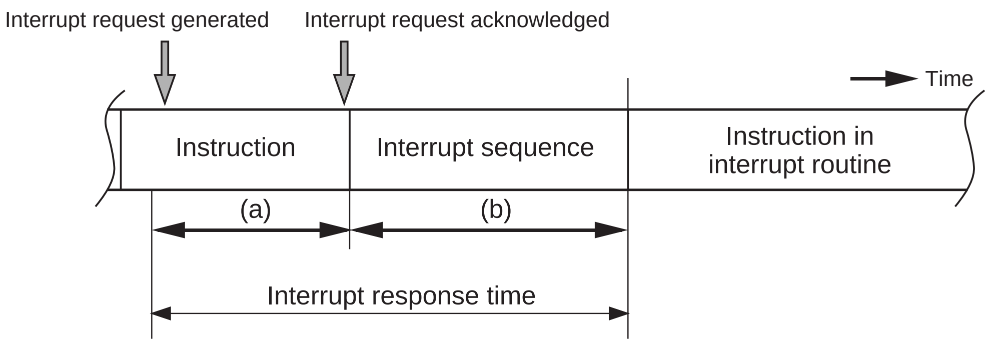
(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000_{16}. After this, the corresponding interrupt request bit becomes "0".
(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to “0” (the U flag, however, does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed).
(4) Saves the content of the temporary register (Note) within the CPU in the stack area.
(5) Saves the content of the program counter (PC) in the stack area.
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

After the interrupt sequence is completed, the processor resumes executing instructions from the first address of the interrupt routine.

Note: This register cannot be utilized by the user.

Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 1.25 shows the interrupt response time.


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
(b)
Interrupt response time

(a) Time from interrupt request is generated to when the instruction then under execution is completed.
(b) Time in which the instruction sequence is executed.

Figure 1.25. Interrupt response time

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction (without wait).

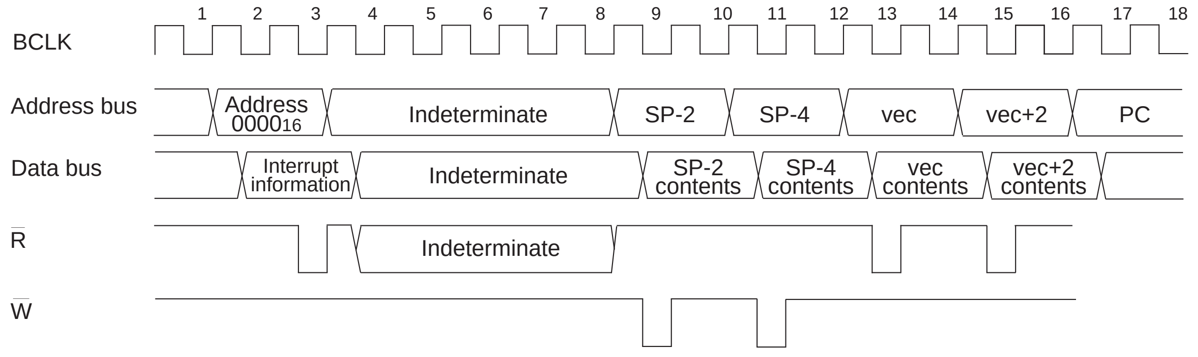
Time (b) is as shown in Table 1.11.

Table 1.11. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-bit bus, without wait	8-bit bus, without wait
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a $\overline{\text{DBC}}$ interrupt; add 1 cycle in the case either of an address match interrupt or of a single-step interrupt.

Note 2: Locate an interrupt vector address in an even address, if possible.



The indeterminate segment is dependent on the queue buffer.
If the queue buffer is ready to take an instruction, a read cycle occurs.

Figure 1.26. Time required for executing the interrupt sequence

Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table 1.12 is set in the IPL.

Table 1.12. Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer	7
Reset	0
Other	Not changed

Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the 4 high-order bits of the program counter, and 4 high-order bits and 8 low-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 low-order bits of the program counter. Figure 1.27 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).

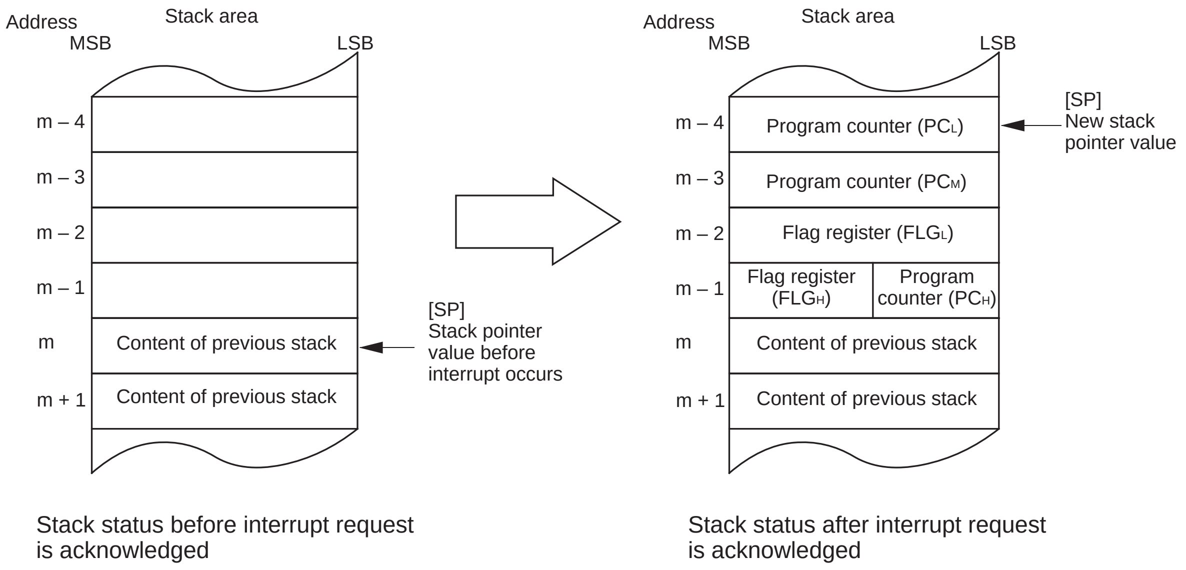
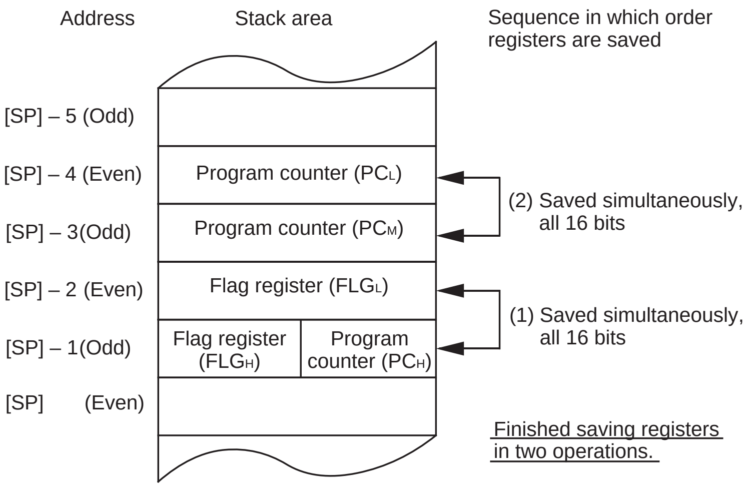

Address
Stack area
MSB
LSB
m – 4
m – 3
m – 2
m – 1
m
m + 1
Content of previous stack
Content of previous stack
[SP]
Stack pointer value before interrupt occurs
Stack status before interrupt request is acknowledged
Address
Stack area
MSB
LSB
m – 4
m – 3
m – 2
m – 1
m
m + 1
Program counter (PCL)
Program counter (PCM)
Flag register (FLGL)
Flag register (FLGH)
Program counter (PCH)
Content of previous stack
Content of previous stack
[SP]
New stack pointer value
Stack status after interrupt request is acknowledged

Figure 1.27. State of stack before and after acceptance of interrupt request

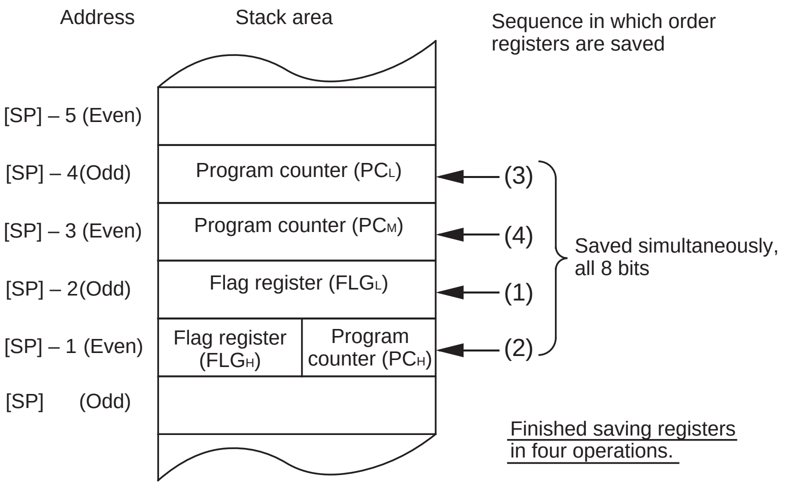
Under development

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer (Note), at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure 1.28 shows the operation of the saving registers.

Note: Stack pointer indicated by U flag.



(1) Stack pointer (SP) contains even number



(2) Stack pointer (SP) contains odd number

Note: [SP] denotes the initial value of the stack pointer (SP) when interrupt request is acknowledged.
After registers are saved, the SP content is [SP] minus 4.

Figure 1.28. Operation of saving registers

Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

Return the other registers saved by software within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted.

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure 1.29 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

Interrupt Priority Level Judge Circuit

This circuit selects the interrupt with the highest priority level when two or more interrupts are generated simultaneously.

Figure 1.30 shows the interrupt resolution circuit.

Under development

Reset > $\overline{\text{DBC}}$ > Watchdog timer > Peripheral I/O > Single step > Address match

Figure 1.29. Hardware interrupts priorities

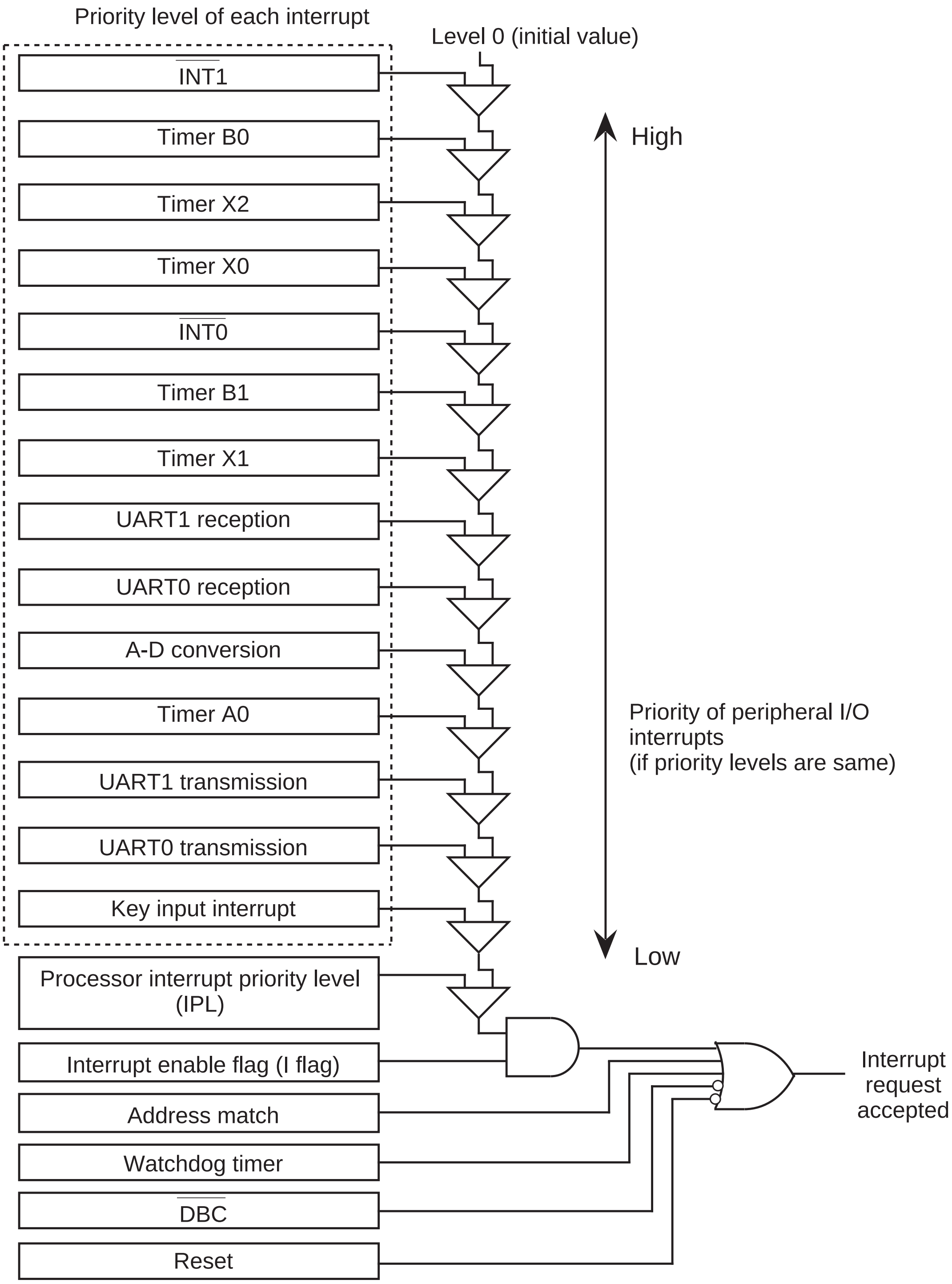


Figure 1.30. Interrupt resolution circuit

Key Input Interrupt

If the direction register of any of $P0_0$ to $P0_7$ is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. Figure 1.31 shows the block diagram of the key input interrupt. Note that if an "L" level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

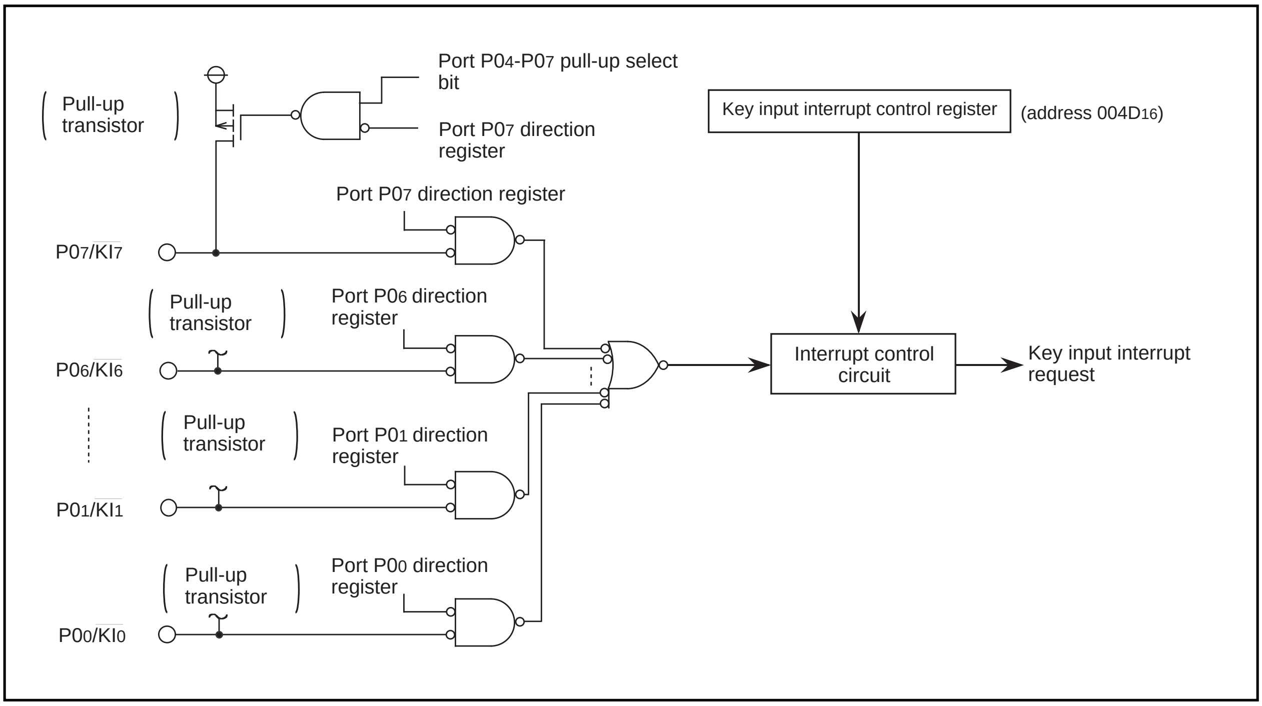

Pull-up transistor
Port P04-P07 pull-up select bit
Port P07 direction register
Key input interrupt control register
(address 004D16)
Port P07 direction register
P07/KI7
Pull-up transistor
Port P06 direction register
P06/KI6
Interrupt control circuit
Key input interrupt request
Pull-up transistor
Port P01 direction register
P01/KI1
Pull-up transistor
Port P00 direction register
P00/KI0

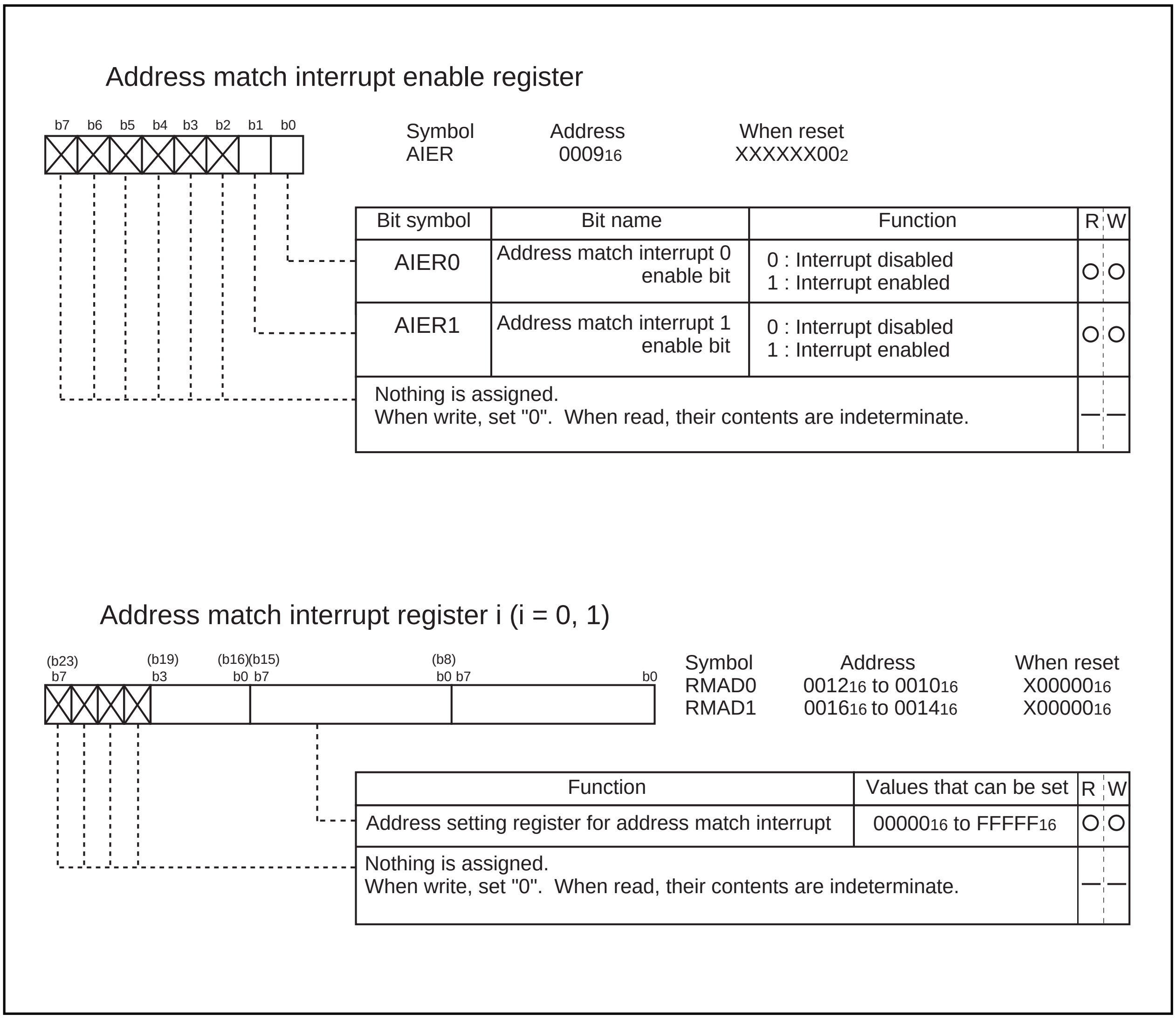
Figure 1.31. Block diagram of key input interrupt

Under development

Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Two address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL).

Figure 1.32 shows the address match interrupt-related registers.



Address match interrupt enable register

Symbol	Address	When reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Address match interrupt register i (i = 0, 1)

Symbol	Address	When reset
RMAD0	0012_{16} to 0010_{16}	$X00000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X00000_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.		—	—

Figure 1.32. Address match interrupt-related registers

Under development

Precautions for Interrupts

(1) Reading address 00000_{16}

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

  The interrupt request bit of the certain interrupt written in address 00000_{16} will then be set to “0”.

  Reading address 00000_{16} by software sets enabled highest priority interrupt source request bit to “0”.

  Though the interrupt is generated, the interrupt routine may not be executed.

  Do not read address 00000_{16} by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000_{16}. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. Concerning the first instruction immediately after reset, generating any interrupts is prohibited.

(3) External interrupt

- Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to pins $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ regardless of the CPU operation clock.
- When changing a polarity of pins $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, the interrupt request bit may become "1". Clear the interrupt request bit after changing the polarity. Figure 1.33 shows the switching condition of $\overline{\text{INT}}$ interrupt request.

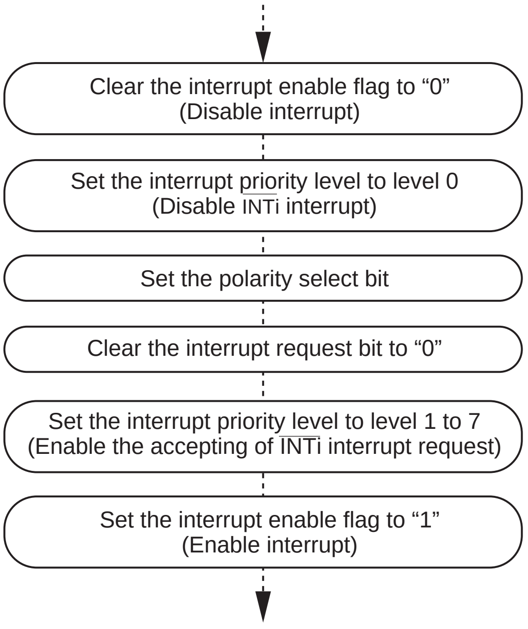

Clear the interrupt enable flag to “0”
(Disable interrupt)
Set the interrupt priority level to level 0
(Disable $\overline{\text{INT}}$i interrupt)
Set the polarity select bit
Clear the interrupt request bit to “0”
Set the interrupt priority level to level 1 to 7
(Enable the accepting of $\overline{\text{INT}}$i interrupt request)
Set the interrupt enable flag to “1”
(Enable interrupt)

Figure 1.33. Switching condition of $\overline{\text{INT}}$ interrupt request

(4) Changing interrupt control register

See "Changing Interrupt Control Register".

Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. A watchdog timer interrupt is generated when an underflow occurs in the watchdog timer. When X_{IN} is selected for the BCLK, bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or by 128). When X_{CIN} is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address $000F_{16}$).

When X_{IN} is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (16 or 128) x watchdog timer count (32768)}}{\text{BCLK}}$$

When X_{CIN} is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (2) x watchdog timer count (32768)}}{\text{BCLK}}$$

For example, when BCLK is 10MHz and the prescaler division ratio is set to 16, the watchdog timer cycle is approximately 52.4 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address $000E_{16}$) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address $000E_{16}$).
Figure 1.34 shows the block diagram of the watchdog timer. Figure 1.35 shows the watchdog timer-related registers.

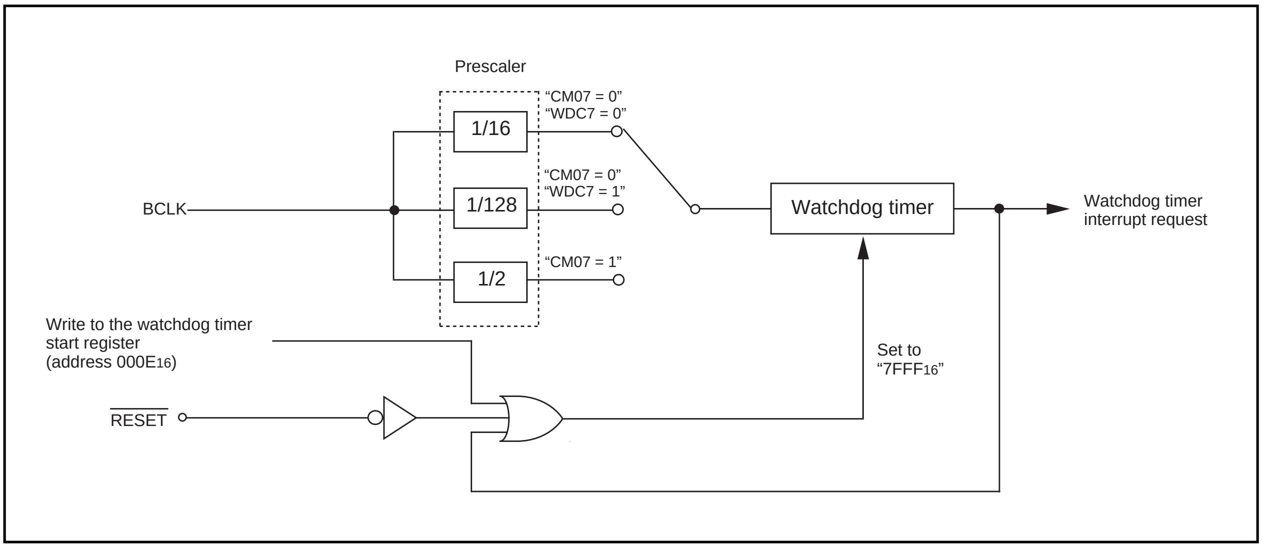
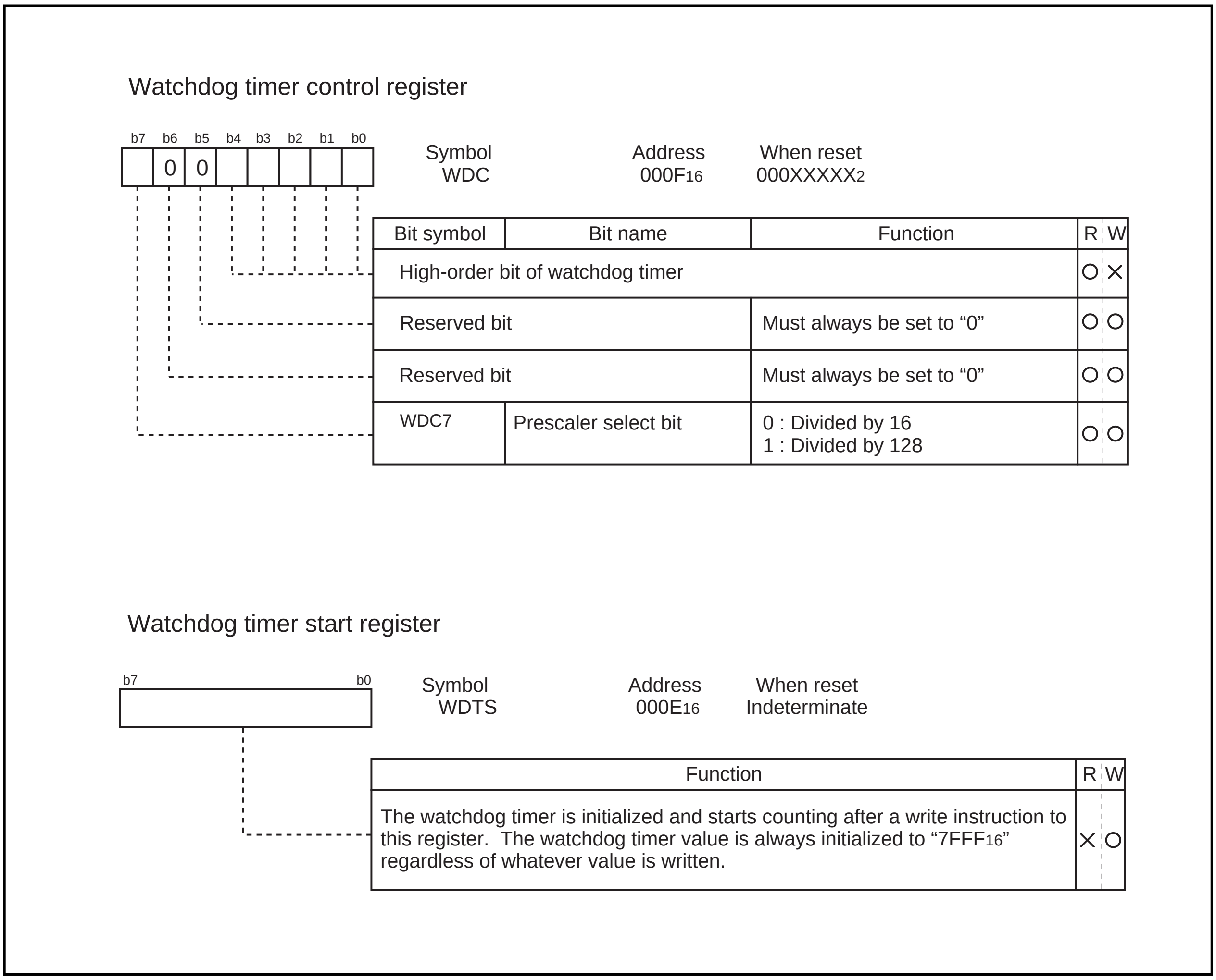


Figure 1.34. Block diagram of watchdog timer

Under development



Watchdog timer control register

Symbol	Address	When reset
WDC	000F$_{16}$	000XXXXX$_2$

Bit symbol	Bit name	Function	R	W
High-order bit of watchdog timer			○	×
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	○	○

Watchdog timer start register

Symbol	Address	When reset
WDTS	000E$_{16}$	Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “7FFF$_{16}$” regardless of whatever value is written.	×	○

Figure 1.35. Watchdog timer control and start registers

Timer

There are six 16-bit timers. These timers can be classified by function into timer A (one), timers B (two) and timers X (three). All these timers function independently. Figure 1.36 show the block diagram of timers.

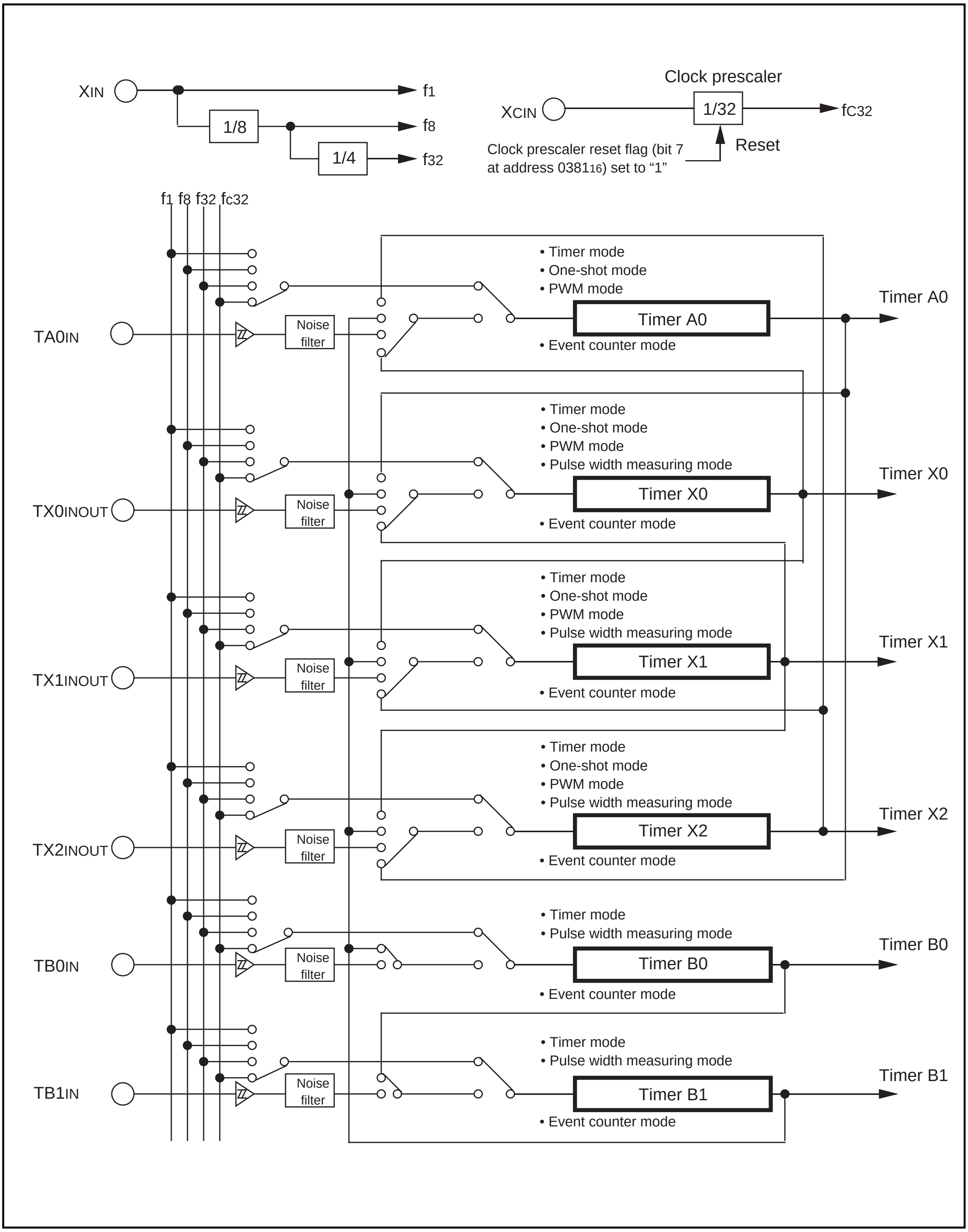


Figure 1.36. Timer block diagram

Under development

Timer A

Figure 1.37 shows the block diagram of timer A. Figures 1.38 to 1.40 show the timer A-related registers.

Use the timer A0 mode register bits 0 and 1 to choose the desired mode.

Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches “0000$_{16}$”.
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

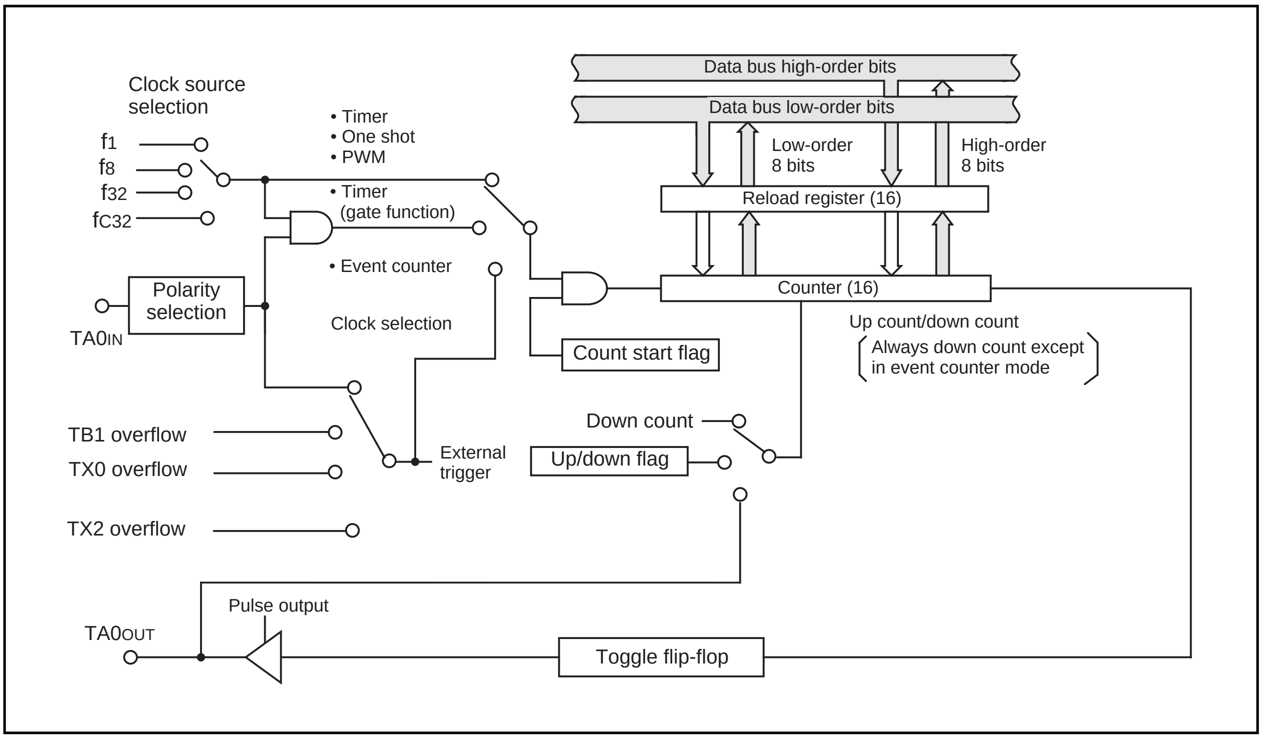


Figure 1.37. Block diagram of timer A

Timer A0 mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: TA0MR
Address: 0396$_{16}$
When reset: 00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 1.38. Timer A-related registers (1)

Timer A0 register (Note)

(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TA0	0387_{16},0386_{16}	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00_{16} to FF_{16} (High-order addresses) 00_{16} to FE_{16} (Low-order addresses)	×	○

Note: Read and write data in 16-bit units.

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Up/down flag

b7 b6 b5 b4 b3 b2 b1 b0

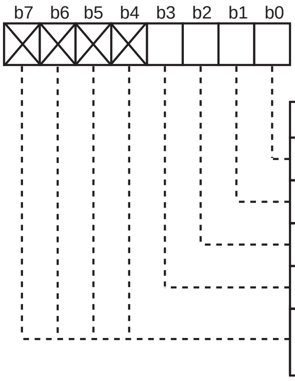
Symbol	Address	When reset
UDF	0384_{16}	$XXX0XXX0_2$

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TA0P	Timer A0 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Figure 1.39. Timer A-related registers (2)

One-shot start flag

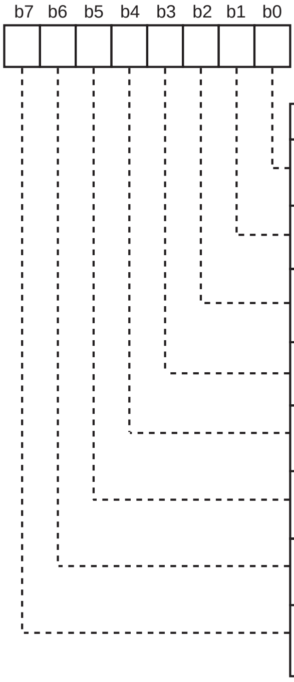
Symbol	Address	When reset
ONSF	0382_{16}	$XXXX0000_2$



Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TX0OS	Timer X0 one-shot start flag		○	○
TX1OS	Timer X1 one-shot start flag		○	○
TX2OS	Timer X2 one-shot start flag		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Trigger select register

Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

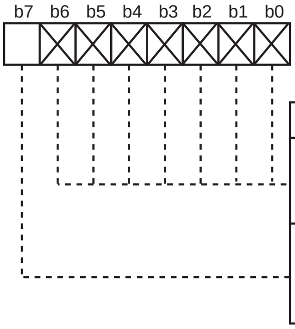


Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0IN is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX2 overflow is selected 1 1 : TX0 overflow is selected	○	○
TA0TGH			○	○
TX0TGL	Timer X0 event/trigger select bit	b3 b2 0 0 : Input on TX0INOUT is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TX1 overflow is selected	○	○
TX0TGH			○	○
TX1TGL	Timer X1 event/trigger select bit	b5 b4 0 0 : Input on TX1INOUT is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX0 overflow is selected 1 1 : TX2 overflow is selected	○	○
TX1TGH			○	○
TX2TGL	Timer X2 event/trigger select bit	b7 b6 0 0 : Input on TX2INOUT is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX1 overflow is selected 1 1 : TA0 overflow is selected	○	○
TX2TGH			○	○

Note: Set the corresponding port direction register to “0”(input mode).

Clock prescaler reset flag

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$



Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.40. Timer A-related registers (3)

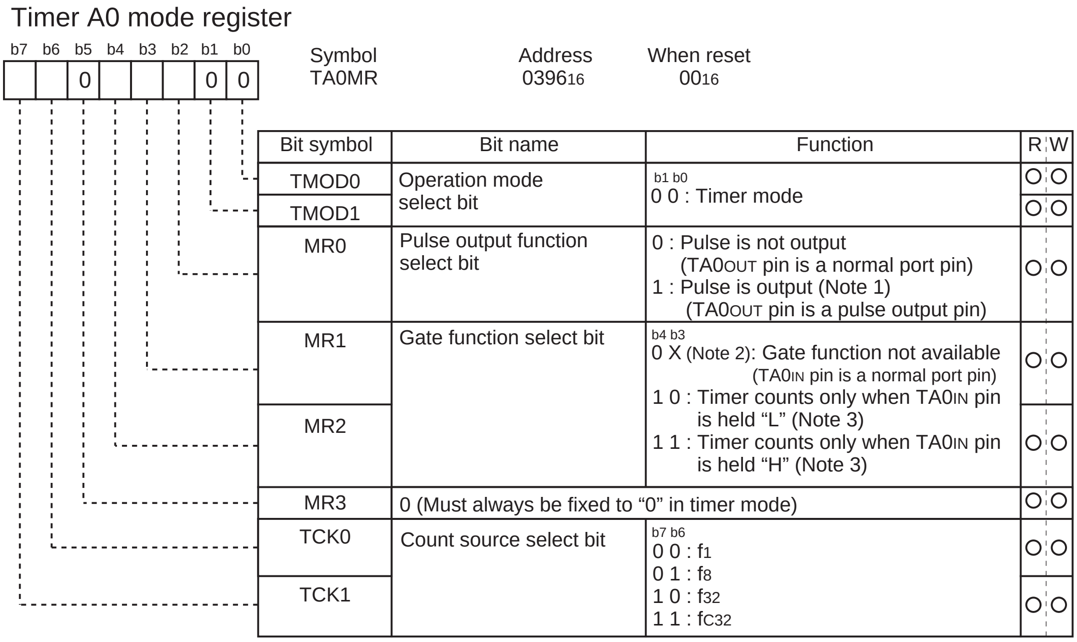
Under development

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.13.) Figure 1.41 shows the timer A0 mode register in timer mode.

Table 1.13. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TA0IN pin function	Programmable I/O port or gate input
TA0OUT pin function	Programmable I/O port or pulse output
Read from timer	Count value can be read out by reading timer A0 register
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TA0IN pin's input signal • Pulse output function Each time the timer underflows, the TA0OUT pin's polarity is reversed



Timer A0 mode register

Symbol TA0MR
Address 0396_{16}
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA0OUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TA0OUT pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X (Note 2): Gate function not available (TA0IN pin is a normal port pin) 1 0 : Timer counts only when TA0IN pin is held "L" (Note 3) 1 1 : Timer counts only when TA0IN pin is held "H" (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to "1" (output mode).
Note 2: The bit can be "0" or "1".
Note 3: Set the corresponding port direction register to "0" (input mode).

Figure 1.41. Timer A0 mode register in timer mode

Under development

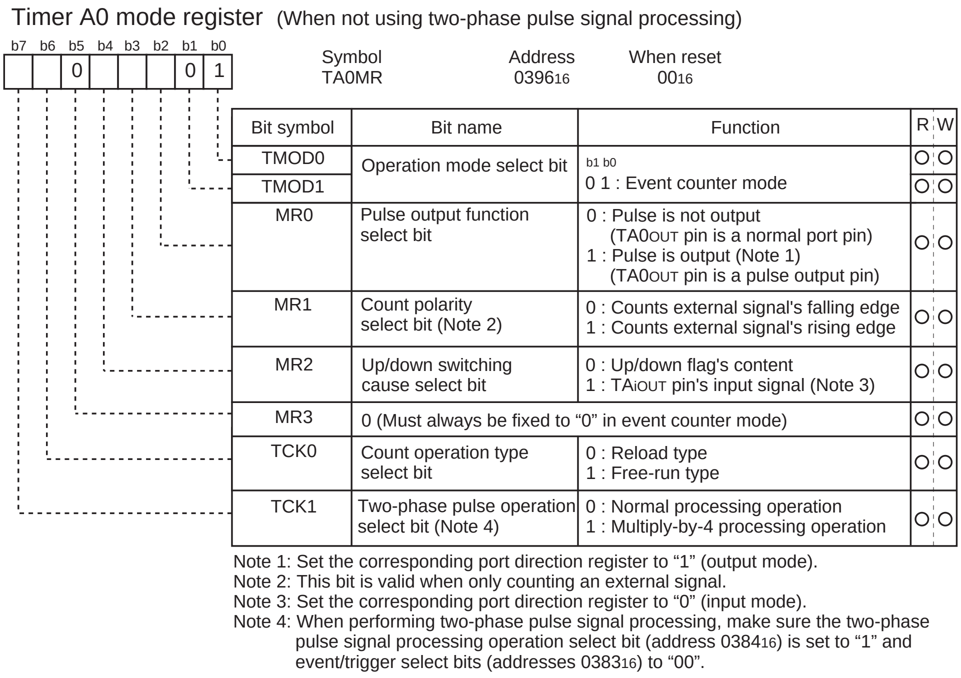
(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timer A0 can count a single-phase and a two-phase external signal. Table 1.14 lists timer specifications when counting a single-phase external signal. Figure 1.42 shows the timer A0 mode register in event counter mode. Table 1.15 lists timer specifications when counting a two-phase external signal. Figure 1.43 shows the timer A0 mode register in event counter mode.

Table 1.14. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TA0IN pin (effective edge can be selected by software) • TB1 overflow, TX0 overflow, TX2 overflow
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register con tents before continuing counting (Note)
Divide ratio	1/ (FFFF16 - n + 1) for up count 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TA0IN pin function	Programmable I/O port or count source input
TA0OUT pin function	Programmable I/O port, pulse output, or up/down count select input
Read from timer	Count value can be read out by reading timer A0 register
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TA0OUT pin's polarity is reversed

Note: This does not apply when the free-run function is selected.


Timer A0 mode register (When not using two-phase pulse signal processing)
b7 b6 b5 b4 b3 b2 b1 b0
0
0 1

Symbol	Address	When reset
TA0MR	039616	0016

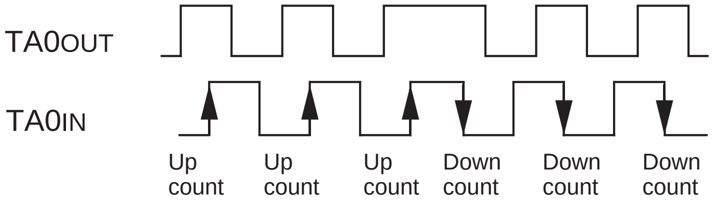
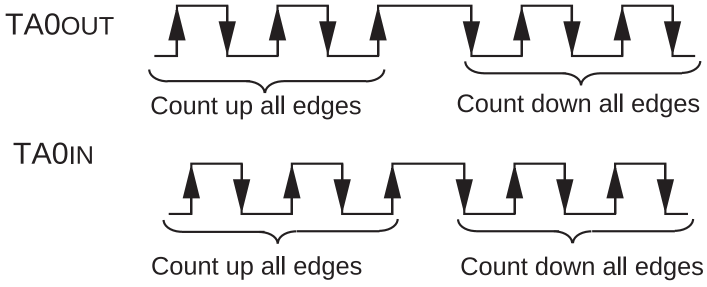
Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA0OUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TA0OUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 2)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAiOUT pin's input signal (Note 3)	○	○
MR3	0 (Must always be fixed to "0" in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse operation select bit (Note 4)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: Set the corresponding port direction register to "1" (output mode).
Note 2: This bit is valid when only counting an external signal.
Note 3: Set the corresponding port direction register to "0" (input mode).
Note 4: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 038416) is set to "1" and event/trigger select bits (addresses 038316) to "00".

Figure 1.42. Timer A0 mode register in event counter mode

Under development

Table 1.15. Timer specifications in event counter mode (when processing two-phase pulse signal)

Item	Specification
Count source	• Two-phase pulse signals input to TA0IN or TA0OUT pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	• 1/ ($FFFF_{16}$ - n + 1) for up count • 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TA0IN pin function	Two-phase pulse input
TA0OUT pin function	Two-phase pulse input
Read from timer	Count value can be read out by reading timer A0 register
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function	• Normal processing operation The timer counts up rising edges or counts down falling edges on the TA0IN pin when input signal on the TA0OUT pin is “H”  TA0OUT TA0IN Up count Up count Up count Down count Down count Down count • Multiply-by-4 processing operation If the phase relationship is such that the TA0IN pin goes “H” when the input signal on the TA0OUT pin is “H”, the timer counts up rising and falling edges on the TA0OUT and TA0IN pins. If the phase relationship is such that the TA0IN pin goes “L” when the input signal on the TA0OUT pin is “H”, the timer counts down rising and falling edges on the TA0OUT and TA0IN pins.  TA0OUT Count up all edges Count down all edges TA0IN Count up all edges Count down all edges

Note: This does not apply when the free-run function is selected.

Under development

Timer A0 mode register
(When using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0	0	0	1

Symbol	Address	When reset
TA0MR	0396_{16}	00_{16}

	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
MR1	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
MR2	1 (Must always be “1” when using two-phase pulse signal processing)		○	○
MR3	0 (Must always be “0” when using two-phase pulse signal processing)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation select bit (Note)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0384_{16}) is set to “1”. Also, always be sure to set the event/trigger select bit (addresses 0383_{16}) to “00”.

Figure 1.43. Timer A0 mode register in event counter mode

Under development

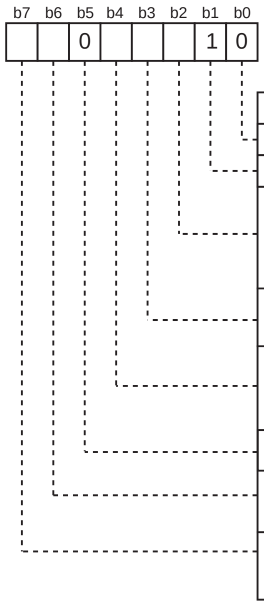
(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.16.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.44 shows the timer A0 mode register in one-shot timer mode.

Table 1.16. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TA0$_{IN}$ pin function	Programmable I/O port or trigger input
TA0$_{OUT}$ pin function	Programmable I/O port or pulse output
Read from timer	When timer A0 register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)

Timer A0 mode register



Symbol TA0MR
Address 0396_{16}
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TA0$_{OUT}$ pin is a normal port pin) 1 : Pulse is output (Note 1) (TA0$_{OUT}$ pin is a pulse output pin)	○	○
MR1	External trigger select bit (Note 2)	0 : Falling edge of TA0$_{IN}$ pin's input signal (Note 3) 1 : Rising edge of TA0$_{IN}$ pin's input signal (Note 3)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register	○	○
MR3	0 (Must always be "0" in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to "1" (output mode).
Note 2: Valid only when the TA0$_{IN}$ pin is selected by the event/trigger select bit (addresses 0383_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 3: Set the corresponding port direction register to "0" (input mode).

Figure 1.44. Timer A0 mode register in one-shot timer mode

Under development

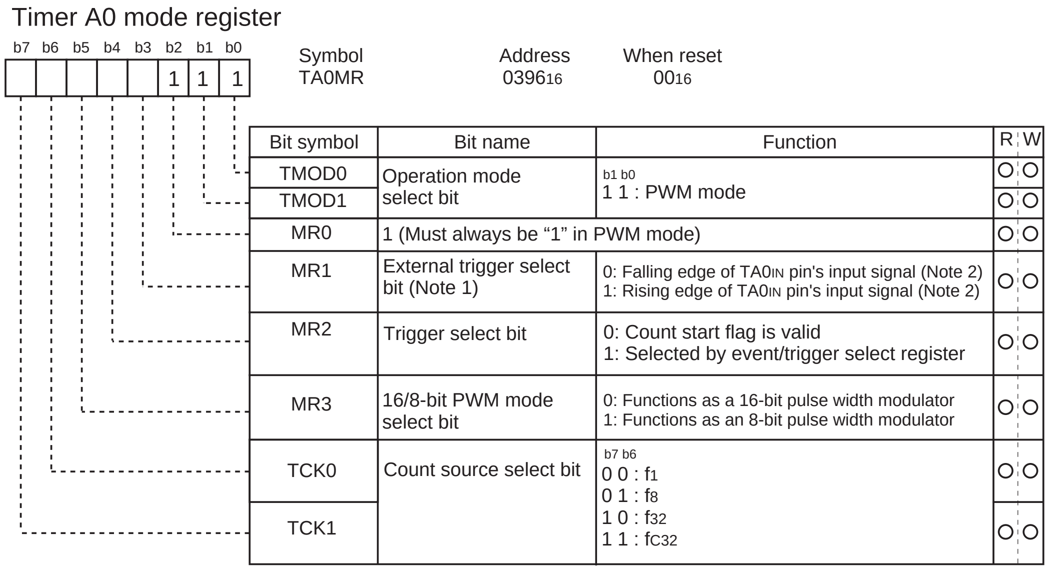
(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.17.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.45 shows the timer A0 mode register in pulse width modulation mode. Figure 1.46 shows the example of how a 16-bit pulse width modulator operates. Figure 1.47 shows the example of how an 8-bit pulse width modulator operates.

Table 1.17. Timer specifications in pulse width modulation mode

Item		Specification
Count source		f_1, f_8, f_{32}, fc_{32}
Count operation		• The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM		• High level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM		• High level width n ×(m+1) / fi n : values set to timer A0 register’s high-order address • Cycle time $(2^{8}-1)$ ×(m+1) / fi m : values set to timer A0 register’s low-order address
Count start condition		• External trigger is input • The timer overflows • The count start flag is set (= 1)
Count stop condition		• The count start flag is reset (= 0)
Interrupt request generation timing	8 bits PWM	• Set value of "H" level width is except FF_{16}, 00_{16} : PWM pulse goes “L” • Set value of "H" level width is FF_{16}, 00_{16} : Timing that count value goes to 01_{16}
	16 bits PWM	• Set value of "H" level width is except $FFFF_{16}$, 0000_{16} : PWM pulse goes “L” • Set value of "H" level width is $FFFF_{16}$, 0000_{16} : Timing that count value goes to 0001_{16}
$TA0_{IN}$ pin function		Programmable I/O port or trigger input
$TA0_{OUT}$ pin function		Pulse output
Read from timer		When timer A0 register is read, it indicates an indeterminate value
Write to timer		• When counting stopped :When a value is written to timer A0 register, it is written to both reload register and counter • When counting in progress : When a value is written to timer A0 register, it is written to only reload register (Transferred to counter at next reload time)

Note: When set value of "H" level width is 00_{16} or 0000_{16}, pulse outputs "L" level and inversion value, FF_{16} or $FFFF_{16}$ is set to timer.



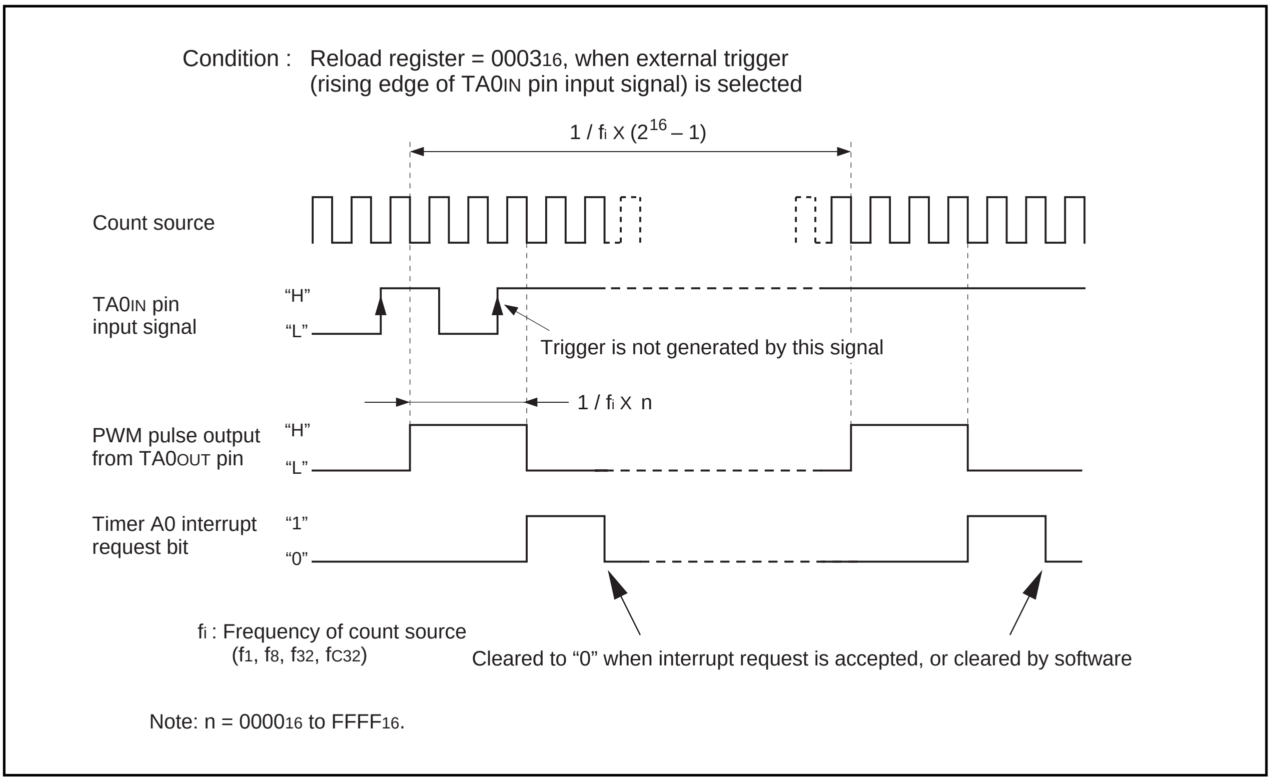
Symbol TA0MR
Address 0396_{16}
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be “1” in PWM mode)		○	○
MR1	External trigger select bit (Note 1)	0: Falling edge of $TA0_{IN}$ pin's input signal (Note 2) 1: Rising edge of $TA0_{IN}$ pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select register	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Note 1: Valid only when the $TA0_{IN}$ pin is selected by the event/trigger select bit (addresses 0383_{16}). If timer overflow is selected, this bit can be “1” or “0”.
Note 2: Set the corresponding port direction register to “0” (input mode).
Note 3: Set the corresponding port direction register to “1” (output mode) when the pulse is output.

Figure 1.45. Timer A0 mode register in pulse width modulation mode

Under development

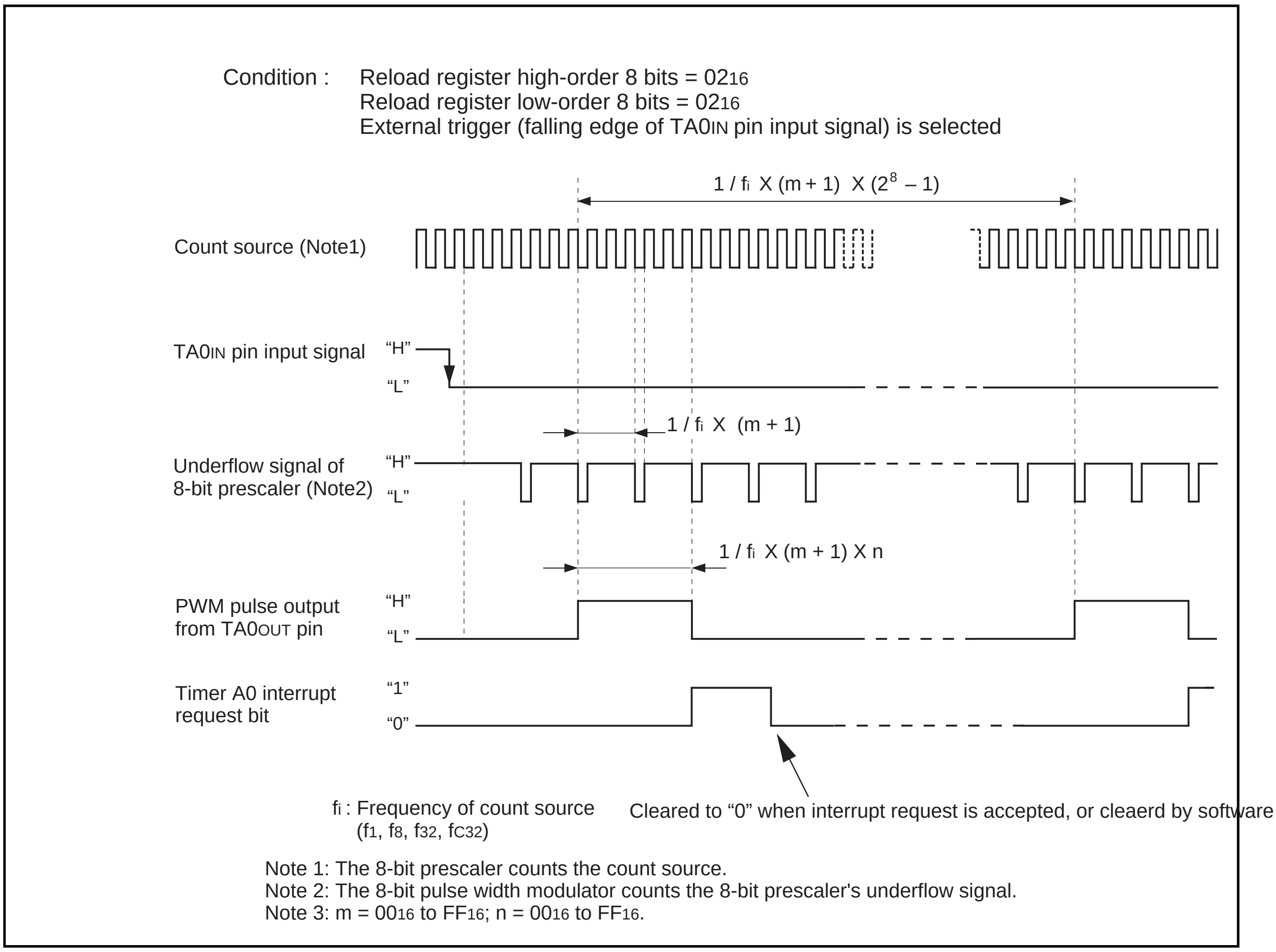

Condition : Reload register = 0003₁₆, when external trigger (rising edge of TA0IN pin input signal) is selected

1 / fi X (2^16 – 1)
Count source
TA0IN pin input signal
"H"
"L"
Trigger is not generated by this signal
1 / fi X n
PWM pulse output from TA0OUT pin
"H"
"L"
Timer A0 interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

fi : Frequency of count source (f_1, f_8, f_{32}, f_{C32})

Note: n = 0000₁₆ to FFFF₁₆.

Figure 1.46. Example of how a 16-bit pulse width modulator operates


Condition : Reload register high-order 8 bits = 02₁₆
Reload register low-order 8 bits = 02₁₆
External trigger (falling edge of TA0IN pin input signal) is selected

1 / fi X (m + 1) X (2^8 – 1)
Count source (Note1)
TA0IN pin input signal
"H"
"L"
1 / fi X (m + 1)
Underflow signal of 8-bit prescaler (Note2)
"H"
"L"
1 / fi X (m + 1) X n
PWM pulse output from TA0OUT pin
"H"
"L"
Timer A0 interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleaerd by software

fi : Frequency of count source (f_1, f_8, f_{32}, f_{C32})

Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00₁₆ to FF₁₆; n = 00₁₆ to FF₁₆.

Figure 1.47. Example of how an 8-bit pulse width modulator operates

Under development

Timer B

Figure 1.48 shows the block diagram of timer B. Figures 1.49 and 1.50 show the timer B-related registers. Use the timer Bi mode register (i = 0, 1) bits 0 and 1 to choose the desired mode.
Timer B has three operation modes listed as follows:

- Timer mode : The timer counts an internal count source.
- Event counter mode : The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode : The timer measures an external signal's pulse period or pulse width.

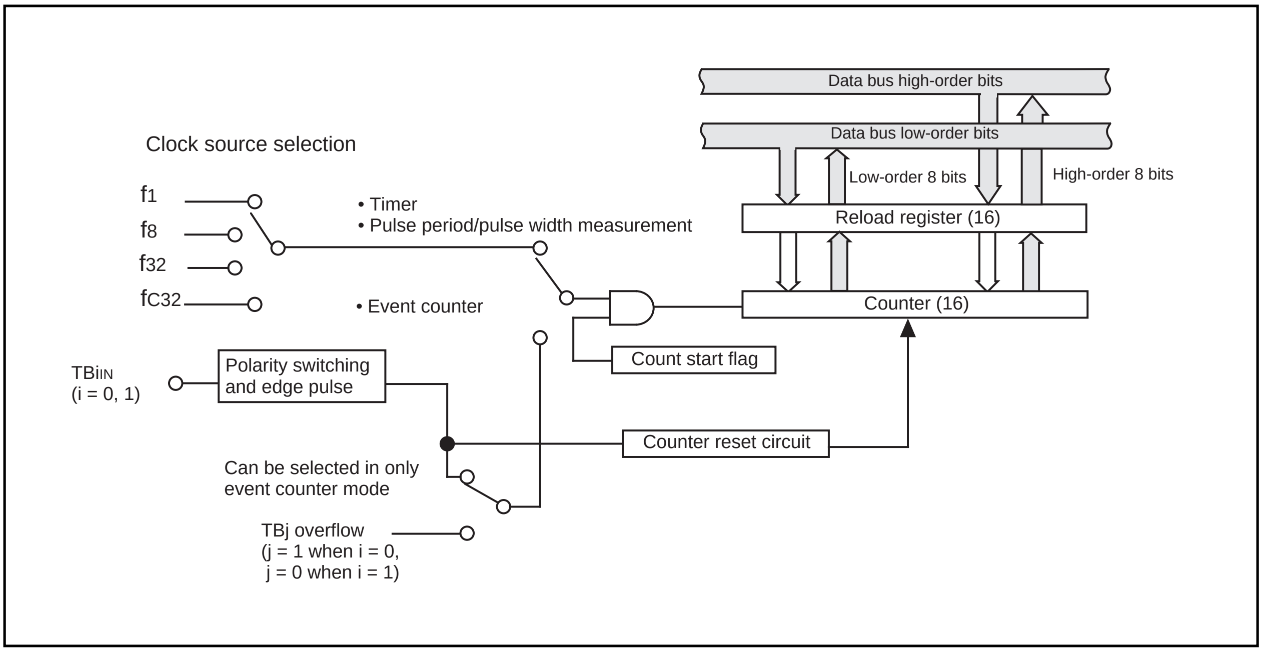


Figure 1.48. Block diagram of timer B

Timer Bi mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TBiMR(i = 0, 1)	$039B_{16}$, $039C_{16}$	$00XX0000_2$

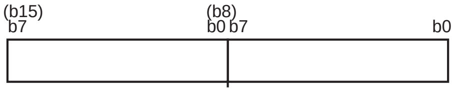
Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1) × (Note 2)	○ ×
MR3			○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Timer B0.
Note 2: Timer B1.

Figure 1.49. Timer B-related registers (1)

Timer Bi register (Note)

Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate

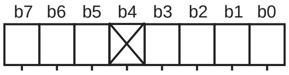


Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000_{16} to $FFFF_{16}$	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

Note1: Read and write data in 16-bit units.

Count start flag

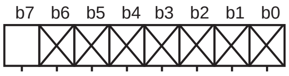
Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$



Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Clock prescaler reset flag

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$



Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.50. Timer B-related registers (2)

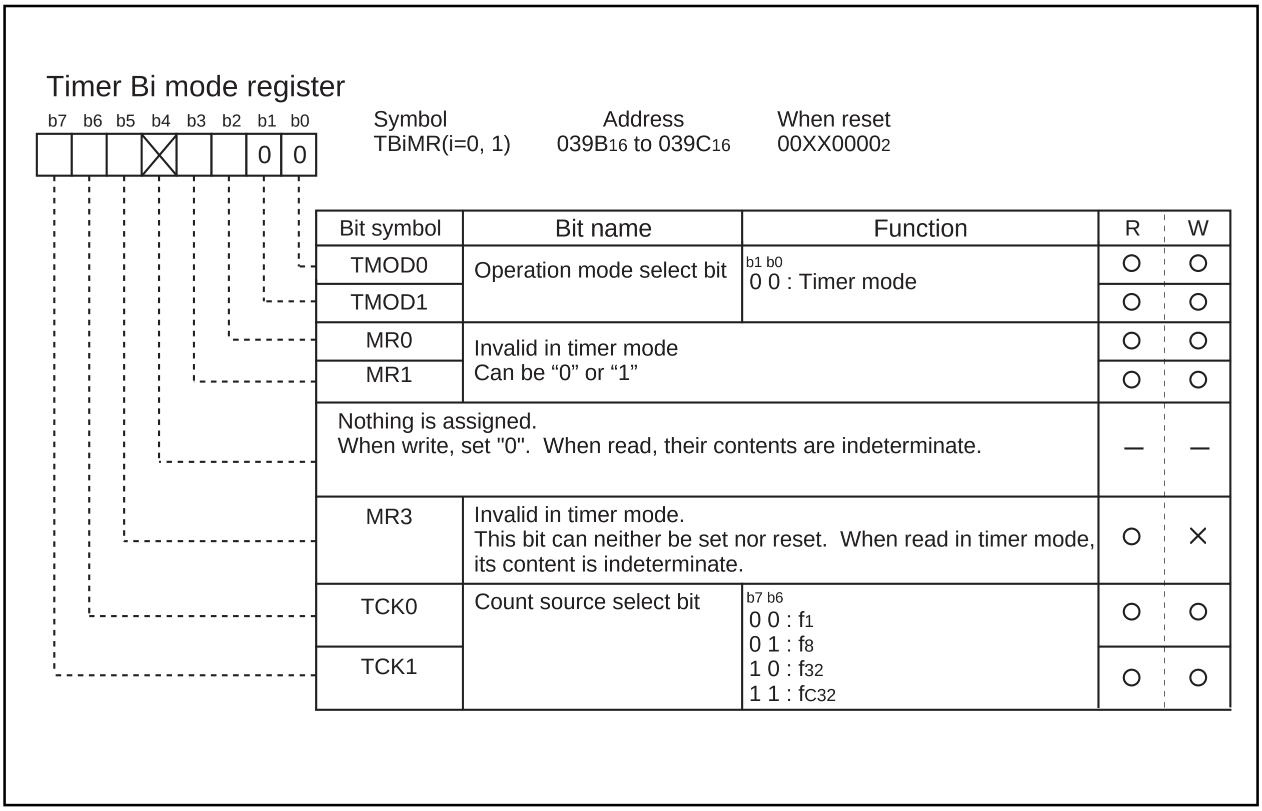
Under development

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.18.) Figure 1.51 shows the timer Bi mode register in timer mode.

Table 1.18. Timer specifications in timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBi_{IN} pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)



Timer Bi mode register

Symbol	Address	When reset
TBiMR(i=0, 1)	$039B_{16}$ to $039C_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be “0” or “1”		○	○
MR1			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
MR3	Invalid in timer mode. This bit can neither be set nor reset. When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Figure 1.51. Timer Bi mode register in timer mode

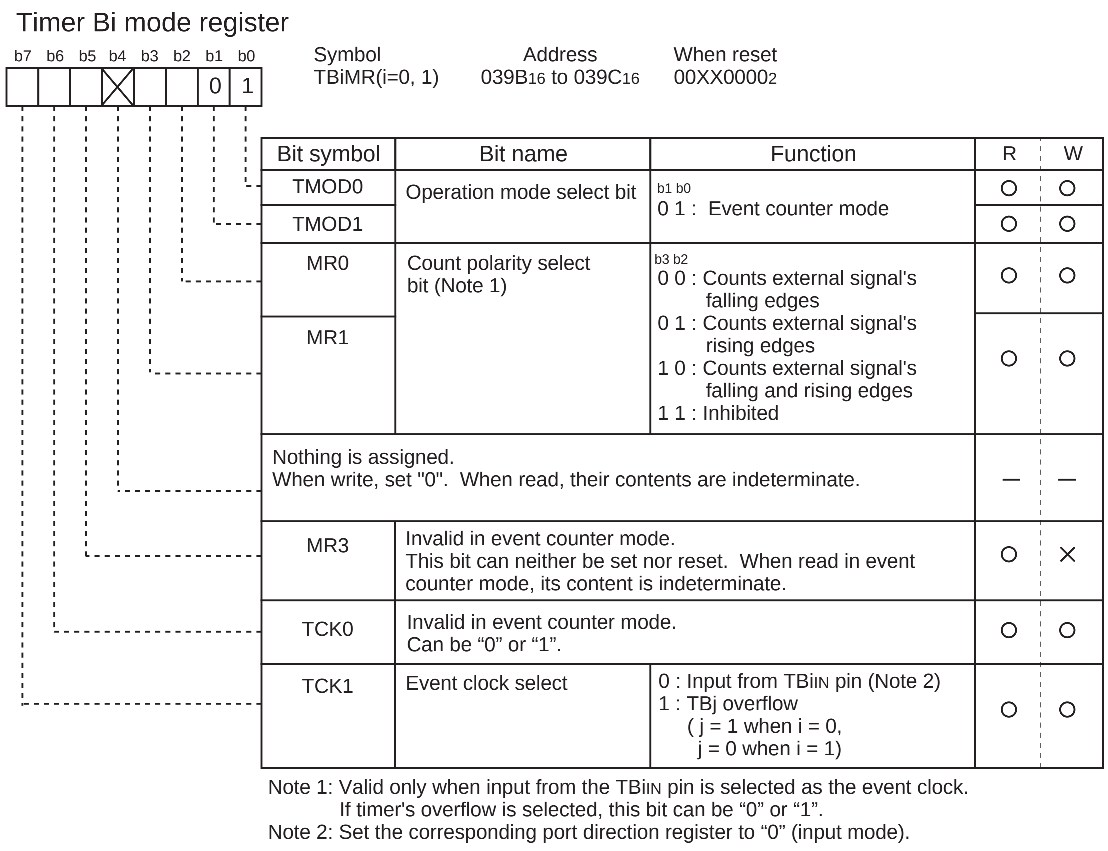
Under development

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 1.19.) Figure 1.52 shows the timer Bi mode register in event counter mode.

Table 1.19. Timer specifications in event counter mode

Item	Specification
Count source	• External signals input to TBi$_{IN}$ pin • Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBi$_{IN}$ pin function	Count source input
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)



Timer Bi mode register

Symbol	Address	When reset
TBiMR(i=0, 1)	$039B_{16}$ to $039C_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Inhibited	○	○
MR1			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
MR3	Invalid in event counter mode. This bit can neither be set nor reset. When read in event counter mode, its content is indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be “0” or “1”.		○	○
TCK1	Event clock select	0 : Input from TBi$_{IN}$ pin (Note 2) 1 : TBj overflow (j = 1 when i = 0, j = 0 when i = 1)	○	○

Note 1: Valid only when input from the TBi$_{IN}$ pin is selected as the event clock.
If timer's overflow is selected, this bit can be “0” or “1”.
Note 2: Set the corresponding port direction register to “0” (input mode).

Figure 1.52. Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

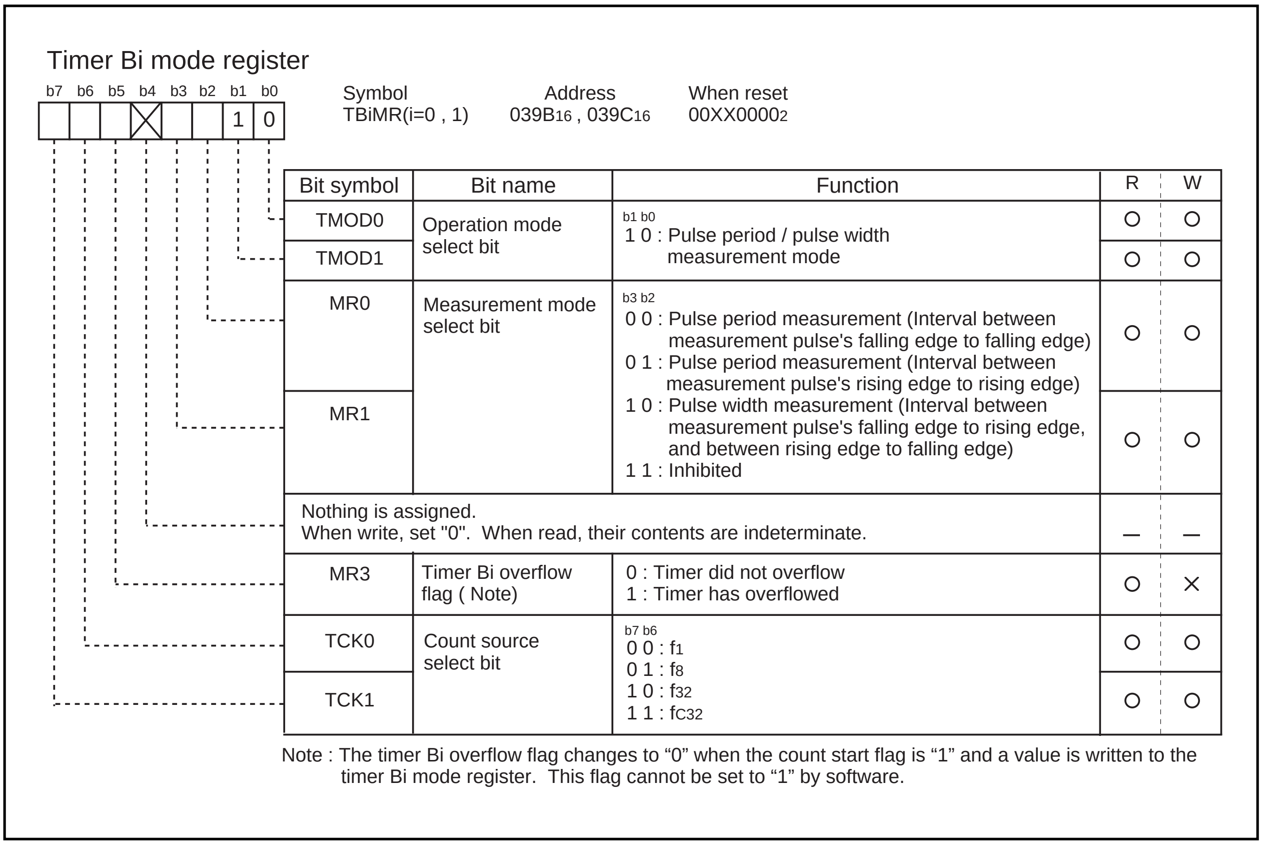
In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 1.20.) Figure 1.53 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 1.54 shows the operation timing when measuring a pulse period. Figure 1.55 shows the operation timing when measuring a pulse width.

Table 1.20. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Up count • Counter value “0000_{16}” is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to “1”. The timer Bi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Bi mode register.)
TBi_{IN} pin function	Measurement pulse input
Read from timer	When timer Bi register is read, it indicates the reload register’s content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.

Note 2: The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer.



Timer Bi mode register

b7 b6 b5 b4 b3 b2 b1 b0
1 0

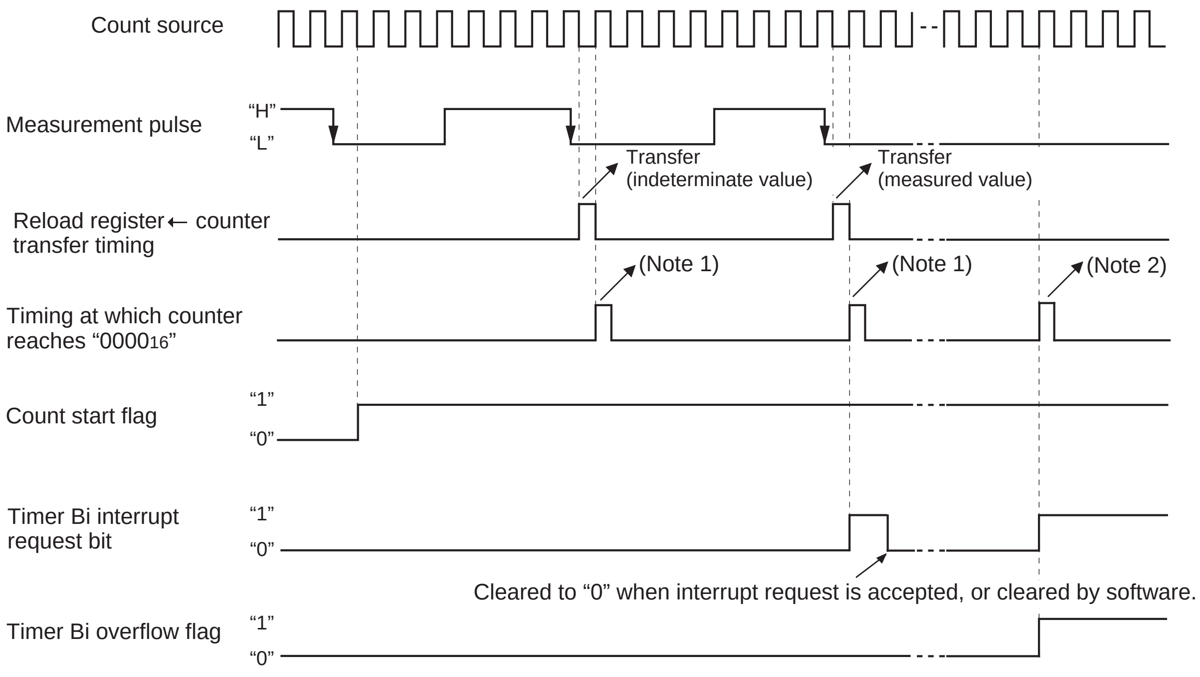
Symbol	Address	When reset
TBiMR(i=0 , 1)	$039B_{16}$, $039C_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–
MR3	Timer Bi overflow flag (Note)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note : The timer Bi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Bi mode register. This flag cannot be set to “1” by software.

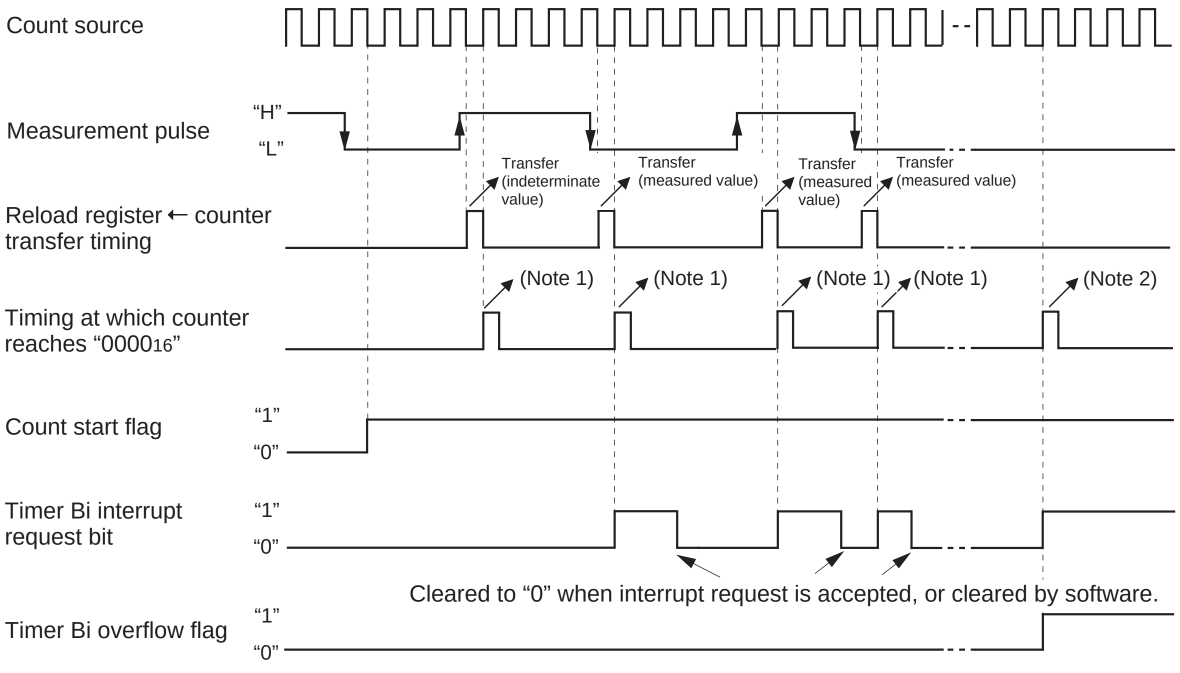
Figure 1.53. Timer Bi mode register in pulse period/pulse width measurement mode

When measuring measurement pulse time interval from falling edge to falling edge


Count source
Measurement pulse
“H”
“L”
Transfer
(indeterminate value)
Transfer
(measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches “0000₁₆”
Count start flag
“1”
“0”
Timer Bi interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software.
Timer Bi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 1.54. Operation timing when measuring a pulse period



Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 1.55. Operation timing when measuring a pulse width

Under development

Timer X

Figure 1.56 shows the block diagram of timer X. Figures 1.57 to 1.59 show the timer X-related registers. Use the timer Xi mode register bits 0 and 1 to choose the desired mode.

Timer X has the five operation modes listed as follows:

- Timer mode : The timer counts an internal count source.
- Event counter mode : The timer counts pulses from an external source or a timer overflow.
- One-shot timer mode : The timer stops counting when the count reaches “0000_{16}”.
- Pulse period/pulse width measuring mode : The timer measures an external signal's pulse period or pulse width.
- Pulse width modulation (PWM) mode : The timer outputs pulses of a given width.

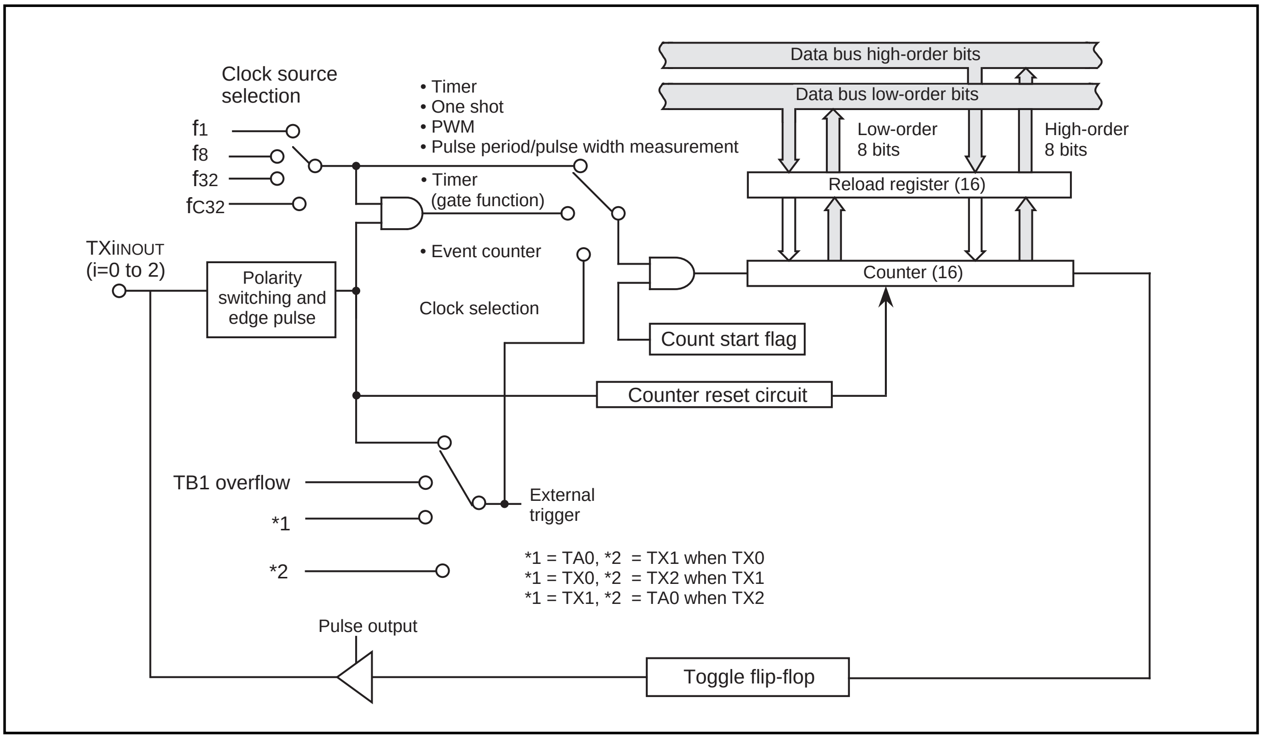


Figure 1.56. Block diagram of timer X

Timer Xi mode register

b7	b6	b5	b4	b3	b2	b1	b0

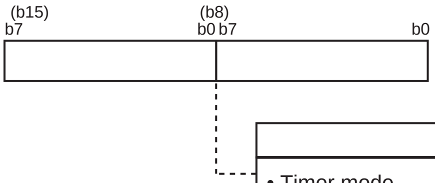
Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397_{16} to 0399_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode or pulse period/ pulse width measurement mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 1.57. Timer X-related registers (1)

Under development

Timer Xi register (Note)

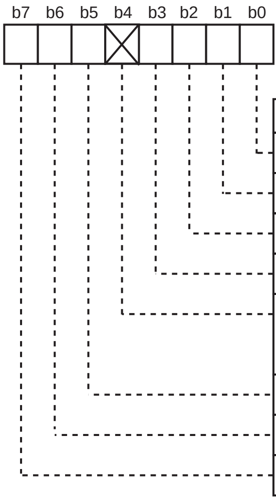


Symbol	Address	When reset
TX0	0389_{16},0388_{16}	Indeterminate
TX1	$038B_{16}$,$038A_{16}$	Indeterminate
TX2	$038D_{16}$,$038C_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	———	○	×
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00_{16} to FF_{16} (High-order addresses) 00_{16} to FF_{16} (Low-order addresses)	×	○

Note: Read and write data in 16-bit units.

Count start flag

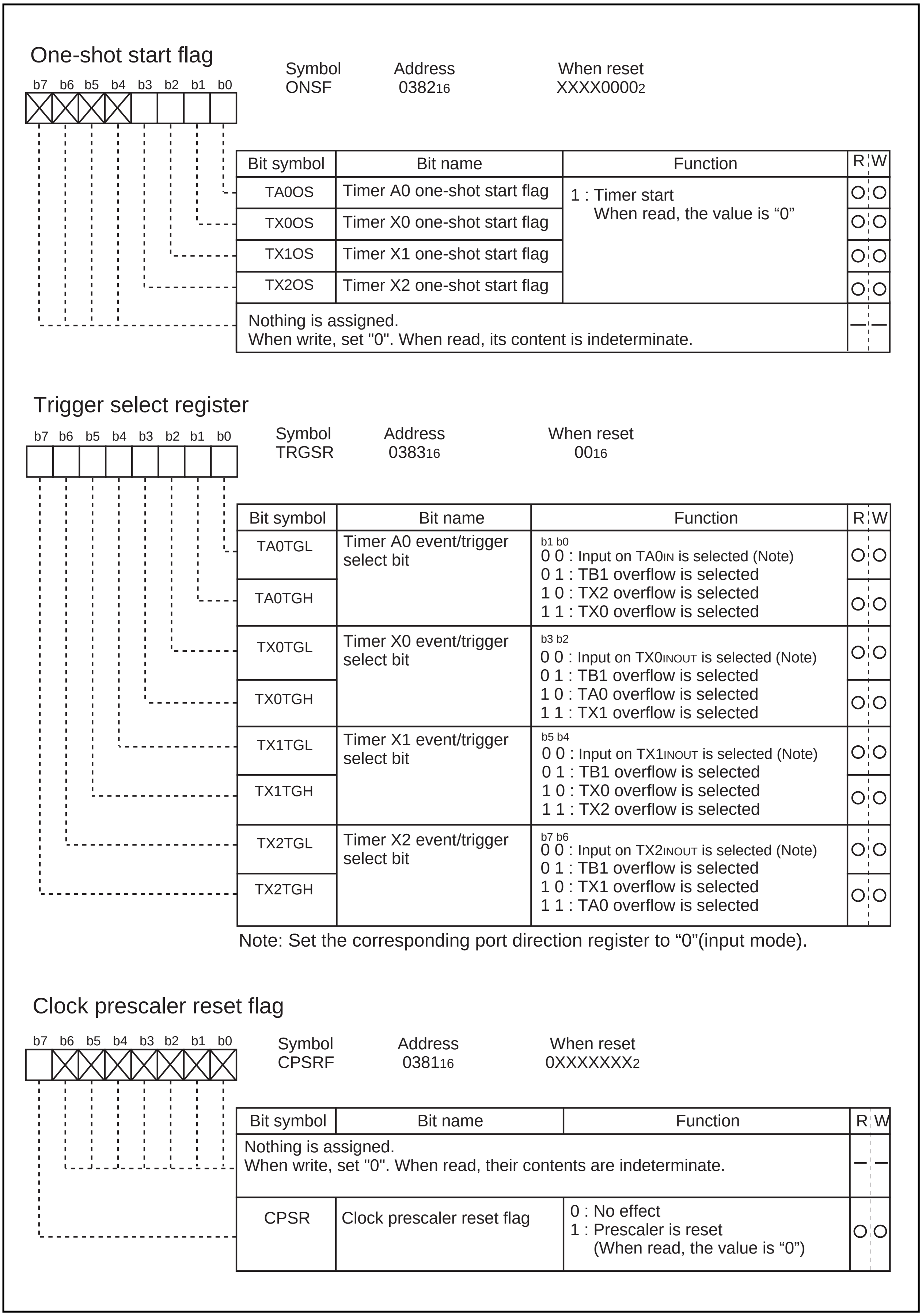


Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0" When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Figure 1.58. Timer X-related registers (2)

Under development



One-shot start flag

Symbol	Address	When reset
ONSF	0382_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TX0OS	Timer X0 one-shot start flag		○	○
TX1OS	Timer X1 one-shot start flag		○	○
TX2OS	Timer X2 one-shot start flag		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Trigger select register

Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0$_{IN}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX2 overflow is selected 1 1 : TX0 overflow is selected	○	○
TA0TGH			○	○
TX0TGL	Timer X0 event/trigger select bit	b3 b2 0 0 : Input on TX0$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TX1 overflow is selected	○	○
TX0TGH			○	○
TX1TGL	Timer X1 event/trigger select bit	b5 b4 0 0 : Input on TX1$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX0 overflow is selected 1 1 : TX2 overflow is selected	○	○
TX1TGH			○	○
TX2TGL	Timer X2 event/trigger select bit	b7 b6 0 0 : Input on TX2$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX1 overflow is selected 1 1 : TA0 overflow is selected	○	○
TX2TGH			○	○

Note: Set the corresponding port direction register to “0”(input mode).

Clock prescaler reset flag

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.59. Timer X-related registers (3)

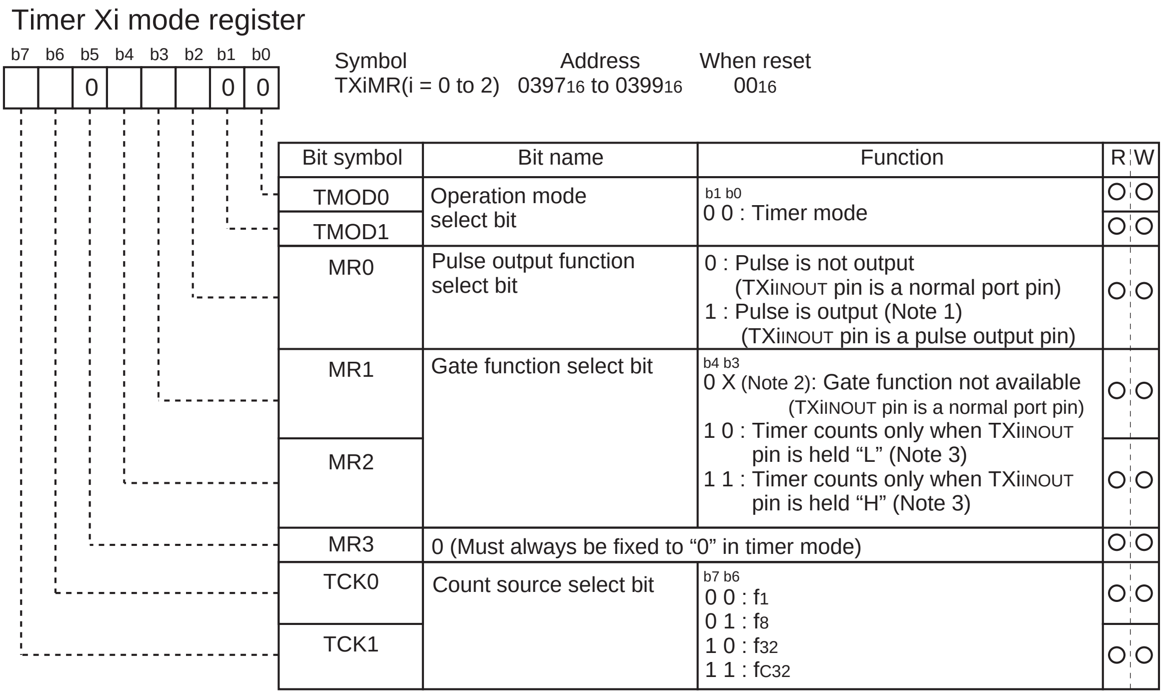
Under development

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.21.) Figure 1.60 shows the timer Xi mode register in timer mode.

Table 1.21. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TXiINOUT pin function	Programmable I/O port, gate input or pulse output
Read from timer	Count value can be read out by reading timer Xi register
Write to timer	• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TXiINOUT pin's input signal • Pulse output function Each time the timer underflows, the TXiINOUT pin's polarity is reversed



Timer Xi mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	0

Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397_{16} to 0399_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TXiINOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TXiINOUT pin is a pulse output pin)	○	○
MR1	Gate function select bit	b4 b3 0 X (Note 2): Gate function not available (TXiINOUT pin is a normal port pin) 1 0 : Timer counts only when TXiINOUT pin is held "L" (Note 3) 1 1 : Timer counts only when TXiINOUT pin is held "H" (Note 3)	○	○
MR2			○	○
MR3	0 (Must always be fixed to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to "1" (output mode). Gate function cannot be selected when pulse output function is selected.
Note 2: The bit can be "0" or "1".
Note 3: Set the corresponding port direction register to "0" (input mode). Pulse output function cannot be selected when gate function is selected.

Figure 1.60. Timer Xi mode register in timer mode

Under development

(2) Event counter mode

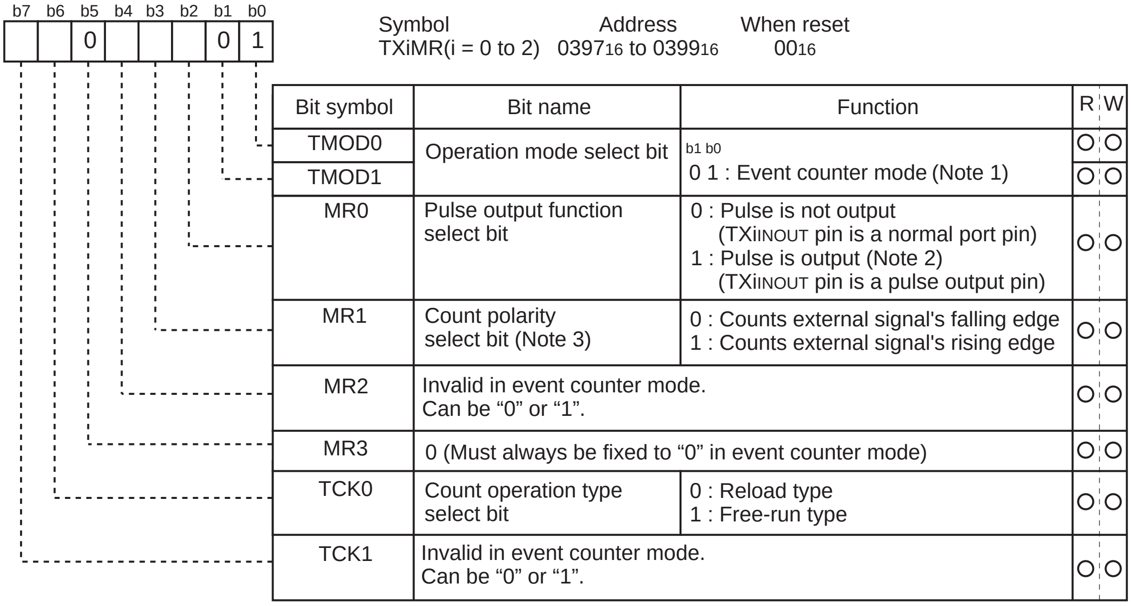
In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 1.22.) Figure 1.61 shows the timer Xi mode register in event counter mode.

Table 1.22. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TXiINOUT pin (effective edge can be selected by software) • TB1 overflow, TA0 overflow, TXi overflow
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting (Note)
Divide ratio	1/ (n + 1)　　n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TXiINOUT pin function	Programmable I/O port, count source input or pulse output
Read from timer	Count value can be read out by reading timer Xi register
Write to timer	• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer underflows, the TXiINOUT pin's polarity is reversed

Note: This does not apply when the free-run function is selected.

Timer Xi mode register



Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397_{16} to 0399_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode (Note 1)	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TXiINOUT pin is a normal port pin) 1 : Pulse is output (Note 2) (TXiINOUT pin is a pulse output pin)	○	○
MR1	Count polarity select bit (Note 3)	0 : Counts external signal's falling edge 1 : Counts external signal's rising edge	○	○
MR2	Invalid in event counter mode. Can be “0” or “1”.		○	○
MR3	0 (Must always be fixed to “0” in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Invalid in event counter mode. Can be “0” or “1”.		○	○

Note 1: Count source is selected by event/trigger select bit(address 0383_{16}) in event counter mode.
Note 2: Set the corresponding port direction register to “1” (output mode). TXiINOUT pin input is not selected as count source when pulse output function is selected.
Note 3: This bit is valid when only counting an external signal.

Figure 1.61. Timer Xi mode register in event counter mode

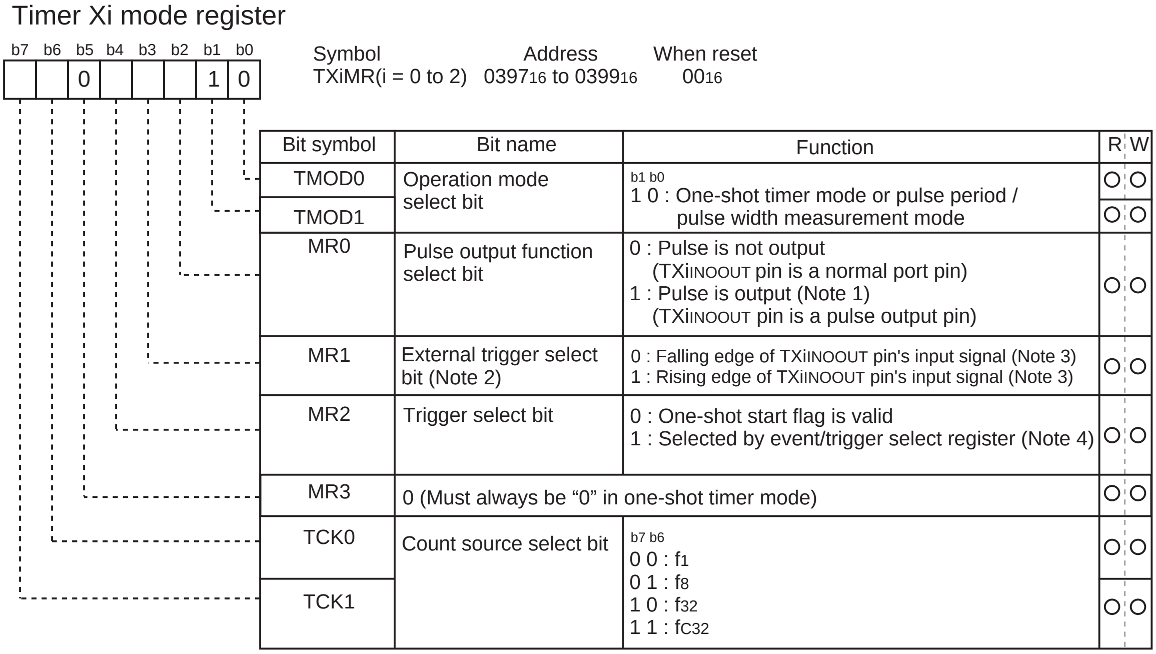
Under development

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.23.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.62 shows the timer Xi mode register in one-shot timer mode.

Table 1.23. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TXiINOUT pin function	Programmable I/O port, trigger input or pulse output
Read from timer	When timer Xi register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)



Symbol TXiMR(i = 0 to 2) Address 0397_{16} to 0399_{16} When reset 00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode or pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Pulse output function select bit	0 : Pulse is not output (TXiINOOUT pin is a normal port pin) 1 : Pulse is output (Note 1) (TXiINOOUT pin is a pulse output pin)	○	○
MR1	External trigger select bit (Note 2)	0 : Falling edge of TXiINOOUT pin's input signal (Note 3) 1 : Rising edge of TXiINOOUT pin's input signal (Note 3)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register (Note 4)	○	○
MR3	0 (Must always be "0" in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Set the corresponding port direction register to "1" (output mode). External trigger cannot be selected as count start condition when pulse output function is selected.
Note 2: Valid only when the TXiINOUT pin is selected by the event/trigger select bit (addresses 0383_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 3: Set the corresponding port direction register to "0" (input mode).
Note 4: Pulse output function cannot be selected when TXiINOUT pin is selected by the event/trigger select bit (addresses 0383_{16}).

Figure 1.62. Timer Xi mode register in one-shot timer mode

Under development

(4) Pulse period/pulse width measurement mode

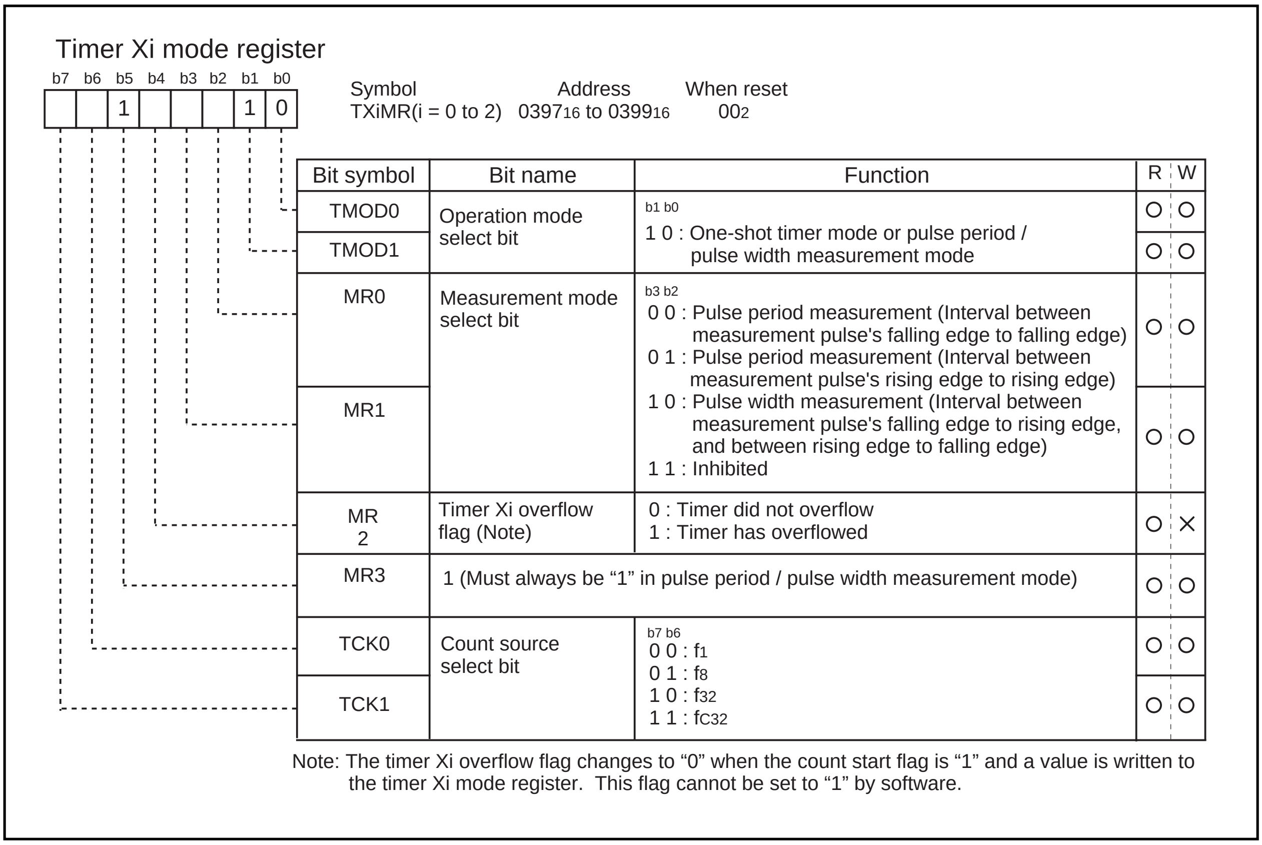
In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 1.24.) Figure 1.63 shows the timer Xi mode register in pulse period/pulse width measurement mode. Figure 1.64 shows the operation timing when measuring a pulse period. Figure 1.65 shows the operation timing when measuring a pulse width.

Table 1.24. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• Up count • Counter value "0000_{16}" is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • When an overflow occurs. (Simultaneously, the timer Xi overflow flag changes to "1". The timer Xi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Xi mode register.)
TXiINOUT pin function	Measurement pulse input
Read from timer	When timer Xi register is read, it indicates the reload register's content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.

Note 2: The value read out from the timer Xi register is indeterminate until the second effective edge is input after the timer.



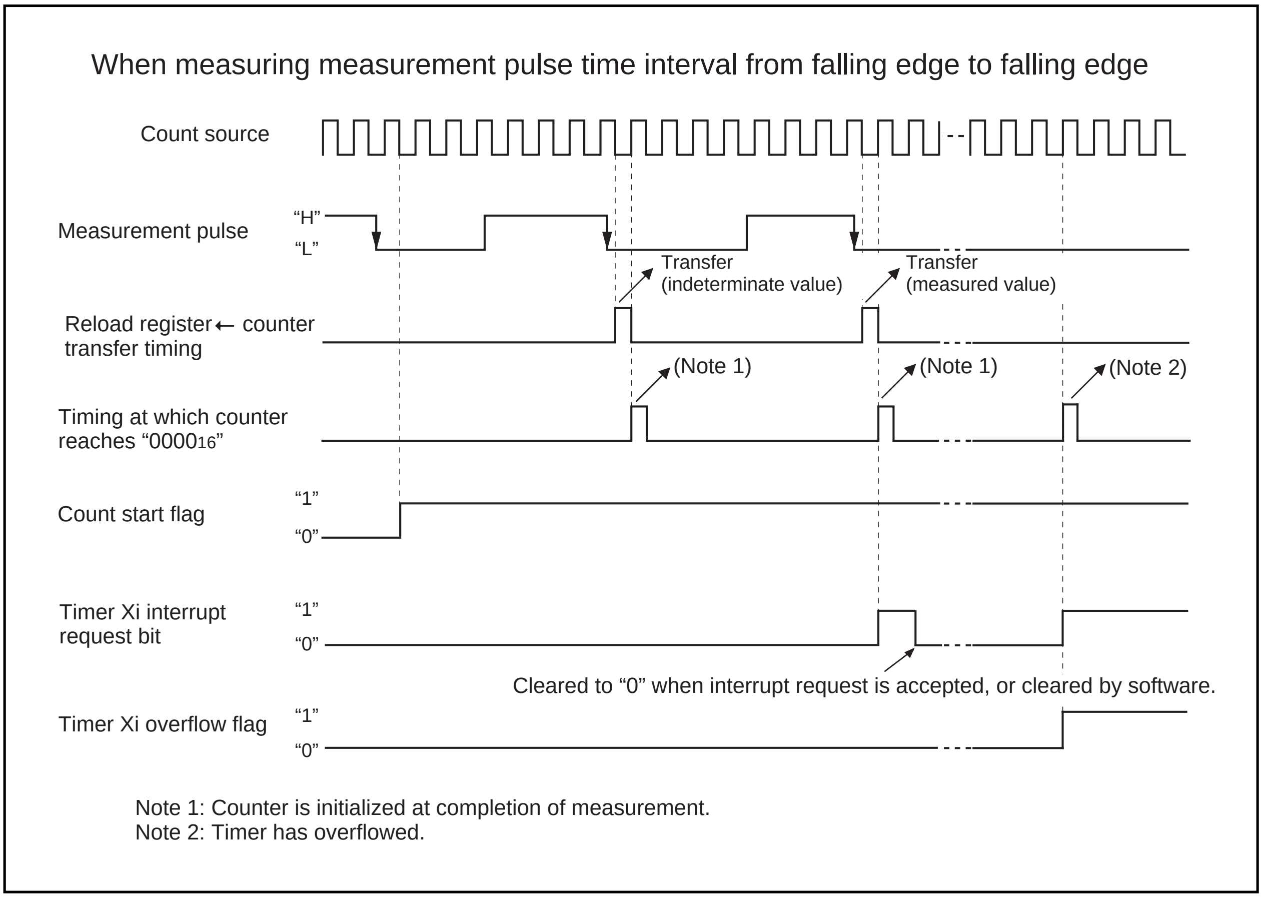
Symbol TXiMR(i = 0 to 2) Address 0397_{16} to 0399_{16} When reset 00_2

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode or pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
MR 2	Timer Xi overflow flag (Note)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
MR3	1 (Must always be "1" in pulse period / pulse width measurement mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note: The timer Xi overflow flag changes to "0" when the count start flag is "1" and a value is written to the timer Xi mode register. This flag cannot be set to "1" by software.

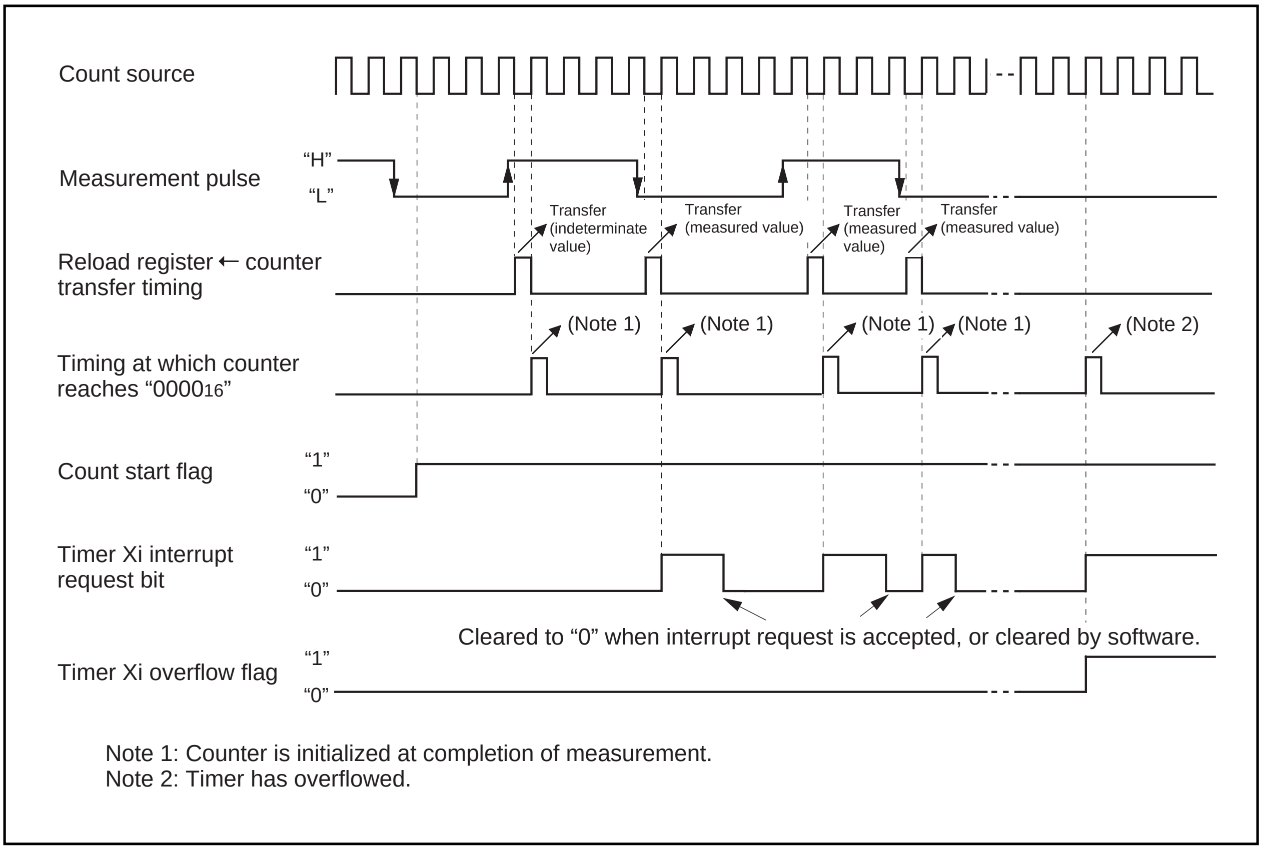
Figure 1.63. Timer Xi mode register in pulse period/pulse width measurement mode

Under development


When measuring measurement pulse time interval from falling edge to falling edge
Count source
Measurement pulse
“H”
“L”
Transfer (indeterminate value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches “0000₁₆”
Count start flag
“1”
“0”
Timer Xi interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software.
Timer Xi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 1.64. Operation timing when measuring a pulse period


Count source
Measurement pulse
“H”
“L”
Transfer (indeterminate value)
Transfer (measured value)
Transfer (measured value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches “0000₁₆”
Count start flag
“1”
“0”
Timer Xi interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software.
Timer Xi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 1.65. Operation timing when measuring a pulse width

Under development

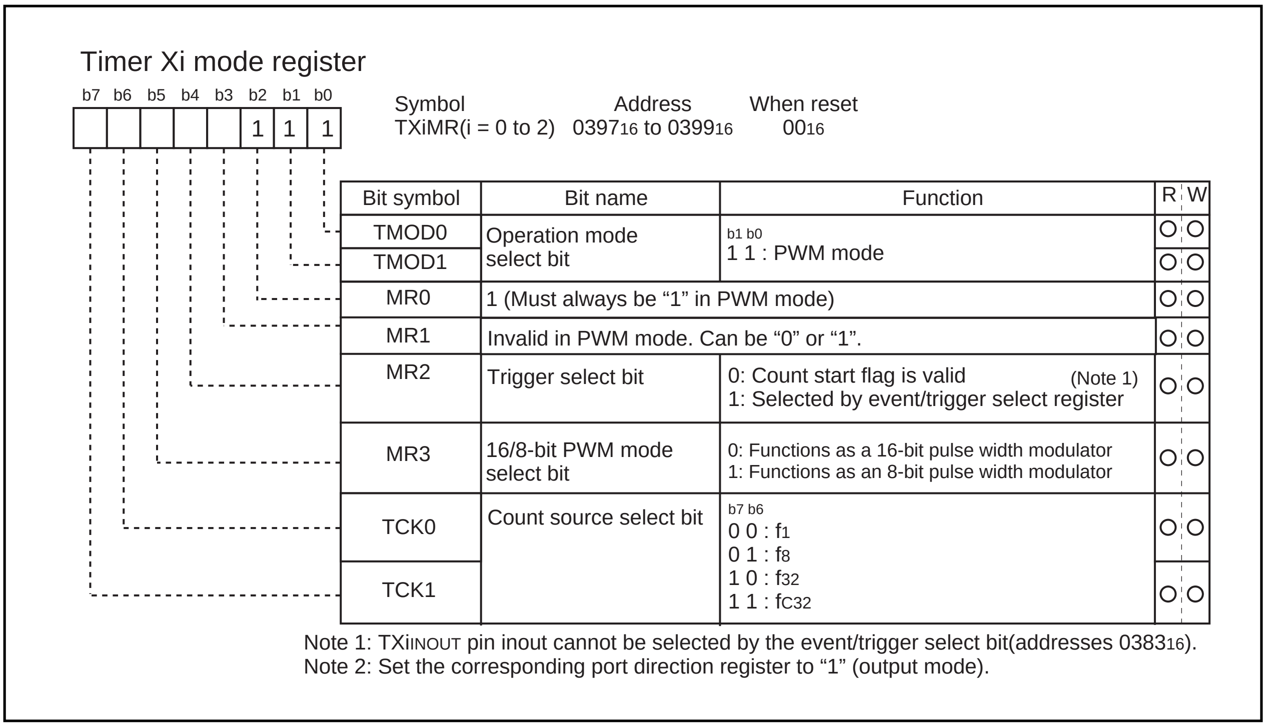
(5) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.25.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.66 shows the timer Xi mode register in pulse width modulation mode. Figure 1.67 shows the example of how a 16-bit pulse width modulator operates. Figure 1.68 shows the example of how an 8-bit pulse width modulator operates.

Table 1.25. Timer specifications in pulse width modulation mode

Item		Specification
Count source		f_1, f_8, f_{32}, f_{C32}
Count operation		• Down counts (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM		• "H" level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM		• "H" level width $n \times (m+1)$/ fi n:values set to timer Xi register’s high-order address • Cycle time $(2^8-1) \times (m+1)$ / fi m : values set to timer Xi register’s low-order address
Count start condition		• The timer overflows • The count start flag is set (= 1)
Count stop condition		• The count start flag is reset (= 0)
Interrupt request generation timing	8 bits PWM	• Set value of "H" level width is except FF_{16}, 00_{16} : PWM pulse goes “L” • Set value of "H" level width is FF_{16}, 00_{16} : Timing that count value goes to 01_{16}
	16 bits PWM	• Set value of "H" level width is except $FFFF_{16}$, 0000_{16} : PWM pulse goes “L” • Set value of "H" level width is $FFFF_{16}$, 0000_{16} : Timing that count value goes to 0001_{16}
TXi_{INOUT} pin function		Pulse output
Read from timer		When timer Xi register is read, it indicates an indeterminate value
Write to timer		• When counting stopped When a value is written to timer Xi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Xi register, it is written to only reload register (Transferred to counter at next reload time)

Note: When set value of "H" level width is 00_{16} or 0000_{16}, pulse outputs "L" level and inversion value, FF_{16} or $FFFF_{16}$ is set to timer.



Timer Xi mode register

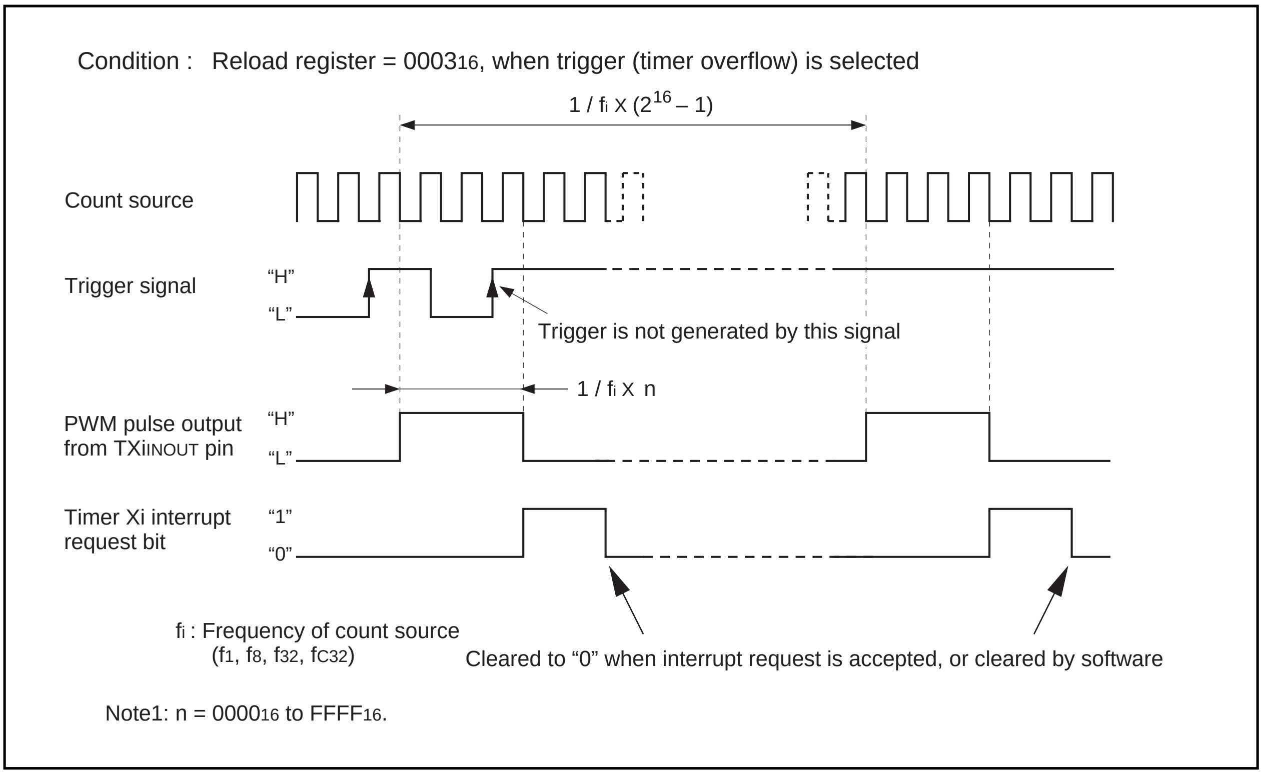
Symbol	Address	When reset
TXiMR(i = 0 to 2)	0397_{16} to 0399_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : PWM mode	○	○
TMOD1			○	○
MR0	1 (Must always be “1” in PWM mode)		○	○
MR1	Invalid in PWM mode. Can be “0” or “1”.		○	○
MR2	Trigger select bit	0: Count start flag is valid (Note 1) 1: Selected by event/trigger select register	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: TXi_{INOUT} pin inout cannot be selected by the event/trigger select bit(addresses 0383_{16}).
Note 2: Set the corresponding port direction register to “1” (output mode).

Figure 1.66. Timer Xi mode register in pulse width modulation mode

Under development



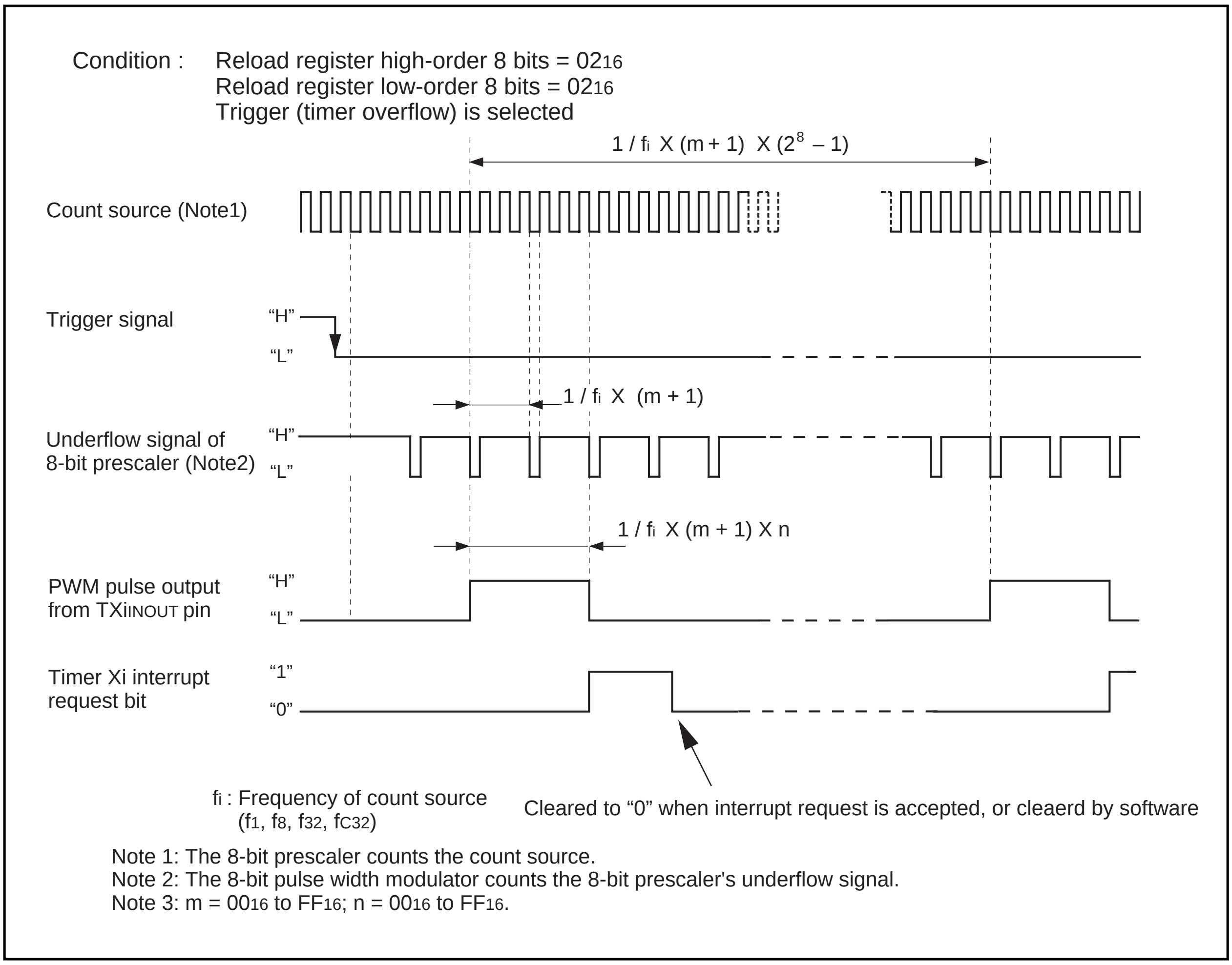
Condition : Reload register = 0003_{16}, when trigger (timer overflow) is selected

1 / fi X (2^16 – 1)
Count source
Trigger signal
"H"
"L"
Trigger is not generated by this signal
1 / fi X n
PWM pulse output from TXiINOUT pin
"H"
"L"
Timer Xi interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

fi : Frequency of count source
(f1, f8, f32, fC32)

Note1: n = 0000_{16} to $FFFF_{16}$.

Figure 1.67. Example of how a 16-bit pulse width modulator operates



Condition : Reload register high-order 8 bits = 02_{16}
Reload register low-order 8 bits = 02_{16}
Trigger (timer overflow) is selected

1 / fi X (m + 1) X (2^8 – 1)
Count source (Note1)
Trigger signal
"H"
"L"
1 / fi X (m + 1)
Underflow signal of 8-bit prescaler (Note2)
"H"
"L"
1 / fi X (m + 1) X n
PWM pulse output from TXiINOUT pin
"H"
"L"
Timer Xi interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleaerd by software

fi : Frequency of count source
(f1, f8, f32, fC32)

Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00_{16} to FF_{16}; n = 00_{16} to FF_{16}.

Figure 1.68. Example of how an 8-bit pulse width modulator operates

Serial I/O

Serial I/O is configured as two channels: UART0 and UART1.

UART0 and UART1 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.69 shows the block diagram of UART0 and UART1. Figure 1.70 shows the block diagram of the transmit/receive unit.

UART0 has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses $03A0_{16}$ and $03A8_{16}$) determine whether UART0 is used as a clock synchronous serial I/O or as a UART.

UART1 is used as a UART only.

Figures 1.71 through 1.73 show the registers related to UARTi.

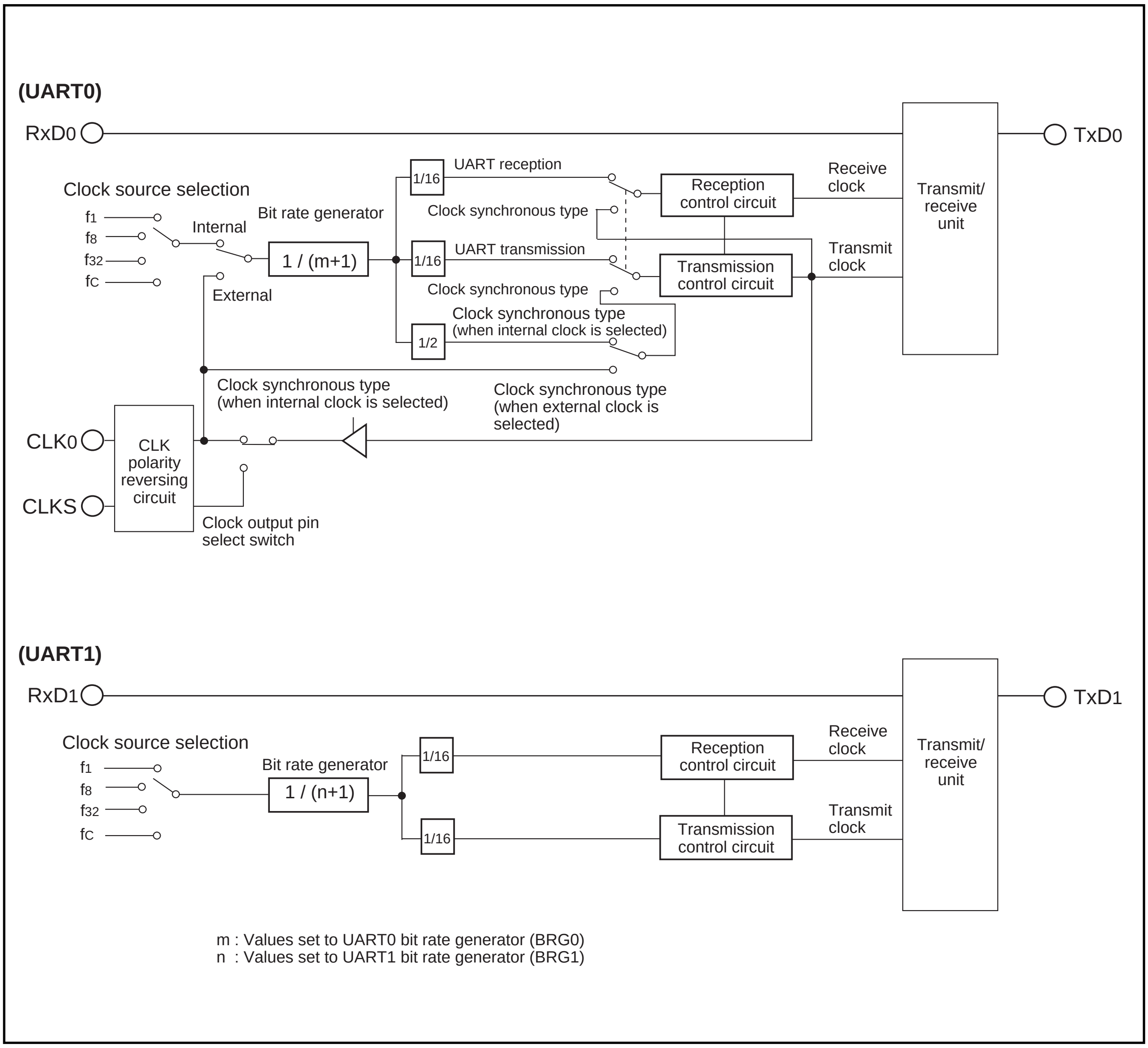
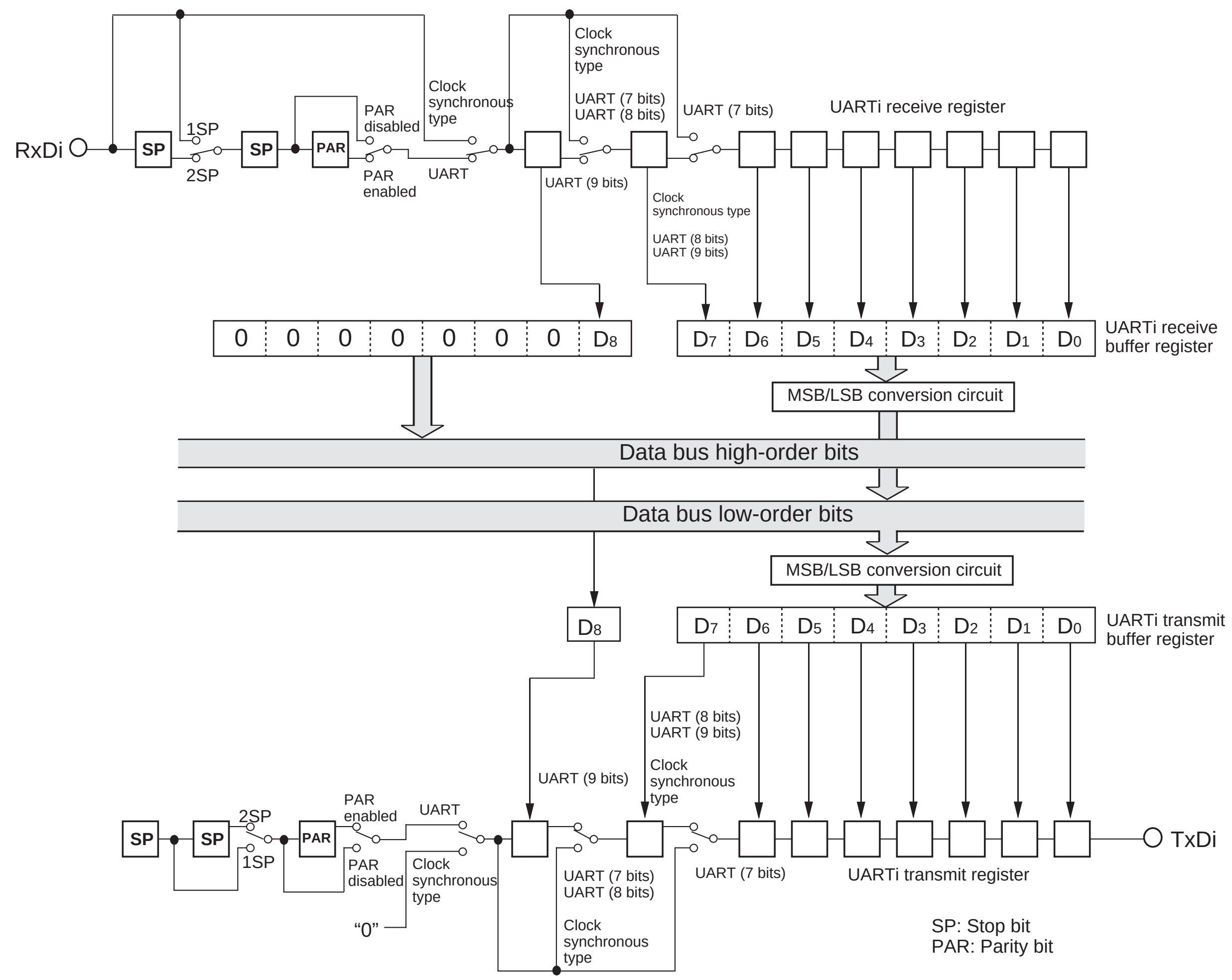


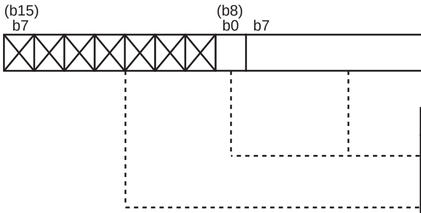
Figure 1.69. Block diagram of UARTi (i = 0, 1)



Note: UART1 cannot be used in clock synchronous serial I/O.

Figure 1.70. Block diagram of transmit/receive unit

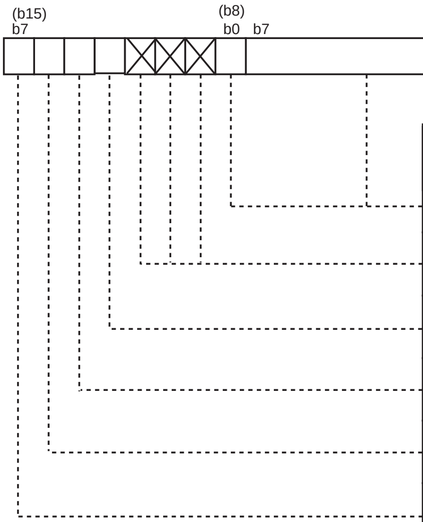
UARTi transmit buffer register



Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate

	R	W
Transmit data	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.	–	–

UARTi receive buffer register

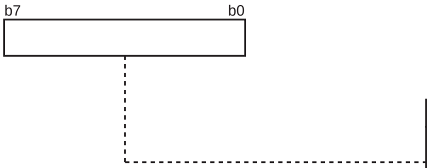


Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	——	Receive data	Receive data	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				–	–
OER	Overrun error flag (Note)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note)	Invalid	0 : No error 1 : Error found	○	×

Note: Bits 15 through 12 are set to “0” when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$ and $03A8_{16}$) are set to “000_2” or the receive enable bit is set to “0”. (Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$, and $03AE_{16}$) is read out.

UARTi bit rate generator



Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate

	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	×	○

Figure 1.71. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
UiMR(i=0,1)	$03A0_{16}$, $03A8_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 1)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit (Note 2)	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: UART1 can use only internal clock. Must set this bit to “1”.

UARTi transmit/receive control register 0

b7	b6	b5	b4	b3	b2	b1	b0
			1		0		

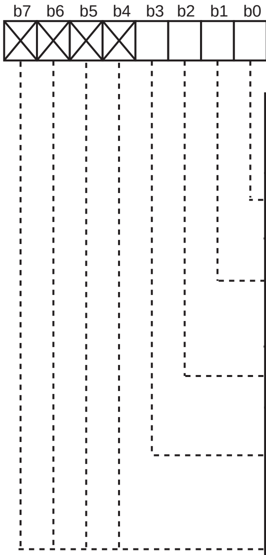
Symbol	Address	When reset
UiC0(i=0,1)	$03A4_{16}$, $03AC_{16}$	08_{16}

Bit symbol	Bit name	Function (Note) (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	○	○
CLK1				○	○
Set this bit to “0”.				○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
Set this bit to “1”.				○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be “0”	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be “0”	○	○

Note: UART1 cannot be used in clock synchronous serial I/O.

Figure 1.72. Serial I/O-related registers (2)

UARTi transmit/receive control register 1

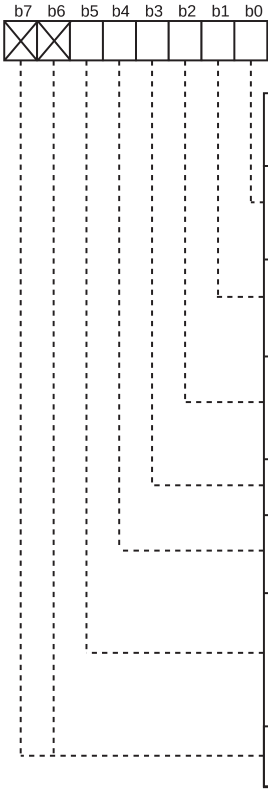


Symbol	Address	When reset
UiC1(i=0,1)	$03A5_{16}$,$03AD_{16}$	02_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				—	—

Note: UART1 cannot be used in clock synchronous serial I/O.

UART transmit/receive control register 2



Symbol	Address	When reset
UCON	$03B0_{16}$	$XX000000_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	Set this bit to “0”.	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
Set this bit to “0”.				○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note 2)	0 : Normal mode (CLK output is CLK0 only) 1 : Transfer clock output from multiple pins function selected	Must always be “0”	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				—	—

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: When using multiple pins to output the transfer clock, the following requirements must be met:
- UART0 internal/external clock select bit (bit 3 at address $03A0_{16}$) = “0”.

Figure 1.73. Serial I/O-related registers (3)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. (See Table 1.26.) Figure 1.65 shows the UART0 transmit/receive mode register.

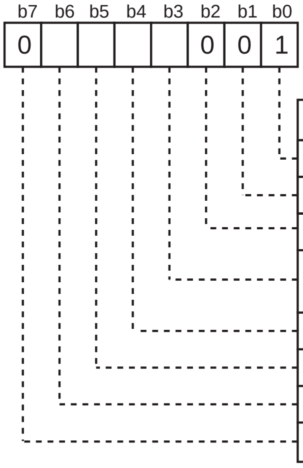
Table 1.26. Specifications of clock synchronous serial I/O mode

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at address $03A0_{16}$ = "0") : fi/ 2(n+1) (Note 1) fi = f_1, f_8, f_{32}, fc • When external clock is selected (bit 3 at address $03A0_{16}$ = "1") : Input from CLK0 pin
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at address $03A5_{16}$) = "1" – Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLK0 polarity select bit (bit 6 at address $03A4_{16}$) = "0": CLK0 input level = "H" – CLK0 polarity select bit (bit 6 at address $03A4_{16}$) = "1": CLK0 input level = "L"
Reception start conditio	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at address $03A5_{16}$) = "1" – Transmit enable bit (bit 0 at address $03A5_{16}$) = "1" – Transmit buffer empty flag (bit 1 at address $03A5_{16}$) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: – CLK0 polarity select bit (bit 6 at address $03A4_{16}$) = "0": CLK0 input level = "H" – CLK0 polarity select bit (bit 6 at address $03A4_{16}$) = "1": CLK0 input level = "L"
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bit 0 at address $03B0_{16}$) = "0": Interrupts requested when data transfer from UART0 transfer buffer register to UART0 transmit register is completed – Transmit interrupt cause select bit (bit 0 at address $03B0_{16}$) = "1": Interrupts requested when data transmission from UART0 transfer register is completed • When receiving – Interrupts requested when data transfer from UART0 receive register to UART0 receive buffer register is completed
Error detection	• Overrun error (Note 2) This error occurs when the next data is ready before contents of UART0 receive buffer register are read out
Select function	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection UART0 transfer clock can be chosen by software to be output from one of the two pins set

Note 1: "n" denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

Note 2: If an overrun error occurs, the UART0 receive buffer will have the next data written in. Note also that the UART0 receive interrupt request bit is not set to "1".

UART0 transmit/receive mode registers



Symbol	Address	When reset
U0MR	03A0₁₆	00₁₆

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Must always be "0" in clock synchronous serial I/O mode)		○	○

Figure 1.74. UART0 transmit/receive mode register in clock synchronous serial I/O mode

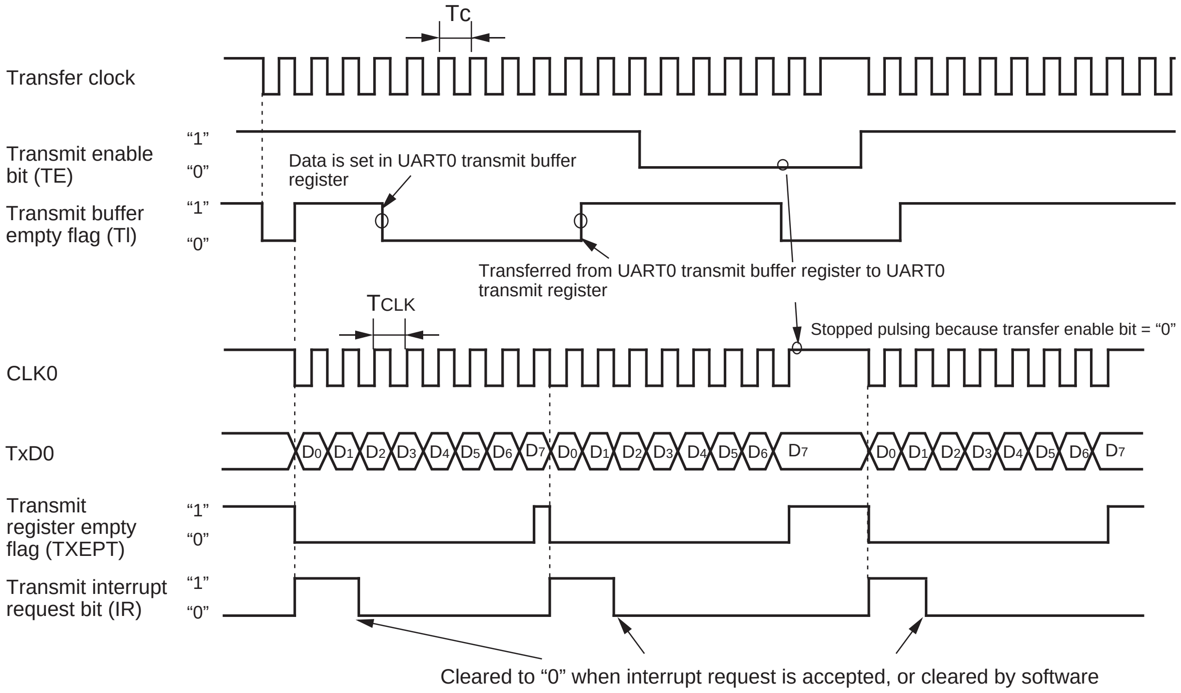
Table 1.27 lists the functions of the input/output pins during clock synchronous serial I/O mode. Note that for a period from when the UART0 operation mode is selected to when transfer starts, the TxD0 pin outputs a "H". (If the N-channel open-drain is selected, this pin is in floating state.)

Table 1.27. Input/output pin functions in clock synchronous serial I/O mode

Pin name	Function	Method of selection
TxD0 (P5$_0$)	Serial data output	Port P5$_0$ direction register (bit 0 at address 03EB$_{16}$)= "1" (Outputs dummy data when performing reception only)
RxD0 (P5$_1$)	Serial data input	Port P5$_1$ direction register (bit 1 at address 03EB$_{16}$)= "0" (Can be used as an input port when performing transmission only)
CLK0 (P5$_2$)	Transfer clock output	Internal/external clock select bit (bit 3 at address 03A0$_{16}$) = "0"
	Transfer clock input	Internal/external clock select bit (bit 3 at address 03A0$_{16}$) = "1" Port P5$_2$ direction register (bit 2 at address 03EB$_{16}$) = "0"

Under
development

• Example of transmit timing (when internal clock is selected)


Tc
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Transmit buffer empty flag (TI)
"1"
"0"
TCLK
CLK0
TxD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5 D6 D7
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"

Data is set in UART0 transmit buffer register

Transferred from UART0 transmit buffer register to UART0 transmit register

Stopped pulsing because transfer enable bit = "0"

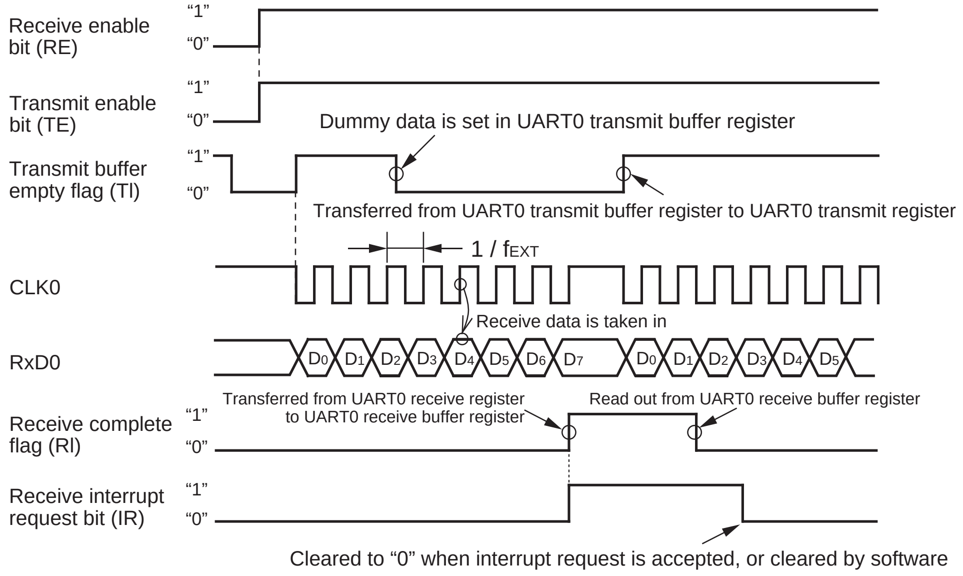
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- Internal clock is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$T_C = T_{CLK} = 2(n + 1) / f_i$
f_i: frequency of BRG0 count source (f_1, f_8, f_{32}, fc)
n: value set to BRG0

• Example of receive timing (when external clock is selected)


Receive enable bit (RE)
"1"
"0"
Transmit enable bit (TE)
"1"
"0"
Transmit buffer empty flag (TI)
"1"
"0"
1 / fEXT
CLK0
RxD0
D0 D1 D2 D3 D4 D5 D6 D7 D0 D1 D2 D3 D4 D5
Receive complete flag (RI)
"1"
"0"
Receive interrupt request bit (IR)
"1"
"0"

Dummy data is set in UART0 transmit buffer register

Transferred from UART0 transmit buffer register to UART0 transmit register

Receive data is taken in

Transferred from UART0 receive register to UART0 receive buffer register

Read out from UART0 receive buffer register

Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- CLK polarity select bit = "0".

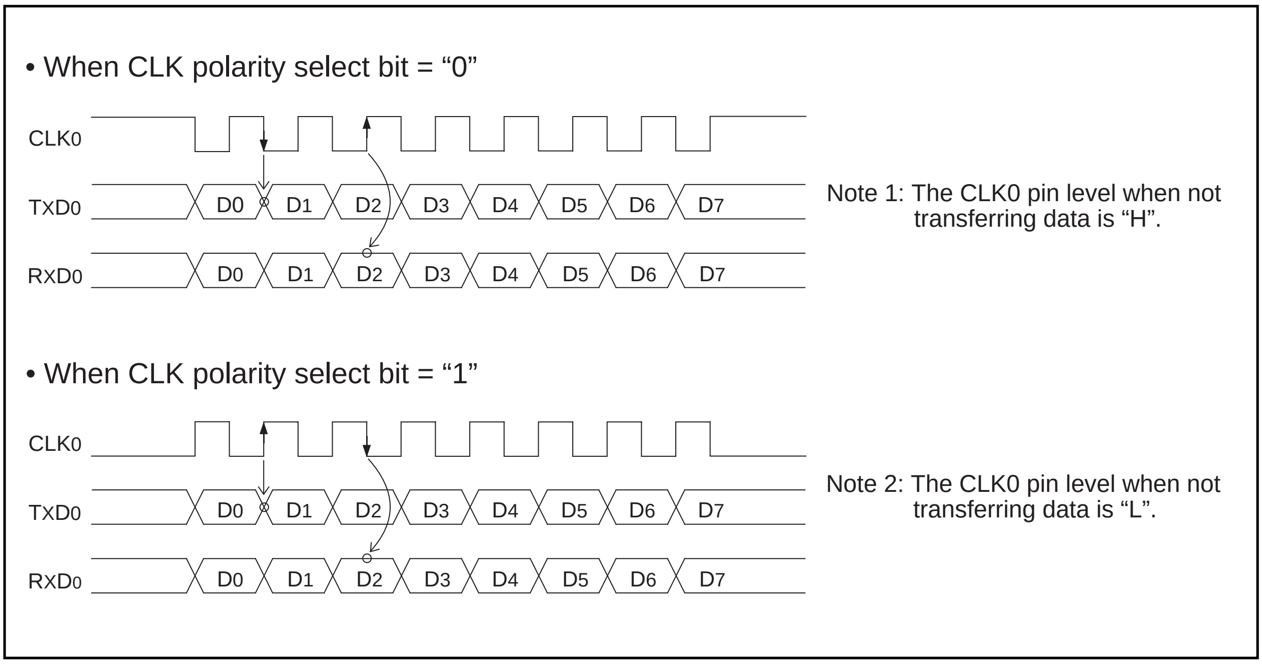
f_{EXT}: frequency of external clock

Meet the following conditions are met when the CLK input before data reception = "H"
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UART0 transmit buffer register

Figure 1.75. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.76, the CLK polarity select bit (bit 6 at addresses $03A4_{16}$) allows selection of the polarity of the transfer clock.


• When CLK polarity select bit = “0”

Note 1: The CLK0 pin level when not transferring data is “H”.

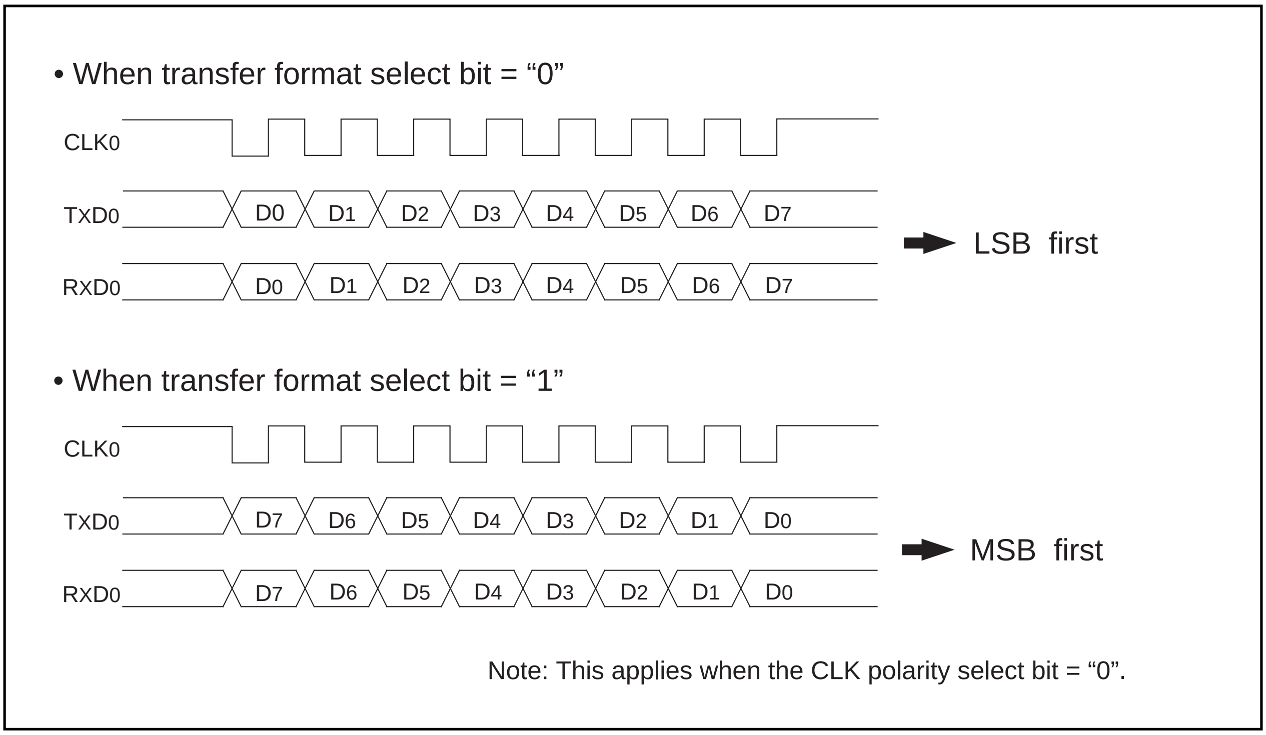
• When CLK polarity select bit = “1”

Note 2: The CLK0 pin level when not transferring data is “L”.

Figure 1.76. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.77, when the transfer format select bit (bit 7 at addresses $03A4_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.


• When transfer format select bit = “0”

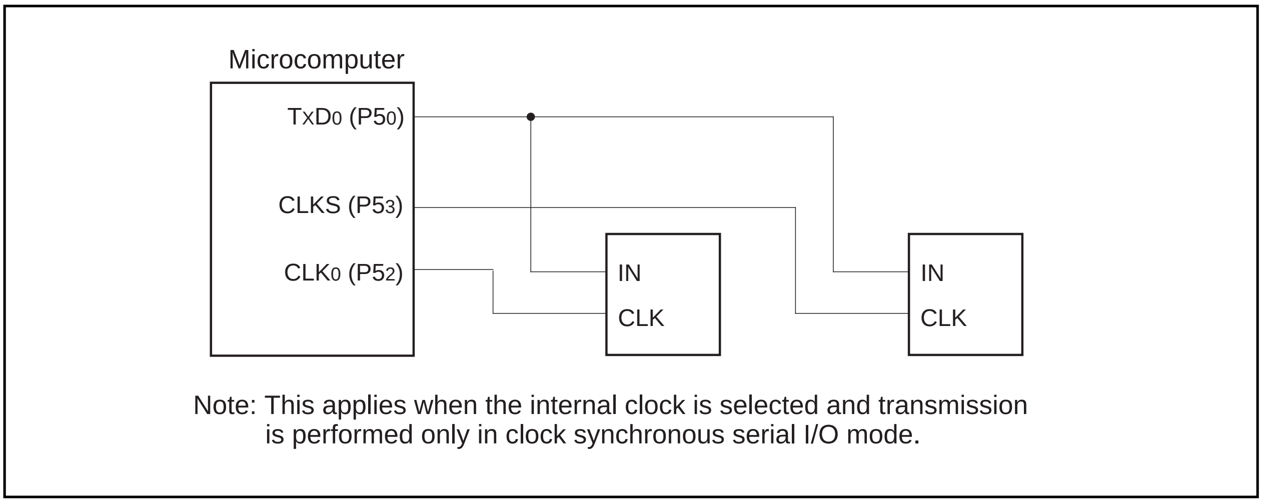
• When transfer format select bit = “1”

Note: This applies when the CLK polarity select bit = “0”.

Figure 1.77. Transfer format

(c) Transfer clock output from multiple pins function

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the CLK and CLKS select bit (bits 4 and 5 at address $03B0_{16}$). (See Figure 1.78.) The multiple pins function is valid only when the internal clock is selected for UART0.



Note: This applies when the internal clock is selected and transmission is performed only in clock synchronous serial I/O mode.

Figure 1.78. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address $03B0_{16}$) is set to “1”, the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

Under development

(2) Clock asynchronous serial I/O (UART) mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. (See Table 1.28.) Figure 1.79 shows the UARTi transmit/receive mode register.

Table 1.28. Specifications of UART Mode

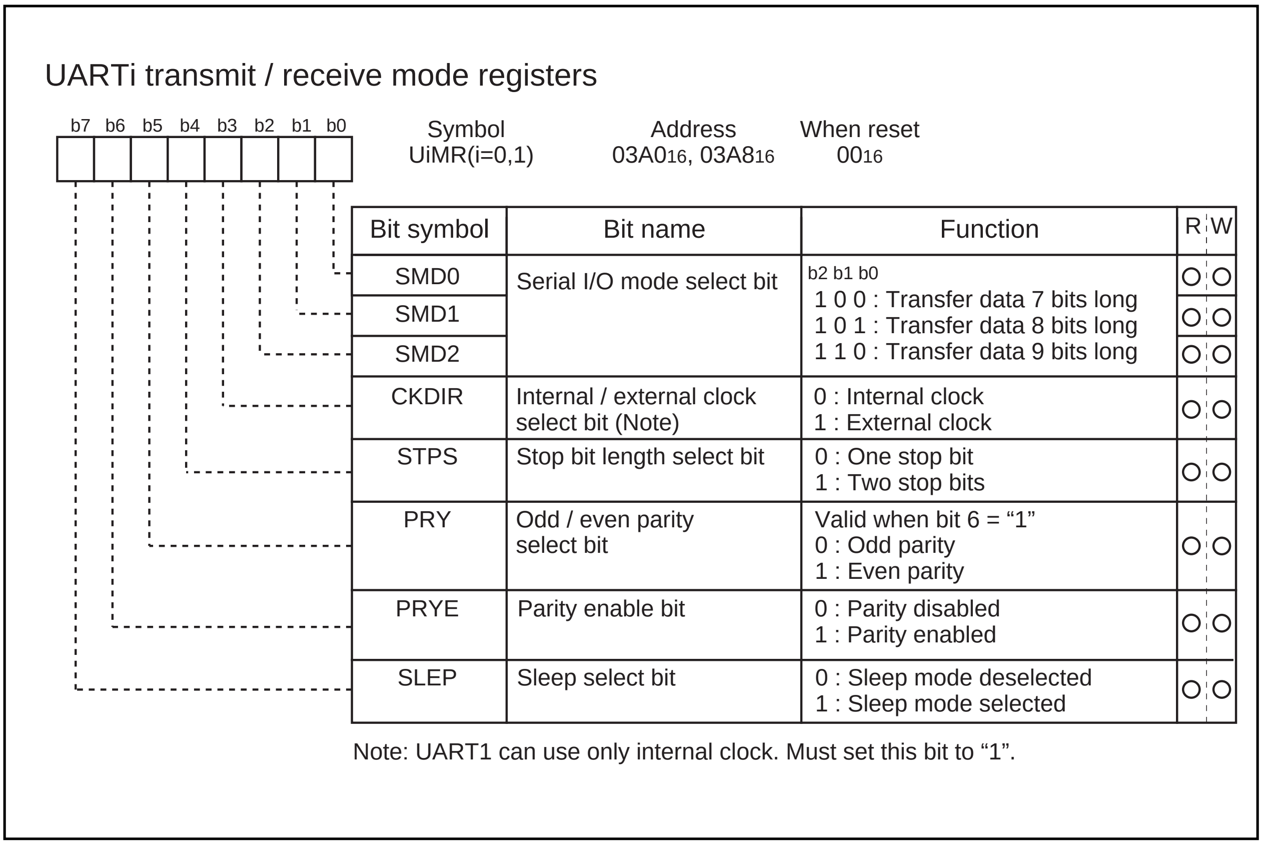
Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses $03A0_{16}$, $03A8_{16}$ = “0”) : fi/16(n+1) (Note 1) fi = f_1, f_8, f_{32}, f_C • When external clock is selected (bit 3 at addresses $03A0_{16}$=“1”) : f_{EXT}/16(n+1) (Note 1) (Note 2)
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses $03A5_{16}$, $03AD_{16}$) = “1” - Transmit buffer empty flag (bit 1 at addresses $03A5_{16}$, $03AD_{16}$) = “0”
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses $03A5_{16}$, $03AD_{16}$) = “1” - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address $03B0_{16}$) = “0”: Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address $03B0_{16}$) = “1”: Interrupts requested when data transmission from UARTi transfer register is completed • When receiving - Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed
Error detection	• Overrun error (Note 3) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1’s in parity and character bits does not match the number of 1’s set • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	• Sleep mode selection This mode is used to transfer data to and from one of multiple slave microcomputers

Note 1: ‘n’ denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

Note 2: f_{EXT} is input from the CLK0 pin. Since UART1 does not have this pin, cannot select external clock.

Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit is not set to “1”.

Under development



UARTi transmit / receive mode registers

Symbol	Address	When reset
UiMR(i=0,1)	$03A0_{16}$, $03A8_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit (Note)	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note: UART1 can use only internal clock. Must set this bit to “1”.

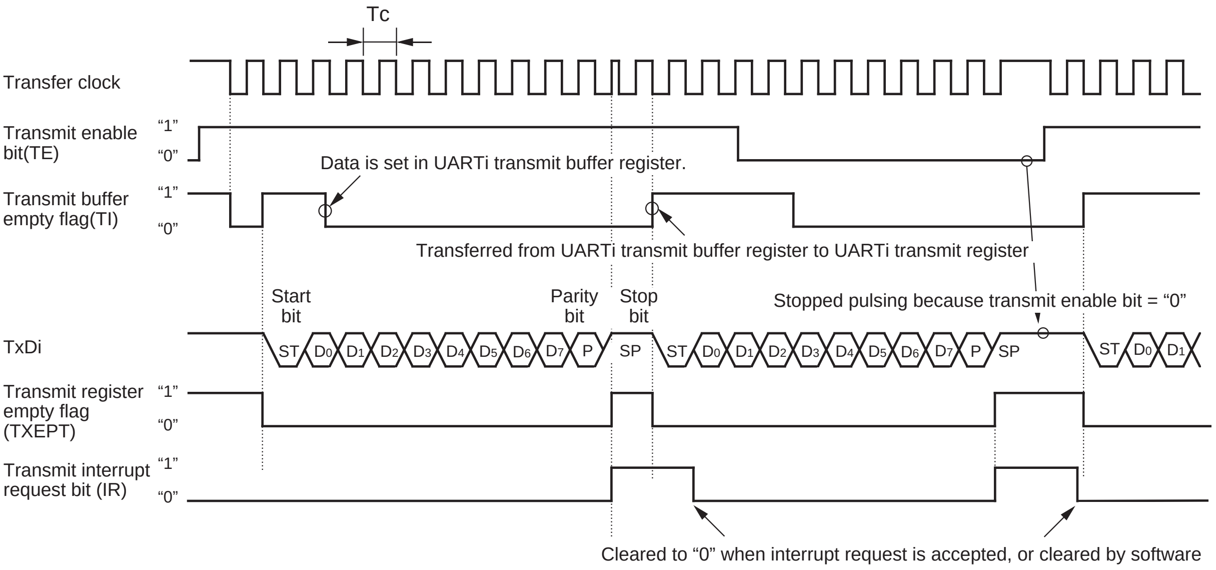
Figure 1.79. UARTi transmit/receive mode register in UART mode

Table 1.29 lists the functions of the input/output pins during UART mode. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a “H”. (If the N-channel open-drain is selected, this pin is in floating state.)

Table 1.29. Input/output pin functions in UART mode

Pin name	Function	Method of selection
TxDi ($P5_0$, $P4_0$)	Serial data output	Port $P5_1$ and $P4_2$ direction register (bit 0 at address $03EB_{16}$, bit 0 at address $03EA_{16}$)= “1” (Can be used as an input port when performing reception only)
RxDi ($P5_1$, $P4_2$)	Serial data input	Port $P5_1$ and $P4_2$ direction register (bit 1 at address $03EB_{16}$, bit 2 at address $03EA_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLK0 ($P5_2$)	Programmable I/O port	Internal/external clock select bit (bit 3 at address $03A0_{16}$) = “0”
	Transfer clock input	Internal/external clock select bit (bit 3 at address $03A0_{16}$) = “1”

- Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)

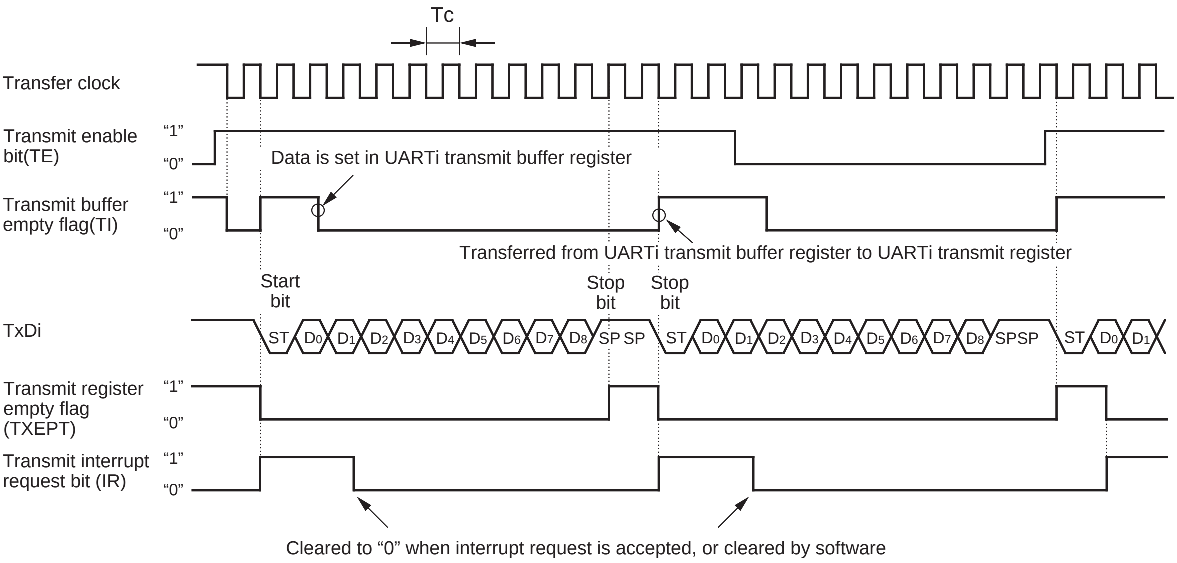

Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register.
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Parity bit
Stop bit
Stopped pulsing because transmit enable bit = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32, fc)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

- Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)


Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag(TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Stop bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

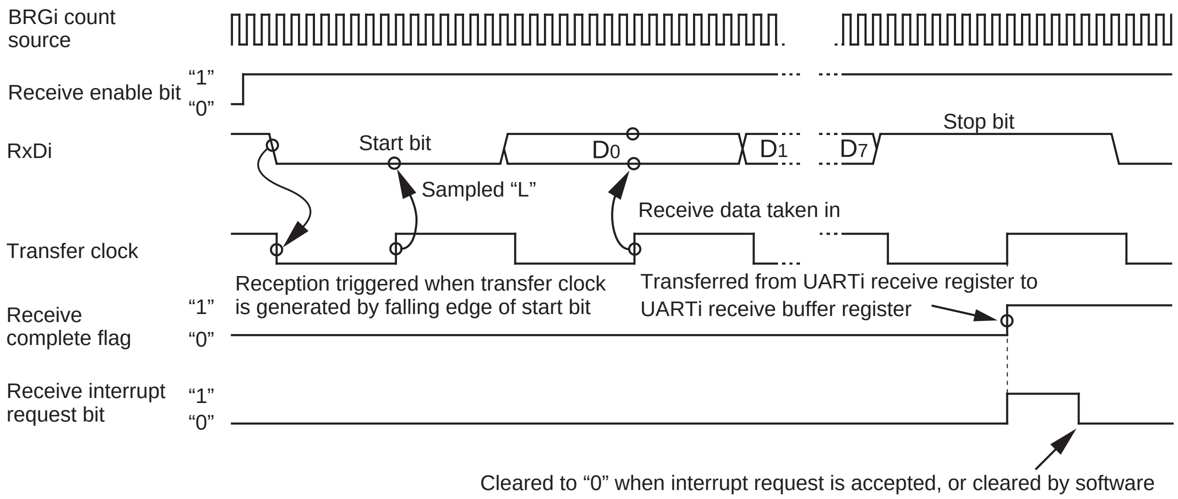
The above timing applies to the following settings :
- Parity is disabled.
- Two stop bits.
- Transmit interrupt cause select bit = "0".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 1.80. Typical transmit timings in UART mode

Under development

- Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)


BRGi count source
Receive enable bit
"1"
"0"
RxDi
Start bit
D0
D1
D7
Stop bit
Sampled "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock is generated by falling edge of start bit
Transferred from UARTi receive register to UARTi receive buffer register
Receive complete flag
"1"
"0"
Receive interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

The above timing applies to the following settings :
- Parity is disabled.
- One stop bit.

Figure 1.81. Typical receive timing in UART mode

(a) Sleep mode

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses 03A0$_{16}$, 03A8$_{16}$) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins $P6_0$ to $P6_7$, and $P5_0$ to $P5_4$ also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address $03D7_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (V_{REF}) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from V_{REF}, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of $03D7_{16}$ to connect V_{REF}.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table 1.30 shows the performance of the A-D converter. Figure 1.82 shows the block diagram of the A-D converter, and Figures 1.83 and 1.84 show the A-D converter-related registers.

Table 1.30. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV_{CC} (V_{CC})
Operating clock ϕ_{AD} (Note 2)	V_{CC} = 5V f_{AD}, divide-by-2 of f_{AD}, divide-by-4 of f_{AD}, f_{AD}=f(X_{IN}) V_{CC} = 3V divide-by-2 of f_{AD}, divide-by-4 of f_{AD}, f_{AD}=f(X_{IN})
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V_{CC} = 5V • Without sample and hold function ±3LSB • With sample and hold function (8-bit resolution) ±2LSB • With sample and hold function (10-bit resolution) ±3LSB V_{CC} = 3V • Without sample and hold function (8-bit resolution) ±2LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN_0 to AN_7) + 5 pins ($AN5_0$ to $AN5_4$)
A-D conversion start condition	• Software trigger A-D conversion starts when the A-D conversion start flag changes to "1"
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 ϕ_{AD} cycles, 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles, 10-bit resolution: 33 ϕ_{AD} cycles

Note 1: Does not depend on use of sample and hold function.

Note 2: Without sample and hold function, set the ϕ_{AD} frequency to 250kHz min.
With the sample and hold function, set the ϕ_{AD} frequency to 1MHz min.

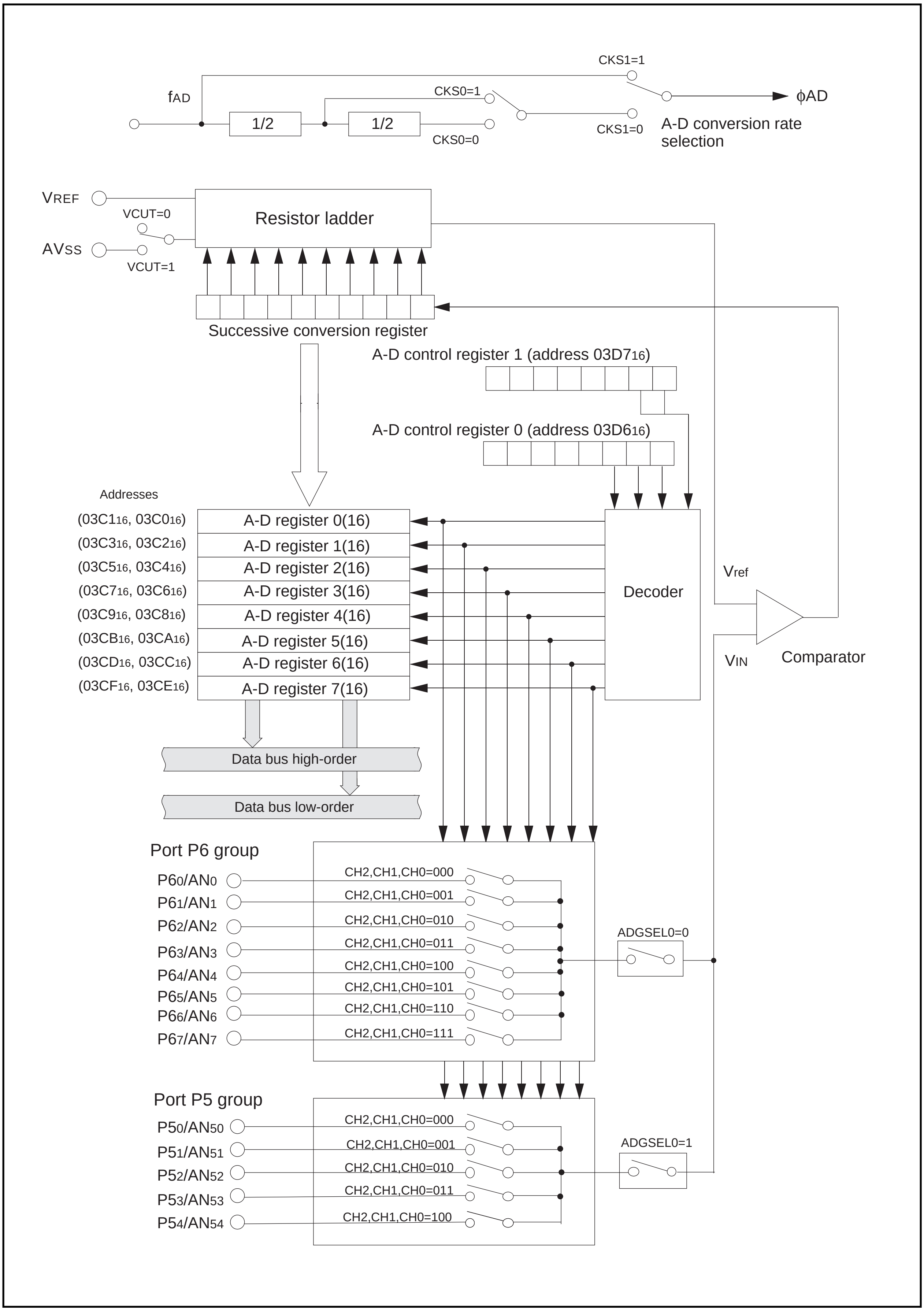


Figure 1.82. Block diagram of A-D converter

Under development

A-D control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
		0					

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0						

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins) When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

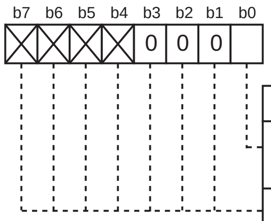
Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.
Note 3: If the repeat sweep mode is selected for the port P5 group, the contents of A-D registers 5 to 7 are indeterminate.

Figure 1.83. A-D converter-related registers (1)

Under development

A-D control register 2 (Note)

Symbol	Address	When reset
ADCON2	$03D4_{16}$	$XXXX0000_2$

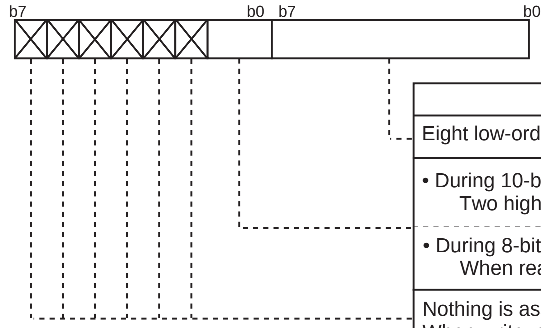


Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their content is indeterminate.			–	–

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

Symbol	Address	When reset
ADi(i=0 to 7)	$03C0_{16}$ to $03CF_{16}$	Indeterminate



Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. When write, set "0". When read, their content is indeterminate.	–	–

Figure 1.84. A-D converter-related registers (2)

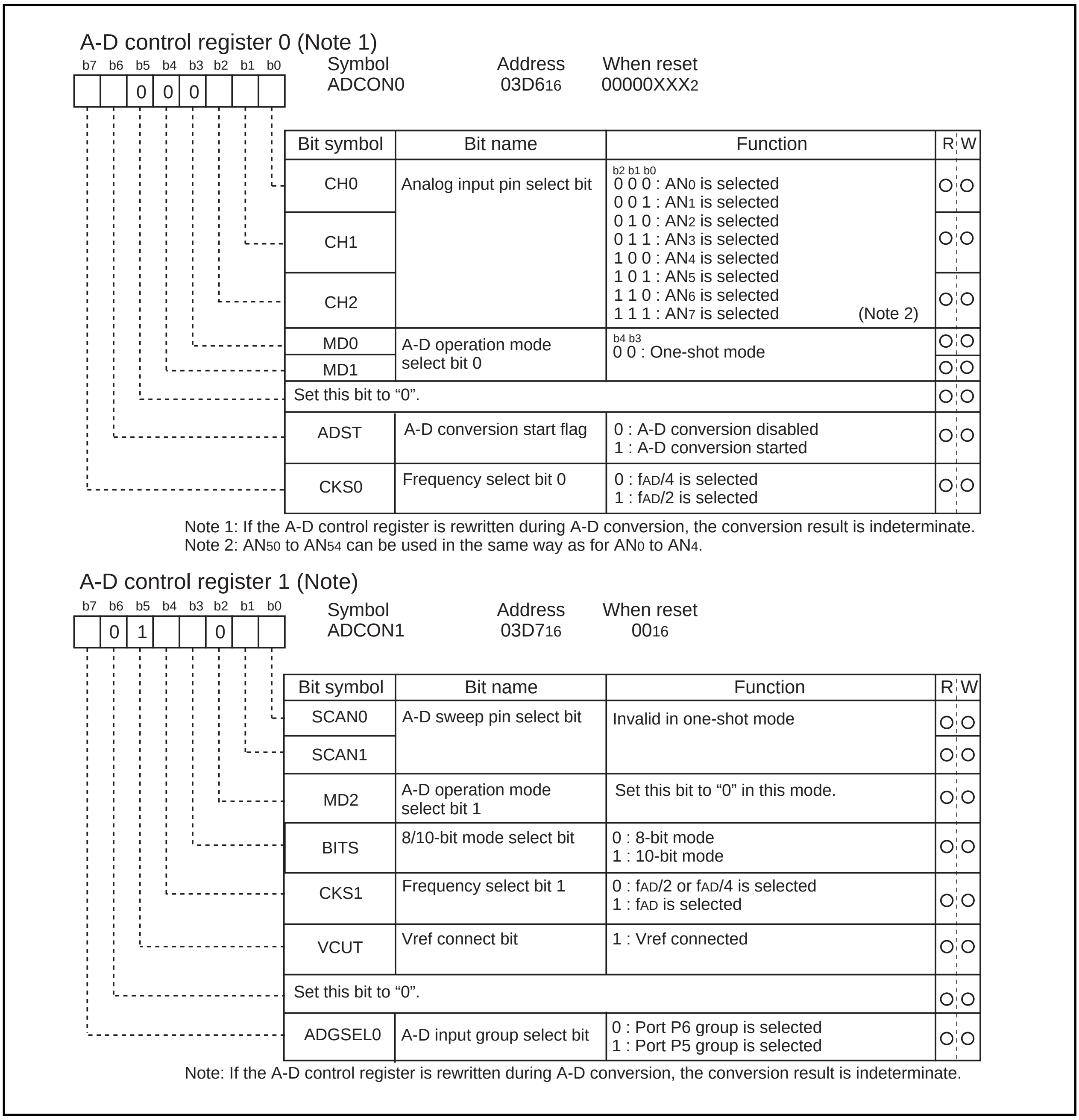
(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. (See Table 1.31.) Figure 1.85 shows the A-D control register in one-shot mode.

Table 1.31. One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to “0”) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN$_0$ to AN$_7$, as selected (Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

Note : AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.



A-D control register 0 (Note 1)

Symbol ADCON0 Address 03D6$_{16}$ When reset 00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.

A-D control register 1 (Note)

Symbol ADCON1 Address 03D7$_{16}$ When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 1.85. A-D conversion register in one-shot mode

Under development

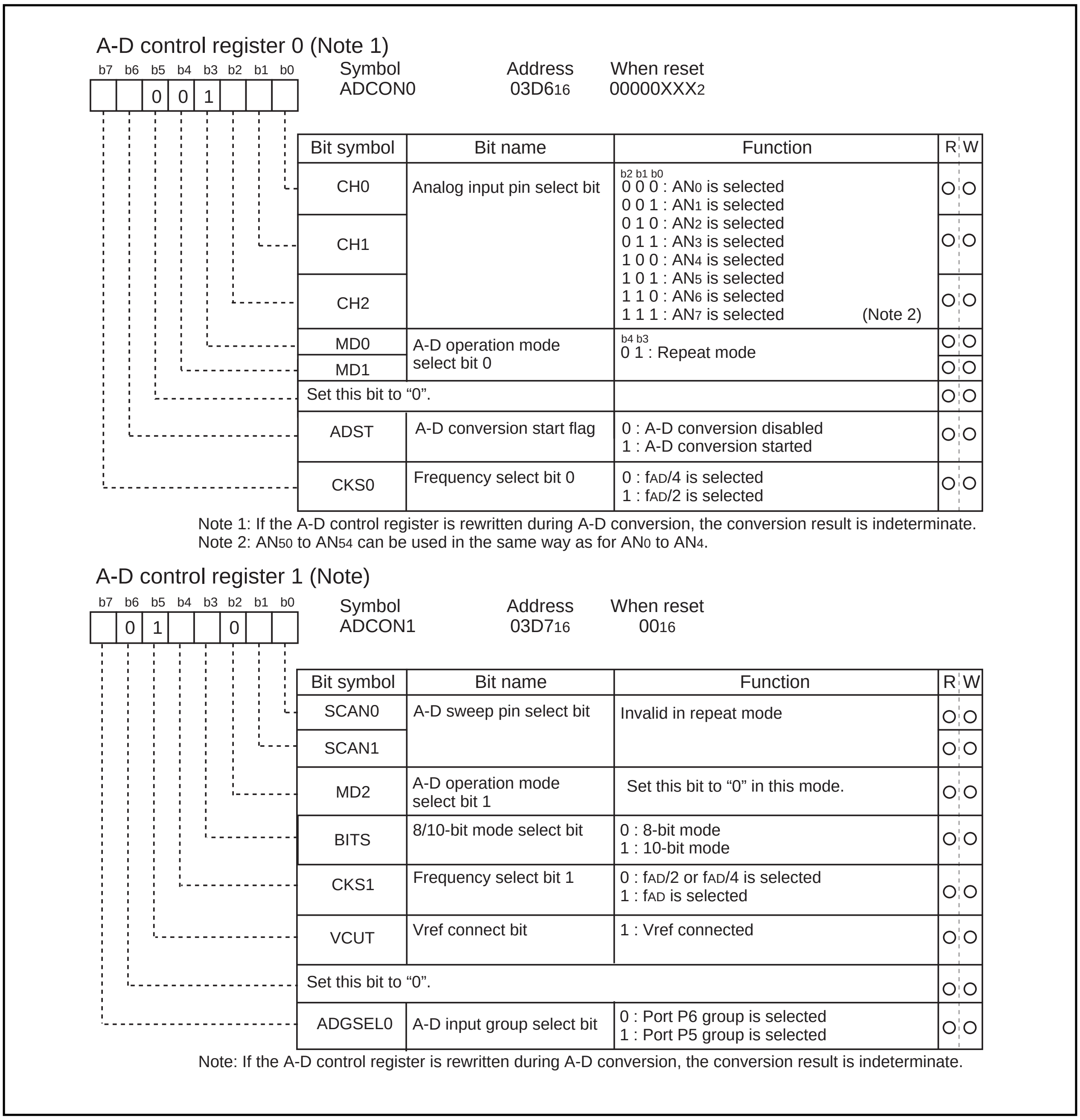
(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. (See Table 1.32.) Figure 1.86 shows the A-D control register in repeat mode.

Table 1.32. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN_0 to AN_7, as selected (Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

Note : AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.



A-D control register 0 (Note 1)

Symbol ADCON0 Address $03D6_{16}$ When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.

A-D control register 1 (Note)

Symbol ADCON1 Address $03D7_{16}$ When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Figure 1.86. A-D conversion register in repeat mode

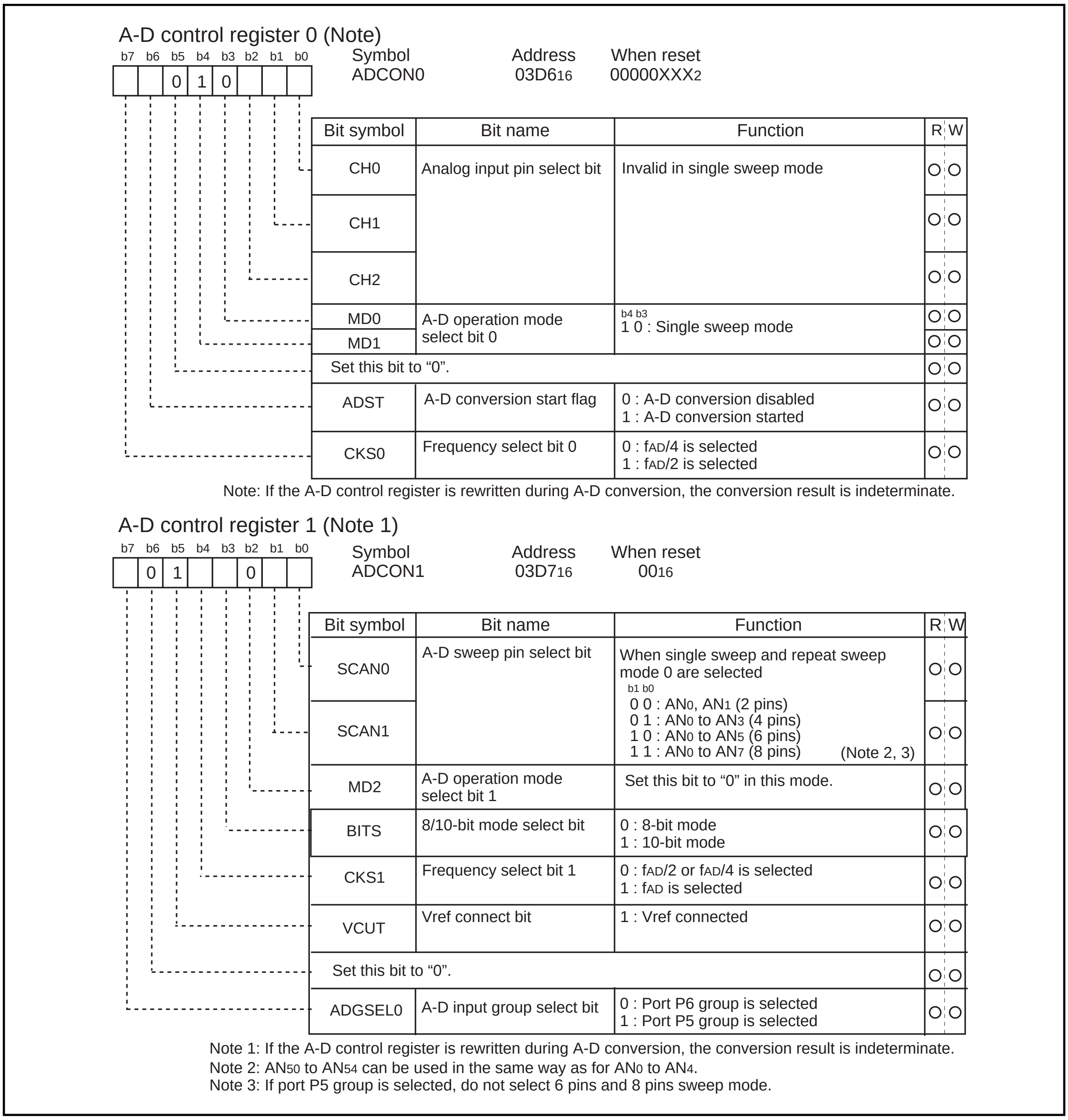
(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. (See Table 1.33.) Figure 1.87 shows the A-D control register in single sweep mode.

Table 1.33. Single sweep mode specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing “1” to A-D converter start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to “0”.) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN_0 and AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)(Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

Note : AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.



A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	0			

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f_{AD}/4 is selected 1 : f_{AD}/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			0		

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : f_{AD}/2 or f_{AD}/4 is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.
Note 3: If port P5 group is selected, do not select 6 pins and 8 pins sweep mode.

Figure 1.87. A-D conversion register in single sweep mode

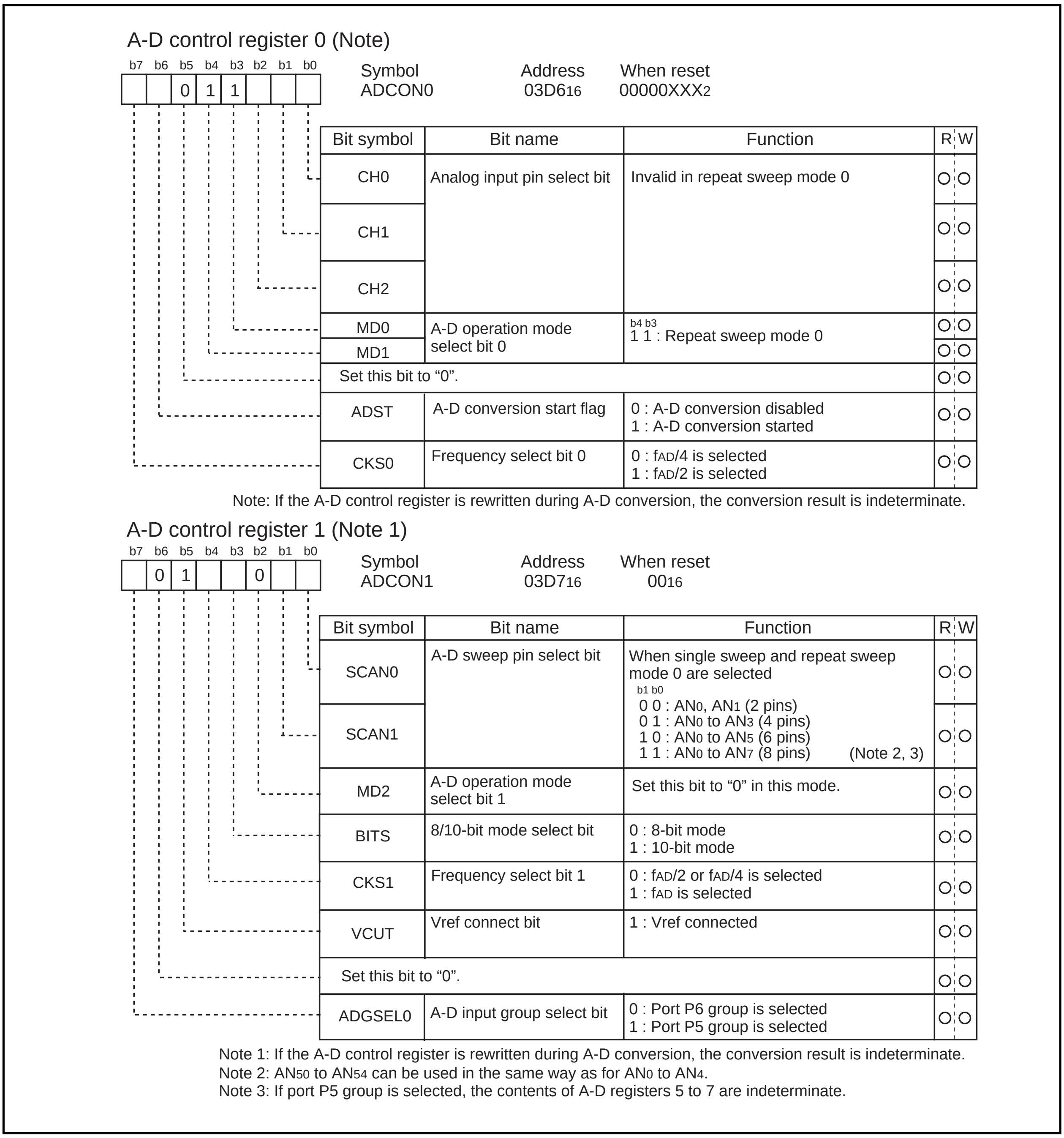
(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. (See Table 1.34.) Figure 1.88 shows the A-D control register in repeat sweep mode 0.

Table 1.34. Repeat sweep mode 0 specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN_0 and AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins)(Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

Note : AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.



A-D control register 0 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
		0	1	1			

Symbol ADCON0 Address $03D6_{16}$ When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0
	0	1			0		

Symbol ADCON1 Address $03D7_{16}$ When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set this bit to “0” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.
Note 3: If port P5 group is selected, the contents of A-D registers 5 to 7 are indeterminate.

Figure 1.88. A-D conversion register in repeat sweep mode 0

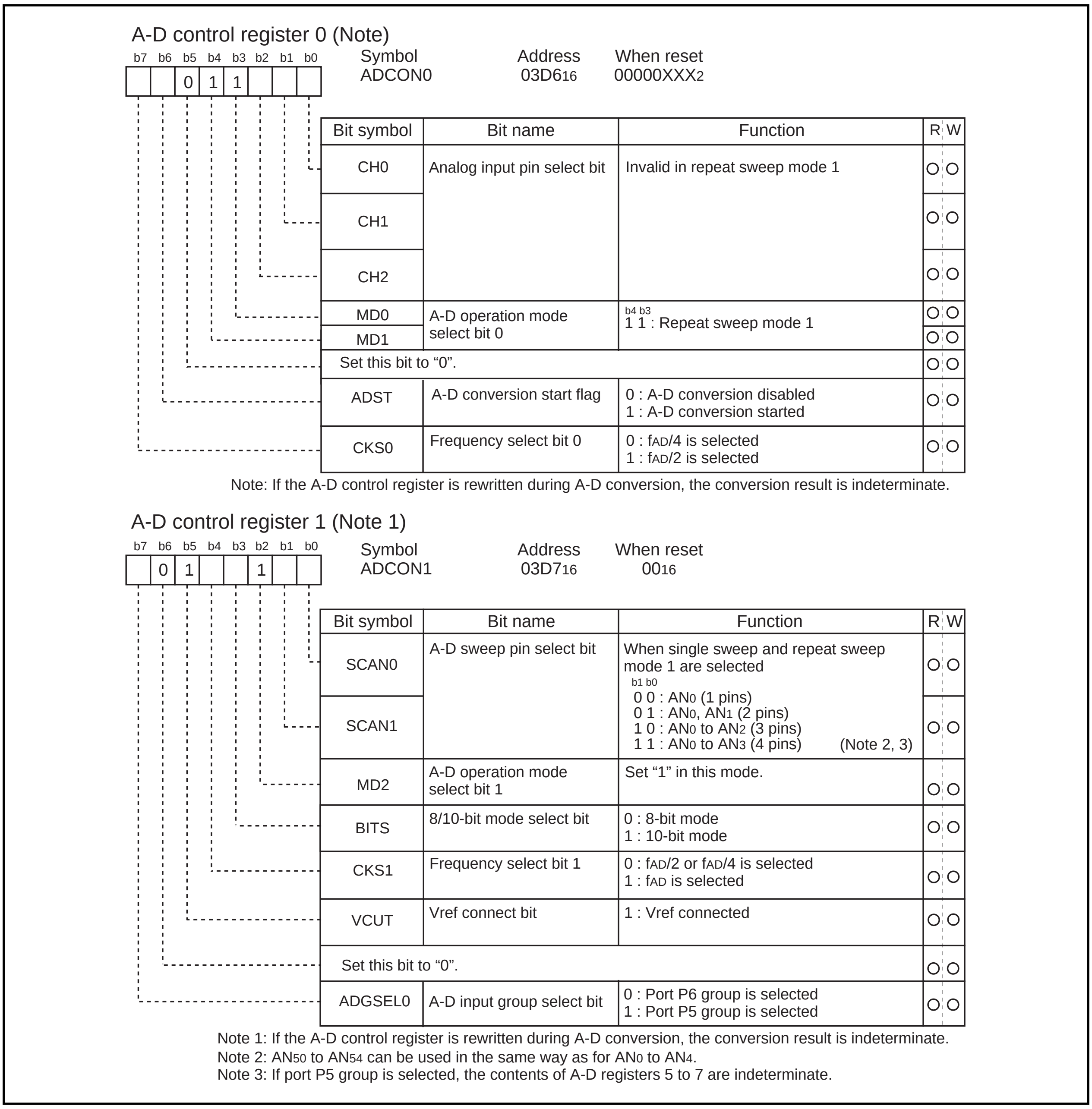
(5) Repeat sweep mode 1

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin select bit. (See Table 1.35.) Figure 1.89 shows the A-D control register in repeat sweep mode 1.

Table 1.35. Repeat sweep mode 1 specifications

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN_0 selected AN_0 → AN_1 → AN_0 → AN_2 → AN_0 → AN_3, etc
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN_0 (1 pin), AN_0 and AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins) (Note)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)

Note : AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.



A-D control register 0 (Note)

Symbol	Address	When reset
ADCON0	$03D6_{16}$	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

Symbol	Address	When reset
ADCON1	$03D7_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 1 are selected b1 b0 0 0 : AN_0 (1 pins) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins) (Note 2, 3)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	Set “1” in this mode.	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.
Note 3: If port P5 group is selected, the contents of A-D registers 5 to 7 are indeterminate.

Figure 1.89. A-D conversion register in repeat sweep mode 1

• Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address $03D4_{16}$) to “1”. When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 ϕ_{AD} cycle is achieved with 8-bit resolution and 33 ϕ_{AD} with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

Programmable I/O Ports

There are 43 programmable I/O ports: P0 to P7. Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set. The port P1 allows the drive capacity of its N-channel output transistor to be set as necessary.

Figures 1.90 to 1.92 show the programmable I/O ports.

Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.

To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices, they function as outputs regardless of the contents of the direction registers. See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

Figure 1.93 shows the direction registers.

These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.

(2) Port registers

Figure 1.94 shows the port registers.

These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.

(3) Pull-up control registers

Figure 1.95 shows the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

(4) Port P1 drive capacity control register

Figure 1.95 shows a structure of the port P1 drive capacity control register.

This register is used to control the drive capacity of the port P1's N-channel output transistor. Each bit in this register corresponds one for one to the port pins.

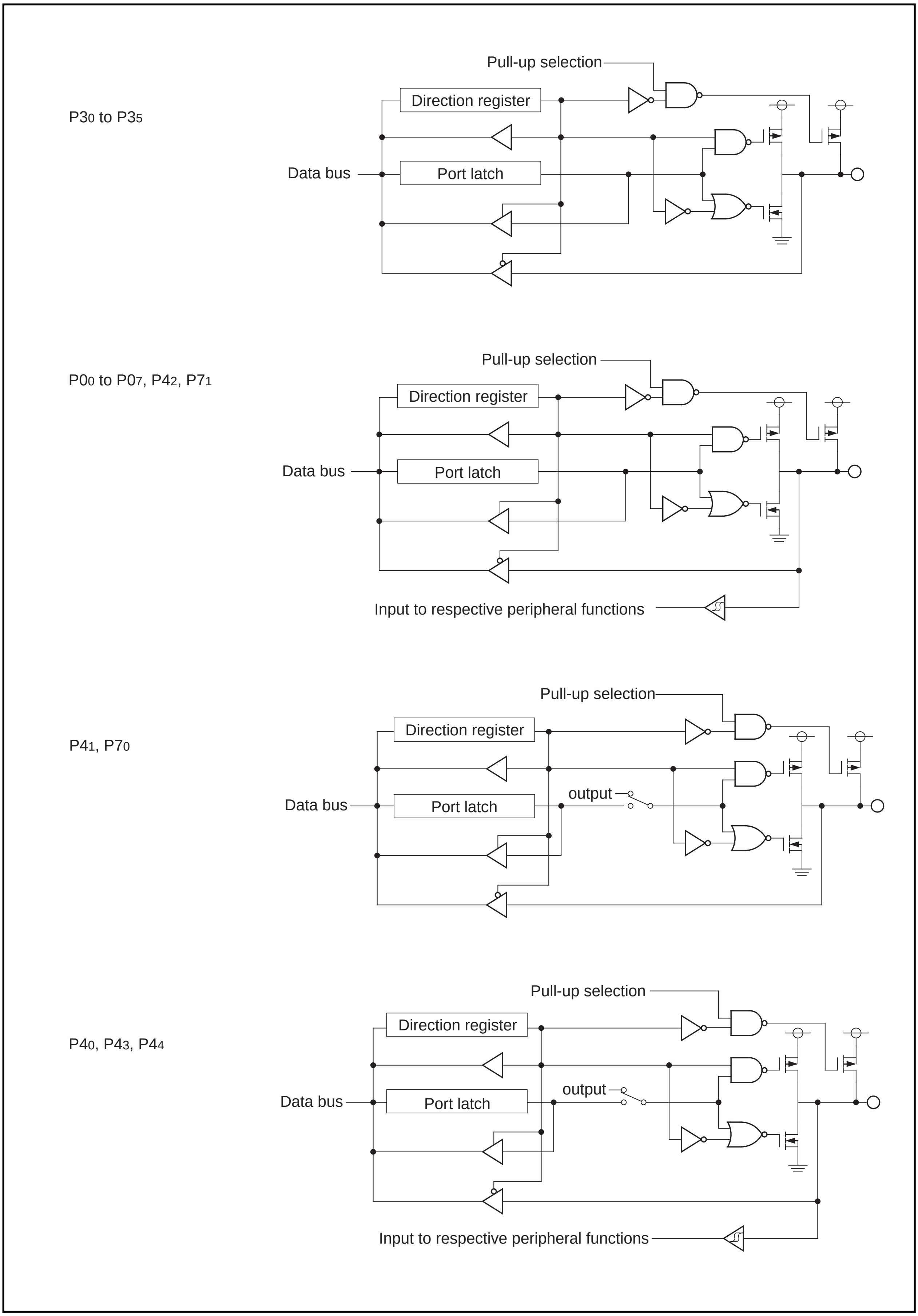


Figure 1.90. Programmable I/O ports (1)

Under development

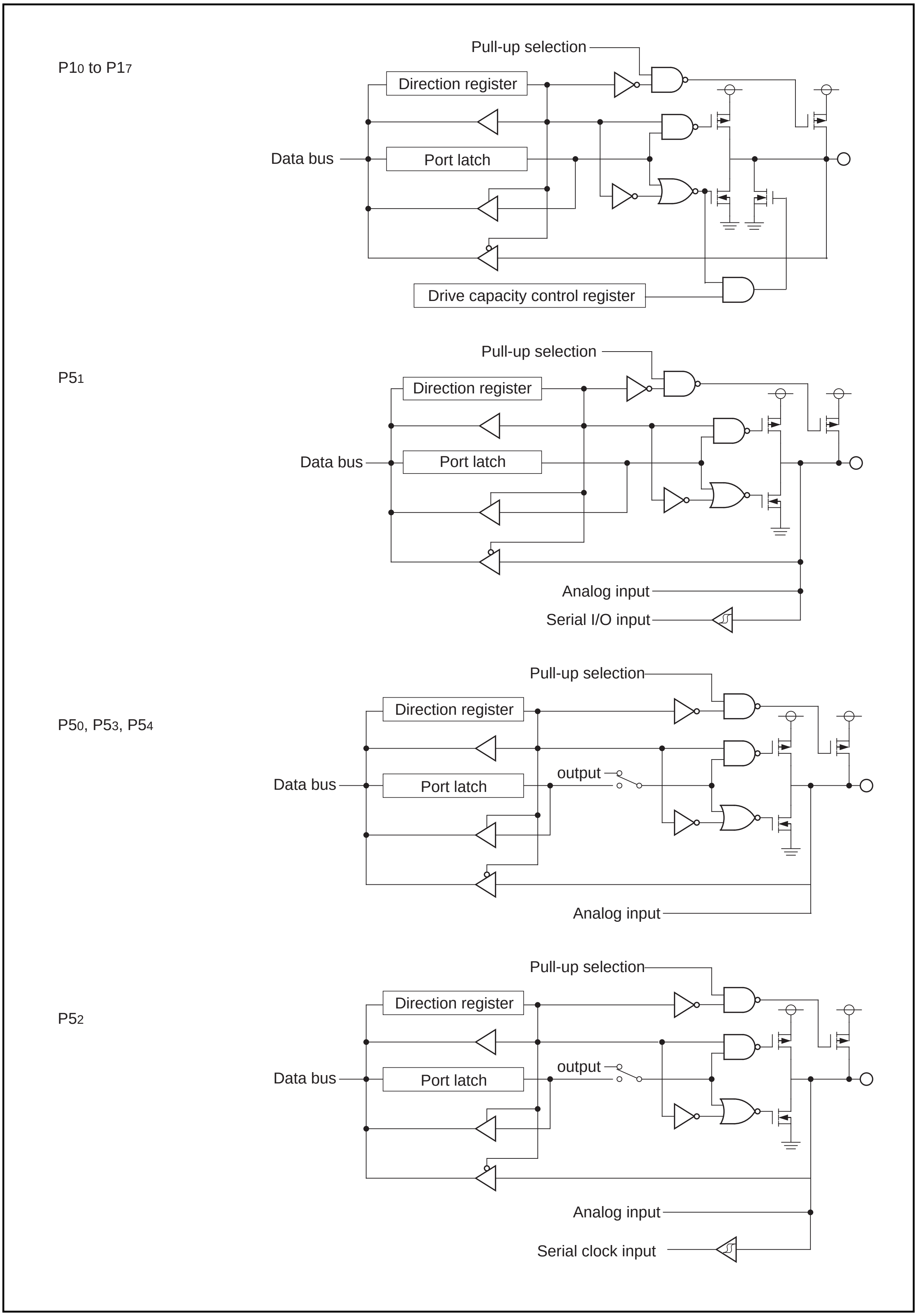


Figure 1.91. Programmable I/O ports (2)

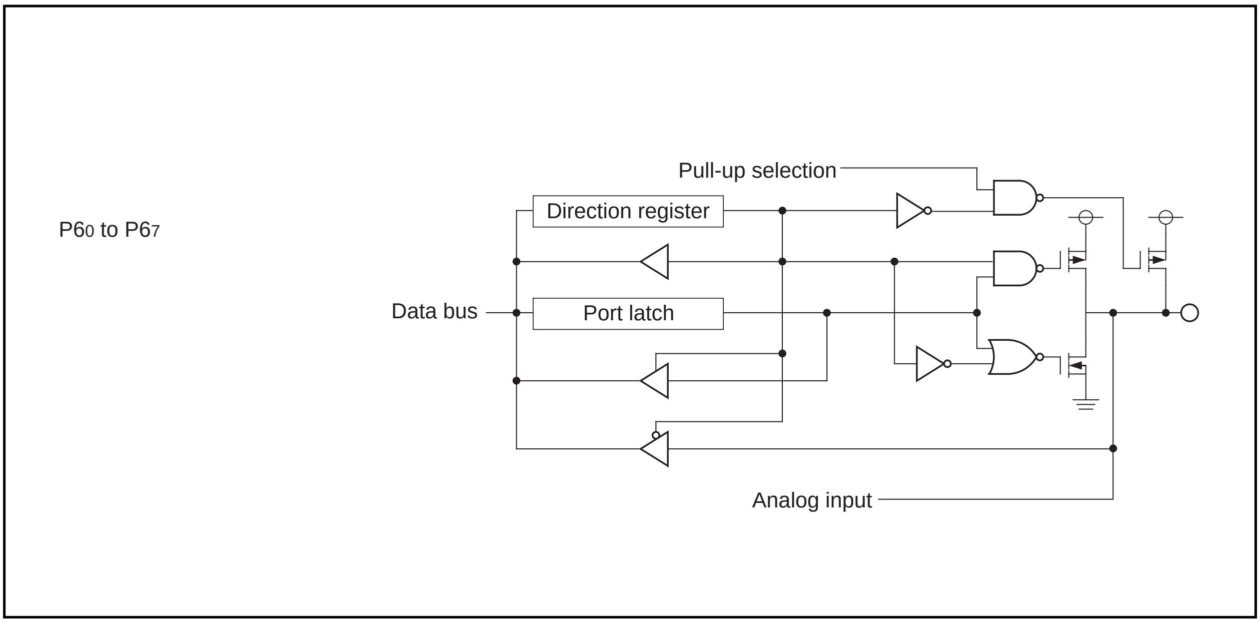
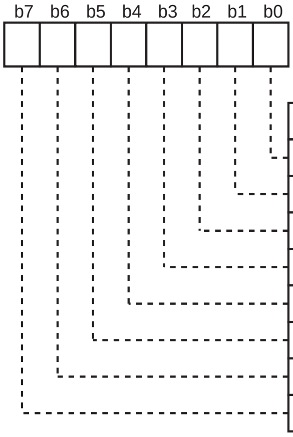


Figure 1.92. Programmable I/O ports (3)

Under development

Port Pi direction register (Note 1)



Symbol	Address	When reset
PDi (i = 0 to 7)	03E2$_{16}$, 03E3$_{16}$, 03E7$_{16}$, 03EA$_{16}$, 03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$	00$_{16}$ 00$_{16}$

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7 except 2)	○	○
PDi_1	Port Pi$_1$ direction register		○	○
PDi_2	Port Pi$_2$ direction register		○	○
PDi_3	Port Pi$_3$ direction register		○	○
PDi_4	Port Pi$_4$ direction register		○	○
PDi_5	Port Pi$_5$ direction register		○	○
PDi_6	Port Pi$_6$ direction register		○	○
PDi_7	Port Pi$_7$ direction register		○	○

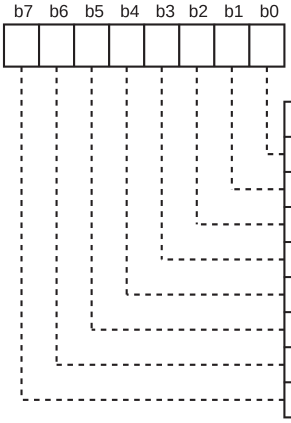
Note 1: Set bit 2 of protect register (address 000A$_{16}$) to “1” before rewriting to the port P4 direction register.

Note 2: Nothing is assigned in direction register of P3$_6$, P3$_7$, P4$_6$, P4$_7$, P5$_5$ to p5$_7$, P7$_2$ to P7$_7$. These bits can either be set nor reset. When read, its contents are indeterminate.

Figure 1.93. Direction register

Port Pi register

Symbol	Address	When reset
Pi (i = 0 to 7)	$03E0_{16}$, $03E1_{16}$, $03E5_{16}$, $03E8_{16}$, $03E9_{16}$, $03EC_{16}$, $03ED_{16}$	Indeterminate Indeterminate



Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : “L” level data 1 : “H” level data (i = 0 to 7 except 2)	○	○
Pi_1	Port Pi$_1$ register		○	○
Pi_2	Port Pi$_2$ register		○	○
Pi_3	Port Pi$_3$ register		○	○
Pi_4	Port Pi$_4$ register		○	○
Pi_5	Port Pi$_5$ register		○	○
Pi_6	Port Pi$_6$ register		○	○
Pi_7	Port Pi$_7$ register		○	○

Note: Nothing is assigned in direction register of P3$_6$, P3$_7$, P4$_6$, P4$_7$, P5$_5$ to p5$_7$, P7$_2$ to P7$_7$. This bit can either be set nor reset. When read, its content is indeterminate.

Figure 1.94. Port register

Pull-up control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PUR0	03FC$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PU00	P0$_0$ to P0$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	P0$_4$ to P0$_7$ pull-up		○	○
PU02	P1$_0$ to P1$_3$ pull-up		○	○
PU03	P1$_4$ to P1$_7$ pull-up		○	○
—	—		○	○
—	—		○	○
PU06	P3$_0$ to P3$_3$ pull-up		○	○
PU07	P3$_4$ to P3$_5$ pull-up		○	○

Pull-up control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PUR1	03FD$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PU10	P4$_0$ to P4$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU11	P4$_4$ to P4$_7$ pull-up		○	○
PU12	P5$_0$ to P5$_3$ pull-up		○	○
PU13	P5$_4$ pull-up		○	○
PU14	P6$_0$ to P6$_3$ pull-up		○	○
PU15	P6$_4$ to P6$_7$ pull-up		○	○
PU16	P7$_0$ to P7$_1$ pull-up		○	○
—	—		○	○

Port P1 drive capacity control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
DRR	03FE$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
DRR0	Port P1$_0$ drive capacuty	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	○	○
DRR1	Port P1$_1$ drive capacuty		○	○
DRR2	Port P1$_2$ drive capacuty		○	○
DRR3	Port P1$_3$ drive capacuty		○	○
DRR4	Port P1$_4$ drive capacuty		○	○
DRR5	Port P1$_5$ drive capacuty		○	○
DRR6	Port P1$_6$ drive capacuty		○	○
DRR7	Port P1$_7$ drive capacuty		○	○

Figure 1.95. Pull-up control register

Under development

Example connection of unused pins

Table 1.36. Example connection of unused pins

Pin name	Connection
Ports P0, P1, P3 to P7	After setting for input mode, connect every pin to V_{SS} (pull-down); or after setting for output mode, leave these pins open.
X_{OUT} (Note)	Open
AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}	Connect to V_{SS}

Note: With external clock input to X_{IN} pin.

Usage Precaution

Timer A (timer mode)

(1) Reading the timer A0 register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer A0 register with the reload timing gets “$FFFF_{16}$”. Reading the timer A0 register after setting a value in the timer A0 register with a count halted but before the counter starts counting gets a proper value.

Timer A (event counter mode)

(1) Reading the timer A0 register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer A0 register with the reload timing gets “$FFFF_{16}$” by underflow or “0000_{16}” by overflow. Reading the timer A0 register after setting a value in the timer A0 register with a count halted but before the counter starts counting gets a proper value.

(2) When stop counting in free run type, set timer again.

Timer A (one-shot timer mode)

(1) Setting the count start flag to “0” while a count is in progress causes as follows:
- The counter stops counting and a content of reload register is reloaded.
- The $TA0_{OUT}$ pin outputs “L” level.
- The interrupt request generated and the timer A0 interrupt request bit goes to “1”.

(2) The timer A0 interrupt request bit goes to “1” if the timer's operation mode is set using any of the following procedures:
- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer A0 interrupt (interrupt request bit), set timer A0 interrupt request bit to “0” after the above listed changes have been made.

Timer A (pulse width modulation mode)

(1) The timer A0 interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:
- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer A0 interrupt (interrupt request bit), set timer A0 interrupt request bit to “0” after the above listed changes have been made.

(2) Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the $TA0_{OUT}$ pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer A0 interrupt request bit goes to “1”. If the $TA0_{OUT}$ pin is outputting an “L” level in this instance, the level does not change, and the timer A0 interrupt request bit does not becomes “1”.

Timer B (timer mode, event counter mode)

(1) Reading the timer Bi register while a count is in progress allows reading , with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing gets “$FFFF_{16}$”. Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

Timer B (pulse period/pulse width measurement mode)

(1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to “1”.

(2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

Timer X (timer mode)

(1) Reading the timer Xi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Xi register with the reload timing gets “$FFFF_{16}$”. Reading the timer A0 register after setting a value in the timer Xi register with a count halted but before the counter starts counting gets a proper value.

Timer X (event counter mode)

(1) Reading the timer Xi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Xi register with the reload timing gets “$FFFF_{16}$” by underflow or “0000_{16}” by overflow. Reading the timer Xi register after setting a value in the timer Xi register with a count halted but before the counter starts counting gets a proper value.

(2) When stop counting in free run type, set timer again.

Timer X (one-shot timer mode)

(1) Setting the count start flag to “0” while a count is in progress causes as follows:

- The counter stops counting and a content of reload register is reloaded.
- The TXiINOUT pin outputs “L” level.
- The interrupt request generated and the timer Xi interrupt request bit goes to “1”.

(2) The timer Xi interrupt request bit goes to “1” if the timer's operation mode is set using any of the following procedures:

- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Xi interrupt (interrupt request bit), set timer Xi interrupt request bit to “0” after the above listed changes have been made.

Timer X (pulse width modulation mode)

(1) The timer Xi interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:

- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Xi interrupt (interrupt request bit), set timer Xi interrupt request bit to “0” after the above listed changes have been made.

(2) Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TXi_{INOUT} pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer Xi interrupt request bit goes to “1”. If the TXi_{INOUT} pin is outputting an “L” level in this instance, the level does not change, and the timer Xi interrupt request bit does not becomes “1”.

Timer X (pulse period/pulse width measurement mode)

(1) If changing the measurement mode select bit is set after a count is started, the timer Xi interrupt request bit goes to “1”.

(2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Xi interrupt request is not generated.

A-D Converter

(1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).
In particular, when the Vref connection bit is changed from “0” to “1”, start A-D conversion after an elapse of 1 µs or longer.

(2) When changing A-D operation mode, select analog input pin again.

(3) Using one-shot mode or single sweep mode
Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)

(4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1
Use the undivided main clock as the internal CPU clock.

Stop Mode and Wait Mode

(1) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to “L” level until main clock oscillation is stabilized.

(2) When shifting to WAIT mode or STOP mode, the program stops after reading 8 bytes from the WAIT instruction and the instruction that sets all clock stop bits to “1” in the instruction queue. Therefore, insert a minimum of 8 NOPs after the WAIT instruction and the instruction that sets all clock stop bits to “1”.

Interrupts

(1) Reading address 00000_{16}

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.

  The interrupt request bit of the certain interrupt written in address 00000_{16} will then be set to “0”.

  Reading address 00000_{16} by software sets enabled highest priority interrupt source request bit to “0”.

  Though the interrupt is generated, the interrupt routine may not be executed.

  Do not read address 00000_{16} by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000_{16}. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt.

  Concerning the first instruction immediately after reset, generating any interrupt is prohibited.

(3) External interrupt

- When changing a polarity of pins $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$, the interrupt request bit may become "1". Clear the interrupt request bit after changing the polarity.

(4) Changing interrupt control register

See "Changing Interrupt Control Register".

Under development

Electrical characteristics

Table 1.37. Absolute maximum ratings

Symbol	Parameter	Condition	Rated value	Unit
Vcc	Supply voltage		- 0.3 to 7	V
AVcc	Analog supply voltage		- 0.3 to 7	V
V_I	Input voltage $\overline{\text{RESET}}$, CNVss, $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$, V_{REF}, X_{IN}		- 0.3 to Vcc + 0.3 (Note 1)	V
V_O	Output voltage $P0_0$ to $P0_7$, $P1_0$ to $P1_7$, $P3_0$ to $P3_5$, $P4_0$ to $P4_5$, $P5_0$ to $P5_4$, $P6_0$ to $P6_7$, $P7_0$, $P7_1$, V_{REF}, X_{IN}		- 0.3 to Vcc + 0.3	V
P_d	Power dissipation	Ta = 25 °C	1000 (Note 2)	mW
T_{opr}	Operating ambient temperature		- 20 to 85 (Note 3)	°C
T_{stg}	Storage temperature		- 40 to 150 (Note 4)	°C

Note 1: When writing to frash MCU, CNVss is –0.3 to 13 (V) .

Note 2: Flat package (56P6S-A) is 300 mW.

Note 3: Extended operating temperature version: -40 to 85 °C.

Note 4: Extended operating temperature version: -65 to 150 °C.

VCC = 5V

Table 1.38. Recommended operating conditions (Note 1)

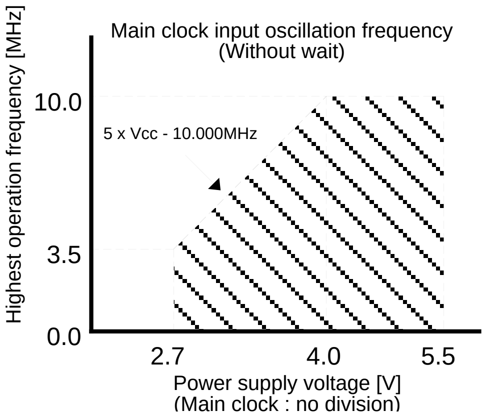
Symbol	Parameter				Standard			Unit
					Min	Typ.	Max.	
Vcc	Supply voltage (Note 2)	Mask ROM version			2.7	5.0	5.5	V
		Flash memory version			4.0	5.0	5.5	
AVcc	Analog supply voltage					Vcc		V
Vss	Supply voltage					0		V
AVss	Analog supply voltage					0		V
V_{IH}	HIGH input voltage	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P3$_0$ to P3$_5$, P4$_0$ to P4$_5$, P5$_0$ to P5$_4$, P6$_0$ to P6$_7$, P7$_0$, P7$_1$, X_{IN}, RESET, CNV_{SS},			0.8Vcc		Vcc	V
V_{IL}	LOW input voltage	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P3$_0$ to P3$_5$, P4$_0$ to P4$_5$, P5$_0$ to P5$_4$, P6$_0$ to P6$_7$, P7$_0$, P7$_1$, X_{IN}, RESET, CNV_{SS}			0		0.2Vcc	V
$I_{OH\ (peak)}$	HIGH peak output current	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P3$_0$ to P3$_5$, P4$_0$ to P4$_5$, P5$_0$ to P5$_4$, P6$_0$ to P6$_7$, P7$_0$, P7$_1$					- 10.0	mA
$I_{OL\ (peak)}$	LOW peak output current	P0$_0$ to P0$_7$, P3$_0$ to P3$_5$, P4$_0$ to P4$_5$, P5$_0$ to P5$_4$, P6$_0$ to P6$_7$, P7$_0$, P7$_1$					10.0	mA
$I_{OL\ (peak)}$	LOW peak output current	P1$_0$ to P1$_7$		HIGHPOWER			30.0	mA
				LOWPOWER			10.0	
$I_{OH\ (avg)}$	HIGH average output current	P0$_0$ to P0$_7$, P1$_0$ to P1$_7$, P3$_0$ to P3$_5$, P4$_0$ to P4$_5$, P5$_0$ to P5$_4$, P6$_0$ to P6$_7$, P7$_0$, P7$_1$					- 5.0	mA
$I_{OL\ (avg)}$	LOW average output current	P0$_0$ to P0$_7$, P3$_0$ to P3$_5$, P4$_0$ to P4$_5$, P5$_0$ to P5$_4$, P6$_0$ to P6$_7$, P7$_0$, P7$_1$					5.0	mA
$I_{OL\ (avg)}$	LOW average output current	P1$_0$ to P1$_7$		HIGHPOWER			15.0	mA
				LOWPOWER			5.0	
f (X_{IN})	Main clock input oscillation frequency	Without wait	Mask ROM version	V_{CC}=4.0V to 5.5V	0		10	MHz
				V_{CC}=2.7V to 4.0V	0		5 x V_{CC} - 10.000	MHz
			Flash memory version	V_{CC}=4.0V to 5.5V	0		10	MHz
		With wait	Mask ROM version	V_{CC}=4.0V to 5.5V	0		10	MHz
				V_{CC}=2.7V to 4.0V	0		2.31 x V_{CC} +0.760	MHz
			Flash memory version	V_{CC}=4.0V to 5.5V	0		10	MHz
f (X_{CIN})	Subclock oscillation frequency					32.768	50	kHz

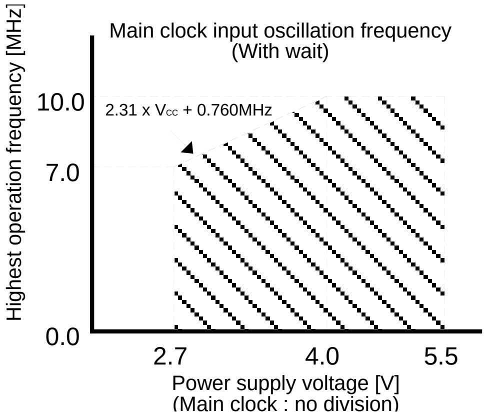
Note 1: Unless otherwise noted: VCC = 2.7V to 5.5V, Vss = 0V, Ta = – 20 to 85°C (Extended operating temperature version:– 40 to 85°C). Flash version: VCC = 4.0V to 5.5V, Vss = 0V, Ta = – 20 to 85°C (Extended operating temperature version:– 40 to 85°C.)

Note 2: Flash version: VCC = 4.0V to 5.5V

Note 3: The average output current is an average value measured over 100ms.

Note 4: Keep output current as follows:
The sum of port P3 and P4 I_{OL} (peak) is under 40 mA. The sum of port P1 I_{OL} (peak) is under 60 mA. The sum of port P1, P3 and P4 I_{OH} (peak) is under 40 mA. The sum of port P0, P5, P6 and P7 I_{OL} (peak) is under 80 mA. The sum of port P0, P5, P6 and P7 I_{OH} (peak) is under 80 mA.





V_{CC} = 5V

Table 1.39. Electrical characteristics (Note)

Symbol	Parameter			Measuring condition		Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	P0$_0$ to P0$_7$,P1$_0$ to P1$_7$,P3$_0$ to P3$_5$, P4$_0$ to P4$_5$,P5$_0$ to P5$_4$,P6$_0$ to P6$_7$,P7$_0$,P7$_1$		I_{OH} = - 5 mA		3.0			V
V_{OH}	HIGH output voltage	P0$_0$ to P0$_7$,P1$_0$ to P1$_7$,P3$_0$ to P3$_5$, P4$_0$ to P4$_5$,P5$_0$ to P5$_4$,P6$_0$ to P6$_7$,P7$_0$,P7$_1$		I_{OH} = - 200 µA		4.7			V
V_{OH}	HIGH output voltage	X_{OUT}	HIGHPOWER	I_{OH} = - 1 mA		3.0			V
			LOWPOWER	I_{OH} = - 0.5 mA		3.0			
V_{OH}	HIGH output voltage	X_{COUT}	HIGHPOWER	No load			3.0		V
			LOWPOWER	No load			1.6		
V_{OL}	LOW output voltage	P0$_0$ to P0$_7$,P3$_0$ to P3$_5$,P4$_0$ to P4$_5$ P5$_0$ to P5$_4$,P6$_0$ to P6$_7$,P7$_0$,P7$_1$		I_{OL} = 5 mA				2.0	V
V_{OL}	LOW output voltage	P0$_0$ to P0$_7$,P3$_0$ to P3$_5$,P4$_0$ to P4$_5$ P5$_0$ to P5$_4$,P6$_0$ to P6$_7$,P7$_0$,P7$_1$		I_{OL} = 200 µA				0.45	V
V_{OL}	LOW output voltage	P1$_0$ to P1$_7$	HIGHPOWER	I_{OL} = 15mA				2.0	V
			LOWPOWER	I_{OL} = 5 mA				2.0	
V_{OL}	LOW output voltage	P1$_0$ to P1$_7$	HIGHPOWER	I_{OL} = 200 µA				0.3	V
			LOWPOWER	I_{OL} = 200 µA				0.45	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	I_{OH} = 1 mA				2.0	V
			LOWPOWER	I_{OH} = 0.5 mA				2.0	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	No load			0		V
			LOWPOWER	No load			0		
V_{T+} -V_{T-}	Hysteresis	TA0$_{IN}$,TX0$_{INOUT}$,TX1$_{INOUT}$ TX2$_{INOUT}$,TB0$_{IN}$,TB1$_{IN}$ $\overline{INT_0}$,$\overline{INT_1}$, CLK$_0$				0.2		0.8	V
V_{T+} -V_{T-}	Hysteresis	$\overline{RESET}$				0.2		1.8	V
I_{IH}	HIGH input current	P0$_0$ to P0$_7$,P1$_0$ to P1$_7$,P3$_0$ to P3$_5$, P4$_0$ to P4$_5$,P5$_0$ to P5$_4$,P6$_0$ to P6$_7$ P7$_0$,P7$_1$,X_{IN}, RESET, CNVss		V_I = 5V				5.0	µA
I_{IL}	LOW input current	P0$_0$ to P0$_7$,P1$_0$ to P1$_7$,P3$_0$ to P3$_5$, P4$_0$ to P4$_5$,P5$_0$ to P5$_4$,P6$_0$ to P6$_7$, P7$_0$,P7$_1$,X_{IN}, RESET, CNVss		V_I = 0V				-5.0	µA
R_{PULLUP}	Pull-up resister	P0$_0$ to P0$_7$,P1$_0$ to P1$_7$,P3$_0$ to P3$_5$, P4$_0$ to P4$_5$,P5$_0$ to P5$_4$,P6$_0$ to P6$_7$,P7$_0$,P7$_1$		V_I = 0V		30.0	50.0	167.0	kΩ
R_{XIN}	Feedback resister	X_{IN}					1.0		MΩ
R_{XCIN}	Feedback resister	X_{CIN}					6.0		MΩ
V_{RAM}	RAM retention voltage			When clock is stopped		2.0			V
Icc	Power supply current			I/O pin has no load	f(X_{IN})=10MHz Square wave, no division		19.0	38.0	mA
					f(X_{CIN})=32kHz Square wave		90.0		µA
					f(X_{CIN})=32kHz With wait		4.0		µA
					Ta=25°C when clock is stopped			1.0	µA
					Ta=85°C when clock is stopped			20.0	

Note: Unless otherwise noted: V_{CC} = 5V, V_{SS} = 0V at Ta = 25^{o}C, f(X_{IN}) = 10MHz)

Under development

V_{CC} = 5V

Table 1.40. A-D conversion characteristics

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		V_{REF} =V_{CC}			10	Bits
–	Absolute accuracy	Sample & hold function not available	V_{REF} =V_{CC} = 5V			±3	LSB
		Sample & hold function available(10bit)	V_{REF} =V_{CC}= 5V			±3	LSB
		Sample & hold function available(8bit)	V_{REF} = V_{CC} = 5V			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC}	10		40	kohm
t_{CONV}	Conversion time(10bit)			3.3			µs
t_{CONV}	Conversion time(8bit)			2.8			µs
t_{SAMP}	Sampling time			0.3			µs
V_{REF}	Reference voltage			2		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 1.41. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		15	ns
t_f	External clock fall time		15	ns

Table 1.42. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	100		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	40		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	40		ns

Table 1.43. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	400		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	200		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	200		ns

Table 1.44. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	200		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	100		ns

Table 1.45. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA0IN input HIGH pulse width	100		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	100		ns

Table 1.46. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TA0OUT input cycle time	2000		ns
$t_{w(UPH)}$	TA0OUT input HIGH pulse width	1000		ns
$t_{w(UPL)}$	TA0OUT input LOW pulse width	1000		ns
$t_{su(UP-TIN)}$	TA0OUT input setup time	400		ns
$t_{h(TIN-UP)}$	TA0OUT input hold time	400		ns

V_{CC} = 5V

Timing requirements (referenced to V_{CC} = 5V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 1.47. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width (counted on both edges)	80		ns

Table 1.48. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	400		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width	200		ns

Table 1.49. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	400		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width	200		ns

Table 1.50. Timer X input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi$_{INOUT}$ input cycle time	100		ns
$t_{w(TXH)}$	TXi$_{INOUT}$ input HIGH pulse width	40		ns
$t_{w(TXL)}$	TXi$_{INOUT}$ input LOW pulse width	40		ns

Table 1.51. Timer X input (gate input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi$_{INOUT}$ input cycle time	400		ns
$t_{w(TXH)}$	TXi$_{INOUT}$ input HIGH pulse width	200		ns
$t_{w(TXL)}$	TXi$_{INOUT}$ input LOW pulse width	200		ns

Table 1.52. Timer X input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi$_{INOUT}$ input cycle time	200		ns
$t_{w(TXH)}$	TXi$_{INOUT}$ input HIGH pulse width	100		ns
$t_{w(TXL)}$	TXi$_{INOUT}$ input LOW pulse width	100		ns

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 1.53. Timer X input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	400		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	200		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	200		ns

Table 1.54. Timer X input (pulse width measurement mode)

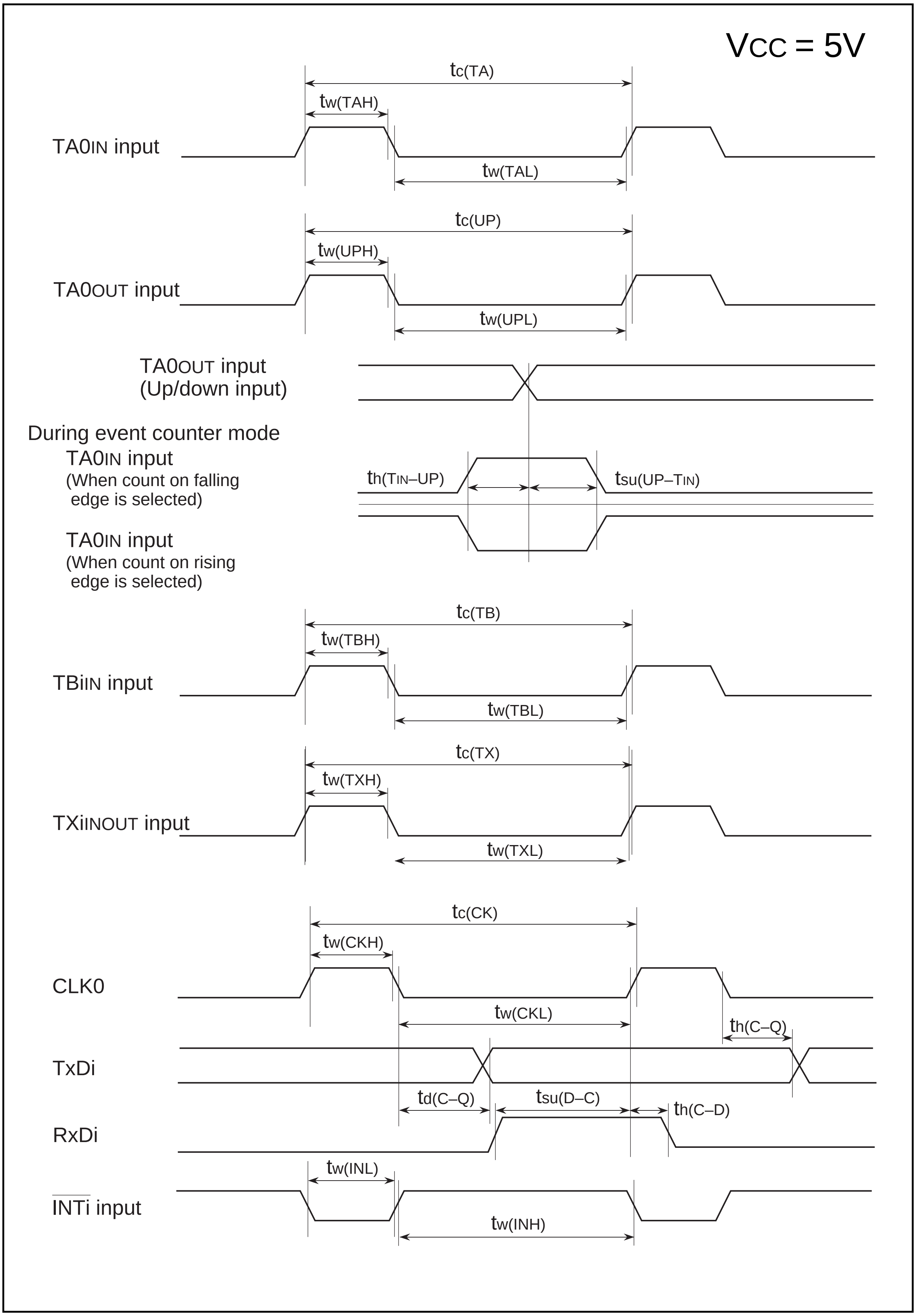
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXiINOUT input cycle time	400		ns
$t_{w(TXH)}$	TXiINOUT input HIGH pulse width	200		ns
$t_{w(TXL)}$	TXiINOUT input LOW pulse width	200		ns

Table 1.55. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK0 input cycle time	200		ns
$t_{w(CKH)}$	CLK0 input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLK0 input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 1.56. External interrupt $\overline{\text{INTi}}$ inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input LOW pulse width	250		ns


VCC = 5V
tc(TA)
tw(TAH)
TA0IN input
tw(TAL)
tc(UP)
tw(UPH)
TA0OUT input
tw(UPL)
TA0OUT input
(Up/down input)
During event counter mode
TA0IN input
(When count on falling edge is selected)
th(TIN–UP)
tsu(UP–TIN)
TA0IN input
(When count on rising edge is selected)
tc(TB)
tw(TBH)
TBiIN input
tw(TBL)
tc(TX)
tw(TXH)
TXiINOUT input
tw(TXL)
tc(CK)
tw(CKH)
CLK0
tw(CKL)
th(C–Q)
TxDi
td(C–Q)
tsu(D–C)
th(C–D)
RxDi
tw(INL)
INTi input
tw(INH)

Under development

V_{CC} = 3V

Table 1.57. Electrical characteristics (Note 1)

Symbol	Parameter			Measuring condition		Standard			Unit
						Min.	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OH} = - 1mA		2.5			V
V_{OH}	HIGH output voltage	X_{OUT}	HIGHPOWER	I_{OH} = - 1 mA		2.5			V
			LOWPOWER	I_{OH} = - 50 μA		2.5			
V_{OH}	HIGH output voltage	X_{COUT}	HIGHPOWER	No load			3.0		V
			LOWPOWER	No load			1.6		
V_{OL}	LOW output voltage	$P0_0$ to $P0_7$,$P3_0$ to $P3_5$,$P4_0$ to $P4_5$ $P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		I_{OL} = 1 mA				0.5	V
V_{OL}	LOW output voltage	$P1_0$ to $P1_7$	HIGHPOWER	I_{OL} = 3 mA				0.5	V
			LOWPOWER	I_{OL} = 1 mA				0.5	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	I_{OH} = 0.1 mA				0.5	V
			LOWPOWER	I_{OH} = 50 μA				0.5	
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	No load			0		V
			LOWPOWER	No load			0		
V_{T+} -V_{T-}	Hysteresis	$TA0_{IN}$,$TX0_{INOUT}$,$TX1_{INOUT}$, $TX2_{INOUT}$,$TB0_{IN}$,$TB1_{IN}$ $\overline{INT_0}$,$\overline{INT_1}$, CLK_0				0.2		0.8	V
V_{T+} -V_{T-}	Hysteresis	$\overline{RESET}$				0.2		1.8	V
I_{IH}	HIGH input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$, $P7_0$,$P7_1$,X_{IN}, RESET, CNVss		V_I = 3V				4.0	μA
I_{IL}	LOW input current	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$, $P7_0$,$P7_1$,X_{IN}, RESET, CNVss		V_I = 0V				-4.0	μA
R_{PULLUP}	Pull-up resistor	$P0_0$ to $P0_7$,$P1_0$ to $P1_7$,$P3_0$ to $P3_5$, $P4_0$ to $P4_5$,$P5_0$ to $P5_4$,$P6_0$ to $P6_7$,$P7_0$,$P7_1$		V_I = 0V		66.0	120.0	500.0	kΩ
R_{XIN}	Feedback resistor	X_{IN}					3.0		MΩ
R_{XIN}	Feedback resistor	X_{IN}					10.0		MΩ
V_{RAM}	RAM retention voltage			When clock is stopped		2.0			V
Icc	Power supply current			I/O pin has no load	f(X_{IN})=7MHz Square wave, no division		6.0	15.0	mA
					f(X_{CIN})=32kHz Square wave		40.0		μA
					f(X_{CIN})=32kHz With wait. Oscillation capacity HIGH (Note 2)		2.8		μA
					f(X_{CIN})=32kHz With wait. Oscillation capacity LOW (Note 2)		0.9		μA
					Ta=25°C when clock is stopped			1.0	μA
					Ta=85°C when clock is stopped			20.0	

Note 1: Unless otherwise noted: V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C, f(X_{IN}) = 7MHz, with wait)

Note 2: With one timer operated using f_{C32}.

V_{CC} = 3V

Table 1.58. A-D conversion characteristics

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max.	
–	Resolution		V_{REF} =V_{CC}			10	Bits
–	Absolute accuracy	Sample & hold function not available (8bit)	V_{REF} =V_{CC} = 3V, $Ø_{AD}$ = f_{AD}/2			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} =V_{CC}	10		40	kohm
t_{CONV}	Conversion time(8bit)			14.0			µs
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

VCC = 3V

Timing requirements (referenced to VCC = 3V, VSS = 0V at Ta = 25°C unless otherwise specified)

Table 1.59. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	143		ns
$t_{w(H)}$	External clock input HIGH pulse width	60		ns
$t_{w(L)}$	External clock input LOW pulse width	60		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Table 1.60. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	150		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	60		ns

Table 1.61. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	600		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	300		ns

Table 1.62. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TA0IN input cycle time	300		ns
$t_{w(TAH)}$	TA0IN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	150		ns

Table 1.63. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TA0IN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TA0IN input LOW pulse width	150		ns

Table 1.64. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TA0OUT input cycle time	3000		ns
$t_{w(UPH)}$	TA0OUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TA0OUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TA0OUT input setup time	600		ns
$t_{h(TIN-UP)}$	TA0OUT input hold time	600		ns

V_{CC} = 3V

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Ta = 25°C unless otherwise specified)

Table 1.65. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBi_{IN} input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width (counted on both edges)	160		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width (counted on both edges)	160		ns

Table 1.66. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	600		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width	300		ns

Table 1.67. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi_{IN} input cycle time	600		ns
$t_{w(TBH)}$	TBi_{IN} input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi_{IN} input LOW pulse width	300		ns

Table 1.68. Timer X input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi_{INOUT} input cycle time	150		ns
$t_{w(TXH)}$	TXi_{INOUT} input HIGH pulse width	60		ns
$t_{w(TXL)}$	TXi_{INOUT} input LOW pulse width	60		ns

Table 1.69. Timer X input (gate input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi_{INOUT} input cycle time	600		ns
$t_{w(TXH)}$	TXi_{INOUT} input HIGH pulse width	300		ns
$t_{w(TXL)}$	TXi_{INOUT} input LOW pulse width	300		ns

Table 1.70. Timer X input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TXi_{INOUT} input cycle time	300		ns
$t_{w(TXH)}$	TXi_{INOUT} input HIGH pulse width	150		ns
$t_{w(TXL)}$	TXi_{INOUT} input LOW pulse width	150		ns

$V_{CC} = 3V$

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Ta = 25^{o}C unless otherwise specified)

Table 1.71. Timer X input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TX_{iINOUT} input cycle time	600		ns
$t_{w(TXH)}$	TX_{iINOUT} input HIGH pulse width	300		ns
$t_{w(TXL)}$	TX_{iINOUT} input LOW pulse width	300		ns

Table 1.72. Timer X input (pulse width measurement mode)

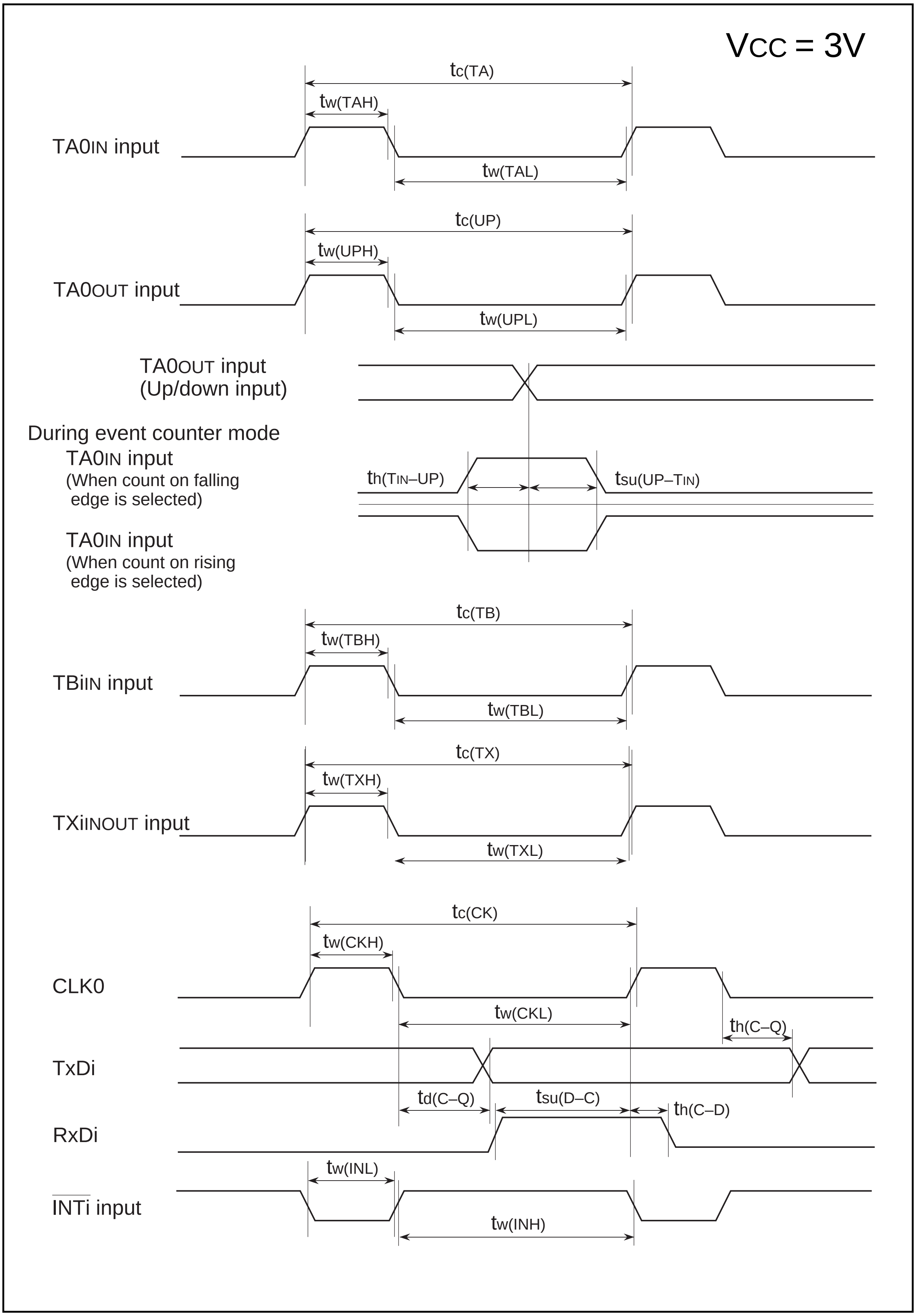
Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TX)}$	TX_{iINOUT} input cycle time	600		ns
$t_{w(TXH)}$	TX_{iINOUT} input HIGH pulse width	300		ns
$t_{w(TXL)}$	TX_{iINOUT} input LOW pulse width	300		ns

Table 1.73. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLK0 input cycle time	300		ns
$t_{w(CKH)}$	CLK0 input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLK0 input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	50		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 1.74. External interrupt $\overline{INTi}$ inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns


VCC = 3V
tc(TA)
tw(TAH)
TA0IN input
tw(TAL)
tc(UP)
tw(UPH)
TA0OUT input
tw(UPL)
TA0OUT input
(Up/down input)
During event counter mode
TA0IN input
(When count on falling edge is selected)
th(TIN–UP)
tsu(UP–TIN)
TA0IN input
(When count on rising edge is selected)
tc(TB)
tw(TBH)
TBiIN input
tw(TBL)
tc(TX)
tw(TXH)
TXiINOUT input
tw(TXL)
tc(CK)
tw(CKH)
CLK0
tw(CKL)
th(C–Q)
TxDi
td(C–Q)
tsu(D–C)
th(C–D)
RxDi
tw(INL)
INTi input
tw(INH)

Under development

Outline Performance

Table AA-1 shows the outline performance of the M30201 (flash memory version).

Table AA-1. Outline Performance of the M30201 (flash memory version)

Item		Performance
Power supply voltage		4.0V to 5.5 V (f(XIN)=10MHz)
Program/erase voltage		VPP=12V ± 5% (f(XIN)=10MHz)
		VCC=5V ± 5% (f(XIN)=10MHz)
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	See Figure 1.AA.3.
	Boot ROM area	One division (4 Kbytes) (Note 1)
Program method		In units of byte
Erase method		Collective erase
Program/erase control method		Program/erase control by software command
Number of commands		6 commands
Program/erase count		100 times
ROM code protect		Parallel I/O mode is supported.

Note: The boot ROM area contains a standard serial I/O mode control program which is stored in it when shipped from the factory. This area can be erased and programmed in only parallel I/O mode.

Preliminary

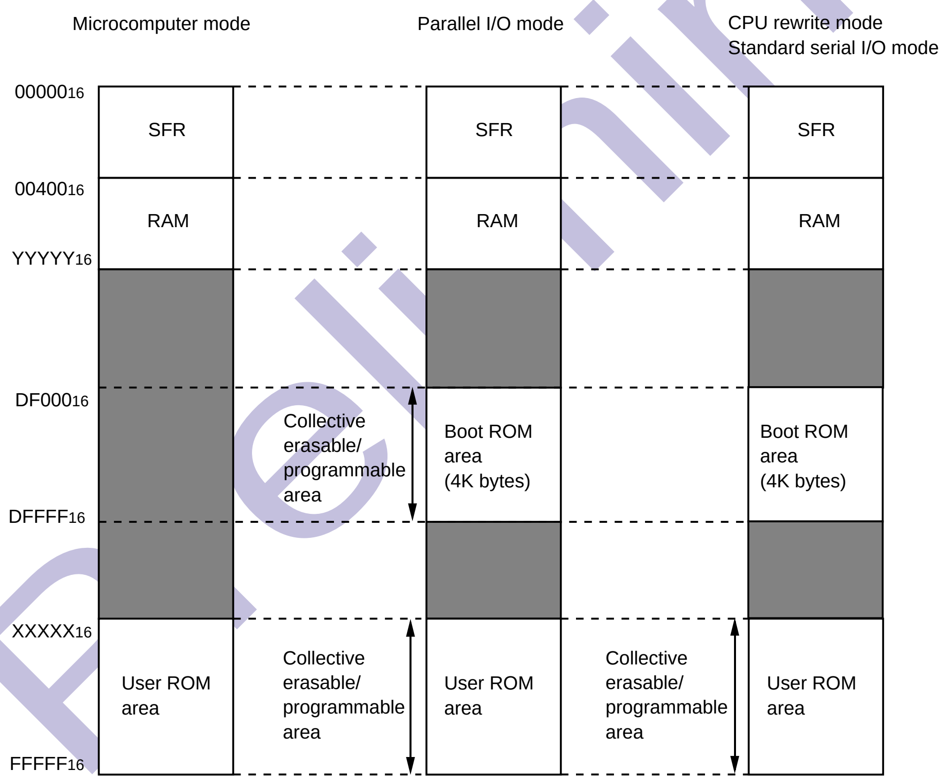
Under development

Flash Memory

The M30201 (flash memory version) contains the NOR type of flash memory that requires a high-voltage V_{PP} power supply for program/erase operations, in addition to the V_{CC} power supply for device operation. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

In addition to the ordinary user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This boot ROM area can be rewritten in only parallel I/O mode.

Preliminary



Note 1: In CPU rewrite and standard serial I/O modes, the user ROM is the only erasable/programmable area.
Note 2: In parallel I/O mode, the area to be erased/programmed can be selected by the address A17 input. The user ROM area is selected when this address input is high and the boot ROM area is selected when this address input is low.

Type No.	$XXXXX_{16}$	$YYYYY_{16}$
M30201F6	$F4000_{16}$	$00BFF_{16}$

Figure AA-3. Block diagram of flash memory version

Under development

CPU Rewrite Mode

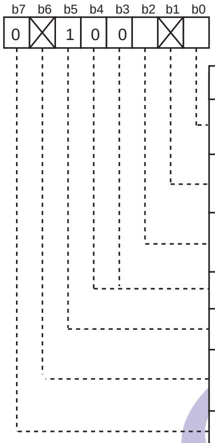
In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU). In CPU rewrite mode, the flash memory can be operated on by reading or writing to the flash memory control register and flash command register. Figure BB-1, Figure BB-2 show the flash memory control register, and flash command register respectively.

Also, in CPU rewrite mode, the CNVSS pin is used as the VPP power supply pin. Apply the power supply voltage, VPPH, from an external source to this pin.

In CPU rewrite mode, only the user ROM area shown in Figure AA-3 can be rewritten; the boot ROM area cannot be rewritten. Make sure the program and block commands are issued for only the user ROM area. The control program for CPU rewrite mode can be stored in either user ROM or boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to internal RAM before it can be executed.

Preliminary

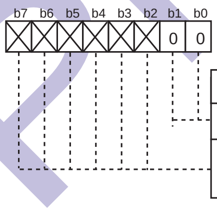
Flash memory control register 0



Symbol FCON0 Address 03B4₁₆ When reset 00100000₂

Bit symbol	Bit name	Function	R	W
FCON00	CPU rewrite mode select bit	0: CPU rewrite mode is invalid 1: CPU rewrite mode is valid	○	○
Reserved bit		This bit can not write. The value, if read, turns out to be indeterminate.	—	—
FCON02	CPU rewrite mode monitor flag	0: CPU rewrite mode is invalid 1: CPU rewrite mode is valid	○	—
Reserved bit		Must always be set to "0".	○	○
Reserved bit		Must always be set to "1".	○	○
Nothing is assigned. In an attempt to write this bit, write "0". The value, if read, turns out to be "0".			—	—
Reserved bit		Must always be set to "0".	○	○

Flash memory control register 1

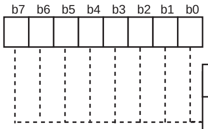


Symbol FCON1 Address 03B5₁₆ When reset XXXXXX00₂

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to "0".	○	○
Nothing is assigned. In an attempt to write these bits, write "0". The value, if read, turns out to be indeterminate.			—	—

Figure BB-1. Flash memory control register

Flash command register



Symbol FCMD Address 03B6₁₆ When reset 00₁₆

Function	R	W
Writing of software command <Software command name> <Command code> •Read command "00₁₆" •Program command "40₁₆" •Program verify command "C0₁₆" •Erase command "20₁₆" +"20₁₆" •Erase verify command "A0₁₆" •Reset command "FF₁₆" +"FF₆"	×	○

Figure BB-2. Flash command register

Under development

Preliminary

Microcomputer Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure AA-3 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNV$_{SS}$ pin low (V_{SS}). In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the P5$_2$ pin high (V_{CC}), the CNV$_{SS}$ pin high(V_{PPH}), the CPU starts operating using the control program in the boot ROM area. This mode is called the “boot” mode. The control program in the boot ROM area can also be used to rewrite the user ROM area.

CPU rewrite mode operation procedure

The internal flash memory can be operated on to program, read, verify, or erase it while being placed on-board by writing commands from the CPU to the flash memory control register (addresses 03B4$_{16}$, 03B5$_{16}$) and flash command register (address 03B6$_{16}$). Note that when in CPU rewrite mode, the boot ROM area cannot be accessed for program, read, verify, or erase operations. Before this can be accomplished, a CPU write control program must be written into the boot ROM area in parallel input/output mode. The following shows a CPU rewrite mode operation procedure.

<Start procedure (Note 1)>

(1) Apply V_{PP}H to the CNV$_{SS}$/V_{PP} pin and V_{CC} to the port P5$_2$ pin for reset release. Or the user can jump from the user ROM area to the boot ROM area using the JMP instruction and execute the CPU write control program. In this case, set the CPU write mode select bit of the flash memory control register to “1” before applying V_{PP}H to the CNV$_{SS}$/V_{PP} pin.

(2) After transferring the CPU write control program from the boot ROM area to the internal RAM, jump to this control program in RAM. (The operations described below are controlled by this program.)

(3) Set the CPU rewrite mode select bit to “1”.

(4) Read the CPU rewrite mode monitor flag to see that the CPU rewrite mode is enabled.

(5) Execute operation on the flash memory by writing software commands to the flash command register.

Note 1: In addition to the above, various other operations need to be performed, such as for entering the data to be written to flash memory from an external source (e.g., serial I/O), initializing the ports, and writing to the watchdog timer.

<Clearing procedure>

(1) Apply V_{SS} to the CNV$_{SS}$/V_{PP} pin.

(2) Set the CPU rewrite mode select bit to “0”.

Under development

Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

During erase/program mode, set BCLK to one of the following frequencies by changing the divide ratio:

5 MHz or less when wait bit (bit 7 at address 0005_{16}) = 0 (without internal access wait state)

10 MHz or less when wait bit (bit 7 at address 0005_{16}) = 1 (with internal access wait state)

(2) Instructions inhibited against use

The instructions listed below cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory:

UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts inhibited against use

No interrupts can be used that look up the fixed vector table in the flash memory area. Maskable interrupts may be used by setting the interrupt vector table in a location outside the flash memory area.

Preliminary

Under development

Software Commands

Table BB-1 lists the software commands available with the M30201 (flash memory version).
When CPU rewrite mode is enabled, write software commands to the flash command register to specify the operation to erase or program.
The content of each software command is explained below.

Table BB-1. List of Software Commands (CPU Rewrite Mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read	Write	$03B6_{16}$	00_{16}			
Program	Write	$03B6_{16}$	40_{16}	Write	Program address	Program data
Program verify	Write	$03B6_{16}$	$C0_{16}$	Read	Verify address	Verify data
Erase	Write	$03B6_{16}$	20_{16}	Write	$03B6_{16}$	20_{16}
Erase verify	Write	$03B6_{16}$	$A0_{16}$	Read	Verify address	Verify data
Reset	Write	$03B6_{16}$	FF_{16}	Write	$03B6_{16}$	FF_{16}

Read Command (00_{16})

The read mode is entered by writing the command code "00_{16}" to the flash command register in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D_0–D_7), 8 bits at a time.
The read mode is retained intact until another command is written.
After reset and after the reset command is executed, the read mode is set.

Program Command (40_{16})

The program mode is entered by writing the command code "40_{16}" to the flash command register in the first bus cycle. When the user execute an instruction to write byte data to the desired address (e.g., STE instruction) in the second bus cycle, the flash memory control circuit executes the program operation. The program operation requires approximately 20 µs. Wait for 20 µs or more before the user go to the next processing.
During program operation, the watchdog timer remains idle, with the value "$7FFF_{16}$" set in it.

Note 1: The write operation is not completed immediately by writing a program command once. The user must always execute a program-verify command after each program command executed. And if verification fails, the user need to execute the program command repeatedly until the verification passes. See Figure 1.BB.3 for an example of a programming flowchart.

Preliminary

Under development

Program-verify command ($C0_{16}$)

The program-verify mode is entered by writing the command code “$C0_{16}$” to the flash command register in the first bus cycle. When the user execute an instruction (e.g., LDE instruction) to read byte data from the address to be verified (the previously programmed address) in the second bus cycle, the content that has actually been written to the address is read out from the memory.

The CPU compares this read data with the data that it previously wrote to the address using the program command. If the compared data do not match, the user need to execute the program and program-verify operations one more time.

Erase command (20_{16} + 20_{16})

The flash memory control circuit executes an erase operation by writing command code “20_{16}” to the flash command register in the first bus cycle and the same command code to the flash command register again in the second bus cycle. The erase operation requires approximately 20 ms. Wait for 20 ms or more before the user go to the next processing.

Before this erase command can be performed, all memory locations to be erased must have had data “00_{16}” written to by using the program and program-verify commands. During erase operation, the watchdog timer remains idle, with the value “$7FFF_{16}$ set in it.

Note 1: The erase operation is not completed immediately by writing an erase command once. The user must always execute an erase-verify command after each erase command executed. And if verification fails, the user need to execute the erase command repeatedly until the verification passes. See Figure BB-3 for an example of an erase flowchart.

Erase-verify command ($A0_{16}$)

The erase-verify mode is entered by writing the command code “$A0_{16}$” to the flash command register in the first bus cycle. When the user execute an instruction to read byte data from the address to be verified (e.g., LDE instruction) in the second bus cycle, the content of the address is read out.

The CPU must sequentially erase-verify memory contents one address at a time, over the entire area erased. If any address is encountered whose content is not “FF_{16}” (not erased), the CPU must stop erase-verify at that point and execute erase and erase-verify operations one more time.

Note 1: If any unerased memory location is encountered during erase-verify operation, be sure to execute erase and erase-verify operations one more time. In this case, however, the user does not need to write data “00_{16}” to memory before erasing.

Preliminary

Reset command (FF16 + FF16)

The reset command is used to stop the program command or the erase command in the middle of operation. After writing command code “4016” or “2016” twice to the flash command register, write command code “FF16” to the flash command register in the first bus cycle and the same command code to the flash command register again in the second bus cycle. The program command or erase command is disabled, with the flash memory placed in read mode.

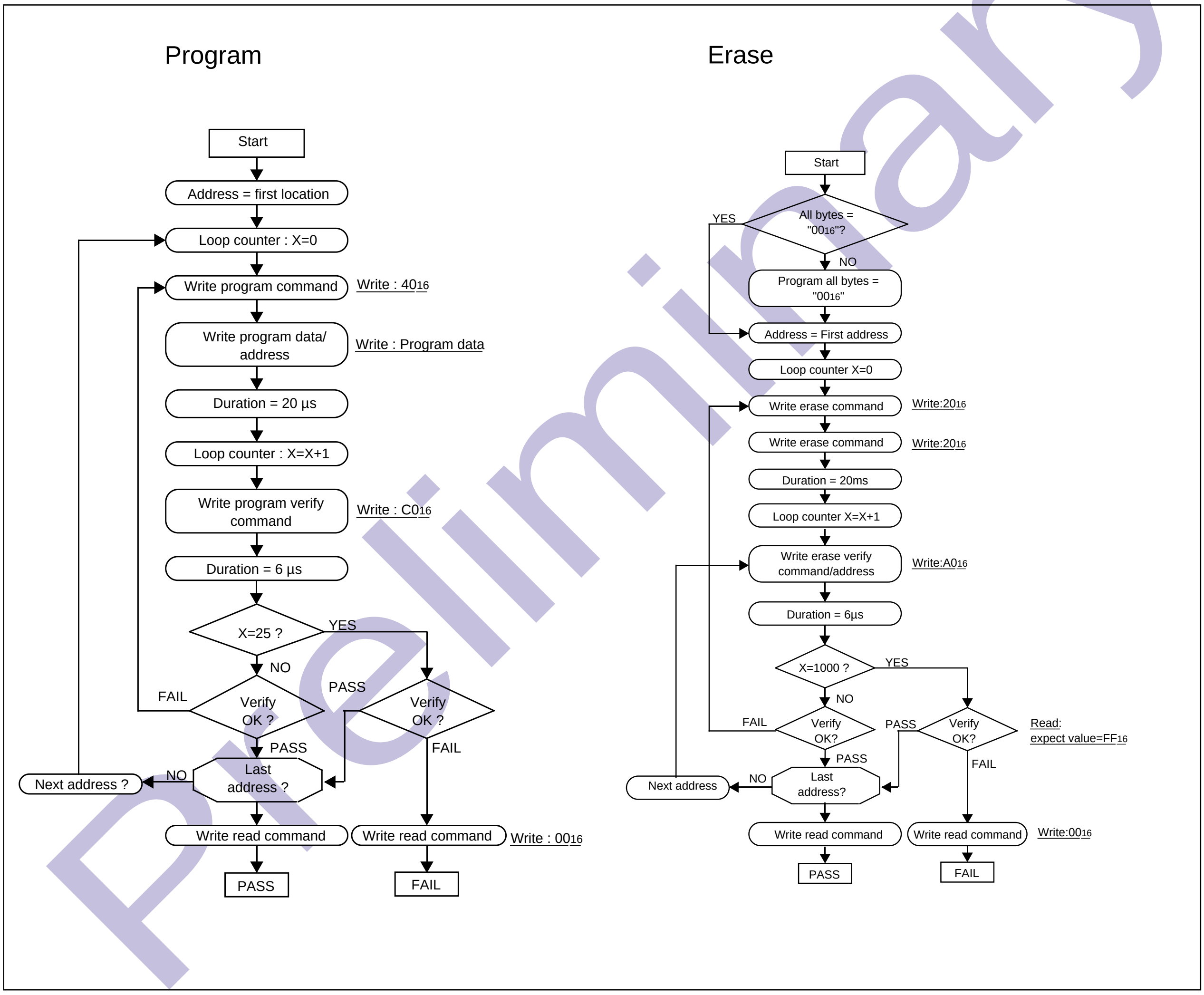


Figure BB-3. Program and erase execution flowchart in the CPU rewrite mode

Preliminary

Under development

Description of Pin Function (Flash Memory Parallel I/O Mode)

Pin name	Signal name	I/O	Function
VCC,VSS	Power supply input		Apply 5 V ± 10 % to the Vcc pin and 0 V to the Vss pin.
CNVSS	CNVSS	I	Apply 12 V ± 5 % to the CNVSS pin.
$\overline{\text{RESET}}$	Reset input	I	Connect this pin to VSS.
XIN	Clock input	I	Connect a ceramic or crystal resonator between the XIN and XOUT pins. When entering an externally derived clock, enter it from XIN and leave XOUT open.
XOUT	Clock output	O	
AVCC, AVSS	Analog power supply input		Connect AVSS to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Connect this pin to VSS.
P00 to P07	Data I/O D0 to D7	I/O	These are data D0–D7 input/output pins.
P10 to P17	Address input A8 to A15	I	These are address A8–A15 input pins.
P30 to P33	Address input A4 to A7	I	These are address A4–A7 input pins.
P34 to P35	Input port P3	I	Enter low signals to these pins.
P40	$\overline{\text{WE}}$ input	I	This is a $\overline{\text{WE}}$ input pin.
P41	$\overline{\text{OE}}$ input	I	This is a $\overline{\text{OE}}$ input pin.
P43	$\overline{\text{CE}}$ input	I	This is a $\overline{\text{CE}}$ input pin.
P42, P44, P45	Input port P4	I	Enter high signals or low signals to these pins.
P50	Address input A17	I	This is address A17 input pin.
P51	VRFY input	I	Apply VIH (5 V) to this pin when VPP = VPPH (12 V), or VIL (0 V) when VPP = VPPL (5 V).
P52	Input port P5	I	Enter low signal to this pin.
P53, P54	Input port P5	I	Enter high signals or low signals to these pins.
P60 to P63	Address input A0 to A3	I	These are address A0–A3 input pins.
P64 to P67	Input port P6	I	Enter high signals or low signals to these pins.
P70 to P71	Input port P7	I	Enter high signals or low signals to these pins.

Preliminary

Under development

Preliminary

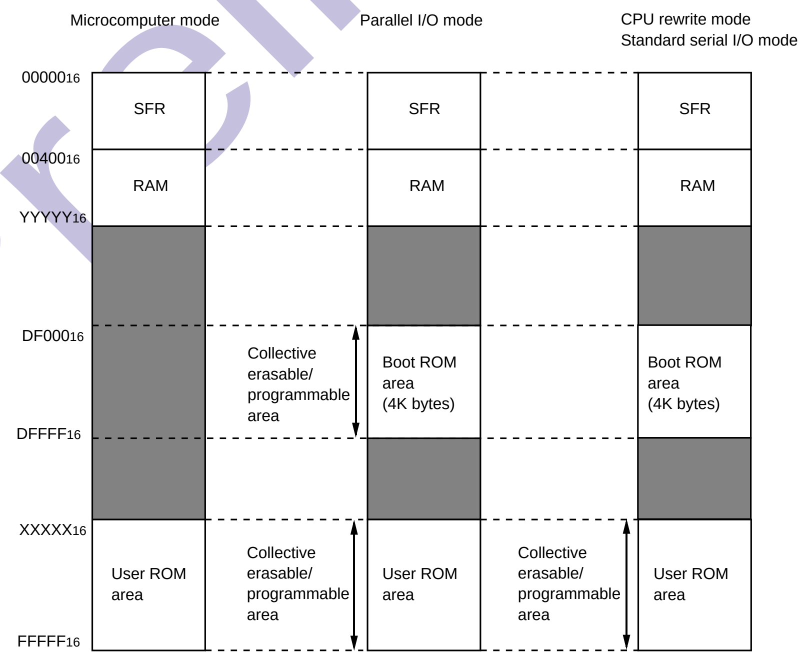
Parallel I/O Mode

The parallel I/O mode is entered by making connections shown in Figures CC-2 and CC-3 and then turning the V_{PPH} power supply on. In this mode, the M30201 (flash memory version) operates in a manner similar to the NOR flash memory M5M28F101 from Mitsubishi. Note, however, that there are some differences with regard to the functions not available with the microcomputer (function of read device identification code) and matters related to memory capacity.
Table CC-2 shows pin relationship between the M30201 and M5M28F101 in parallel I/O mode.

Table CC-2. Pin relationship in parallel I/O mode

	M30201(flash memory version)	M5M28F101
V_{CC}	V_{CC}	V_{CC}
V_{SS}	V_{SS}	V_{SS}
Address input	$P6_0$ to $P6_3$, $P3_0$ to $P3_3$, $P1_0$ to $P1_7$, $P5_0$	A_0 to A_{15}, A_{17}
Data I/O	$P0_0$ to $P0_7$	D_0 to D_7
$\overline{OE}$ input	$P4_1$	$\overline{OE}$
$\overline{CE}$ input	$P4_3$	$\overline{CE}$
$\overline{WE}$ input	$P4_0$	$\overline{WE}$
V_{RFY} input (Note)	$P5_1$	———

Note: The V_{RFY} input only selects read-only or read/write mode, and does not have any pin associated with it on the M5M28F101.



Note 1: In CPU rewrite and standard serial I/O modes, the user ROM is the only erasable/programmable area.
Note 2: In parallel I/O mode, the area to be erased/programmed can be selected by the address A_{17} input. The user ROM area is selected when this address input is high and the boot ROM area is selected when this address input is low.

Type No.	$XXXXX_{16}$	$YYYYY_{16}$
M30201F6	$F4000_{16}$	$00BFF_{16}$

Figure CC-1. Block diagram of flash memory version

Under development

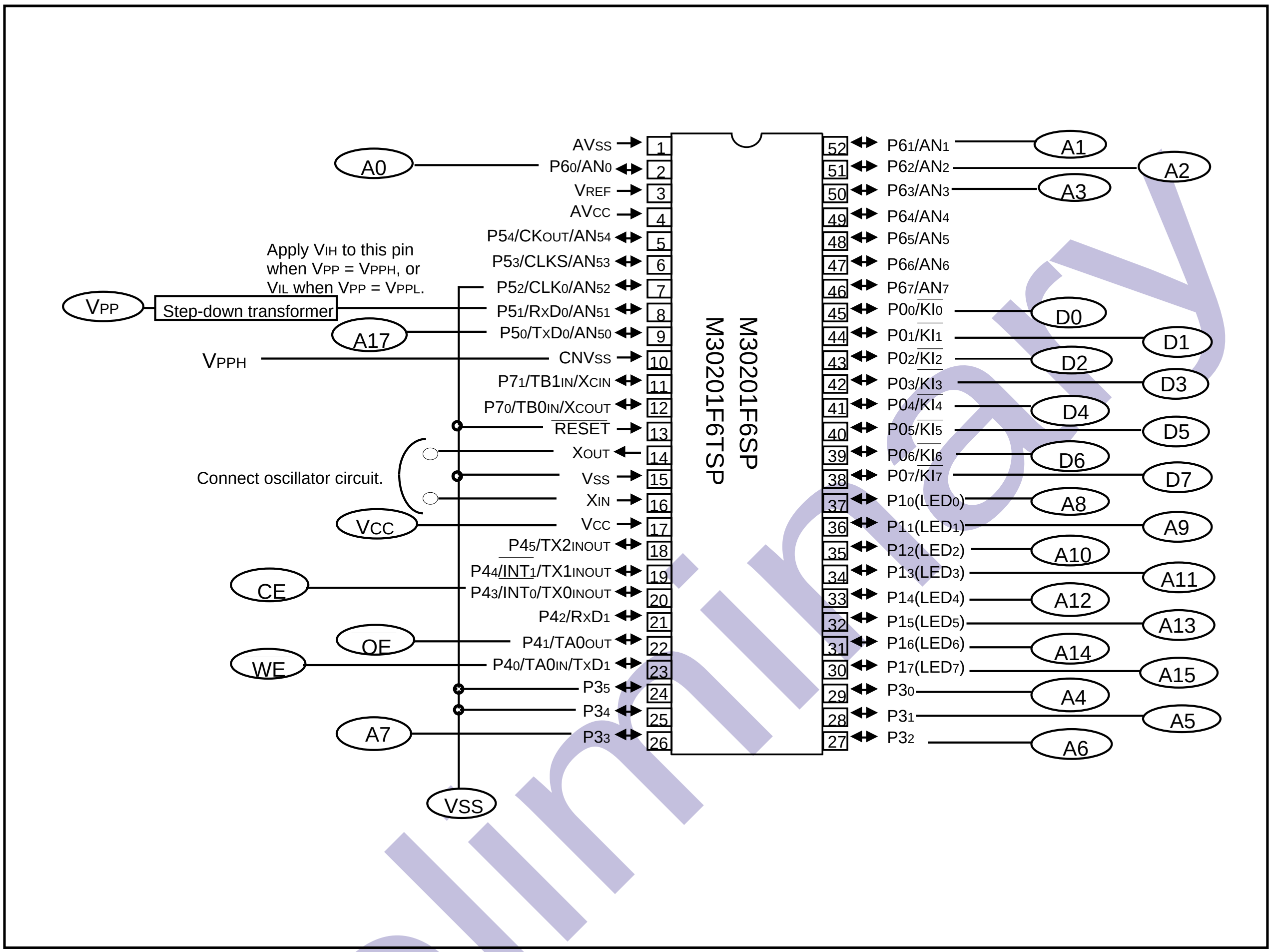
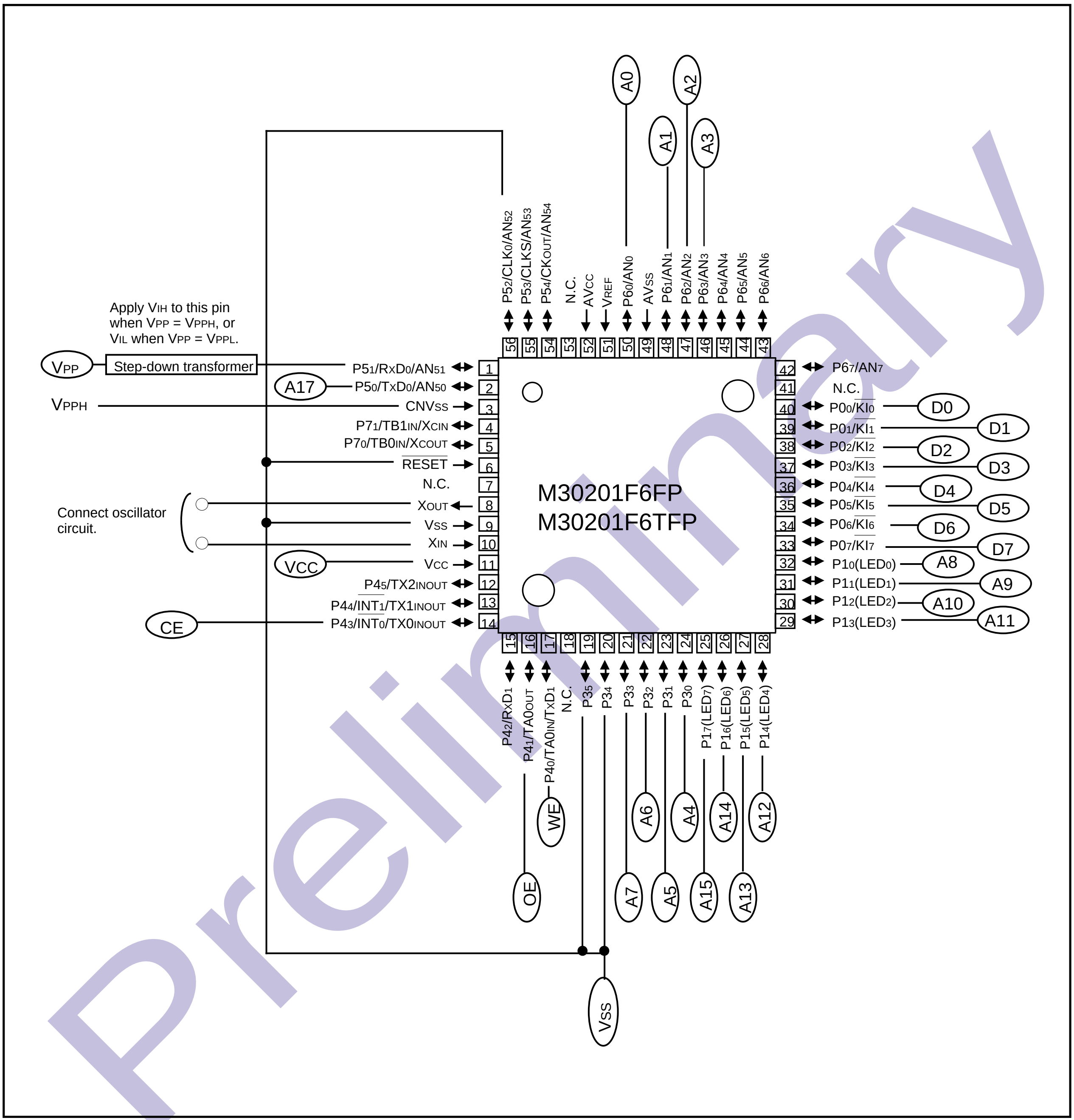


Figure CC-2. Pin connection diagram in parallel I/O mode (1)

Preliminary

Under development



Preliminary

Figure CC-3. Pin connection diagram in parallel I/O mode (2)

Under development

User ROM and Boot ROM Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure CC-1 can be rewritten.

In the boot ROM area, an erase block operation is applied to only one 4 K byte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the Mitsubishi factory. Therefore, using the device in standard serial input/output mode, the user does not need to write to the boot ROM area.

Functional Outline (Parallel I/O Mode)

In parallel I/O mode, bus operation modes—Read, Output Disable, Standby, and Write—are selected by the status of the $\overline{CE}$, $\overline{OE}$, $\overline{WE}$, V_{RFY}, and CNV_{SS} input pins.

The contents of erase, program, and other operations are selected by writing a software command. The data in memory can only be read out by a read after software command input.

Program and erase operations are controlled using software commands.

Preliminary

Table CC-3. Relationship between control signals and bus operation modes

Mode \ Pin name		$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	V_{RFY}	V_{PP}	D_0 to D_7
Read only	Read	V_{IL}	V_{IL}	V_{IH}	V_{IL}	V_{PPH}	Data output
	Output disabled	V_{IL}	V_{IH}	V_{IH}	V_{IL}	V_{PPH}	Hi-Z
	Stand by	V_{IH}	X	X	V_{IL}	V_{PPH}	Hi-Z
Read/ Write	Read	V_{IL}	V_{IL}	V_{IH}	V_{IH}	V_{PPH}	Data output
	Output disabled	V_{IL}	V_{IH}	V_{IH}	V_{IH}	V_{PPH}	Hi-Z
	Stand by	V_{IH}	X	X	V_{IH}	V_{PPH}	Hi-Z
	Write	V_{IL}	V_{IH}	V_{IL}	V_{IH}	V_{PPH}	Data input

Note: X can be V_{IL} or V_{IH}.

Under development

The following explains about bus operation modes, software commands, and status register.

Bus Operation Modes

Read-only mode is entered by applying V_{PPH} to the CNV_{SS} pin and a low voltage to the V_{RFY} pin. Read-only mode has three states: Read, Output Disable, and Standby which are selected by setting the $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ pins high or low.

Read-write mode is entered by applying V_{PPH} to the CNV_{SS} pin and a high voltage to the V_{RFY} pin. Read-write mode has four states: Read, Output Disable, Standby, and Write which are selected by setting the $\overline{CE}$, $\overline{OE}$, and $\overline{WE}$ pins high or low.

Read

The Read mode is entered by pulling the $\overline{WE}$ pin high when the $\overline{CE}$ and $\overline{OE}$ pins are low. In Read mode, the data corresponding to each software command entered is output from the data I/O pins D_0–D_7.

Output Disable

The Output Disable mode is entered by pulling the $\overline{CE}$ pin low and the $\overline{WE}$ and $\overline{OE}$ pins high. Also, the data I/O pins are placed in the high-impedance state.

Standby

The Standby mode is entered by driving the $\overline{CE}$ pin high. Also, the data I/O pins are placed in the high-impedance state.

Write

The Write mode is entered by applying V_{PPH} to the CNV_{SS} pin and a high voltage to the V_{RFY} pin and then pulling the $\overline{WE}$ pin low when the $\overline{CE}$ pin is low and $\overline{OE}$ pin is high. In this mode, the device accepts the software commands or write data entered from the data I/O pins. A program, erase, or some other operation is initiated depending on the content of the software command entered here. The input data such as address is latched at the falling edge of $\overline{WE}$ pin. The input data such as software command is latched at the rising edge of $\overline{WE}$ pin.

Preliminary

Under development

Software Commands

Table CC-4 lists the software commands available with the M30201 (flash memory version). By entering a software command from the data I/O pins (D_0–D_7) in Write mode, specify the content of the operation, such as erase or program operation, to be performed.

The following explains the content of each software command.

Table CC-4. Software command list (parallel I/O mode)

Command	First bus cycle			Second bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read	Write	x	00_{16}			
Program	Write	x	40_{16}	Write	Program address	Program data
Program verify	Write	x	$C0_{16}$	Read	x	Verify data
Erase	Write	x	20_{16}	Write	x	20_{16}
Erase verify	Write	Verify address	$A0_{16}$	Read	x	Verify data
Reset	Write	x	FF_{16}	Write	x	FF_{16}

Read Command (00_{16})

The read mode is entered by writing the command code “00_{16}” in the first bus cycle. When an address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data I/O pins (D_0–D_7).

The read mode is retained intact until another command is written.

After reset and after the reset command is executed, the read mode is set.

Program Command (40_{16})

The program mode is entered by writing the command code “40_{16}” in the first bus cycle. When an address and data to be program is write in the second bus cycle, the flash memory control circuit executes the program operation. The program operation requires approximately 20 µs. Wait for 20 µs or more before the user go to the next processing.

Note 1: The write operation is not completed immediately by writing a program command once. The user must always execute a program-verify command after each program command executed. And if verification fails, the user need to execute the program command repeatedly until the verification passes. See Figure CC-4 for an example of a programming flowchart.

Preliminary

Under development

Program-verify command ($C0_{16}$)

The program-verify mode is entered by writing the command code “$C0_{16}$” in the first bus cycle and the verify data is output from the data I/O pins (D_0–D_7) in the second bus cycle.

Erase command (20_{16} + 20_{16})

The flash memory control circuit executes an erase operation by writing command code “20_{16}” in the first bus cycle and the same command code again in the second bus cycle. The erase operation requires approximately 20 ms. Wait for 20 ms or more before the user go to the next processing. Before this erase command can be performed, all memory locations to be erased must have had data “00_{16}” written to by using the program and program-verify commands.

Note 1: The erase operation is not completed immediately by writing an erase command once. The user must always execute an erase-verify command after each erase command executed. And if verification fails, the user need to execute the erase command repeatedly until the verification passes. See Figure CC-4 for an example of an erase flowchart.

Erase-verify command ($A0_{16}$)

The erase-verify mode is entered by writing the command code “$A0_{16}$” in the first bus cycle and the verify data is output from the data I/O pins (D_0–D_7) in the second bus cycle.

Note 1: If any unerased memory location is encountered during erase-verify operation, be sure to execute erase and erase-verify operations one more time. In this case, however, the user does not need to write data “00_{16}” to memory before erasing.

Preliminary

Under development

Reset command (FF_{16} + FF_{16})

The reset command is used to stop the program command or the erase command in the middle of operation. After writing command code “40_{16}” or “20_{16}” twice, write command code “FF_{16}” in the first bus cycle and the same command code again in the second bus cycle. The program command or erase command is disabled, with the flash memory placed in read mode.

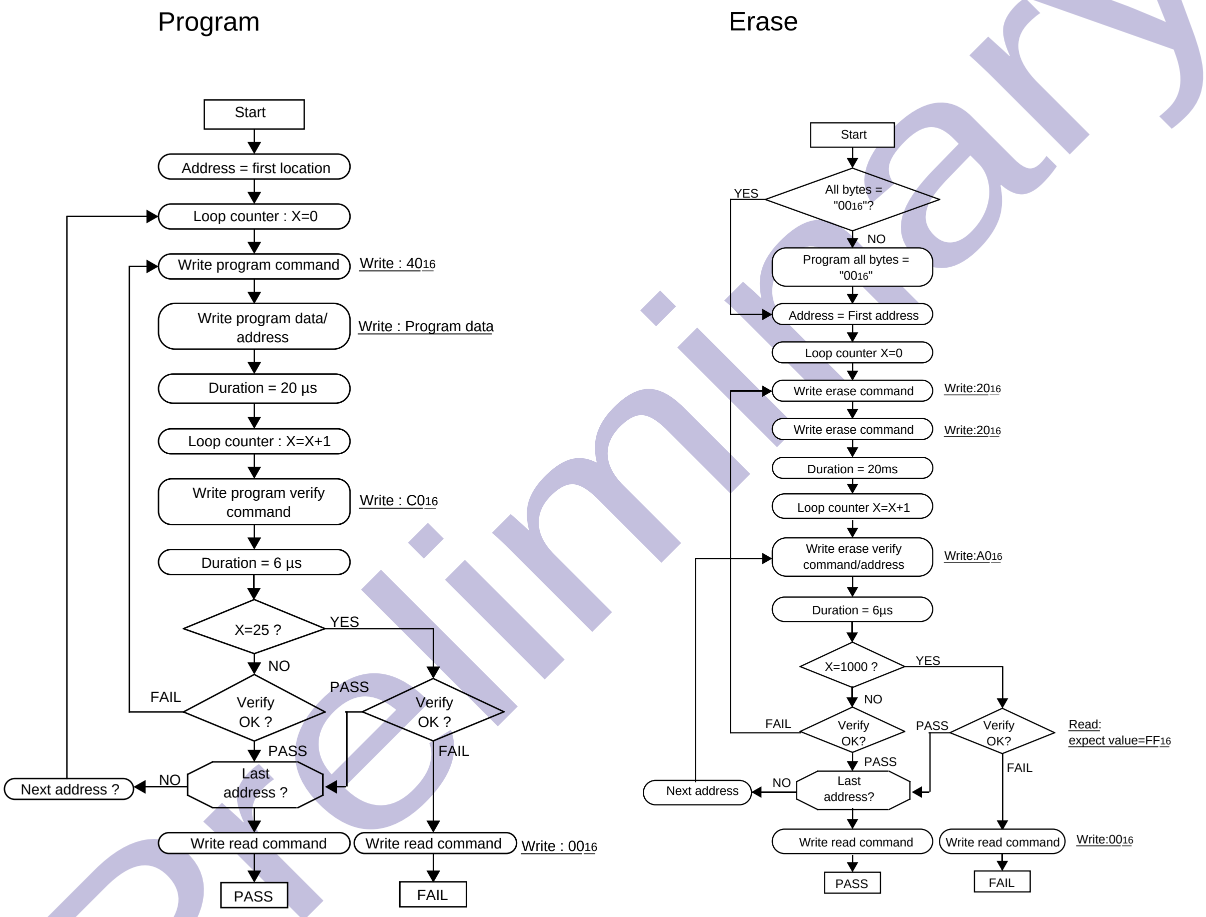


Figure CC-4. Program and erase execution flowchart in the CPU rewrite mode

Preliminary

Under development

Protect function

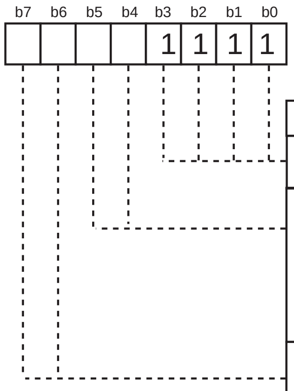
In parallel I/O mode, the internal flash memory has the “protect function” available. This function protects the flash memory contents from being read or rewritten easily.

Depending on the content at the protect control address (FFFFF16) in parallel I/O mode, this function inhibits the flash memory contents against read or modification. The protect control address (FFFFF16) is shown in Figure CC-5 . (This address exists in the user ROM area.)

The protect function is enabled by setting one of the two protect set bits to “0”, so that the internal flash memory contents are inhibited against read or modification. The protect function is disabled by setting both of the two protect reset bits to “00”, so that the internal flash memory contents can be read or modified. Once the protect function is set, the user cannot change settings of the protect clear bits while in parallel I/O mode. Settings of the protect reset bits can only be changed in CPU rewrite mode.

Protect control address

Symbol ROMCP
Address FFFFF16
When shipping FF16



Bit symbol	Bit name	Function
Reserved bit		Always set to "1".
ROMCR	Protect reset bit	b5 b4 00: Protect removed 01: Protect set bit effective 10: Protect set bit effective 11: Protect set bit effective
ROMCP	Protect set bit	b7 b6 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled

Note 1: When protect is turned on, the flash memory version is protected against readout or modification in parallel I/O mode.
Note 2: The protect reset bits can be used to turn off protect . However, since these bits cannot be changed in parallel I/O mode, they need to be rewritten in CPU rewrite mode.

Figure CC-5. Protect control address

Under development

Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
VCC,VSS	Power input		Apply 5V ± 10 % to Vcc pin and 0 V to Vss pin.
CNVSS	CNVSS	I	Apply 12V ± 5 % to this pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
AVCC, AVSS	Analog power supply input		Connect AVSS to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for AD from this pin.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open.
P30 to P35	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P45	Input port P4	I	Input "H" or "L" level signal or open.
P54	Input port P5	I	Input "H" or "L" level signal or open.
P50	TxD output	O	Serial data output pin.
P51	RxD input	I	Serial data input pin.
P52	SCLK input	I	Serial clock input pin.
P53	BUSY output	O	BUSY signal output pin.
P60 to P67	Input port P6	I	Input "H" or "L" level signal or open.
P70 to P71	Input port P7	I	Input "H" or "L" level signal or open.

Preliminary

Under development

Mode setup method

Signal	Value
CNVss	VPPH
RESET	Vss → Vcc

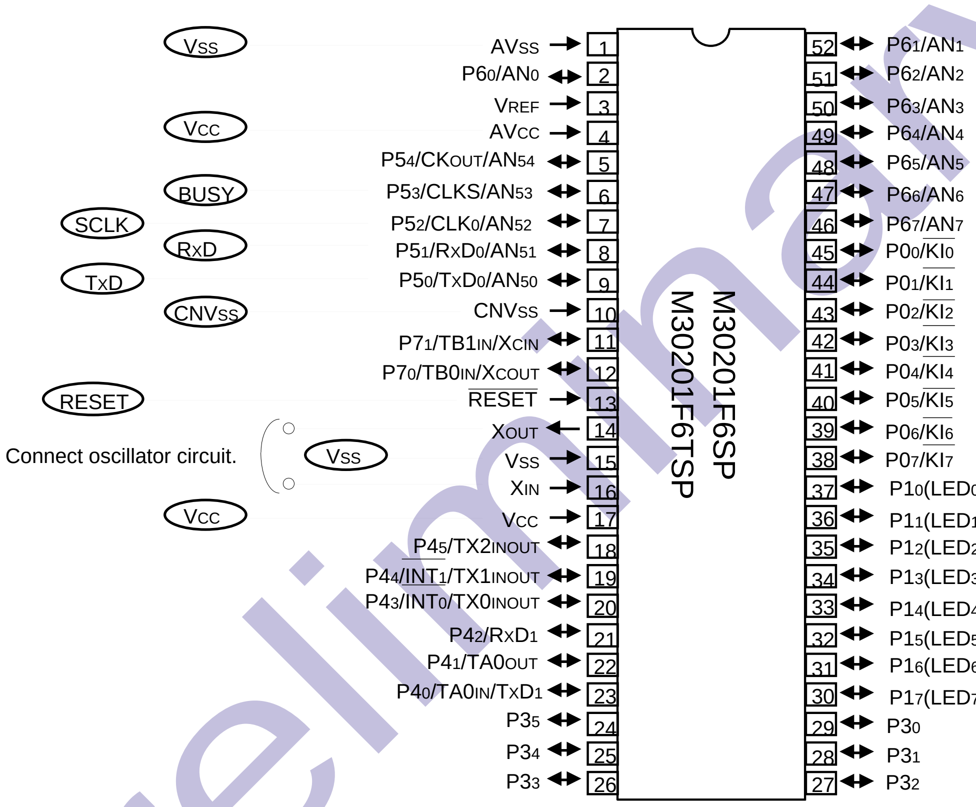
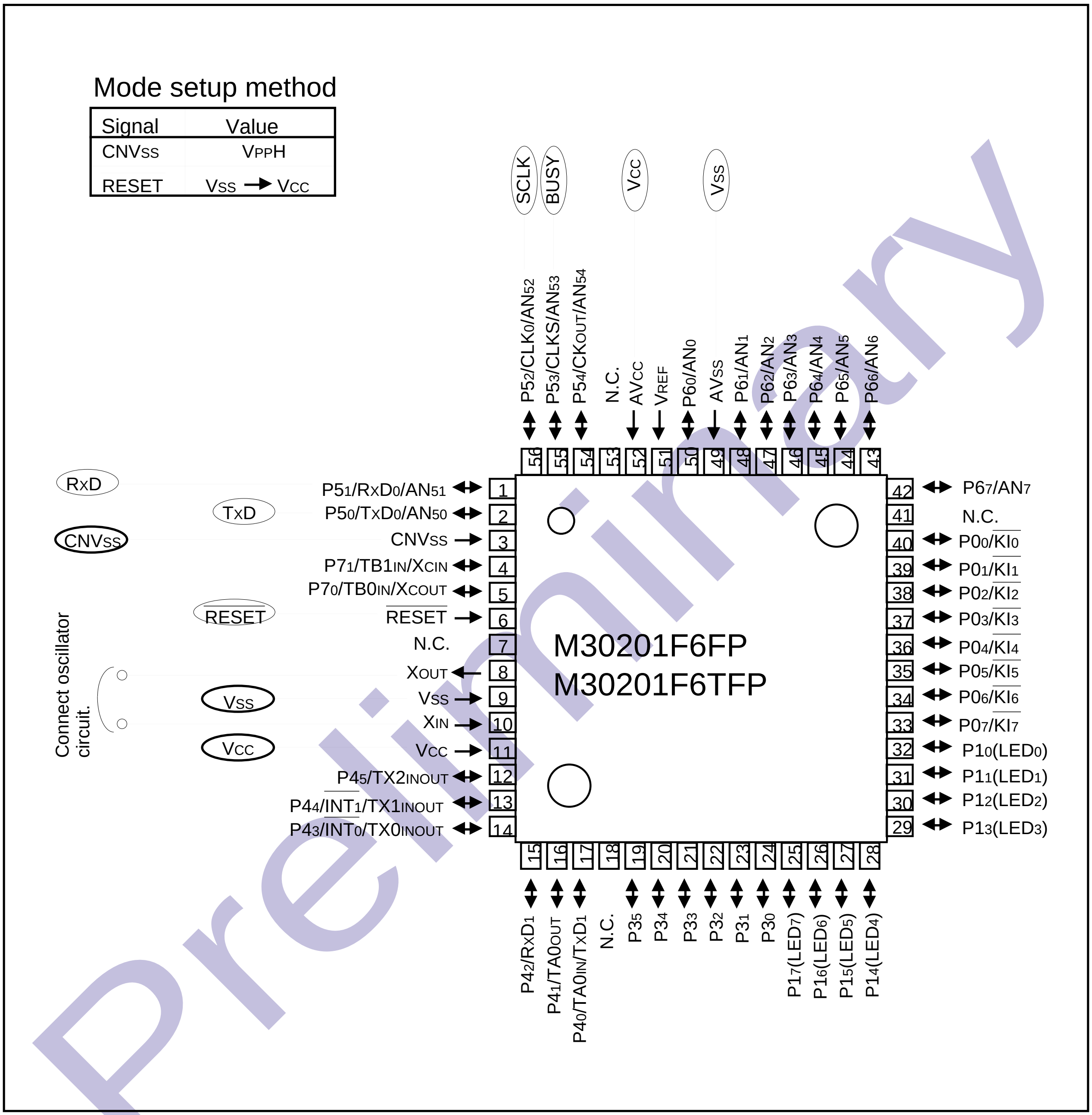


Figure DD-1. Pin connections for serial I/O mode (1)

Preliminary

Under development



Mode setup method

Signal	Value
CNVSS	VPPH
RESET	VSS → VCC

Preliminary

Figure DD-2. Pin connections for serial I/O mode (2)

Under development

Standard Serial I/O Mode

The standard serial I/O mode serially inputs and outputs the software commands, addresses and data necessary for operating (read, program, erase, etc.) the internal flash memory. It uses a purpose-specific serial programmer.

The standard serial I/O mode differs from the parallel I/O mode in that the CPU controls operations like rewriting (uses the CPU rewrite mode) in the flash memory or serial input for rewriting data. The standard serial I/O mode is started by clearing the reset with V$_{PPH}$ at the CNVss pin. (For the normal microprocessor mode, set CNVss to “L”.)

This control program is written in the boot ROM area when shipped from Mitsubishi Electric. Therefore, if the boot ROM area is rewritten in the parallel I/O mode, the standard serial I/O mode cannot be used.

Figures DD-1 and DD-2 show the pin connections for the standard serial I/O mode. Serial data I/O uses three UART0 pins: CLK$_0$, RxD$_0$, and TxD$_0$ and port P5$_3$ (BUSY).

The CLK$_0$ pin is the transfer clock input pin and it transfers the external transfer clock. The TxD$_0$ pin outputs the CMOS signal. The P5$_3$ (BUSY) pin outputs an “L” level when reception setup ends and an “H” level when the reception operation starts. Transmission and reception data is transferred serially in 8-byte blocks.

In the standard serial I/O mode, only the user ROM area shown in Figure CC-1 can be rewritten, the boot ROM area cannot.

The standard serial I/O mode has a 7-byte ID code. When the flash memory is not blank and the ID code does not match the content of the flash memory, the command sent from the programmer is not accepted.

Function Overview (Standard Serial I/O Mode)

In the standard serial I/O mode, software commands, addresses and data are input and output between the flash memory and an external device (serial programmer, etc.) using a clock synchronized serial I/O (UART0) and P5$_3$. In reception, the software commands, addresses and program data are synchronized with the rise of the transfer clock input to the CLK$_0$ pin and input into the flash memory via the RxD$_0$ pin. In transmission, the read data and status are synchronized with the fall of the transfer clock and output to the outside from the TxD$_0$ pin.

The TxD$_1$ pin is CMOS output. Transmission is in 8-bit blocks and LSB first.

When busy, either during transmission or reception, or while executing an erase operation or program, the P5$_3$ (BUSY) pin is “H” level. Accordingly, do not start the next transmission until the P5$_3$ (BUSY) pin is “L” level.

Also, data in memory and the status register can be read after inputting a software command. It is possible to check flash memory operating status or whether a program or erase operation ended successfully or in error by reading the status register.

Software commands and the status register are explained here following.

Preliminary

Under development

Software Commands

Table DD-1 lists software commands. In the standard serial I/O mode, erase operations, programs and reading are controlled by transferring software commands via the RxD pin. Software commands are explained here below.

Preliminary

Table DD-1. Software commands (Standard serial I/O mode)

	Control command		2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verificate
1	Page read	FF$_{16}$	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41$_{16}$	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Erase all unlocked blocks	A7$_{16}$	D0$_{16}$						Not acceptable
4	Read status register	70$_{16}$	SRD output	SRD1 output					Acceptable
5	Clear status register	50$_{16}$							Not acceptable
6	Read lockbit status	71$_{16}$	Address (middle)	Address (high)	Lock bit data output				Not acceptable
7	ID check function	F5$_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
8	Download function	FA$_{16}$	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
9	Version data output function	FB$_{16}$	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
14	Boot area output function	FC$_{16}$	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable

Note1: Shading indicates transfer from flash memory microcomputer to serial programmer. All other data is transferred from the serial programmer to the flash memory microcomputer.
Note2: SRD refers to status register data. SRD1 refers to status register 1 data.
Note3: All commands can be accepted when the flash memory is totally blank.

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Send the “FF_{16}” command code in the 1st byte of the transmission.

(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the rise of the clock.

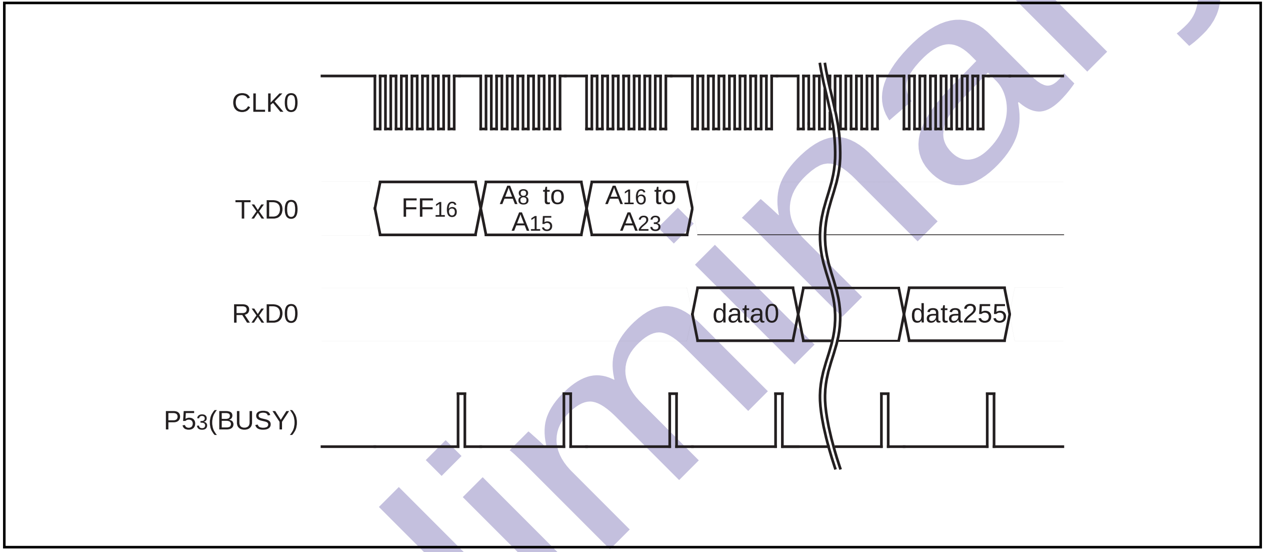


Figure DD-3. Timing for page read

Read Status Register Command

This command reads status information. When the “70_{16}” command code is sent in the 1st byte of the transmission, the contents of the status register (SRD) specified in the 2nd byte of the transmission and the contents of status register 1 (SRD1) specified in the 3rd byte of the transmission are read.

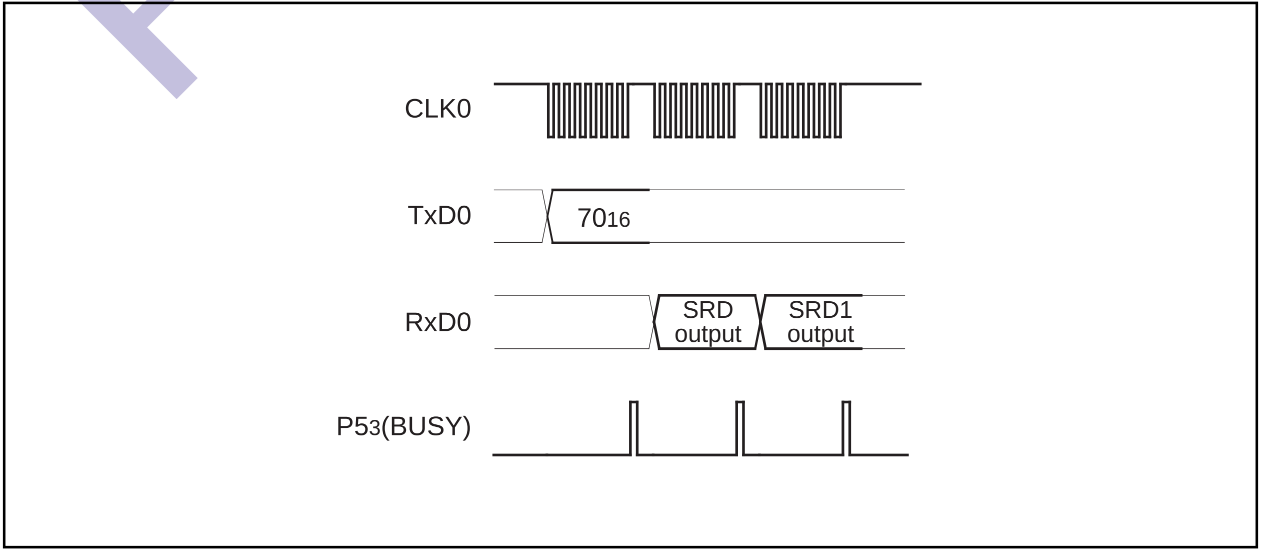


Figure DD-4. Timing for reading the status register

Preliminary

Under development

Clear Status Register Command

This command clears the bits (SR3–SR4) which are set when the status register operation ends in error. When the "50_{16}" command code is sent in the 1st byte of the transmission, the aforementioned bits are cleared. When the clear status register operation ends, the $P5_3$ (BUSY) signal changes from the "H" to the "L" level.

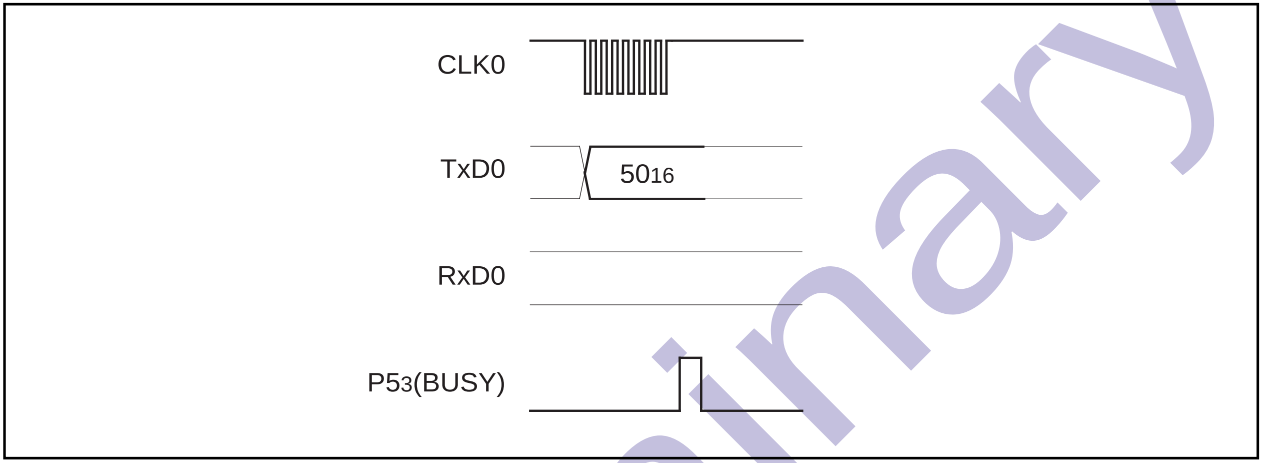


Figure DD-5. Timing for clearing the status register

Preliminary

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Send the "41_{16}" command code in the 1st byte of the transmission.
(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.
(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the $P5_3$ (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

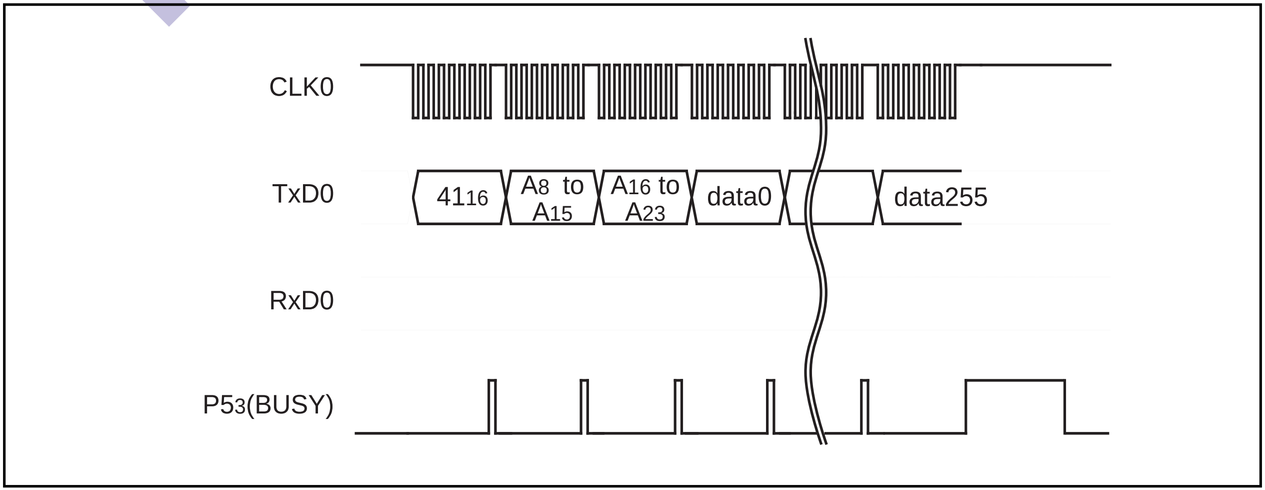


Figure DD-6. Timing for the page program

Under development

Preliminary

Erase All Unlocked Blocks Command

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

(1) Send the "$A7_{16}$" command code in the 1st byte of the transmission.

(2) Send the verify command code "$D0_{16}$" in the 2nd byte of the transmission. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the $P5_3$ (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register.

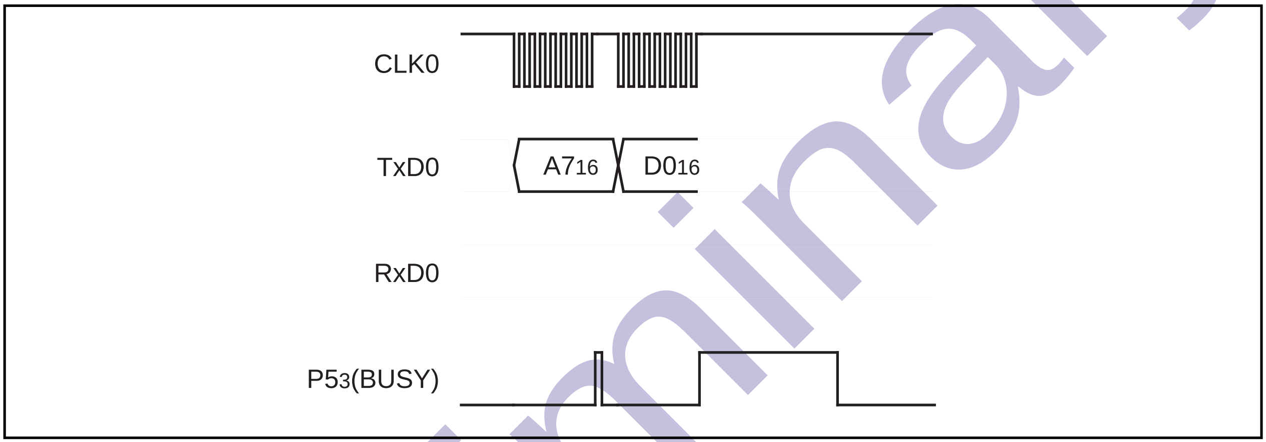


Figure DD-8. Timing for erasing all unlocked blocks

Read Lock Bit Status Command

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

(1) Send the "71_{16}" command code in the 1st byte of the transmission.

(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.

(3) The lock bit data of the specified block is output in the 4th byte of the transmission. Write the highest address of the specified block for addresses A_8 to A_{23}.

The M30201 (flash memory version) does not have the lock bit, so the read value is always "1" (block unlock).

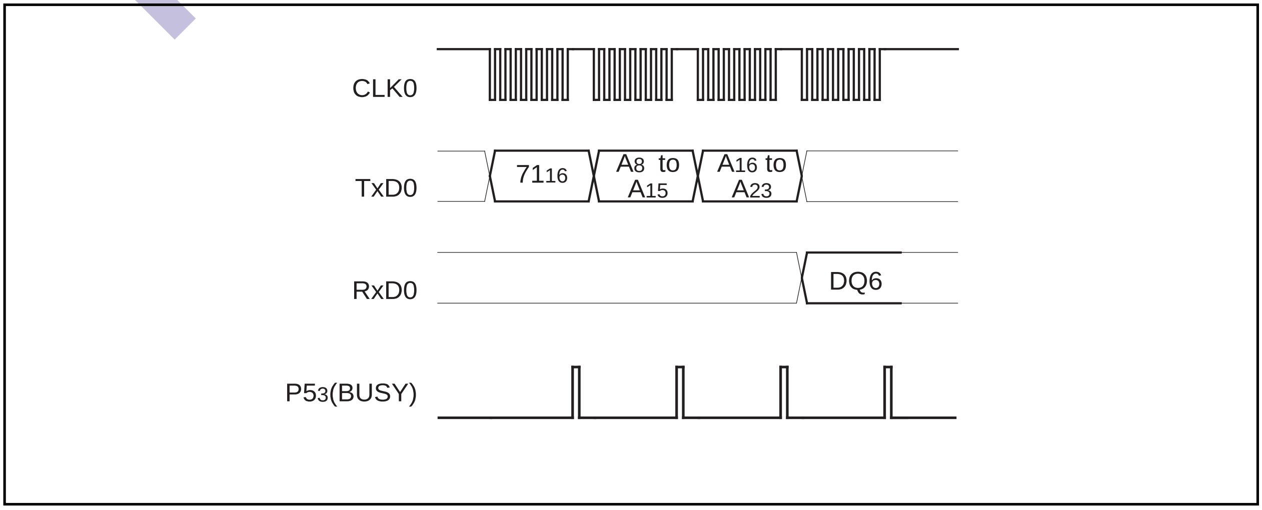


Figure DD-10. Timing for reading lock bit status

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Send the “FA_{16}” command code in the 1st byte of the transmission.
(2) Send the program size in the 2nd and 3rd bytes of the transmission.
(3) Send the check sum in the 4th byte of the transmission. The check sum is added to all data sent in the 5th byte onward.
(4) The program to execute is sent in the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

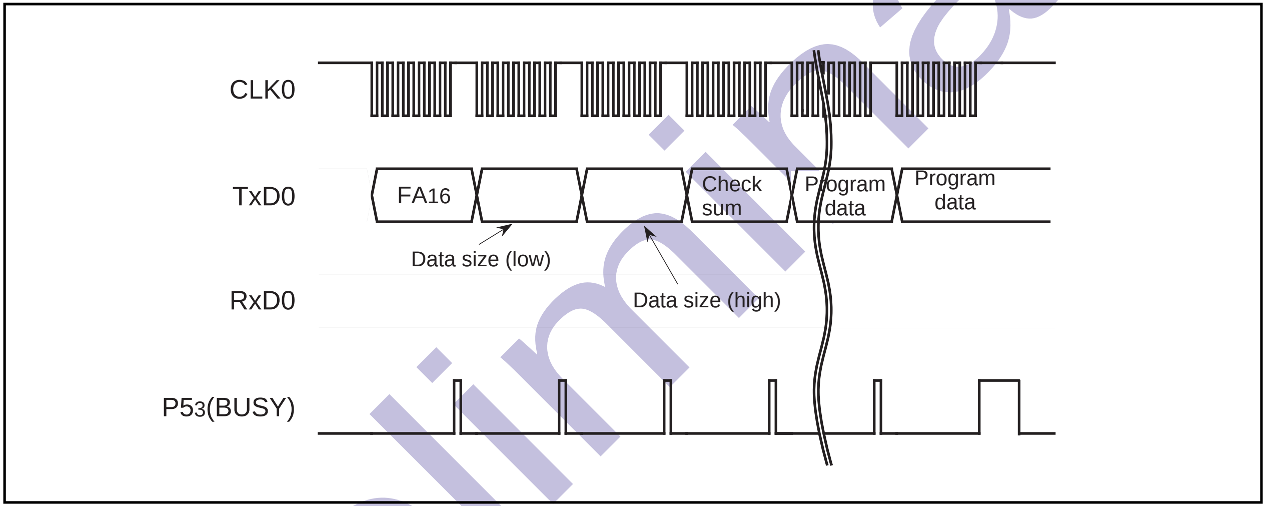


Figure DD-13. Timing for download

Preliminary

Under development

Preliminary

Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

(1) Send the "FB_{16}" command code in the 1st byte of the transmission.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

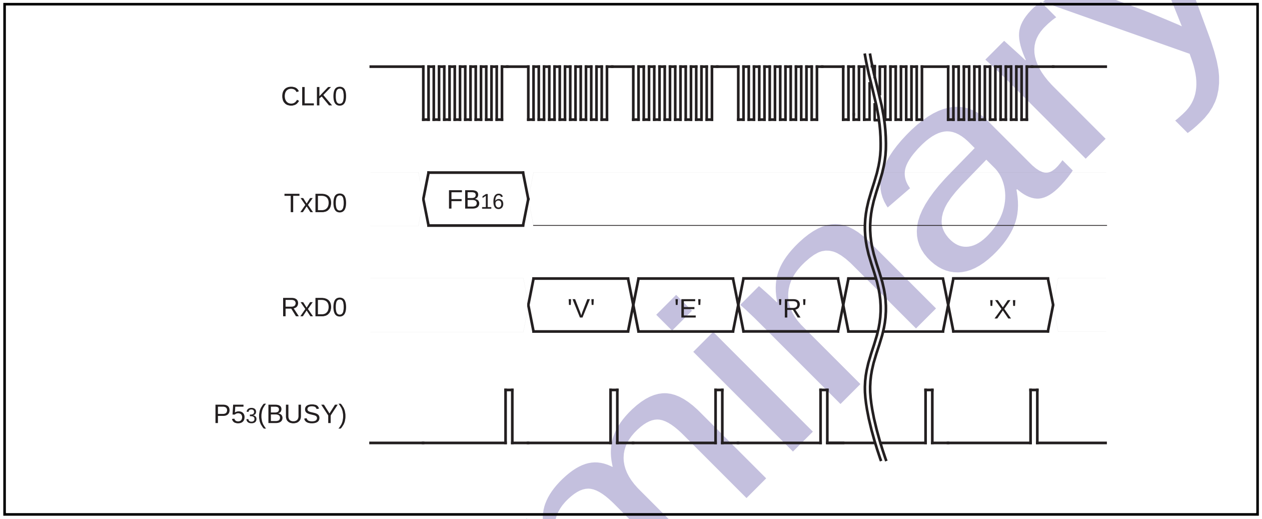


Figure DD-14. Timing for version information output

Boot Area Output Command

This command outputs the control program stored in the boot area in one page blocks (256 bytes). Execute the boot area output command as explained here following.

(1) Send the "FC_{16}" command code in the 1st byte of the transmission.

(2) Send addresses A_8 to A_{15} and A_{16} to A_{23} in the 2nd and 3rd bytes of the transmission respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first, in sync with the rise of the clock.

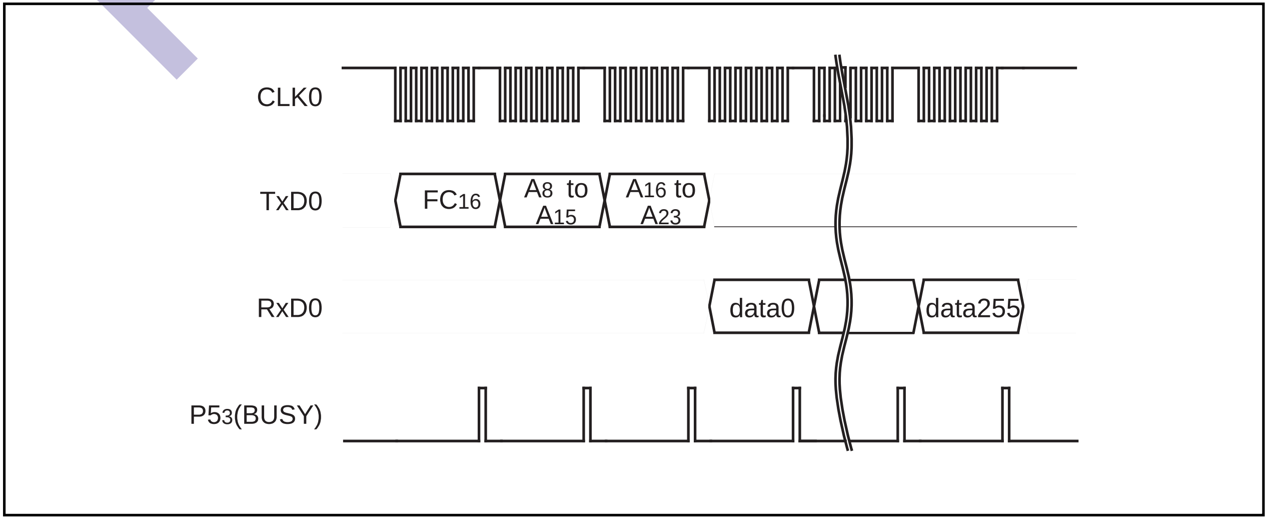


Figure DD-15. Timing for boot area output

Under development

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Send the “F5$_{16}$” command code in the 1st byte of the transmission.

(2) Send addresses A_0 to A_7, A_8 to A_{15} and A_{16} to A_{23} of the 1st byte of the ID code in the 2nd, 3rd and 4th bytes of the transmission respectively.

(3) Send the number of data sets of the ID code in the 5th byte.

(4) The ID code is sent in the 6th byte onward, starting with the 1st byte of the code.

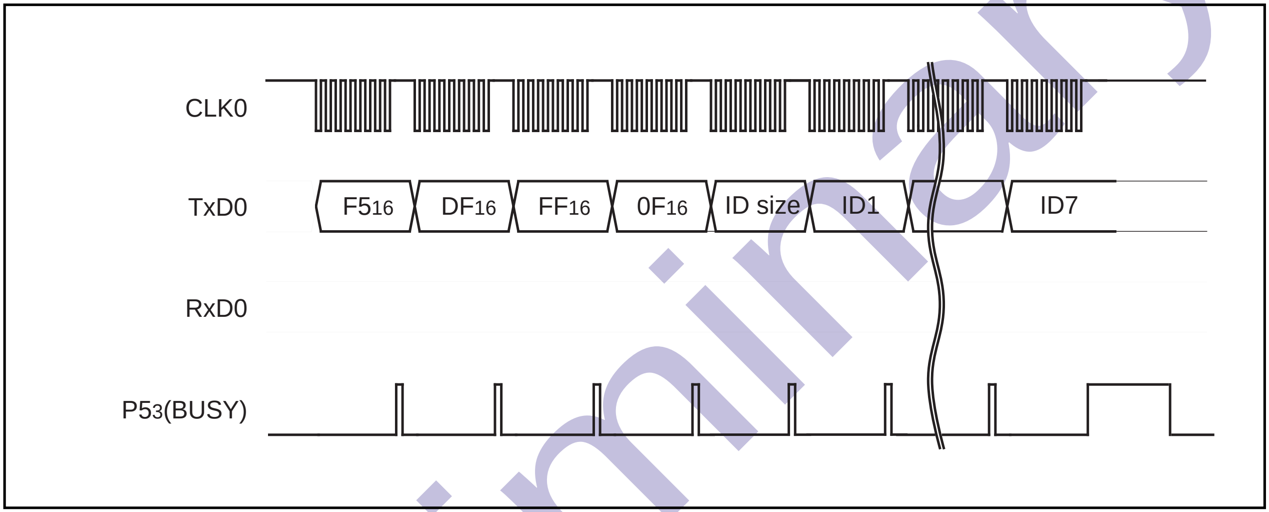
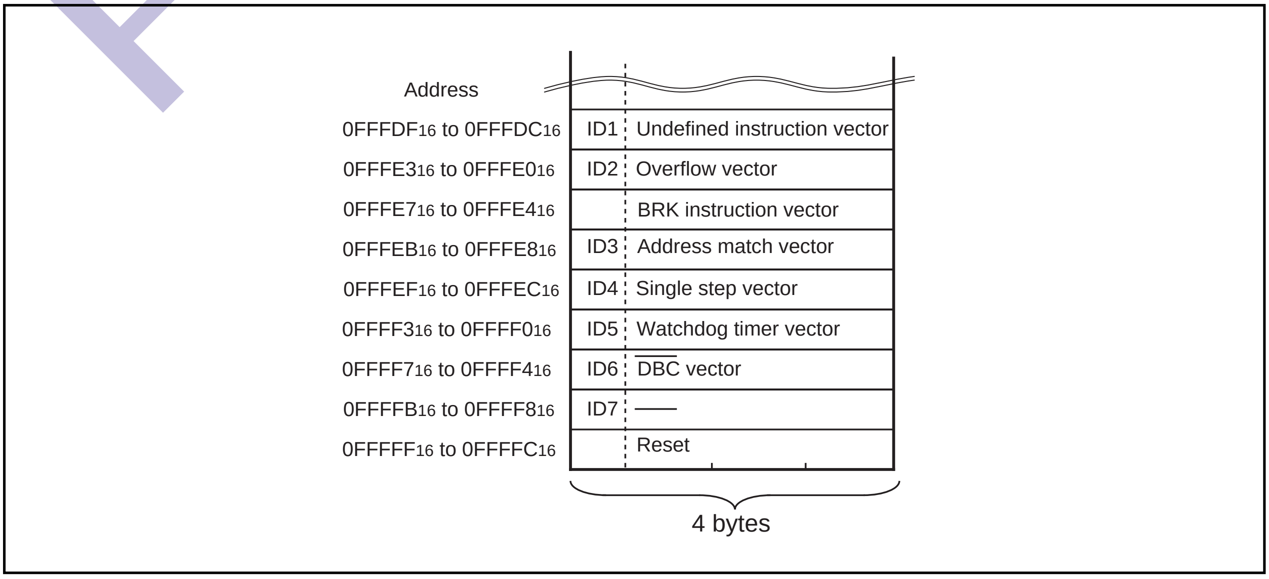


Figure DD-16. Timing for the ID check

Preliminary

ID Code

When the flash memory is not blank, the ID code sent from the serial programmer and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the serial programmer is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFDF$_{16}$, 0FFFE3$_{16}$, 0FFFEB$_{16}$, 0FFFEF$_{16}$, 0FFFF3$_{16}$, and 0FFFF7$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.



Address		
0FFFDF$_{16}$ to 0FFFDC$_{16}$	ID1	Undefined instruction vector
0FFFE3$_{16}$ to 0FFFE0$_{16}$	ID2	Overflow vector
0FFFE7$_{16}$ to 0FFFE4$_{16}$		BRK instruction vector
0FFFEB$_{16}$ to 0FFFE8$_{16}$	ID3	Address match vector
0FFFEF$_{16}$ to 0FFFEC$_{16}$	ID4	Single step vector
0FFFF3$_{16}$ to 0FFFF0$_{16}$	ID5	Watchdog timer vector
0FFFF7$_{16}$ to 0FFFF4$_{16}$	ID6	$\overline{\text{DBC}}$ vector
0FFFFB$_{16}$ to 0FFFF8$_{16}$	ID7	——
0FFFFF$_{16}$ to 0FFFFC$_{16}$		Reset

Figure DD-17. ID code storage addresses

Under development

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}). Table DD-2 gives the definition of each status register bit. After clearing the reset, the status register outputs “80_{16}”.

Table DD-2. Status register (SRD)

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Status bit	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase bit	Terminated in error	Terminated normally
SR4 (bit4)	Program bit	Terminated in error	Terminated normally
SR3 (bit3)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Preliminary

Status Bit (SR7)

The status bit indicates the operating status of the flash memory. When power is turned on, “1” (ready) is set for it. The bit is set to “0” (busy) during an auto write or auto erase operation, but it is set back to “1” when the operation ends.

Erase Bit (SR5)

The erase bit reports the operating status of the auto erase operation. If an erase error occurs, it is set to “1”. When the erase status is cleared, it is set to “0”.

Program Bit (SR4)

The program bit reports the operating status of the auto write operation. If a write error occurs, it is set to “1”. When the program status is cleared, it is set to “0”.

Under development

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).

Table DD-3 gives the definition of each status register 1 bit. "00_{16}" is output when power is turned ON and the flag status is maintained even after the reset.

Table DD-3. Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 Not verified 01 Verification mismatch 10 Reserved 11 Verified	
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

Check Sum Consistency Bit (SR12)

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

ID Check Completed Bits (SR11 and SR10)

These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Reception Time Out (SR9)

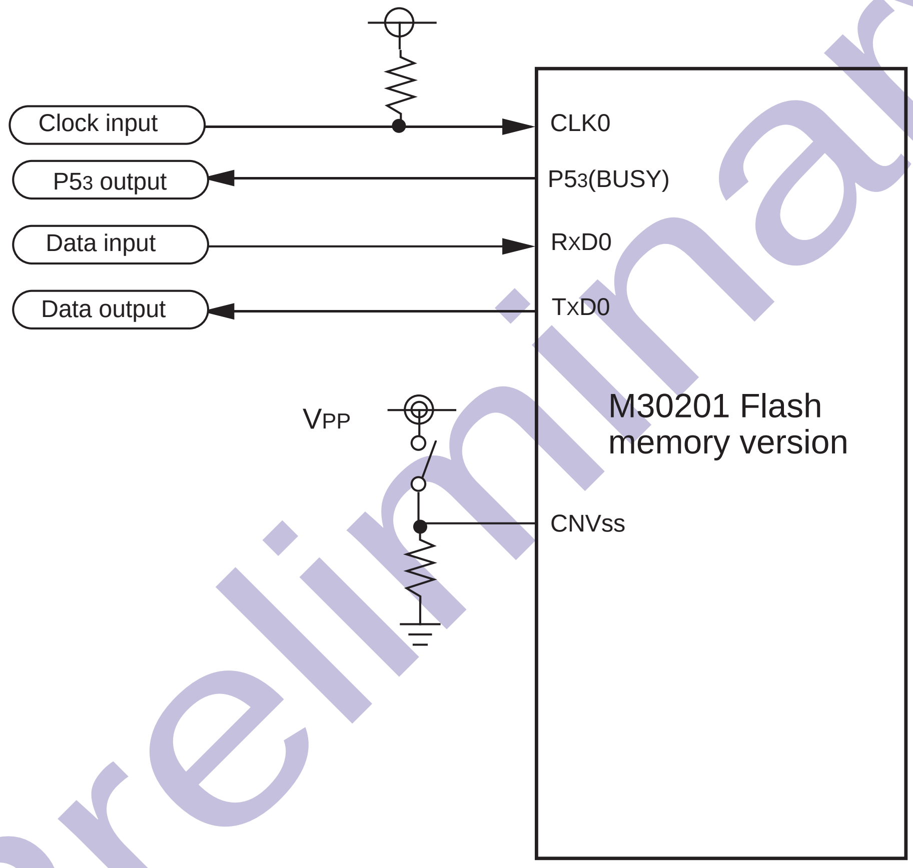
This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

Preliminary

Under development

Example Circuit Application for The Standard Serial I/O Mode

The below figure shows a circuit application for the standard serial I/O mode. Control pins will vary according to programmer, therefore see the programmer manual for more information.



Preliminary

(1) Control pins and external circuitry will vary according to programmer. For more information, see the programmer manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure DD-20. Example circuit application for the standard serial I/O mode

Chapter 2

Peripheral Functions Usage

Under development

2.1 Protect

2.1.1 Overview

'Protect' is a function that causes a value held in a register to be unchanged even when a program runs away. The following is an overview of the protect function:

(1) Registers affected by the protect function

The registers affected by the protect function are:

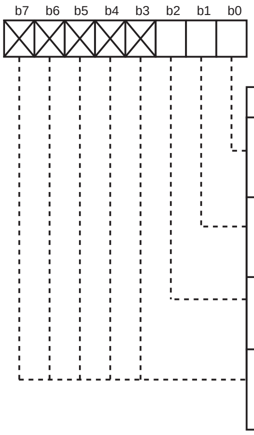
(a) System clock control registers 0, 1 (addresses 0006_{16} and 0007_{16})
(b) Processor mode registers 0, 1 (addresses 0004_{16} and 0005_{16})
(c) Port P4 direction register (address $03EA_{16}$)

The values in registers (1) through (3) cannot be changed in write-protect state. To change values in the registers, put the individual registers in write-enabled state.

(2) Protect register

Figure 2.1.1 shows protect register.

Protect register



Symbol	Address	When reset
PRCR	$000A_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P4 direction register (address $03EA_{16}$) (Note)	0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.			—	—

Note: Writing a value to an address after "1" is written to this bit returns the bit

Figure 2.1.1. Protect register

2.1.2 Protect Operation

The following explains the protect operation. Figure 2.1.2 shows the set-up procedure.

Operation
(1) Setting "1" in the write-enable bit of system clock control registers 0 and 1 causes system clock control register 0 and system clock control register 1 to be in write-enabled state.
(2) The contents of system clock control register 0 and that of system clock control register 1 are changed.
(3) Setting "0" in the write-enable bit of system control registers 0 and 1 causes system clock control register 0 and system control register 1 to be in write-inhibited state.
(4) To change the contents of processor mode register 0 and that of processor mode register 1, follow the same steps as in dealing with system clock control registers.
(5) The write-enable bit of port P4 direction register goes to "0" when the next write instruction is executed after write-enabled state is readied. Make changes in input/output immediately after the instruction that sets "1" in the write-enable bit of port P4 direction register (avoid causing an interrupt).

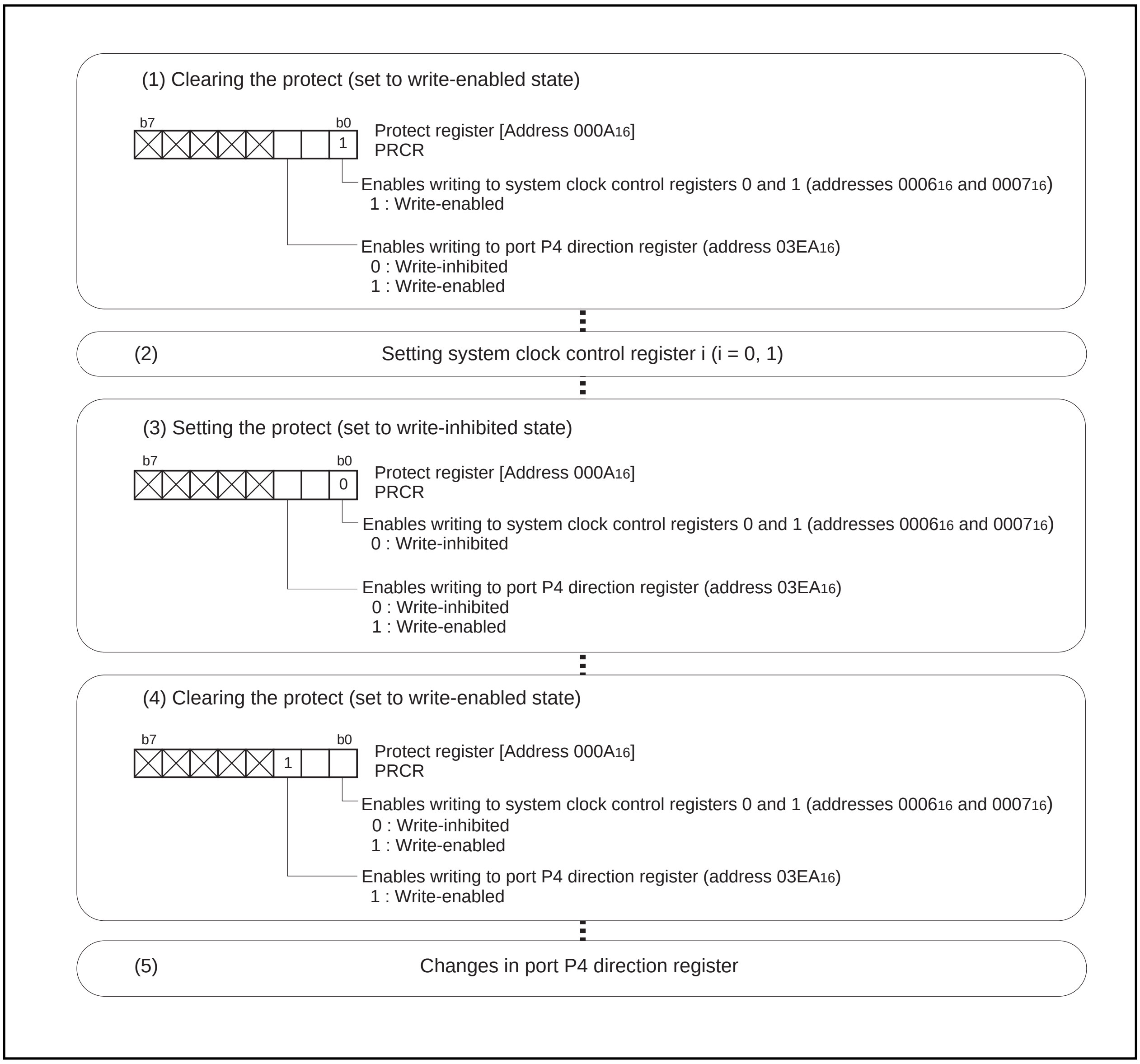

(1) Clearing the protect (set to write-enabled state)
b7
b0
1
Protect register [Address 000A$_{16}$]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 0006$_{16}$ and 0007$_{16}$)
1 : Write-enabled
Enables writing to port P4 direction register (address 03EA$_{16}$)
0 : Write-inhibited
1 : Write-enabled
(2)
Setting system clock control register i (i = 0, 1)
(3) Setting the protect (set to write-inhibited state)
b7
b0
0
Protect register [Address 000A$_{16}$]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 0006$_{16}$ and 0007$_{16}$)
0 : Write-inhibited
Enables writing to port P4 direction register (address 03EA$_{16}$)
0 : Write-inhibited
1 : Write-enabled
(4) Clearing the protect (set to write-enabled state)
b7
b0
1
Protect register [Address 000A$_{16}$]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 0006$_{16}$ and 0007$_{16}$)
0 : Write-inhibited
1 : Write-enabled
Enables writing to port P4 direction register (address 03EA$_{16}$)
1 : Write-enabled
(5)
Changes in port P4 direction register

Figure 2.1.2. Set-up procedure for protect function

2.1.3 Precaution for Protect

(1) The write-enable bit of port P4 direction register goes to “0” when the next write instruction is executed after write-enabled state is readied. Make changes in input/output immediately after the instruction that sets “1” in the write-enable bit of port P4 direction register (avoid causing an interrupt).

Under development

2.2 Timer A

2.2.1 Overview

The following is an overview for timer A, a 16-bit timer.

(1) Mode

Timer A operates in one of the four modes:

(a) Timer mode

In this mode, the internal count source is counted. Two functions can be selected: the pulse output function that reverses output from a port every time an overflow occurs, or the gate function which controls the count start/stop according to the input signal from a port.

- Timer mode operation ... P168
- Timer mode, gate function operation ... P170
- Timer mode, pulse output function operation ... P172

(b) Event counter mode

This mode counts the pulses from the outside and the number of overflows in other timers. The free-run type, in which nothing is reloaded from the reload register, can be selected when an underflow occurs. The pulse output function can also be selected. Please refer to the timer mode explanation for details, as the operation is identical.

- Event counter mode operation ... P174
- Event counter mode, free run type operation ... P176

Furthermore, Timer A has a 2-phase pulse signal processing function which generates an up count or down count in the event counter mode, depending on the phase of the two input signals.

- Operation of the 2-phase pulse signal processing function in normal event counter mode ... P178
- Operation of the 2-phase pulse signal processing function in 4-multiplication mode ... P180

(c) One-shot timer mode

In this mode, the timer is started by the trigger and stops when the timer goes to “0”. The trigger can be selected from the following 3 types: an external input signal, an overflow of the timer, or a software trigger. The pulse output function can also be selected. Please refer to the timer mode explanation for details, as the operation is identical.

- Operation in one-shot timer mode effected by software ... P182
- Operation in one-shot timer mode effected by an external trigger ... P184

(d) Pulse width modulation (PWM) mode

In this mode, the arbitrary pulses are successively output. Either a 16-bit fixed-period PWM mode or 8-bit variable-period mode can be selected. The trigger for initiating output can also be selected. Please refer to the one-shot timer mode explanation for details, as the operation is identical.

- 16-bit PWM mode operation ... P186
- 8-bit PWM mode operation ... P188

(2) Count source

The internal count source can be selected from f1, f_8, f_{32}, and f_{C32}. Clocks f_1, f_8, and f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively. Clock f_{C32} is derived by dividing the CPU's secondary clock by 32.

(3) Frequency division ratio

In timer mode or pulse width modulation mode, [the value set in the timer register + 1] becomes the frequency division ratio. In event counter mode, [the set value + 1] becomes the frequency division ratio when a down count is performed, or [$FFFF_{16}$ - the set value + 1] becomes the frequency division ratio when an up count is performed. In one-shot timer mode, the value set in the timer register becomes the frequency division ratio.

The counter overflows (or underflows) when a count source equal to a frequency division ratio is input, and an interrupt occurs. For the pulse output function, the output from the port varies (the value in the port register does not vary).

(4) Reading the timer

Either in timer mode or in event counter mode, reading the timer register takes out the count at that moment. Read it in 16-bit units. The data either in one-shot timer mode or in pulse width modulation mode is indeterminate.

(5) Writing to the timer

To write to the timer register when a count is in progress, the value is written only to the reload register. When writing to the timer register when a count is stopped, the value is written both to the reload register and to the counter. Write a value in 16-bit units.

(6) Relation between the input/output to/from the timer and the direction register

With the output function of the timer, set the direction register of the relevant port to input. To input an external signal to the timer, set the direction register of the relevant port to input.

(7) Pins related to timer A

(a) $TA0_{IN}$	Input pins to timer A.
(b) $TA0_{OUT}$	Output pins from timer A. They become input pins to timer A when event counter mode is active.

Under development

(8) Registers related to timer A

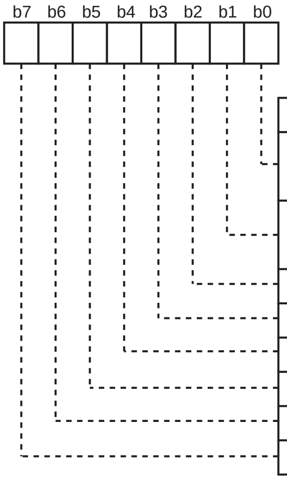
Figure 2.2.1 shows the memory map of timer A-related registers. Figures 2.2.2 through 2.2.5 show timer A-related registers.

0055_{16}	Timer A0 interrupt control register (TA0IC)
≈	≈
0380_{16}	Count start flag (TABSR)
0381_{16}	Clock prescaler reset flag (CPSRF)
0382_{16}	One-shot start flag (ONSF)
0383_{16}	Trigger select register (TRGSR)
0384_{16}	Up-down flag (UDF)
0385_{16}	
0386_{16}	Timer A0 (TA0)
0387_{16}	
≈	≈
0396_{16}	Timer A0 mode register (TA0MR)

Figure 2.2.1. Memory map of timer A-related registers

Timer A0 mode register

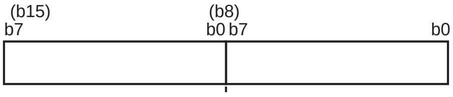
Symbol	Address	When reset
TA0MR	0396_{16}	00_{16}



Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 2.2.2. Timer A-related registers (1)

Timer A0 register (Note)



Symbol TA0

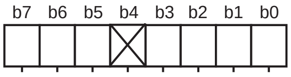
Address 0387_{16},0386_{16}

When reset Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$	×	○
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000_{16} to $FFFE_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00_{16} to FF_{16} (High-order addresses) 00_{16} to FE_{16} (Low-order addresses)	×	○

Note: Read and write data in 16-bit units.

Count start flag



Symbol TABSR

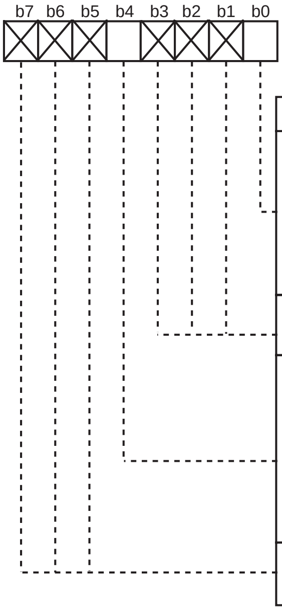
Address 0380_{16}

When reset $000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Figure 2.2.3. Timer A-related registers (2)

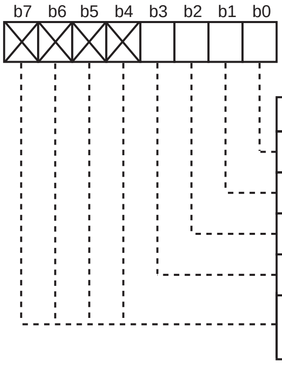
Up/down flag



Symbol	Address	When reset
UDF	0384_{16}	$XXX0XXX0_2$

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TA0P	Timer A0 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to “0”	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

One-shot start flag



Symbol	Address	When reset
ONSF	0382_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TX0OS	Timer X0 one-shot start flag		○	○
TX1OS	Timer X1 one-shot start flag		○	○
TX2OS	Timer X2 one-shot start flag		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Figure 2.2.4. Timer A-related registers (3)

Trigger select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TRGSR	0383$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0$_{IN}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX2 overflow is selected 1 1 : TX0 overflow is selected	○	○
TA0TGH			○	○
TX0TGL	Timer X0 event/trigger select bit	b3 b2 0 0 : Input on TX0$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TX1 overflow is selected	○	○
TX0TGH			○	○
TX1TGL	Timer X1 event/trigger select bit	b5 b4 0 0 : Input on TX1$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX0 overflow is selected 1 1 : TX2 overflow is selected	○	○
TX1TGH			○	○
TX2TGL	Timer X2 event/trigger select bit	b7 b6 0 0 : Input on TX2$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX1 overflow is selected 1 1 : TA0 overflow is selected	○	○
TX2TGH			○	○

Note: Set the corresponding port direction register to “0”(input mode).

Clock prescaler reset flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPSRF	0381$_{16}$	0XXXXXXX$_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 2.2.5. Timer A-related registers (4)

2.2.2 Operation of Timer A (timer mode)

In timer mode, choose functions from those listed in Table 2.2.1. Operations of the circled items are described below. Figure 2.2.6 shows the operation timing, and Figure 2.2.7 shows the set-up procedure.

Table 2.2.1. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Pulse output function	O	No pulses output
		Pulses output
Gate function	O	No gate function
		Performs count only for the period in which the $TA0_{IN}$ pin is at "L" level
		Performs count only for the period in which the $TA0_{IN}$ pin is at "H" level

Operation (1) Setting the count start flag to “1” causes the counter to perform a down count on the count source.

(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

(3) Setting the count start flag to “0” causes the counter to hold its value and to stop.

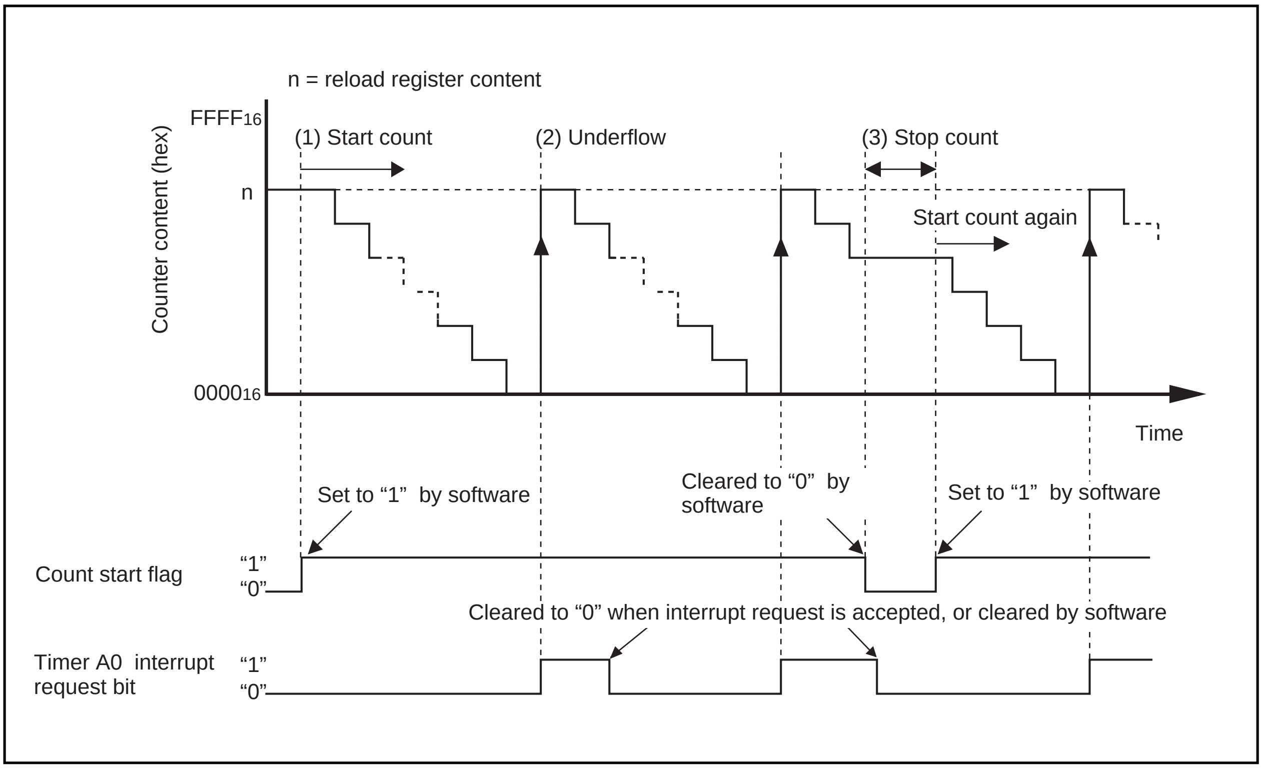
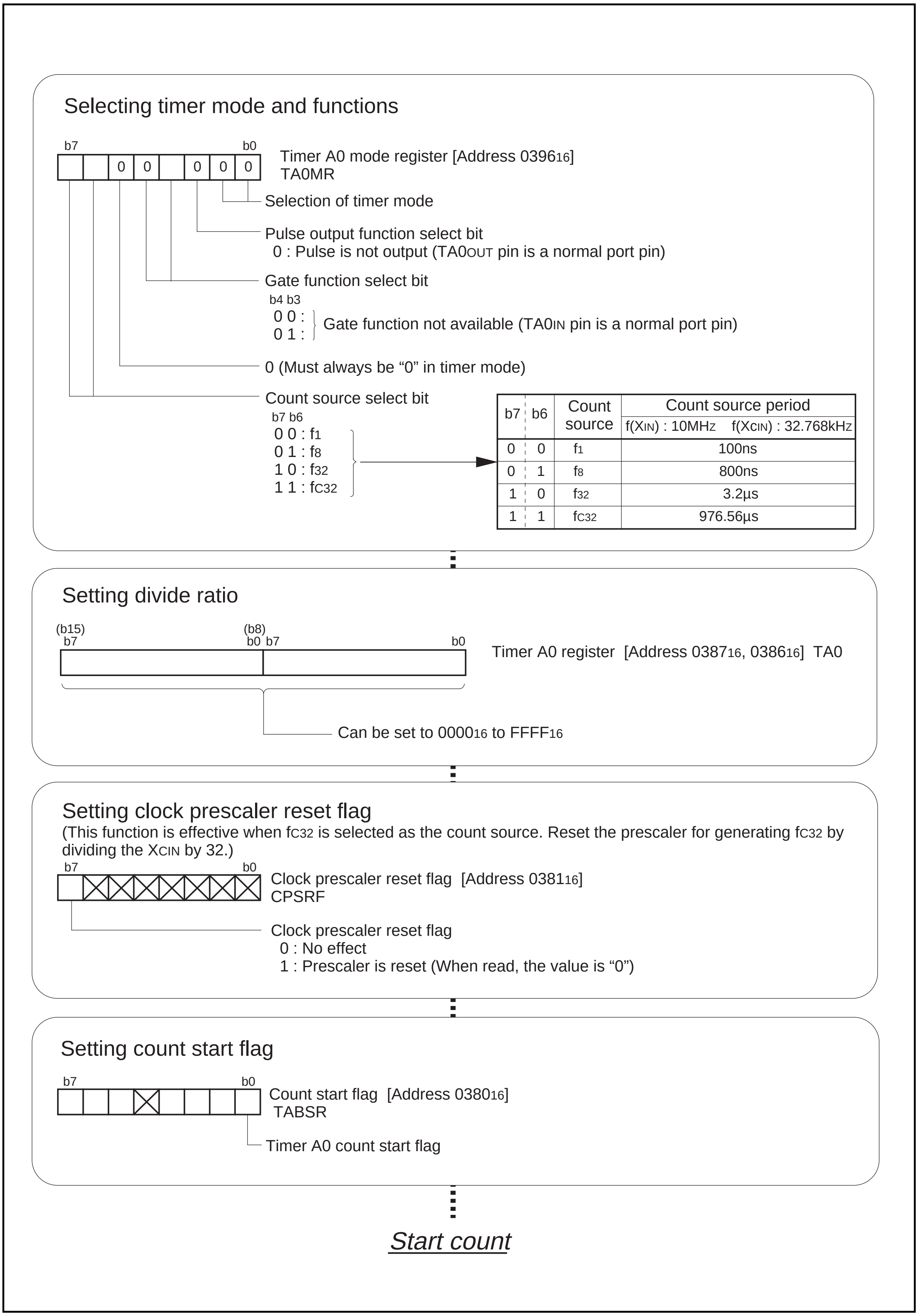


Figure 2.2.6. Operation timing of timer mode

Under development


Selecting timer mode and functions
b7
b0
0 0 0 0 0
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Gate function select bit
b4 b3
0 0 :
0 1 :
Gate function not available (TA0IN pin is a normal port pin)
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
Setting divide ratio
(b15)
b7
(b8)
b0 b7
b0
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 0000₁₆ to FFFF₁₆
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7
b0
Clock prescaler reset flag [Address 0381₁₆]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

b7	b6	Count source	Count source period f(XIN) : 10MHz	Count source period f(XCIN) : 32.768kHz
0	0	f1	100ns	
0	1	f8	800ns	
1	0	f32	3.2µs	
1	1	fC32		976.56µs

Figure 2.2.7. Set-up procedure of timer mode

Under development

2.2.3 Operation of Timer A (timer mode, gate function selected)

In timer mode, choose functions from those listed in Table 2.2.2. Operations of the circled items are described below. Figure 2.2.8 shows the operation timing, and Figure 2.2.9 shows the set-up procedure.

Table 2.2.2. Choosed functions

Item	Set-up	
Count source	O	Internal count source(f_1 / f_8 / f_{32} / fc_{32})
Pulse output function	O	No pulses output
		Pulses output
Gate function		No gate function
		Performs count only for the period in which the $TA0_{IN}$ pin is at "L" level
	O	Performs count only for the period in which the $TA0_{IN}$ pin is at "H" level

Operation (1) When the count start flag is set to "1" and the $TA0_{IN}$ pin inputs at "H" level, the counter performs a down count on the count source.

(2) When the $TA0_{IN}$ pin inputs at "L" level, the counter holds its value and stops.

(3) If an underflow occurs, the content of the reload register is reloaded and the count continues. At this time, the timer A0 interrupt request bit goes to "1".

(4) Setting the count start flag to "0" causes the counter to hold its value and to stop.

Note • Make the pulse width of the signal input to the $TA0_{IN}$ pin not less than two cycles of the count source.

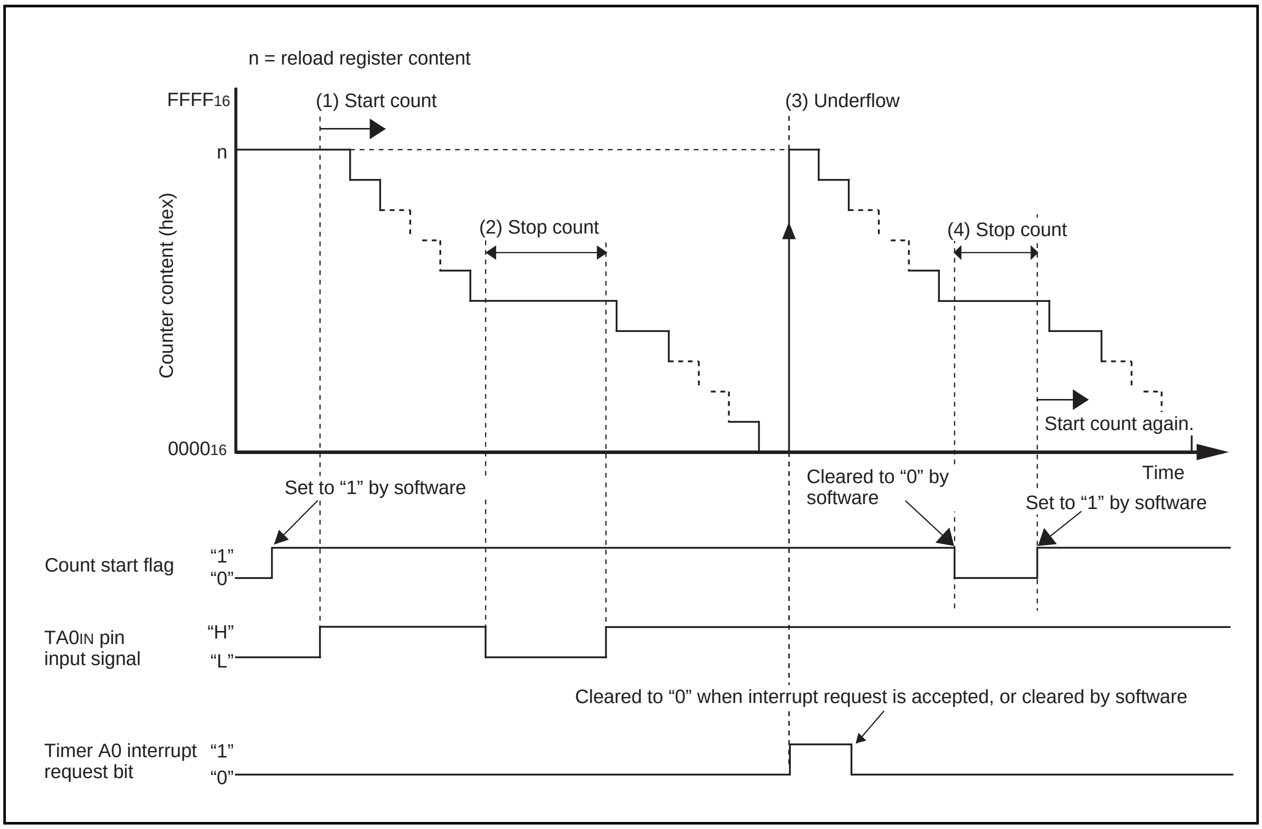
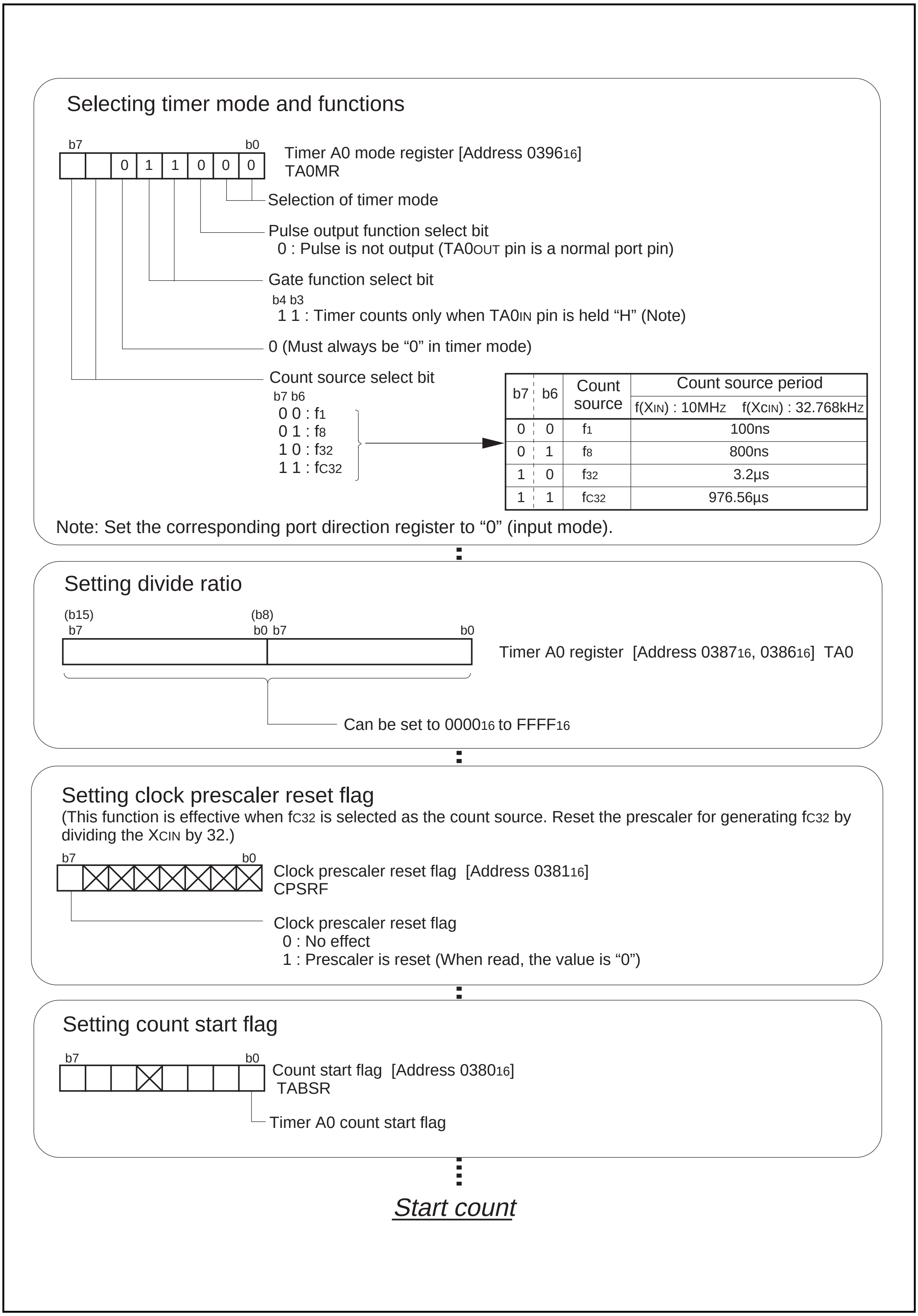


Figure 2.2.8. Operation timing of timer mode, gate function selected


Selecting timer mode and functions
b7
b0
0 1 1 0 0 0
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of timer mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Gate function select bit
b4 b3
1 1 : Timer counts only when TA0IN pin is held “H” (Note)
0 (Must always be “0” in timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
Note: Set the corresponding port direction register to “0” (input mode).
Setting divide ratio
(b15)
b7
(b8)
b0 b7
b0
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 0000₁₆ to FFFF₁₆
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7
b0
Clock prescaler reset flag [Address 0381₁₆]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f1	100ns
0	1	f8	800ns
1	0	f32	3.2µs
1	1	fC32	976.56µs

Figure 2.2.9. Set-up procedure of timer mode, gate function selected

2.2.4 Operation of Timer A (timer mode, pulse output function selected)

In timer mode, choose functions from those listed in Table 2.2.3. Operations of the circled items are described below. Figure 2.2.10 shows the operation timing, and Figure 2.2.11 shows the set-up procedure.

Table 2.2.3. Choosed functions

Item	Set-up	
Count source	O	Internal count source(f_1 / f_8 / f_{32} / fc_{32})
Pulse output function		No pulses output
	O	Pulses output
Gate function	O	No gate function
		Performs count only for the period in which the TA0_{IN} pin is at "L" level
		Performs count only for the period in which the TA0_{IN} pin is at "H" level

Operation (1) Setting the count start flag to "1" causes the counter to perform a down count on the count source.

(2) If an underflow occurs, the content of the reload register is reloaded and the count continues. At this time, the timer A0 interrupt request bit goes to "1". Also, the output polarity of the TA0_{OUT} pin reverses.

(3) Setting the count start flag to "0" causes the counter to hold its value and to stop. Also, the TA0_{OUT} pin outputs an "L" level.

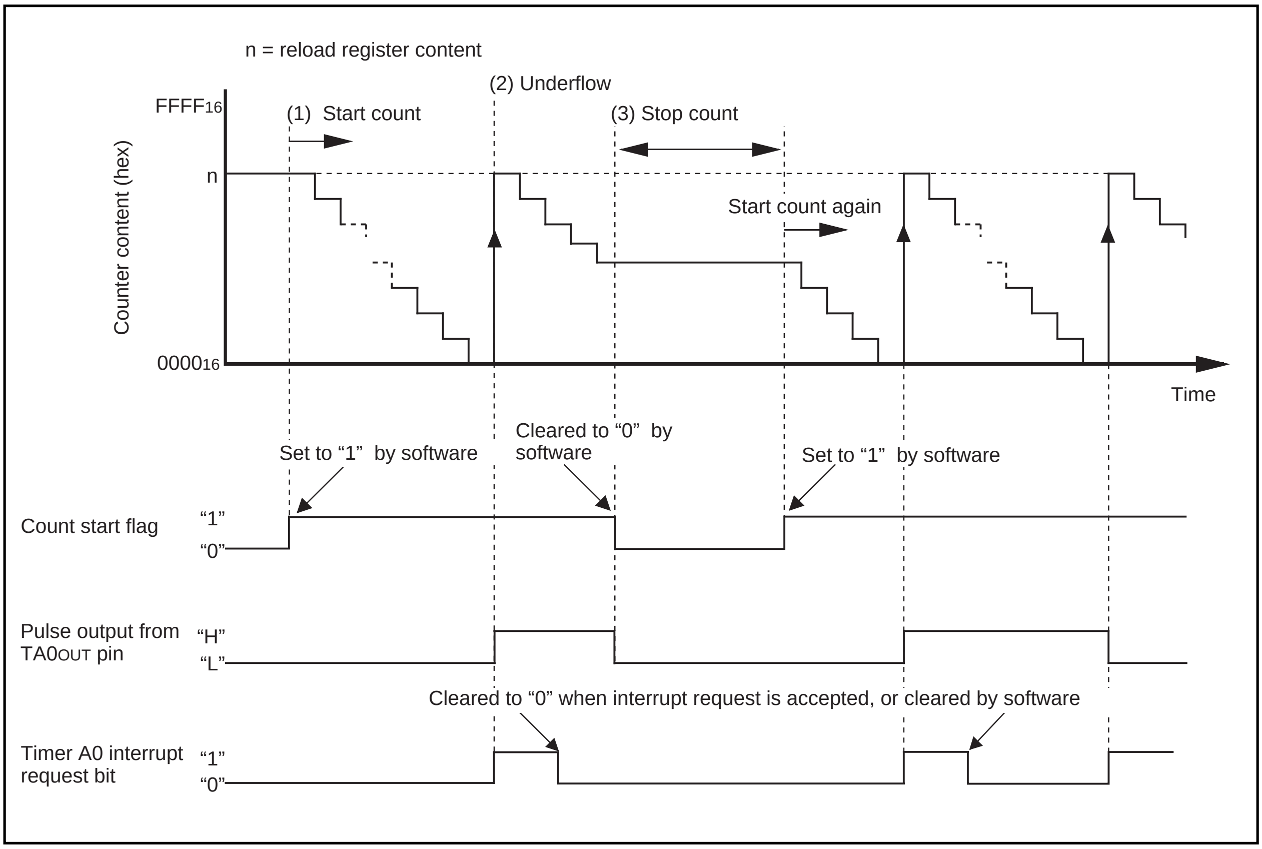
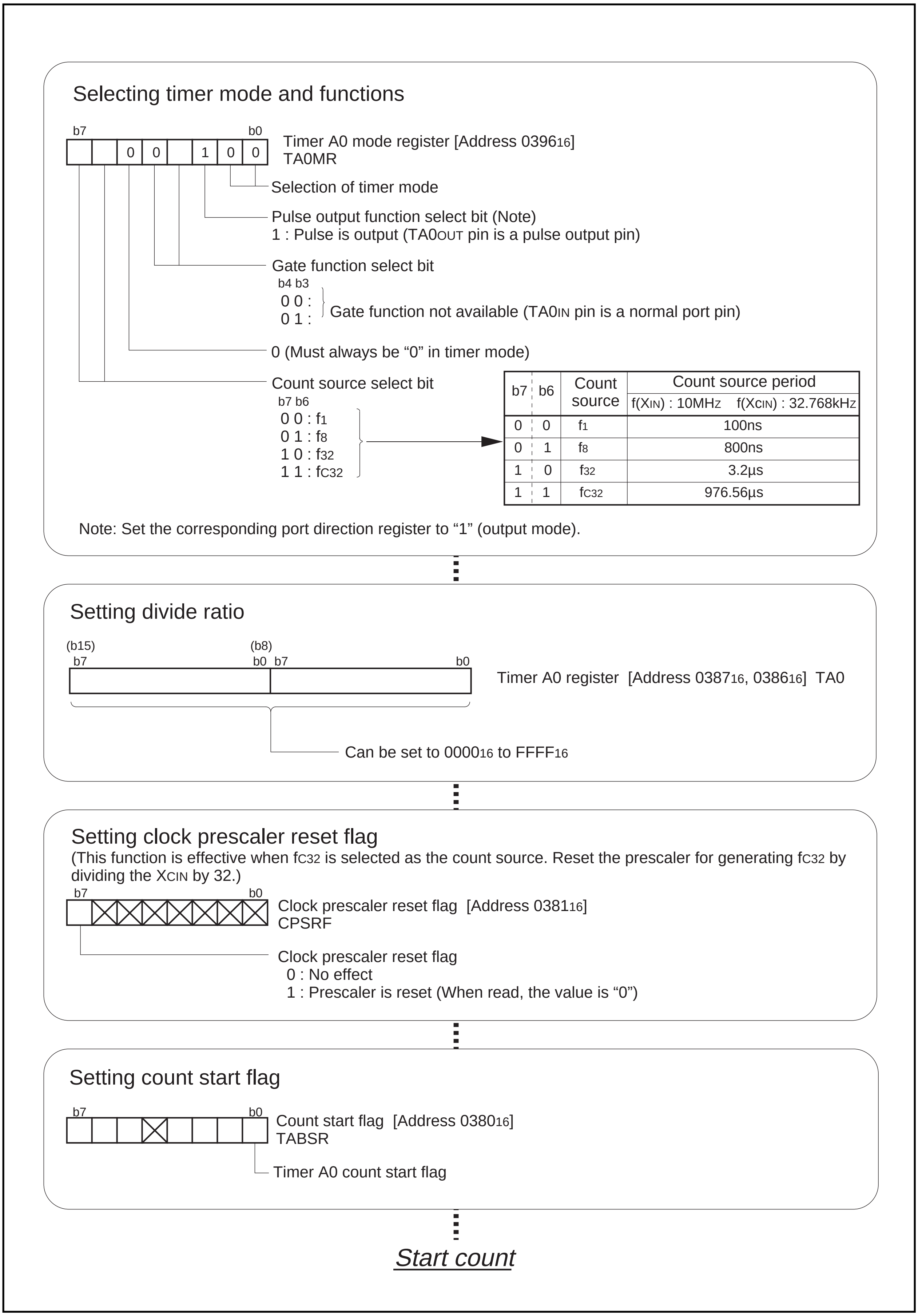

n = reload register content
(2) Underflow
FFFF16
(1) Start count
(3) Stop count
Counter content (hex)
n
Start count again
000016
Time
Cleared to "0" by software
Set to "1" by software
Set to "1" by software
Count start flag
"1"
"0"
Pulse output from TA0OUT pin
"H"
"L"
Cleared to "0" when interrupt request is accepted, or cleared by software
Timer A0 interrupt request bit
"1"
"0"

Figure 2.2.10. Operation timing of timer mode, pulse output function selected


Selecting timer mode and functions
b7
b0
0 0 1 0 0
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of timer mode
Pulse output function select bit (Note)
1 : Pulse is output (TA0OUT pin is a pulse output pin)
Gate function select bit
b4 b3
0 0 :
0 1 :
Gate function not available (TA0IN pin is a normal port pin)
0 (Must always be “0” in timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
Note: Set the corresponding port direction register to “1” (output mode).
Setting divide ratio
(b15)
b7
(b8)
b0
b7
b0
Timer A0 register [Address 038716, 038616] TA0
Can be set to 000016 to FFFF16
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7
b0
Clock prescaler reset flag [Address 038116]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)
Setting count start flag
b7
b0
Count start flag [Address 038016]
TABSR
Timer A0 count start flag
Start count

b7	b6	Count source	Count source period f(XIN) : 10MHz	Count source period f(XCIN) : 32.768kHz
0	0	f1	100ns	
0	1	f8	800ns	
1	0	f32	3.2µs	
1	1	fC32		976.56µs

Figure 2.2.11. Set-up procedure of timer mode, pulse output function selected

Under development

2.2.5 Operation of Timer A (event counter mode, reload type selected)

In event counter mode, choose functions from those listed in Table 2.2.4. Operations of the circled items are described below. Figure 2.2.12 shows the operation timing, and Figure 2.2.13 shows the set-up procedure.

Table 2.2.4. Choosed functions

Item	Set-up		Item	Set-up	
Count source	O	Input signal to TA0IN (counting falling edges)	Pulse output function	O	No pulses output
					Pulses output
		Input signal to TA0IN (counting rising edges)	Count operation type	O	Reload type
					Free-run type
		Timer overflow (TB1/TX0/TX2 overflow)	Factor for switching between up and down	O	Content of up/down flag
					Input signal to TA0OUT

Operation
(1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.
(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer A0 interrupt request bit goes to “1”.
(3) If switching from an up count to a down count or vice versa while a count is in progress, the switch takes effect from the next effective edge of the count source.
(4) Setting the count start flag to “0” causes the counter to hold its value and to stop.
(5) If an overflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

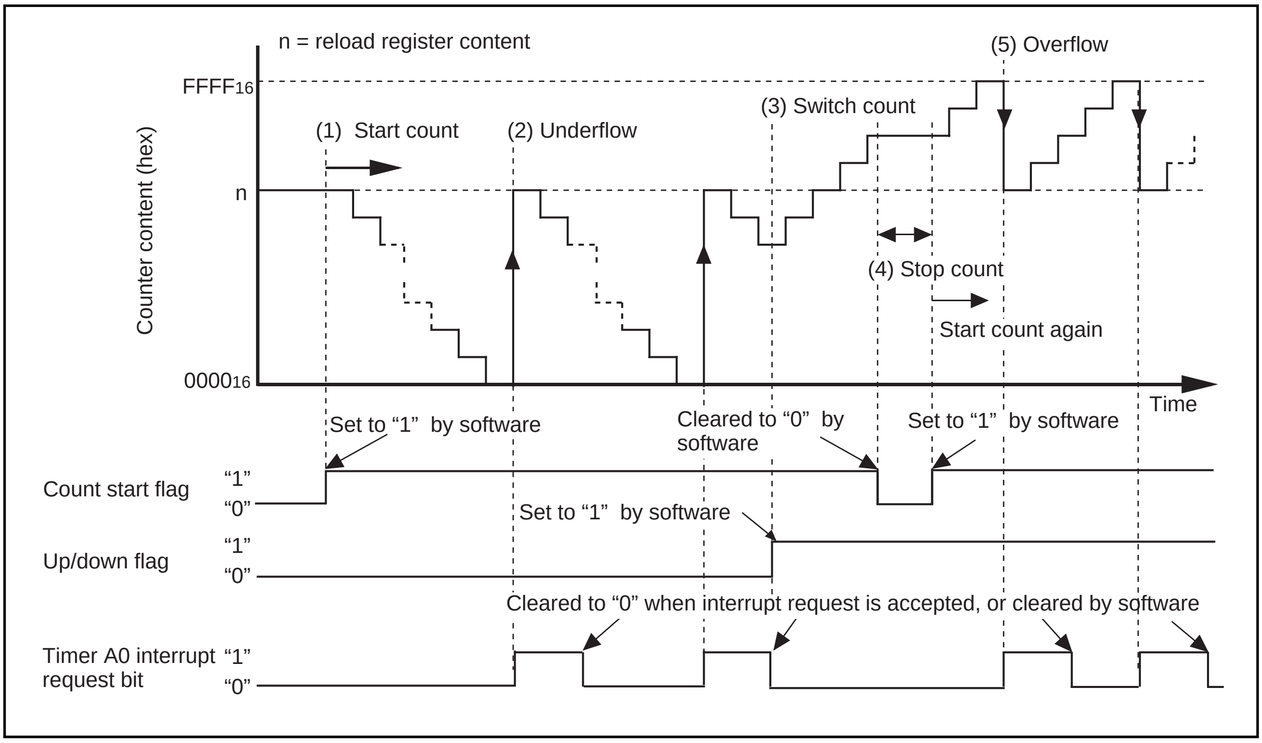


Figure 2.2.12. Operation timing of event counter mode, reload type selected

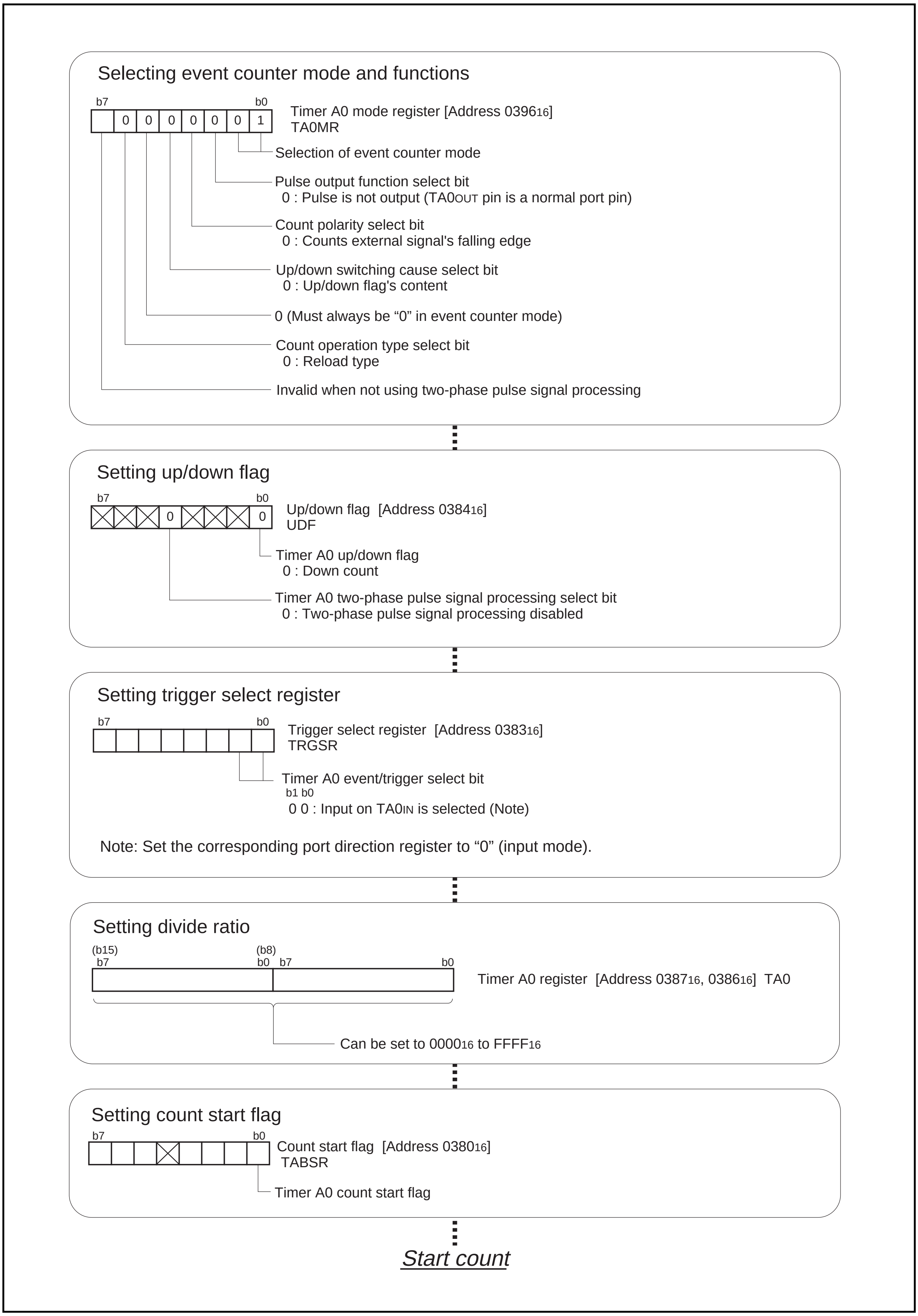

Selecting event counter mode and functions
b7
b0
0 0 0 0 0 0 1
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Count polarity select bit
0 : Counts external signal's falling edge
Up/down switching cause select bit
0 : Up/down flag's content
0 (Must always be “0” in event counter mode)
Count operation type select bit
0 : Reload type
Invalid when not using two-phase pulse signal processing
Setting up/down flag
b7
b0
0 0
Up/down flag [Address 0384₁₆]
UDF
Timer A0 up/down flag
0 : Down count
Timer A0 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
Setting trigger select register
b7
b0
Trigger select register [Address 0383₁₆]
TRGSR
Timer A0 event/trigger select bit
b1 b0
0 0 : Input on TA0IN is selected (Note)
Note: Set the corresponding port direction register to “0” (input mode).
Setting divide ratio
(b15)
b7
(b8)
b0
b7
b0
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 0000₁₆ to FFFF₁₆
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

Figure 2.2.13. Set-up procedure of event counter mode, reload type selected

2.2.6 Operation of Timer A (event counter mode, free run type selected)

In event counter mode, choose functions from those listed in Table 2.2.5. Operations of the circled items are described below. Figure 2.2.14 shows the operation timing, and Figure 2.2.15 shows the set-up procedure.

Table 2.2.5. Choosed functions

Item	Set-up		Item	Set-up	
Count source	O	Input signal to $TA0_{IN}$ (counting falling edges)	Pulse output function	O	No pulses output
					Pulses output
		Input signal to $TA0_{IN}$ (counting rising edges)	Count operation type		Reload type
				O	Free-run type
		Timer overflow (TB1/TX0/TX2 overflow)	Factor for switching between up and down	O	Content of up/down flag
					Input signal to $TA0_{OUT}$

Operation (1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.

(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

(3) If switching from an up count to a down count or vice versa while a count is in progress, the switch takes effect from the next effective edge of the count source.

(4) Even if an overflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

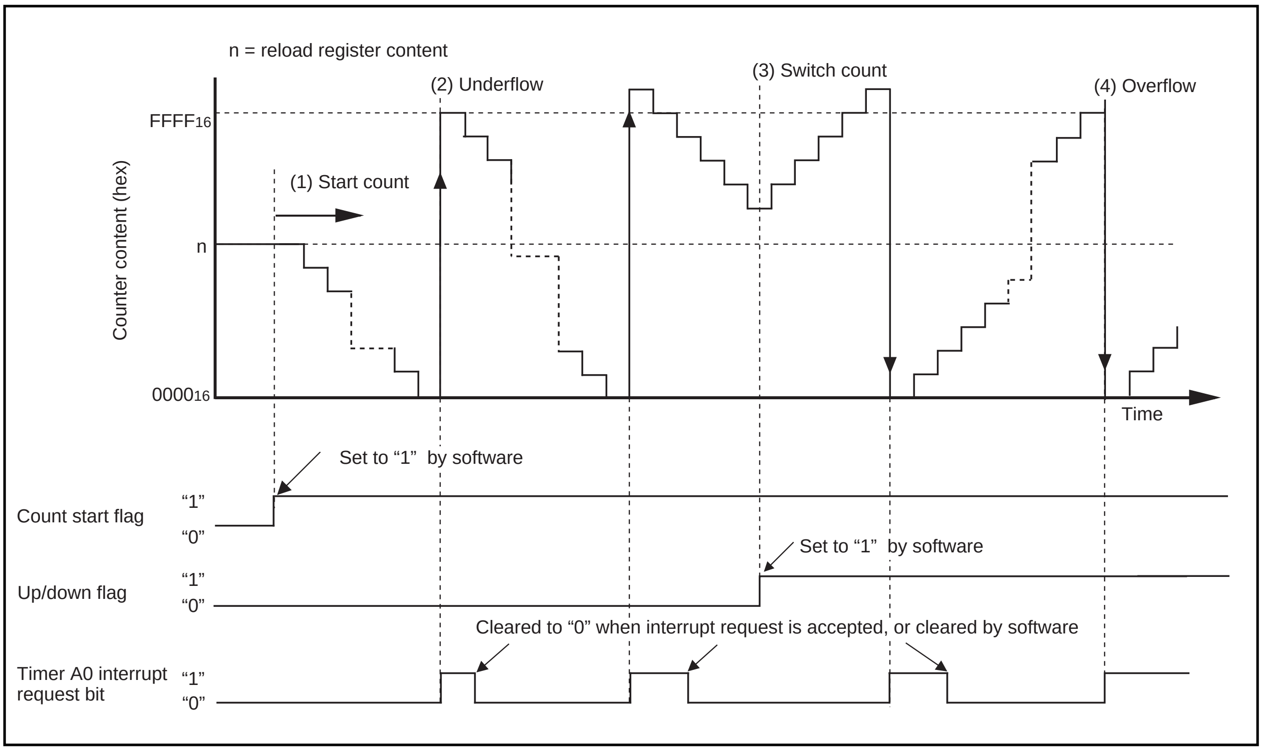


Figure 2.2.14. Operation timing of event counter mode, free run type selected

Under development

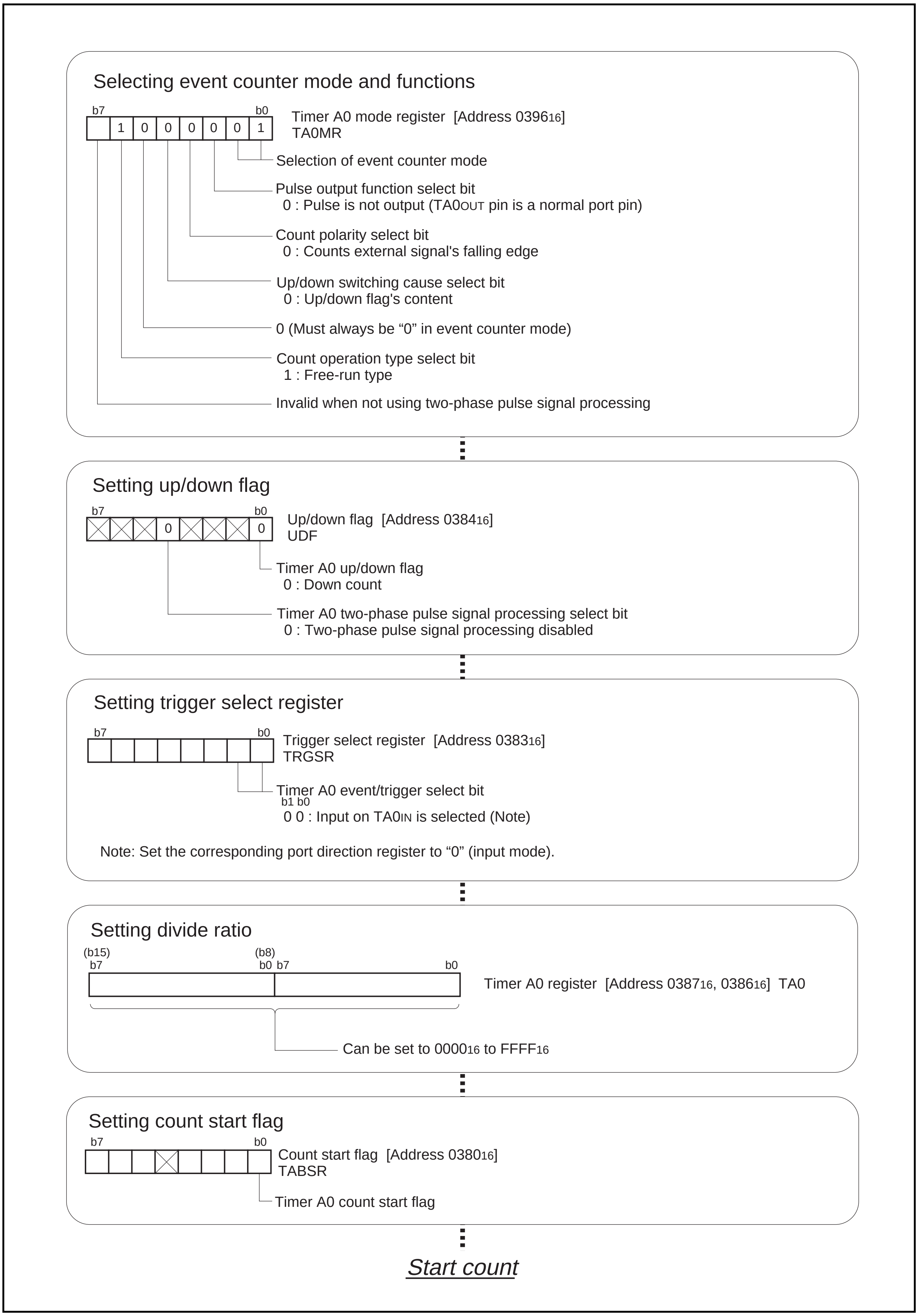

Selecting event counter mode and functions
b7
b0
1 0 0 0 0 0 1
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output (TA0OUT pin is a normal port pin)
Count polarity select bit
0 : Counts external signal's falling edge
Up/down switching cause select bit
0 : Up/down flag's content
0 (Must always be “0” in event counter mode)
Count operation type select bit
1 : Free-run type
Invalid when not using two-phase pulse signal processing
Setting up/down flag
b7
b0
0 0
Up/down flag [Address 0384₁₆]
UDF
Timer A0 up/down flag
0 : Down count
Timer A0 two-phase pulse signal processing select bit
0 : Two-phase pulse signal processing disabled
Setting trigger select register
b7
b0
Trigger select register [Address 0383₁₆]
TRGSR
Timer A0 event/trigger select bit
b1 b0
0 0 : Input on TA0IN is selected (Note)
Note: Set the corresponding port direction register to “0” (input mode).
Setting divide ratio
(b15)
b7
(b8)
b0 b7
b0
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 0000₁₆ to FFFF₁₆
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

Figure 2.2.15. Set-up procedure of event counter mode, free run type selected

Under development

2.2.7 Operation of timer A (2-phase pulse signal process in event counter mode, normal mode selected)

In processing 2-phase pulse signals in event counter mode, choose functions from those listed in Table 2.2.6. Operations of the circled items are described below. Figure 2.2.16 shows the operation timing, and Figure 2.2.17 shows the set-up procedure.

Table 2.2.6. Choosed functions

Item	Set-up	
Count operation type		Reload type
	O	Free run type
2-phase pulses process	O	Normal processing
		4-multiplication processing

Operation (1) Setting the count start flag to “1” causes the counter to count effective edges of the count source.

(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

(3) Even if an overflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

Note • The up count or down count conditions are as follows:

If a rising edge is present at the TA0IN pin when the input signal level to the TA0OUT pin is “H”, an up count is performed.

If a falling edge is present at the TA0IN pin when the input signal level to the TA0OUT pin is “H”, a down count is performed.

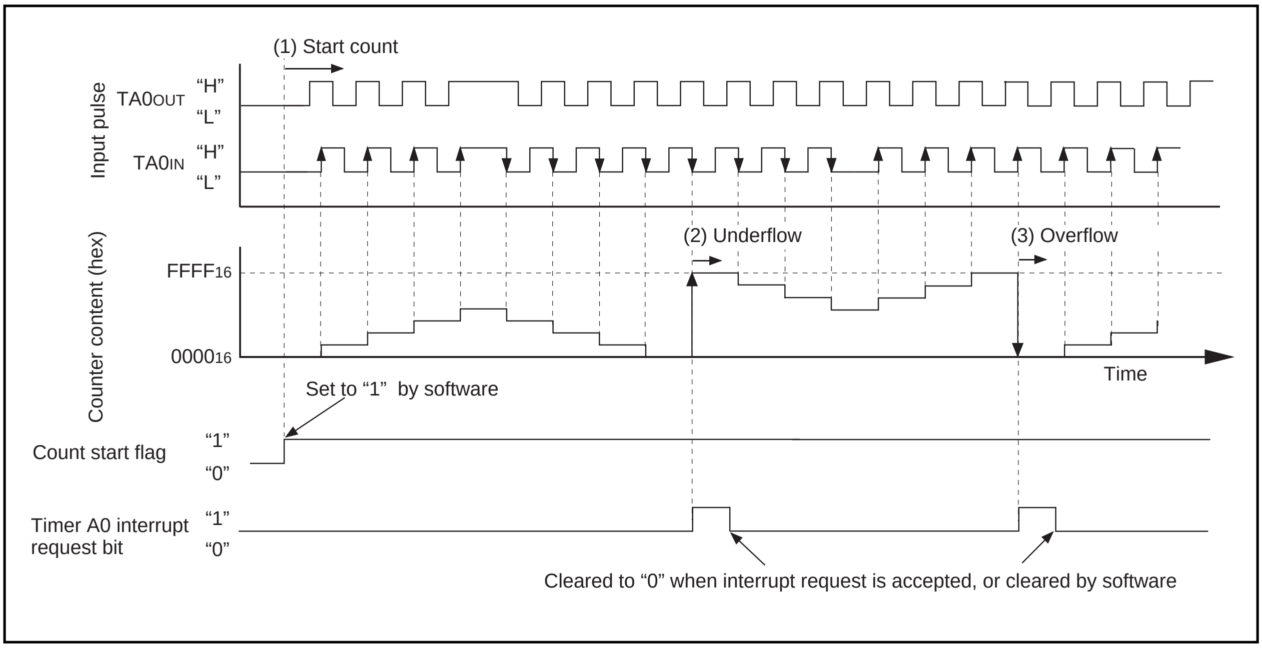
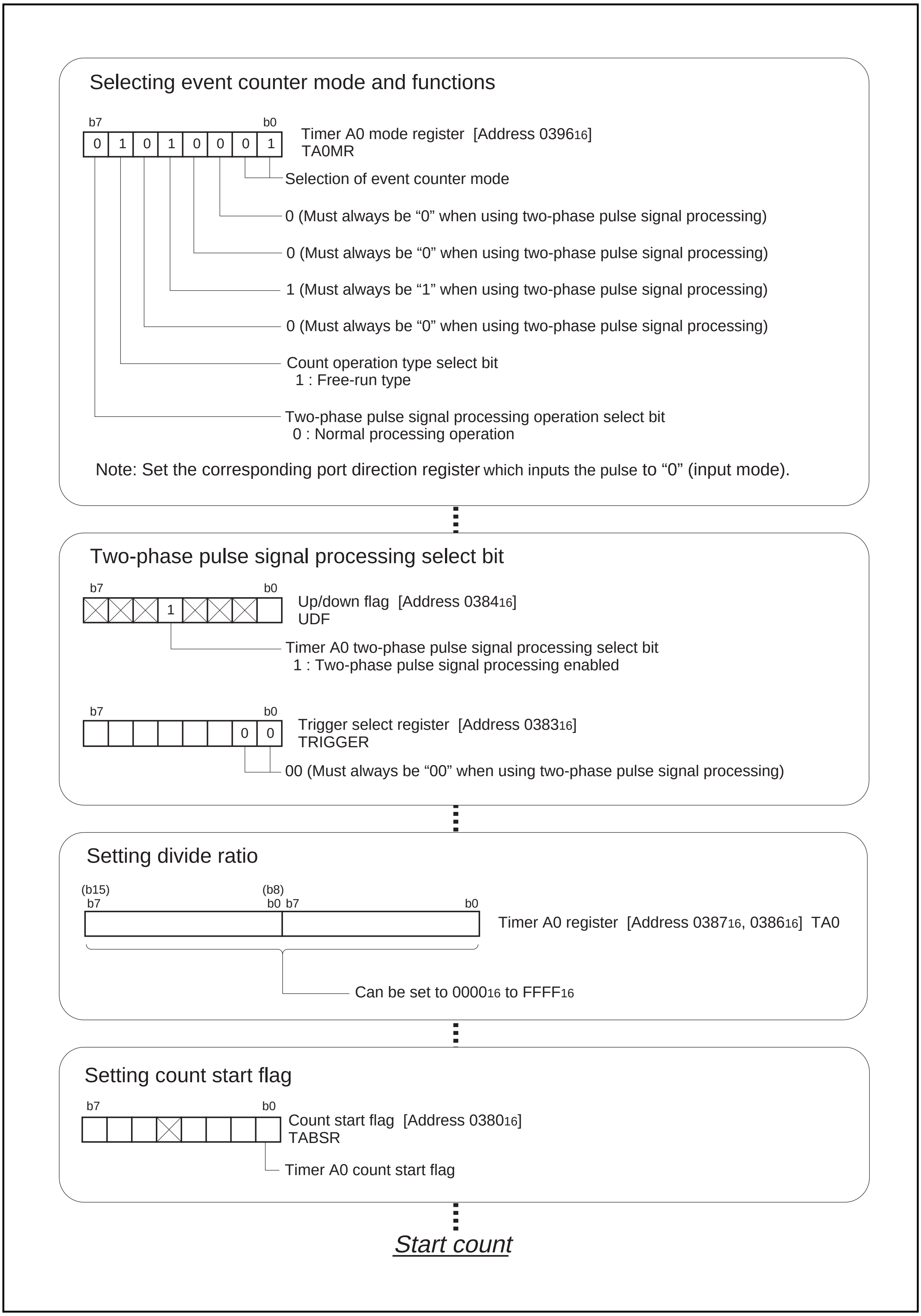


Figure 2.2.16. Operation timing of 2-phase pulse signal process in event counter mode, normal mode selected

Under development


Selecting event counter mode and functions
b7
b0
0 1 0 1 0 0 0 1
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of event counter mode
0 (Must always be “0” when using two-phase pulse signal processing)
0 (Must always be “0” when using two-phase pulse signal processing)
1 (Must always be “1” when using two-phase pulse signal processing)
0 (Must always be “0” when using two-phase pulse signal processing)
Count operation type select bit
1 : Free-run type
Two-phase pulse signal processing operation select bit
0 : Normal processing operation
Two-phase pulse signal processing select bit
b7
b0
1
Up/down flag [Address 0384₁₆]
UDF
Timer A0 two-phase pulse signal processing select bit
1 : Two-phase pulse signal processing enabled
b7
b0
0 0
Trigger select register [Address 0383₁₆]
TRIGGER
00 (Must always be “00” when using two-phase pulse signal processing)
Setting divide ratio
(b15)
b7
(b8)
b0 b7
b0
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 0000₁₆ to FFFF₁₆
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

Note: Set the corresponding port direction register which inputs the pulse to “0” (input mode).

Figure 2.2.17. Set-up procedure of 2-phase pulse signal process in event counter mode, normal mode selected

2.2.8 Operation of timer A (2-phase pulse signal process in event counter mode, multiply-by-4 mode selected)

In processing 2-phase pulse signals in event counter mode, choose functions from those listed in Table 2.2.7. Operations of the circled items are described below. Figure 2.2.18 shows the operation timing, and Figure 2.2.19 shows the set-up procedure.

Table 2.2.7. Choosed functions

Item	Set-up		Item	Set-up	
Count operation type		Reload type	Processing 2 phase pulses		Normal processing
	O	Free run type		O	4-multiplication processing

Operation (1) Setting the count start flag to “1” causes the counter to count effective edges of the count source.

(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

(3) Even if an overflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer A0 interrupt request bit goes to “1”.

Note • The up count or down count conditions are as follows:

Table 2.2.8. The up count or down count conditions

	Input signal to the TA0OUT pin	Input signal to the TA0IN pin		Input signal to the TA0OUT pin	Input signal to the TA0IN pin
Up count	“H” level	Rising	Down count	“H” level	Falling
	“L” level	Falling		“L” level	Rising
	Rising	“L” level		Rising	“H” level
	Falling	“H” level		Falling	“L” level

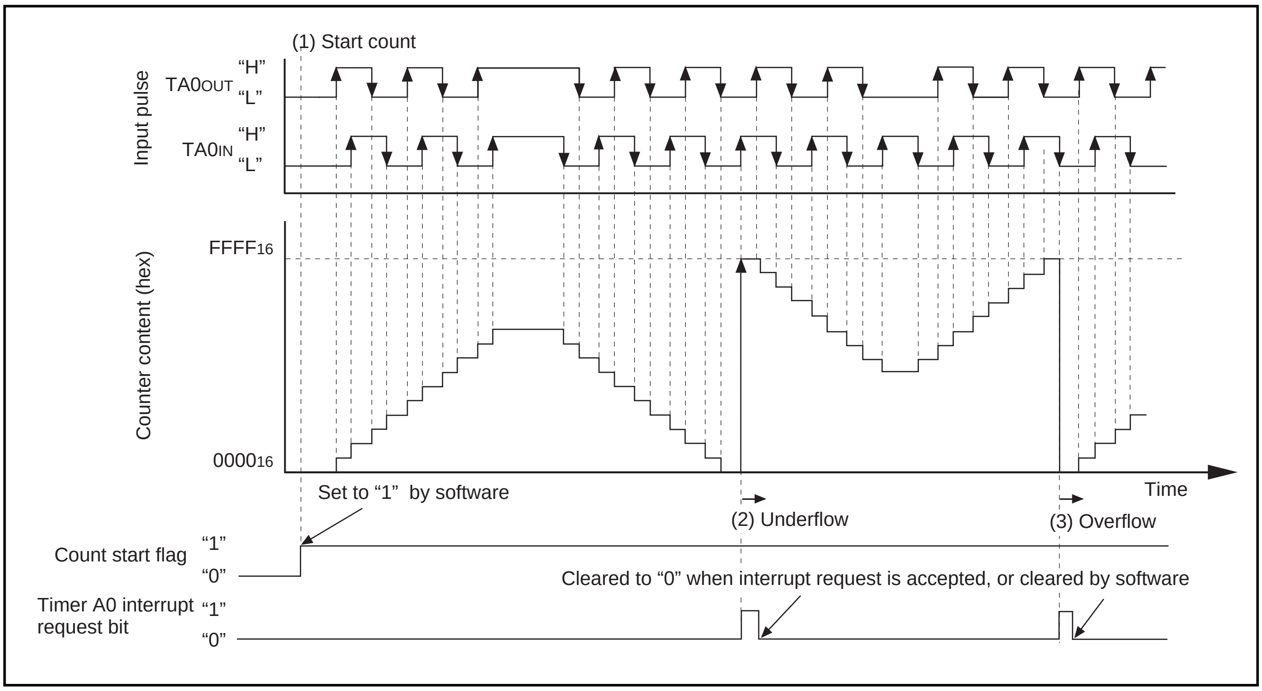
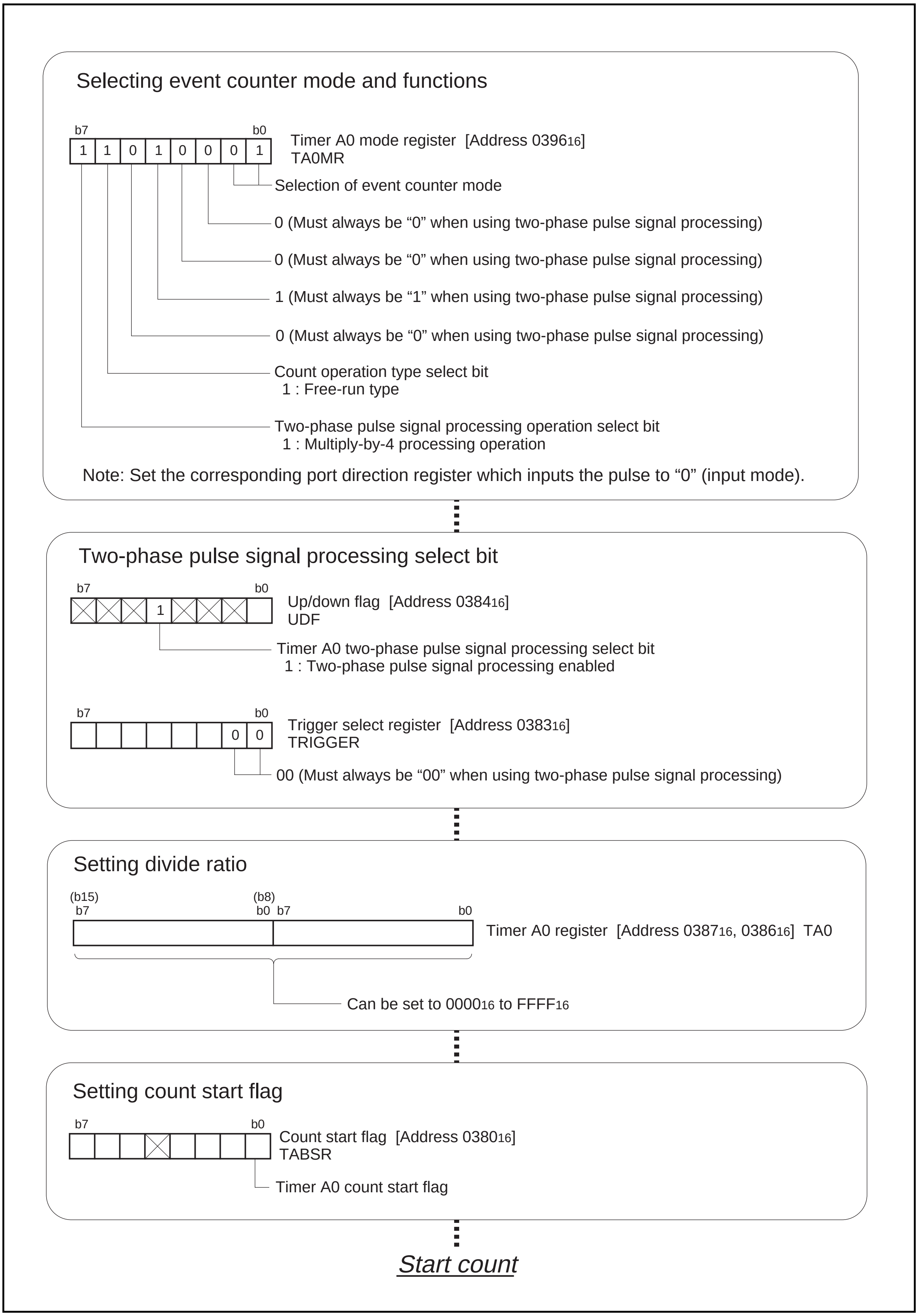


Figure 2.2.18. Operation timing of 2-phase pulse signal process in event counter mode, multiply-by-4 mode selected


Selecting event counter mode and functions
b7
b0
1 1 0 1 0 0 0 1
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of event counter mode
0 (Must always be "0" when using two-phase pulse signal processing)
0 (Must always be "0" when using two-phase pulse signal processing)
1 (Must always be "1" when using two-phase pulse signal processing)
0 (Must always be "0" when using two-phase pulse signal processing)
Count operation type select bit
1 : Free-run type
Two-phase pulse signal processing operation select bit
1 : Multiply-by-4 processing operation
Two-phase pulse signal processing select bit
b7
b0
1
Up/down flag [Address 0384₁₆]
UDF
Timer A0 two-phase pulse signal processing select bit
1 : Two-phase pulse signal processing enabled
b7
b0
0 0
Trigger select register [Address 0383₁₆]
TRIGGER
00 (Must always be "00" when using two-phase pulse signal processing)
Setting divide ratio
(b15)
b7
(b8)
b0
b7
b0
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 0000₁₆ to FFFF₁₆
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

Note: Set the corresponding port direction register which inputs the pulse to "0" (input mode).

Figure 2.2.19. Set-up procedure of2-phase pulse signal process in event counter mode, multiply-by-4 mode selected

2.2.9 Operation of Timer A (one-shot timer mode)

In one-shot timer mode, choose functions from those listed in Table 2.2.9. Operations of the circled items are described below. Figure 2.2.20 shows the operation timing, and Figure 2.2.21 shows the set-up procedure.

Table 2.2.9. Choosed functions

Item		Set-up
Count source	**O**	Internal count source (f_1 / f_8 / f_{32} / f_{C32})
Pulse output function		No pulses output
	O	Pulses output
Count start condition		External trigger input (falling edge of input signal to the $\mathrm{TA0_{IN}}$ pin)
		External trigger input (rising edge of input signal to the $\mathrm{TA0_{IN}}$ pin)
		Timer overflow (TB1/TX0/TX2 overflow)
	O	Writing “1” to the one-shot start flag

Operation (1) Setting the one-shot start flag to “1” with the count start flag set to “1” causes the counter to perform a down count on the count source. At this time, the $\mathrm{TA0_{OUT}}$ pin outputs an “H” level.

(2) The instant the value of the counter becomes “0000_{16}”, the $\mathrm{TA0_{OUT}}$ pin outputs an “L” level, and the counter reloads the content of the reload register and stops counting. At this time, the timer A0 interrupt request bit goes to “1”.

(3) If a trigger occurs while a count is in progress, the counter reloads the value in the reload register again and continues counting. The reload timing is in step with the next count source input after the trigger.

(4) Setting the count start flag to “0” causes the counter to stop and to reload the content of the reload register. Also, the $\mathrm{TA0_{OUT}}$ pin outputs an “L” level. At this time, the timer A0 interrupt request bit goes to “1”.

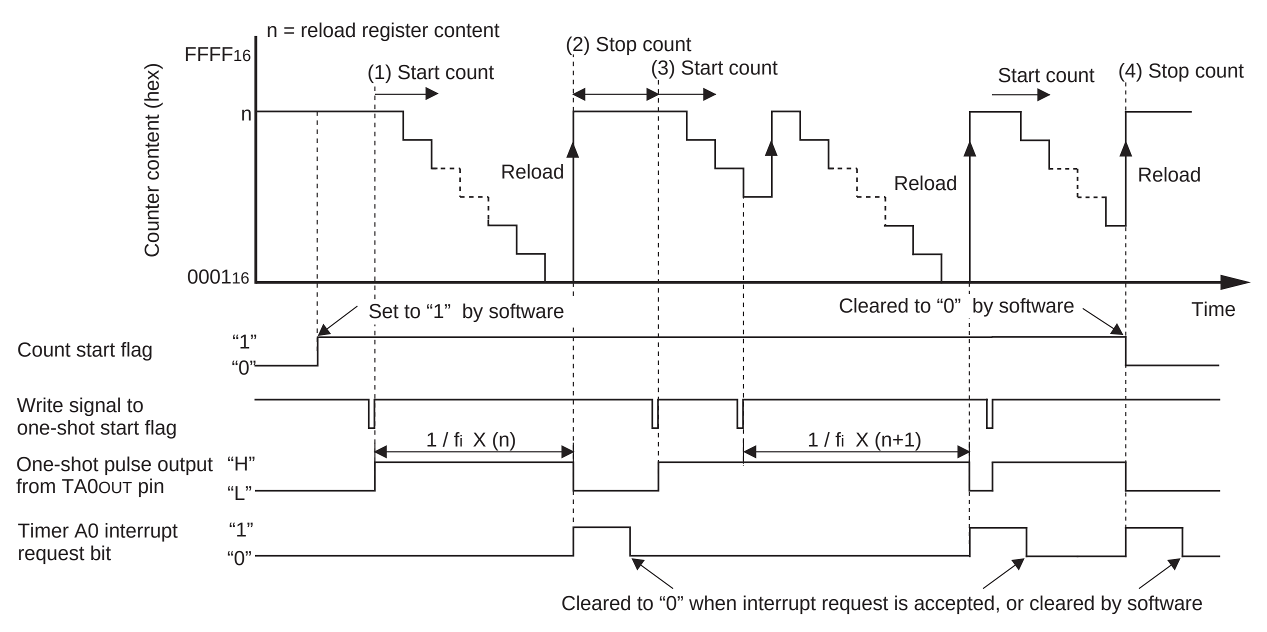
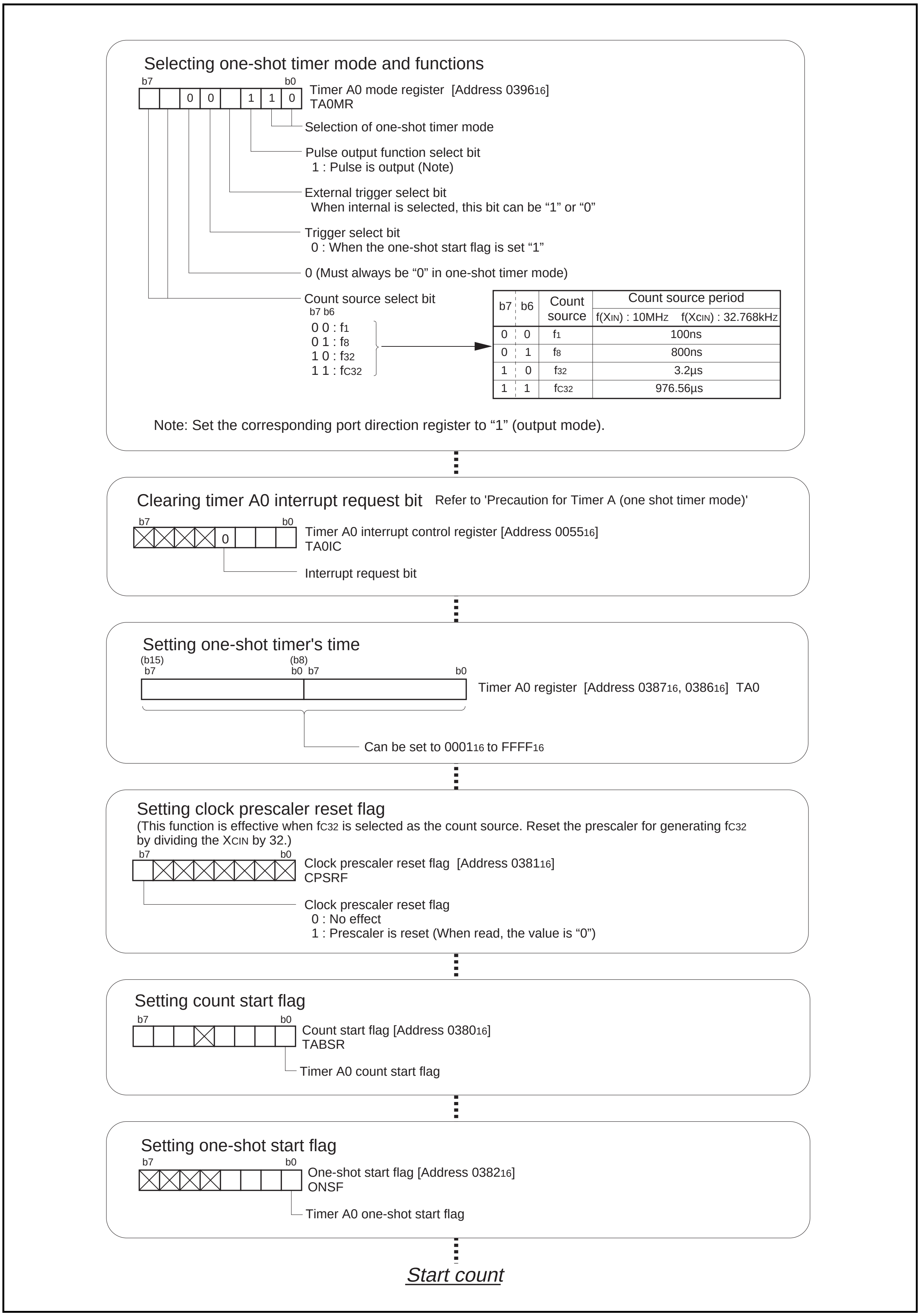

n = reload register content
FFFF16
n
000116
Counter content (hex)
(1) Start count
(2) Stop count
(3) Start count
Start count
(4) Stop count
Reload
Reload
Reload
Time
Set to “1” by software
Cleared to “0” by software
Count start flag
“1”
“0”
Write signal to one-shot start flag
1 / fi X (n)
1 / fi X (n+1)
One-shot pulse output from TA0OUT pin
“H”
“L”
Timer A0 interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software

Figure 2.2.20. Operation timing of one-shot mode



Selecting one-shot timer mode and functions

Timer A0 mode register [Address 0396_{16}]
TA0MR

Selection of one-shot timer mode

Pulse output function select bit
1 : Pulse is output (Note)

External trigger select bit
When internal is selected, this bit can be "1" or "0"

Trigger select bit
0 : When the one-shot start flag is set "1"

0 (Must always be "0" in one-shot timer mode)

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period	
			$f(X_{IN})$: 10MHz	$f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Note: Set the corresponding port direction register to "1" (output mode).

Clearing timer A0 interrupt request bit Refer to 'Precaution for Timer A (one shot timer mode)'

Timer A0 interrupt control register [Address 0055_{16}]
TA0IC

Interrupt request bit

Setting one-shot timer's time

Timer A0 register [Address 0387_{16}, 0386_{16}] TA0

Can be set to 0001_{16} to $FFFF_{16}$

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")

Setting count start flag

Count start flag [Address 0380_{16}]
TABSR

Timer A0 count start flag

Setting one-shot start flag

One-shot start flag [Address 0382_{16}]
ONSF

Timer A0 one-shot start flag

Start count

Figure 2.2.21. Set-up procedure of one-shot mode

Under development

2.2.10 Operation of Timer A (one-shot timer mode, external trigger selected)

In one-shot timer mode, choose functions from those listed in Table 2.2.10. Operations of the circled items are described below. Figure 2.2.22 shows the operation timing, and Figure 2.2.23 shows the set-up procedure.

Table 2.2.10. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Pulse output function		No pulses output
	O	Pulses output
Count start condition		External trigger input (falling edge of input signal to the TA0$_{IN}$ pin)
	O	External trigger input (rising edge of input signal to the TA0$_{IN}$ pin)
		Timer overflow (TB1/TX0/TX2 overflow)
		Writing “1” to the one-shot start flag

Operation (1) If the TA0$_{IN}$ pin input level changes from “L” to “H” with the count start flag set to “1”, the counter performs a down count on the count source. At this time, the TA0$_{OUT}$ pin output level goes to “H” level.

(2) If the value of the counter becomes “0000$_{16}$”, the TA0$_{OUT}$ pin outputs an “L” level, and the counter reloads the content of the reload register and stops counting. At this time, the timer A0 interrupt request bit goes to “1”.

(3) If a trigger occurs while a count is in progress, the counter reloads the value of the reload register again and continues counting. The reload timing is in step with the next count source input after the trigger.

(4) Setting the count start flag to “0” causes the counter to stop and to reload the content of the reload register. Also, the TA0$_{OUT}$ pin outputs an “L” level. At this time, the timer A0 interrupt request bit goes to “1”.

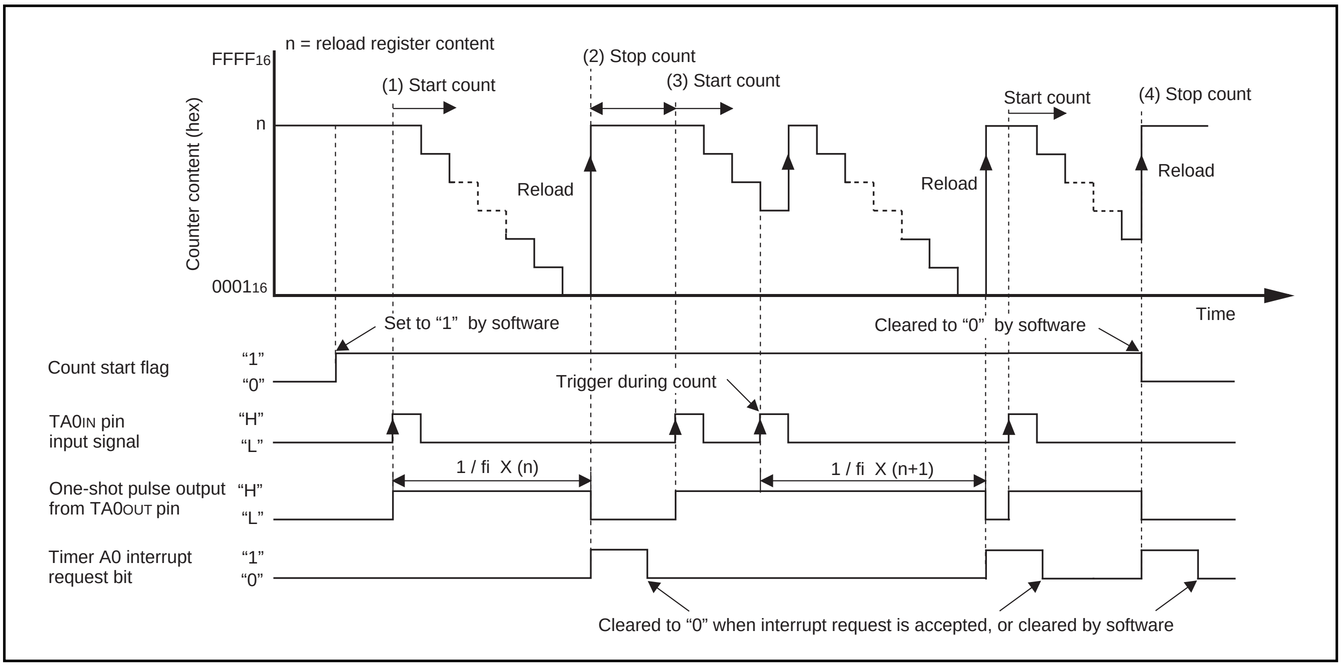
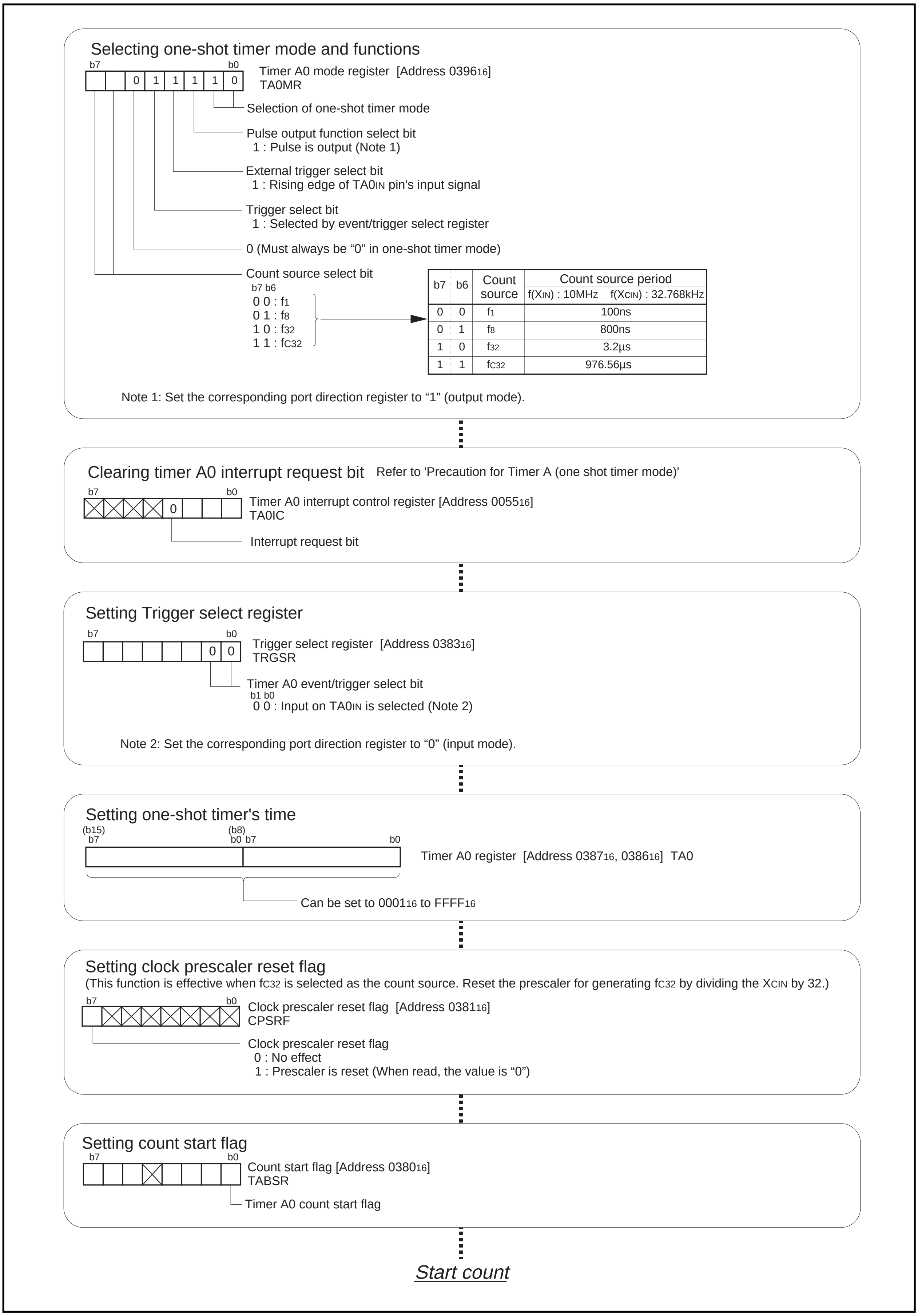


Figure 2.2.22. Operation timing of one-shot mode, external trigger selected

Under development


Selecting one-shot timer mode and functions
b7
b0
0 1 1 1 1 0
Timer A0 mode register [Address 0396_{16}]
TA0MR
Selection of one-shot timer mode
Pulse output function select bit
1 : Pulse is output (Note 1)
External trigger select bit
1 : Rising edge of $TA0_{IN}$ pin's input signal
Trigger select bit
1 : Selected by event/trigger select register
0 (Must always be "0" in one-shot timer mode)
Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : fc_{32}

b7	b6	Count source	Count source period $f(X_{IN})$: 10MHz	 $f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	fc_{32}		976.56µs

Note 1: Set the corresponding port direction register to "1" (output mode).

Clearing timer A0 interrupt request bit
b7
b0
0
Timer A0 interrupt control register [Address 0055_{16}]
TA0IC
Interrupt request bit

Refer to 'Precaution for Timer A (one shot timer mode)'

Setting Trigger select register
b7
b0
0 0
Trigger select register [Address 0383_{16}]
TRGSR
Timer A0 event/trigger select bit
b1 b0
0 0 : Input on $TA0_{IN}$ is selected (Note 2)

Note 2: Set the corresponding port direction register to "0" (input mode).

Setting one-shot timer's time
(b15)
b7
(b8)
b0 b7
b0
Timer A0 register [Address 0387_{16}, 0386_{16}] TA0
Can be set to 0001_{16} to $FFFF_{16}$
Setting clock prescaler reset flag

(This function is effective when fc_{32} is selected as the count source. Reset the prescaler for generating fc_{32} by dividing the X_{CIN} by 32.)

b7
b0
Clock prescaler reset flag [Address 0381_{16}]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count start flag
b7
b0
Count start flag [Address 0380_{16}]
TABSR
Timer A0 count start flag
Start count

Figure 2.2.23. Set-up procedure of one-shot mode, external trigger selected

Under development

2.2.11 Operation of Timer A (pulse width modulation mode, 16-bit PWM mode selected)

In pulse width modulation mode, choose functions from those listed in Table 2.2.11. Operations of the circled items are described below. Figure 2.2.24 shows the operation timing, and Figure 2.2.25 shows the set-up procedure.

Table 2.2.11. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / f_{C32})
PWM mode	O	16-bit PWM
		8-bit PWM
Count start condition		External trigger input (falling edge of input signal to the TA0$_{IN}$ pin)
	O	External trigger input (rising edge of input signal to the TA0$_{IN}$ pin)
		Timer overflow (TB1/TX0/TX2 overflow)

Operation (1) If the TA0$_{IN}$ pin input level changes from “L” to “H” with the count start flag set to “1”, the counter performs a down count on the count source. Also, the TA0$_{OUT}$ pin outputs an “H” level.

(2) The TA0$_{OUT}$ pin output level changes from “H” to “L” when a set time period elapses. At this time, the timer A0 interrupt request bit goes to “1”.

(3) The counter reloads the content of the reload register every time PWM pulses are output for one cycle, and continues counting.

(4) Setting the count start flag to “0” causes the counter to hold its value and to stop. Also, the TA0$_{OUT}$ outputs an “L” level.

Note
- PWM pulse cycle is $(2^{16}-1)/fi$, whereas H level duration is n/fi. However, when “0000$_{16}$” is set for the timer A0 register, the PWM output is “L” level for the entire period, and an interrupt request is generated for every PWM output cycle. Also, when “FFFF$_{16}$” is set for the timer A0 register, the PWM output is “H” level for the entire period, and an interrupt request is generated for every PWM output cycle.
  (fi: Count source frequency f_1, f_8, f_{32}, f_{C32} n: Timer value)

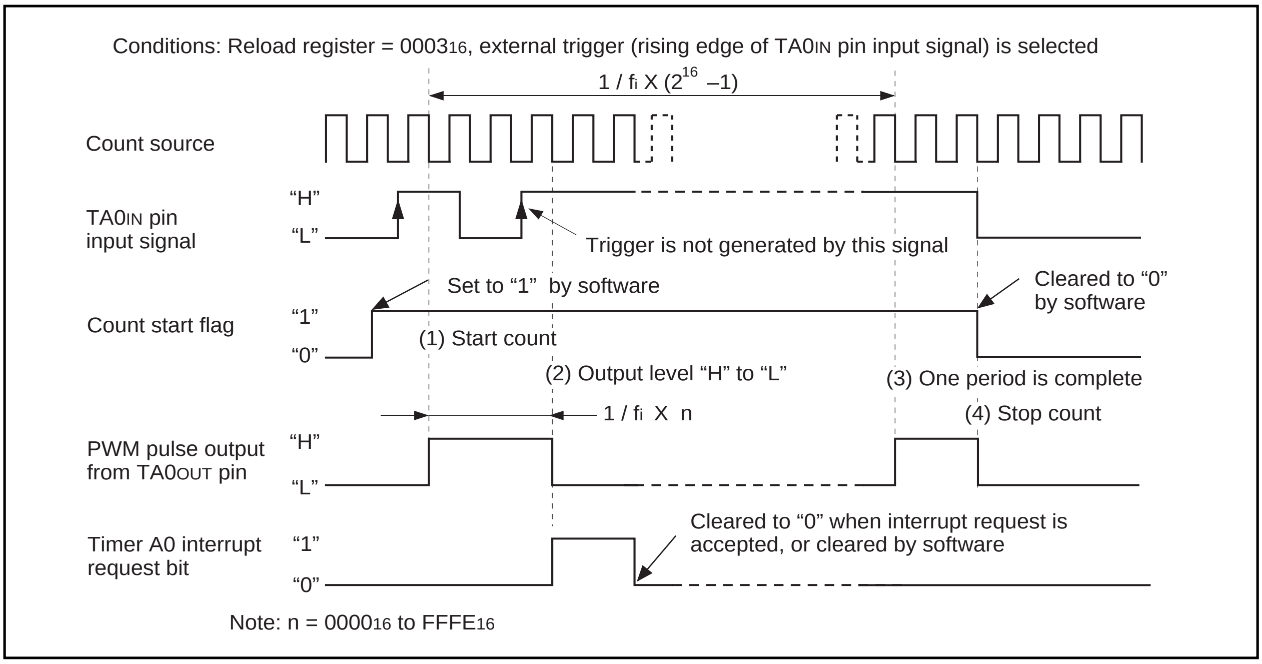
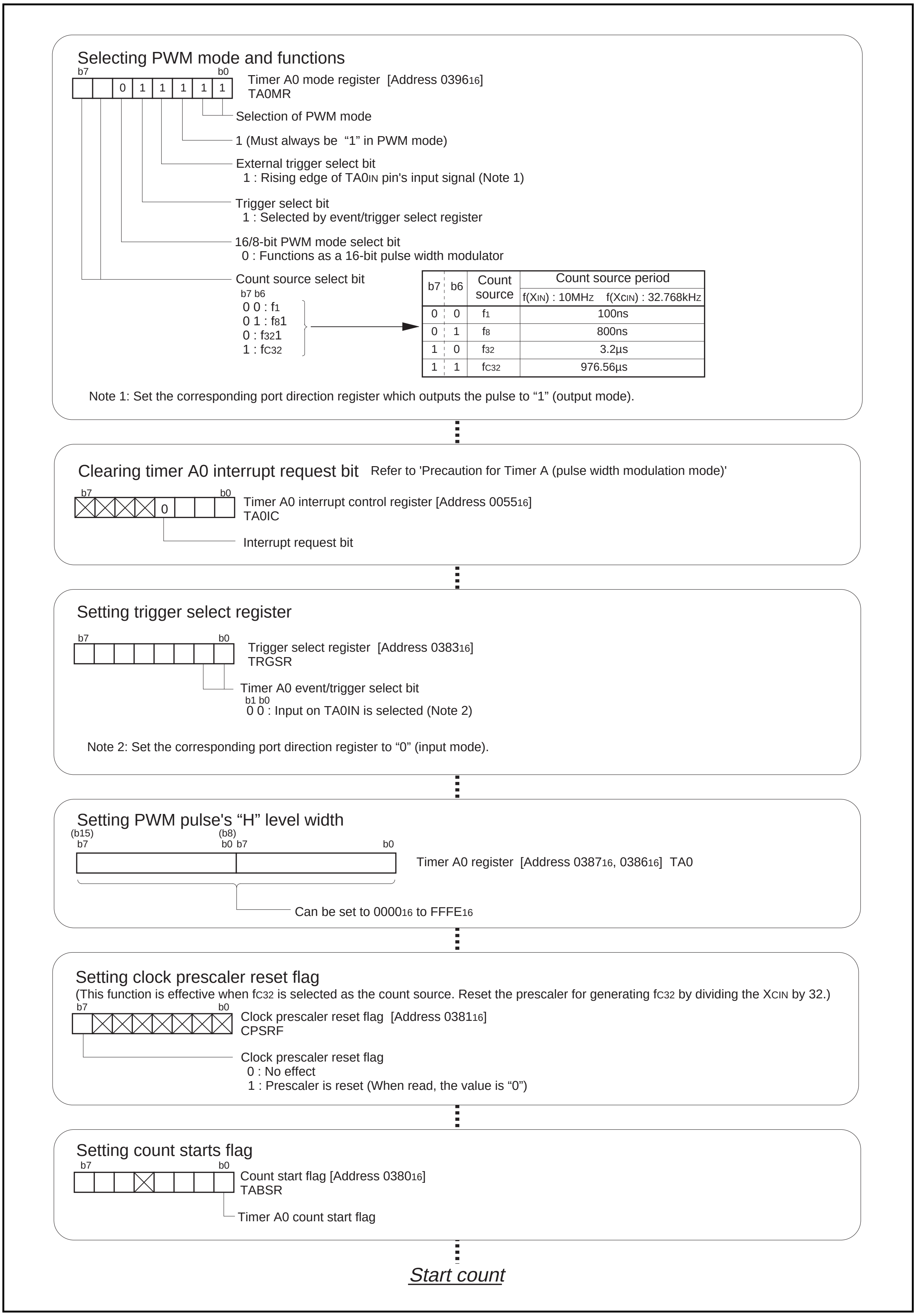

Conditions: Reload register = 0003₁₆, external trigger (rising edge of TA0IN pin input signal) is selected
1 / fi X (2^16 –1)
Count source
TA0IN pin input signal
“H”
“L”
Trigger is not generated by this signal
Set to “1” by software
Cleared to “0” by software
Count start flag
“1”
“0”
(1) Start count
(2) Output level “H” to “L”
(3) One period is complete
1 / fi X n
(4) Stop count
PWM pulse output from TA0OUT pin
“H”
“L”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer A0 interrupt request bit
“1”
“0”
Note: n = 0000₁₆ to FFFE₁₆

Figure 2.2.24. Operation timing of pulse width modulation mode, 16-bit PWM mode selected



Selecting PWM mode and functions

Timer A0 mode register [Address 0396_{16}]
TA0MR

Selection of PWM mode

1 (Must always be "1" in PWM mode)

External trigger select bit
1 : Rising edge of TA0IN pin's input signal (Note 1)

Trigger select bit
1 : Selected by event/trigger select register

16/8-bit PWM mode select bit
0 : Functions as a 16-bit pulse width modulator

Count source select bit
b7 b6
0 0 : f_1
0 1 : $f_8$1
0 : f_{32}1
1 : f_{C32}

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Note 1: Set the corresponding port direction register which outputs the pulse to "1" (output mode).

Clearing timer A0 interrupt request bit Refer to 'Precaution for Timer A (pulse width modulation mode)'

Timer A0 interrupt control register [Address 0055_{16}]
TA0IC

Interrupt request bit

Setting trigger select register

Trigger select register [Address 0383_{16}]
TRGSR

Timer A0 event/trigger select bit
b1 b0
0 0 : Input on TA0IN is selected (Note 2)

Note 2: Set the corresponding port direction register to "0" (input mode).

Setting PWM pulse's "H" level width

Timer A0 register [Address 0387_{16}, 0386_{16}] TA0

Can be set to 0000_{16} to $FFFE_{16}$

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the XCIN by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")

Setting count starts flag

Count start flag [Address 0380_{16}]
TABSR

Timer A0 count start flag

Start count

Figure 2.2.25. Set-up procedure of pulse width modulation mode, 16-bit PWM mode selected

Under development

2.2.12 Operation of Timer A (pulse width modulation mode, 8-bit PWM mode selected)

In pulse width modulation mode, choose functions from those listed in Table 2.2.12. Operations of the circled items are described below. Figure 2.2.26 shows the operation timing, and Figure 2.2.27 shows the set-up procedure.

Table 2.2.12. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
PWM mode		16-bit PWM
	O	8-bit PWM
Count start condition	O	External trigger input (falling edge of input signal to the $TA0_{IN}$ pin)
		External trigger input (rising edge of input signal to the $TA0_{IN}$ pin)
		Timer overflow (TB1/TX0/TX2 overflow)

Operation (1) If the $TA0_{IN}$ pin input level changes from “H” to “L” with the count start flag set to “1”, the counter performs a down count on the count source. Also, the $TA0_{OUT}$ pin outputs an “H” level.

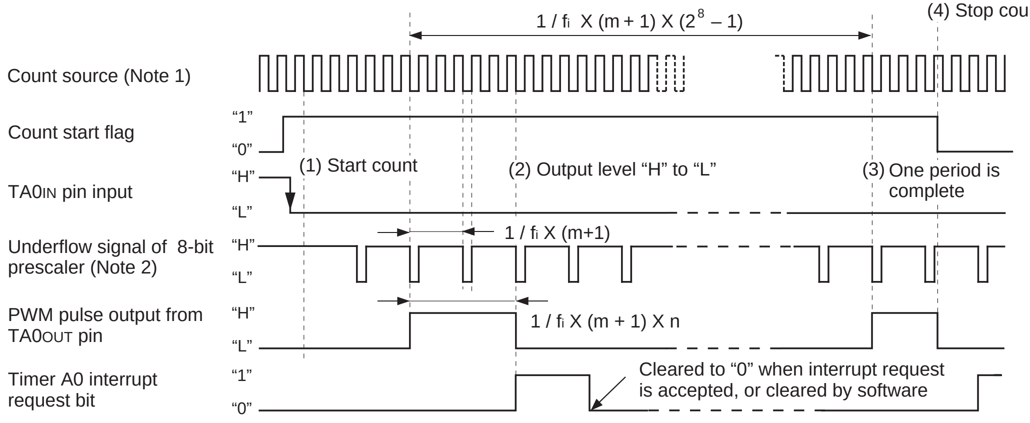
(2) The $TA0_{OUT}$ pin output level changes from “H” to “L” when a set time period elapses. At this time, the timer A0 interrupt request bit goes to “1”.

(3) The counter reloads the content of the reload register every time PWM pulses are output for one cycle, and continues counting.

(4) Setting the count start flag to “0” causes the counter to hold its value and to stop. Also, the $TA0_{OUT}$ pin outputs an “L” level.

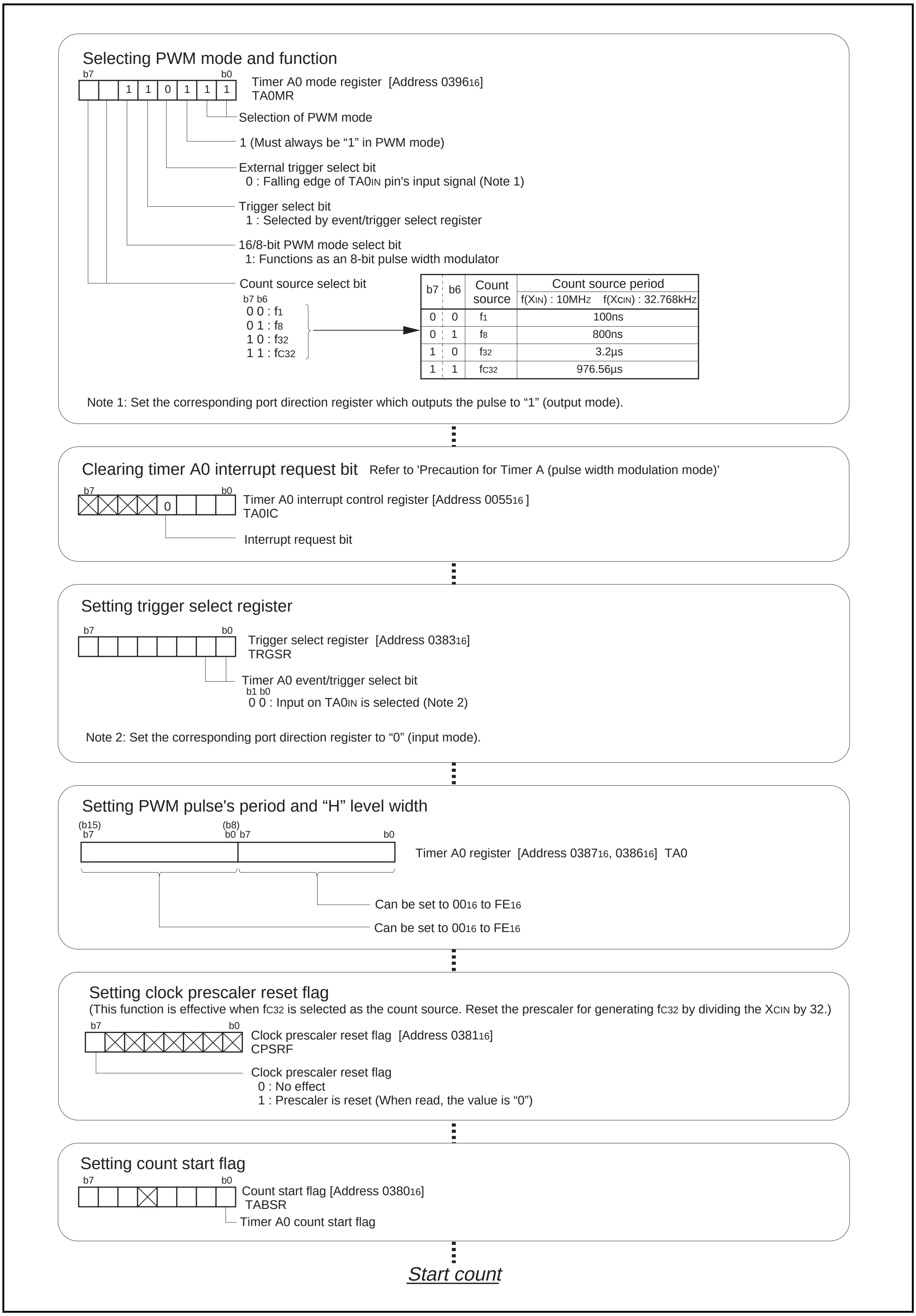
Note • PWM pulse cycle is (m + 1(x (2^8 -1)/fi, whereas “H” level duration is n x (m + 1)/fi. However, when “00_{16}” is set for the significant 8 bits of the timer A0 register, the PWM output is “L” level for the entire period, and an interrupt request is generated for every PWM output cycle. Also, when “FF_{16}” is set for the significant 8 bits of the timer A0 register, the PWM output is “H” level for the entire period, and an interrupt request is generated for every PWM output cycle.
(fi: Count source frequency f_1, f_8, f_{32}, fc_{32} n: Timer value)

Conditions: Reload register high-order 8 bits = 02_{16}
Reload register low-order 8 bits = 02_{16}
External trigger (falling edge of $TA0_{IN}$ pin input signal) is selected


1 / fi X (m + 1) X (2^8 – 1)
(4) Stop count
Count source (Note 1)
Count start flag
“1”
“0”
TA0IN pin input
“H”
“L”
(1) Start count
(2) Output level “H” to “L”
(3) One period is complete
Underflow signal of 8-bit prescaler (Note 2)
“H”
“L”
1 / fi X (m+1)
PWM pulse output from TA0OUT pin
“H”
“L”
1 / fi X (m + 1) X n
Timer A0 interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software

Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00_{16} to FF_{16}; n = 00_{16} to FF_{16}.

Figure 2.2.26. Operation timing of pulse width modulation mode, with 8-bit PWM mode selected


Selecting PWM mode and function
b7
b0
1 1 0 1 1 1
Timer A0 mode register [Address 0396₁₆]
TA0MR
Selection of PWM mode
1 (Must always be “1” in PWM mode)
External trigger select bit
0 : Falling edge of TA0IN pin's input signal (Note 1)
Trigger select bit
1 : Selected by event/trigger select register
16/8-bit PWM mode select bit
1: Functions as an 8-bit pulse width modulator
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
Clearing timer A0 interrupt request bit
Refer to 'Precaution for Timer A (pulse width modulation mode)'
0
Timer A0 interrupt control register [Address 0055₁₆]
TA0IC
Interrupt request bit
Setting trigger select register
Trigger select register [Address 0383₁₆]
TRGSR
Timer A0 event/trigger select bit
b1 b0
0 0 : Input on TA0IN is selected (Note 2)
Setting PWM pulse's period and “H” level width
(b15)
(b8)
b7
Timer A0 register [Address 0387₁₆, 0386₁₆] TA0
Can be set to 00₁₆ to FE₁₆
Can be set to 00₁₆ to FE₁₆
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
Clock prescaler reset flag [Address 0381₁₆]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)
Setting count start flag
Count start flag [Address 0380₁₆]
TABSR
Timer A0 count start flag
Start count

b7	b6	Count source	Count source period f(XIN) : 10MHz	f(XCIN) : 32.768kHz
0	0	f1	100ns	
0	1	f8	800ns	
1	0	f32	3.2µs	
1	1	fC32		976.56µs

Note 1: Set the corresponding port direction register which outputs the pulse to “1” (output mode).

Note 2: Set the corresponding port direction register to “0” (input mode).

Figure 2.2.27. Set-up procedure of pulse width modulation mode, 8-bit PWM mode selected

2.2.13 Precautions for Timer A (timer mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer A0 register, then set the flag to “1”.

(2) Reading the timer A0 register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer A0 register with the reload timing shown in Figure 2.2.28 gets “$FFFF_{16}$”. Reading the timer A0 register after setting a value in the timer A0 register with a count halted but before the counter starts counting gets a proper value.

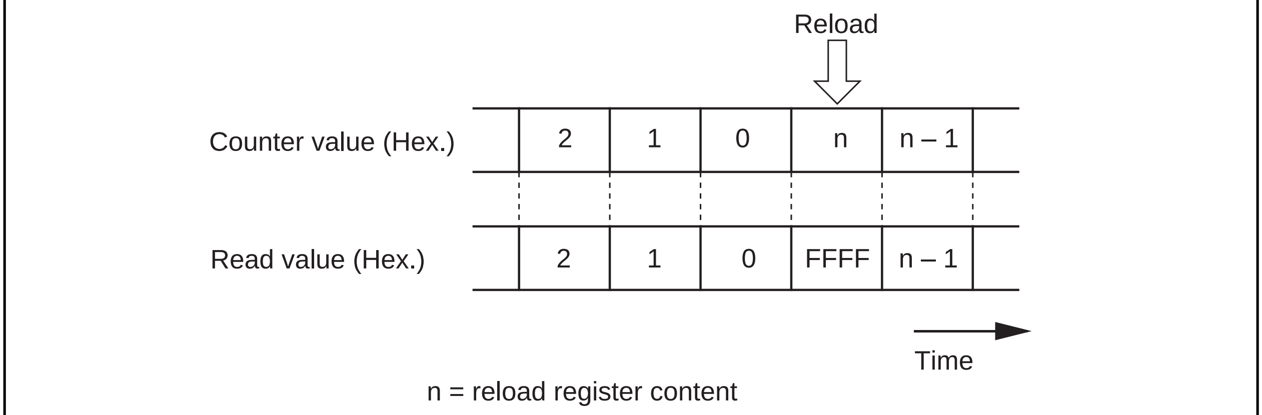


Figure 2.2.28. Reading timer A0 register

2.2.14 Precautions for Timer A (event counter mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer A0 register, then set the flag to “1”.

(2) Reading the timer A0 register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer A0 register with the reload timing shown in Figure 2.2.29 gets “FFFF$_{16}$” by underflow or “0000$_{16}$” by overflow. Reading the timer A0 register after setting a value in the timer A0 register with a count halted but before the counter starts counting gets a proper value.

(3) Please note the standards for the differences between the 2 pulses used in the 2-phase pulse signals input signals to the $TA0_{IN}$ pin and $TA0_{OUT}$ pin as shown in Figure 2.2.30.

(4) When free run type is selected, if count is stopped, set a value in the timer A0 register again.

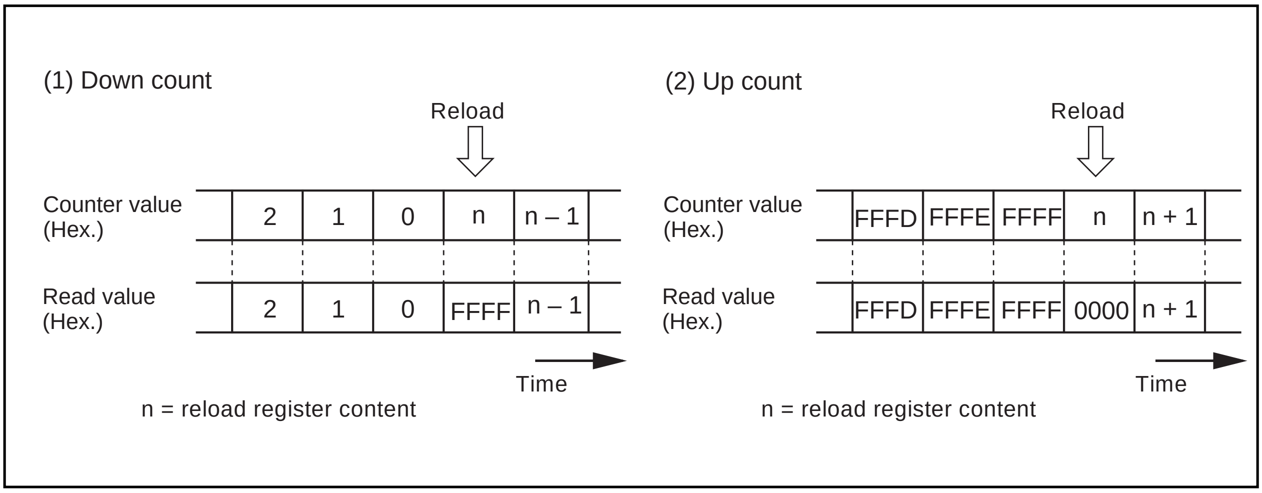
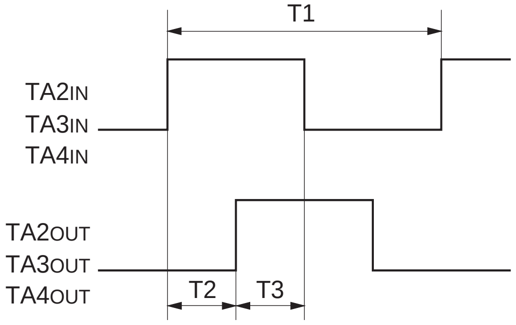


Figure 2.2.29. Reading timer A0 register



Vcc = 5V, f(X_{IN}) = 10MHz

T1 (Min.)	T2, T3 (Min.)
800ns	200ns

Vcc = 3V, f(X_{IN}) = 7MHz, one-wait

T1 (Min.)	T2, T3 (Min.)
2µs	500ns

Figure 2.2.30. Standard of 2-phase pulses

2.2.15 Precautions for Timer A (one-shot timer mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer A0 register, then set the flag to “1”.

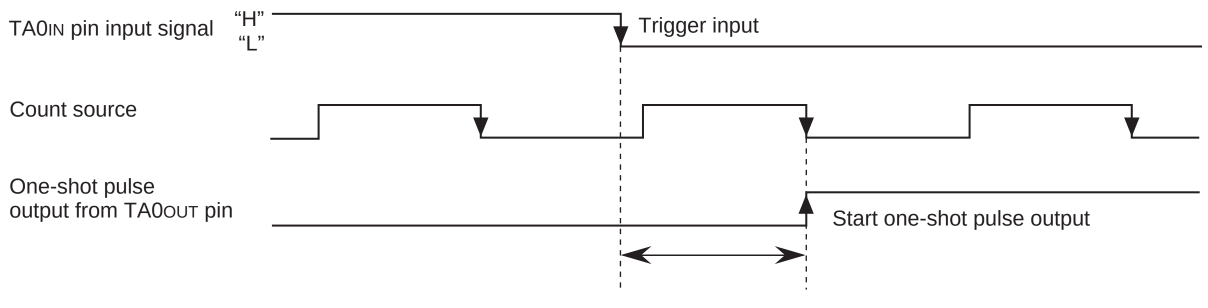
(2) Setting the count start flag to “0” while a count is in progress causes as follows:
- The counter stops counting and a content of reload register is reloaded.
- The TA0OUT pin outputs “L” level.
- The interrupt request generated and the timer A0 interrupt request bit goes to “1”.

(3) The output from the one-shot timer synchronizes with the count source generated internally. Therefore, when an external trigger has been selected, a delay of one cycle of the maximum count source occurs between the trigger input to the TA0IN pin and the one-shot timer output.

(4) The timer A0 interrupt request bit goes to “1” if the timer's operation mode is set using any of the following procedures:
- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer A0 interrupt (interrupt request bit), set timer A0 interrupt request bit to “0” after the above listed changes have been made.

(5) If a trigger occurs while a count is in progress, after the counter performs one down count following the reoccurrence of a trigger, the reload register contents are reloaded, and the count continues. To generate a trigger while a count is in progress, generate the second trigger after an elapse longer than one cycle of the timer's count source after the previous trigger occurred.


TA0IN pin input signal
“H”
“L”
Trigger input
Count source
One-shot pulse output from TA0OUT pin
Start one-shot pulse output

Note: The above applies when an external trigger (falling edge of TA0IN pin input signal) is selected.

Figure 2.2.31. One-shot timer delay

Under development

2.2.16 Precautions for Timer A (pulse width modulation mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer A0 register, then set the flag to “1”.

(2) The timer A0 interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:

- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer A0 interrupt (interrupt request bit), set timer A0 interrupt request bit to “0” after the above listed changes have been made.

(3) Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TA0OUT pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer A0 interrupt request bit goes to “1”. If the TA0OUT pin is outputting an “L” level in this instance, the level does not change, and the timer A0 interrupt request bit does not becomes “1”.

(4) Normal PWM output is restored according to the interrupt request generate timing, both in the case of 16-bit PWM and 8-bit PWM, when PWM output is either “H” or “L” level for the entire period. This holds only when a value other than “0000_{16}” or “$FFFF_{16}$” is set during 16-bit PWM, or a value other than “00_{16}” or “FF_{16}” is set during 8-bit PWM.

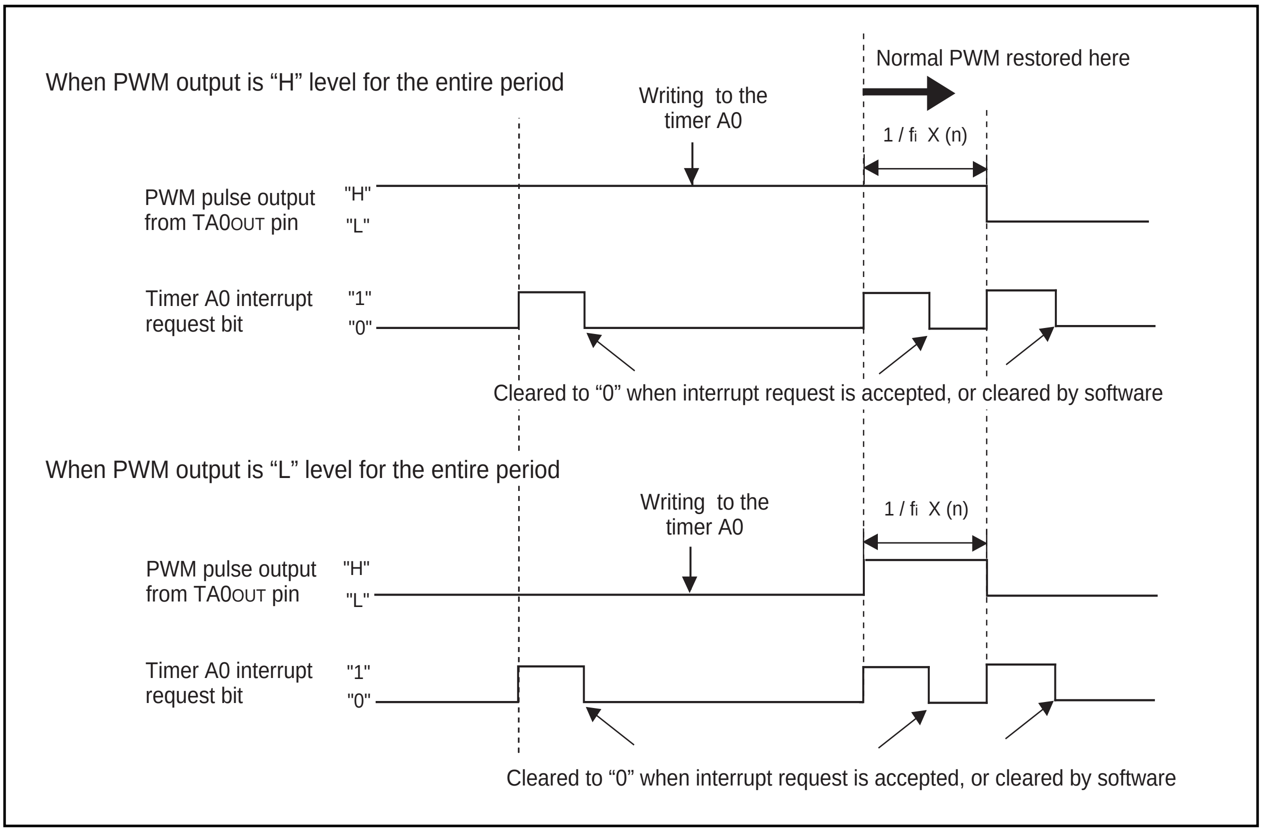

When PWM output is “H” level for the entire period
Normal PWM restored here
Writing to the timer A0
1 / fi X (n)
PWM pulse output from TA0OUT pin
"H"
"L"
Timer A0 interrupt request bit
"1"
"0"
Cleared to “0” when interrupt request is accepted, or cleared by software
When PWM output is “L” level for the entire period
Writing to the timer A0
1 / fi X (n)
PWM pulse output from TA0OUT pin
"H"
"L"
Timer A0 interrupt request bit
"1"
"0"
Cleared to “0” when interrupt request is accepted, or cleared by software

Figure 2.2.32. Operation timing of PWM output mode

2.3 Timer B

2.3.1 Overview

The following is an overview for timer B, a 16-bit timer.

(1) Mode

Timer B operates in one of three modes:

(a) Timer mode

The internal count source is counted.

- Operation in timer mode P198

(b) Event counter mode

The number of pulses coming from outside and the number of the timer overflows are counted.

- Operation in event counter mode P200

(c) Pulse period measurement/pulse width measurement mode

External pulse period or external pulse widths are measured. If pulse period measurement mode is selected, the periods of input pulses are continuously measured. If pulse width measurement mode is selected, widths of “H” level pulses and those of “L” level pulses are continuously measured.

- Operation in pulse period measurement mode P202
- Operation in pulse width measurement mode P204

(2) Count source

An internal count source can be selected from f_1, f_8, f_{32}, and f_{C32}. f_1, f_8, and f_{32} are clocks obtained by dividing the CPU main clock by 1, 8, and 32 respectively. f_{C32} is the clock obtained by dividing the CPU secondary clock by 32.

(3) Frequency division ratio

The frequency division ratio equals [the value set in the timer register + 1]. The counter underflows when a count source equal to a frequency division ratio is input, and an interrupt request occurs.

(4) Reading the timer

In timer mode or event counter mode, the count value at the time of reading the timer register will be read. Read the register in 16-bit increments. In both the pulse period measurement mode and pulse width measurement mode, an indeterminate value is read until the second effective edge is input after a count is started, otherwise, the measurement results are read.

(5) Writing to the timer

When writing to the timer register while a count is in progress, the value is written only to the reload register. When writing to the timer register while a count has stopped, the value is written both to the reload register and the count. Write the value in 16-bit increments. The timer register cannot be written to in either the pulse period measurement mode or the pulse width measurement mode.

Under development

(6) Input to the timer and the direction register

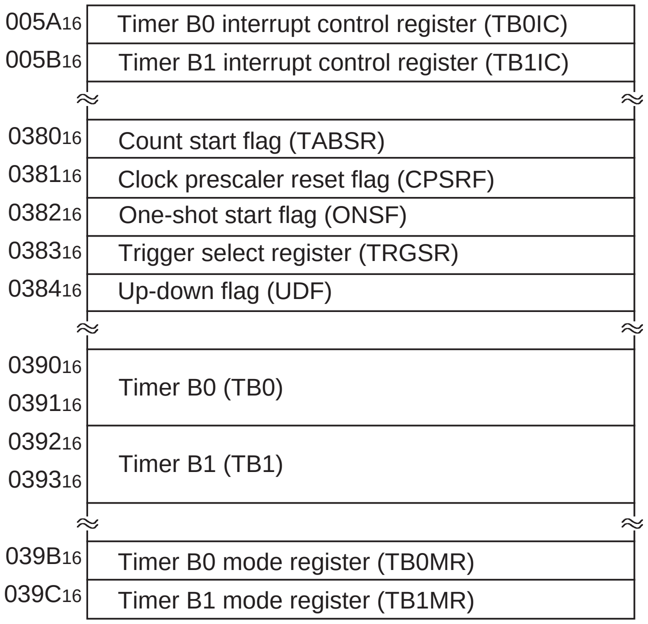
To input an external signal to the timer, set the direction register of the relevant port to input.

(7) Pins related to timer B

(a) $TB0_{IN}$, $TB1_{IN}$ Input pins to timer B.

(8) Registers related to timer B

Figure 2.3.1 shows the memory map of timer B-related registers. Figures 2.3.2 and 2.3.3 show timer B-related registers.

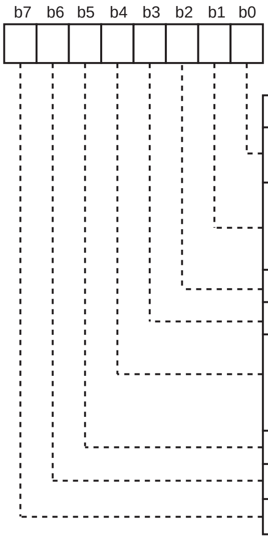


$005A_{16}$	Timer B0 interrupt control register (TB0IC)
$005B_{16}$	Timer B1 interrupt control register (TB1IC)
≈	≈
0380_{16}	Count start flag (TABSR)
0381_{16}	Clock prescaler reset flag (CPSRF)
0382_{16}	One-shot start flag (ONSF)
0383_{16}	Trigger select register (TRGSR)
0384_{16}	Up-down flag (UDF)
≈	≈
0390_{16} 0391_{16}	Timer B0 (TB0)
0392_{16} 0393_{16}	Timer B1 (TB1)
≈	≈
$039B_{16}$	Timer B0 mode register (TB0MR)
$039C_{16}$	Timer B1 mode register (TB1MR)

Figure 2.3.1. Memory map of timer B-related registers

Under development

Timer Bi mode register



Symbol TBiMR(i = 0, 1)
Address 039B$_{16}$, 039C$_{16}$
When reset 00XX0000$_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1)	○
			× (Note 2)	×
MR3			○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Timer B0.
Note 2: Timer B1.

Figure 2.3.2. Timer B-related registers (1)

Timer Bi register (Note)

(b15) b7 … (b8) b0 b7 … b0

Symbol	Address	When reset
TB0	0391_{16}, 0390_{16}	Indeterminate
TB1	0393_{16}, 0392_{16}	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000_{16} to $FFFF_{16}$	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000_{16} to $FFFF_{16}$	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

Note1: Read and write data in 16-bit units.

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0380_{16}	$000X0000_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Clock prescaler reset flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 2.3.3. Timer B-related registers (2)

2.3.2 Operation of Timer B (timer mode)

In timer mode, choose functions from those listed in Table 2.3.1. Operations of the circled items are described below. Figure 2.3.4 shows the operation timing, and Figure 2.3.5 shows the set-up procedure.

Table 2.3.1. Choosed functions

Item	Set-up	
Count source	**O**	Internal count source (f_1 / f_8 / f_{32} / fc_{32})

Operation (1) Setting the count start flag to “1” causes the counter to perform a down count on the count source.

(2) If an underflow occurs, the content of the reload register is reloaded, and the counter continues counting. At this time, the timer Bi interrupt request bit goes to “1”.

(3) Setting the count start flag to “0” causes the counter to hold its value and to stop.

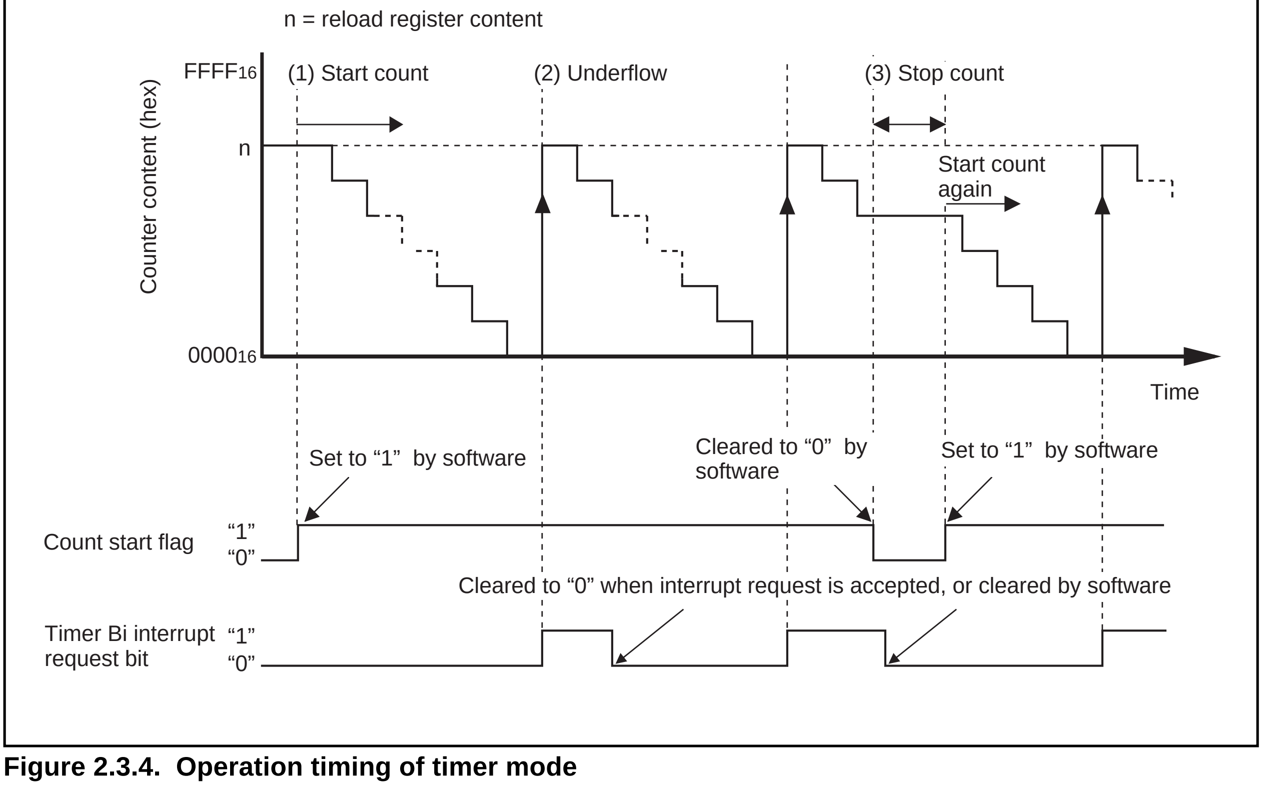
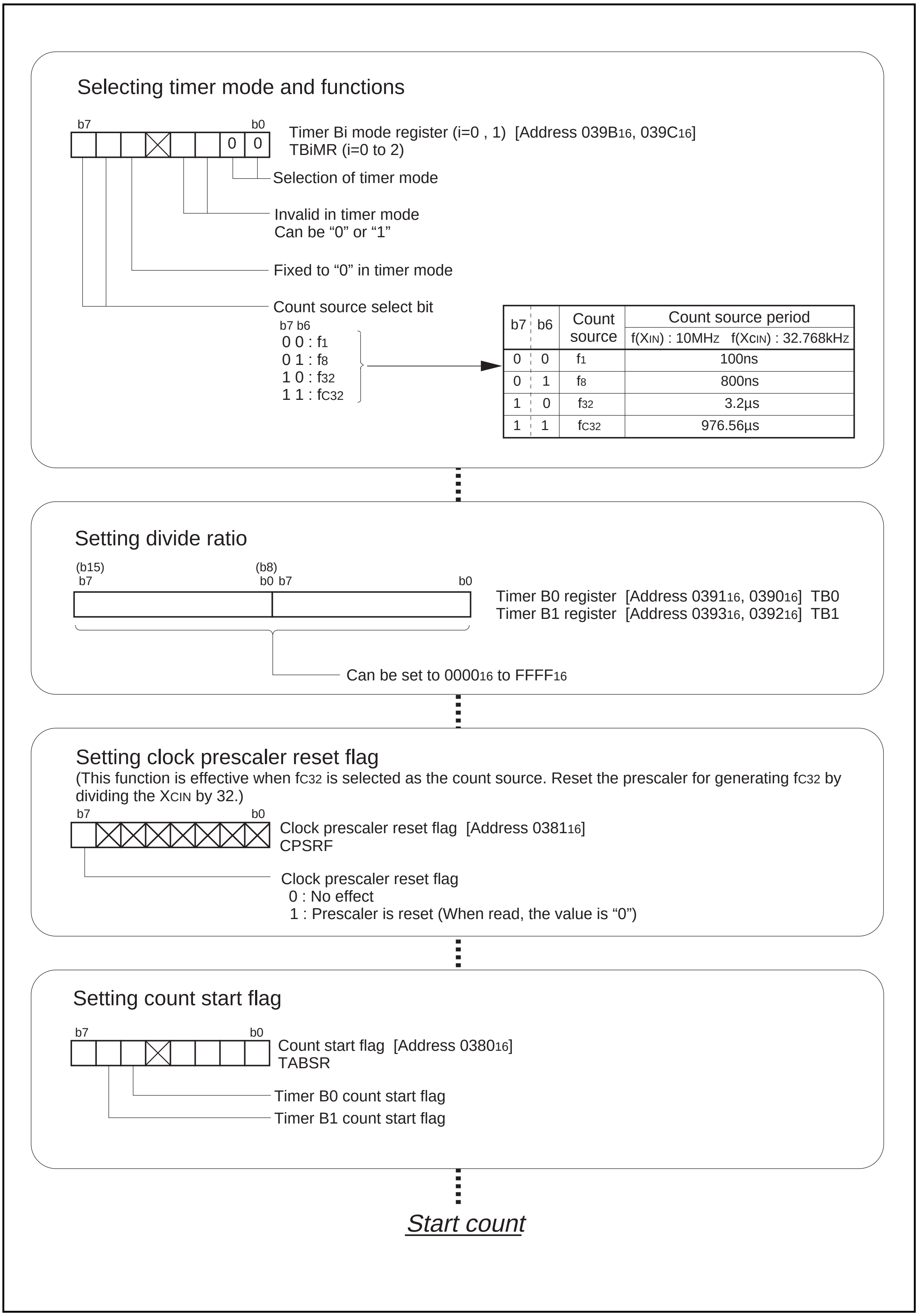


Figure 2.3.4. Operation timing of timer mode



Selecting timer mode and functions

Timer Bi mode register (i=0 , 1) [Address 039B$_{16}$, 039C$_{16}$]
TBiMR (i=0 to 2)

Selection of timer mode

Invalid in timer mode
Can be “0” or “1”

Fixed to “0” in timer mode

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period	
			f(X_{IN}) : 10MHz	f(X_{CIN}) : 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Setting divide ratio

Timer B0 register [Address 0391$_{16}$, 0390$_{16}$] TB0
Timer B1 register [Address 0393$_{16}$, 0392$_{16}$] TB1

Can be set to 0000$_{16}$ to FFFF$_{16}$

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 0381$_{16}$]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380$_{16}$]
TABSR

Timer B0 count start flag
Timer B1 count start flag

Start count

Figure 2.3.5. Set-up procedure of timer mode

2.3.3 Operation of Timer B (event counter mode)

In event counter mode, choose functions from those listed in Table 2.3.2. Operations of the circled items are described below. Figure 2.3.6 shows the operation timing, and Figure 2.3.7 shows the set-up procedure.

Table 2.3.2. Choosed functions

Item	Set-up	
Count source	O	Input signal to the TBiIN pin (counting falling edges)
		Input signal to the TBiIN pin (counting rising edges)
		Input signal to the TBiIN pin (counting rising edges and falling edges)
		Timer overflow(TBj overflow)

Operation (1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.

(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer Bi interrupt request bit goes to “1”.

(3) Setting the count start flag to “0” causes the counter to hold its value and to stop.

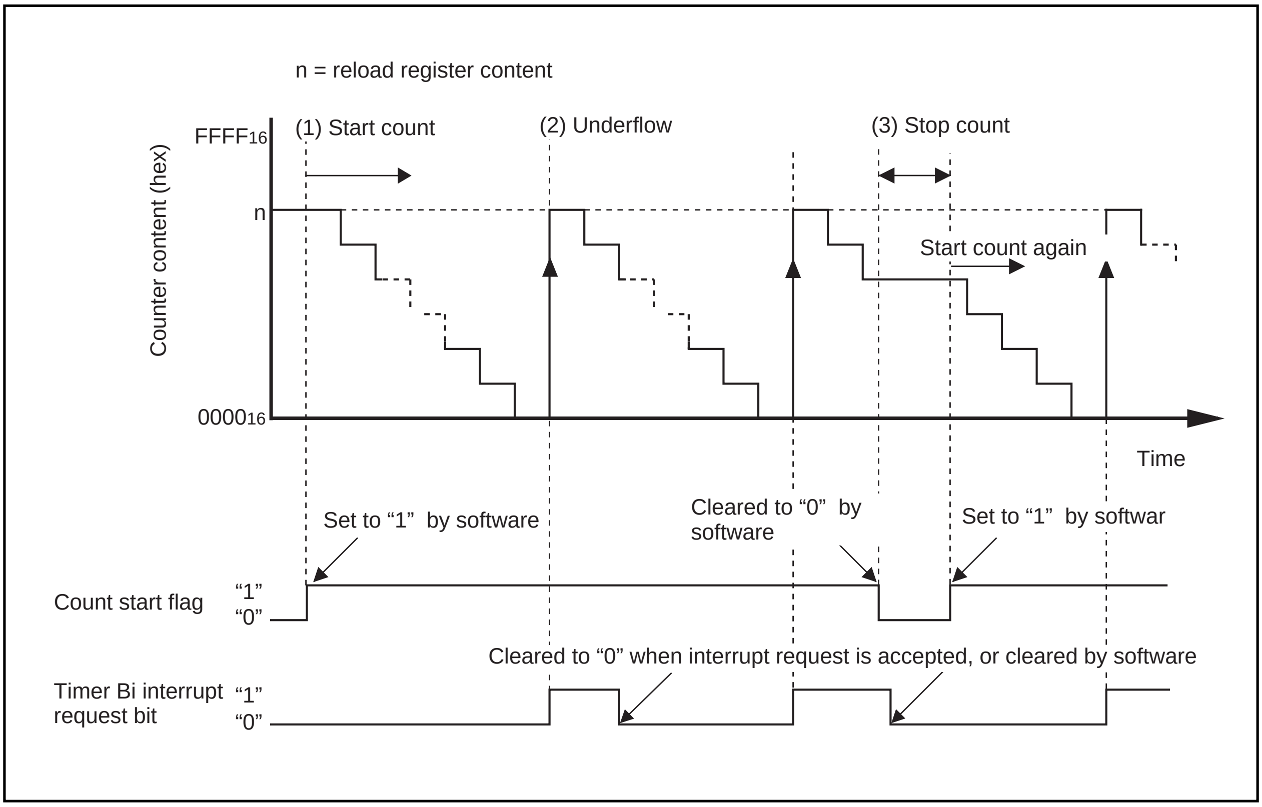


Figure 2.3.6. Operation timing of event counter mode

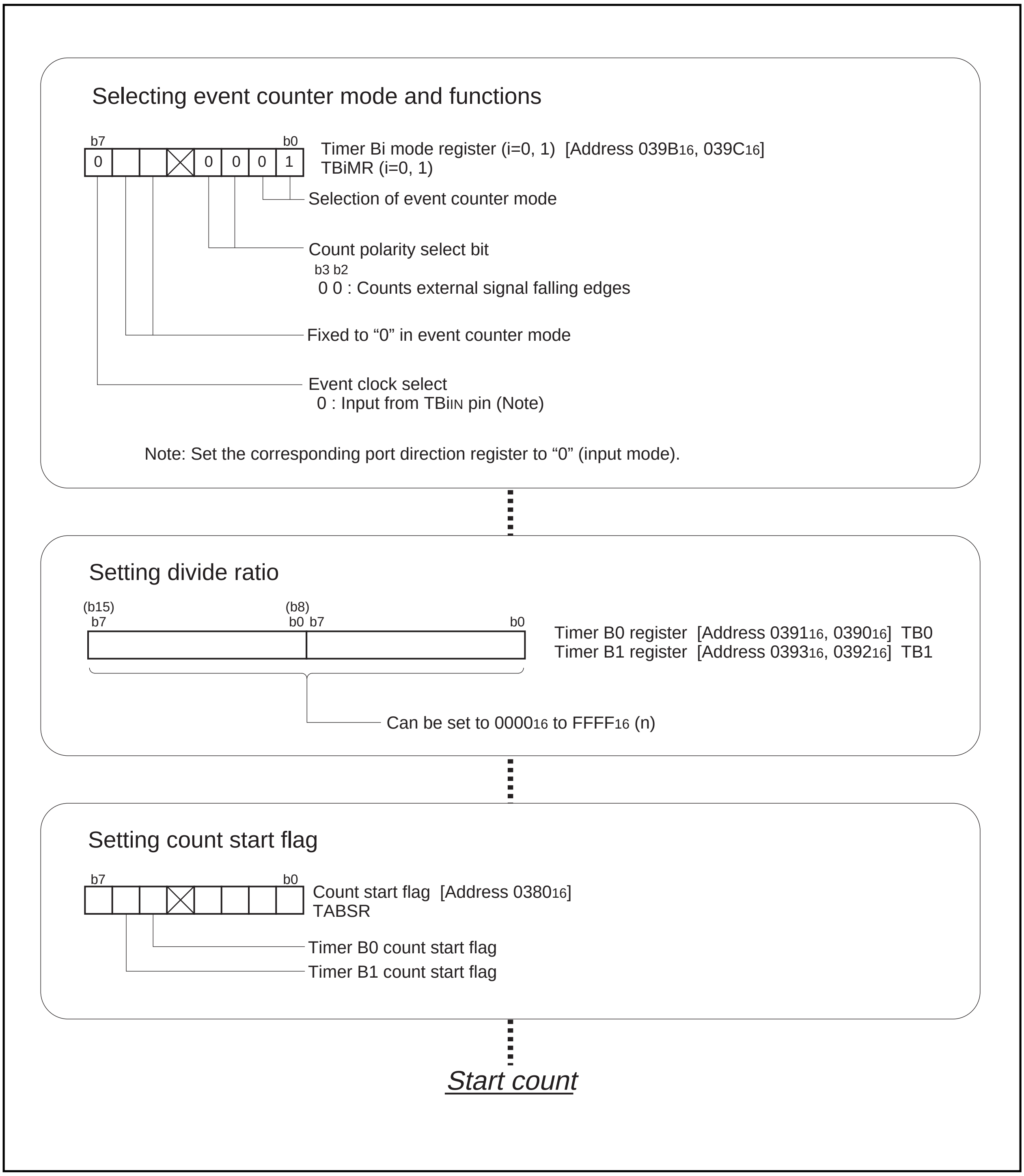

Selecting event counter mode and functions
b7
b0
0
0
0
0
1
Timer Bi mode register (i=0, 1) [Address $039B_{16}$, $039C_{16}$]
TBiMR (i=0, 1)
Selection of event counter mode
Count polarity select bit
b3 b2
0 0 : Counts external signal falling edges
Fixed to "0" in event counter mode
Event clock select
0 : Input from TBi_{IN} pin (Note)
Note: Set the corresponding port direction register to "0" (input mode).
Setting divide ratio
(b15)
b7
(b8)
b0 b7
b0
Timer B0 register [Address 0391_{16}, 0390_{16}] TB0
Timer B1 register [Address 0393_{16}, 0392_{16}] TB1
Can be set to 0000_{16} to $FFFF_{16}$ (n)
Setting count start flag
b7
b0
Count start flag [Address 0380_{16}]
TABSR
Timer B0 count start flag
Timer B1 count start flag
Start count

Figure 2.3.7. Set-up procedure of event counter mode

2.3.4 Operation of Timer B (pulse period measurement mode)

In pulse period/pulse width measurement mode, choose functions from those listed in Table 2.3.3. Operations of the circled items are described below. Figure 2.3.8 shows the operation timing, and Figure 2.3.9 shows the set-up procedure.

Table 2.3.3. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / f_{C32})
Measurement mode	O	Pulse period measurement (interval between measurement pulse falling edge to falling edge)
		Pulse period measurement (interval between measurement pulse rising edge to rising edge)
		Pulse width measurement (interval between measurement pulse falling edge to rising edge, and between rising edge to falling edge)

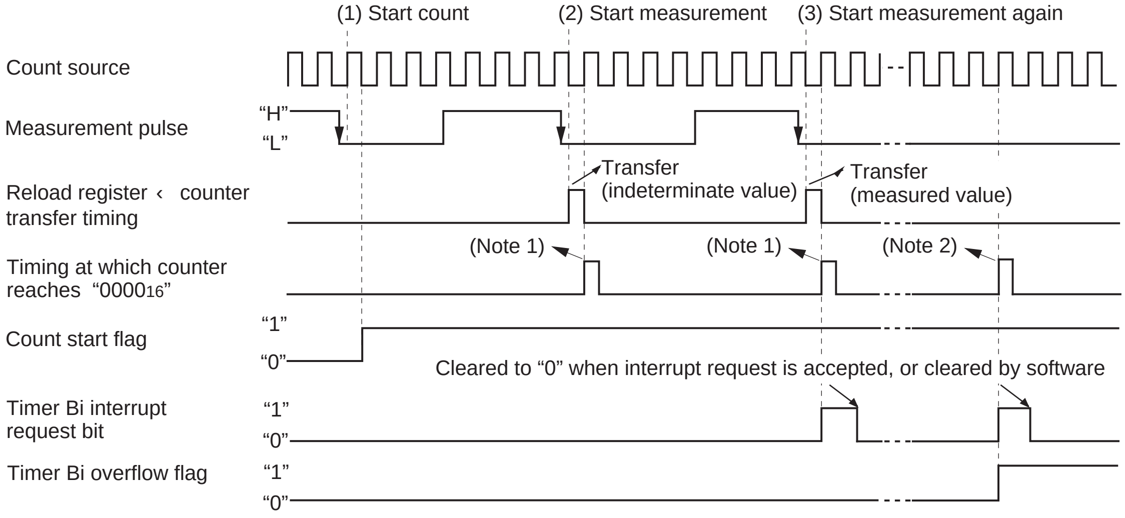
Operation (1) Setting the count start flag to “1” causes the counter to start counting the count source.

(2) If a measurement pulse changes from “H” to “L”, the value of the counter goes to “0000$_{16}$”, and measurement is started. In this instance, an indeterminate value is transferred to the reload register. The timer Bi interrupt request does not generate.

(3) If a measurement pulse changes from “H” to “L” again, the value of the counter is transferred to the reload register, and the timer Bi interrupt request bit goes to “1”. Then the value of the counter becomes “0000$_{16}$”, and the measurement is started again.

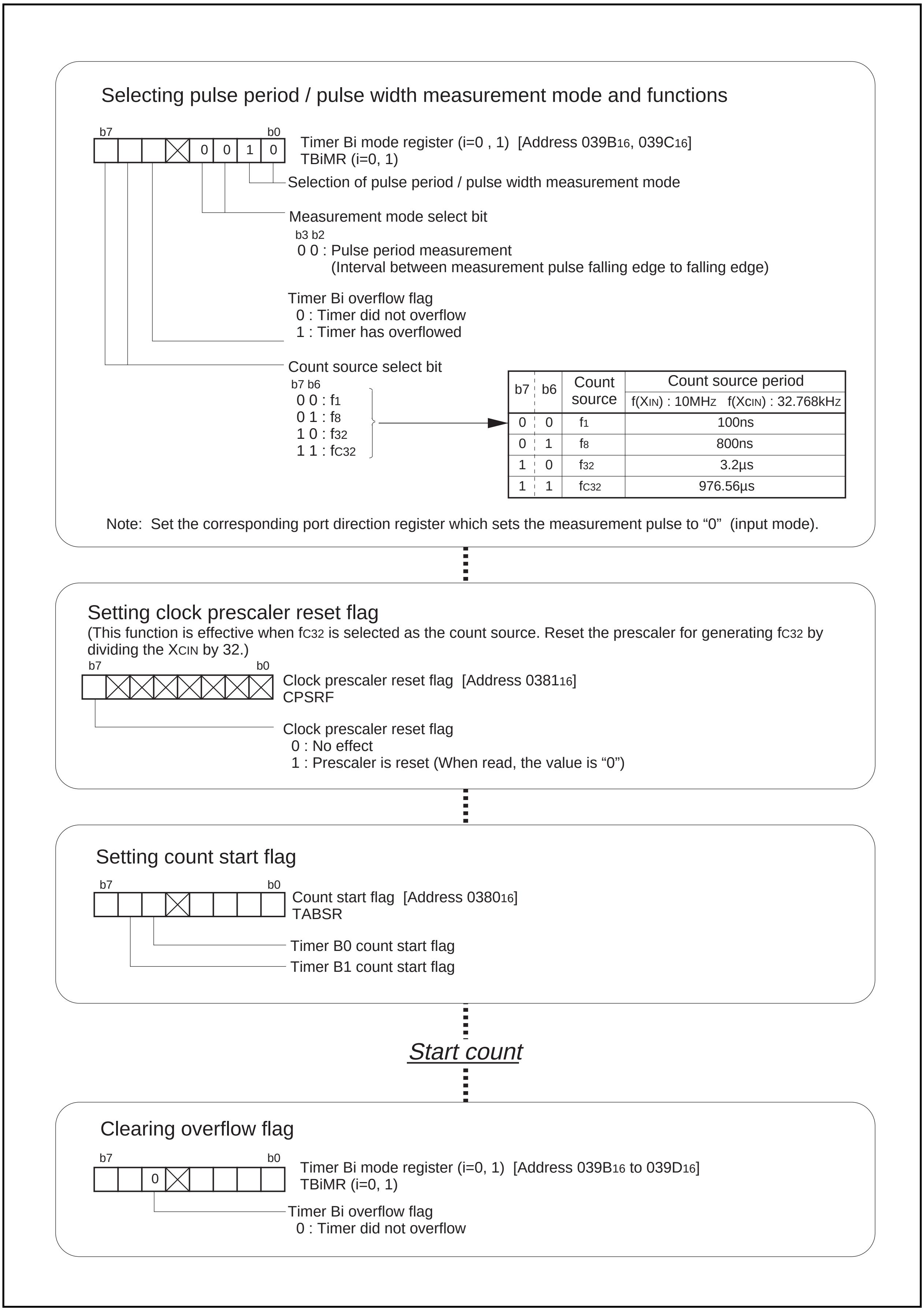
Note
- The timer Bi interrupt request bit goes to “1” when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the timer Bi overflow flag within the interrupt routine.
- The value of the counter at the beginning of a count is indeterminate. Thus there can be instances in which the timer Bi overflow flag goes to “1” immediately after a count is performed.
- The timer Bi overflow flag goes to “0” if timer Bi mode register is written to when the count start flag is “1”. This flag cannot be set to “1” by software.

Measurement of pulse time interval from falling edge to falling edge


(1) Start count
(2) Start measurement
(3) Start measurement again
Count source
Measurement pulse
“H”
“L”
Reload register ← counter transfer timing
Transfer (indeterminate value)
Transfer (measured value)
Timing at which counter reaches “0000$_{16}$”
(Note 1)
(Note 1)
(Note 2)
Count start flag
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Bi interrupt request bit
“1”
“0”
Timer Bi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 2.3.8. Operation timing of pulse period measurement mode



Selecting pulse period / pulse width measurement mode and functions

Timer Bi mode register (i=0 , 1) [Address 039B16, 039C16]
TBiMR (i=0, 1)

Selection of pulse period / pulse width measurement mode

Measurement mode select bit
b3 b2
0 0 : Pulse period measurement
(Interval between measurement pulse falling edge to falling edge)

Timer Bi overflow flag
0 : Timer did not overflow
1 : Timer has overflowed

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period	
			f(XIN) : 10MHz	f(XCIN) : 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Note: Set the corresponding port direction register which sets the measurement pulse to “0” (input mode).

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 038116]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 038016]
TABSR

Timer B0 count start flag
Timer B1 count start flag

Start count

Clearing overflow flag

Timer Bi mode register (i=0, 1) [Address 039B16 to 039D16]
TBiMR (i=0, 1)

Timer Bi overflow flag
0 : Timer did not overflow

Figure 2.3.9. Set-up procedure of pulse period measurement mode

2.3.5 Operation of Timer B (pulse width measurement mode)

In pulse period/pulse width measurement mode, choose functions from those listed in Table 2.3.4. Operations of the circled items are described below. Figure 2.3.10 shows the operation timing, and Figure 2.3.11 shows the set-up procedure.

Table 2.3.4. Choosed functions

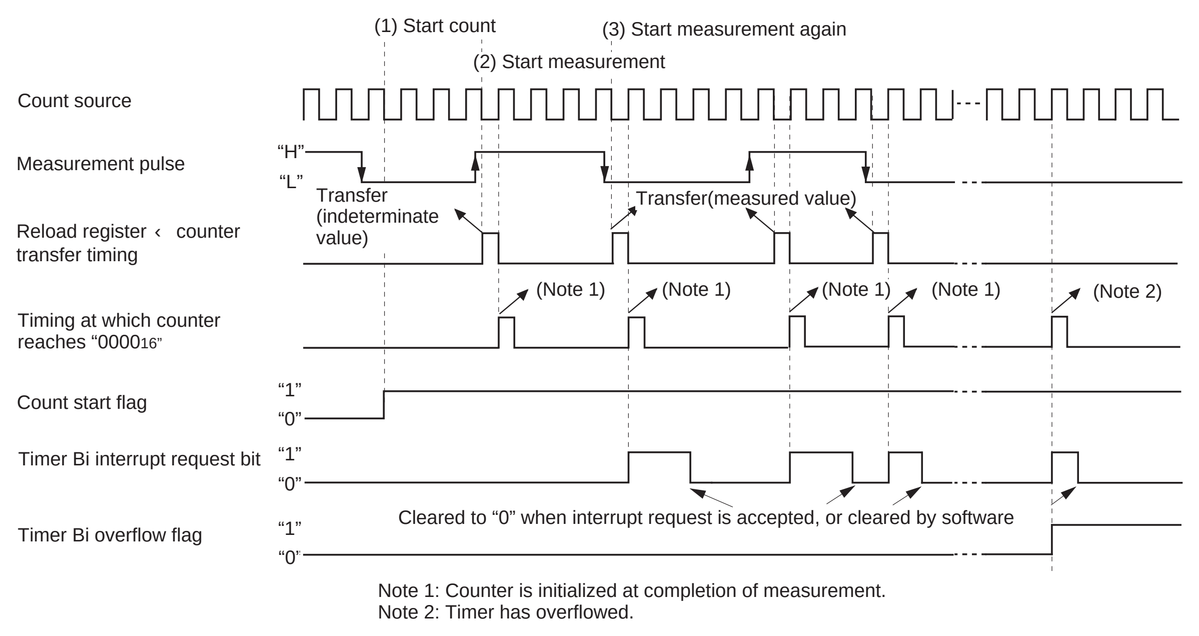
Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Measurement mode		Pulse period measurement (interval between measurement pulse falling edge to falling edge)
		Pulse period measurement (interval between measurement pulse rising edge to rising edge)
	O	Pulse width measurement (interval between measurement pulse falling edge to rising edge, and between rising edge to falling edge)

Operation (1) Setting the count start flag to “1” causes the counter to start counting the count source.

(2) If an effective edge of a pulse to be measured is input, the value of the counter goes to “0000_{16}”, and measurement is started. In this instance, an indeterminate value is transferred to the reload register. The timer Bi interrupt request does not generate.

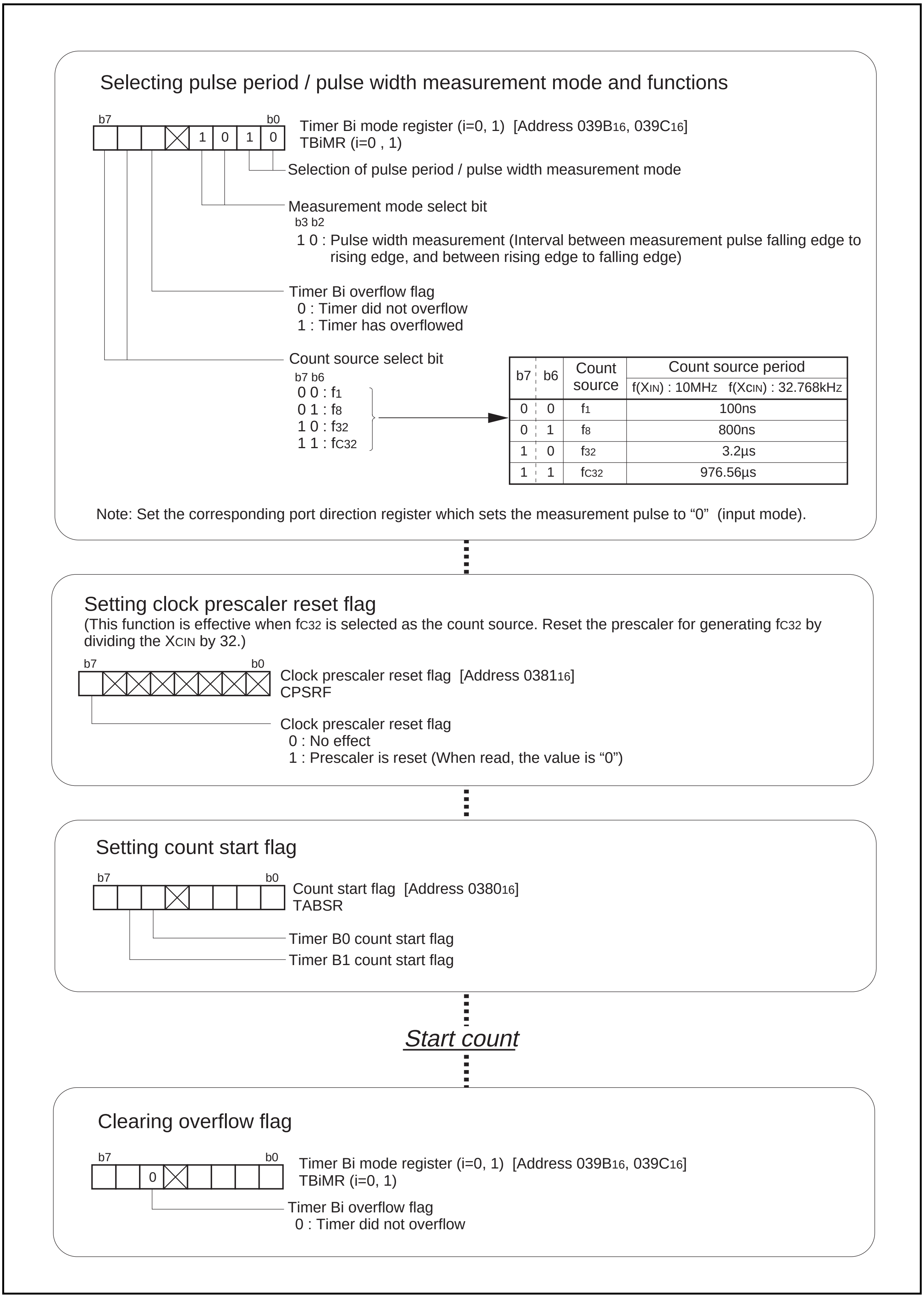
(3) If an effective edge of a pulse to be measured is input again, the value of the counter is transferred to the reload register, and the timer Bi interrupt request bit goes to “1”. Then the value of the counter becomes “0000_{16}”, and measurement is started again.

Note
- The timer Bi interrupt request bit goes to “1” when an effective edge of a pulse to be measured is input or timer Bi is overflows. The factor of interrupt request can be determined by use of the timer Bi overflow flag within the interrupt routine.
- The value of the counter at the beginning of a count is indeterminate. Thus there can be instances in which the timer Bi overflow flag goes to “1” immediately after a count is performed.
- The timer Bi overflow flag goes to “0” if timer Bi mode register is written to when the count start flag is “1”. This flag cannot be set to “1” by software.


(1) Start count
(3) Start measurement again
(2) Start measurement
Count source
Measurement pulse
“H”
“L”
Transfer (indeterminate value)
Transfer(measured value)
Reload register ‹ counter transfer timing
(Note 1)
(Note 1)
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches “0000₁₆”
Count start flag
“1”
“0”
Timer Bi interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Bi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 2.3.10. Operation timing of pulse width measurement mode



Selecting pulse period / pulse width measurement mode and functions

Timer Bi mode register (i=0, 1) [Address 039B$_{16}$, 039C$_{16}$]
TBiMR (i=0 , 1)

Selection of pulse period / pulse width measurement mode

Measurement mode select bit
b3 b2
1 0 : Pulse width measurement (Interval between measurement pulse falling edge to rising edge, and between rising edge to falling edge)

Timer Bi overflow flag
0 : Timer did not overflow
1 : Timer has overflowed

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Note: Set the corresponding port direction register which sets the measurement pulse to “0” (input mode).

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 0381$_{16}$]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380$_{16}$]
TABSR

Timer B0 count start flag
Timer B1 count start flag

Start count

Clearing overflow flag

Timer Bi mode register (i=0, 1) [Address 039B$_{16}$, 039C$_{16}$]
TBiMR (i=0, 1)

Timer Bi overflow flag
0 : Timer did not overflow

Figure 2.3.11. Set-up procedure of pulse width measurement mode

2.3.6 Precautions for Timer B (timer mode, event counter mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Bi register, then set the flag to “1”.

(2) Reading the timer Bi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing shown in Figure 2.3.12 gets “FFFF$_{16}$”. Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

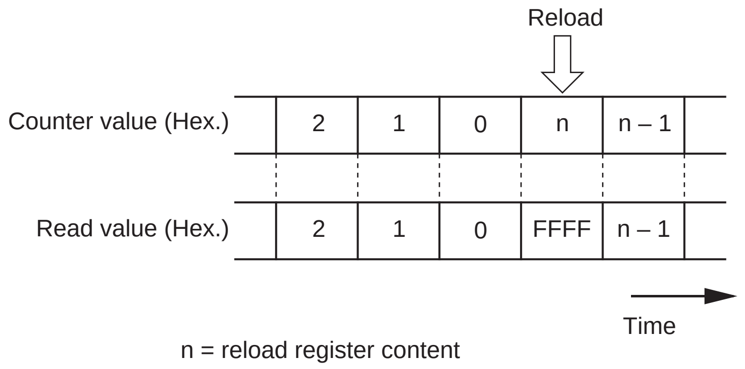


Figure 2.3.12. Reading timer Bi register

2.3.7 Precautions for Timer B (pulse period/pulse width measurement mode)

(1) The timer Bi interrupt request bit goes to “1” when an effective edge of a measurement pulse is input or timer Bi is overflowed. The factor of interrupt request can be determined by use of the timer Bi overflow flag within the interrupt routine.

(2) If the timer overflow occurs simultaneously with the input of a measurement pulse, and if the interrupt factor cannot be determined from the timer Bi overflow flag, connect the timers and count the number of overflows.

(3) When reset, the timer Bi overflow flag goes to “1”. This flag cannot be set to “0” by writing to the timer Bi mode register when the count start flag is “1”.

(4) Use the timer Bi interrupt request bit to detect only overflows. Use the timer Bi overflow flag only to determine the interrupt factor within the interrupt routine.

(5) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

(6) The value of the counter is indeterminate at the beginning of a count. Therefore the timer Bi overflow flag may go to “1” immediately after a count is started.

(7) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to “1”.

(8) If the input signal to the TBiIN pin is affected by noise, precise measurement may not be performed in some cases. It is recommended to see that measurements fall within a specific range by use of software.

(9) For pulse width measurement, pulse widths are successively measured. Use software to check whether the measurement result is an “H” level width or an “L” level width.

Under development

2.4 Timer X

2.4.1 Overview

The following is an overview for timer X, a 16-bit timer.

(1) Mode

Timer X operates in one of the four modes:

(a) Timer mode

In this mode, the internal count source is counted. Two functions can be selected: the pulse output function that reverses output from a port every time an overflow occurs, or the gate function which controls the count start/stop according to the input signal from a port.

- Timer mode operation P212
- Timer mode, gate function operation P214
- Timer mode, pulse output function operation P216

(b) Event counter mode

This mode counts the pulses from the outside and the number of overflows in other timers. The free-run type, in which nothing is reloaded from the reload register, can be selected when an underflow occurs. The pulse output function can also be selected.

- Event counter mode operation P218
- Event counter mode, free run type operation P220

(c) One-shot timer mode

In this mode, the timer is started by the trigger and stops when the timer goes to “0”. The trigger can be selected from the following 3 types: an external input signal, an overflow of the timer, or a software trigger.

- One-shot timer mode operation P222

(d) Pulse period measurement/pulse width measurement mode

External pulse period or external pulse widths are measured. If pulse period measurement mode is selected, the periods of input pulses are continuously measured. If pulse width measurement mode is selected, widths of “H” level pulses and those of “L” level pulses are continuously measured.

- Operation in pulse period measurement mode P224
- Operation in pulse width measurement mode P226

(d) Pulse width modulation (PWM) mode

In this mode, the arbitrary pulses are successively output. Either a 16-bit fixed-period PWM mode or 8-bit variable-period mode can be selected. The trigger for initiating output can also be selected.

- 16-bit PWM mode operation P228
- 8-bit PWM mode operation P230

(2) Count source

The internal count source can be selected from f1, f_8, f_{32}, and f_{C32}. Clocks f_1, f_8, and f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively. Clock f_{C32} is derived by dividing the CPU's secondary clock by 32.

Under development

(3) Frequency division ratio

In timer mode or pulse width modulation mode, [the value set in the timer register + 1] becomes the frequency division ratio. In event counter mode, [the set value + 1] becomes the frequency division ratio when a down count is performed, or [$FFFF_{16}$ - the set value + 1] becomes the frequency division ratio when an up count is performed. In one-shot timer mode, the value set in the timer register becomes the frequency division ratio.

The counter overflows (or underflows) when a count source equal to a frequency division ratio is input, and an interrupt occurs. For the pulse output function, the output from the port varies (the value in the port register does not vary).

(4) Reading the timer

Either in timer mode or in event counter mode, reading the timer register takes out the count at that moment. Read it in 16-bit units. The data either in one-shot timer mode or in pulse width modulation mode is indeterminate. In both the pulse period measurement mode and pulse width measurement mode, an indeterminate value is read until the second effective edge is input after a count is started, otherwise, the measurement results are read.

(5) Writing to the timer

When writing to the timer register while a count is in progress, the value is written only to the reload register. When writing to the timer register while a count has stopped, the value is written both to the reload register and the count. Write the value in 16-bit increments. The timer register cannot be written to in either the pulse period measurement mode or the pulse width measurement mode.

(6) Relation between the input/output to/from the timer and the direction register

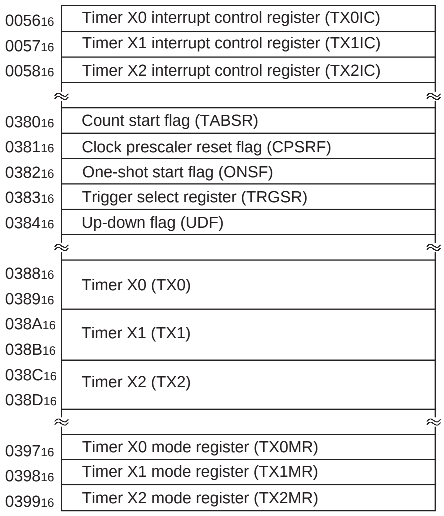
With the output function of the timer, set the direction register of the relevant port to input. To input an external signal to the timer, set the direction register of the relevant port to input. However, pulse output cannot be selected when inputting an external signal to the timer, and vice-versa.

(7) Pins related to timer X

(a) $TX0_{INOUT}$, $TX1_{INOUT}$, $TX2_{INOUT}$ Input/output pins to timer X.

(8) Registers related to timer X

Figure 2.4.1 shows the memory map of timer X-related registers. Figures 2.4.2 and 2.4.3 show timer X-related registers.

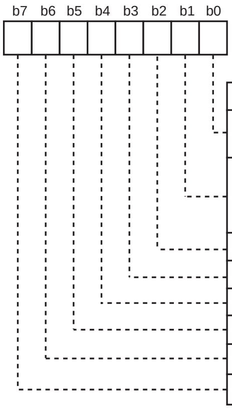


0056_{16}	Timer X0 interrupt control register (TX0IC)
0057_{16}	Timer X1 interrupt control register (TX1IC)
0058_{16}	Timer X2 interrupt control register (TX2IC)
≈	≈
0380_{16}	Count start flag (TABSR)
0381_{16}	Clock prescaler reset flag (CPSRF)
0382_{16}	One-shot start flag (ONSF)
0383_{16}	Trigger select register (TRGSR)
0384_{16}	Up-down flag (UDF)
≈	≈
0388_{16} 0389_{16}	Timer X0 (TX0)
$038A_{16}$ $038B_{16}$	Timer X1 (TX1)
$038C_{16}$ $038D_{16}$	Timer X2 (TX2)
≈	≈
0397_{16}	Timer X0 mode register (TX0MR)
0398_{16}	Timer X1 mode register (TX1MR)
0399_{16}	Timer X2 mode register (TX2MR)

Figure 2.4.1. Memory map of timer X-related registers

Under development

Timer Xi mode register

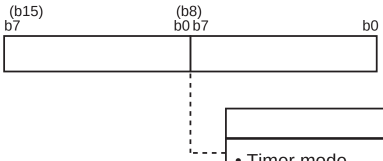


Symbol TXiMR(i = 0 to 2)
Address 0397$_{16}$ to 0399$_{16}$
When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode or pulse period/ pulse width measurement mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Must set "00" to operation mode select bit when using timer X2 of M30200.

Timer Xi register (Note)

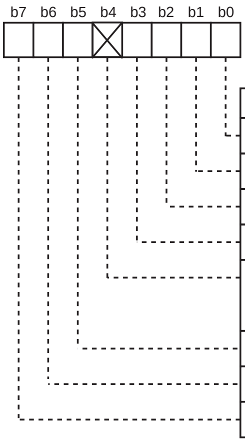


Symbol	Address	When reset
TX0	0389$_{16}$,0388$_{16}$	Indeterminate
TX1	038B$_{16}$,038A$_{16}$	Indeterminate
TX2	038D$_{16}$,038C$_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000$_{16}$ to FFFF$_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000$_{16}$ to FFFF$_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000$_{16}$ to FFFF$_{16}$	×	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	———	○	×
• Pulse width modulation mode (16-bit PWM) Functions as a 16-bit pulse width modulator	0000$_{16}$ to FFFE$_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00$_{16}$ to FF$_{16}$ (High-order addresses) 00$_{16}$ to FF$_{16}$ (Low-order addresses)	×	○

Note: Read and write data in 16-bit units.

Count start flag



Symbol TABSR
Address 0380$_{16}$
When reset 000X0000$_2$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TX0S	Timer X0 count start flag		○	○
TX1S	Timer X1 count start flag		○	○
TX2S	Timer X2 count start flag		○	○
Nothing is assigned. When write, set "0" When read, their contents are indeterminate.			—	—
TB0S	Timer B0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB1S	Timer B1 count start flag		○	○
CDCS	Clock devided count start flag		○	○

Figure 2.4.2. Timer X-related registers (1)

One-shot start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
ONSF	0382_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TX0OS	Timer X0 one-shot start flag		○	○
TX1OS	Timer X1 one-shot start flag		○	○
TX2OS	Timer X2 one-shot start flag		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Trigger select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TRGSR	0383_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Input on TA0$_{IN}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX2 overflow is selected 1 1 : TX0 overflow is selected	○	○
TA0TGH			○	○
TX0TGL	Timer X0 event/trigger select bit	b3 b2 0 0 : Input on TX0$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TX1 overflow is selected	○	○
TX0TGH			○	○
TX1TGL	Timer X1 event/trigger select bit	b5 b4 0 0 : Input on TX1$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX0 overflow is selected 1 1 : TX2 overflow is selected	○	○
TX1TGH			○	○
TX2TGL	Timer X2 event/trigger select bit	b7 b6 0 0 : Input on TX2$_{INOUT}$ is selected (Note) 0 1 : TB1 overflow is selected 1 0 : TX1 overflow is selected 1 1 : TA0 overflow is selected	○	○
TX2TGH			○	○

Note: Set the corresponding port direction register to “0”(input mode).

Clock prescaler reset flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPSRF	0381_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 2.4.3. Timer X-related registers (2)

2.4.2 Operation of Timer X (timer mode)

In timer mode, choose functions from those listed in Table 2.4.1. Operations of the circled items are described below. Figure 2.4.4 shows the operation timing, and Figure 2.4.5 shows the set-up procedure.

Table 2.4.1. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Pulse output function	O	No pulses output
		Pulses output
Gate function	O	No gate function
		Performs count only for the period in which the TXiINOUT pin is at "L" level
		Performs count only for the period in which the TXiINOUT pin is at "H" level

Operation (1) Setting the count start flag to “1” causes the counter to perform a down count on the count source.

(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer Xi interrupt request bit goes to “1”.

(3) Setting the count start flag to “0” causes the counter to hold its value and to stop.

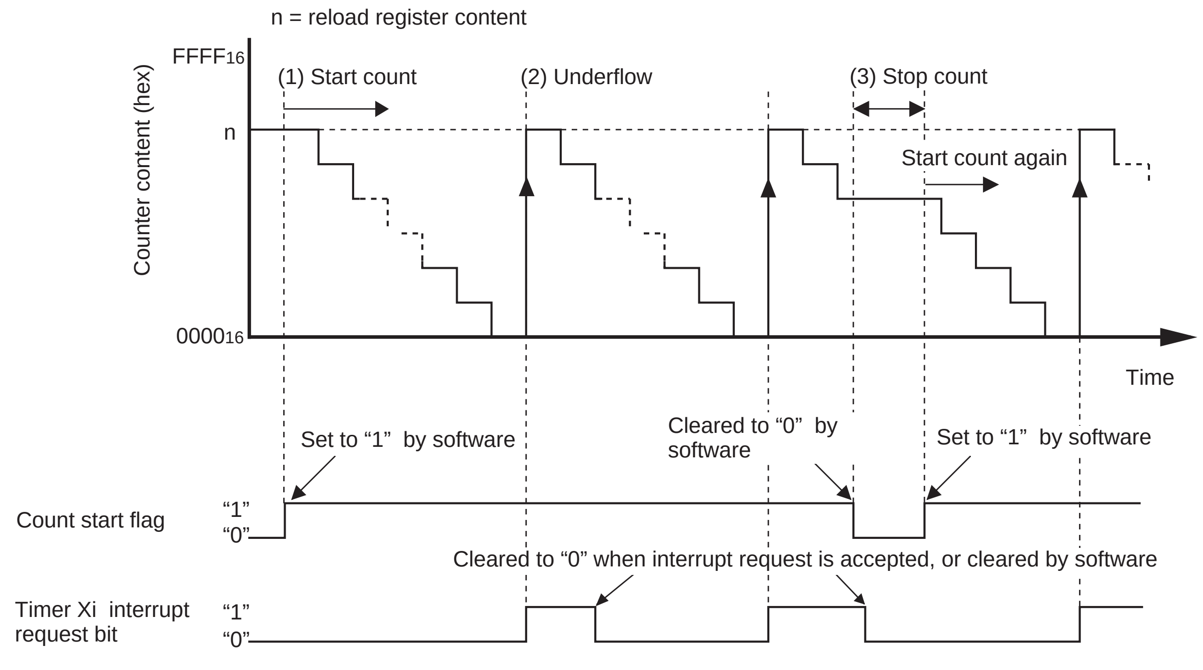
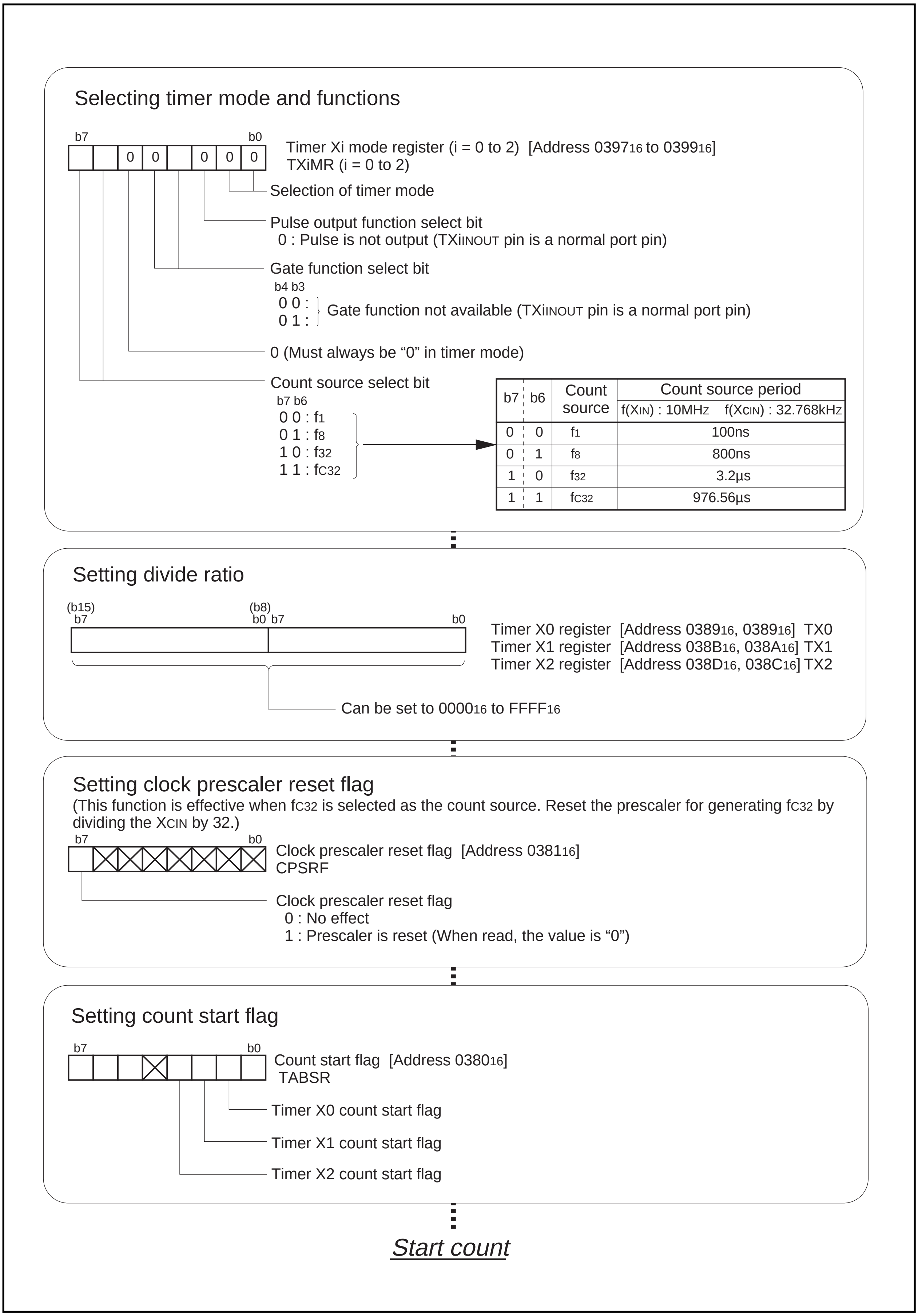


Figure 2.4.4. Operation timing of timer mode

Under development



Selecting timer mode and functions

Timer Xi mode register (i = 0 to 2) [Address 0397$_{16}$ to 0399$_{16}$]
TXiMR (i = 0 to 2)

Selection of timer mode

Pulse output function select bit
0 : Pulse is not output (TXiINOUT pin is a normal port pin)

Gate function select bit
b4 b3
0 0 :
0 1 : Gate function not available (TXiINOUT pin is a normal port pin)

0 (Must always be “0” in timer mode)

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period f(XIN) : 10MHz	 f(XCIN) : 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Setting divide ratio

Timer X0 register [Address 0389$_{16}$, 0389$_{16}$] TX0
Timer X1 register [Address 038B$_{16}$, 038A$_{16}$] TX1
Timer X2 register [Address 038D$_{16}$, 038C$_{16}$] TX2

Can be set to 0000$_{16}$ to FFFF$_{16}$

Setting clock prescaler reset flag

(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the XCIN by 32.)

Clock prescaler reset flag [Address 0381$_{16}$]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380$_{16}$]
TABSR

Timer X0 count start flag

Timer X1 count start flag

Timer X2 count start flag

Start count

Figure 2.4.5. Set-up procedure of timer mode

Under development

2.4.3 Operation of Timer X (timer mode, gate function selected)

In timer mode, choose functions from those listed in Table 2.4.2. Operations of the circled items are described below. Figure 2.4.6 shows the operation timing, and Figure 2.4.7 shows the set-up procedure.

Table 2.4.2. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / f_{C32})
Pulse output function	O	No pulses output
		Pulses output
Gate function		No gate function
		Performs count only for the period in which the TXiINOUT pin is at "L" level
	O	Performs count only for the period in which the TXiINOUT pin is at "H" level

Operation (1) When the count start flag is set to "1" and the TXiINOUT pin inputs at "H" level, the counter performs a down count on the count source.

(2) When the TXiINOUT pin inputs at "L" level, the counter holds its value and stops.

(3) If an underflow occurs, the content of the reload register is reloaded and the count continues. At this time, the timer Xi interrupt request bit goes to "1".

(4) Setting the count start flag to "0" causes the counter to hold its value and to stop.

Note • Make the pulse width of the signal input to the TXiINOUT pin not less than two cycles of the count source.

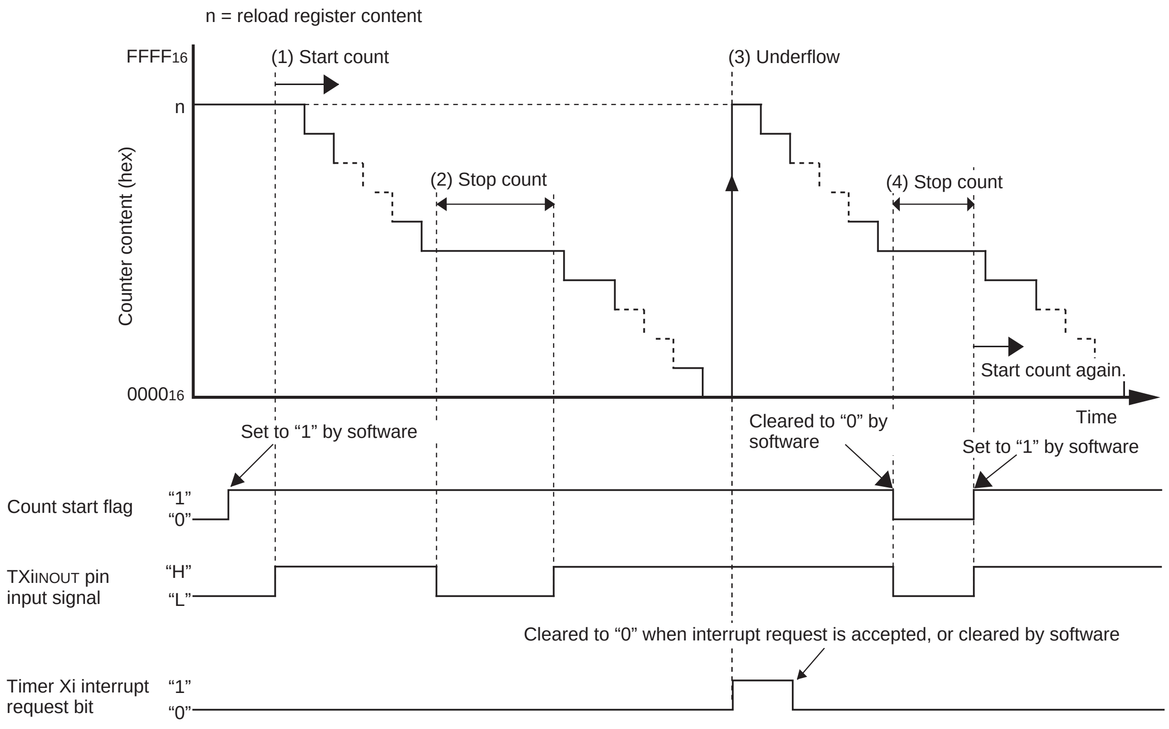
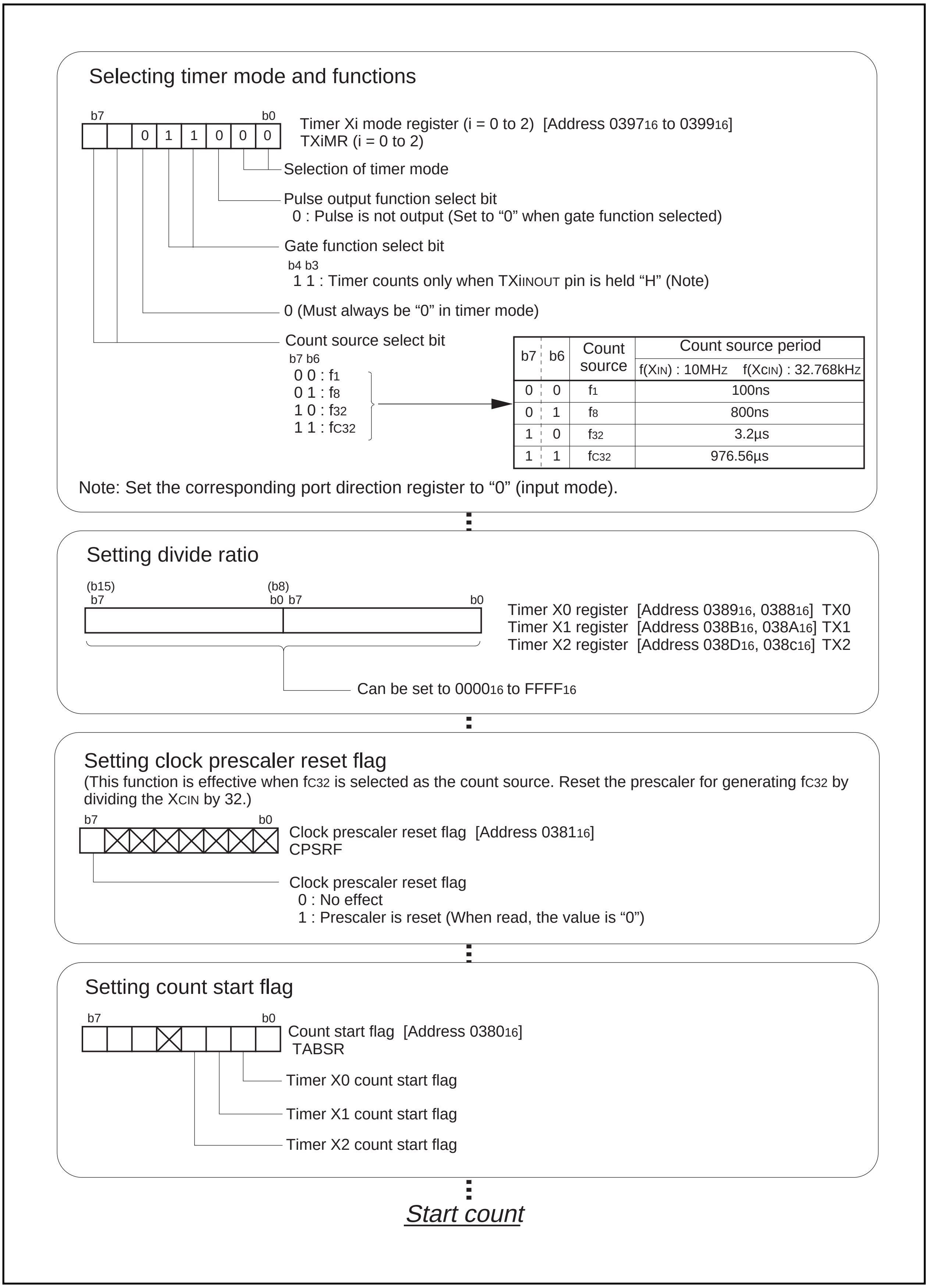


Figure 2.4.6. Operation timing of timer mode, gate function selected



Selecting timer mode and functions

Timer Xi mode register (i = 0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i = 0 to 2)

Selection of timer mode

Pulse output function select bit
0 : Pulse is not output (Set to “0” when gate function selected)

Gate function select bit
b4 b3
1 1 : Timer counts only when TXi_{INOUT} pin is held “H” (Note)

0 (Must always be “0” in timer mode)

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period	
			$f(X_{IN})$: 10MHz	$f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Note: Set the corresponding port direction register to “0” (input mode).

Setting divide ratio

Timer X0 register [Address 0389_{16}, 0388_{16}] TX0
Timer X1 register [Address $038B_{16}$, $038A_{16}$] TX1
Timer X2 register [Address $038D_{16}$, $038C_{16}$] TX2

Can be set to 0000_{16} to $FFFF_{16}$

Setting clock prescaler reset flag

(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380_{16}]
TABSR

Timer X0 count start flag

Timer X1 count start flag

Timer X2 count start flag

Start count

Figure 2.4.7. Set-up procedure of timer mode, gate function selected

2.4.4 Operation of Timer X (timer mode, pulse output function selected)

In timer mode, choose functions from those listed in Table 2.4.3. Operations of the circled items are described below. Figure 2.4.8 shows the operation timing, and Figure 2.4.9 shows the set-up procedure.

Table 2.4.3. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Pulse output function		No pulses output
	O	Pulses output
Gate function	O	No gate function
		Performs count only for the period in which the TXiINOUT pin is at "L" level
		Performs count only for the period in which the TXiINOUT pin is at "H" level

Operation
(1) Setting the count start flag to "1" causes the counter to perform a down count on the count source.
(2) If an underflow occurs, the content of the reload register is reloaded and the count continues. At this time, the timer Xi interrupt request bit goes to "1". Also, the output polarity of the TXiINOUT pin reverses.
(3) Setting the count start flag to "0" causes the counter to hold its value and to stop. Also, the TXiINOUT pin outputs an "L" level.

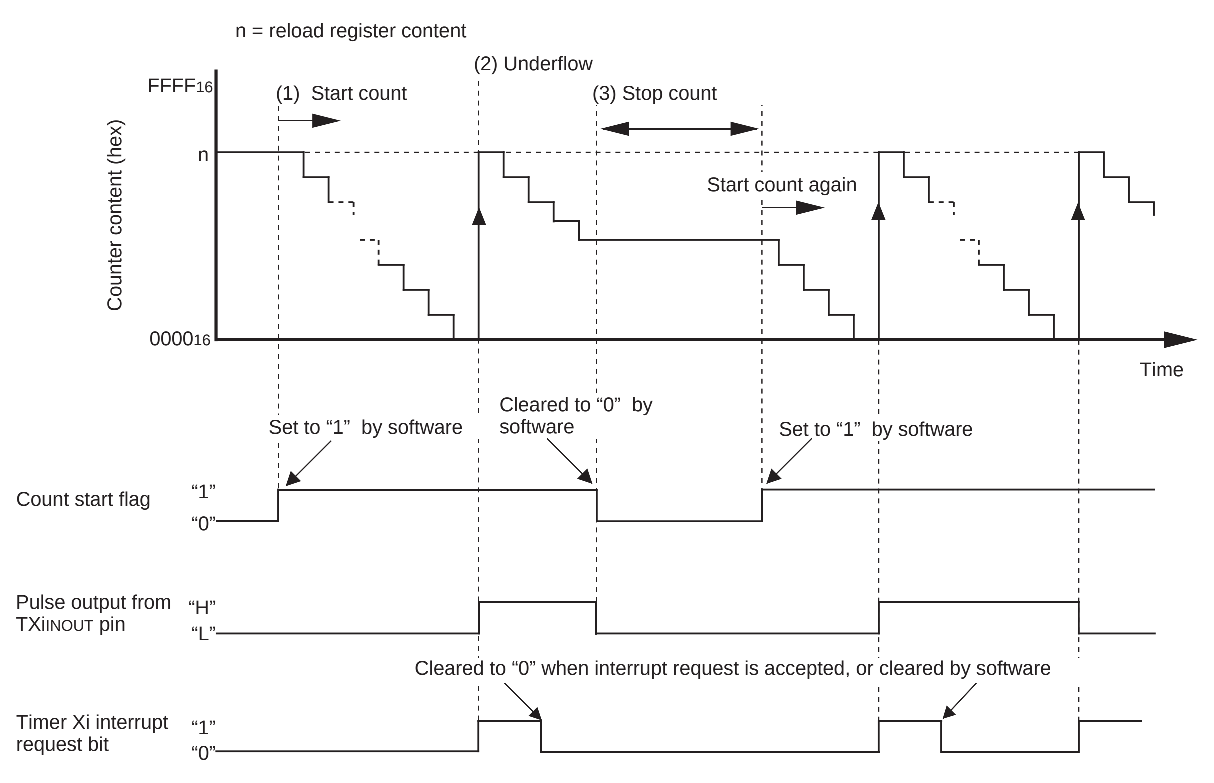
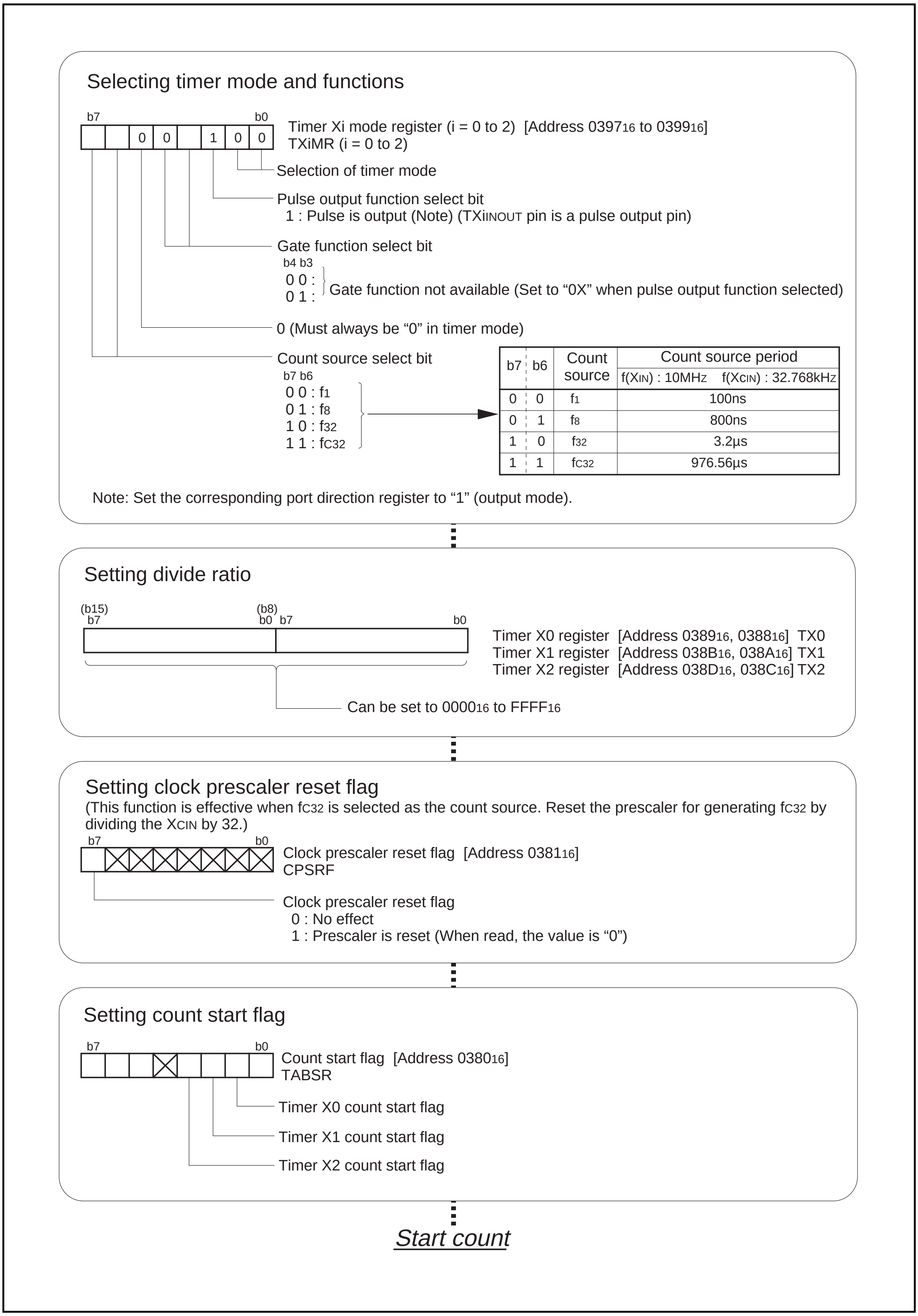

n = reload register content
(2) Underflow
FFFF16
(1) Start count
(3) Stop count
n
Start count again
Counter content (hex)
000016
Time
Cleared to "0" by software
Set to "1" by software
Set to "1" by software
Count start flag
"1"
"0"
Pulse output from TXiINOUT pin
"H"
"L"
Cleared to "0" when interrupt request is accepted, or cleared by software
Timer Xi interrupt request bit
"1"
"0"

Figure 2.4.8. Operation timing of timer mode, pulse output function selected



Selecting timer mode and functions

Timer Xi mode register (i = 0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i = 0 to 2)

Selection of timer mode

Pulse output function select bit
1 : Pulse is output (Note) (TXiINOUT pin is a pulse output pin)

Gate function select bit
b4 b3
0 0 :
0 1 :
Gate function not available (Set to “0X” when pulse output function selected)

0 (Must always be “0” in timer mode)

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period	
			f(XIN) : 10MHz	f(XCIN) : 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}	976.56µs	

Note: Set the corresponding port direction register to “1” (output mode).

Setting divide ratio

Timer X0 register [Address 0389_{16}, 0388_{16}] TX0
Timer X1 register [Address $038B_{16}$, $038A_{16}$] TX1
Timer X2 register [Address $038D_{16}$, $038C_{16}$] TX2

Can be set to 0000_{16} to $FFFF_{16}$

Setting clock prescaler reset flag

(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the XCIN by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380_{16}]
TABSR

Timer X0 count start flag

Timer X1 count start flag

Timer X2 count start flag

Start count

Figure 2.4.9. Set-up procedure of timer mode, pulse output function selected

Under development

2.4.5 Operation of Timer X (event counter mode, reload type selected)

In event counter mode, choose functions from those listed in Table 2.4.4. Operations of the circled items are described below. Figure 2.4.10 shows the operation timing, and Figure 2.4.11 shows the set-up procedure.

Table 2.4.4. Choosed functions

Item	Set-up	
Count source		Input signal to TXi$_{\text{INOUT}}$(counting falling edges)
		Input signal to TXi$_{\text{INOUT}}$(counting rising edges)
	O	Timer overflow(TB1/TA0/TXi overflow)
Pulse output function		No pulses output
	O	Pulses output
Count operation type	O	Reload type
		Free-run type

Operation (1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.

(2) If an underflow occurs, the content of the reload register is reloaded, and the count continues. At this time, the timer Xi interrupt request bit goes to “1”.

(3) Setting the count start flag to “0” causes the counter to hold its value and to stop.

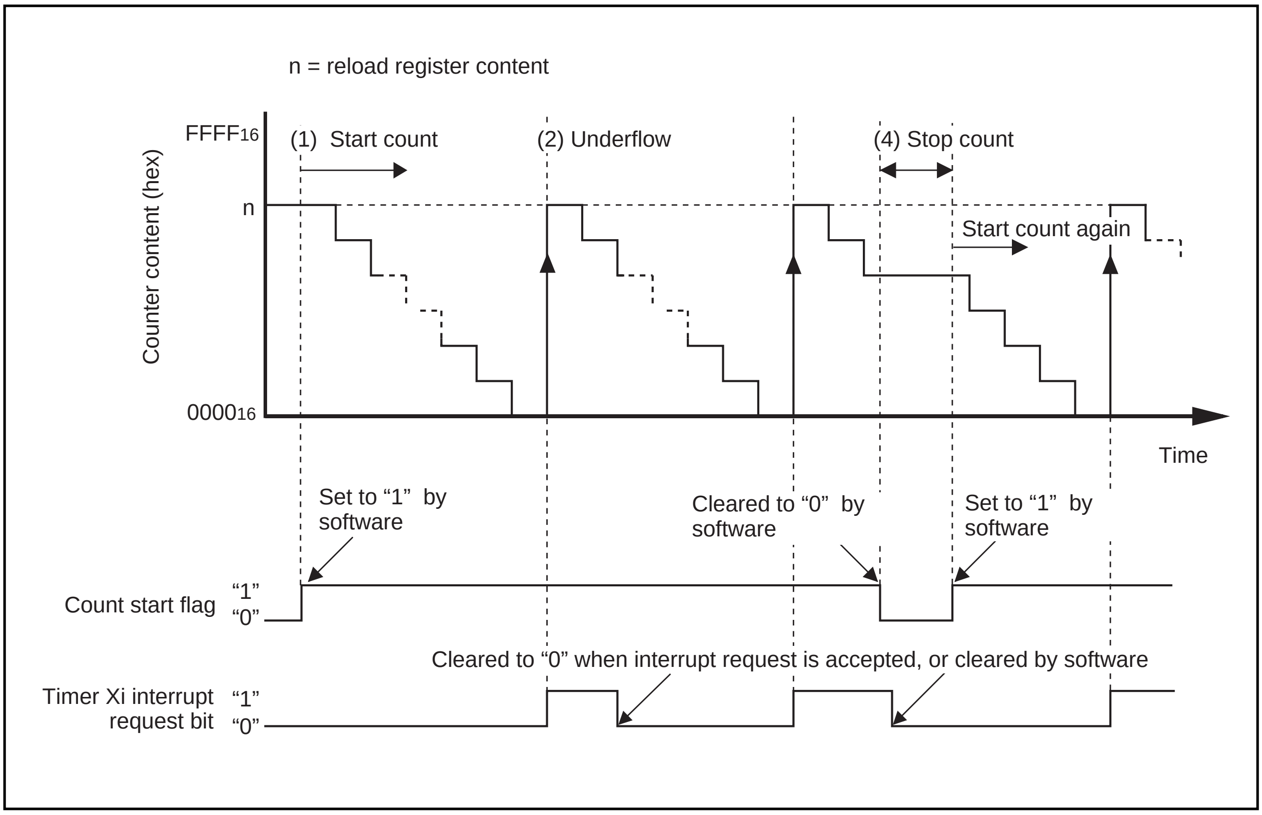


Figure 2.4.10. Operation timing of event counter mode, reload type selected

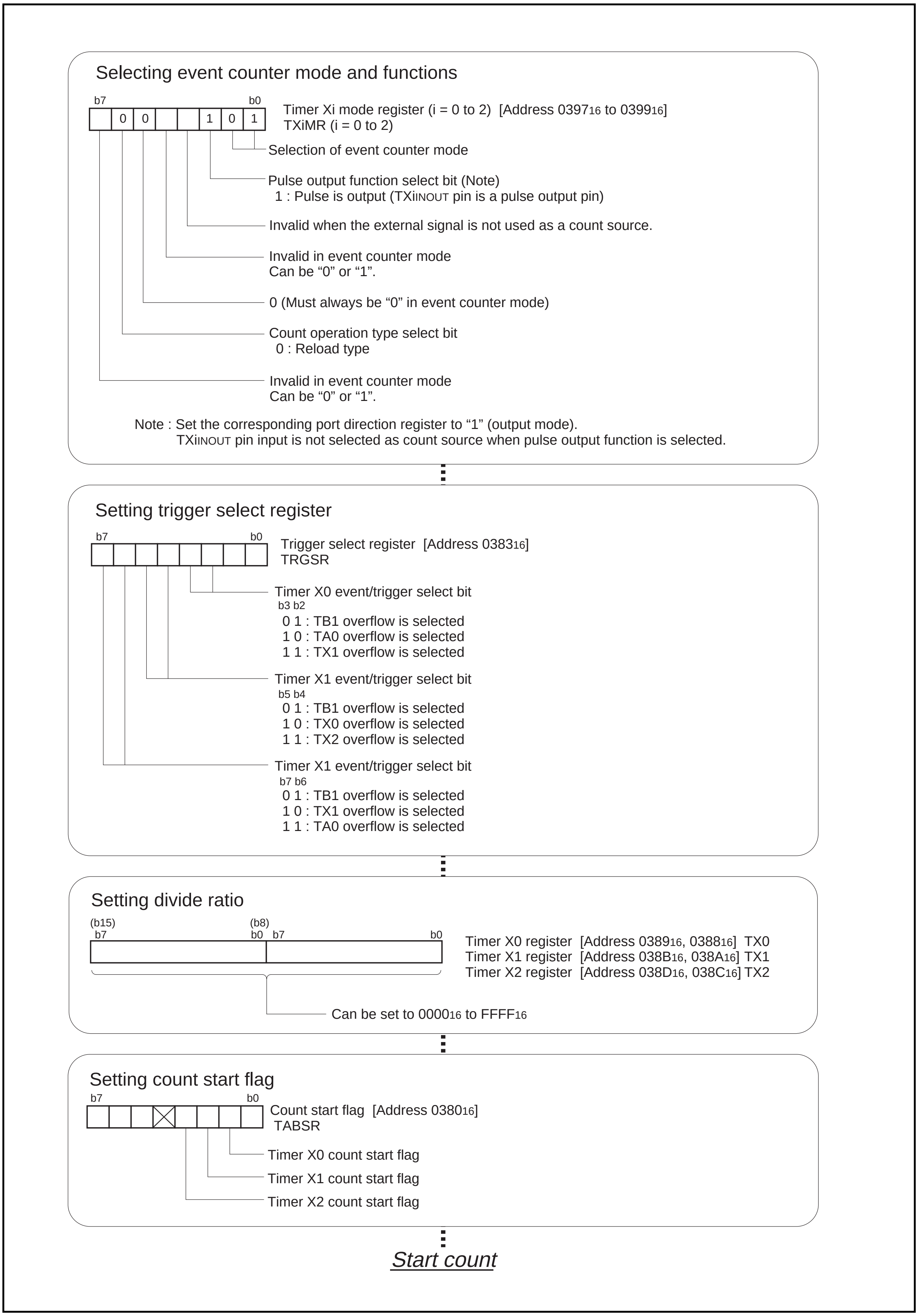

Selecting event counter mode and functions
b7
b0
0 0 1 0 1
Timer Xi mode register (i = 0 to 2) [Address 0397₁₆ to 0399₁₆]
TXiMR (i = 0 to 2)
Selection of event counter mode
Pulse output function select bit (Note)
1 : Pulse is output (TXiINOUT pin is a pulse output pin)
Invalid when the external signal is not used as a count source.
Invalid in event counter mode
Can be “0” or “1”.
0 (Must always be “0” in event counter mode)
Count operation type select bit
0 : Reload type
Invalid in event counter mode
Can be “0” or “1”.
Note : Set the corresponding port direction register to “1” (output mode).
TXiINOUT pin input is not selected as count source when pulse output function is selected.
Setting trigger select register
b7
b0
Trigger select register [Address 0383₁₆]
TRGSR
Timer X0 event/trigger select bit
b3 b2
0 1 : TB1 overflow is selected
1 0 : TA0 overflow is selected
1 1 : TX1 overflow is selected
Timer X1 event/trigger select bit
b5 b4
0 1 : TB1 overflow is selected
1 0 : TX0 overflow is selected
1 1 : TX2 overflow is selected
Timer X1 event/trigger select bit
b7 b6
0 1 : TB1 overflow is selected
1 0 : TX1 overflow is selected
1 1 : TA0 overflow is selected
Setting divide ratio
(b15)
b7
(b8)
b0
b7
b0
Timer X0 register [Address 0389₁₆, 0388₁₆] TX0
Timer X1 register [Address 038B₁₆, 038A₁₆] TX1
Timer X2 register [Address 038D₁₆, 038C₁₆] TX2
Can be set to 0000₁₆ to FFFF₁₆
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag
Start count

Figure 2.4.11. Set-up procedure of event counter mode, reload type selected

2.4.6 Operation of Timer X (event counter mode, free run type selected)

In event counter mode, choose functions from those listed in Table 2.4.5. Operations of the circled items are described below. Figure 2.4.12 shows the operation timing, and Figure 2.4.13 shows the set-up procedure.

Table 2.4.5. Choosed functions

Item	Set-up	
Count source	**O**	Input signal to TXi$_{INOUT}$(counting falling edges)
		Input signal to TXi$_{INOUT}$(counting rising edges)
		Timer overflow(TB1/TA0/TXi overflow)
Pulse output function	**O**	No pulses output
		Pulses output
Count operation type		Reload type
	O	Free-run type

Operation
(1) Setting the count start flag to “1” causes the counter to count the falling edges of the count source.
(2) Even if an underflow occurs, the content of the reload register is not reloaded, but the count continues. At this time, the timer Xi interrupt request bit goes to “1”.
(3) Setting the count start flag to “0” causes the counter to hold its value and to stop.

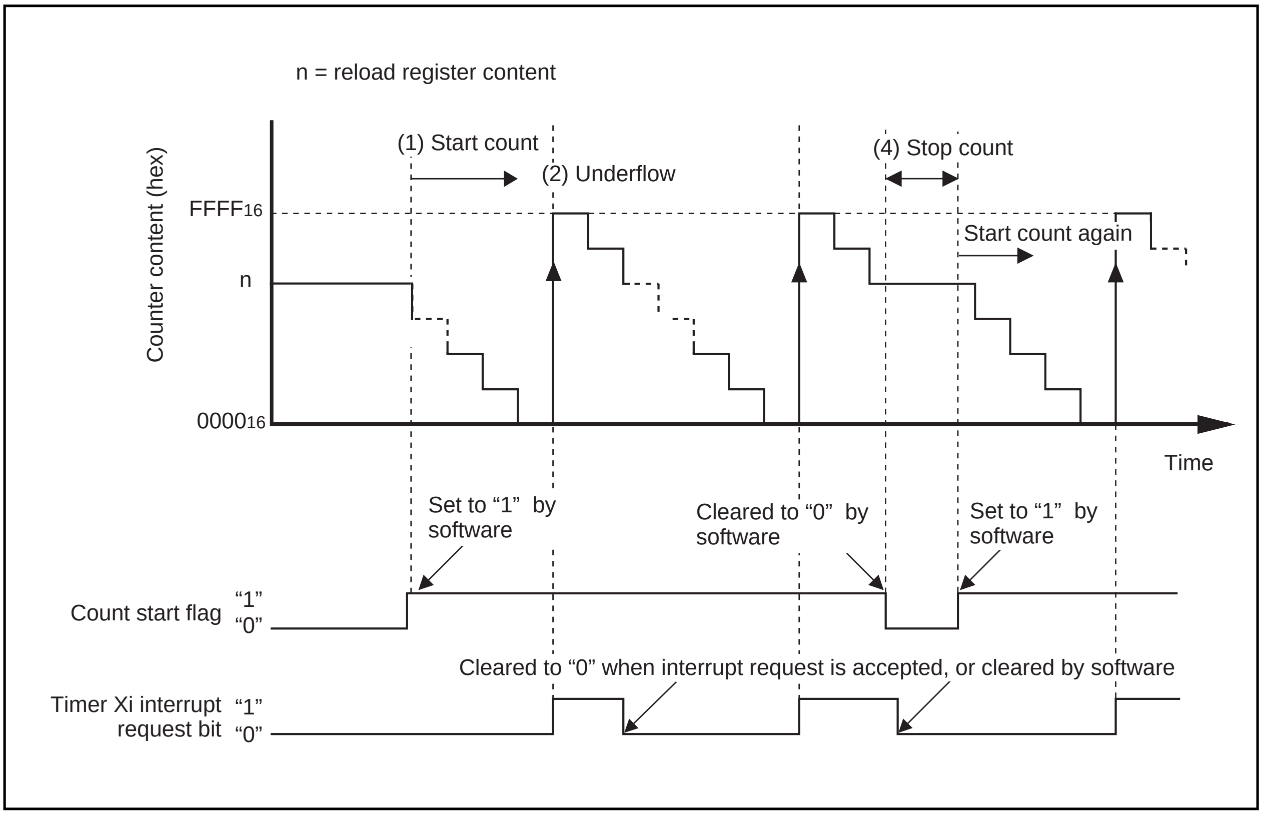


Figure 2.4.12. Operation timing of event counter mode, free run type selected

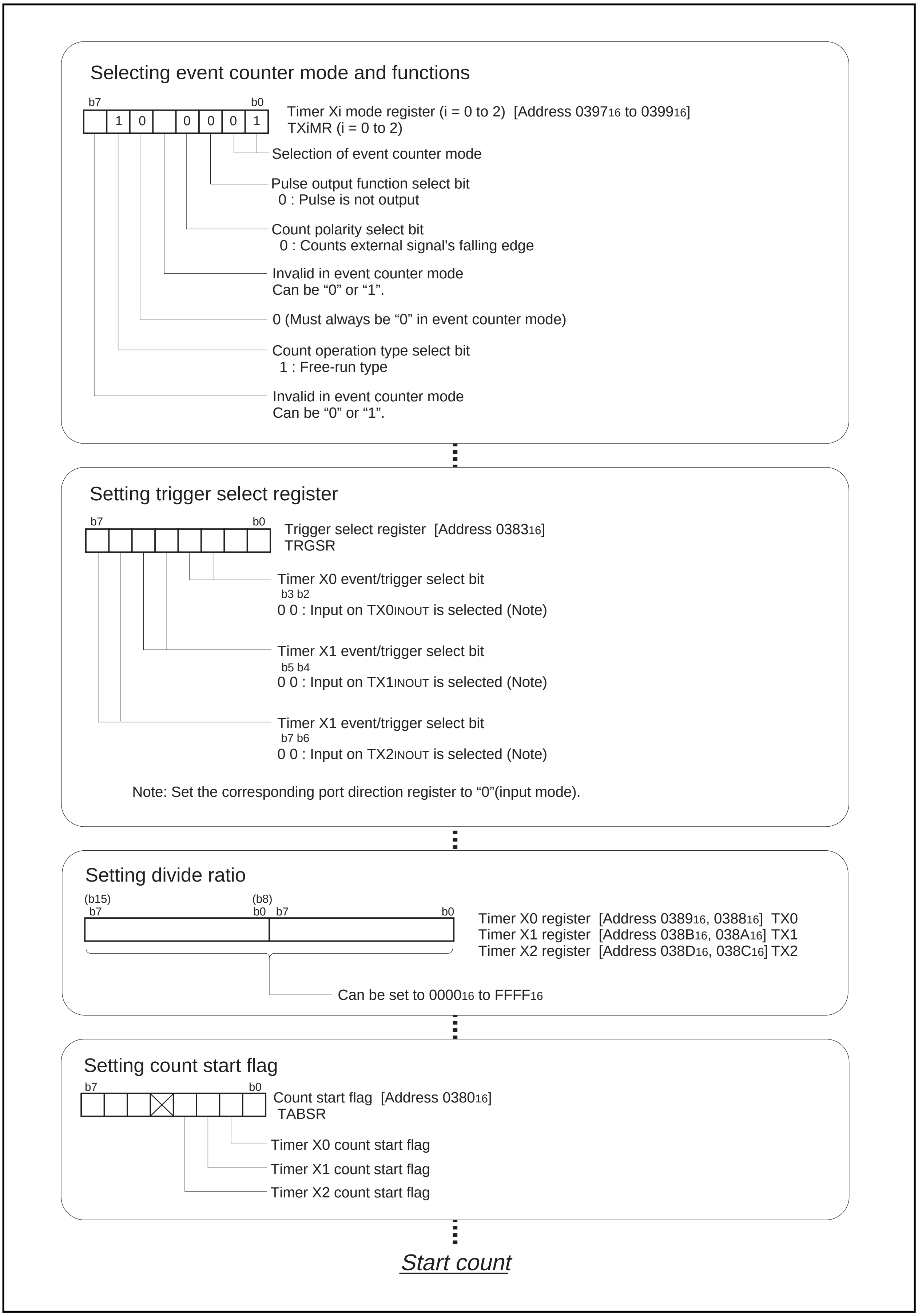

Selecting event counter mode and functions
b7
b0
1 0 0 0 0 1
Timer Xi mode register (i = 0 to 2) [Address 0397₁₆ to 0399₁₆]
TXiMR (i = 0 to 2)
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output
Count polarity select bit
0 : Counts external signal's falling edge
Invalid in event counter mode
Can be "0" or "1".
0 (Must always be "0" in event counter mode)
Count operation type select bit
1 : Free-run type
Invalid in event counter mode
Can be "0" or "1".
Setting trigger select register
b7
b0
Trigger select register [Address 0383₁₆]
TRGSR
Timer X0 event/trigger select bit
b3 b2
0 0 : Input on TX0INOUT is selected (Note)
Timer X1 event/trigger select bit
b5 b4
0 0 : Input on TX1INOUT is selected (Note)
Timer X1 event/trigger select bit
b7 b6
0 0 : Input on TX2INOUT is selected (Note)
Note: Set the corresponding port direction register to "0"(input mode).
Setting divide ratio
(b15)
b7
(b8)
b0
b7
b0
Timer X0 register [Address 0389₁₆, 0388₁₆] TX0
Timer X1 register [Address 038B₁₆, 038A₁₆] TX1
Timer X2 register [Address 038D₁₆, 038C₁₆] TX2
Can be set to 0000₁₆ to FFFF₁₆
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag
Start count

Figure 2.4.13. Set-up procedure of event counter mode, free run type selected

2.4.7 Operation of Timer X (one-shot timer mode)

In one-shot timer mode, choose functions from those listed in Table 2.4.6. Operations of the circled items are described below. Figure 2.4.14 shows the operation timing, and Figure 2.4.15 shows the set-up procedure.

Table 2.4.6. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / f_{C32})
Pulse output function		No pulses output
	O	Pulses output
Count start condition		External trigger input (falling edge of input signal to the TXi_{INOUT} pin)
		External trigger input (rising edge of input signal to the TXi_{INOUT} pin)
		Timer overflow (TB1/TX0/TXi overflow)
	O	Writing “1” to the one-shot start flag

Operation
(1) Setting the one-shot start flag to “1” with the count start flag set to “1” causes the counter to perform a down count on the count source. At this time, the TXi_{INOUT} pin outputs an “H” level.
(2) The instant the value of the counter becomes “0000$_{16}$”, the TXi_{INOUT} pin outputs an “L” level, and the counter reloads the content of the reload register and stops counting. At this time, the timer Xi interrupt request bit goes to “1”.
(3) If a trigger occurs while a count is in progress, the counter reloads the value in the reload register again and continues counting. The reload timing is in step with the next count source input after the trigger.
(4) Setting the count start flag to “0” causes the counter to stop and to reload the content of the reload register. Also, the TXi_{INOUT} pin outputs an “L” level. At this time, the timer Xi interrupt request bit goes to “1”.

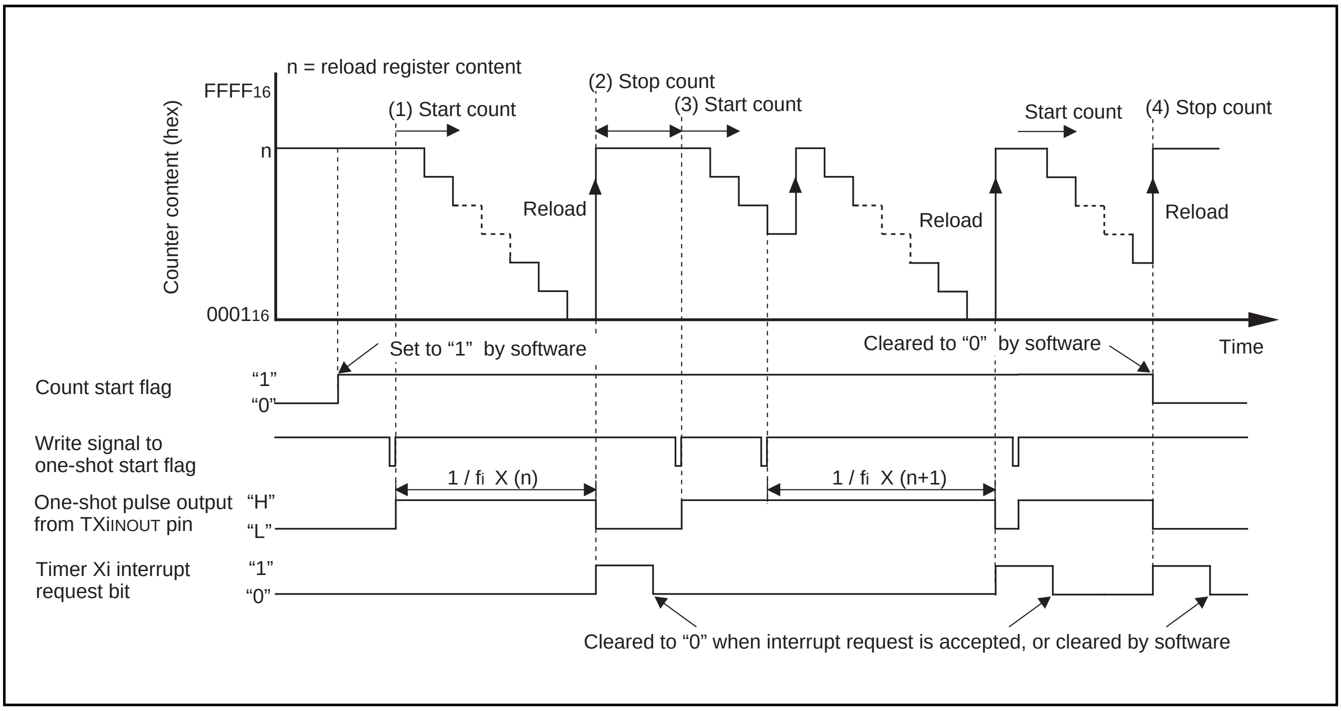
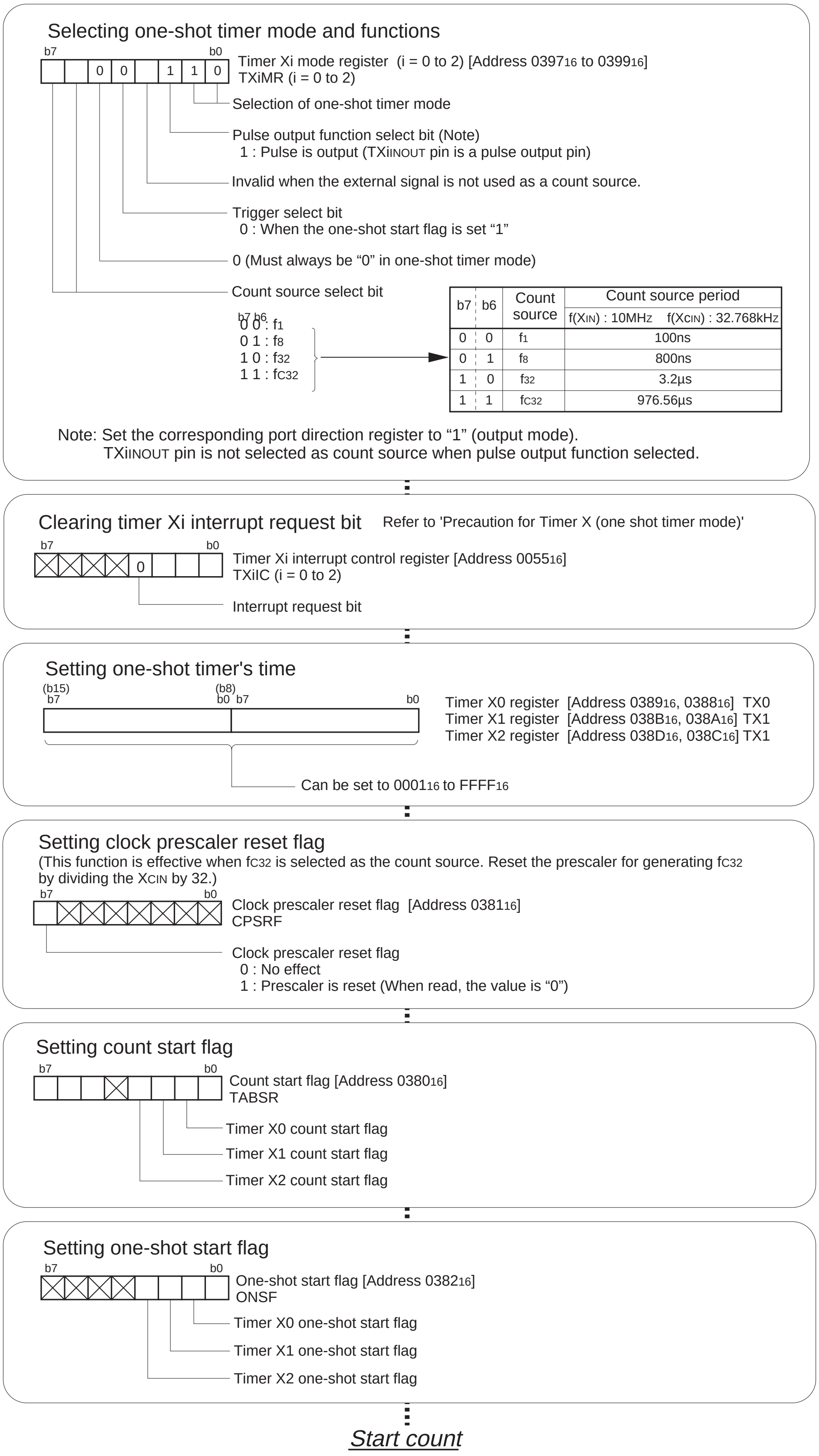

n = reload register content
FFFF16
Counter content (hex)
n
000116
(1) Start count
(2) Stop count
(3) Start count
Start count
(4) Stop count
Reload
Reload
Reload
Time
Set to “1” by software
Cleared to “0” by software
Count start flag
“1”
“0”
Write signal to one-shot start flag
1 / fi X (n)
1 / fi X (n+1)
One-shot pulse output from TXiINOUT pin
“H”
“L”
Timer Xi interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software

Figure 2.4.14. Operation timing of one-shot mode


Selecting one-shot timer mode and functions
b7
b0
0
0
1
1
0
Timer Xi mode register (i = 0 to 2) [Address 0397₁₆ to 0399₁₆]
TXiMR (i = 0 to 2)
Selection of one-shot timer mode
Pulse output function select bit (Note)
1 : Pulse is output (TXiINOUT pin is a pulse output pin)
Invalid when the external signal is not used as a count source.
Trigger select bit
0 : When the one-shot start flag is set “1”
0 (Must always be “0” in one-shot timer mode)
Count source select bit
b7 b6
0 0 : f1
0 1 : f8
1 0 : f32
1 1 : fC32
Note: Set the corresponding port direction register to “1” (output mode).
TXiINOUT pin is not selected as count source when pulse output function selected.
Clearing timer Xi interrupt request bit
Refer to 'Precaution for Timer X (one shot timer mode)'
b7
b0
0
Timer Xi interrupt control register [Address 0055₁₆]
TXiIC (i = 0 to 2)
Interrupt request bit
Setting one-shot timer's time
(b15)
b7
(b8)
b0
b7
b0
Timer X0 register [Address 0389₁₆, 0388₁₆] TX0
Timer X1 register [Address 038B₁₆, 038A₁₆] TX1
Timer X2 register [Address 038D₁₆, 038C₁₆] TX1
Can be set to 0001₁₆ to FFFF₁₆
Setting clock prescaler reset flag
(This function is effective when fC32 is selected as the count source. Reset the prescaler for generating fC32 by dividing the XCIN by 32.)
b7
b0
Clock prescaler reset flag [Address 0381₁₆]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)
Setting count start flag
b7
b0
Count start flag [Address 0380₁₆]
TABSR
Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag
Setting one-shot start flag
b7
b0
One-shot start flag [Address 0382₁₆]
ONSF
Timer X0 one-shot start flag
Timer X1 one-shot start flag
Timer X2 one-shot start flag
Start count

b7	b6	Count source	Count source period	
			f(XIN) : 10MHz	f(XCIN) : 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Figure 2.4.15. Set-up procedure of one-shot mode

Under development

2.4.8 Operation of Timer X (pulse period measurement mode)

In pulse period/pulse width measurement mode, choose functions from those listed in Table 2.4.7. Operations of the circled items are described below. Figure 2.4.16 shows the operation timing, and Figure 2.4.17 shows the set-up procedure.

Table 2.4.7. Choosed functions

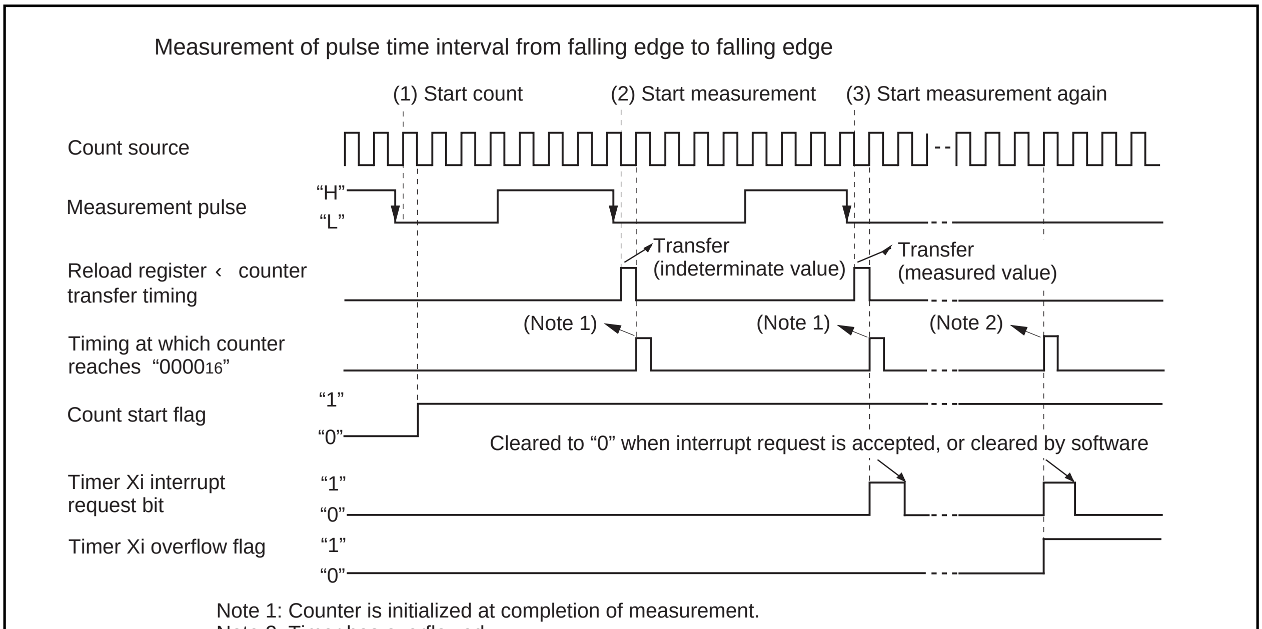
Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Measurement mode	O	Pulse period measurement (interval between measurement pulse falling edge to falling edge)
		Pulse period measurement (interval between measurement pulse rising edge to rising edge)
		Pulse width measurement (interval between measurement pulse falling edge to rising edge, and between rising edge to falling edge)

Operation (1) Setting the count start flag to “1” causes the counter to start counting the count source.

(2) If a measurement pulse changes from “H” to “L”, the value of the counter goes to “0000_{16}”, and measurement is started. In this instance, an indeterminate value is transferred to the reload register. The timer Xi interrupt request does not generate.

(3) If a measurement pulse changes from “H” to “L” again, the value of the counter is transferred to the reload register, and the timer Xi interrupt request bit goes to “1”. Then the value of the counter becomes “0000_{16}”, and the measurement is started again.

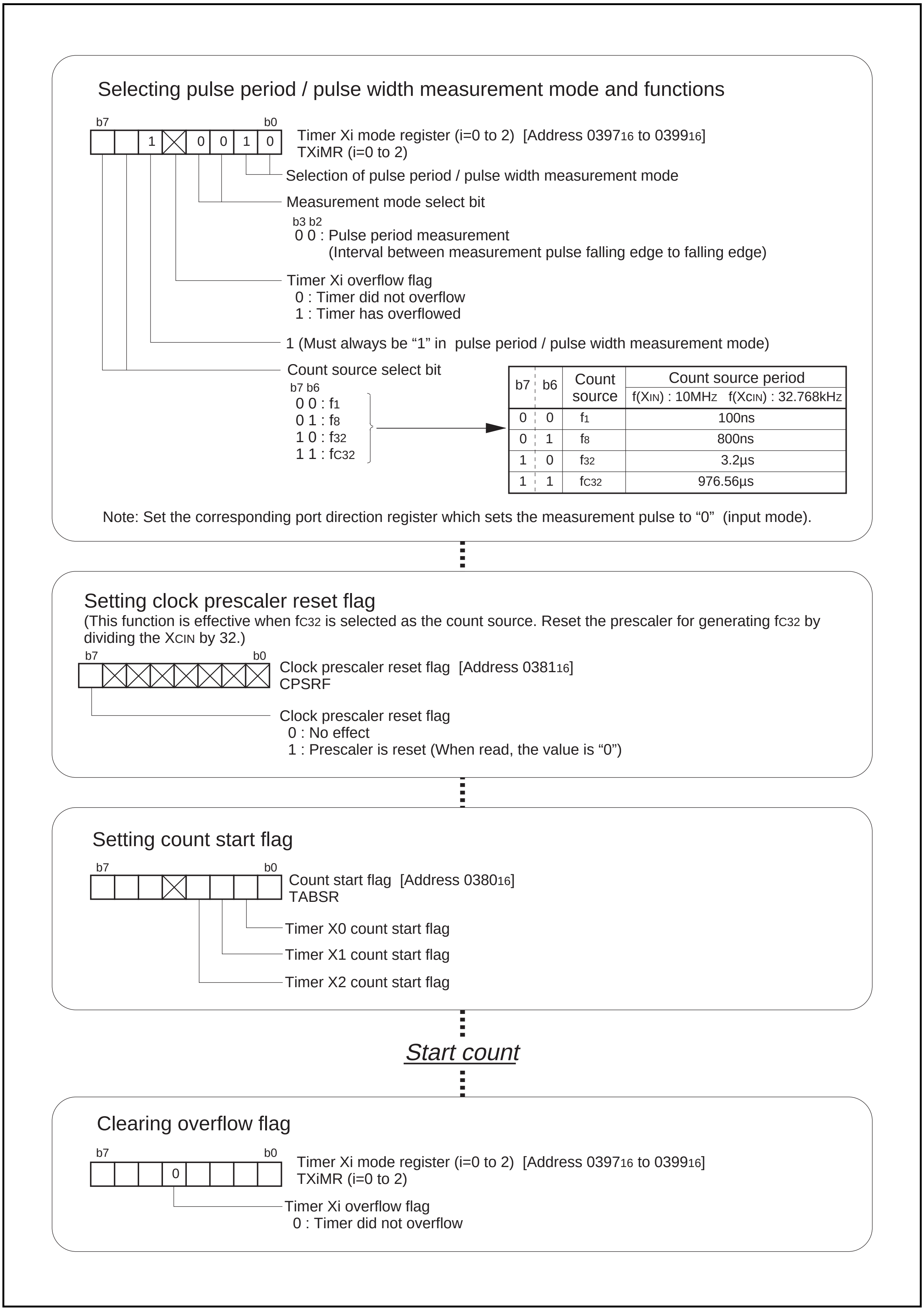
Note
- The timer Xi interrupt request bit goes to “1” when an effective edge of a measurement pulse is input or timer Xi is overflowed. The factor of interrupt request can be determined by use of the timer Xi overflow flag within the interrupt routine.
- The value of the counter at the beginning of a count is indeterminate. Thus there can be instances in which the timer Xi overflow flag goes to “1” immediately after a count is performed.
- The timer Xi overflow flag goes to “0” if timer Xi mode register is written to when the count start flag is “1”. This flag cannot be set to “1” by software.


Measurement of pulse time interval from falling edge to falling edge
(1) Start count
(2) Start measurement
(3) Start measurement again
Count source
Measurement pulse
“H”
“L”
Reload register ← counter transfer timing
Transfer (indeterminate value)
Transfer (measured value)
Timing at which counter reaches “0000_{16}”
(Note 1)
(Note 1)
(Note 2)
Count start flag
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Xi interrupt request bit
“1”
“0”
Timer Xi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 2.4.16. Operation timing of pulse period measurement mode

Under development



Selecting pulse period / pulse width measurement mode and functions

Timer Xi mode register (i=0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i=0 to 2)

Selection of pulse period / pulse width measurement mode

Measurement mode select bit
b3 b2
0 0 : Pulse period measurement
(Interval between measurement pulse falling edge to falling edge)

Timer Xi overflow flag
0 : Timer did not overflow
1 : Timer has overflowed

1 (Must always be “1” in pulse period / pulse width measurement mode)

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Note: Set the corresponding port direction register which sets the measurement pulse to “0” (input mode).

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the XCIN by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380_{16}]
TABSR

Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag

Start count

Clearing overflow flag

Timer Xi mode register (i=0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i=0 to 2)

Timer Xi overflow flag
0 : Timer did not overflow

Figure 2.4.17. Set-up procedure of pulse period measurement mode

2.4.9 Operation of Timer X (pulse width measurement mode)

In pulse period/pulse width measurement mode, choose functions from those listed in Table 2.4.8. Operations of the circled items are described below. Figure 2.4.18 shows the operation timing, and Figure 2.4.19 shows the set-up procedure.

Table 2.4.8. Choosed functions

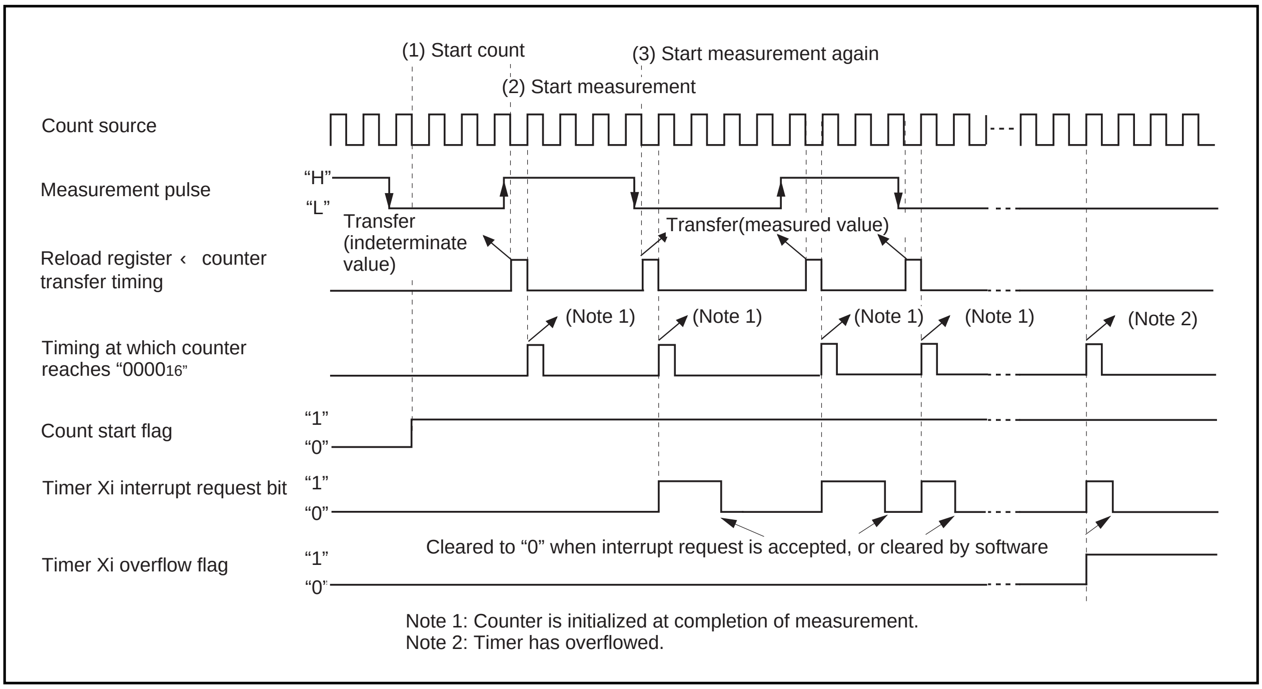
Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
Measurement mode		Pulse period measurement (interval between measurement pulse falling edge to falling edge)
		Pulse period measurement (interval between measurement pulse rising edge to rising edge)
	O	Pulse width measurement (interval between measurement pulse falling edge to rising edge, and between rising edge to falling edge)

Operation (1) Setting the count start flag to “1” causes the counter to start counting the count source.

(2) If an effective edge of a pulse to be measured is input, the value of the counter goes to “0000_{16}”, and measurement is started. In this instance, an indeterminate value is transferred to the reload register. The timer Xi interrupt request does not generate.

(3) If an effective edge of a pulse to be measured is input again, the value of the counter is transferred to the reload register, and the timer Xi interrupt request bit goes to “1”. Then the value of the counter becomes “0000_{16}”, and measurement is started again.

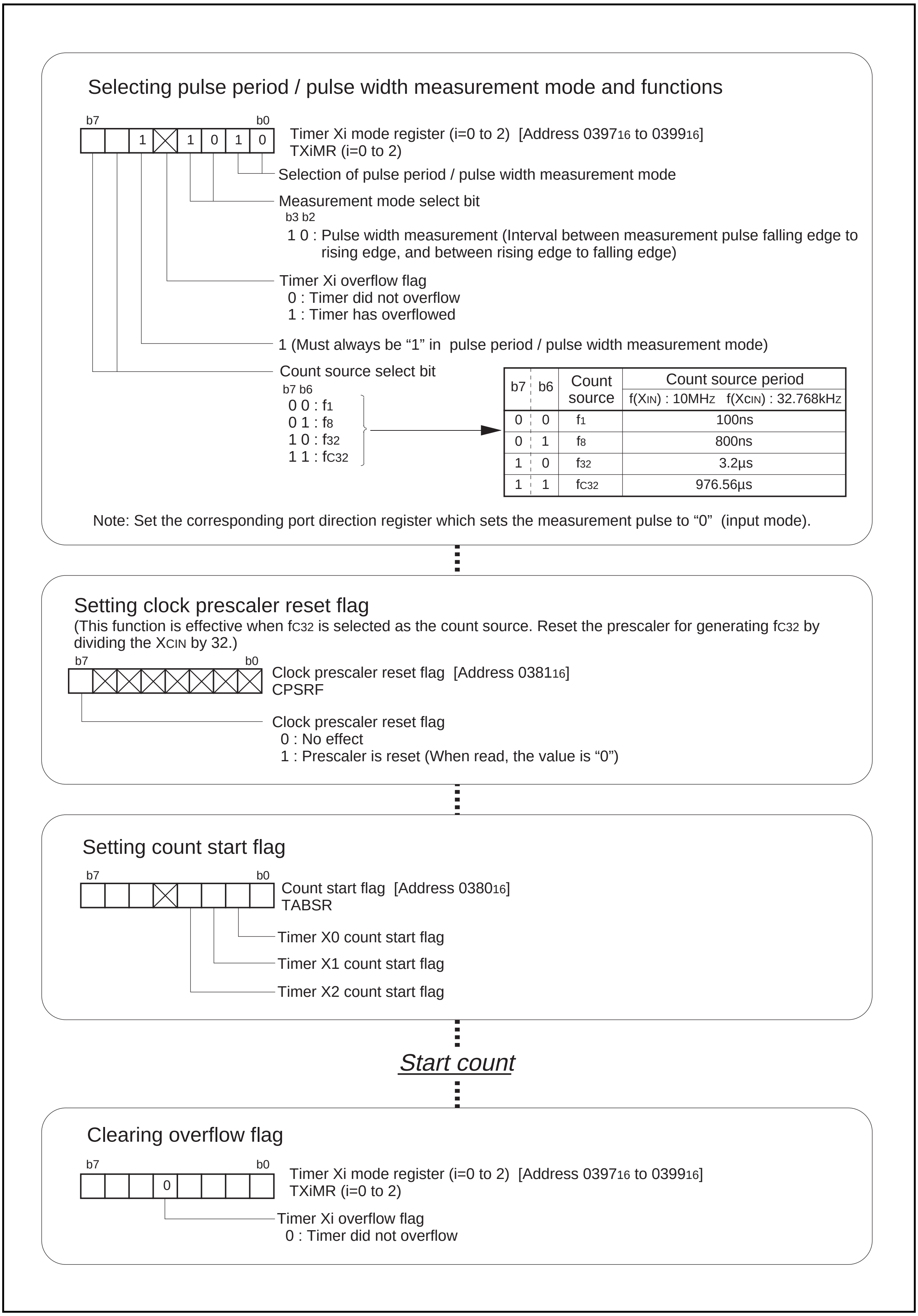
Note
- The timer Xi interrupt request bit goes to “1” when an effective edge of a pulse to be measured is input or timer Xi is overflows. The factor of interrupt request can be determined by use of the timer Xi overflow flag within the interrupt routine.
- The value of the counter at the beginning of a count is indeterminate. Thus there can be instances in which the timer Xi overflow flag goes to “1” immediately after a count is performed.
- The timer Xi overflow flag goes to “0” if timer Xi mode register is written to when the count start flag is “1”. This flag cannot be set to “1” by software.


(1) Start count
(3) Start measurement again
(2) Start measurement
Count source
Measurement pulse
“H”
“L”
Transfer (indeterminate value)
Transfer(measured value)
Reload register ‹ counter transfer timing
(Note 1)
(Note 1)
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches “0000₁₆”
Count start flag
“1”
“0”
Timer Xi interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Xi overflow flag
“1”
“0”

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 2.4.18. Operation timing of pulse width measurement mode

Under development



Selecting pulse period / pulse width measurement mode and functions

Timer Xi mode register (i=0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i=0 to 2)

Selection of pulse period / pulse width measurement mode

Measurement mode select bit
b3 b2
1 0 : Pulse width measurement (Interval between measurement pulse falling edge to rising edge, and between rising edge to falling edge)

Timer Xi overflow flag
0 : Timer did not overflow
1 : Timer has overflowed

1 (Must always be “1” in pulse period / pulse width measurement mode)

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period $f(X_{IN})$: 10MHz	$f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns	
0	1	f_8	800ns	
1	0	f_{32}	3.2µs	
1	1	f_{C32}		976.56µs

Note: Set the corresponding port direction register which sets the measurement pulse to “0” (input mode).

Setting clock prescaler reset flag

(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is “0”)

Setting count start flag

Count start flag [Address 0380_{16}]
TABSR

Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag

Clearing overflow flag

Timer Xi mode register (i=0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i=0 to 2)

Timer Xi overflow flag
0 : Timer did not overflow

Figure 2.4.19. Set-up procedure of pulse width measurement mode

Under development

2.4.10 Operation of Timer X (pulse width modulation mode, 16-bit PWM mode selected)

In pulse width modulation mode, choose functions from those listed in Table 2.4.9. Operations of the circled items are described below. Figure 2.4.20 shows the operation timing, and Figure 2.4.21 shows the set-up procedure.

Table 2.4.9. Choosed functions

Item	Set-up	
Count source	**O**	Internal count source (f_1 / f_8 / f_{32} / fc_{32})
PWM mode	**O**	16-bit PWM
		8-bit PWM
Count start condition	**O**	Timer overflow (TB1/TA0/TXi overflow)

Operation (1) Selected timer overflow is generated with the count start flag set to “1”, the counter performs a down count on the count source. Also, the TXiINOUT pin outputs an “H” level.

(2) The TXiINOUT pin output level changes from “H” to “L” when a set time period elapses. At this time, the timer Xi interrupt request bit goes to “1”.

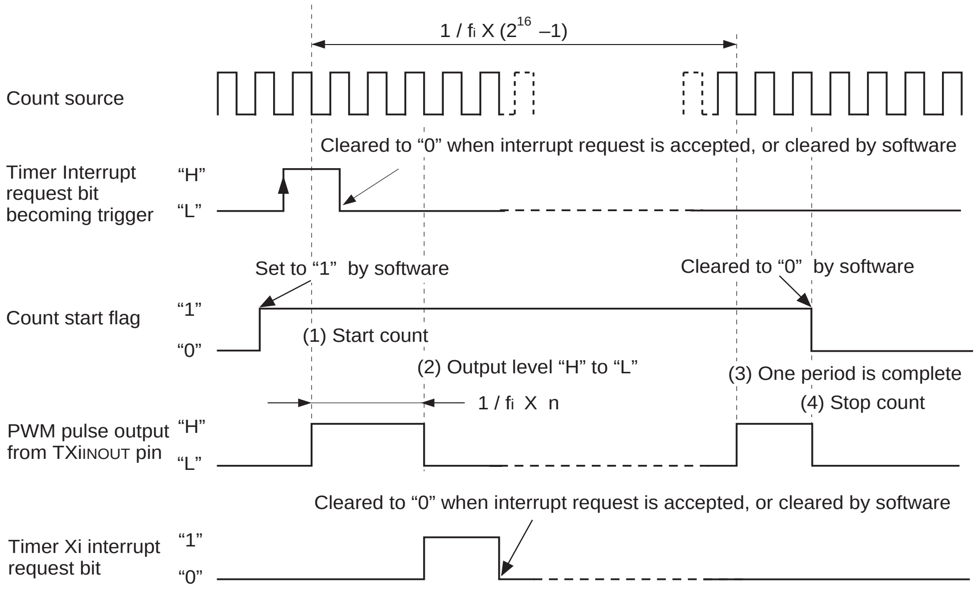
(3) The counter reloads the content of the reload register every time PWM pulses are output for one cycle, and continues counting.

(4) Setting the count start flag to “0” causes the counter to hold its value and to stop. Also, the TXiINOUT outputs an “L” level.

Note

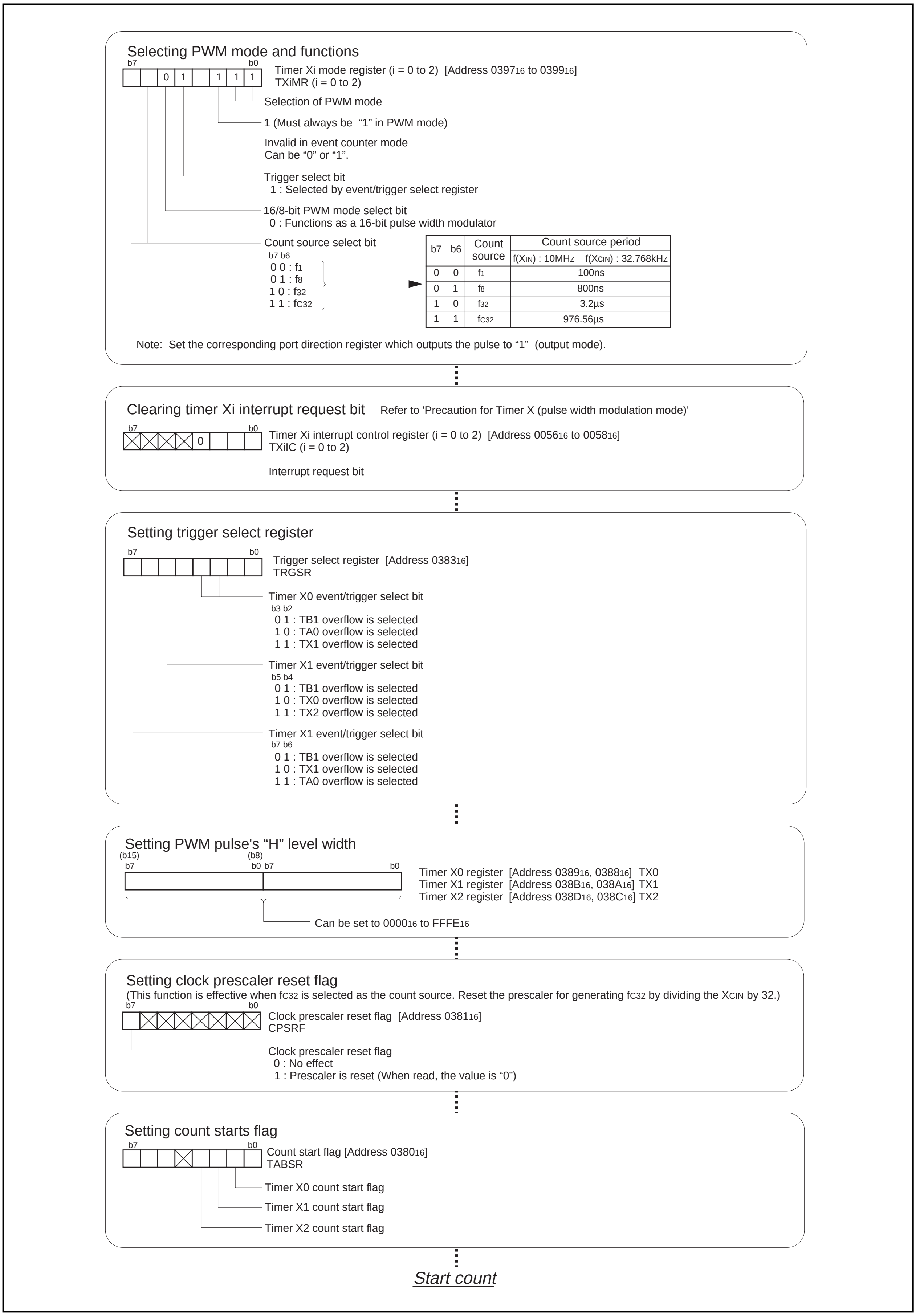
- PWM pulse cycle is $(2^{16} -1)/fi$, whereas “H” level duration is n/fi. However, when “0000_{16}” is set for the timer A0 register, the PWM output is “L” level for the entire period, and an interrupt request is generated for every PWM output cycle. Also, when “$FFFF_{16}$” is set for the timer A0 register, the PWM output is “H” level for the entire period, and an interrupt request is generated for every PWM output cycle.
  (fi: Count source frequency f_1, f_8, f_{32}, fc_{32} n: Timer value)

Conditions: Reload register = 0003_{16}, when timer overflow is selected in trigger


1 / fi X (2^16 –1)
Count source
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Interrupt request bit becoming trigger
“H”
“L”
Set to “1” by software
Cleared to “0” by software
Count start flag
“1”
“0”
(1) Start count
(2) Output level “H” to “L”
(3) One period is complete
1 / fi X n
(4) Stop count
PWM pulse output from TXiINOUT pin
“H”
“L”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Xi interrupt request bit
“1”
“0”

Note: n = 0000_{16} to $FFFE_{16}$

Figure 2.4.20. Operation timing of pulse width modulation mode, 16-bit PWM mode selected



Selecting PWM mode and functions

Timer Xi mode register (i = 0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i = 0 to 2)

Selection of PWM mode

1 (Must always be "1" in PWM mode)

Invalid in event counter mode
Can be "0" or "1".

Trigger select bit
1 : Selected by event/trigger select register

16/8-bit PWM mode select bit
0 : Functions as a 16-bit pulse width modulator

Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : f_{C32}

b7	b6	Count source	Count source period $f(X_{IN})$: 10MHz $f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Note: Set the corresponding port direction register which outputs the pulse to "1" (output mode).

Clearing timer Xi interrupt request bit Refer to 'Precaution for Timer X (pulse width modulation mode)'

Timer Xi interrupt control register (i = 0 to 2) [Address 0056_{16} to 0058_{16}]
TXiIC (i = 0 to 2)

Interrupt request bit

Setting trigger select register

Trigger select register [Address 0383_{16}]
TRGSR

Timer X0 event/trigger select bit
b3 b2
0 1 : TB1 overflow is selected
1 0 : TA0 overflow is selected
1 1 : TX1 overflow is selected

Timer X1 event/trigger select bit
b5 b4
0 1 : TB1 overflow is selected
1 0 : TX0 overflow is selected
1 1 : TX2 overflow is selected

Timer X1 event/trigger select bit
b7 b6
0 1 : TB1 overflow is selected
1 0 : TX1 overflow is selected
1 1 : TA0 overflow is selected

Setting PWM pulse's "H" level width

Timer X0 register [Address 0389_{16}, 0388_{16}] TX0
Timer X1 register [Address $038B_{16}$, $038A_{16}$] TX1
Timer X2 register [Address $038D_{16}$, $038C_{16}$] TX2

Can be set to 0000_{16} to $FFFE_{16}$

Setting clock prescaler reset flag
(This function is effective when f_{C32} is selected as the count source. Reset the prescaler for generating f_{C32} by dividing the X_{CIN} by 32.)

Clock prescaler reset flag [Address 0381_{16}]
CPSRF

Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")

Setting count starts flag

Count start flag [Address 0380_{16}]
TABSR

Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag

Start count

Figure 2.4.21. Set-up procedure of pulse width modulation mode, 16-bit PWM mode selected

2.4.11 Operation of Timer X (pulse width modulation mode, 8-bit PWM mode selected)

In pulse width modulation mode, choose functions from those listed in Table 2.4.10. Operations of the circled items are described below. Figure 2.4.22 shows the operation timing, and Figure 2.4.22 shows the set-up procedure.

Table 2.4.10. Choosed functions

Item	Set-up	
Count source	O	Internal count source (f_1 / f_8 / f_{32} / f_{C32})
PWM mode		16-bit PWM
	O	8-bit PWM
Count start condition	O	Timer overflow (TB1/TA0/TXi overflow)

Operation (1) Selected timer overflow is generated with the count start flag set to “1”, the counter performs a down count on the count source. Also, the TXiINOUT pin outputs an “H” level.

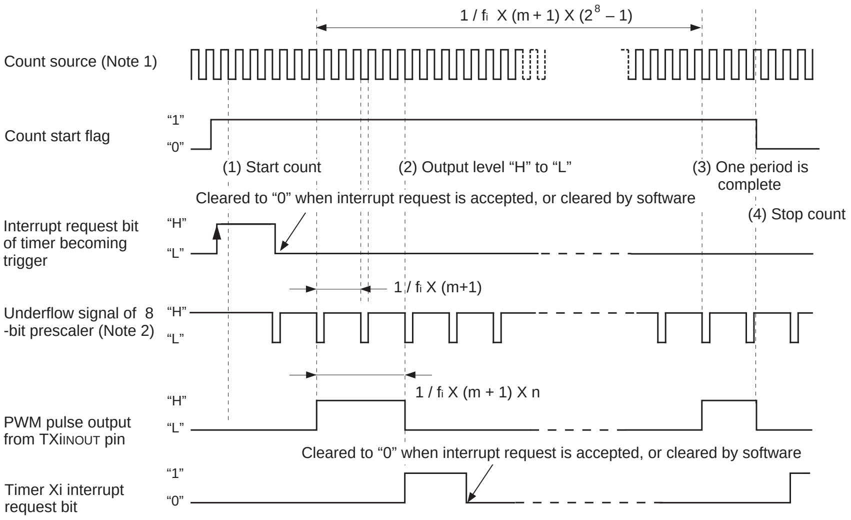
(2) The TXiINOUT pin output level changes from “H” to “L” when a set time period elapses. At this time, the timer Xi interrupt request bit goes to “1”.

(3) The counter reloads the content of the reload register every time PWM pulses are output for one cycle, and continues counting.

(4) Setting the count start flag to “0” causes the counter to hold its value and to stop. Also, the TXiOUT pin outputs an “L” level.

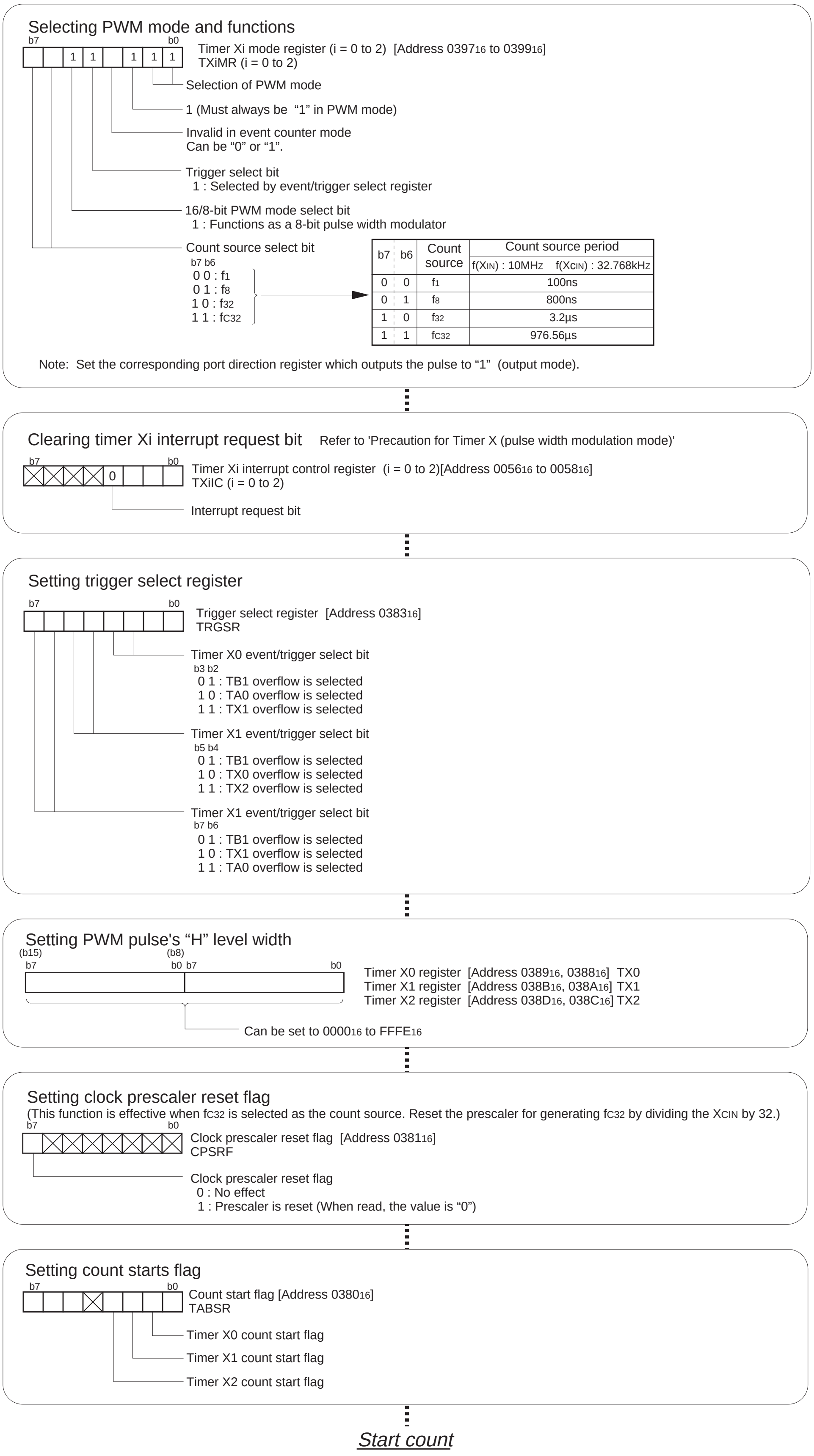
Note • PWM pulse cycle is (m + 1(x (2^8 -1)/fi, whereas “H” level duration is n x (m + 1)/fi. However, when “00_{16}” is set for the significant 8 bits of the timer A0 register, the PWM output is “L” level for the entire period, and an interrupt request is generated for every PWM output cycle. Also, when “FF_{16}” is set for the significant 8 bits of the timer A0 register, the PWM output is “H” level for the entire period, and an interrupt request is generated for every PWM output cycle.
(fi: Count source frequency f_1, f_8, f_{32}, f_{C32} n: Timer value)

Conditions: Reload register high-order 8 bits = 02_{16}
Reload register low-order 8 bits = 02_{16}
When timer overflow is selected in trigger


1 / fi X (m + 1) X (2^8 – 1)
Count source (Note 1)
Count start flag
“1”
“0”
(1) Start count
(2) Output level “H” to “L”
(3) One period is complete
Cleared to “0” when interrupt request is accepted, or cleared by software
(4) Stop count
Interrupt request bit of timer becoming trigger
“H”
“L”
1 / fi X (m+1)
Underflow signal of 8 -bit prescaler (Note 2)
“H”
“L”
1 / fi X (m + 1) X n
PWM pulse output from TXiINOUT pin
“H”
“L”
Cleared to “0” when interrupt request is accepted, or cleared by software
Timer Xi interrupt request bit
“1”
“0”

Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00_{16} to FF_{16}; n = 00_{16} to FF_{16}.

Figure 2.4.22. Operation timing of pulse width modulation mode, with 8-bit PWM mode selected


Selecting PWM mode and functions
b7
b0
1
1
1
1
1
Timer Xi mode register (i = 0 to 2) [Address 0397_{16} to 0399_{16}]
TXiMR (i = 0 to 2)
Selection of PWM mode
1 (Must always be "1" in PWM mode)
Invalid in event counter mode
Can be "0" or "1".
Trigger select bit
1 : Selected by event/trigger select register
16/8-bit PWM mode select bit
1 : Functions as a 8-bit pulse width modulator
Count source select bit
b7 b6
0 0 : f_1
0 1 : f_8
1 0 : f_{32}
1 1 : fc_{32}
Note: Set the corresponding port direction register which outputs the pulse to "1" (output mode).
Clearing timer Xi interrupt request bit
Refer to 'Precaution for Timer X (pulse width modulation mode)'
b7
b0
0
Timer Xi interrupt control register (i = 0 to 2)[Address 0056_{16} to 0058_{16}]
TXiIC (i = 0 to 2)
Interrupt request bit
Setting trigger select register
b7
b0
Trigger select register [Address 0383_{16}]
TRGSR
Timer X0 event/trigger select bit
b3 b2
0 1 : TB1 overflow is selected
1 0 : TA0 overflow is selected
1 1 : TX1 overflow is selected
Timer X1 event/trigger select bit
b5 b4
0 1 : TB1 overflow is selected
1 0 : TX0 overflow is selected
1 1 : TX2 overflow is selected
Timer X1 event/trigger select bit
b7 b6
0 1 : TB1 overflow is selected
1 0 : TX1 overflow is selected
1 1 : TA0 overflow is selected
Setting PWM pulse's "H" level width
(b15)
(b8)
b7
b0 b7
b0
Timer X0 register [Address 0389_{16}, 0388_{16}] TX0
Timer X1 register [Address $038B_{16}$, $038A_{16}$] TX1
Timer X2 register [Address $038D_{16}$, $038C_{16}$] TX2
Can be set to 0000_{16} to $FFFE_{16}$
Setting clock prescaler reset flag
(This function is effective when fc_{32} is selected as the count source. Reset the prescaler for generating fc_{32} by dividing the X_{CIN} by 32.)
b7
b0
Clock prescaler reset flag [Address 0381_{16}]
CPSRF
Clock prescaler reset flag
0 : No effect
1 : Prescaler is reset (When read, the value is "0")
Setting count starts flag
b7
b0
Count start flag [Address 0380_{16}]
TABSR
Timer X0 count start flag
Timer X1 count start flag
Timer X2 count start flag
Start count

b7	b6	Count source	Count source period $f(X_{IN})$: 10MHz $f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	fc_{32}	976.56µs

Figure 2.4.23. Set-up procedure of pulse width modulation mode, 8-bit PWM mode selected

2.4.12 Precautions for Timer X (timer mode, event counter mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Xi register, then set the flag to “1”.

(2) Reading the timer Xi register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Xi register with the reload timing shown in Figure 2.4.24 gets “FFFF$_{16}$”. Reading the timer Xi register after setting a value in the timer Xi register with a count halted but before the counter starts counting gets a proper value.

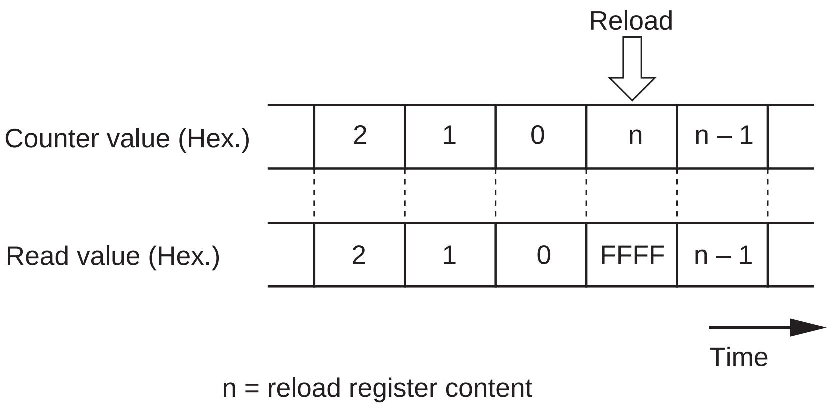


Figure 2.4.24. Reading timer Xi register

2.4.13 Precautions for Timer X (one-shot timer mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Xi register, then set the flag to “1”.

(2) Setting the count start flag to “0” while a count is in progress causes as follows:

- The counter stops counting and a content of reload register is reloaded.
- The TXiINOUT pin outputs “L” level.
- The interrupt request generated and the timer Xi interrupt request bit goes to “1”.

(3) The timer Xi interrupt request bit goes to “1” if the timer's operation mode is set using any of the following procedures:

- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Xi interrupt (interrupt request bit), set timer Xi interrupt request bit to “0” after the above listed changes have been made.

(4) If a trigger occurs while a count is in progress, after the counter performs one down count following the reoccurrence of a trigger, the reload register contents are reloaded, and the count continues. To generate a trigger while a count is in progress, generate the second trigger after an elapse longer than one cycle of the timer's count source after the previous trigger occurred.

2.4.14 Precautions for Timer X (pulse period/pulse width measurement mode)

(1) The timer Xi interrupt request bit goes to “1” when an effective edge of a measurement pulse is input or timer Xi is overflowed. The factor of interrupt request can be determined by use of the timer Xi overflow flag within the interrupt routine.

(2) If the timer overflow occurs simultaneously with the input of a measurement pulse, and if the interrupt factor cannot be determined from the timer Xi overflow flag, connect the timers and count the number of overflows.

(3) When reset, the timer Xi overflow flag goes to “1”. This flag cannot be set to “0” by writing to the timer Xi mode register when the count start flag is “1”.

(4) Use the timer Xi interrupt request bit to detect only overflows. Use the timer Xi overflow flag only to determine the interrupt factor within the interrupt routine.

(5) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Xi interrupt request is not generated.

(6) The value of the counter is indeterminate at the beginning of a count. Therefore the timer Xi overflow flag may go to “1” immediately after a count is started.

(7) If changing the measurement mode select bit is set after a count is started, the timer Xi interrupt request bit goes to “1”.

(8) If the input signal to the TX_{iINOUT} pin is affected by noise, precise measurement may not be performed in some cases. It is recommended to see that measurements fall within a specific range by use of software.

(9) For pulse width measurement, pulse widths are successively measured. Use software to check whether the measurement result is an “H” level width or an “L” level width.

2.4.15 Precautions for Timer X (pulse width modulation mode)

(1) To clear reset, the count start flag is set to “0”. Set a value in the timer Xi register, then set the flag to “1”.

(2) The timer Xi interrupt request bit becomes “1” if setting operation mode of the timer in compliance with any of the following procedures:

- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Xi interrupt (interrupt request bit), set timer Xi interrupt request bit to “0” after the above listed changes have been made.

(3) Setting the count start flag to “0” while PWM pulses are being output causes the counter to stop counting. If the TXiINOUT pin is outputting an “H” level in this instance, the output level goes to “L”, and the timer Xi interrupt request bit goes to “1”. If the TXiINOUT pin is outputting an “L” level in this instance, the level does not change, and the timer Xi interrupt request bit does not becomes “1”.

(4) Normal PWM output is restored according to the interrupt request generate timing, both in the case of 16-bit PWM and 8-bit PWM, when PWM output is either “H” or “L” level for the entire period. This holds only when a value other than “0000_{16}” or “$FFFF_{16}$” is set during 16-bit PWM, or a value other than “00_{16}” or “FF_{16}” is set during 8-bit PWM.

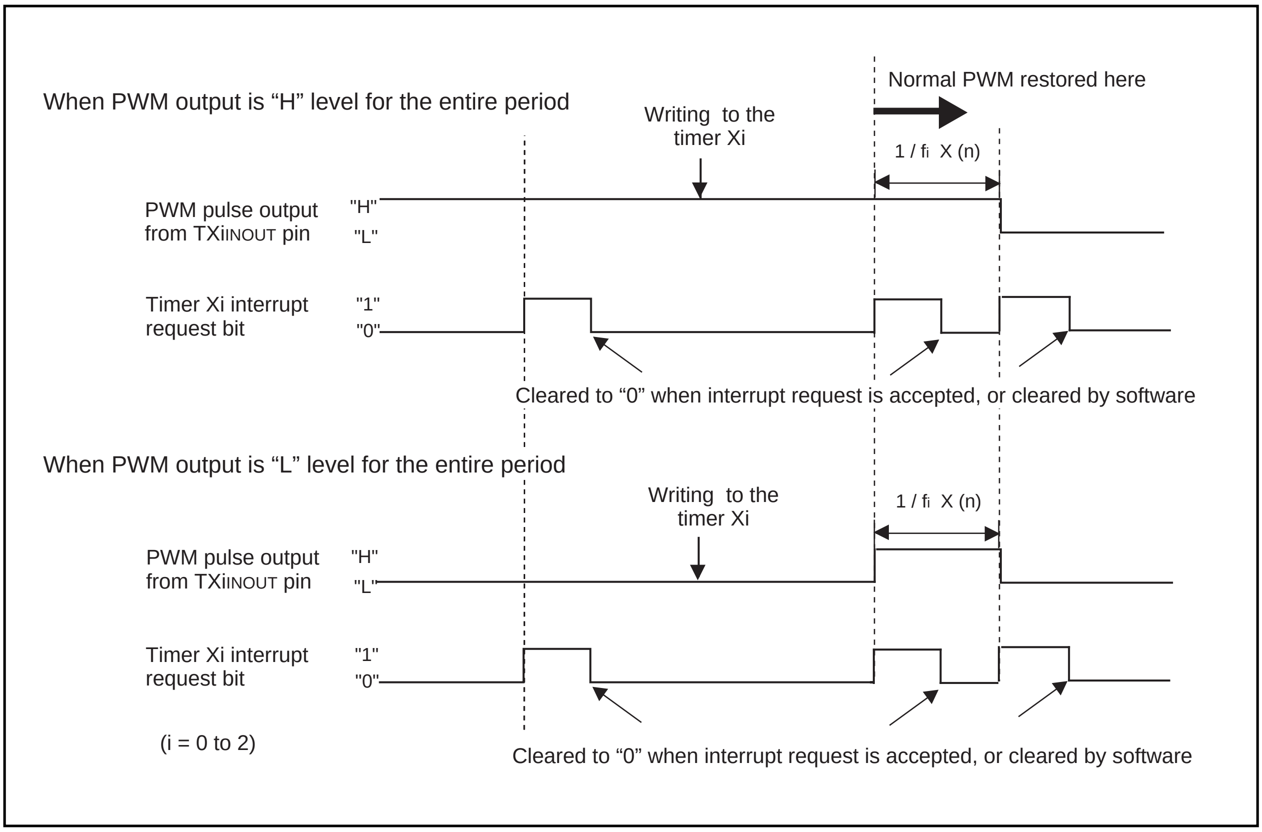

When PWM output is “H” level for the entire period
Normal PWM restored here
Writing to the timer Xi
1 / fi X (n)
PWM pulse output from TXiINOUT pin
"H"
"L"
Timer Xi interrupt request bit
"1"
"0"
Cleared to “0” when interrupt request is accepted, or cleared by software
When PWM output is “L” level for the entire period
Writing to the timer Xi
1 / fi X (n)
PWM pulse output from TXiINOUT pin
"H"
"L"
Timer Xi interrupt request bit
"1"
"0"
(i = 0 to 2)
Cleared to “0” when interrupt request is accepted, or cleared by software

Figure 2.4.25. Operation timing of PWM output mode

2.5 Clock-Synchronous Serial I/O

2.5.1 Overview

Clock-synchronous serial I/O carries out 8-bit data communications in synchronization with the clock. The following is an overview of the clock-synchronous serial I/O.

(1) Transmission/reception format

8-bit data

(2) Transfer rate

If the internal clock is selected as the transfer clock, the divide-by-2 frequency, resulting from the bit rate generator division, becomes the transfer rate. The bit rate generator count source can be selected from the following: f_1, f_8, f_{32}, and f_C. Clocks f_1, f_8 and f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively. Clock f_C is derived by dividing the CPU's sub clock by 1 respectively.

Furthermore, if an external clock is selected as the transfer clock, the clock frequency input to the CLK pin becomes the transfer rate.

(3) Error detection

Only overrun error can be detected. Overrun error is an error that occurs when the next data is made ready before the reception buffer register is read.

(4) How to deal with an error

When receiving data, read an error flag and reception data simultaneously to determine which error has occurred. If the data read is erroneous, initialize the error flag and the UART0 receive buffer register, then receive the data again.

To initialize the UART0 receive buffer register

1. Set the receive enable bit to “0” (disable reception).
2. Set the serial I/O mode select bit to “000_2” (invalid serial I/O).
3. Set the serial I/O mode select bit.
4. Set the receive enable bit to “1” again (enable reception).

To transmit data again due to an error on the reception side, set the UART0 transmit buffer register again, then transmit the data again.

To set the UART0 transmit buffer register again

1. Set the serial I/O mode select bits to “000_2” (invalidate serial I/O).
2. Set the serial I/O mode select bits again.
3. Set the transmit enable bit to “1” (enable transmission), then set transmission data in the UART0 transmit buffer register.

(5) Function selection

For clock-synchronous serial I/O, the following functions can be selected:

(a) Function for choosing polarity

This function switches the polarity of the transfer clock. The following operations are available:

- Data is input at the falling edge of the transfer clock, and is output at the rising edge.
- Data is input at the rising edge of the transfer clock, and is output at the falling edge.

(b) Function for choosing which bit to transmit first

This function is to choose whether to transmit data from bit 0 or from bit 7. Choose either of the following:

• LSB first	Data is transmitted from bit 0.
• MSB first	Data is transmitted from bit 7.

(c) Function for choosing successive reception mode

Successive reception mode is a mode in which reading the receive buffer register makes the reception-enabled status ready. In this mode, there is no need to write dummy data to the transmit buffer register so as to make the reception-enabled status ready. But at the time of starting reception, read the receive buffer register into a dummy manner.

• Normal mode	Writing dummy data to the transmit buffer register makes the reception enabled status ready.
• Successive reception mode	Reading the reception buffer register makes the reception-enabled status ready.

(d) Function for outputting transfer clock to multiple pins

This function is to switch among pins to output the transfer clock. This function is effective only when selecting the internal clock. Switching among pins for outputting the transfer clock allows data transmission to two external ICs in a time-sharing manner.

(e) Function for choosing a transmission interrupt factor

The timing to generate a transmission interrupt can be selected from the following: the instant the transmission buffer is emptied or the instant the transmission register is emptied. When transmission buffer empty timing is selected, an interrupt occurs when transmitted data is moved from the transmission buffer to the transmission register. Therefore, data can be transmitted in succession. When transmission register empty timing is selected, an interrupt occurs when data transmission is complete.

Following are some examples in which various functions (a) through (e) are selected:

- Transmission Operation WITH: transmission at falling edge of transfer clock, LSB First, interrupt at instant transmission buffer is emptied; WITHOUT transfer clock output to multiple pins function P242
- Transmission Operation WITH: transmission at falling edge of transfer clock, LSB First, interrupt at instant transmission is completed; WITH transfer clock output to multiple pins function (UART0 selection available) P246
- Reception WITH: reception at falling edge of transfer clock, LSB First, successive reception mode disabled; WITHOUT transfer clock output to multiple pins function P250

(6) Input/output to the serial I/O and the direction register

To input an external signal to the serial I/O, set the direction register of the relevant port to input. To output signal from the serial I/O, set the direction register of the relevant port to output.

Under development

(7) Pins related to the serial I/O

• CLK0 pin	Input/output pins for the transfer clock
• RxD0, RxD1 pins	Input pins for data
• TxD0, TxD1 pins	Output pins for data (Since TxD2 pin is N-channel open drain, this pin needs pull-up resistor.)
• CLKS pin	Output pin for transfer clock. Can be used as transfer clock output pin in the transfer clock output to multiple pins function.

Note : UART1 cannot be used in clock-synchronous serial I/O mode.

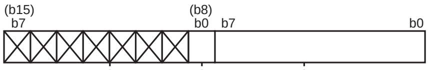
(8) Registers related to the serial I/O

Figure 2.5.1 shows the memory map of serial I/O-related registers, and Figures 2.5.2 to 2.5.4 show serial I/O-related registers.

0051_{16}	UART0 transmit interrupt control register (S0TIC)
0052_{16}	UART0 receive interrupt control register (S0RIC)
0053_{16}	UART1 transmit interrupt control regster(S1TIC)
0054_{16}	UART1 receive interrupt control register(S1RIC)
≈	≈
$03A0_{16}$	UART0 transmit/receive mode register (U0MR)
$03A1_{16}$	UART0 bit rate generator (U0BRG)
$03A2_{16}$	UART0 transmit buffer register (U0TB)
$03A3_{16}$	
$03A4_{16}$	UART0 transmit/receive control register 0 (U0C0)
$03A5_{16}$	UART0 transmit/receive control register 1 (U0C1)
$03A6_{16}$	UART0 receive buffer register (U0RB)
$03A7_{16}$	
$03A8_{16}$	UART1 transmit/receive mode register (U1MR)
$03A9_{16}$	UART1 bit rate generator (U1BRG)
$03AA_{16}$	UART1 transmit buffer register (U1TB)
$03AB_{16}$	
$03AC_{16}$	UART1 transmit/receive control register 0 (U1C0)
$03AD_{16}$	UART1 transmit/receive control register 1 (U1C1)
$03AE_{16}$	UART1 receive buffer register (U1RB)
$03AF_{16}$	
$03B0_{16}$	UART transmit/receive control register 2 (UCON)
$03B1_{16}$	

Figure 2.5.1. Memory map of serial I/O-related registers

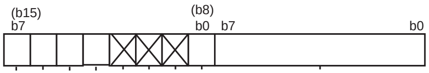
UARTi transmit buffer register



Symbol	Address	When reset
U0TB	03A3$_{16}$, 03A2$_{16}$	Indeterminate
U1TB	03AB$_{16}$, 03AA$_{16}$	Indeterminate

	R	W
Transmit data	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.	–	–

UARTi receive buffer register

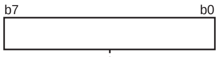


Symbol	Address	When reset
U0RB	03A7$_{16}$, 03A6$_{16}$	Indeterminate
U1RB	03AF$_{16}$, 03AE$_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	———	Receive data	Receive data	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				–	–
OER	Overrun error flag (Note)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note)	Invalid	0 : No error 1 : Error found	○	×

Note: Bits 15 through 12 are set to “0” when the serial I/O mode select bit (bits 2 to 0 at addresses 03A0$_{16}$ and 03A8$_{16}$) are set to “000$_2$” or the receive enable bit is set to “0”. (Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses 03A6$_{16}$, and 03AE$_{16}$) is read out.

UARTi bit rate generator



Symbol	Address	When reset
U0BRG	03A1$_{16}$	Indeterminate
U1BRG	03A9$_{16}$	Indeterminate

	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00$_{16}$ to FF$_{16}$	×	○

Figure 2.5.2. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
UiMR(i=0,1)	$03A0_{16}$, $03A8_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 1)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit (Note 2)	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: UART1 can use only internal clock. Must set this bit to “1”.

UARTi transmit/receive control register 0

b7	b6	b5	b4	b3	b2	b1	b0
			1		0		

Symbol	Address	When reset
UiC0(i=0,1)	$03A4_{16}$, $03AC_{16}$	08_{16}

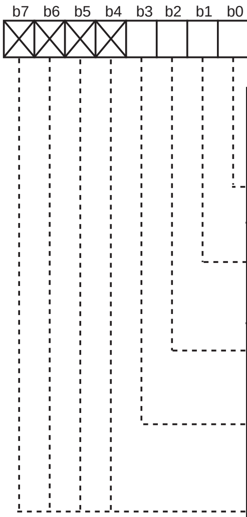
Bit symbol	Bit name	Function (Note) (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	○	○
CLK1				○	○
	Set this bit to “0”.			○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
	Set this bit to “1”.			○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be “0”	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be “0”	○	○

Note: UART1 cannot be used in clock synchronous serial I/O.

Figure 2.5.3. Serial I/O-related registers (2)

Under development

UARTi transmit/receive control register 1

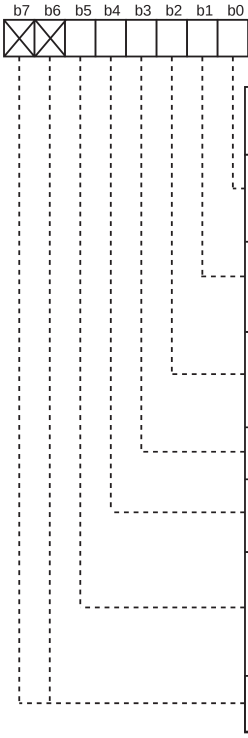


Symbol UiC1(i=0,1)
Address $03A5_{16}$,$03AD_{16}$
When reset 02_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				–	–

Note: UART1 cannot be used in clock synchronous serial I/O.

UART transmit/receive control register 2



Symbol UCON
Address $03B0_{16}$
When reset $XX000000_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	Set this bit to “0”.	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
Set this bit to “0”.				○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note 2)	0 : Normal mode (CLK output is CLK0 only) 1 : Transfer clock output from multiple pins function selected	Must always be “0”	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				–	–

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: When using multiple pins to output the transfer clock, the following requirements must be met:
- UART0 internal/external clock select bit (bit 3 at address $03A0_{16}$) = “0”.

Figure 2.5.4. Serial I/O-related registers (3)

2.5.2 Operation of Serial I/O (transmission in clock-synchronous serial I/O mode)

In transmitting data in clock-synchronous serial I/O mode, choose functions from those listed in Table 2.5.1. Operations of the circled items are described below. Figure 2.5.5 shows the operation timing, and Figures 2.5.6 and 2.5.7 show the set-up procedures.

Table 2.5.1. Choosed functions

Item	Set-up	
Transfer clock source	**O**	Internal clock (f_1 / f_8 / f_{32} / f_c)
		External clock (CLK0 pin)
CLK polarity	**O**	Output transmission data at the falling edge of the transfer clock
		Output transmission data at the rising edge of the transfer clock
Transfer clock	**O**	LSB first
		MSB first
Transmission interrupt factor	**O**	Transmission buffer empty
		Transmission complete
Output transfer clock to multiple pins (Note)	**O**	Not selected
		Selected

Note: This can be selected only when UART0 is used in combination with the internal clock.

Operation

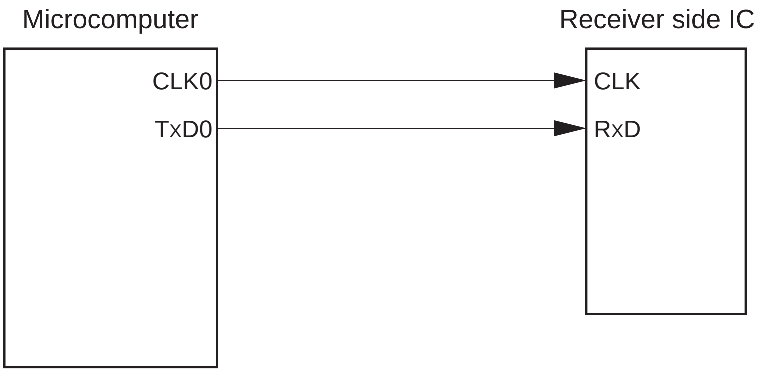
(1) Setting the transmit enable bit to “1” and writing transmission data to the UART0 transmit buffer register makes data transmissible status ready.

(2) In synchronization with the first falling edge of the transfer clock, transmission data held in the UART0 transmit buffer register is transmitted to the UART0 transmit register. At this time, the UART0 transmit interrupt request bit goes to “1”. Also, the first bit of the transmission data is transmitted from the TxD_0 pin. Then the data is transmitted bit by bit from the lower order in synchronization with the falling edges.

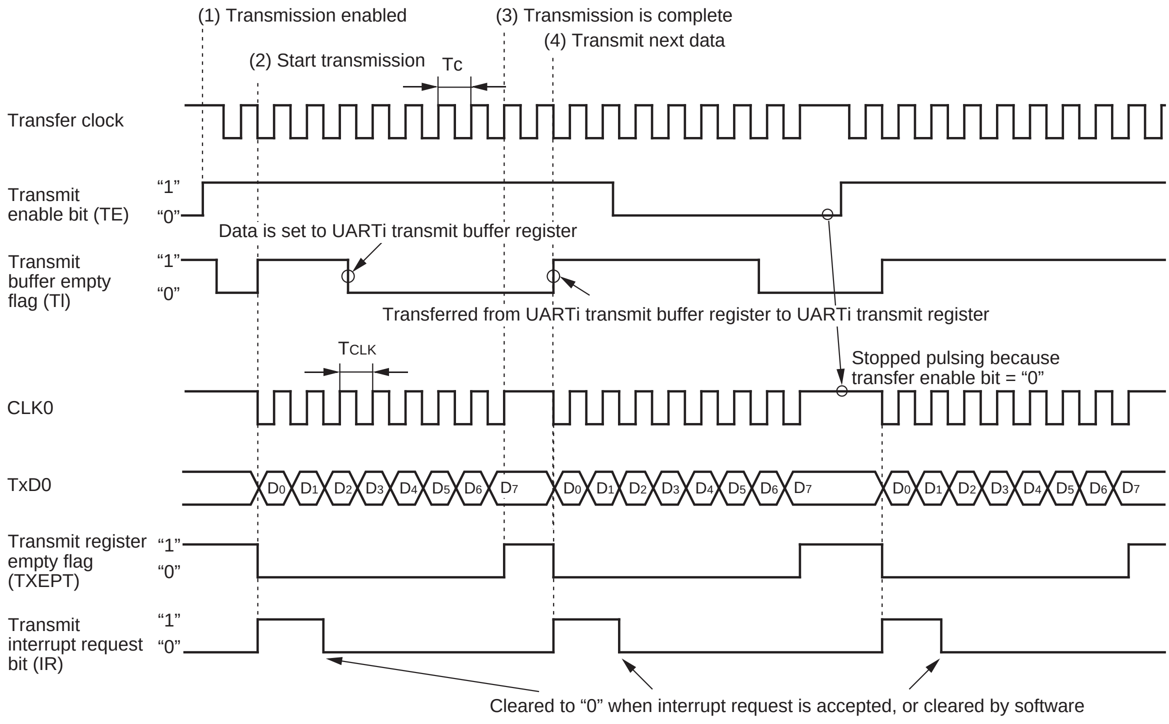
(3) When transmission of 1-byte data is completed, the transmit register empty flag goes to “1”, which indicates that transmission is completed. The transfer clock stops at “H” level.

(4) If the next transmission data is set in the UART0 transmit buffer register while transmission is in progress (before the eighth bit has been transmitted), the data is transmitted in succession.

Example of wiring



Example of operation


(1) Transmission enabled
(2) Start transmission
Tc
(3) Transmission is complete
(4) Transmit next data
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Data is set to UARTi transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
TCLK
Stopped pulsing because transfer enable bit = "0"
CLK0
TxD0
D0 D1 D2 D3 D4 D5 D6 D7
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.
The above timing applies to the following settings:
- Internal clock is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

$T_C = T_{CLK} = 2(n + 1) / fi$
fi: frequency of BRGi count source (f_1, f_8, f_{32}, f_C)
n: value set to BRGi

Figure 2.5.5. Operation timing of transmission in clock-synchronous serial I/O mode

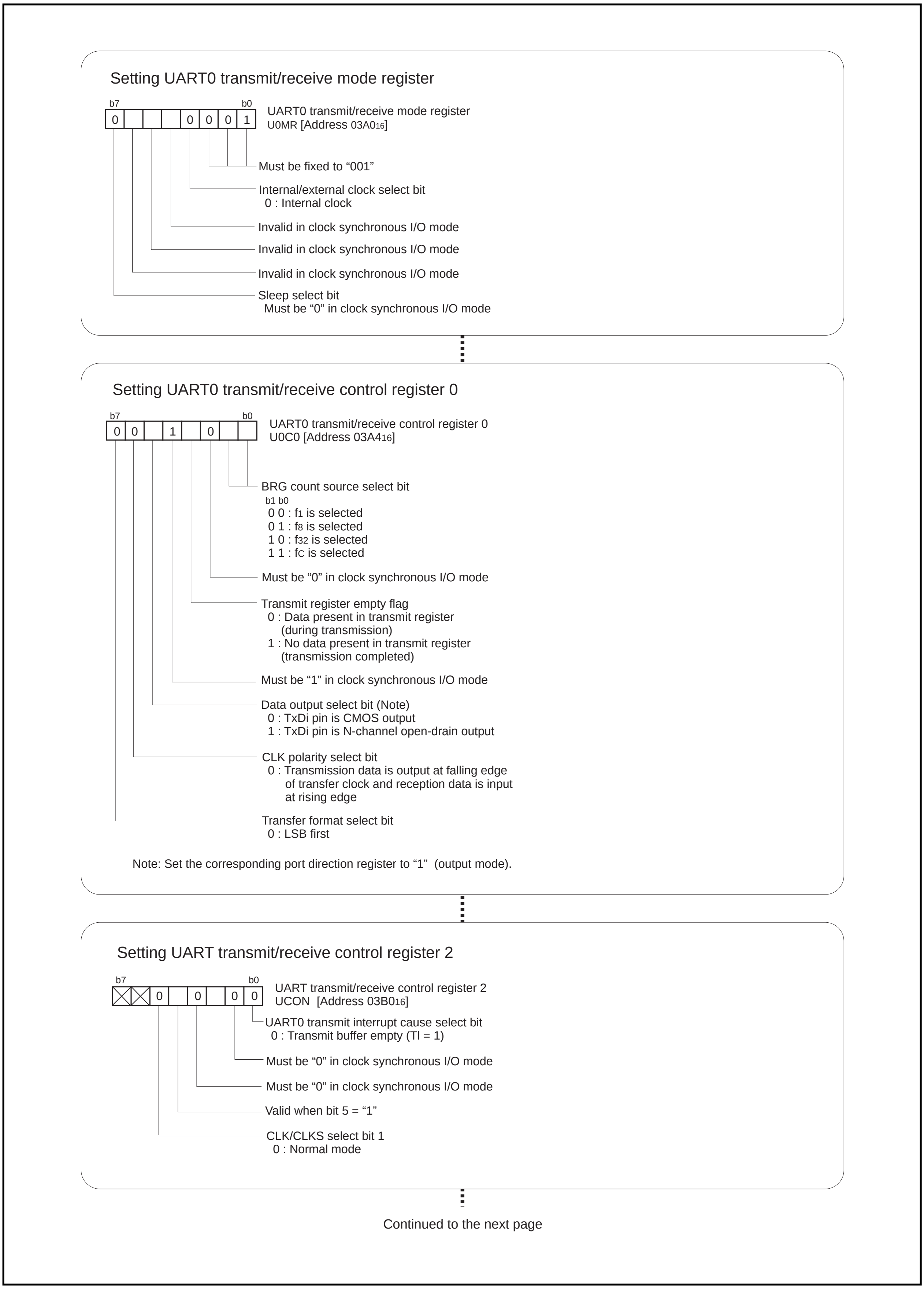
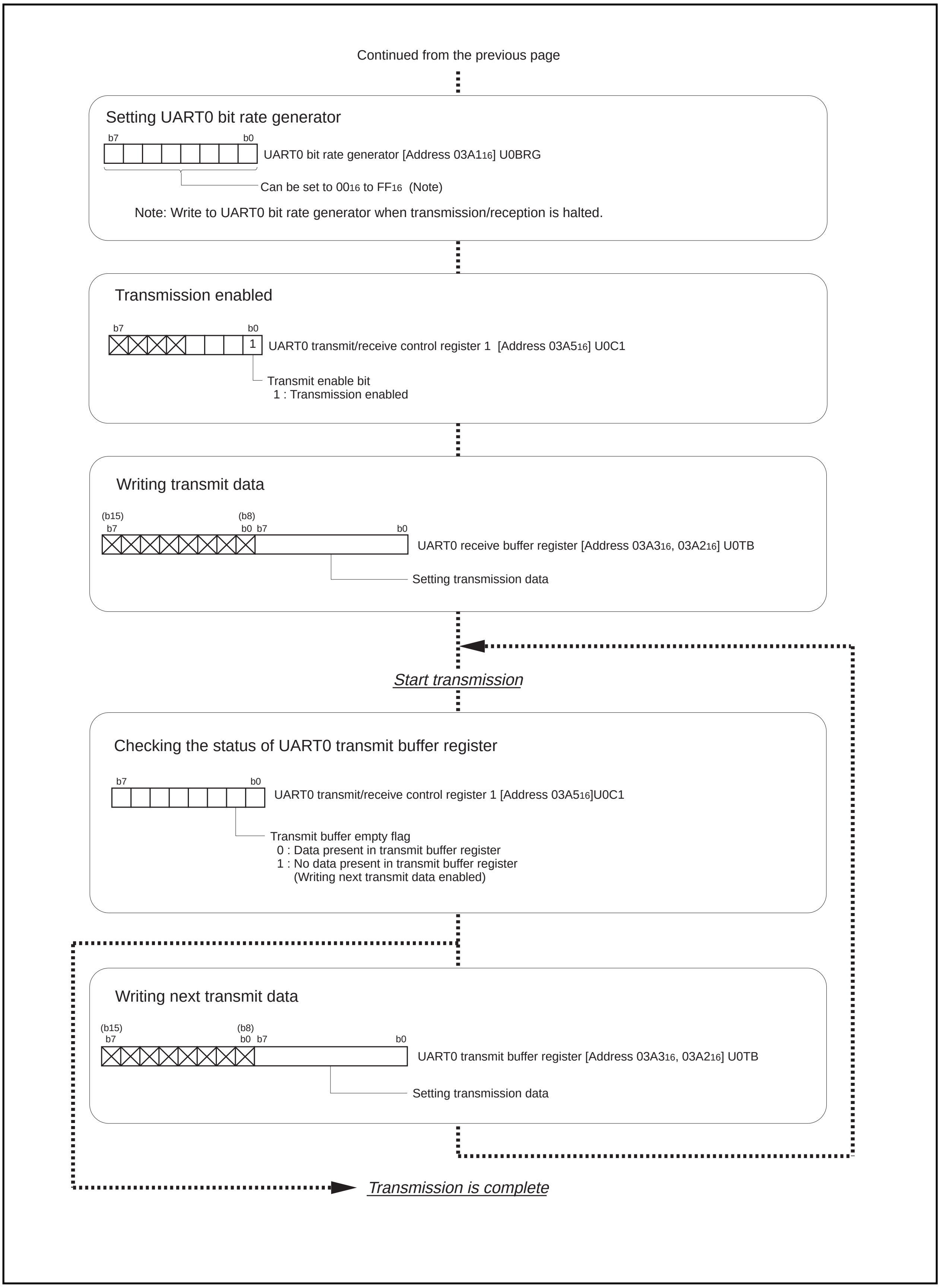

Setting UART0 transmit/receive mode register
b7
b0
0
0
0
0
1
UART0 transmit/receive mode register
U0MR [Address $03A0_{16}$]
Must be fixed to “001”
Internal/external clock select bit
0 : Internal clock
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Sleep select bit
Must be “0” in clock synchronous I/O mode
Setting UART0 transmit/receive control register 0
b7
b0
0
0
1
0
UART0 transmit/receive control register 0
U0C0 [Address $03A4_{16}$]
BRG count source select bit
b1 b0
0 0 : f_1 is selected
0 1 : f_8 is selected
1 0 : f_{32} is selected
1 1 : fc is selected
Must be “0” in clock synchronous I/O mode
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
Must be “1” in clock synchronous I/O mode
Data output select bit (Note)
0 : TxDi pin is CMOS output
1 : TxDi pin is N-channel open-drain output
CLK polarity select bit
0 : Transmission data is output at falling edge of transfer clock and reception data is input at rising edge
Transfer format select bit
0 : LSB first
Note: Set the corresponding port direction register to “1” (output mode).
Setting UART transmit/receive control register 2
b7
b0
0
0
0
0
UART transmit/receive control register 2
UCON [Address $03B0_{16}$]
UART0 transmit interrupt cause select bit
0 : Transmit buffer empty (TI = 1)
Must be “0” in clock synchronous I/O mode
Must be “0” in clock synchronous I/O mode
Valid when bit 5 = “1”
CLK/CLKS select bit 1
0 : Normal mode
Continued to the next page

Figure 2.5.6. Set-up procedure of transmission in clock-synchronous serial I/O mode (1)

Under development


Continued from the previous page
Setting UART0 bit rate generator
b7
b0
UART0 bit rate generator [Address $03A1_{16}$] U0BRG
Can be set to 00_{16} to FF_{16} (Note)
Transmission enabled
b7
b0
1
UART0 transmit/receive control register 1 [Address $03A5_{16}$] U0C1
Transmit enable bit
1 : Transmission enabled
Writing transmit data
(b15)
b7
(b8)
b0 b7
b0
UART0 receive buffer register [Address $03A3_{16}$, $03A2_{16}$] U0TB
Setting transmission data
Start transmission
Checking the status of UART0 transmit buffer register
b7
b0
UART0 transmit/receive control register 1 [Address $03A5_{16}$]U0C1
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register
(Writing next transmit data enabled)
Writing next transmit data
(b15)
b7
(b8)
b0 b7
b0
UART0 transmit buffer register [Address $03A3_{16}$, $03A2_{16}$] U0TB
Setting transmission data
Transmission is complete

Note: Write to UART0 bit rate generator when transmission/reception is halted.

Figure 2.5.7. Set-up procedure of transmission in clock-synchronous serial I/O mode (2)

2.5.3 Operation of the Serial I/O (transmission in clock-synchronous serial I/O mode, transfer clock output from multiple pins function selected)

In transmitting data in clock-synchronous serial I/O mode, choose functions from those listed in Table 2.5.2. Operations of the circled items are described below. Figure 2.5.8 shows the operation timing, and Figures 2.5.9 and 2.5.10 show the set-up procedures.

Table 2.5.2. Choosed functions

Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32} / f_C)
		External clock (CLK0 pin)
CLK polarity	O	Output transmission data at the falling edge of the transfer clock
		Output transmission data at the rising edge of the transfer clock
Transfer clock	O	LSB first
		MSB first
Transmission interrupt factor	O	Transmission buffer empty
		Transmission complete
Output transfer clock to multiple pins (Note)		Not selected
	O	Selected

Note: This can be selected only when UART0 is used in combination with the internal clock.

Operation

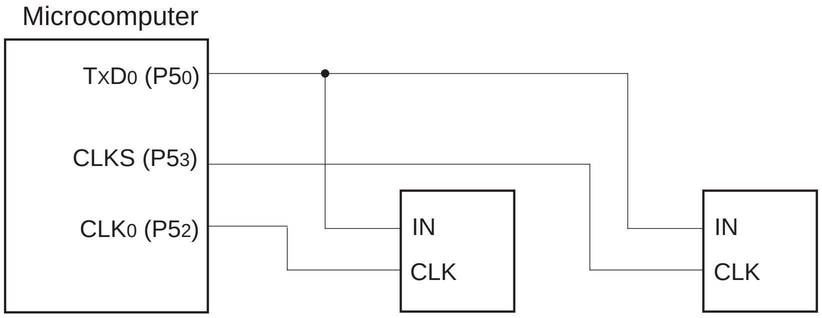
(1) Setting the transmit enable bit to “1” makes data transmissible status ready.

(2) When transmission data is written to the UART0 transmit buffer register, transmission data held in the UART0 transmit buffer register is transmitted to the UART0 transmit register in synchronization with the first falling edge of the transfer clock. At this time, the first bit of the transmission data is transmitted from the TxD_0 pin. Then the data is transmitted bit by bit from the lower order in synchronization with the falling edges of the transfer clock.

(3) When transmission of 1-byte data is completed, the transmit register empty flag goes to “1”, which indicates that the transmission is completed. The transfer clock stops at “H” level. At this time, the UART0 transmit interrupt request bit goes to “1”.

(4) Setting CLK/CLKS select bit 1 to “1” and setting CLK/CLKS select bit 0 to “1” causes the CLKS pin to go to the transfer clock output pin. Change the transfer clock output pin when transmission is halted.

Example of wiring



Note: This applies when performing only transmission with an internal clock selected in the clock synchronous serial I/O mode.

Example of operation

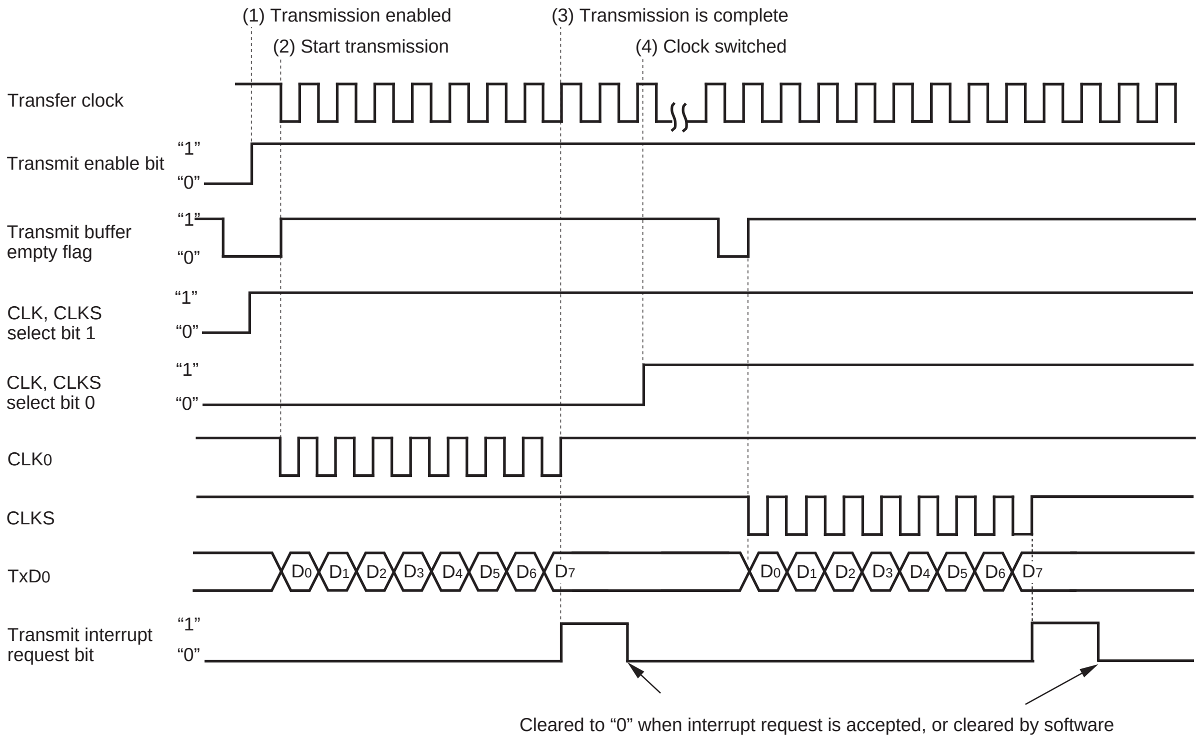
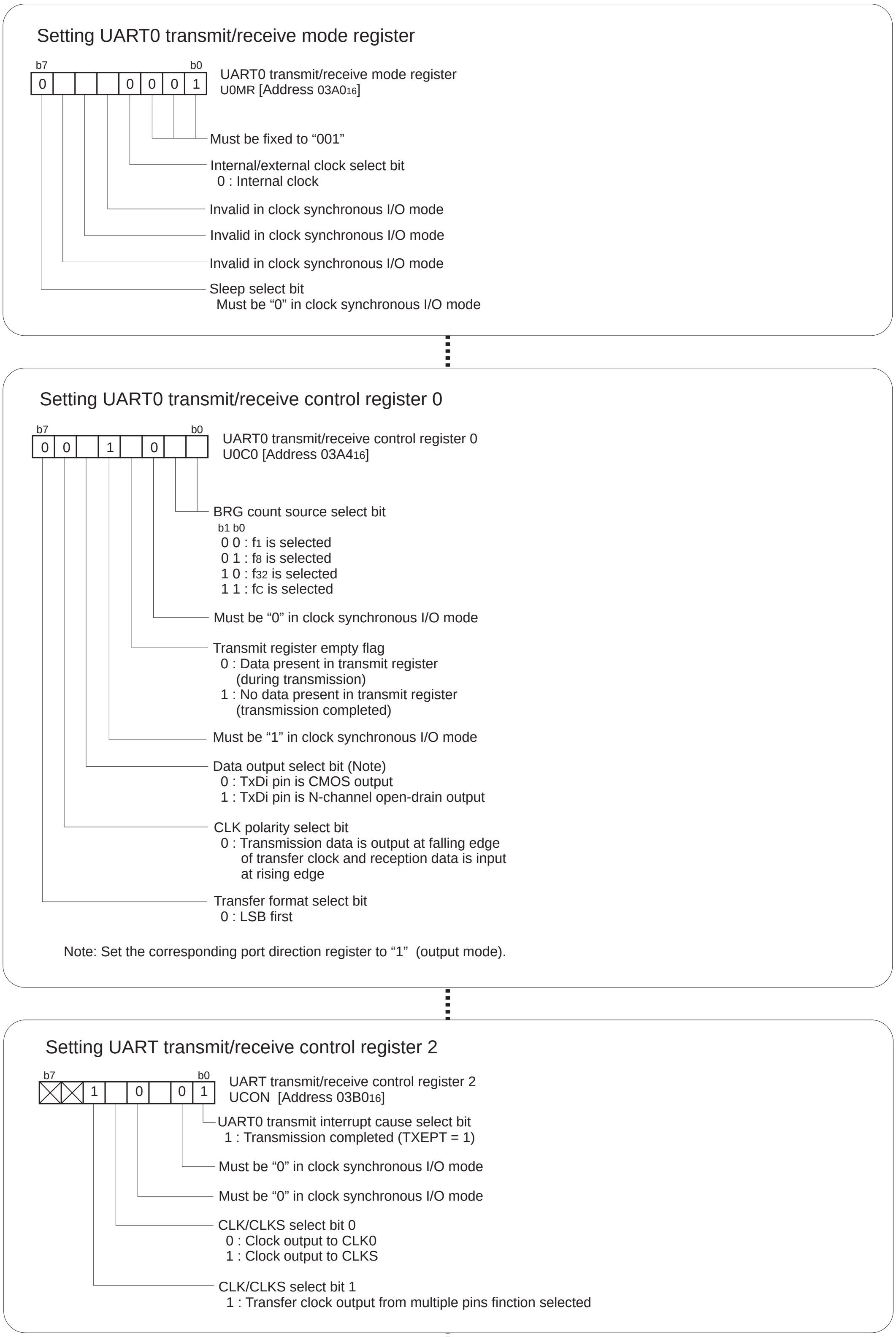


Figure 2.5.8. Operation timing of transmission in clock-synchronous serial I/O mode, transfer clock output from multiple pins function selected


Setting UART0 transmit/receive mode register
b7
b0
0
0 0 0 1
UART0 transmit/receive mode register
U0MR [Address 03A0$_{16}$]
Must be fixed to “001”
Internal/external clock select bit
0 : Internal clock
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Invalid in clock synchronous I/O mode
Sleep select bit
Must be “0” in clock synchronous I/O mode
Setting UART0 transmit/receive control register 0
b7
b0
0 0 1 0
UART0 transmit/receive control register 0
U0C0 [Address 03A4$_{16}$]
BRG count source select bit
b1 b0
0 0 : f$_1$ is selected
0 1 : f$_8$ is selected
1 0 : f$_{32}$ is selected
1 1 : fc is selected
Must be “0” in clock synchronous I/O mode
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
Must be “1” in clock synchronous I/O mode
Data output select bit (Note)
0 : TxDi pin is CMOS output
1 : TxDi pin is N-channel open-drain output
CLK polarity select bit
0 : Transmission data is output at falling edge of transfer clock and reception data is input at rising edge
Transfer format select bit
0 : LSB first
Note: Set the corresponding port direction register to “1” (output mode).
Setting UART transmit/receive control register 2
b7
b0
1 0 0 1
UART transmit/receive control register 2
UCON [Address 03B0$_{16}$]
UART0 transmit interrupt cause select bit
1 : Transmission completed (TXEPT = 1)
Must be “0” in clock synchronous I/O mode
Must be “0” in clock synchronous I/O mode
CLK/CLKS select bit 0
0 : Clock output to CLK0
1 : Clock output to CLKS
CLK/CLKS select bit 1
1 : Transfer clock output from multiple pins finction selected

Continued to the next page

Figure 2.5.9. Set-up procedure of transmission in clock-synchronous serial I/O mode, transfer clock output from multiple pins function selected (1)

Under development

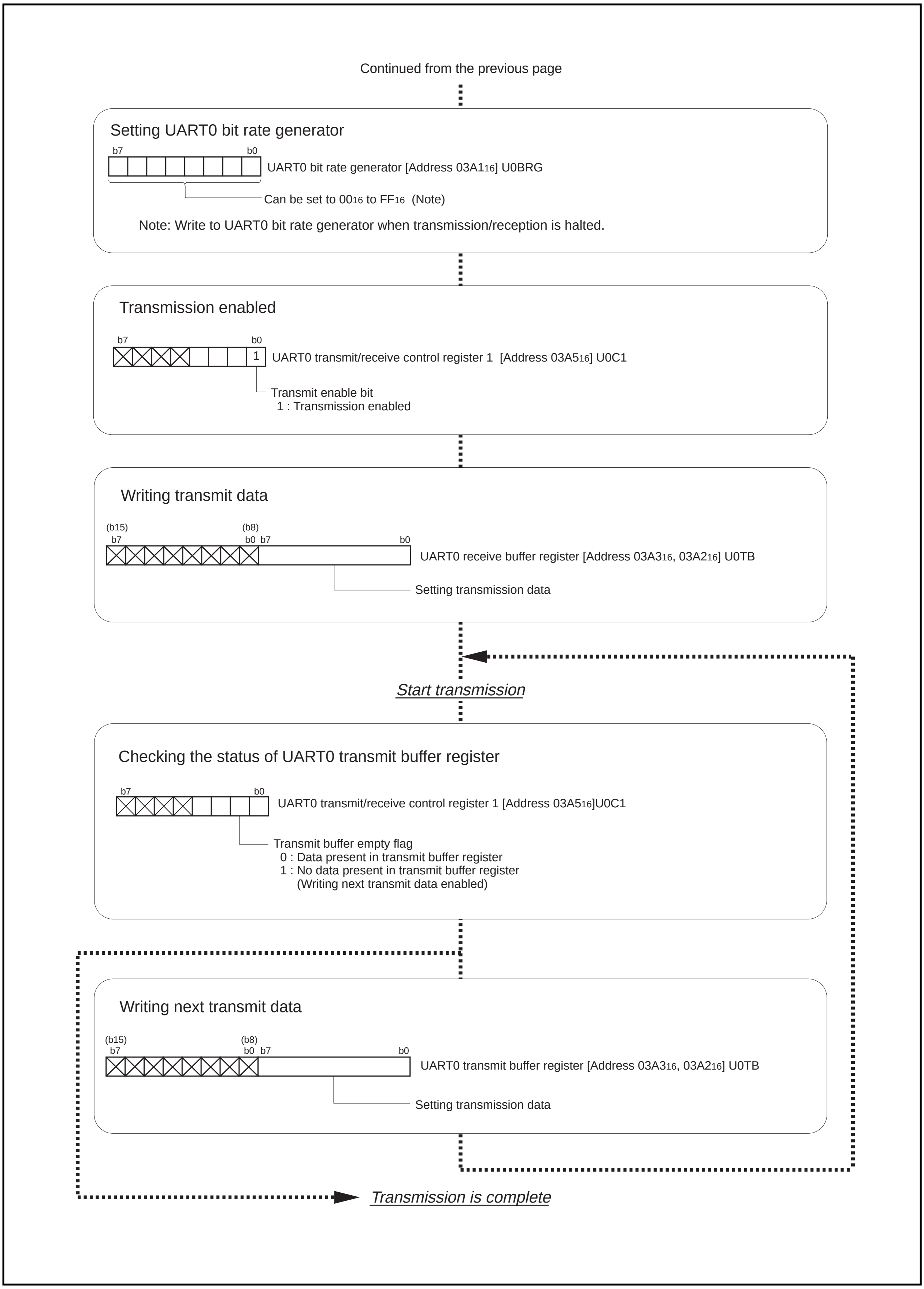

Continued from the previous page
Setting UART0 bit rate generator
b7
b0
UART0 bit rate generator [Address 03A1$_{16}$] U0BRG
Can be set to 00$_{16}$ to FF$_{16}$ (Note)
Note: Write to UART0 bit rate generator when transmission/reception is halted.
Transmission enabled
b7
b0
1
UART0 transmit/receive control register 1 [Address 03A5$_{16}$] U0C1
Transmit enable bit
1 : Transmission enabled
Writing transmit data
(b15)
b7
(b8)
b0 b7
b0
UART0 receive buffer register [Address 03A3$_{16}$, 03A2$_{16}$] U0TB
Setting transmission data
Start transmission
Checking the status of UART0 transmit buffer register
b7
b0
UART0 transmit/receive control register 1 [Address 03A5$_{16}$]U0C1
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register
(Writing next transmit data enabled)
Writing next transmit data
(b15)
b7
(b8)
b0 b7
b0
UART0 transmit buffer register [Address 03A3$_{16}$, 03A2$_{16}$] U0TB
Setting transmission data
Transmission is complete

Figure 2.5.10. Set-up procedure of transmission in clock-synchronous serial I/O mode, transfer clock output from multiple pins function selected (2)

2.5.4 Operation of Serial I/O (reception in clock-synchronous serial I/O mode)

In receiving data in clock-synchronous serial I/O mode, choose functions from those listed in Table 2.5.3. Operations of the circled items are described below. Figure 2.5.11 shows the operation timing, and Figures 2.5.12 and 2.5.13 show the set-up procedures.

Table 2.5.3. Choosed functions

Item	Set-up	
Transfer clock source		Internal clock (f_1 / f_8 / f_{32} / f_c)
	O	External clock (CLK0 pin)
CLK polarity	O	Output transmission data at the falling edge of the transfer clock
		Output transmission data at the rising edge of the transfer clock
Transfer clock	O	LSB first
		MSB first
Continuous receive mode	O	Disabled
		Enabled
Output transfer clock to multiple pins (Note)	O	Not selected
		Selected

Note: This can be selected only when UART0 is used in combination with the internal clock.

Operation (1) Writing dummy data to the UART0 transmit buffer register, setting the receive enable bit to “1”, and the transmit enable bit to “1”, makes the data receivable status ready.

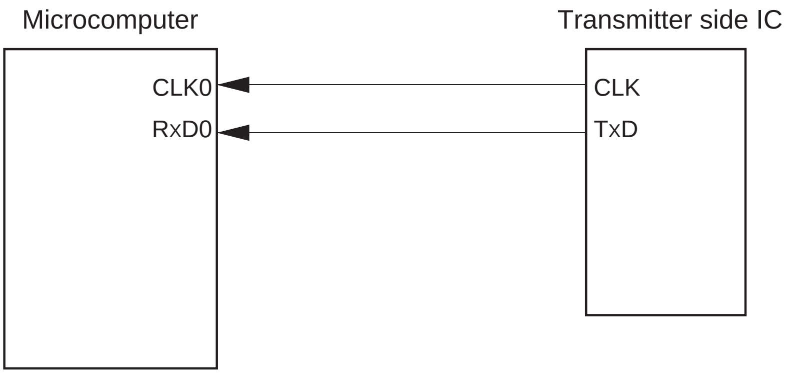
(2) In synchronization with the first rising edge of the transfer clock, the input signal to the RxD0 pin is stored in the highest bit of the UART0 receive register. Then, data is taken in by shifting right the content of the UART0 reception data in synchronization with the rising edges of the transfer clock.

(3) When 1-byte data lines up in the UART0 receive register, the content of the UART0 receive register is transmitted to the UART0 receive buffer register. The transfer clock stops at “H” level. At this time, the receive complete flag and the UART0 receive interrupt request bit goes to “1”.

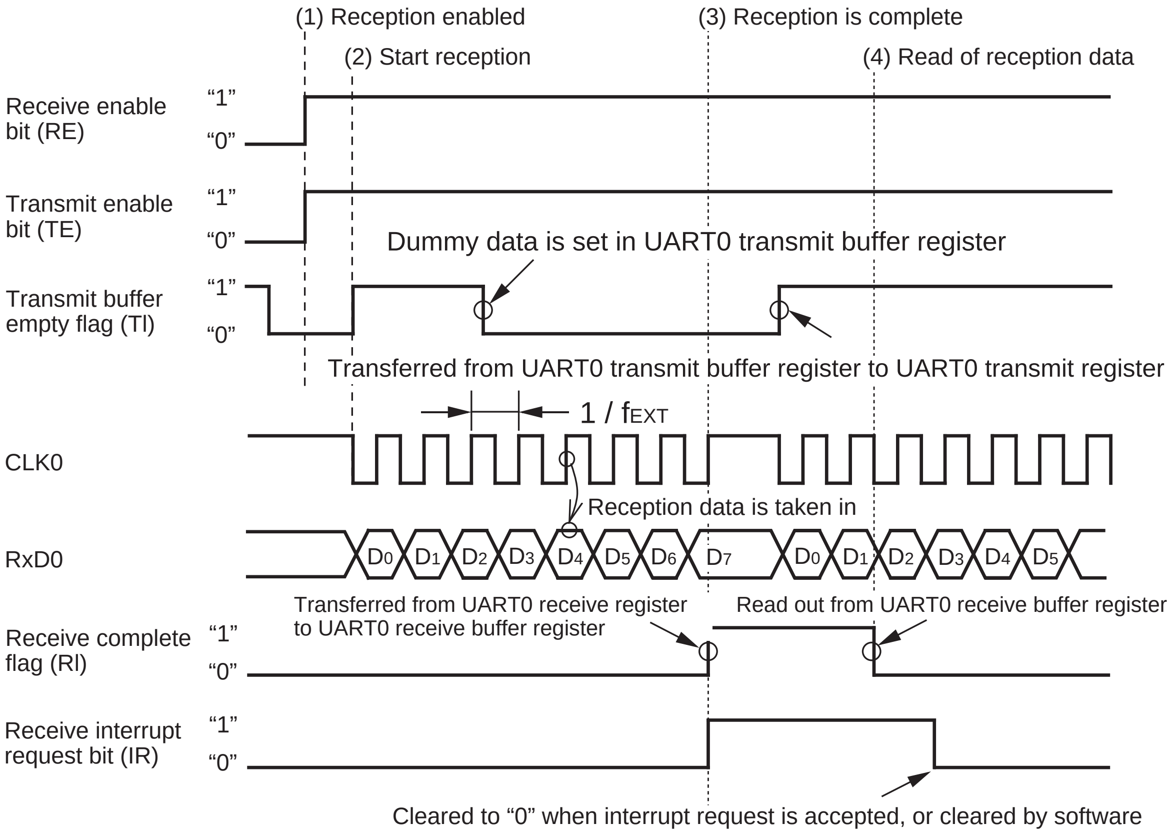
(4) The receive complete flag goes to “0” when the lower-order byte of the UART0 buffer register is read.

Under development

Example of wiring



Example of operation


(1) Reception enabled
(2) Start reception
(3) Reception is complete
(4) Read of reception data
Receive enable bit (RE)
"1"
"0"
Transmit enable bit (TE)
"1"
"0"
Dummy data is set in UART0 transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UART0 transmit buffer register to UART0 transmit register
1 / fEXT
CLK0
Reception data is taken in
RxD0
D0
D1
D2
D3
D4
D5
D6
D7
D0
D1
D2
D3
D4
D5
Transferred from UART0 receive register to UART0 receive buffer register
Read out from UART0 receive buffer register
Receive complete flag (RI)
"1"
"0"
Receive interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

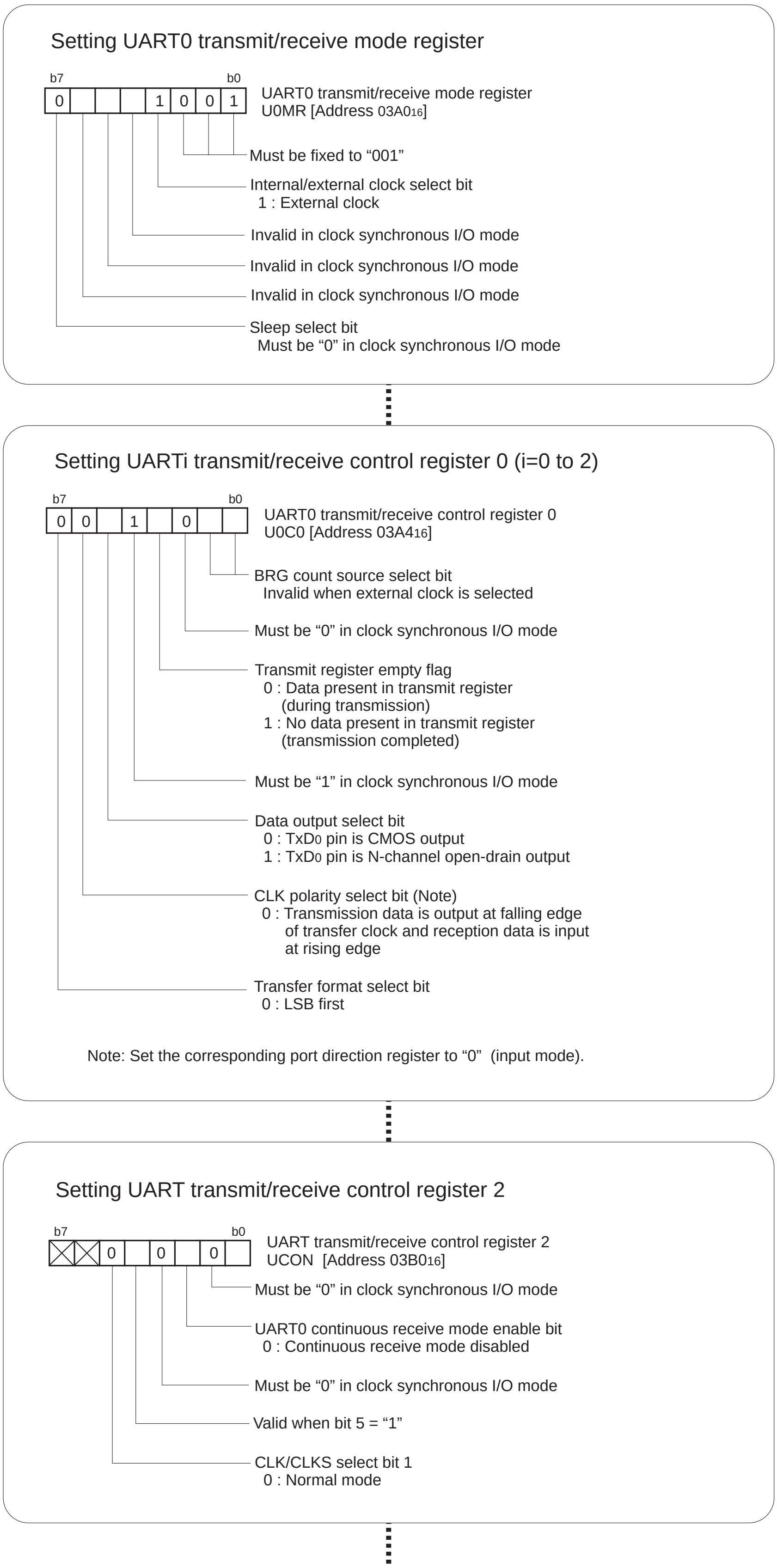
The above timing applies to the following settings:
- External clock is selected.
- CLK polarity select bit = "0".

Make sure that the following conditions are met when the CLK0 pin input ="H" before data reception
- Transmit enable bit fi "1"
- Receive enable bit fi "1"
- Dummy data write to UART0 transmit buffer register

fEXT: frequency of external clock

Figure 2.5.11. Operation timing of reception in clock-synchronous serial I/O mode

Under development



Setting UART0 transmit/receive mode register

UART0 transmit/receive mode register
U0MR [Address 03A0$_{16}$]

Must be fixed to “001”

Internal/external clock select bit
1 : External clock

Invalid in clock synchronous I/O mode

Invalid in clock synchronous I/O mode

Invalid in clock synchronous I/O mode

Sleep select bit
Must be “0” in clock synchronous I/O mode

Setting UARTi transmit/receive control register 0 (i=0 to 2)

UART0 transmit/receive control register 0
U0C0 [Address 03A4$_{16}$]

BRG count source select bit
Invalid when external clock is selected

Must be “0” in clock synchronous I/O mode

Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)

Must be “1” in clock synchronous I/O mode

Data output select bit
0 : TxD$_0$ pin is CMOS output
1 : TxD$_0$ pin is N-channel open-drain output

CLK polarity select bit (Note)
0 : Transmission data is output at falling edge of transfer clock and reception data is input at rising edge

Transfer format select bit
0 : LSB first

Note: Set the corresponding port direction register to “0” (input mode).

Setting UART transmit/receive control register 2

UART transmit/receive control register 2
UCON [Address 03B0$_{16}$]

Must be “0” in clock synchronous I/O mode

UART0 continuous receive mode enable bit
0 : Continuous receive mode disabled

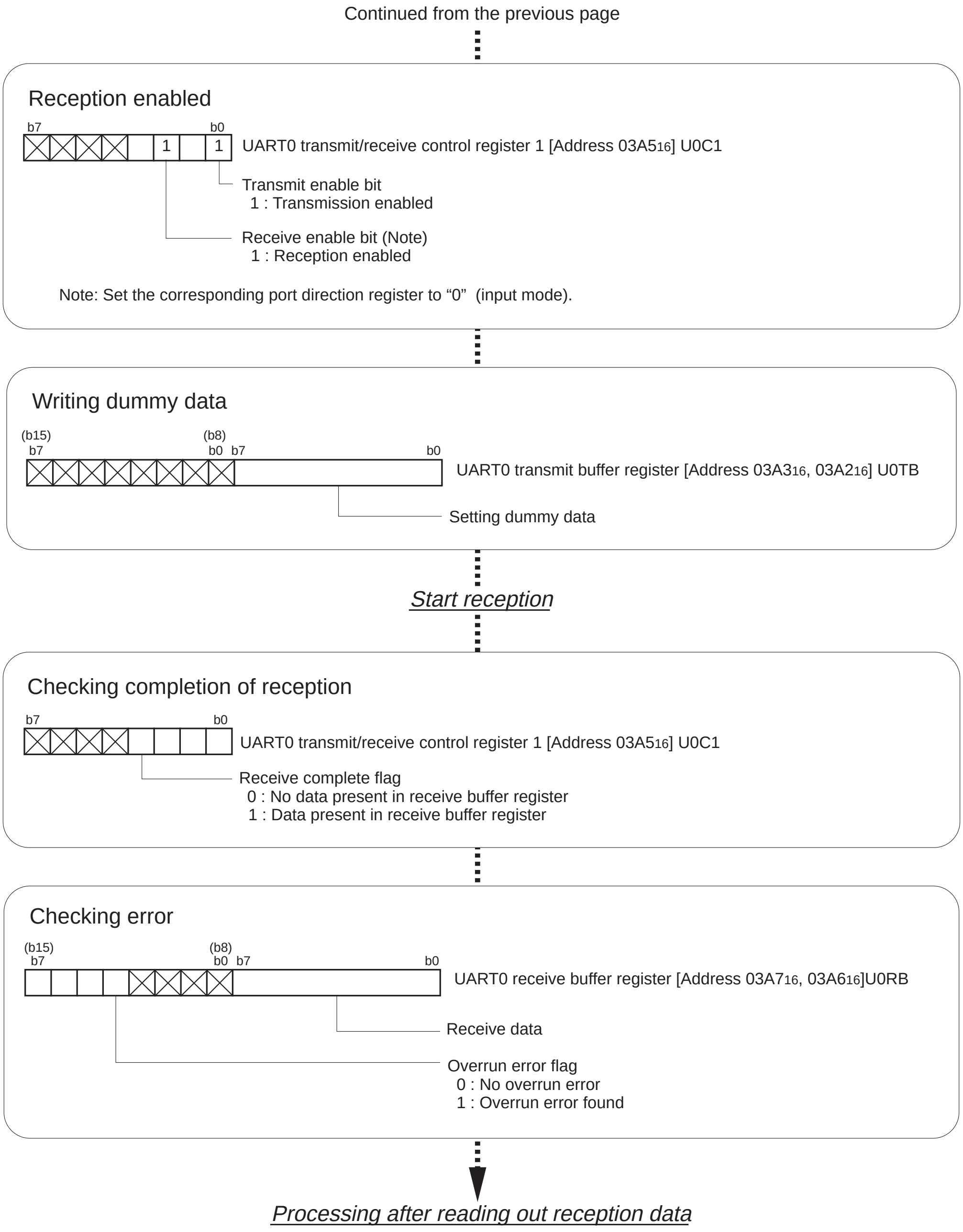
Must be “0” in clock synchronous I/O mode

Valid when bit 5 = “1”

CLK/CLKS select bit 1
0 : Normal mode

Continued to the next page

Figure 2.5.12. Set-up procedure of reception in clock-synchronous serial I/O mode (1)


Continued from the previous page
Reception enabled
b7
b0
1
1
UART0 transmit/receive control register 1 [Address 03A516] U0C1
Transmit enable bit
1 : Transmission enabled
Receive enable bit (Note)
1 : Reception enabled
Writing dummy data
(b15)
b7
(b8)
b0
b7
b0
UART0 transmit buffer register [Address 03A316, 03A216] U0TB
Setting dummy data
Start reception
Checking completion of reception
b7
b0
UART0 transmit/receive control register 1 [Address 03A516] U0C1
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15)
b7
(b8)
b0
b7
b0
UART0 receive buffer register [Address 03A716, 03A616]U0RB
Receive data
Overrun error flag
0 : No overrun error
1 : Overrun error found
Processing after reading out reception data

Note: Set the corresponding port direction register to “0” (input mode).

Figure 2.5.13. Set-up procedure of reception in clock-synchronous serial I/O mode (2)

2.5.5 Precautions for Serial I/O (in clock-synchronous serial I/O)

Transmission

(1) With an external clock selected, perform the following set-up procedure with the CLK0 pin input level = “H” if the CLK polarity select bit = “0” or with the CLK0 pin input level = “L” if the CLK polarity select bit = “1”:

1. Set the transmit enable bit (to “1”)
2. Write transmission data to the UART0 transmit buffer register

Reception

(1) In operating the clock-synchronous serial I/O, operating a transmitter generates a shift clock. Fix settings for transmission even when using the device only for reception. Dummy data is output to the outside from the TxD_0 pin (transmission pin) when receiving data.

(2) With the internal clock selected, setting the transmit enable bit to “1” (transmission-enabled status) and setting dummy data in the UART0 transmission buffer register generates a shift clock.
With the external clock selected, a shift clock is generated when the transmit enable bit is set to “1”, dummy data is set in the UART0 transmit buffer register, and the external clock is input to the CLK0 pin.

(3) In receiving data in succession, an overrun error occurs when the next reception data is made ready in the UART0 receive register with the receive complete flag set to “1” (before the content of the UART0 receive buffer register is read), and overrun error flag is set to “1”. In this instance, the next data is written to the UART0 receive buffer register, so handle with this problem by writing programs on transmission side and reception side so that the previous data is transmitted again.
If an overrun error occurs, the UART0 receive interrupt request bit does not go to “1”.

(4) To receive data in succession, set dummy data in the lower-order byte of the UART0 transmit buffer register every time reception is made.

(5) With an external clock selected, perform the following set-up procedure with the CLK0 pin input level = “H” if the CLK polarity select bit = “0” or with the CLK0 pin input level = “L” if the CLK polarity select bit = “1”:

1. Set receive enable bit (to “1”)
2. Set transmit enable bit (to “1”)
3. Write dummy data to the UART0 transmit buffer register

2.6 Clock-Asynchronous Serial I/O (UART)

2.6.1 Overview

UART handles communications by means of character-by-character synchronization. The transmission side and the reception side are independent of each other, so full-duplex communication is possible. The following is an overview of the clock-asynchronous serial I/O.

(1) Transmission/reception format

Figure 2.6.1 shows the transmission/reception format, and Table 2.6.1 shows the names and functions of transmission data.

- Transfer data length : 7 bits
  - 1ST – 7DATA – 1SP
  - 1ST – 7DATA – 2SP
  - 1ST – 7DATA – 1PAR – 1SP
  - 1ST – 7DATA – 1PAR – 2SP
- Transfer data length : 8 bits
  - 1ST – 8DATA – 1SP
  - 1ST – 8DATA – 2SP
  - 1ST – 8DATA – 1PAR – 1SP
  - 1ST – 8DATA – 1PAR – 2SP
- Transfer data length : 9 bits
  - 1ST – 9DATA – 1SP
  - 1ST – 9DATA – 2SP
  - 1ST – 9DATA – 1PAR – 1SP
  - 1ST – 9DATA – 1PAR – 2SP

ST : Start bit
DATA : Character bit (Transfer data)
PAR : Parity bit
SP : Stop bit

Figure 2.6.1. Transmission/reception format

Table 2.6.1. Transmission data names and functions

Name	Function
ST (start bit)	A 1-bit “L” signal to be added immediately before character bits. This bit signals the start of data transmission.
DATA (character bits)	Transmission data set in the UARTi transmit buffer register.
PAR (parity bit)	A signal to be added immediately after character bits so as to increase data reliability. The level of this signal so varies that the total number of 1's in character bits and this bit always becomes even or odd depending on which parity is chosen, even or odd.
SP (stop bit)	Either 1-bit or 2-bit “H” signal to be added immediately after character bits (after the parity bit if parity is checked). This / they signals the end of data transmission.

(2) Transfer rate

The divide-by-16 frequency, resulting from division in the bit rate generator (BRG), becomes the transfer rate. The count source for the transfer rate register can be selected from f_1, f_8, f_{32}, and the input from the CLK pin. Clocks f_1, f_8, f_{32} are derived by dividing the CPU's main clock by 1, 8, and 32 respectively.

Table 2.6.2. Example of baud rate setting

Baud rate (bps)	BRG's count source	System clock : 10MHz		System clock : 7.3728MHz	
		BRG's set value : n	Actual time (bps)	BRG's set value : n	Actual time (bps)
600	f_8	129 (81_{16})	600	95 ($5F_{16}$)	600
1200	f_8	64 (40_{16})	1201	47 ($2F_{16}$)	1200
2400	f_8	32 (20_{16})	2367	23 (17_{16})	2400
4800	f_1	129 (81_{16})	4807	95 ($5F_{16}$)	4800
9600	f_1	64 (40_{16})	9615	47 ($2F_{16}$)	9600
14400	f_1	42 ($2A_{16}$)	14534	31 ($1F_{16}$)	14400
19200	f_1	32 (20_{16})	18939	23 (17_{16})	19200
28800	f_1	21 (15_{16})	28409	15 (F_{16})	28800
31250	f_1	19 (13_{16})	31250		

(3) An error detection

In clock-asynchronous serial I/O mode, detect errors are shown in Table 2.6.3.

Table 2.6.3. Error detection

Type of error	Description	When the flag turns on	How to clear the flag
Overrun error	• This error occurs when the next data lines up before the content of the UARTi receive buffer register is read. • The next data is written to the UARTi receive buffer register. • The UARTi receive interrupt request bit does not go to “1”.	The error is detected when data is transferred from the UARTi receive register to the UARTi receive buffer register.	• Set the serial I/O mode select bits to “000$_2$”. • Set the receive enable bit to “0”.
Framing error	• This error occurs when the stop bit falls short of the set number of stop bits.		• Set the serial I/O mode select bits to ”000$_2$”. • Set the receive enable bit to “0”. • Read the lower-order byte of the UARTi receive buffer register.
Parity error	• With parity enabled, this error occurs when the total number of 1's in character bits and the parity bit is different from the specified number.		
Error-sum flag	• This flag turns on when any error (overrun, framing, or parity) is detected.		• When all error (overrun, framing, and parity) are removed, the flag is cleared.

(4) How to deal with an error

When receiving data, read an error flag and reception data simultaneously to determine which error has occurred. If the data read is erroneous, initialize the error flag and the UARTi receive buffer register, then receive the data again.

To initialize the UARTi receive buffer register

1. Set the receive enable bit to “0” (disable reception).
2. Set the receive enable bit to “1” again (enable reception).

To transmit data again due to an error on the reception side, set the UARTi transmit buffer register again, then transmit the data again.

To set the UARTi transmit buffer register again

1. Set the serial I/O mode select bits to “000$_2$” (invalidate serial I/O).
2. Set the serial I/O mode select bits again.
3. Set the transmit enable bit to “1” (enable transmission), then set transmission data in the UARTi transmit buffer register.

Under development

(5) Functions selection

In operating UART, the following functions can be used:

(a) Sleep mode

Sleep mode is a mode in which data is transferred to a particular microcomputer among those connected by use of clock-asynchronous serial I/O devices.

The following are examples in which functions (a) to (e) are chosen:

- Transmission WITHOUT: other functions P264
- Reception WITHOUT: other functions P268

(6) Input/output to the serial I/O and the direction register

To input an external signal to the serial I/O, set the direction register of the relevant port to input. To output a signal from the serial I/O, set the direction register of the relevant port to output.

(7) Pins related to the serial I/O

- CLK$_0$ pins :Input pins for the transfer clock
- RxD$_0$, RxD$_1$ pins :Input pins for data
- TxD$_0$, TxD$_1$ pins :Output pins for data

Under development

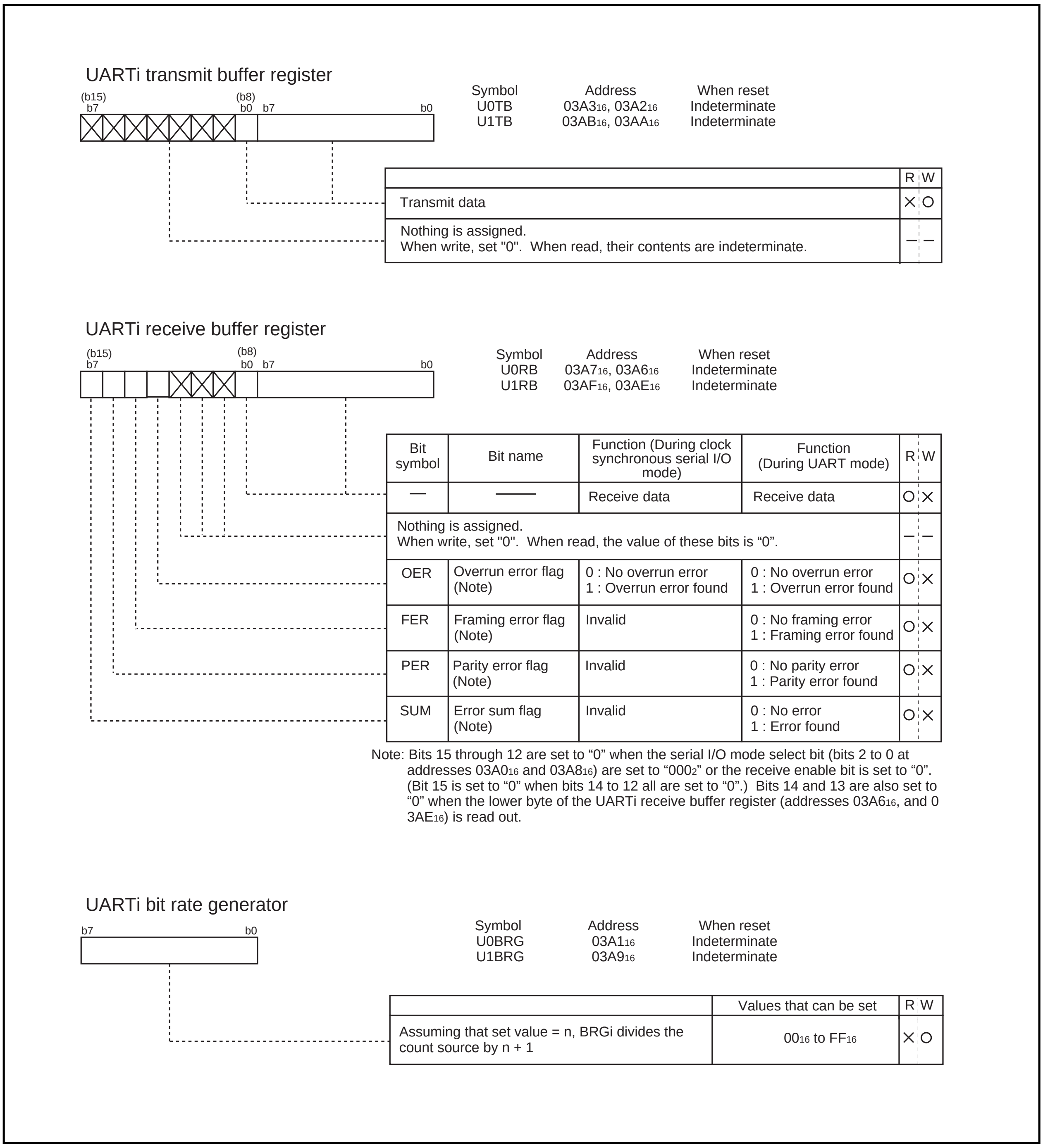
(8) Registers related to the serial I/O

Figure 2.6.2 shows the memory map of serial I/O-related registers, and Figures 2.6.3 to 2.6.7 show UARTi-related registers.

0051_{16}	UART0 transmit interrupt control register (S0TIC)
0052_{16}	UART0 receive interrupt control register (S0RIC)
0053_{16}	UART1 transmit interrupt control regster(S1TIC)
0054_{16}	UART1 receive interrupt control register(S1RIC)
≈	≈
$03A0_{16}$	UART0 transmit/receive mode register (U0MR)
$03A1_{16}$	UART0 bit rate generator (U0BRG)
$03A2_{16}$ $03A3_{16}$	UART0 transmit buffer register (U0TB)
$03A4_{16}$	UART0 transmit/receive control register 0 (U0C0)
$03A5_{16}$	UART0 transmit/receive control register 1 (U0C1)
$03A6_{16}$ $03A7_{16}$	UART0 receive buffer register (U0RB)
$03A8_{16}$	UART1 transmit/receive mode register (U1MR)
$03A9_{16}$	UART1 bit rate generator (U1BRG)
$03AA_{16}$ $03AB_{16}$	UART1 transmit buffer register (U1TB)
$03AC_{16}$	UART1 transmit/receive control register 0 (U1C0)
$03AD_{16}$	UART1 transmit/receive control register 1 (U1C1)
$03AE_{16}$ $03AF_{16}$	UART1 receive buffer register (U1RB)
$03B0_{16}$	UART transmit/receive control register 2 (UCON)

Figure 2.6.2. Memory map of UARTi-related registers

Under development



UARTi transmit buffer register

Symbol	Address	When reset
U0TB	$03A3_{16}$, $03A2_{16}$	Indeterminate
U1TB	$03AB_{16}$, $03AA_{16}$	Indeterminate

	R	W
Transmit data	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.	–	–

UARTi receive buffer register

Symbol	Address	When reset
U0RB	$03A7_{16}$, $03A6_{16}$	Indeterminate
U1RB	$03AF_{16}$, $03AE_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	——	Receive data	Receive data	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				–	–
OER	Overrun error flag (Note)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note)	Invalid	0 : No error 1 : Error found	○	×

Note: Bits 15 through 12 are set to “0” when the serial I/O mode select bit (bits 2 to 0 at addresses $03A0_{16}$ and $03A8_{16}$) are set to “000_2” or the receive enable bit is set to “0”. (Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses $03A6_{16}$, and 0 $3AE_{16}$) is read out.

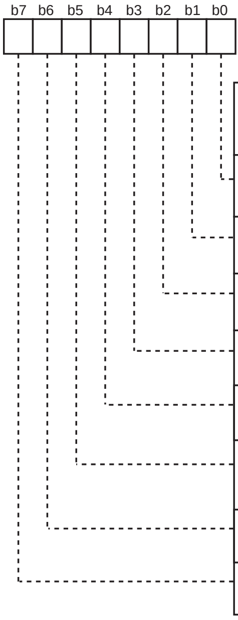
UARTi bit rate generator

Symbol	Address	When reset
U0BRG	$03A1_{16}$	Indeterminate
U1BRG	$03A9_{16}$	Indeterminate

	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00_{16} to FF_{16}	×	○

Figure 2.6.3. UARTi-related registers (1)

UARTi transmit/receive mode register

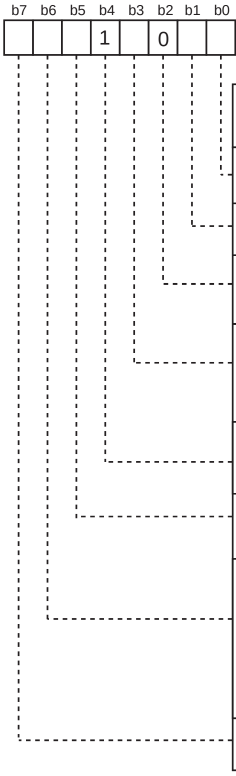


Symbol UiMR(i=0,1) | Address $03A0_{16}$, $03A8_{16}$ | When reset 00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit (Note 1)	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit (Note 2)	0 : Internal clock 1 : External clock	0 : Internal clock 1 : External clock	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Must always be “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: UART1 can use only internal clock. Must set this bit to “1”.

UARTi transmit/receive control register 0



Symbol UiC0(i=0,1) | Address $03A4_{16}$, $03AC_{16}$ | When reset 08_{16}

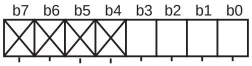
Bit symbol	Bit name	Function (Note) (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : fc is selected	○	○
CLK1				○	○
Set this bit to “0”.				○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
Set this bit to “1”.				○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open-drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open-drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Must always be “0”	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Must always be “0”	○	○

Note: UART1 cannot be used in clock synchronous serial I/O.

Figure 2.6.4. UARTi-related registers (2)

Under development

UARTi transmit/receive control register 1

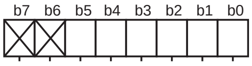


Symbol	Address	When reset
UiC1(i=0,1)	$03A5_{16}$,$03AD_{16}$	02_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				—	—

Note: UART1 cannot be used in clock synchronous serial I/O.

UART transmit/receive control register 2



Symbol	Address	When reset
UCON	$03B0_{16}$	$XX000000_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	Set this bit to “0”.	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Invalid	○	○
Set this bit to “0”.				○	○
CLKMD0	CLK/CLKS select bit 0	Valid when bit 5 = “1” 0 : Clock output to CLK1 1 : Clock output to CLKS1	Invalid	○	○
CLKMD1	CLK/CLKS select bit 1 (Note 2)	0 : Normal mode (CLK output is CLK0 only) 1 : Transfer clock output from multiple pins function selected	Must always be “0”	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				—	—

Note 1: UART1 cannot be used in clock synchronous serial I/O.
Note 2: When using multiple pins to output the transfer clock, the following requirements must be met:
- UART0 internal/external clock select bit (bit 3 at address $03A0_{16}$) = “0”.

Figure 2.6.5. UARTi-related registers (3)

2.6.2 Operation of Serial I/O (transmission in UART mode)

In transmitting data in UART mode, choose functions from those listed in Table 2.6.4. Operations of the circled items are described below. Figure 2.6.6 shows the operation timing, and Figures 2.6.7 and 2.6.8 show the set-up procedures.

Table 2.6.4. Choosed functions

Item	Set-up	
Transfer clock source	O	Internal clock (f_1 / f_8 / f_{32} / f_C)
		External clock (CLK0 pin) (Note)
Transmission interrupt factor		Transmission buffer empty
	O	Transmission complete
Sleep mode	O	Sleep mode off
		Sleep mode selected

Note: UART1 cannot be selected external clock.

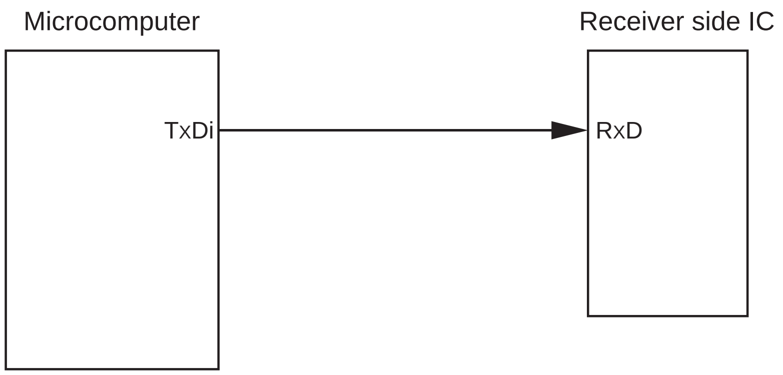
Operation (1) Setting the transmit enable bit to “1” and writing transmission data to the UARTi transmit buffer register readies the data transmissible status.

(2) Transmission data held in the UARTi transmit buffer register is transmitted to the UARTi transmit register. At this time, the first bit (the start bit) of the transmission data is transmitted from the TxDi pin. Then, data is transmitted, bit by bit, in sequence: LSB, ····, MSB, parity bit, and stop bit(s).

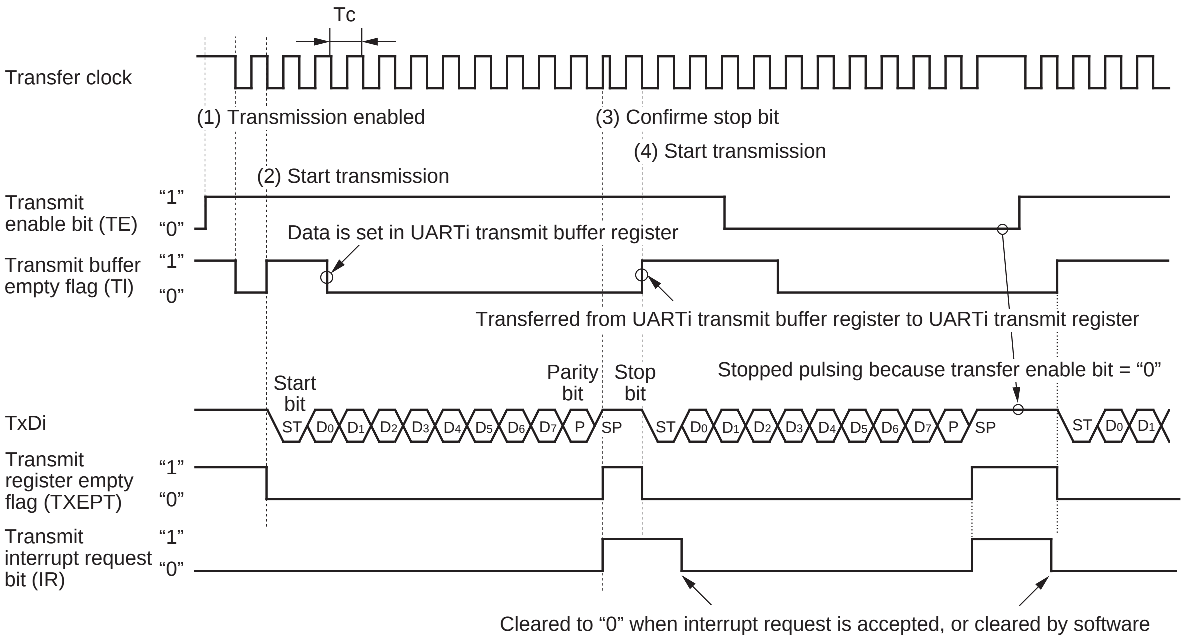
(3) When the stop bit(s) is (are) transmitted, the transmit register empty flag goes to “1”, which indicates that transmission is completed. At this time, the UARTi transmit interrupt request bit goes to “1”. The transfer clock stops at “H” level.

(4) If the transmission condition of the next data is ready when transmission is completed, a start bit is generated following to stop bit(s), and the next data is transmitted.

Example of wiring



Example of operation


Tc
Transfer clock
(1) Transmission enabled
(3) Confirme stop bit
(4) Start transmission
(2) Start transmission
Transmit enable bit (TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Parity bit
Stop bit
Stopped pulsing because transfer enable bit = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
ST D0 D1
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32}, f_C)
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 2.6.6. Operation timing of transmission in UART mode

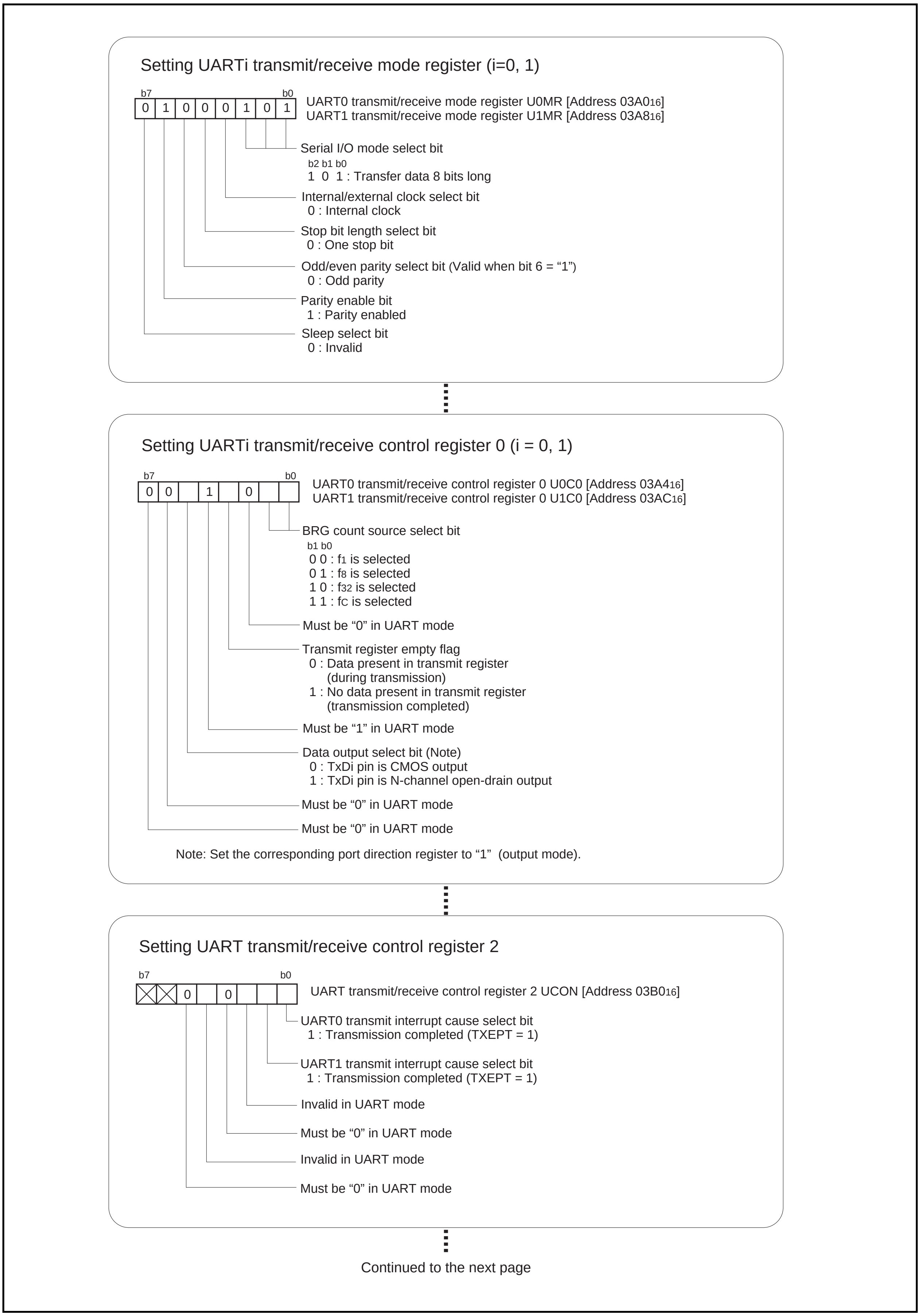

Setting UARTi transmit/receive mode register (i=0, 1)
b7
b0
0 1 0 0 0 1 0 1
UART0 transmit/receive mode register U0MR [Address 03A016]
UART1 transmit/receive mode register U1MR [Address 03A816]
Serial I/O mode select bit
b2 b1 b0
1 0 1 : Transfer data 8 bits long
Internal/external clock select bit
0 : Internal clock
Stop bit length select bit
0 : One stop bit
Odd/even parity select bit (Valid when bit 6 = “1”)
0 : Odd parity
Parity enable bit
1 : Parity enabled
Sleep select bit
0 : Invalid
Setting UARTi transmit/receive control register 0 (i = 0, 1)
b7
b0
0 0 1 0
UART0 transmit/receive control register 0 U0C0 [Address 03A416]
UART1 transmit/receive control register 0 U1C0 [Address 03AC16]
BRG count source select bit
b1 b0
0 0 : f1 is selected
0 1 : f8 is selected
1 0 : f32 is selected
1 1 : fC is selected
Must be “0” in UART mode
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
Must be “1” in UART mode
Data output select bit (Note)
0 : TxDi pin is CMOS output
1 : TxDi pin is N-channel open-drain output
Must be “0” in UART mode
Must be “0” in UART mode
Note: Set the corresponding port direction register to “1” (output mode).
Setting UART transmit/receive control register 2
b7
b0
0 0
UART transmit/receive control register 2 UCON [Address 03B016]
UART0 transmit interrupt cause select bit
1 : Transmission completed (TXEPT = 1)
UART1 transmit interrupt cause select bit
1 : Transmission completed (TXEPT = 1)
Invalid in UART mode
Must be “0” in UART mode
Invalid in UART mode
Must be “0” in UART mode
Continued to the next page

Figure 2.6.7. Set-up procedure of transmission in UART mode (1)

Under development

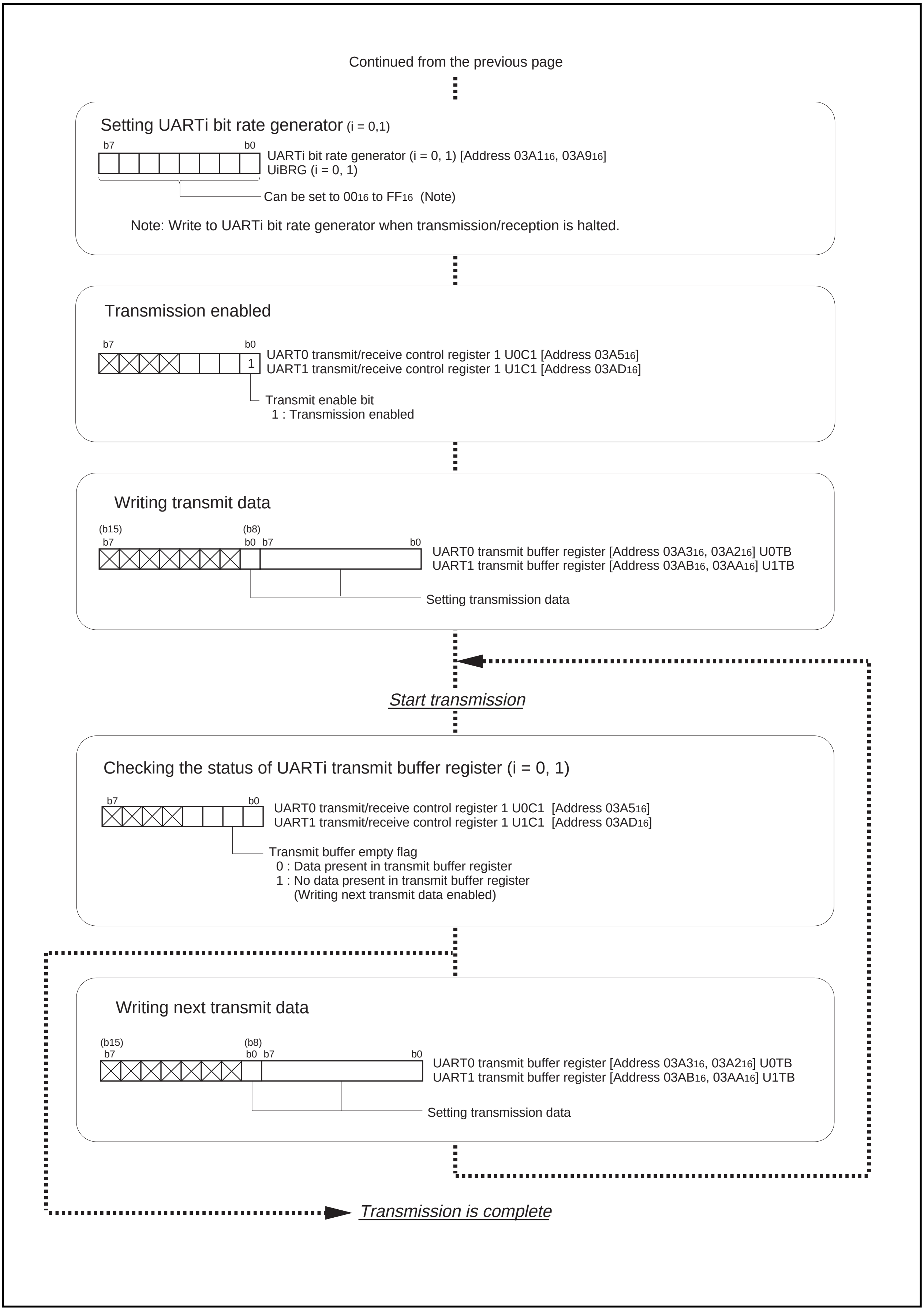

Continued from the previous page
Setting UARTi bit rate generator (i = 0,1)
b7
b0
UARTi bit rate generator (i = 0, 1) [Address $03A1_{16}$, $03A9_{16}$]
UiBRG (i = 0, 1)
Can be set to 00_{16} to FF_{16} (Note)
Note: Write to UARTi bit rate generator when transmission/reception is halted.
Transmission enabled
b7
b0
1
UART0 transmit/receive control register 1 U0C1 [Address $03A5_{16}$]
UART1 transmit/receive control register 1 U1C1 [Address $03AD_{16}$]
Transmit enable bit
1 : Transmission enabled
Writing transmit data
(b15)
b7
(b8)
b0
b7
b0
UART0 transmit buffer register [Address $03A3_{16}$, $03A2_{16}$] U0TB
UART1 transmit buffer register [Address $03AB_{16}$, $03AA_{16}$] U1TB
Setting transmission data
Start transmission
Checking the status of UARTi transmit buffer register (i = 0, 1)
b7
b0
UART0 transmit/receive control register 1 U0C1 [Address $03A5_{16}$]
UART1 transmit/receive control register 1 U1C1 [Address $03AD_{16}$]
Transmit buffer empty flag
0 : Data present in transmit buffer register
1 : No data present in transmit buffer register
(Writing next transmit data enabled)
Writing next transmit data
(b15)
b7
(b8)
b0
b7
b0
UART0 transmit buffer register [Address $03A3_{16}$, $03A2_{16}$] U0TB
UART1 transmit buffer register [Address $03AB_{16}$, $03AA_{16}$] U1TB
Setting transmission data
Transmission is complete

Figure 2.6.8. Set-up procedure of transmission in UART mode (2)

Under development

2.6.3 Operation of Serial I/O (reception in UART mode)

In receiving data in UART mode, choose functions from those listed in Table 2.6.5. Operations of the circled items are described below. Figure 2.6.9 shows the operation timing, and Figures 2.6.10 and 2.6.11 show the set-up procedures.

Table 2.6.5. Choosed functions

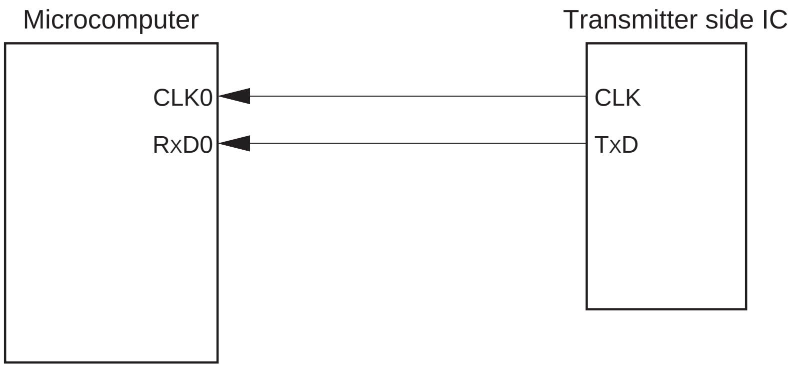
Item	Set-up	
Transfer clock source		Internal clock (f_1 / f_8 / f_{32} / f_C)
	O	External clock (CLK0 pin) (Note)
Sleep mode	**O**	Sleep mode off
		Sleep mode selected

Note: UART1 cannot be selected external clock.

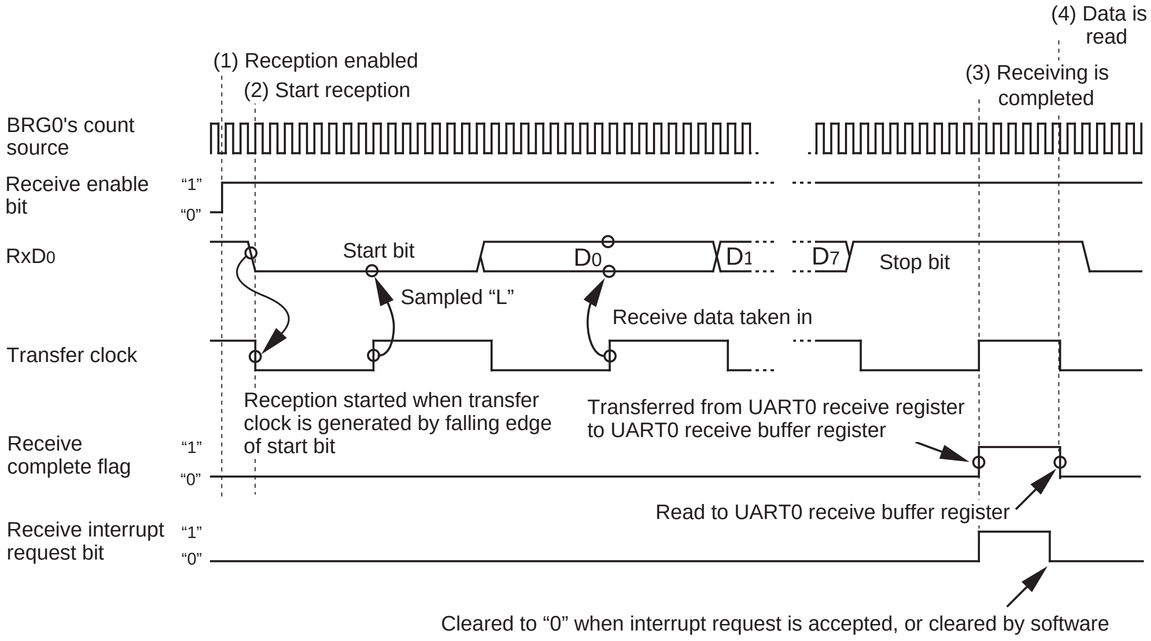
Operation
(1) Setting the receive enable bit to “1” readies data-receivable status.
(2) When the first bit (the start bit) of reception data is received from the RxDi pin. Then, data is received, bit by bit, in sequence: LSB, ····, MSB, and stop bit(s).
(3) When the stop bit(s) is (are) received, the content of the UARTi receive register is transmitted to the UARTi receive buffer register.
At this time, the receive complete flag goes to “1” to indicate that the reception is completed, the UARTi receive interrupt request bit goes to “1”.
(4) The receive complete flag goes to “0” when the lower-order byte of the UARTi buffer register is read.

Under development

Example of wiring



Example of operation


(4) Data is read
(1) Reception enabled
(2) Start reception
(3) Receiving is completed
BRG0's count source
Receive enable bit
"1"
"0"
RxD0
Start bit
D0
D1
D7
Stop bit
Sampled "L"
Receive data taken in
Transfer clock
Reception started when transfer clock is generated by falling edge of start bit
Transferred from UART0 receive register to UART0 receive buffer register
Receive complete flag
"1"
"0"
Read to UART0 receive buffer register
Receive interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Timing of transfer data 8 bits long applies to the following settings :
- Transfer data length is 8 bits.
- Parity is disabled.
- One stop bit

Figure 2.6.9. Operation timing of reception in UART mode

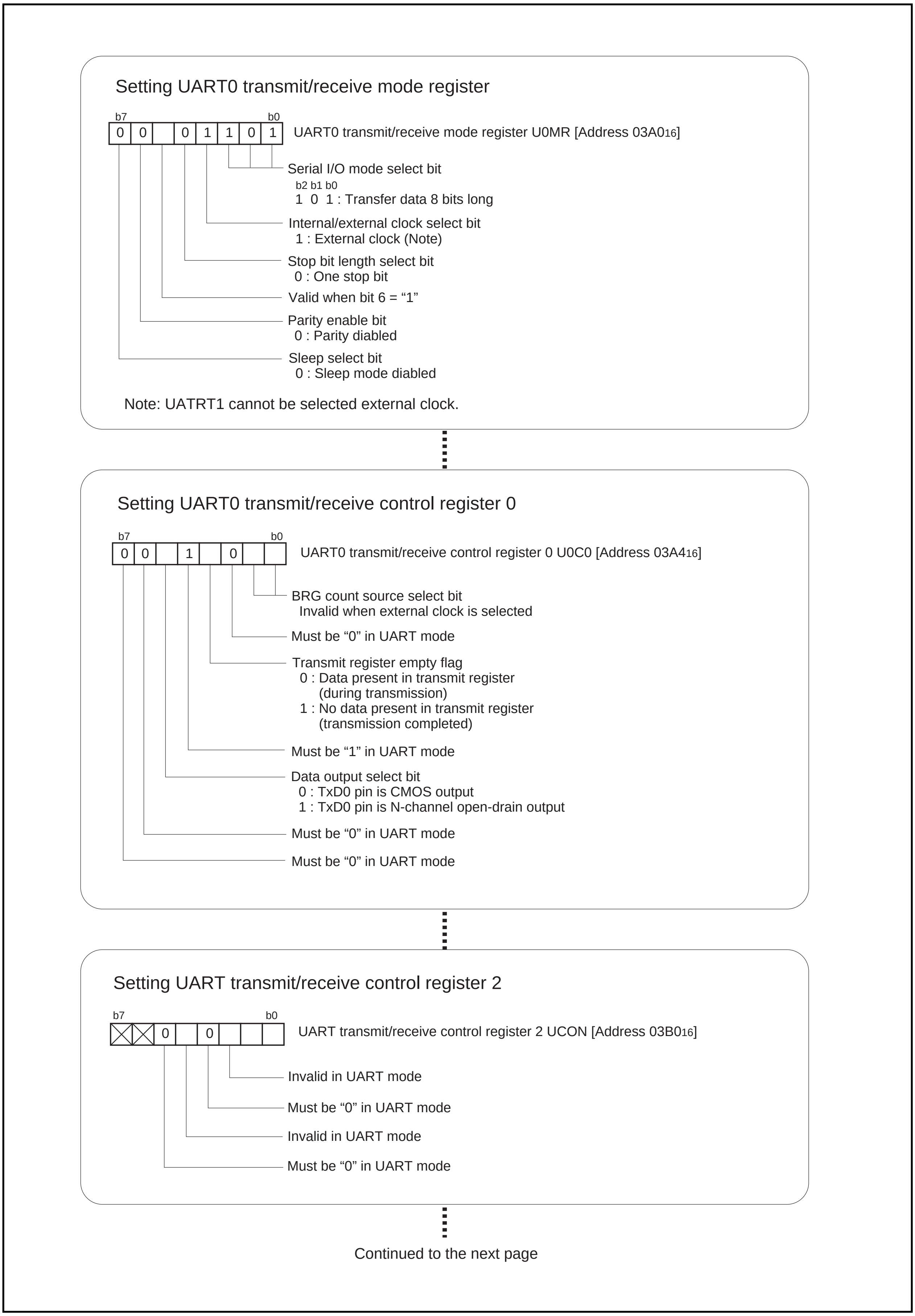
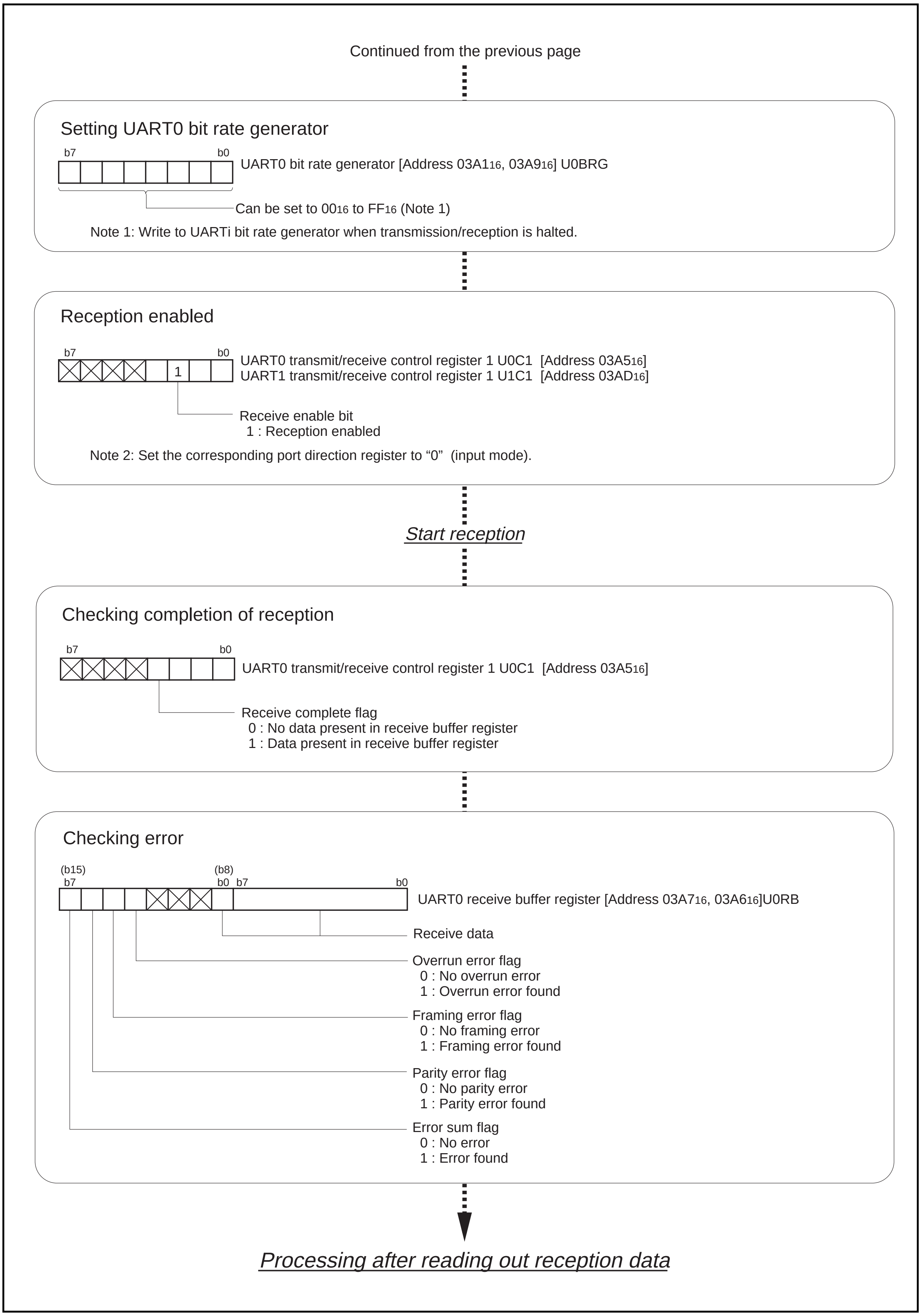

Setting UART0 transmit/receive mode register
b7
b0
0 0 0 1 1 0 1
UART0 transmit/receive mode register U0MR [Address 03A0$_{16}$]
Serial I/O mode select bit
b2 b1 b0
1 0 1 : Transfer data 8 bits long
Internal/external clock select bit
1 : External clock (Note)
Stop bit length select bit
0 : One stop bit
Valid when bit 6 = “1”
Parity enable bit
0 : Parity diabled
Sleep select bit
0 : Sleep mode diabled
Note: UATRT1 cannot be selected external clock.
Setting UART0 transmit/receive control register 0
b7
b0
0 0 1 0
UART0 transmit/receive control register 0 U0C0 [Address 03A4$_{16}$]
BRG count source select bit
Invalid when external clock is selected
Must be “0” in UART mode
Transmit register empty flag
0 : Data present in transmit register (during transmission)
1 : No data present in transmit register (transmission completed)
Must be “1” in UART mode
Data output select bit
0 : TxD0 pin is CMOS output
1 : TxD0 pin is N-channel open-drain output
Must be “0” in UART mode
Must be “0” in UART mode
Setting UART transmit/receive control register 2
b7
b0
0 0
UART transmit/receive control register 2 UCON [Address 03B0$_{16}$]
Invalid in UART mode
Must be “0” in UART mode
Invalid in UART mode
Must be “0” in UART mode
Continued to the next page

Figure 2.6.10. Set-up procedure of reception in UART mode (1)

Under development


Continued from the previous page
Setting UART0 bit rate generator
b7
b0
UART0 bit rate generator [Address 03A1$_{16}$, 03A9$_{16}$] U0BRG
Can be set to 00$_{16}$ to FF$_{16}$ (Note 1)
Reception enabled
b7
b0
1
UART0 transmit/receive control register 1 U0C1 [Address 03A5$_{16}$]
UART1 transmit/receive control register 1 U1C1 [Address 03AD$_{16}$]
Receive enable bit
1 : Reception enabled
Start reception
Checking completion of reception
b7
b0
UART0 transmit/receive control register 1 U0C1 [Address 03A5$_{16}$]
Receive complete flag
0 : No data present in receive buffer register
1 : Data present in receive buffer register
Checking error
(b15)
b7
(b8)
b0
b7
b0
UART0 receive buffer register [Address 03A7$_{16}$, 03A6$_{16}$]U0RB
Receive data
Overrun error flag
0 : No overrun error
1 : Overrun error found
Framing error flag
0 : No framing error
1 : Framing error found
Parity error flag
0 : No parity error
1 : Parity error found
Error sum flag
0 : No error
1 : Error found
Processing after reading out reception data

Note 1: Write to UARTi bit rate generator when transmission/reception is halted.

Note 2: Set the corresponding port direction register to “0” (input mode).

Figure 2.6.11. Set-up procedure of reception in UART mode (2)

2.7 A-D Converter

2.7.1 Overview

The A-D converter used in the M16C/60 group operates on a successive conversion basis. The following is an overview of the A-D converter.

(1) Mode

The A-D converter operates in one of five modes:

(a) One-shot mode

Carries out A-D conversion on input level of one specified pin only once.

(b) Repetition mode

Repeatedly carries out A-D conversion on input level of one specified pin.

(c) Single sweep mode

Carries out A-D conversion on input level of two or more specified pins only once.

(d) Repeated sweep mode 0

Repeatedly carries out A-D conversion on input level of two or more pins.

(e) Repeated sweep mode 1

Repeatedly carries out A-D conversion on input level of two or more pins. This mode is different from the repeated sweep mode 0 in that weights can be assigned to specifing pins control the number of conversion times.

(2) Operation clock

The operation clock in 5 V operation can be selected from the following: f_{AD}, divide-by-2 f_{AD}, and divide-by-4 f_{AD}. In 3 V operation, the selection is divide-by-2 f_{AD} or divide-by-4. The f_{AD} frequency is equal to that of the CPU’s main clock.

(3) Conversion time

Number of conversion for A-D convertor varies depending on resolution as given. Table 2.7.1 shows relation between the A-D converter operation clock and conversion time.

Sample & Hold function selected:

33 cycles for 10-bit resolution, or 28 cycles for 8-bit resolution

No Sample & Hold function:

59 cycles for 10-bit resolution, or 49 cycles for 8-bit resolution

Table 2.7.1. Conversion time every operation clock

Frequency selection bit 1		0		1
Frequency selection bit 0		0	1	Invalid
A-D converter's operation clock		$f_{AD} = \frac{f_{AD}}{4}$	$f_{AD} = \frac{f_{AD}}{2}$	$f_{AD} = f_{AD}$
Min. conversion cycles (Note 1)	8-bit mode	28 X f_{AD}		
	10-bit mode	33 X f_{AD}		
Min. conversion time (Note 2)	8-bit mode	11.2µs	5.6µs	2.8µs
	10-bit mode	13.2µs	6.6µs	3.3µs

Note 1: The number of conversion cycles per one analog input pin.

Note 2: The conversion time per one analog input pin (when f_{AD} = f(X_{IN}) = 10 MHz)

(4) Functions selection

(a) Sample & Hold function

Sample & Hold function samples input voltage when A-D conversion starts and carries out A-D conversion on the voltage sampled. When A-D conversion starts, input voltage is sampled for 3 cycles of the operation clock. When the Sample & Hold function is selected, set the operation clock for A-D conversion to 1 MHz or higher.

(b) 8-bit A-D to 10-bit A-D switching function

Either 8-bit resolution or 10-bit resolution can be selected. When 8-bit resolution is selected, the 8 higher-order bits of the 10-bit A-D are subjected to A-D conversion. The equations for 10-bit resolution and 8-bit resolution are given below:

10-bit resolution $(\mathrm{Vref} \times n / 2^{10}) - (\mathrm{Vref} \times 0.5 / 10^{10})$ (n = 1 to 1023), 0 (n = 0)

8-bit resolution $(\mathrm{Vref} \times n / 2^{8}) - (\mathrm{Vref} \times 0.5 / 2^{10})$ (n = 1 to 256), 0 (n = 0)

(c) Analog input group function

The analog input pins can be switched between the port P6 group (AN$_0$ to AN$_4$) and the port P5 group (AN$_{50}$ to AN$_{54}$).

(d) Connecting or cutting Vref

Cutting Vref allows decrease of the current flowing into the A-D converter. To decrease the microcomputer's power consumption, cut Vref. To carry out A-D conversion, start A-D conversion 1 ms or longer after connecting Vref.

The following are exsamples in which functions (a) through (d) are selected:

- One-shot mode P278
- Repeat mode P280
- Single sweep mode P282
- Repeated sweep mode 0 .... P284
- Repeated sweep mode 1 .... P286

(5) Input to A-D converter and direction register

To use the A-D converter, set the direction register of the relevant port to input.

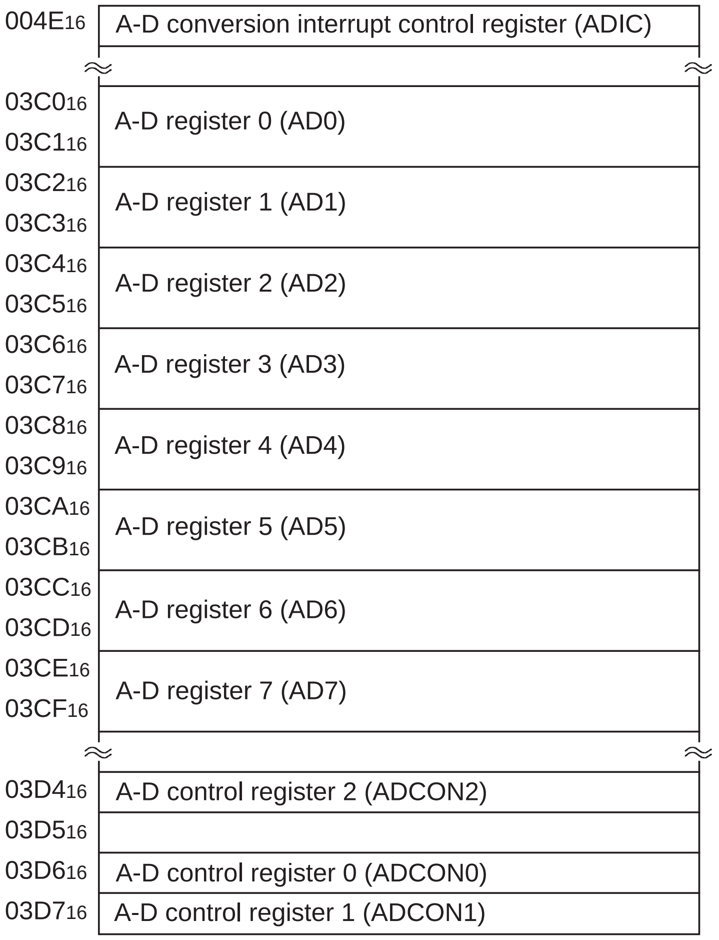
(6) Pins related to A-D converter

(a) AN$_0$ pin through AN$_7$ pin	Input pins of the A-D converter (Port P6 group)
(b) AN$_{50}$ pin through AN$_{57}$ pin	Input pins of the A-D converter (Port P5 group)
(c) AVcc pin	Power source pin of the analog section
(d) VREF pin	Input pin of reference voltage
(e) AVss pin	GND pin of the analog section

Under development

(7) A-D converter and related registers

Figure 2.7.1 shows the memory map of A-D converter-related registers, and Figures 2.7.2 through 2.7.4 show A-D converter-related registers.

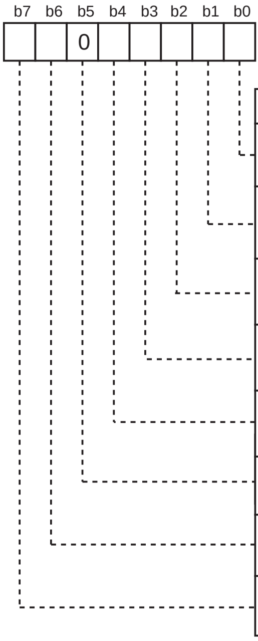


004E$_{16}$	A-D conversion interrupt control register (ADIC)
≈	≈
03C0$_{16}$	A-D register 0 (AD0)
03C1$_{16}$	
03C2$_{16}$	A-D register 1 (AD1)
03C3$_{16}$	
03C4$_{16}$	A-D register 2 (AD2)
03C5$_{16}$	
03C6$_{16}$	A-D register 3 (AD3)
03C7$_{16}$	
03C8$_{16}$	A-D register 4 (AD4)
03C9$_{16}$	
03CA$_{16}$	A-D register 5 (AD5)
03CB$_{16}$	
03CC$_{16}$	A-D register 6 (AD6)
03CD$_{16}$	
03CE$_{16}$	A-D register 7 (AD7)
03CF$_{16}$	
≈	≈
03D4$_{16}$	A-D control register 2 (ADCON2)
03D5$_{16}$	
03D6$_{16}$	A-D control register 0 (ADCON0)
03D7$_{16}$	A-D control register 1 (ADCON1)

Figure 2.7.1. Memory map of A-D converter-related registers

Under development

A-D control register 0 (Note 1)



Symbol	Address	When reset
ADCON0	03D6$_{16}$	00000XXX$_2$

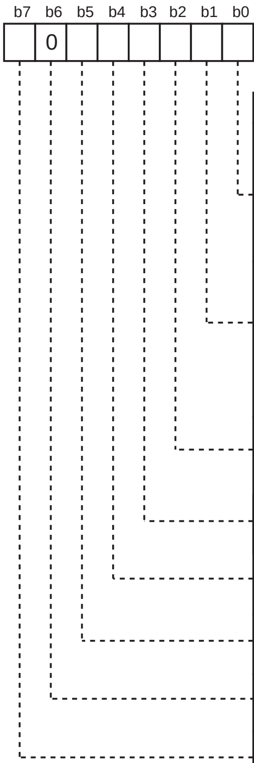
Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 Repeat sweep mode 1	○	○
MD1			○	○
Set this bit to “0”.			○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN$_{50}$ to AN$_{54}$ can be used in the same way as for AN$_0$ to AN$_4$.

Figure 2.7.2. A-D converter-related registers (1)

Under development

A-D control register 1 (Note 1)



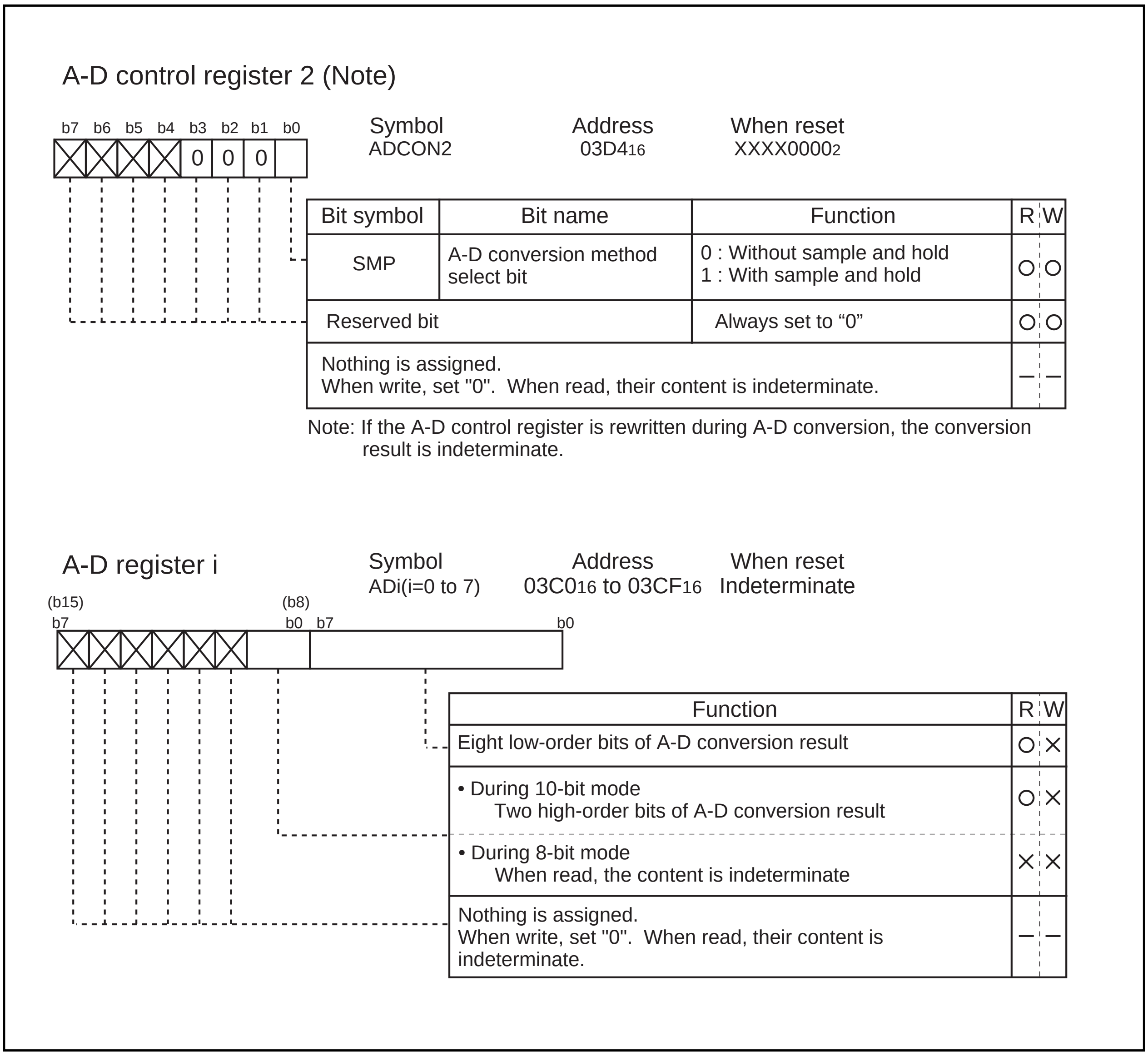
Symbol ADCON1
Address $03D7_{16}$
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins)	○	○
SCAN1		When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins) (Note 2, 3)	○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
Set this bit to “0”.			○	○
ADGSEL0	A-D input group select bit	0 : Port P6 group is selected 1 : Port P5 group is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: AN_{50} to AN_{54} can be used in the same way as for AN_0 to AN_4.
Note 3: If the repeat sweep mode is selected for the port P5 group, the contents of A-D registers 5 to 7 are indeterminate.

Figure 2.7.3. A-D converter-related registers (2)

Under development



A-D control register 2 (Note)

Symbol	Address	When reset
ADCON2	03D4$_{16}$	XXXX0000$_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their content is indeterminate.			–	–

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

Symbol	Address	When reset
ADi(i=0 to 7)	03C0$_{16}$ to 03CF$_{16}$	Indeterminate

Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. When write, set "0". When read, their content is indeterminate.	–	–

Figure 2.7.4. A-D converter-related registers (3)

2.7.2 Operation of A-D converter (one-shot mode)

In one-shot mode, choose functions from those listed in Table 2.7.2. Operations of the circled items are described below. Figure 2.7.5 shows the operation timing, and Figure 2.7.6 shows the set-up procedure.

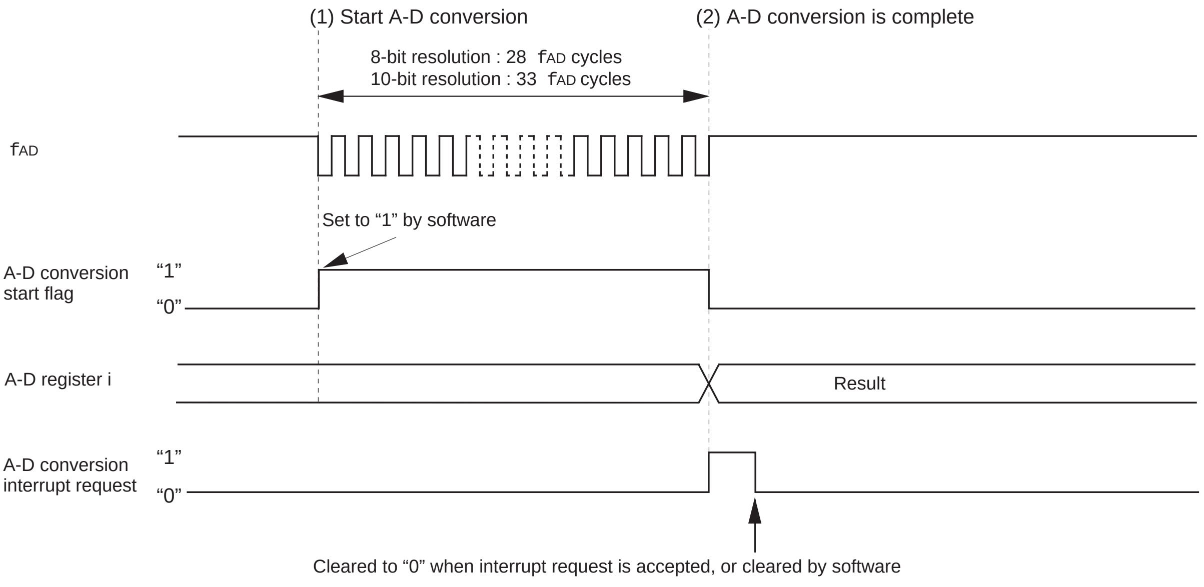
Table 2.7.2. Choosed functions

Item	Set-up	
Operation clock f_{AD}	O	Divided-by-4 f_{AD} / divided-by-2 f_{AD} / f_{AD}
Resolution	O	8-bit / 10-bit
Analog input pin	O	One of AN_0 pin to AN_7 pin (Note)
Sample & Hold		Not activated
	O	Activated

Note : When the port P5 group is selected, analog input pins are changed from AN_0 to AN_4 to pins AN_{50} to AN_{54}.

Operation (1) Setting the A-D conversion start flag to “1” causes the A-D converter to begin operating.

(2) After A-D conversion is completed, the content of the successive comparison register (conversion result) is transmitted to A-D register i. At this time, the A-D conversion interrupt request bit goes to “1”. Also, the A-D conversion start flag goes to “0”, and the A-D converter stops operating.


(1) Start A-D conversion
(2) A-D conversion is complete
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
fAD
Set to “1” by software
A-D conversion start flag
“1”
“0”
A-D register i
Result
A-D conversion interrupt request
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software

Note: When f_{AD} frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 f_{AD} cycles for 8-bit resolution and 59 f_{AD} cycles for 10-bit resolution.

Figure 2.7.5. Operation timing of one-shot mode

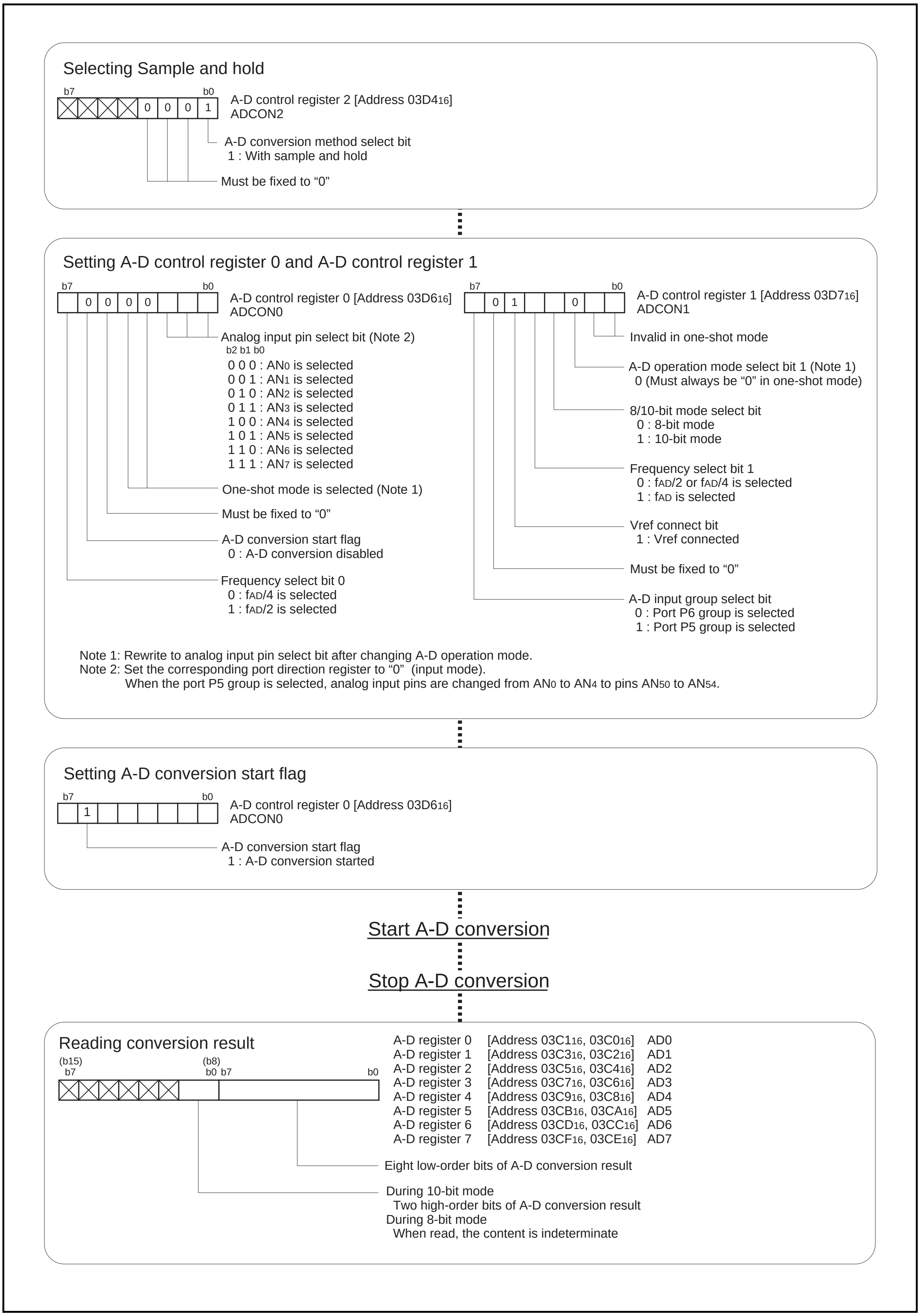

Selecting Sample and hold
b7
b0
0 0 0 1
A-D control register 2 [Address 03D4₁₆]
ADCON2
A-D conversion method select bit
1 : With sample and hold
Must be fixed to “0”
Setting A-D control register 0 and A-D control register 1
b7
b0
0 0 0 0
A-D control register 0 [Address 03D6₁₆]
ADCON0
Analog input pin select bit (Note 2)
b2 b1 b0
0 0 0 : AN0 is selected
0 0 1 : AN1 is selected
0 1 0 : AN2 is selected
0 1 1 : AN3 is selected
1 0 0 : AN4 is selected
1 0 1 : AN5 is selected
1 1 0 : AN6 is selected
1 1 1 : AN7 is selected
One-shot mode is selected (Note 1)
Must be fixed to “0”
A-D conversion start flag
0 : A-D conversion disabled
Frequency select bit 0
0 : fAD/4 is selected
1 : fAD/2 is selected
b7
b0
0 1 0
A-D control register 1 [Address 03D7₁₆]
ADCON1
Invalid in one-shot mode
A-D operation mode select bit 1 (Note 1)
0 (Must always be “0” in one-shot mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1
0 : fAD/2 or fAD/4 is selected
1 : fAD is selected
Vref connect bit
1 : Vref connected
Must be fixed to “0”
A-D input group select bit
0 : Port P6 group is selected
1 : Port P5 group is selected
Note 1: Rewrite to analog input pin select bit after changing A-D operation mode.
Note 2: Set the corresponding port direction register to “0” (input mode).
When the port P5 group is selected, analog input pins are changed from AN0 to AN4 to pins AN50 to AN54.
Setting A-D conversion start flag
b7
b0
1
A-D control register 0 [Address 03D6₁₆]
ADCON0
A-D conversion start flag
1 : A-D conversion started
Start A-D conversion
Stop A-D conversion
Reading conversion result
(b15)
b7
(b8)
b0 b7
b0
A-D register 0 [Address 03C1₁₆, 03C0₁₆] AD0
A-D register 1 [Address 03C3₁₆, 03C2₁₆] AD1
A-D register 2 [Address 03C5₁₆, 03C4₁₆] AD2
A-D register 3 [Address 03C7₁₆, 03C6₁₆] AD3
A-D register 4 [Address 03C9₁₆, 03C8₁₆] AD4
A-D register 5 [Address 03CB₁₆, 03CA₁₆] AD5
A-D register 6 [Address 03CD₁₆, 03CC₁₆] AD6
A-D register 7 [Address 03CF₁₆, 03CE₁₆] AD7
Eight low-order bits of A-D conversion result
During 10-bit mode
Two high-order bits of A-D conversion result
During 8-bit mode
When read, the content is indeterminate

Figure 2.7.6. Set-up procedure of one-shot mode

Under development

2.7.3 Operation of A-D Converter (in repeat mode)

In repeat mode, choose functions from those listed in Table 2.7.3. Operations of the circled items are described below. Figure 2.7.7 shows timing chart, and Figure 2.7.8 shows the set-up procedure.

Table 2.7.3. Choosed functions

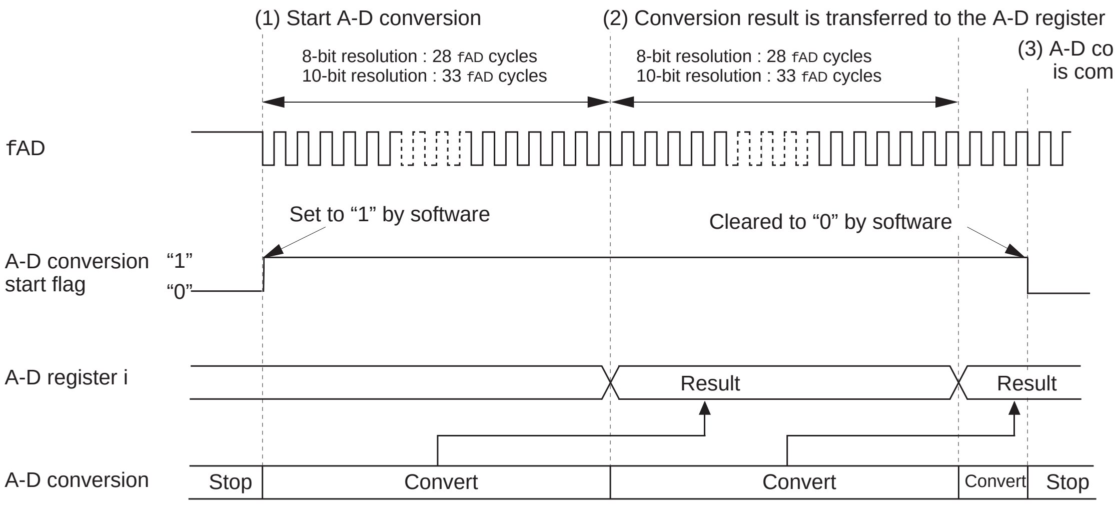
Item	Set-up	
Operation clock fAD	O	Divided-by-4 fAD / divided-by-2 fAD / fAD
Resolution	O	8-bit / 10-bit
Analog input pin	O	One of AN0 pin to AN7 pin (Note)
Sample & Hold		Not activated
	O	Activated

Note : When the port P5 group is selected, analog input pins are changed from AN0 to AN4 to pins AN50 to AN54.

Operation (1) Setting the A-D conversion start flag to “1” causes the A-D converter to start operating.

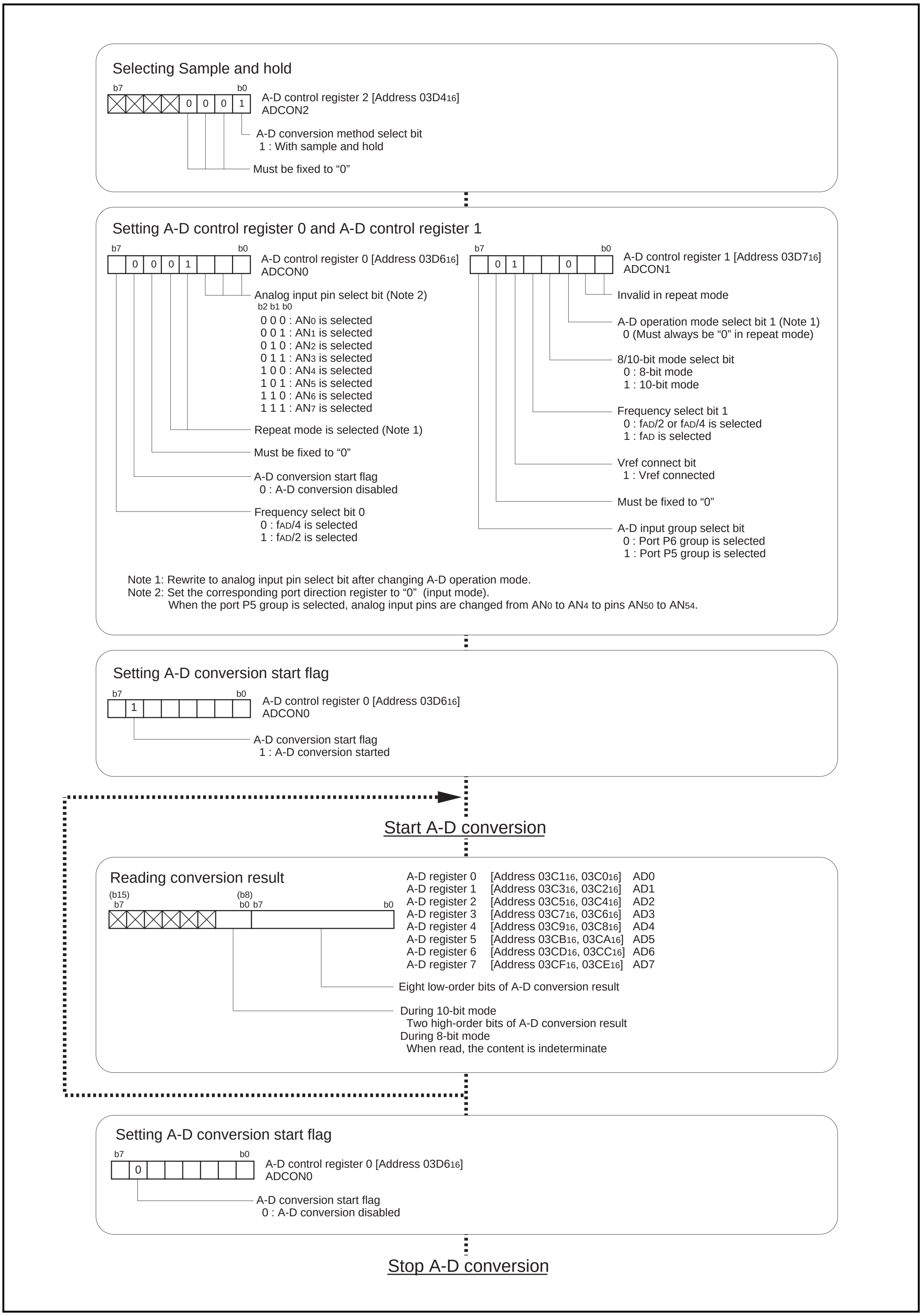
(2) After the first conversion is completed, the content of the successive comparison register (conversion result) is transmitted to A-D register i. The A-D conversion interrupt request bit does not go to “1”.

(3) The A-D converter continues operating until the A-D conversion start flag is set to “0” by software. The conversion result is transmitted to A-D register i every time a conversion is completed.


(1) Start A-D conversion
(2) Conversion result is transferred to the A-D register
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
(3) A-D conversion is complete
fAD
Set to “1” by software
Cleared to “0” by software
A-D conversion start flag
“1”
“0”
A-D register i
Result
Result
A-D conversion
Stop
Convert
Convert
Convert
Stop

Note: When fAD frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 fAD cycles for 8-bit resolution and 59 fAD cycles for 10-bit resolution.

Figure 2.7.7. Operation timing of repeat mode


Selecting Sample and hold
b7
b0
0 0 0 1
A-D control register 2 [Address 03D4₁₆]
ADCON2
A-D conversion method select bit
1 : With sample and hold
Must be fixed to "0"
Setting A-D control register 0 and A-D control register 1
b7
b0
0 0 0 1
A-D control register 0 [Address 03D6₁₆]
ADCON0
Analog input pin select bit (Note 2)
b2 b1 b0
0 0 0 : AN0 is selected
0 0 1 : AN1 is selected
0 1 0 : AN2 is selected
0 1 1 : AN3 is selected
1 0 0 : AN4 is selected
1 0 1 : AN5 is selected
1 1 0 : AN6 is selected
1 1 1 : AN7 is selected
Repeat mode is selected (Note 1)
Must be fixed to "0"
A-D conversion start flag
0 : A-D conversion disabled
Frequency select bit 0
0 : fAD/4 is selected
1 : fAD/2 is selected
b7
b0
0 1 0
A-D control register 1 [Address 03D7₁₆]
ADCON1
Invalid in repeat mode
A-D operation mode select bit 1 (Note 1)
0 (Must always be "0" in repeat mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1
0 : fAD/2 or fAD/4 is selected
1 : fAD is selected
Vref connect bit
1 : Vref connected
Must be fixed to "0"
A-D input group select bit
0 : Port P6 group is selected
1 : Port P5 group is selected
Setting A-D conversion start flag
b7
b0
1
A-D control register 0 [Address 03D6₁₆]
ADCON0
A-D conversion start flag
1 : A-D conversion started
Start A-D conversion
Reading conversion result
(b15)
b7
(b8)
b0 b7
b0
A-D register 0 [Address 03C1₁₆, 03C0₁₆] AD0
A-D register 1 [Address 03C3₁₆, 03C2₁₆] AD1
A-D register 2 [Address 03C5₁₆, 03C4₁₆] AD2
A-D register 3 [Address 03C7₁₆, 03C6₁₆] AD3
A-D register 4 [Address 03C9₁₆, 03C8₁₆] AD4
A-D register 5 [Address 03CB₁₆, 03CA₁₆] AD5
A-D register 6 [Address 03CD₁₆, 03CC₁₆] AD6
A-D register 7 [Address 03CF₁₆, 03CE₁₆] AD7
Eight low-order bits of A-D conversion result
During 10-bit mode
Two high-order bits of A-D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A-D conversion start flag
b7
b0
0
A-D control register 0 [Address 03D6₁₆]
ADCON0
A-D conversion start flag
0 : A-D conversion disabled
Stop A-D conversion

Note 1: Rewrite to analog input pin select bit after changing A-D operation mode.
Note 2: Set the corresponding port direction register to "0" (input mode).
When the port P5 group is selected, analog input pins are changed from AN_0 to AN_4 to pins AN_{50} to AN_{54}.

Figure 2.7.8. Set-up procedure of repeat mode

Under development

2.7.4 Operation of A-D Converter (in single sweep mode)

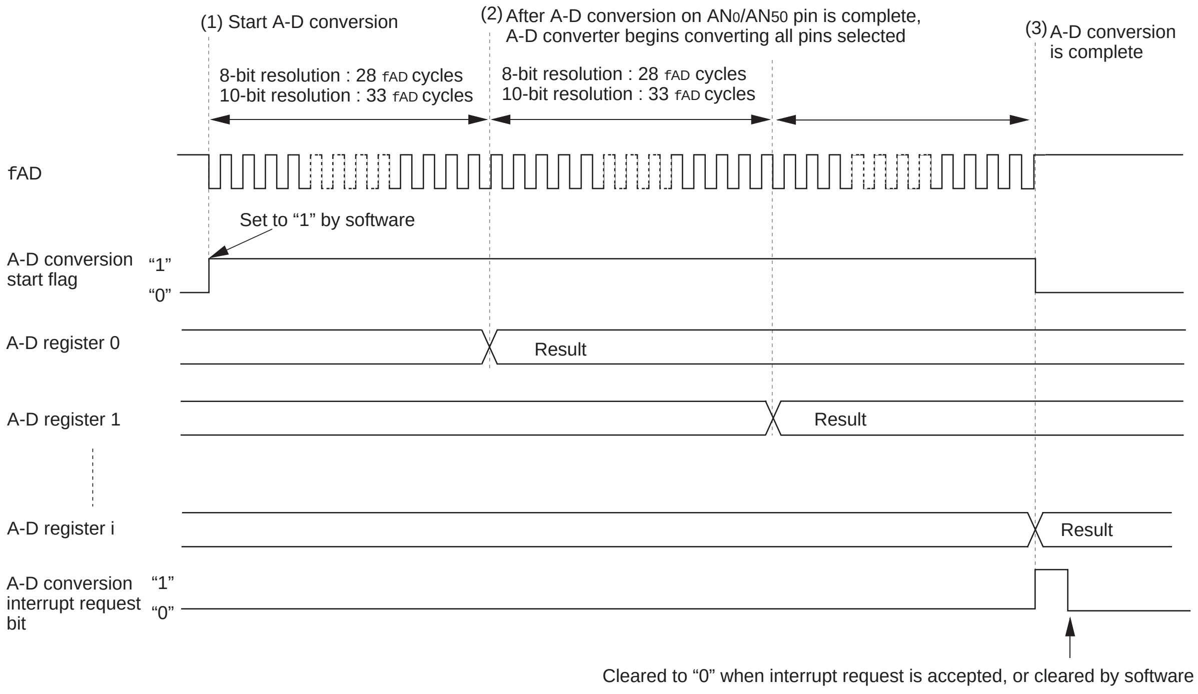
In single sweep mode, choose functions from those listed in Table 2.7.4. Operations of the circled items are described below. Figure 2.7.9 shows timing chart, and Figure 2.7.10 shows the set-up procedure.

Table 2.7.4. Choosed functions

Item	Set-up	
Operation clock f_{AD}	**O**	Divided-by-4 f_{AD} / divided-by-2 f_{AD} / f_{AD}
Resolution	**O**	8-bit / 10-bit
Analog input pin	**O**	AN$_0$ and AN$_1$ (2 pins) / AN$_0$ to AN$_3$ (4 pins) / AN$_0$ to AN$_5$ (6 pins) / AN$_0$ to AN$_7$ (8 pins) (Note)
Sample & Hold		Not activated
	O	Activated

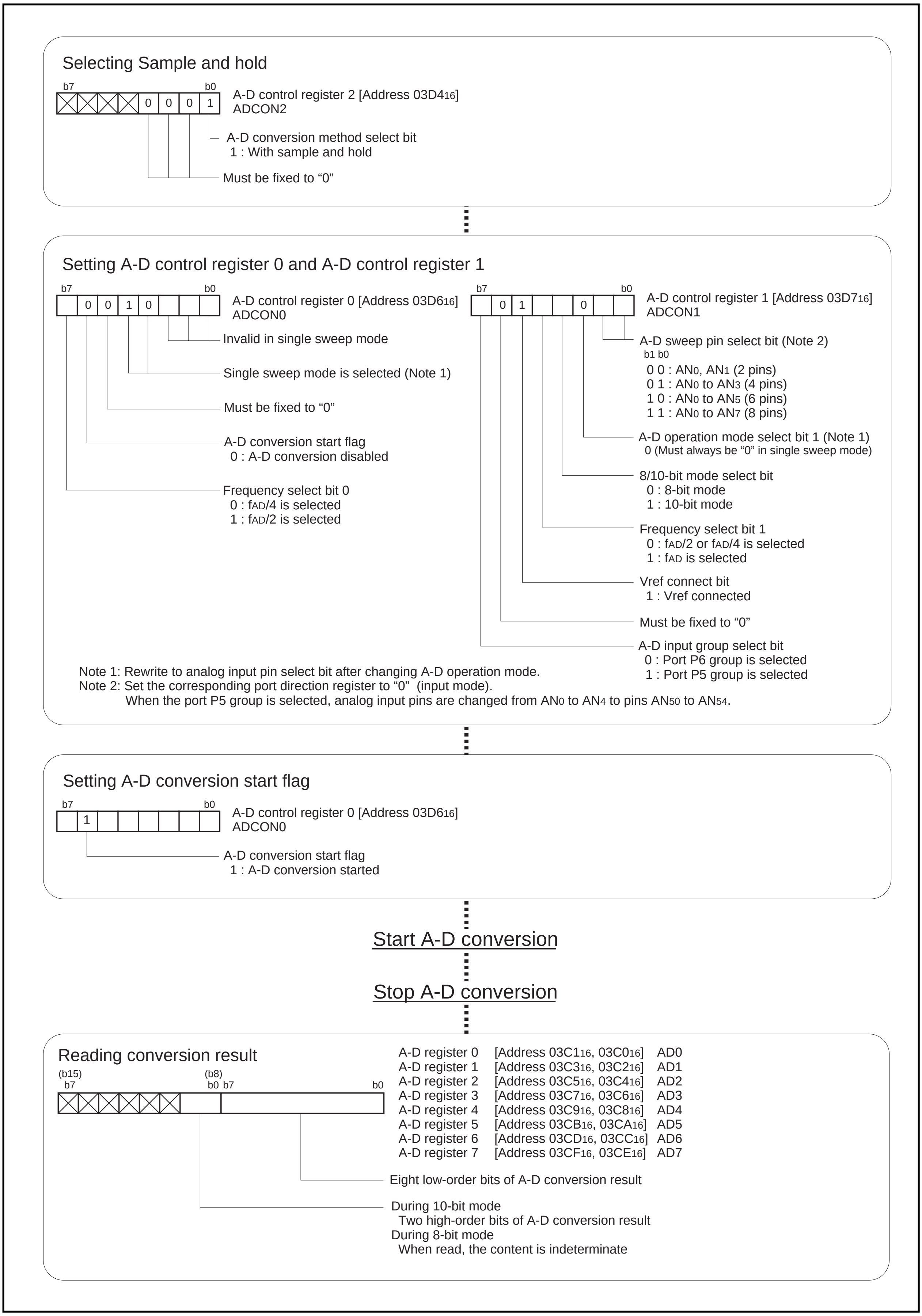
Note : When the port P5 group is selected, analog input pins are changed from AN$_0$ to AN$_4$ to pins AN$_{50}$ to AN$_{54}$.

Operation
(1) Setting the A-D conversion start flag to “1” causes the A-D converter to start the conversion on voltage input to the AN$_0$/AN$_{50}$ pin.
(2) After the A-D conversion of voltage input to the AN$_0$/AN$_{50}$ pin is completed, the content of the successive comparison register (conversion result) is transmitted to A-D register 0. The A-D converter converts all analog input pins selected by the user. The conversion result is transmitted to A-D register i corresponding to each pin, every time conversion on one pin is completed.
(3) When the A-D conversion on all the analog input pins selected is completed, the A-D conversion interrupt request bit goes to “1”. At this time, the A-D conversion start flag goes to “0”. The A-D converter stops operating.


(1) Start A-D conversion
(2) After A-D conversion on AN0/AN50 pin is complete, A-D converter begins converting all pins selected
(3) A-D conversion is complete
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
fAD
Set to “1” by software
A-D conversion start flag
“1”
“0”
A-D register 0
Result
A-D register 1
Result
A-D register i
Result
A-D conversion interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software

Note: When f_{AD} frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 f_{AD} cycles for 8-bit resolution and 59 f_{AD} cycles for 10-bit resolution.

Figure 2.7.9. Operation timing of single sweep mode


Selecting Sample and hold
b7
b0
0 0 0 1
A-D control register 2 [Address 03D4₁₆]
ADCON2
A-D conversion method select bit
1 : With sample and hold
Must be fixed to “0”
Setting A-D control register 0 and A-D control register 1
0 0 1 0
A-D control register 0 [Address 03D6₁₆]
ADCON0
Invalid in single sweep mode
Single sweep mode is selected (Note 1)
Must be fixed to “0”
A-D conversion start flag
0 : A-D conversion disabled
Frequency select bit 0
0 : fAD/4 is selected
1 : fAD/2 is selected
0 1 0
A-D control register 1 [Address 03D7₁₆]
ADCON1
A-D sweep pin select bit (Note 2)
b1 b0
0 0 : AN0, AN1 (2 pins)
0 1 : AN0 to AN3 (4 pins)
1 0 : AN0 to AN5 (6 pins)
1 1 : AN0 to AN7 (8 pins)
A-D operation mode select bit 1 (Note 1)
0 (Must always be “0” in single sweep mode)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1
0 : fAD/2 or fAD/4 is selected
1 : fAD is selected
Vref connect bit
1 : Vref connected
Must be fixed to “0”
A-D input group select bit
0 : Port P6 group is selected
1 : Port P5 group is selected
Setting A-D conversion start flag
1
A-D control register 0 [Address 03D6₁₆]
ADCON0
A-D conversion start flag
1 : A-D conversion started
Start A-D conversion
Stop A-D conversion
Reading conversion result
(b15)
b7
(b8)
b0 b7
b0
Eight low-order bits of A-D conversion result
During 10-bit mode
Two high-order bits of A-D conversion result
During 8-bit mode
When read, the content is indeterminate

Note 1: Rewrite to analog input pin select bit after changing A-D operation mode.
Note 2: Set the corresponding port direction register to “0” (input mode).
When the port P5 group is selected, analog input pins are changed from AN_0 to AN_4 to pins AN_{50} to AN_{54}.

A-D register 0	[Address 03C1₁₆, 03C0₁₆]	AD0
A-D register 1	[Address 03C3₁₆, 03C2₁₆]	AD1
A-D register 2	[Address 03C5₁₆, 03C4₁₆]	AD2
A-D register 3	[Address 03C7₁₆, 03C6₁₆]	AD3
A-D register 4	[Address 03C9₁₆, 03C8₁₆]	AD4
A-D register 5	[Address 03CB₁₆, 03CA₁₆]	AD5
A-D register 6	[Address 03CD₁₆, 03CC₁₆]	AD6
A-D register 7	[Address 03CF₁₆, 03CE₁₆]	AD7

Figure 2.7.10. Set-up procedure of single sweep mode

2.7.5 Operation of A-D Converter (in repeat sweep mode 0)

In repeat sweep 0 mode, choose functions from those listed in Table 2.7.5. Operations of the circled items are described below. Figure 2.7.11 shows timing chart, and Figure 2.7.12 shows the set-up procedure.

Table 2.7.5. Choosed functions

Item	Set-up	
Operation clock f_{AD}	**O**	Divided-by-4 f_{AD} / divided-by-2 f_{AD} / f_{AD}
Resolution	**O**	8-bit / 10-bit
Analog input pin	**O**	AN_0 and AN_1 (2 pins) / AN_0 to AN_3 (4 pins) / AN_0 to AN_5 (6 pins) / AN_0 to AN_7 (8 pins) (Note)
Sample & Hold		Not activated
	O	Activated

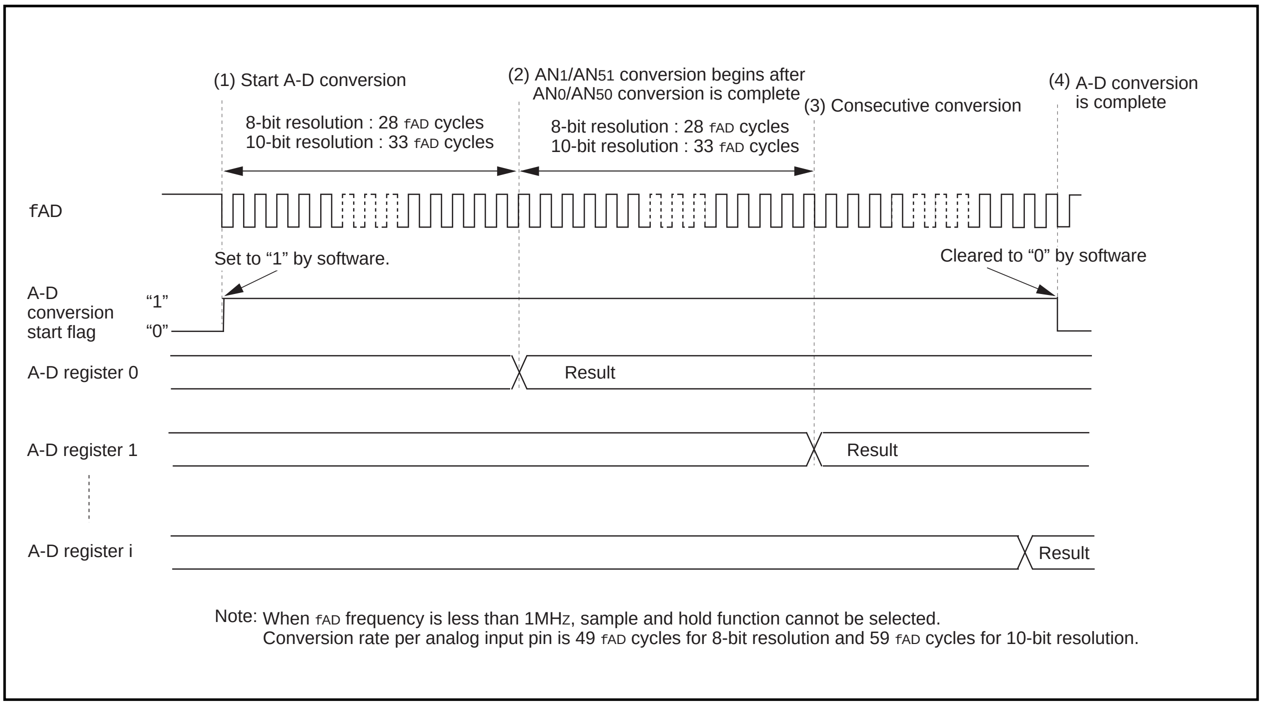
Note : When the port P5 group is selected, analog input pins are changed from AN_0 to AN_4 to pins AN_{50} to AN_{54}.

Operation (1) Setting the A-D conversion start flag to “1” causes the A-D converter to start the conversion on voltage input to the AN_0/AN_{50} pin.

(2) After the A-D conversion of voltage input to the AN_0/AN_{50} pin is completed, the content of the successive comparison register (conversion result) is transmitted to A-D register 0.

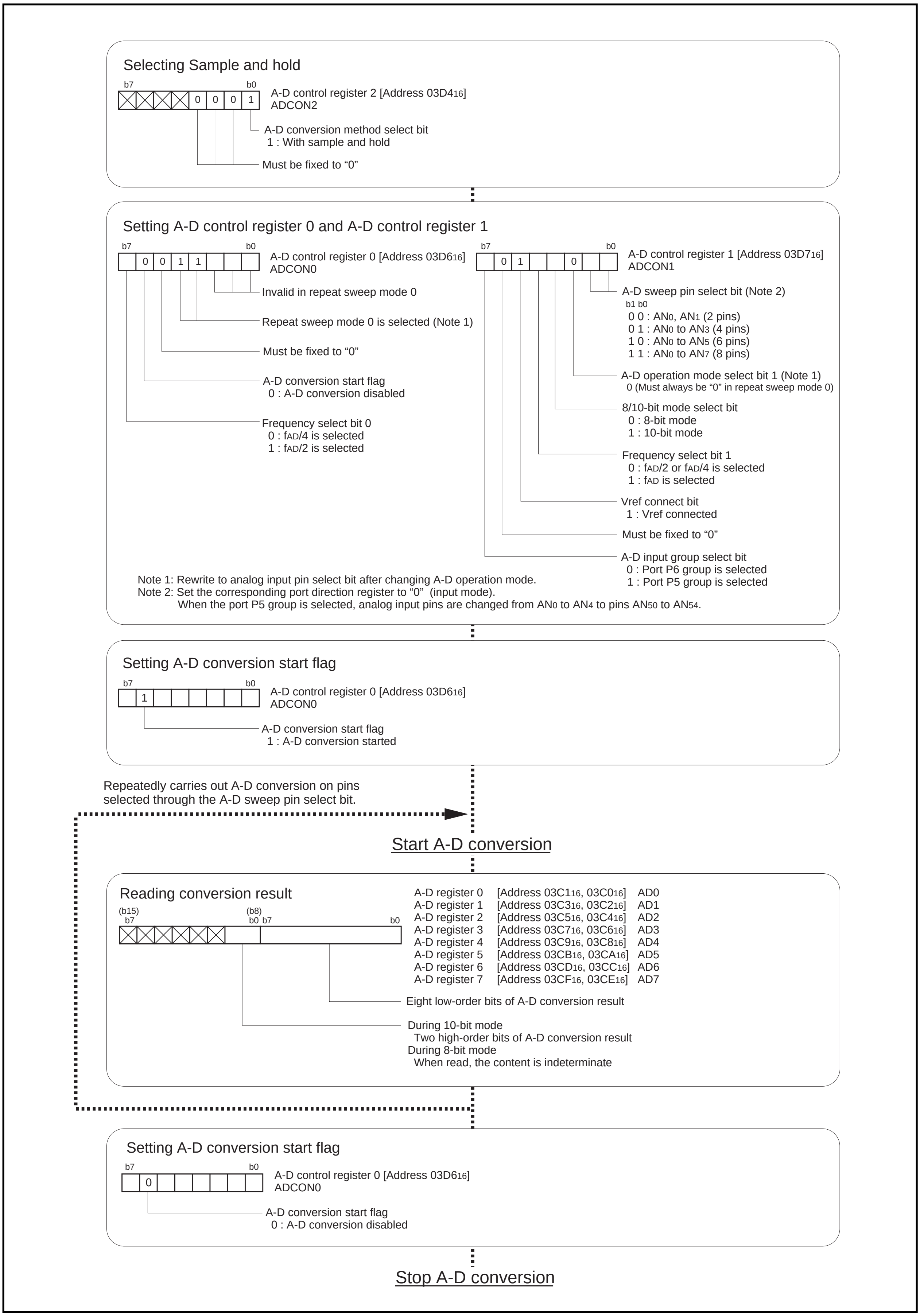
(3) The A-D converter converts all pins selected by the user. The conversion result is transmitted to A-D register i corresponding to each pin every time A-D conversion on the pin is completed. The A-D conversion interrupt request bit does not go to “1”.

(4) The A-D converter continues operating until the A-D conversion start flag is set to “0” by software.


(1) Start A-D conversion
(2) AN1/AN51 conversion begins after AN0/AN50 conversion is complete
(3) Consecutive conversion
(4) A-D conversion is complete
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
fAD
Set to “1” by software.
Cleared to “0” by software
A-D conversion start flag
“1”
“0”
A-D register 0
Result
A-D register 1
Result
A-D register i
Result

Note: When f_{AD} frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 f_{AD} cycles for 8-bit resolution and 59 f_{AD} cycles for 10-bit resolution.

Figure 2.7.11. Operation timing of repeat sweep 0 mode


Selecting Sample and hold
b7
b0
0 0 0 1
A-D control register 2 [Address $03D4_{16}$]
ADCON2
A-D conversion method select bit
1 : With sample and hold
Must be fixed to "0"
Setting A-D control register 0 and A-D control register 1
b7
b0
0 0 1 1
A-D control register 0 [Address $03D6_{16}$]
ADCON0
Invalid in repeat sweep mode 0
Repeat sweep mode 0 is selected (Note 1)
Must be fixed to "0"
A-D conversion start flag
0 : A-D conversion disabled
Frequency select bit 0
0 : f_{AD}/4 is selected
1 : f_{AD}/2 is selected
b7
b0
0 1 0
A-D control register 1 [Address $03D7_{16}$]
ADCON1
A-D sweep pin select bit (Note 2)
b1 b0
0 0 : AN_0, AN_1 (2 pins)
0 1 : AN_0 to AN_3 (4 pins)
1 0 : AN_0 to AN_5 (6 pins)
1 1 : AN_0 to AN_7 (8 pins)
A-D operation mode select bit 1 (Note 1)
0 (Must always be "0" in repeat sweep mode 0)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1
0 : f_{AD}/2 or f_{AD}/4 is selected
1 : f_{AD} is selected
Vref connect bit
1 : Vref connected
Must be fixed to "0"
A-D input group select bit
0 : Port P6 group is selected
1 : Port P5 group is selected
Setting A-D conversion start flag
b7
b0
1
A-D control register 0 [Address $03D6_{16}$]
ADCON0
A-D conversion start flag
1 : A-D conversion started
Repeatedly carries out A-D conversion on pins selected through the A-D sweep pin select bit.
Start A-D conversion
Reading conversion result
(b15)
b7
(b8)
b0 b7
b0
A-D register 0 [Address $03C1_{16}$, $03C0_{16}$] AD0
A-D register 1 [Address $03C3_{16}$, $03C2_{16}$] AD1
A-D register 2 [Address $03C5_{16}$, $03C4_{16}$] AD2
A-D register 3 [Address $03C7_{16}$, $03C6_{16}$] AD3
A-D register 4 [Address $03C9_{16}$, $03C8_{16}$] AD4
A-D register 5 [Address $03CB_{16}$, $03CA_{16}$] AD5
A-D register 6 [Address $03CD_{16}$, $03CC_{16}$] AD6
A-D register 7 [Address $03CF_{16}$, $03CE_{16}$] AD7
Eight low-order bits of A-D conversion result
During 10-bit mode
Two high-order bits of A-D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A-D conversion start flag
b7
b0
0
A-D control register 0 [Address $03D6_{16}$]
ADCON0
A-D conversion start flag
0 : A-D conversion disabled
Stop A-D conversion

Note 1: Rewrite to analog input pin select bit after changing A-D operation mode.
Note 2: Set the corresponding port direction register to "0" (input mode).
When the port P5 group is selected, analog input pins are changed from AN_0 to AN_4 to pins AN_{50} to AN_{54}.

Figure 2.7.12. Set-up procedure of repeat sweep 0 mode

Under development

2.7.6 Operation of A-D Converter (in repeat sweep mode 1)

In repeat sweep 1 mode, choose functions from those listed in Table 2.7.6. Operations of the circled items are described below. Figure 2.7.13 shows ANi pin's sweep sequence, Figure 2.7.14 shows timing chart, and Figure 2.7.15 shows the set-up procedure.

Table 2.7.6. Choosed functions

Item	Set-up	
Operation clock f$_{AD}$	O	Divided-by-4 f$_{AD}$ / divided-by-2 f$_{AD}$ / f$_{AD}$
Resolution	O	8-bit / 10-bit
Analog input pin	O	AN$_0$ (1 pins) / AN$_0$ to AN$_1$ (2 pins) / AN$_0$ to AN$_2$ (3 pins) / AN$_0$ to AN$_3$ (4 pins) (Note)
Sample & Hold		Not activated
	O	Activated

Note : When the port P5 group is selected, analog input pins are changed from AN$_0$ to AN$_4$ to pins AN$_{50}$ to AN$_{54}$.

Operation

(1) Setting the A-D conversion start flag to “1” causes the A-D converter to start the conversion on voltage input to the AN$_0$/AN$_{50}$ pin.

(2) After the A-D conversion on voltage input to the AN$_0$/AN$_{50}$ pin is completed, the content of the successive comparison register (conversion result) is transmitted to A-D register 0.

(3) Every time the A-D converter carries out A-D conversion on a selected analog input pin, the A-D converter carries out A-D conversion on only one unselected pin, and then the A-D converter carries out A-D conversion from the AN0 pin again. (See Figure 2.7.13.) The conversion result is transmitted to A-D register i every time conversion on a pin is completed. The A-D conversion interrupt request bit does not go to “1”.

(4) The A-D converter continues operating until software goes the A-D conversion start flag to “0”.

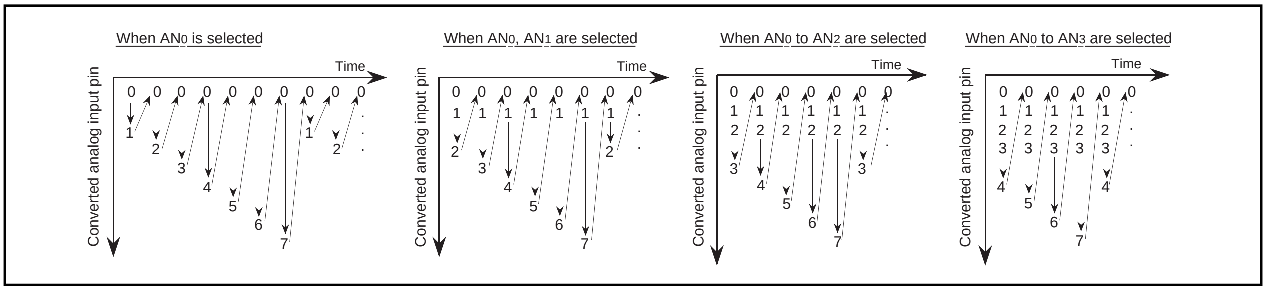
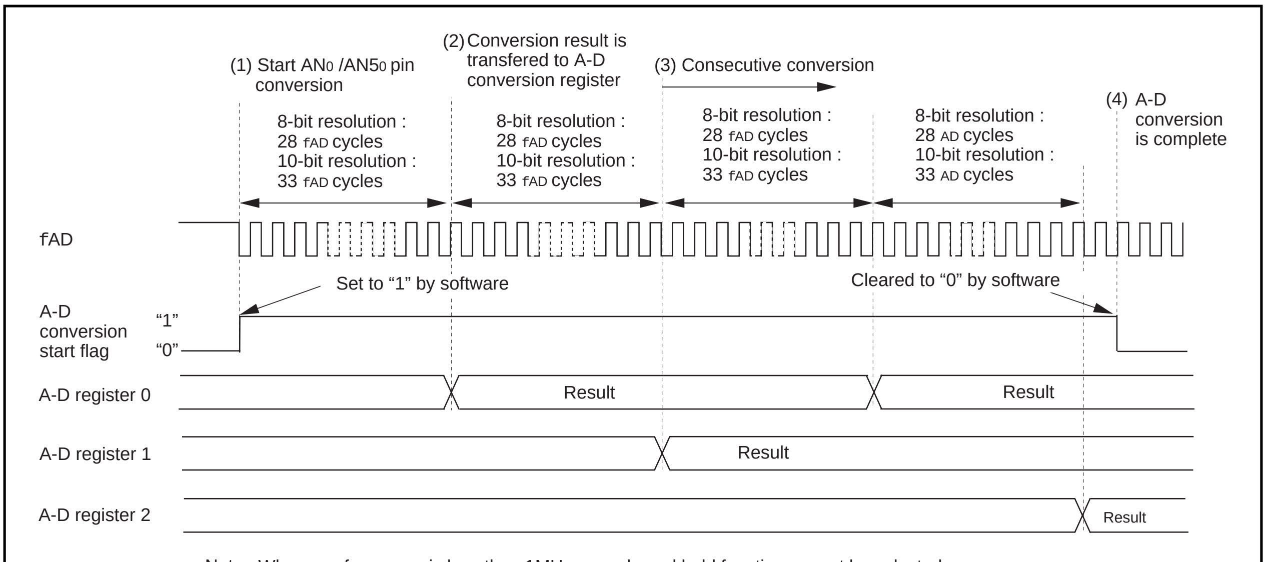

When AN0 is selected
When AN0, AN1 are selected
When AN0 to AN2 are selected
When AN0 to AN3 are selected
Time
Converted analog input pin

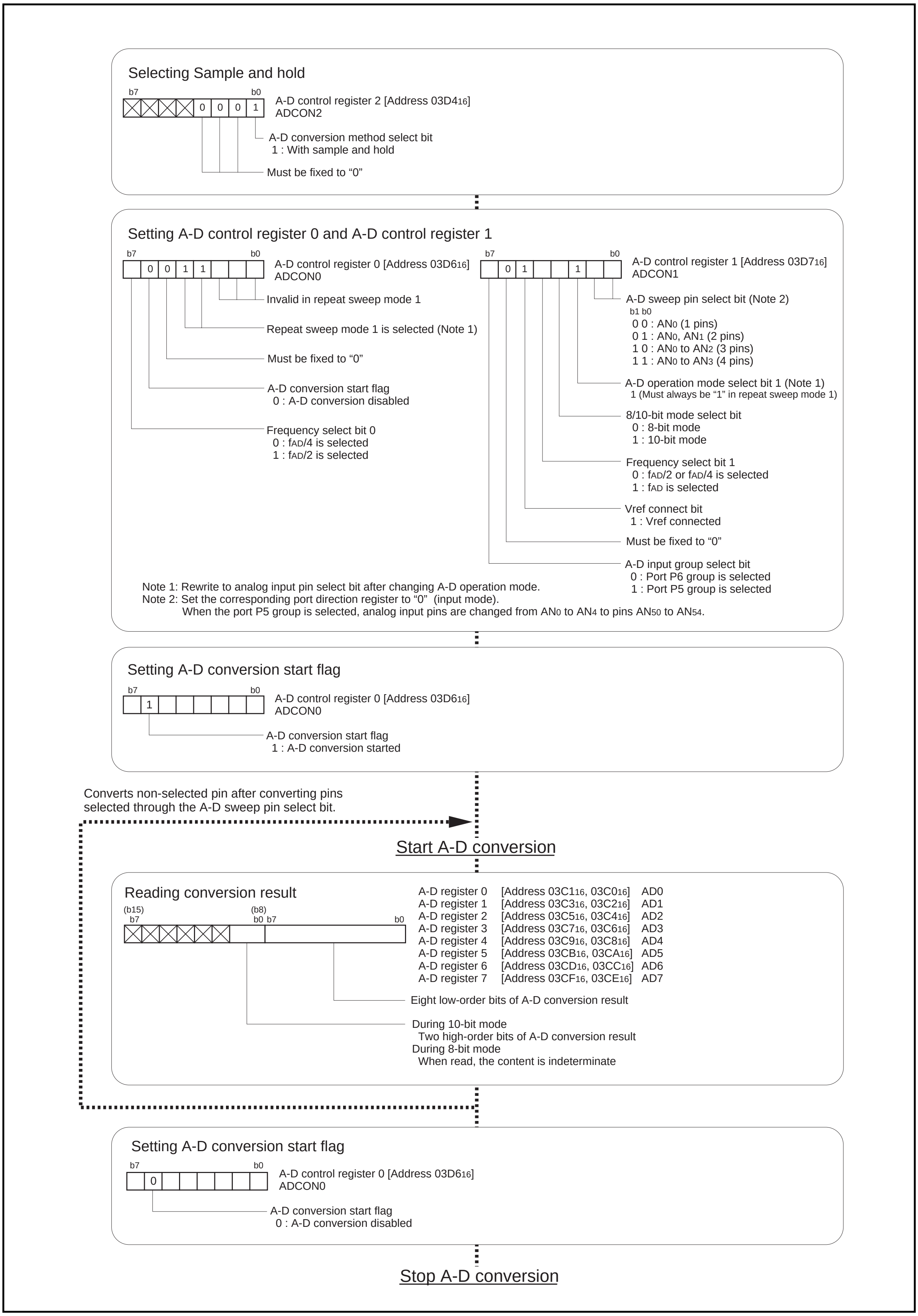
Figure 2.7.13. ANi pin's sweep sequence in repeat sweep mode


(1) Start AN0 /AN50 pin conversion
(2) Conversion result is transfered to A-D conversion register
(3) Consecutive conversion
(4) A-D conversion is complete
8-bit resolution : 28 fAD cycles
10-bit resolution : 33 fAD cycles
fAD
Set to “1” by software
Cleared to “0” by software
A-D conversion start flag
“1”
“0”
A-D register 0
A-D register 1
A-D register 2
Result

Note: When f$_{AD}$ frequency is less than 1MHz, sample and hold function cannot be selected.
Conversion rate per analog input pin is 49 f$_{AD}$ cycles for 8-bit resolution and 59 f$_{AD}$ cycles for 10-bit resolution.

Figure 2.7.14. Operation timing of repeat sweep 1 mode

Under development


Selecting Sample and hold
b7
b0
0 0 0 1
A-D control register 2 [Address $03D4_{16}$]
ADCON2
A-D conversion method select bit
1 : With sample and hold
Must be fixed to “0”
Setting A-D control register 0 and A-D control register 1
b7
b0
0 0 1 1
A-D control register 0 [Address $03D6_{16}$]
ADCON0
Invalid in repeat sweep mode 1
Repeat sweep mode 1 is selected (Note 1)
Must be fixed to “0”
A-D conversion start flag
0 : A-D conversion disabled
Frequency select bit 0
0 : $f_{AD}/4$ is selected
1 : $f_{AD}/2$ is selected
b7
b0
0 1 1
A-D control register 1 [Address $03D7_{16}$]
ADCON1
A-D sweep pin select bit (Note 2)
b1 b0
0 0 : AN_0 (1 pins)
0 1 : AN_0, AN_1 (2 pins)
1 0 : AN_0 to AN_2 (3 pins)
1 1 : AN_0 to AN_3 (4 pins)
A-D operation mode select bit 1 (Note 1)
1 (Must always be “1” in repeat sweep mode 1)
8/10-bit mode select bit
0 : 8-bit mode
1 : 10-bit mode
Frequency select bit 1
0 : $f_{AD}/2$ or $f_{AD}/4$ is selected
1 : f_{AD} is selected
Vref connect bit
1 : Vref connected
Must be fixed to “0”
A-D input group select bit
0 : Port P6 group is selected
1 : Port P5 group is selected
Setting A-D conversion start flag
b7
b0
1
A-D control register 0 [Address $03D6_{16}$]
ADCON0
A-D conversion start flag
1 : A-D conversion started
Converts non-selected pin after converting pins selected through the A-D sweep pin select bit.
Start A-D conversion
Reading conversion result
(b15)
b7
(b8)
b0 b7
b0
A-D register 0 [Address $03C1_{16}$, $03C0_{16}$] AD0
A-D register 1 [Address $03C3_{16}$, $03C2_{16}$] AD1
A-D register 2 [Address $03C5_{16}$, $03C4_{16}$] AD2
A-D register 3 [Address $03C7_{16}$, $03C6_{16}$] AD3
A-D register 4 [Address $03C9_{16}$, $03C8_{16}$] AD4
A-D register 5 [Address $03CB_{16}$, $03CA_{16}$] AD5
A-D register 6 [Address $03CD_{16}$, $03CC_{16}$] AD6
A-D register 7 [Address $03CF_{16}$, $03CE_{16}$] AD7
Eight low-order bits of A-D conversion result
During 10-bit mode
Two high-order bits of A-D conversion result
During 8-bit mode
When read, the content is indeterminate
Setting A-D conversion start flag
b7
b0
0
A-D control register 0 [Address $03D6_{16}$]
ADCON0
A-D conversion start flag
0 : A-D conversion disabled
Stop A-D conversion

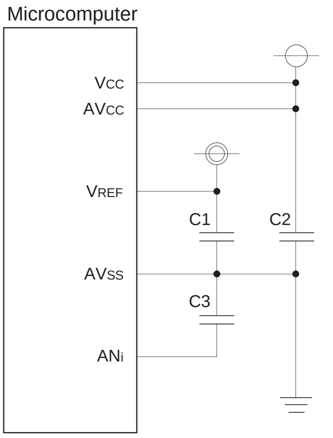
Note 1: Rewrite to analog input pin select bit after changing A-D operation mode.
Note 2: Set the corresponding port direction register to “0” (input mode).
When the port P5 group is selected, analog input pins are changed from AN_0 to AN_4 to pins AN_{50} to AN_{54}.

Figure 2.7.15. Set-up procedure of repeat sweep 1 mode

2.7. 7 Precautions for A-D Converter

(1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).
In particular, when the Vref connection bit is changed from 0 to 1, start A-D conversion after an elapse of 1 ms or longer.

(2) To reduce conversion error due to noise, connect a voltage to the AVcc pin and to the Vref pin from an independent source. It is recommended to connect a capacitor between the AVss pin and the AVcc pin, between the AVss pin and the Vref pin, and between the AVss pin and the analog input pin (ANi/AN5i). Figure 2.7.16 shows the an example of connecting the capacitors to these pins.



Note 1: C≥10.47 μF, C≥ 20.47 μF, C≥ 3100 pF
(for reference)
Note 2: Use thick and shortest possible wiring to connect capacitors.

Figure 2.7.16. Use of capacitors to reduce noice

(3) Set the direction register of the following ports to input: the port corresponding to a pin to be used as an analog input pin and external trigger input pin.

(4) If using the A-D converter with Vcc = 2.7V to 4.0 V:
Use without fAD (no frequency division) for φAD.
Select without the Sample & Hold feature.
Select 8-bit mode.

(5) Rewrite to analog input pin after changing A-D operation mode. The two cannot be set at the same time.

(6) When using the one-shot or single sweep mode
Confirm that A-D conversion is complete before reading the A-D register.
(Note: When A-D conversion interrupt request bit is set, it shows that A-D conversion is completed.)

(7) When using the repeat mode or repeat sweep mode 0 or 1
Use the undivided main clock as the internal CPU clock.

Under development

2.7.8 Method of A-D Conversion (10-bit mode)

(1) The A-D converter compares the reference voltage (Vref) generated internally based on the contents of the successive comparison register with the analog input voltage (VIN) input from the analog input pin. Each bit of the comparison result is stored in the successive comparison register until analog-to-digital conversion (successive comparison method) is complete. If a trigger occurs, the A-D converter carries out the following:

1. Fixes bit 9 of the successive comparison register.

   Compares Vref with VIN: [In this instance, the contents of the successive comparison register are “1000000000_2” (default).]

   Bit 9 of the successive comparison register varies depending on the comparison result as follows.

   If Vref < VIN, then “1” is assigned to bit 9.
   If Vref > VIN, then “0” is assigned to bit 9.

2. Fixes bit 8 of the successive comparison register.

   Sets bit 8 of the successive comparison register to “1”, then compares Vref with VIN.
   Bit 8 of the successive comparison register varies depending on the comparison result as follows:

   If Vref < VIN, then “1” is assigned to bit 8.
   If Vref > VIN, then “0” is assigned to bit 8.

3. Fixes bit 7 through bit 0 of the successive comparison register.

   Carries out step 2 above on bit 7 through bit 0.
   After bit 0 is fixed, the contents of the successive comparison register (conversion result) are transmitted to A-D register i.

Vref is generated based on the latest content of the successive comparison register. Table 2.7.7 shows the relationship of the successive comparison register contents and Vref. Table 2.7.8 shows how the successive comparison register and Vref vary while A-D conversion is in progress. Figure 2.7.17 shows theoretical A-D conversion characteristics.

Table 2.7.7. Relationship of the successive comparison register contents and Vref

Successive approximation register : n	Vref (V)
0	0
1 to1023	$\frac{V_{REF}}{1024} \times n - \frac{V_{REF}}{2048}$

Table 2.7.8. Variation of the successive comparison register and Vref while A-D conversion is in progress (10-bit mode)

	Successive approximation register	Vref change
A-D converter stopped	b9 b0 1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison ↓	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
2nd comparison ↓	n_9 1 0 0 0 0 0 0 0 0 ↑ 1st comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] $\left(\begin{array}{ll} n_9 = 1 & +\frac{V_{REF}}{4} \\ n_9 = 0 & -\frac{V_{REF}}{4} \end{array}\right)$
3rd comparison ↓	n_9 n_8 1 0 0 0 0 0 0 0 ↑ 2nd comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] $\left(\begin{array}{ll} n_8 = 1 & +\frac{V_{REF}}{8} \\ n_8 = 0 & -\frac{V_{REF}}{8} \end{array}\right)$
⋮ ↓	⋮	⋮
10th comparison ↓	n_9 n_8 n_7 n_6 n_5 n_4 n_3 n_2 n_1 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm \ldots\ldots \pm \frac{V_{REF}}{1024} - \frac{V_{REF}}{2048}$ [V]
Conversion complete	n_9 n_8 n_7 n_6 n_5 n_4 n_3 n_2 n_1 n_0 This data transfers to the bit 0 to bit 9 of A-D register.	

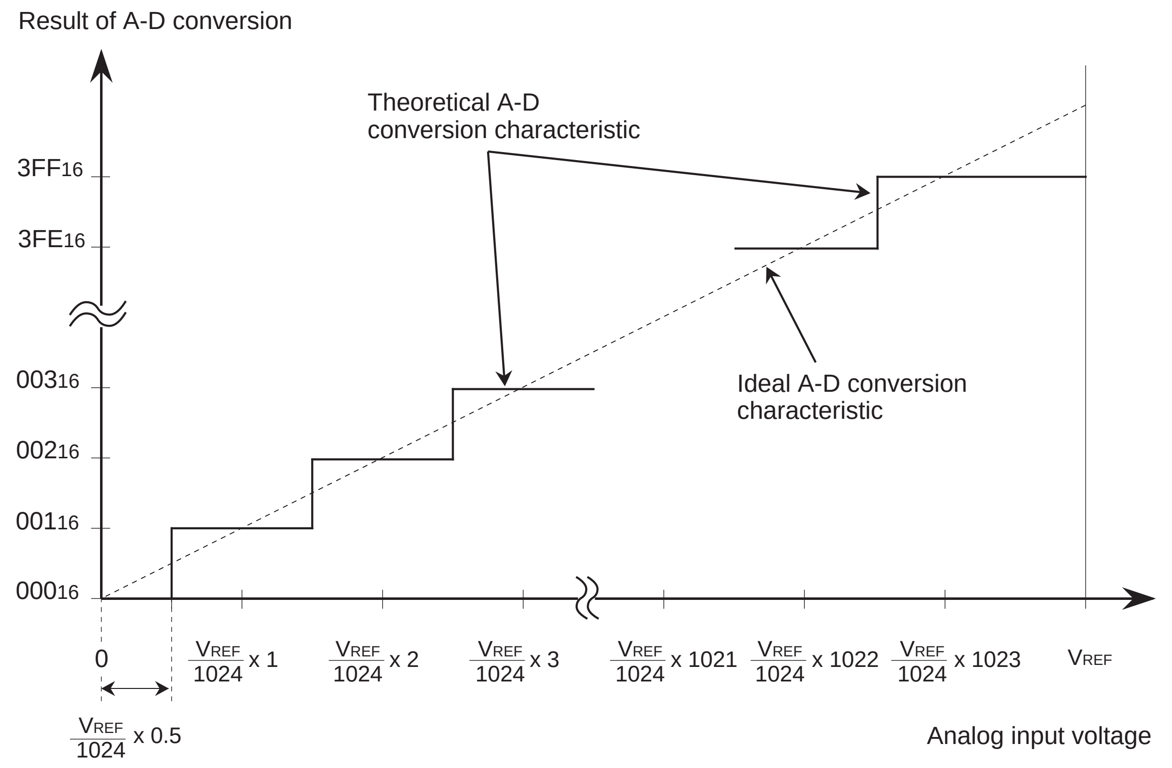


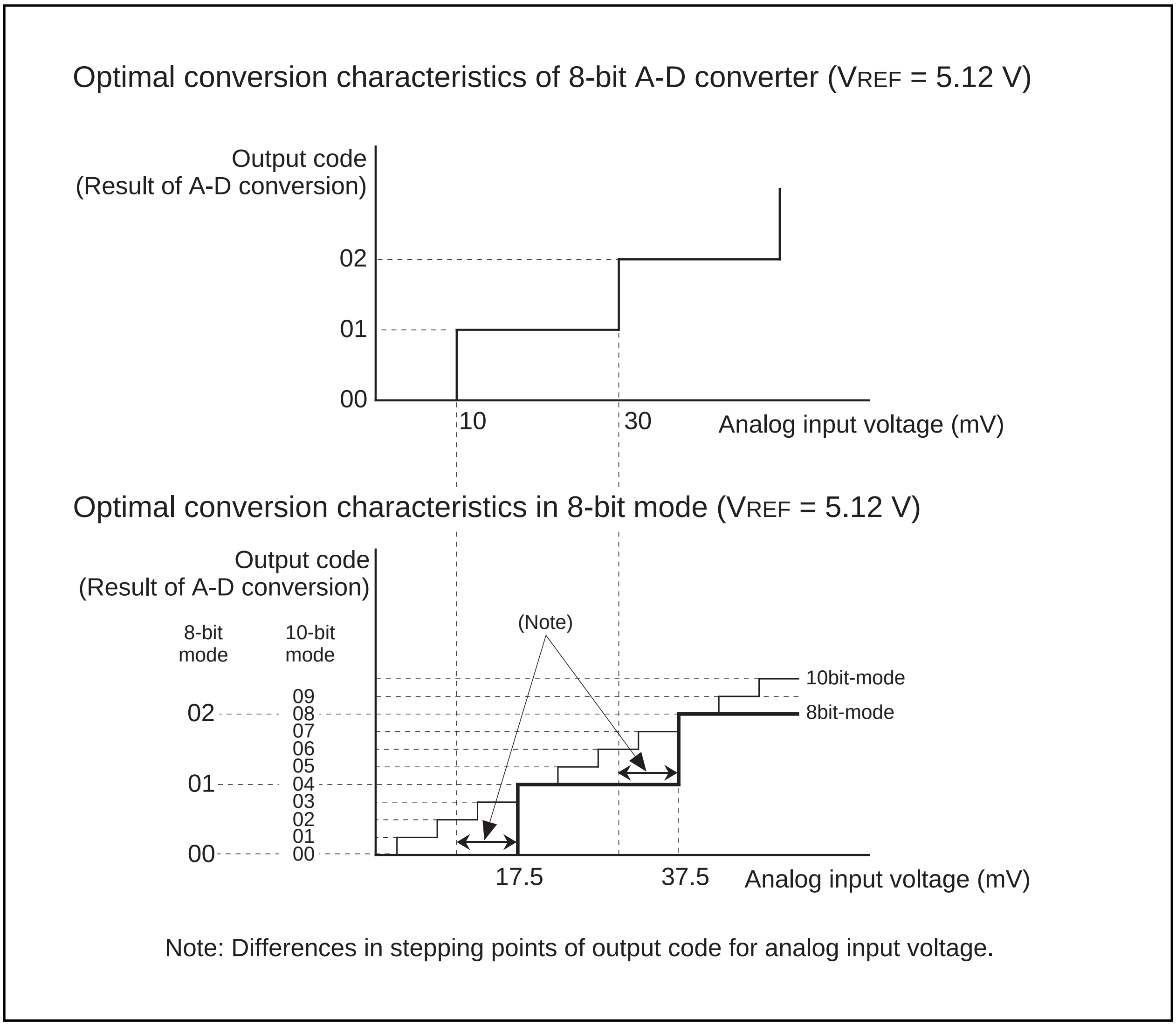
Figure 2.7.17. Theoretical A-D conversion characteristics (10-bit mode)

2.7.9 Method of A-D Conversion (8-bit mode)

(1) In 8-bit mode, 8 higher-order bits of the 10-bit successive comparison register becomes A-D conversion result. Hence, if compared to a result obtained by using an 8-bit A-D converter, the voltage compared is different by 3 VREF/2048 (see what are underscored in Table 2.7.9), and differences in stepping points of output codes occur as shown in Figure 2.7.18.

Table 2.7.9. The comparison voltage in 8-bit mode compared to 8-bit A-D converter

		8-bit mode	8-bit A-D converter
Comparison voltage Vref	n = 0	0	0
	n = 1 to 255	$\frac{V_{REF}}{2^8} \times n - \underline{\frac{V_{REF}}{2^{10}} \times 0.5}$	$\frac{V_{REF}}{2^8} \times n - \underline{\frac{V_{REF}}{2^8} \times 0.5}$


Optimal conversion characteristics of 8-bit A-D converter (VREF = 5.12 V)

Optimal conversion characteristics in 8-bit mode (VREF = 5.12 V)

Note: Differences in stepping points of output code for analog input voltage.

Figure 2.7.18. The level conversion characteristics of 8-bit mode and 8-bit A-D converter

Table 2.7.10. Variation of the successive comparison register and Vref while A-D conversion is in progress (8-bit mode)

	Successive approximation register	Vref change
	b9 … b0	
A-D converter stopped	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2}$ [V]
1st comparison	1 0 0 0 0 0 0 0 0 0	$\frac{V_{REF}}{2} - \frac{V_{REF}}{2048}$ [V]
↓ 2nd comparison	n_9 1 0 0 0 0 0 0 0 0 ↑ 1st comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} - \frac{V_{REF}}{2048}$ [V] $\begin{pmatrix} n_9 = 1 & + \frac{V_{REF}}{4} \\ n_9 = 0 & - \frac{V_{REF}}{4} \end{pmatrix}$
↓ 3rd comparison	n_9 n_8 1 0 0 0 0 0 0 0 ↑ 2nd comparison result	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} - \frac{V_{REF}}{2048}$ [V] $\begin{pmatrix} n_8 = 1 & + \frac{V_{REF}}{8} \\ n_8 = 0 & - \frac{V_{REF}}{8} \end{pmatrix}$
↓ ⋮ ↓	⋮	⋮
8th comparison	n_9 n_8 n_7 n_6 n_5 n_4 n_3 1 0 0	$\frac{V_{REF}}{2} \pm \frac{V_{REF}}{4} \pm \frac{V_{REF}}{8} \pm ...... \pm \frac{V_{REF}}{256} - \frac{V_{REF}}{2048}$ [V]
↓ Conversion complete	$\underbrace{n_9\ n_8\ n_7\ n_6\ n_5\ n_4\ n_3\ n_2}$ 0 0 This data transfers to bit 0 to bit 7 of A-D register.	

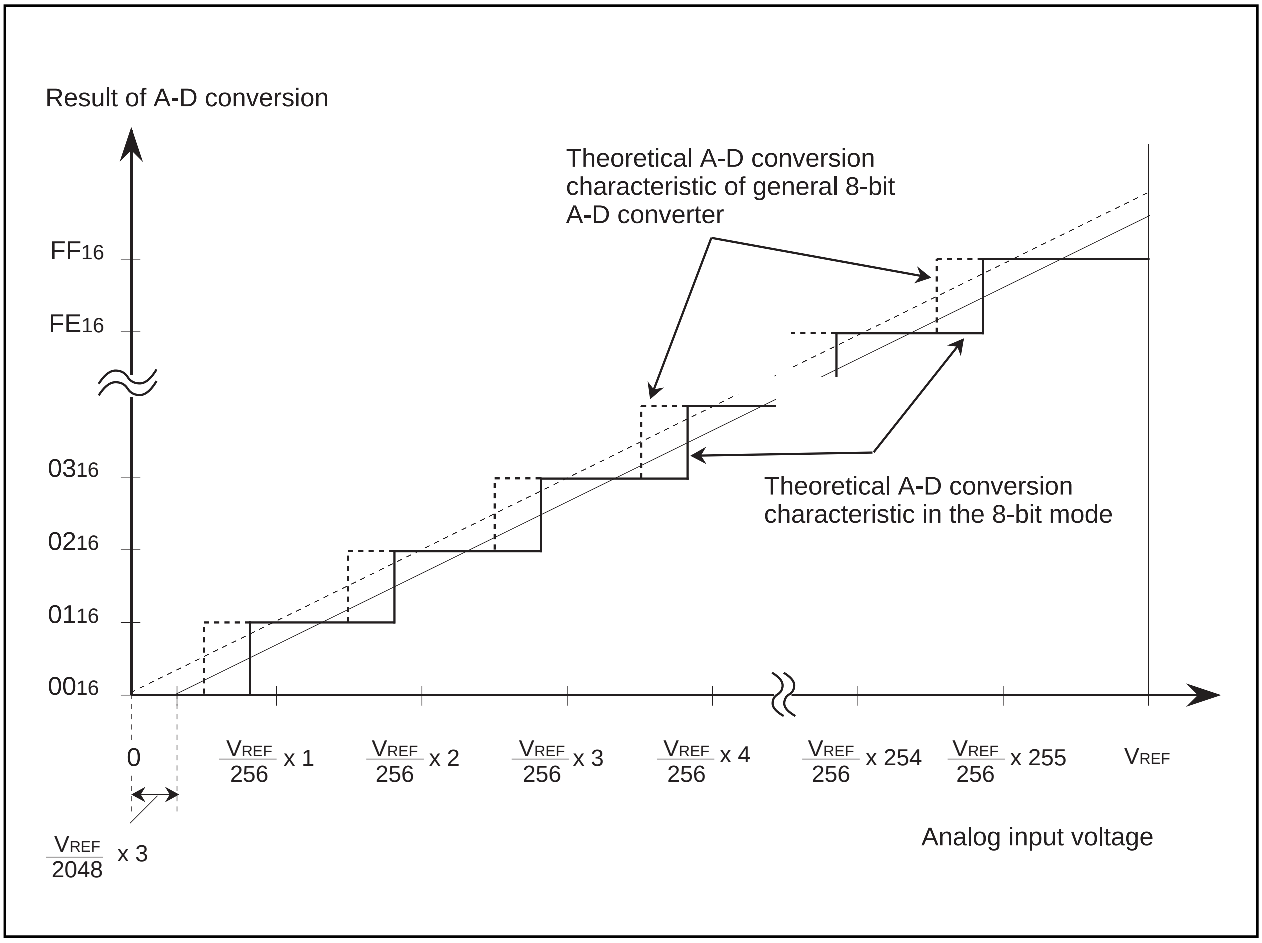


Figure 2.7.19. Theoretical A-D conversion characteristics (8-bit mode)

2.7.10 Absolute Accuracy and Differential Non-Linearity Error

- **Absolute accuracy**

Absolute accuracy is the difference between output code based on the theoretical A-D conversion characteristics, and actual A-D conversion result. When measuring absolute accuracy, the voltage at the middle point of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A-D conversion characteristics, is used as an analog input voltage. For example, if 10-bit resolution is used and if VREF (reference voltage) = 5.12 V, then 1-LSB width becomes 5 mV, and 0 mV, 5 mV, 10 mV, 15 mV, 20 mV, ···· are used as analog input voltages. If analog input voltage is 25 mV, “absolute accuracy = – 3LSB” refers to the fact that actual A-D conversion falls on a range from “002_{16}” to ”008_{16}” though an output code, “005_{16}”, can be expected from the theoretical A-D conversion characteristics. Zero error and full-scale error are included in absolute accuracy.

Also, all the output codes for analog input voltage between VREF and AVcc becomes “$3FF_{16}$”.

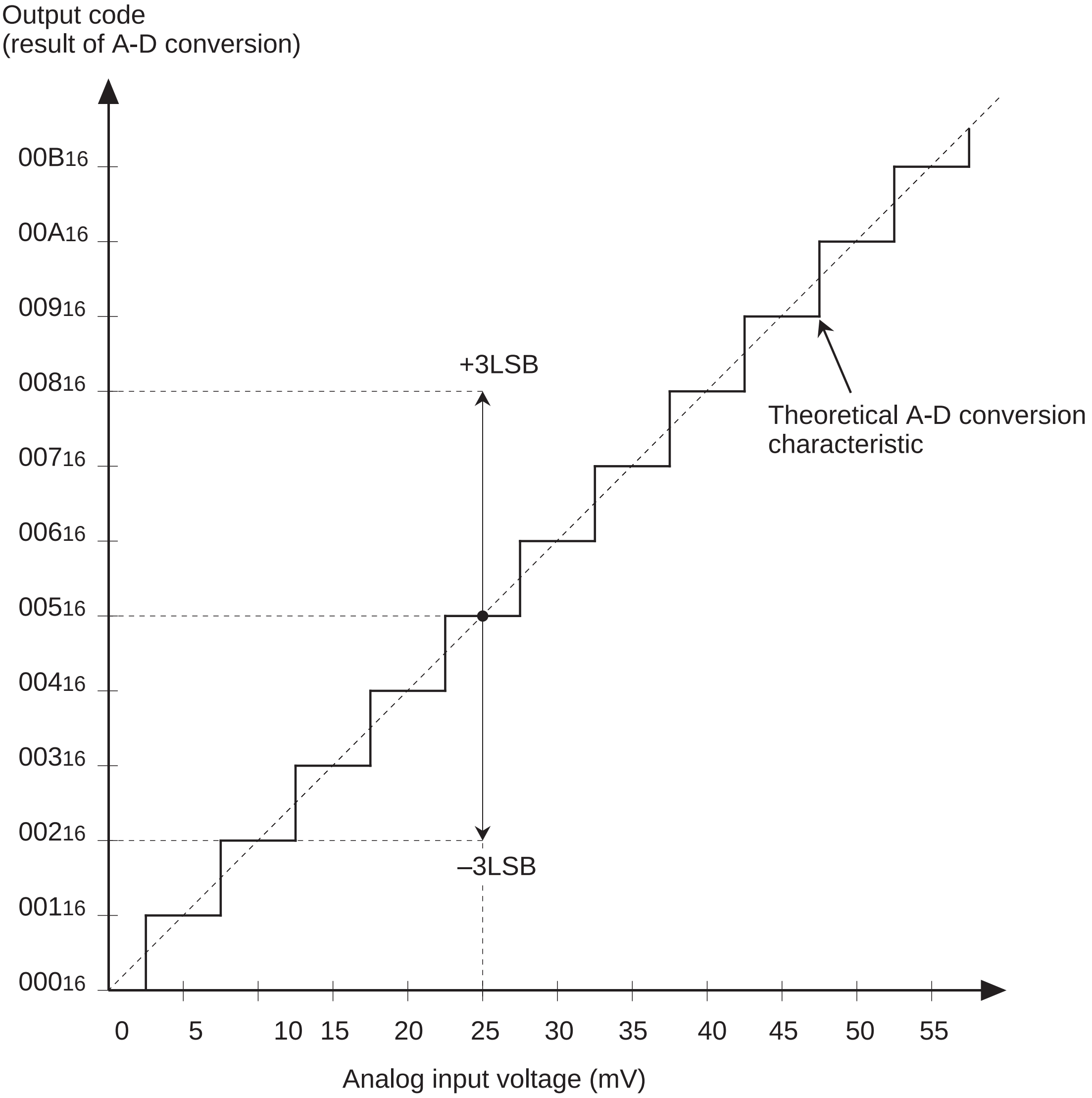


Figure 2.7.20. Absolute accuracy (10-bit resolution)

- **Differential non-linearity error**

  Differential non-linearity error refers to the difference between 1-LSB width based on the theoretical A-D conversion characteristics (an analog input width that can meet the expectation of outputting an equal code) and an actually measured 1-LSB width (analog input voltage width that outputs an equal code). If 10-bit resolution is used and if VREF (reference voltage) = 5.12 V, “differential non-linearity error = – 1LSB” refers to the fact that 1-LSB width actually measured falls on a range from 0 mV to 10 mV though 1-LSB width based on the theoretical A-D conversion characteristics is 5 mV (see 5.2 A-D converter's standard characteristics).

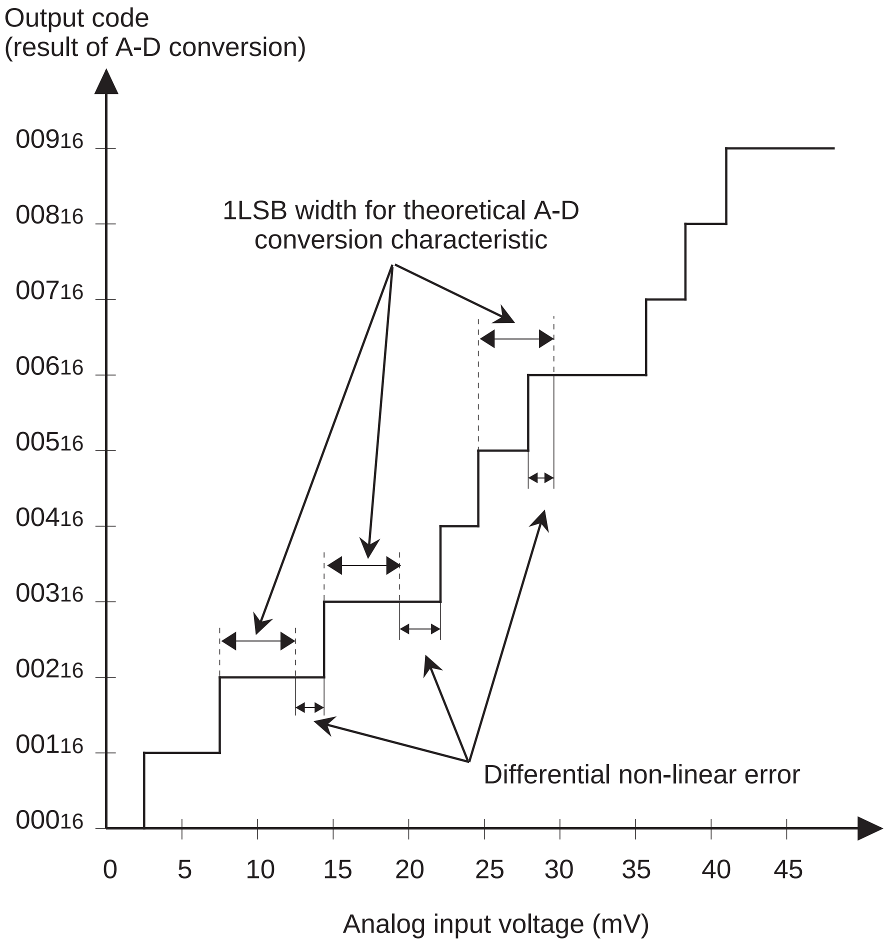
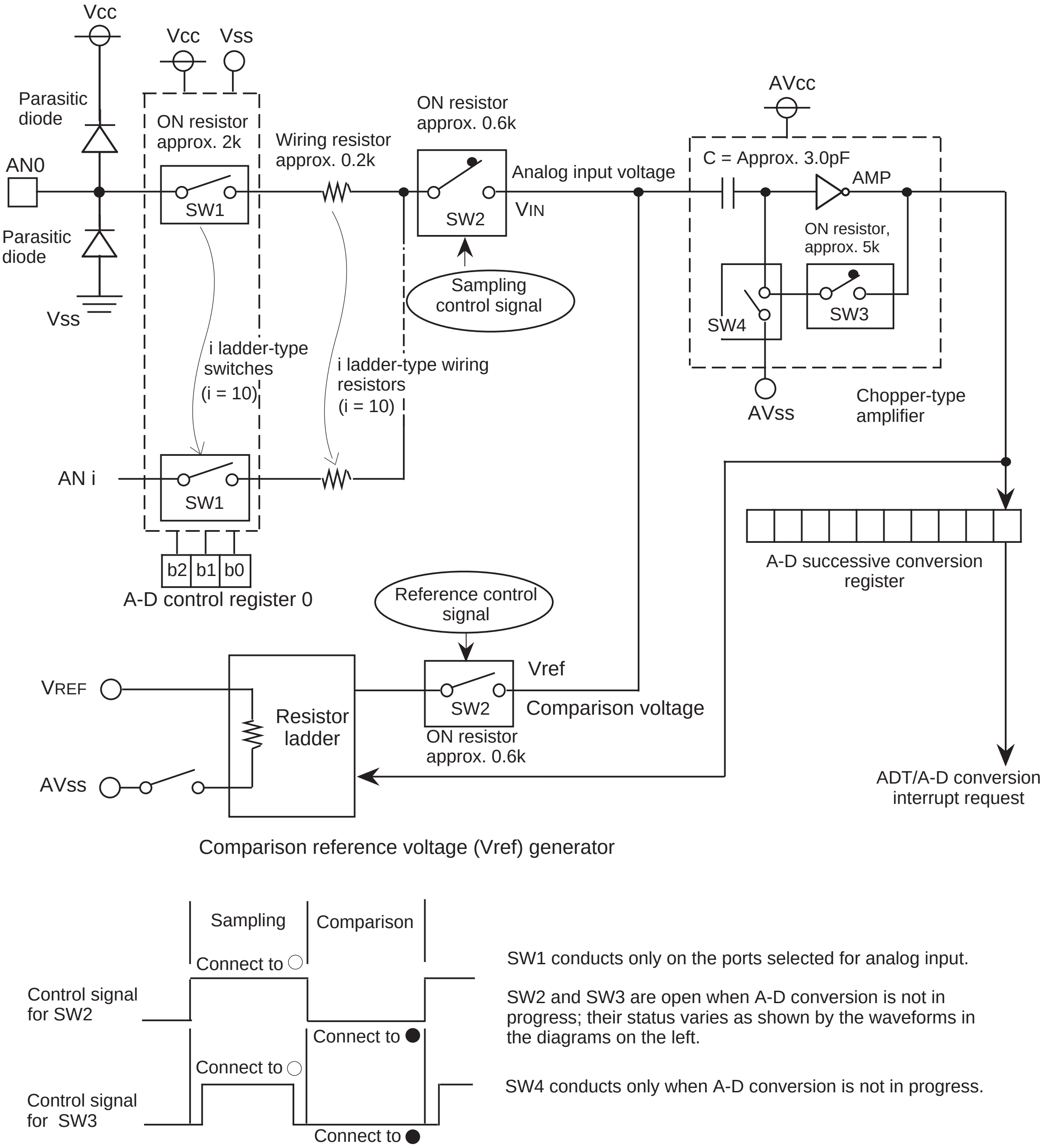


Figure 2.7.21. Differential non-linearity error (10-bit resolution)

2.7.11 Internal Equivalent Circuit of Analog Input

Figure 2.7.22 shows the internal equivalent circuit of analog input.



SW1 conducts only on the ports selected for analog input.

SW2 and SW3 are open when A-D conversion is not in progress; their status varies as shown by the waveforms in the diagrams on the left.

SW4 conducts only when A-D conversion is not in progress.

Warning: Use only as a standard for designing this data.
Mass production may cause some changes in device characteristics.

Figure 2.7.22. Internal equivalent circuit to analog input

Under development

2.7.12 Sensor’s Output Impedance under A-D Conversion

To carry out A-D conversion properly, charging the internal capacitor C shown in Figure 2.7.23 has to be completed within a specified period of time. With T as the specified time, time T is the time that switches SW2 and SW3 are connected to O in Figure 2.7.22. Let output impedance of sensor equivalent circuit be R0, microcomputer’s internal resistance be R, precision (error) of the A-D converter be X, and the A-D converter’s resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

Vc is generally $V_C = V_{IN} \{1 - e^{-\frac{t}{C\,(R0 + R)}}\}$

And when t = T, $V_C = V_{IN} - \frac{X}{Y} V_{IN} = V_{IN}(1 - \frac{X}{Y})$

$$e^{-\frac{T}{C\,(R0 + R)}} = \frac{X}{Y}$$

$$-\frac{T}{C\,(R0 + R)} = \ln \frac{X}{Y}$$

Hence, $R0 = -\dfrac{T}{C \bullet \ln \frac{X}{Y}} - R$

With the model shown in Figure 2.7.29 as an example, when the difference between V_{IN} and V_C becomes 0.1LSB, we find impedance R0 when voltage between pins V_C changes from 0 to V_{IN}-(0.1/1024) V_{IN} in time T. (0.1/1024) means that A-D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A-D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When f(X_{IN}) = 10 MHz, T = 0.3 us in the A-D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3 ms, R = 7.8 kW, C = 3 pF, X = 0.1, and Y = 1024 . Hence,

$$R0 = -\frac{0.3 \text{ X } 10^{-6}}{3.0 \text{ X } 10^{-12} \bullet \ln \frac{0.1}{1024}} - 7.8 \text{ X} 10^3 \fallingdotseq 3.0 \text{ X } 10^3$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A-D converter turns out to be approximately 3.0 kW. Tables 2.7.11 and 2.7.12 show output impedance values based on the LSB values.

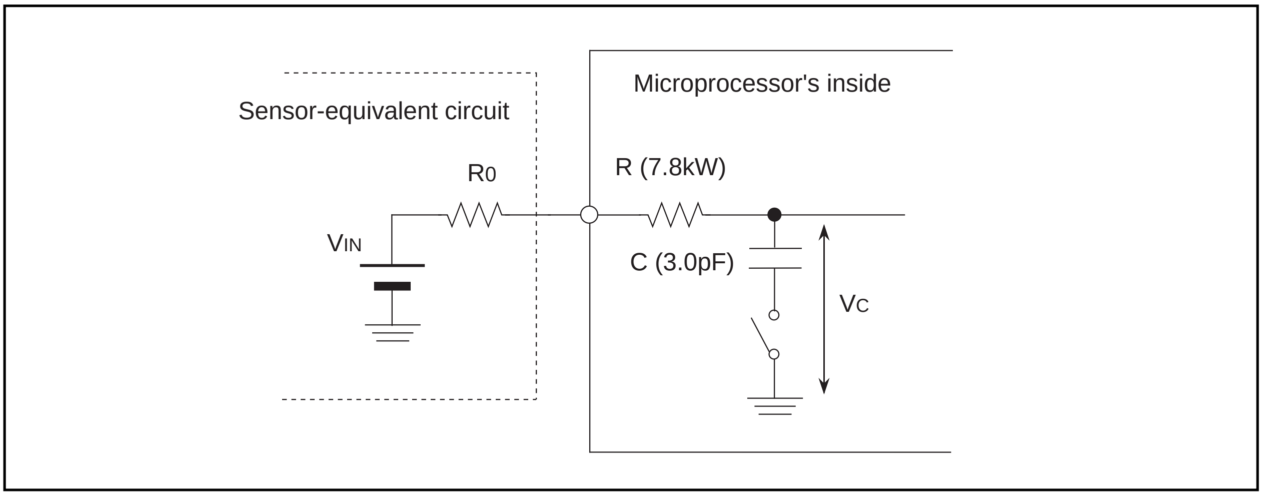


Figure 2.7.23 A circuit equivalent to the A-D conversion terminal

Tables 2.7.11. Output impedance values based on the LSB values (1)

f(Xin) (MHz)	Cycle (ns)	T	Ri + Rf (kohm)	C (pF)	Resolution (LSB)	R0max (kohm)
10	0.1	0.3 (3 x cycle, Sample & hold bit is enabled	7.8	3.0	0.1	3.0
					0.3	4.5
					0.5	5.3
					0.7	5.9
					0.9	6.4
					1.1	6.8
					1.3	7.2
					1.5	7.5
					1.7	7.8
					1.9	8.1
10	0.1	0.2 (2 x cycle, Sample & hold bit is disabled	7.8	3.0	0.1	-0.6
					0.3	0.4
					0.5	0.9
					0.7	1.3
					0.9	1.7
					1.1	2.0
					1.3	2.2
					1.5	2.4
					1.7	2.6
					1.9	2.8

Tables 2.7.12. Output impedance values based on the LSB values (2)

f(Xin) (MHz)	Cycle (ns)	T	Ri + Rf (kohm)	C (pF)	Resolution (LSB)	R0max (kohm)
10	0.1	0.3 (3 x cycle, Sample & hold bit is enabled	7.8	3.0	0.1	4.9
					0.3	7.0
					0.5	8.2
					0.7	9.1
					0.9	9.9
					1.1	10.5
					1.3	11.1
					1.5	11.7
					1.7	12.1
					1.9	12.6
10	0.1	0.2 (2 x cycle, Sample & hold bit is disabled	7.8	3.0	0.1	0.7
					0.3	2.1
					0.5	2.9
					0.7	3.5
					0.9	4.0
					1.1	4.4
					1.3	4.8
					1.5	5.2
					1.7	5.5
					1.9	5.8

2.8 Watchdog Timer

2.8.1 Overview

The watchdog timer can detect a runaway program using its 15-bit timer prescaler. The following is an overview of the watchdog timer.

(1) Watchdog timer start procedure

When reset, the watchdog timer is in stopped state. Writing to the watchdog timer start register initializes the watchdog timer to $7FFF_{16}$ and causes it to start performing a down count. The watchdog timer, once started operating, cannot be stopped by any means other than stopping conditions.

(2) Watchdog timer stop conditions

The watchdog timer stops in any one of the following states:

(a) Period in which the CPU is in stopped state

(b) Period in which the CPU is in waiting state

(3) Watchdog timer initialization

The watchdog timer is initialized to $7FFF_{16}$ in the cases given below, and begins a down count.

(a) When the watchdog timer writes to the watchdog timer start register while a count is in progress

(b) When the watchdog timer underflows

(4) Runaway detection

When the watchdog timer underflows, a watchdog timer interrupt occurs. In writing a program, write to the watchdog timer start register before the watchdog timer underflows. The watchdog timer interrupt occurs regardless of the status of the interrupt enable flag (I flag). In processing a watchdog timer interrupt, set the software reset bit to “1” to reset software.

(5) Watchdog timer cycle

The watchdog timer cycle varies depending on the BCLK and the frequency division ratio of the prescaler selected.

Table 2.8.1. The watchdog timer cycle

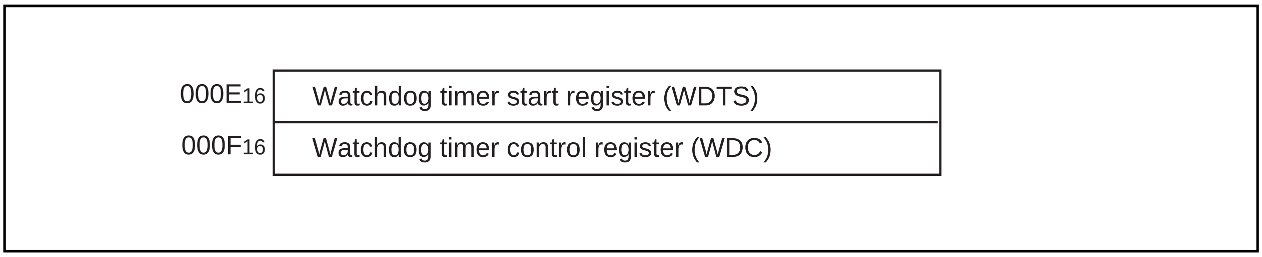
CM07	CM06	CM17	CM16	BCLK	WDC7	Period
0	0	0	0	10MHz	0	Approx. 52.4ms (Note)
					1	Approx. 419.2ms (Note)
0	0	0	1	5MHz	0	Approx. 104.9ms (Note)
					1	Approx. 838.8ms (Note)
0	0	1	0	2.5MHz	0	Approx. 209.7ms (Note)
					1	Approx. 1.68s (Note)
0	0	1	1	0.625MHz	0	Approx. 838.8ms (Note)
					1	Approx. 6.71s (Note)
0	1	Invalid	Invalid	1.25MHz	0	Approx. 419.2ms (Note)
					1	Approx. 3.35s (Note)
1	Invalid	Invalid	Invalid	32kHz	Invalid	Approx. 2s (Note)

Note: An error due to the prescaler occurs.

Under development

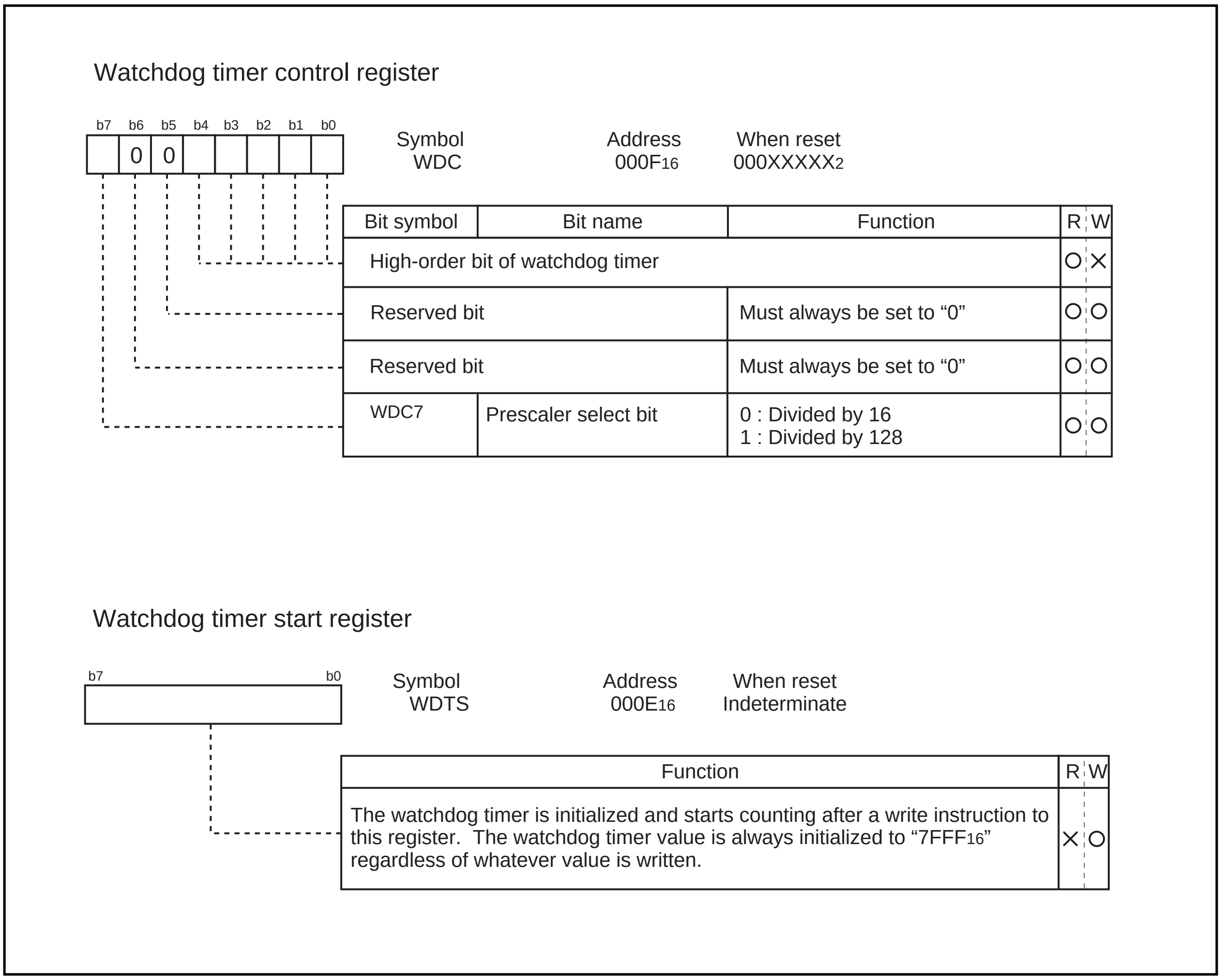
(6) Registers related to the watchdog timer

Figure 2.8.1 shows the memory map of watchdog timer-related registers, and Figure 2.8.2 shows watchdog timer-related registers.



$000E_{16}$	Watchdog timer start register (WDTS)
$000F_{16}$	Watchdog timer control register (WDC)

Figure 2.8.1. Memory map of watchdog timer-related registers



Watchdog timer control register

Symbol	Address	When reset
WDC	$000F_{16}$	$000XXXXX_2$

Bit symbol	Bit name	Function	R	W
High-order bit of watchdog timer			○	×
Reserved bit		Must always be set to “0”	○	○
Reserved bit		Must always be set to “0”	○	○
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	○	○

Watchdog timer start register

Symbol	Address	When reset
WDTS	$000E_{16}$	Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “$7FFF_{16}$” regardless of whatever value is written.	×	○

Figure 2.8.2. Watchdog timer-related registers

2.8.2 Operation of Watchdog Timer

The following is an operation of the watchdog timer. Figure 2.8.3 shows the operation timing, and Figure 2.8.4 shows the set-up procedure.

Operation
(1) Writing to the watchdog timer start register initializes the watchdog timer to $7FFF_{16}$ and causes it to start a down count.
(2) With a count in progress, writing to the watchdog timer start register again initializes the watchdog timer to $7FFF_{16}$ and causes it to resume counting.
(3) Either executing the WAIT instruction or going to the stopped state causes the watchdog timer to hold the count in progress and to stop counting. The watchdog timer resumes counting after returning from the execution of the WAIT instruction or from the stopped state.
(4) If the watchdog timer underflows, it is initialized to $7FFF_{16}$ and continues counting. At this time, a watchdog timer interrupt occurs.

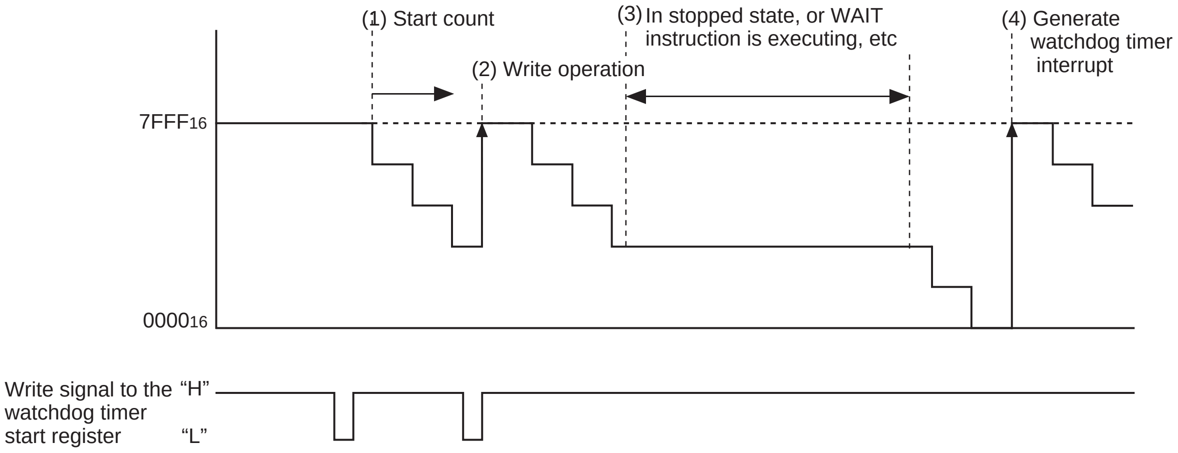
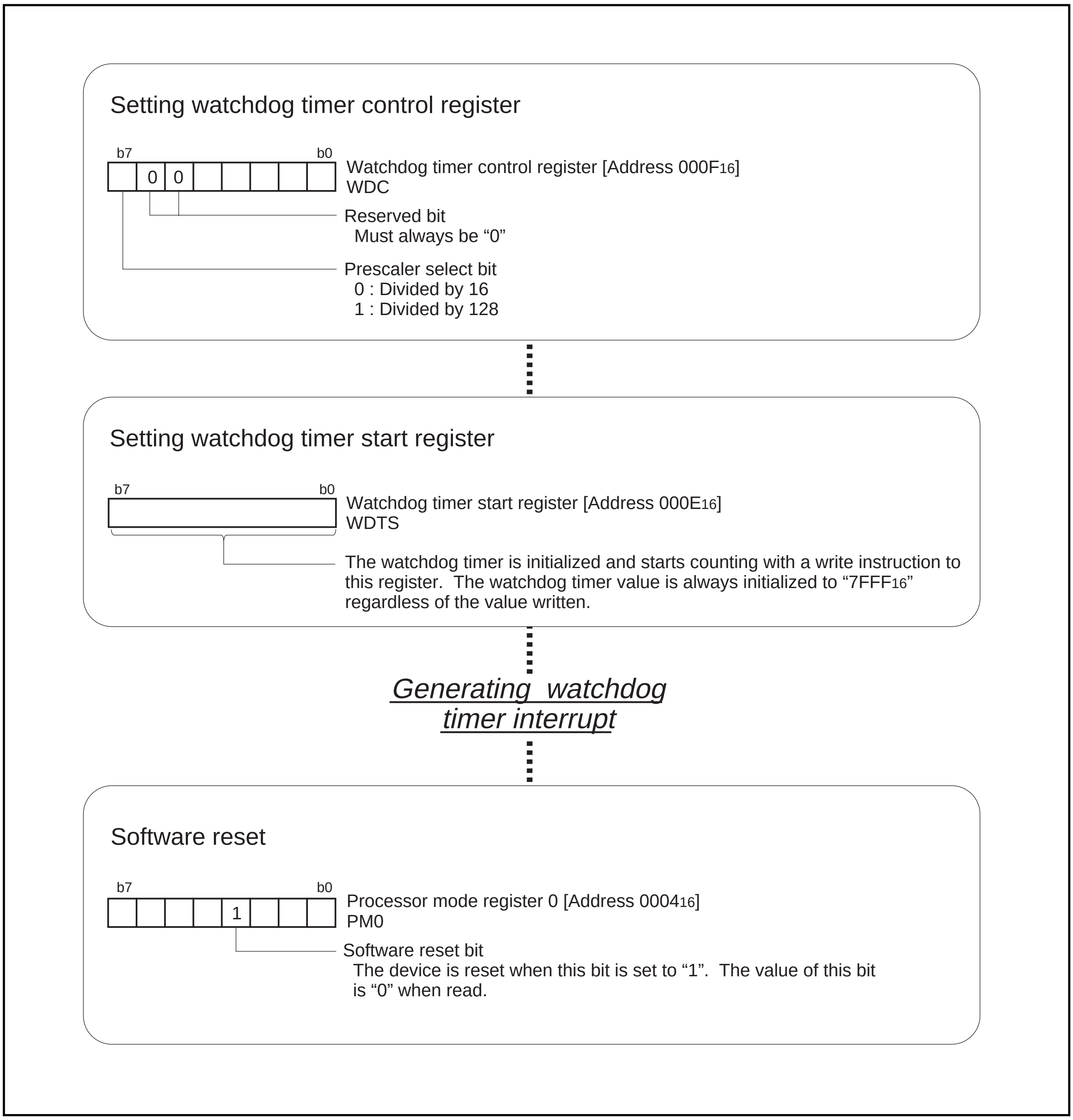

(1) Start count
(3) In stopped state, or WAIT instruction is executing, etc
(4) Generate watchdog timer interrupt
(2) Write operation
7FFF16
000016
Write signal to the watchdog timer start register
"H"
"L"

Figure 2.8.3. Operation timing of watchdog timer

Under development



Setting watchdog timer control register

Watchdog timer control register [Address 000F$_{16}$]
WDC

Reserved bit
Must always be “0”

Prescaler select bit
0 : Divided by 16
1 : Divided by 128

Setting watchdog timer start register

Watchdog timer start register [Address 000E$_{16}$]
WDTS

The watchdog timer is initialized and starts counting with a write instruction to this register. The watchdog timer value is always initialized to “7FFF$_{16}$” regardless of the value written.

Generating watchdog timer interrupt

Software reset

Processor mode register 0 [Address 0004$_{16}$]
PM0

Software reset bit
The device is reset when this bit is set to “1”. The value of this bit is “0” when read.

Figure 2.8.4. Set-up procedure of watchdog timer

2.9 Address Match Interrupt

2.9.1 Overview

The address match interrupt is used for correcting a ROM or for a simplified debugging-purpose monitor. The following is an overview of the address match interrupt.

(1) Enabling/disabling the address match interrupt

The address match interrupt enable bit can be used to enable and disable an address match interrupt. It is affected neither by the processor interrupt priority level (IPL) nor the interrupt enable flag (I flag).

(2) Timing of the address match interrupt

An interrupt occurs immediately before executing the instruction in the address indicated by the address match interrupt register. Set the first address of the instruction in the address match interrupt register. Setting a half address of an instruction or an address of tabulated data does not generate an address match interrupt.

The first instruction of an interrupt routine does not generate an address match interrupt either.

(3) Returning from an address match interrupt

The return address put in the stack when an address match interrupt occurs depends on the instruction not yet executed (the instruction the address match interrupt register indicates). The return address is not put in the stack. For this reason, to return from an address match interrupt, either rewrite the content of the stack and use the REIT instruction or use the POP instruction to restore the stack to the state as it was before the interrupt occurred and return by use of a jump instruction.

Figure 2.9.1 shows unexecuted instructions and corresponding the stacked addresses.

<Instructions whose address is added to by 2 when an address match interrupt occurs>

- 16-bit operation code instructions
- 8-bit operation code instructions given below

ADD.B:S	#IMM8,dest	SUB.B:S	#IMM8,dest	AND.B:S	#IMM8,dest
OR.B:S	#IMM8,dest	MOV.B:S	#IMM8,dest	STZ.B:S	#IMM8,dest
STNZ.B:S	#IMM8,dest	STZX.B:S	#IMM81,#IMM82,dest		
CMP.B:S	#IMM8,dest	PUSHM	src	POPM	dest
JMPS	#IMM8	JSRS	#IMM8		
MOV.B:S	#IMM,dest (However, dest = A0/A1)				

<Instructions whose address is added to by 1 when an address match interrupt occurs>

- Instructions other than those listed above

Figure 2.9.1. Unexecuted instructions and corresponding stacked addresses

(4) How to determine an address match interrupt

Address match interrupts can be set at two different locations. However, both location will have the same vector address. Therefore, it is necessary to determine which interrupt has occurred; address match interrupt 0 or address match interrupt 1. Using the content of the stack, etc., determine which interrupt has occurred according to the first part of the address match interrupt routine.

(5) Registers related to the address match interrupt

Figure 2.9.2 shows the memory map of address match interrupt-related registers, and Figure 2.9.3 shows address match interrupt-related registers.

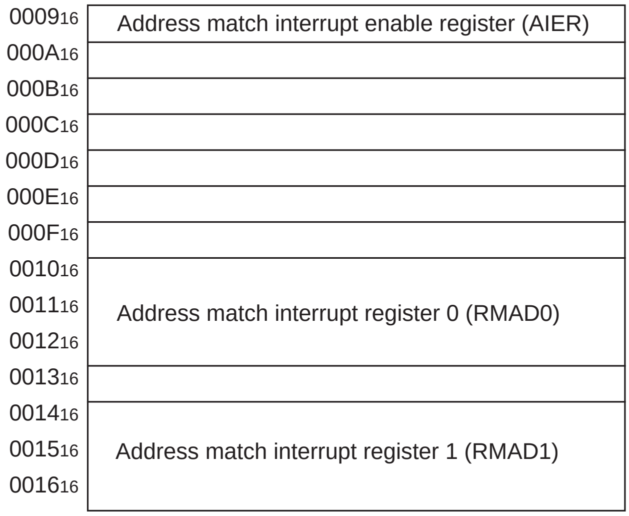
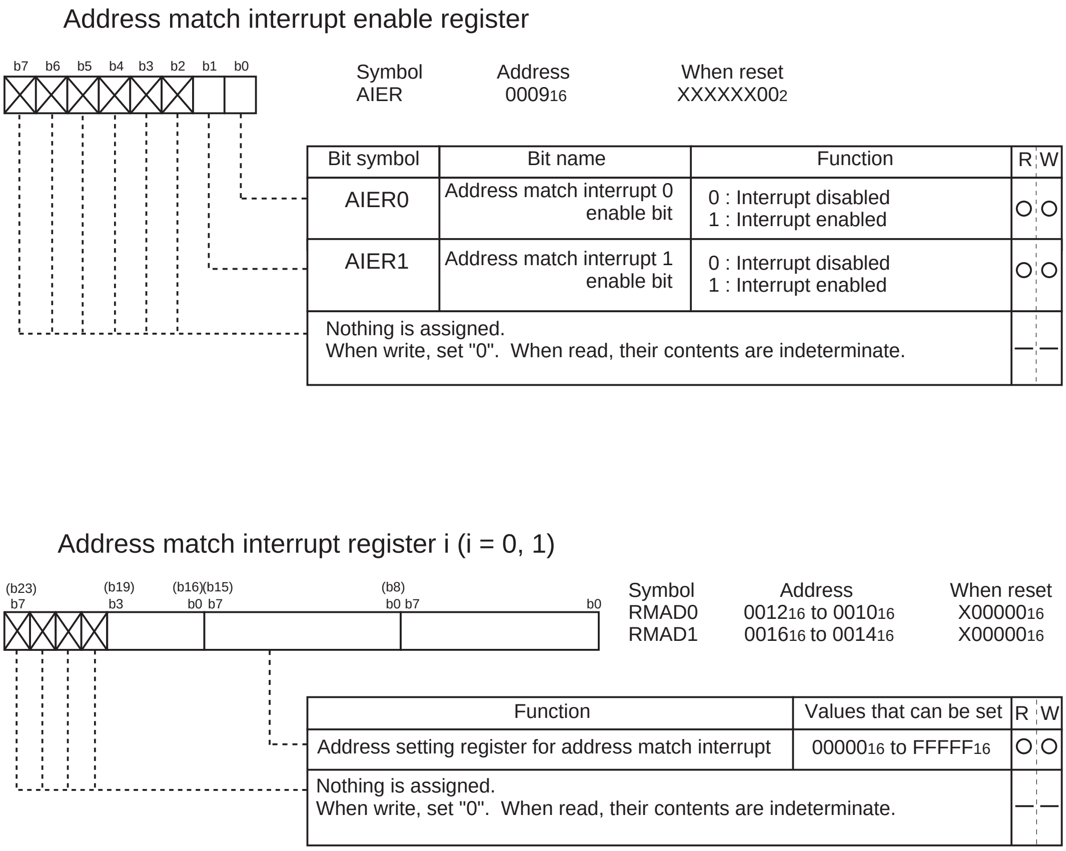


Figure 2.9.2. Memory map of address match interrupt-related registers


Address match interrupt enable register
b7 b6 b5 b4 b3 b2 b1 b0
Address match interrupt register i (i = 0, 1)
(b23) b7
(b19) b3
(b16)(b15) b0 b7
(b8) b0 b7
b0

Address match interrupt enable register

Symbol	Address	When reset
AIER	0009_{16}	$XXXXXX00_2$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Address match interrupt register i (i = 0, 1)

Symbol	Address	When reset
RMAD0	0012_{16} to 0010_{16}	$X00000_{16}$
RMAD1	0016_{16} to 0014_{16}	$X00000_{16}$

Function	Values that can be set	R	W
Address setting register for address match interrupt	00000_{16} to $FFFFF_{16}$	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.		—	—

Figure 2.9.3. Address match interrupt-related registers

Under development

2.9.2 Operation of Address Match Interrupt

The following is an operation of address match interrupt. Figure 2.9.4 shows the set-up procedure of address match interrupt, and Figure 2.9.5 shows the overview of the address match interrupt handling routine.

Operation (1) The address match interrupt handling routine sets an address to be used to cause the address match interrupt register to generate an interrupt.

(2) Setting the address match enable flag to “1” enables an interrupt to occur.

(3) An address match interrupt occurs immediately before the instruction in the address indicated by the address match interrupt register as a program is executed.

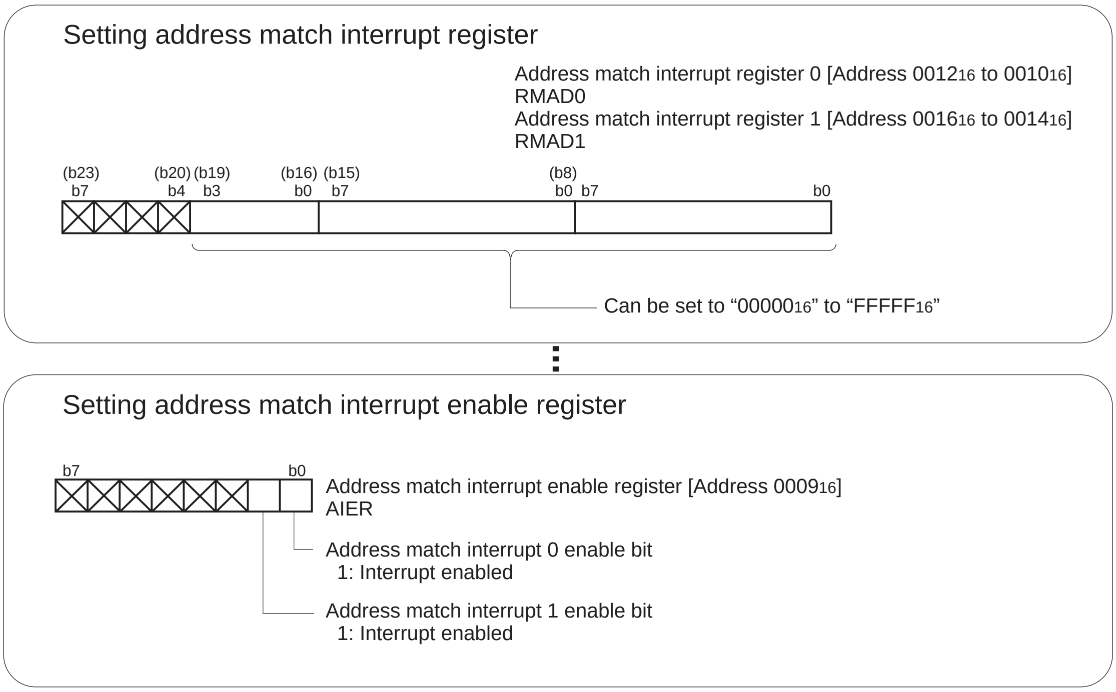
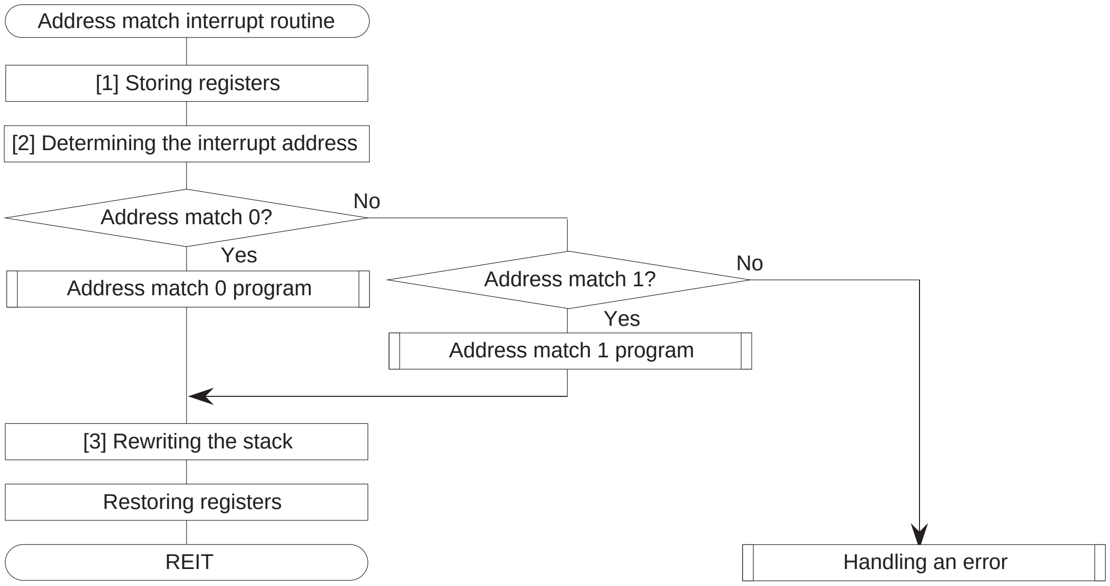

Setting address match interrupt register
Address match interrupt register 0 [Address 0012_{16} to 0010_{16}]
RMAD0
Address match interrupt register 1 [Address 0016_{16} to 0014_{16}]
RMAD1
(b23) b7
(b20) b4
(b19) b3
(b16) b0
(b15) b7
(b8) b0
b7
b0
Can be set to “00000_{16}” to “$FFFFF_{16}$”
Setting address match interrupt enable register
b7
b0
Address match interrupt enable register [Address 0009_{16}]
AIER
Address match interrupt 0 enable bit
1: Interrupt enabled
Address match interrupt 1 enable bit
1: Interrupt enabled

Figure 2.9.4. Set-up procedure of address match interrupt



Explanation:

[1] Storing the contents of the registers holding the main program status to be kept.

[2] Determining the interrupt address
Determining which factor generated the interrupt.

[3] Rewriting the stack
Rewriting the return address.

Figure 2.9.5. Overview of the address match interrupt handling routine

Under development

2.10 Key-Input Interrupt

2.10.1 Overview

Key-input interrupt occurs when a falling edge is input to $P0_0$ through $P0_7$. The following is an overview of the key-input interrupt:

(1) Enabling/disabling the key-input interrupt

The key-input interrupt can be enabled and disabled using the key-input interrupt register. The key-input interrupt is affected by the interrupt priority level (IPL) and the interrupt enable flag (I flag).

(2) Occurrence timing of the key-input interrupt

With key-input interrupt acceptance enabled, pins $P0_0$ through $P0_7$, which are set to input, become key-input interrupt pins ($\overline{KI_0}$ through $\overline{KI_7}$). A key-input interrupt occurs when a falling edge is input to a key-input interrupt pin. At this moment, the level of other key-input interrupt pins must be “H”. No interrupt occurs when the level of other key-input interrupt pins is “L”.

(3) How to determine a key-input interrupt

A key-input interrupt occurs when a falling edge is input to one of eight pins, but each pin has the same vector address.

Therefore, read the input level of pins $P0_0$ through $P0_7$ in the key-input interrupt routine to determine the interrupted pin.

(4) Registers related to the key-input interrupt

Figure 2.10.1 shows the memory map of key-input interrupt-related registers, and Figure 2.10.2 shows key-input interrupt-related registers.

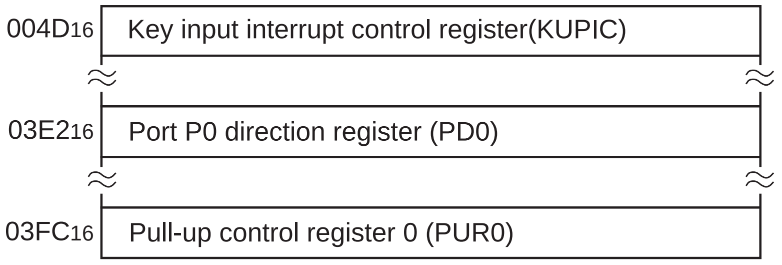
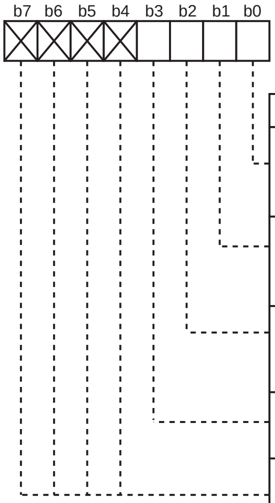

$004D_{16}$
Key input interrupt control register(KUPIC)
$03E2_{16}$
Port P0 direction register (PD0)
$03FC_{16}$
Pull-up control register 0 (PUR0)

Figure 2.10.1. Memory map of key-input interrupt-related registers

Interrupt control register

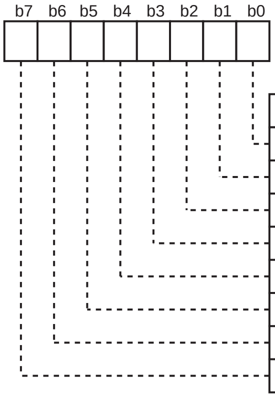


Symbol KUPIC
Address $004D_{16}$
When reset $XXXXX000_2$

Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note)
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

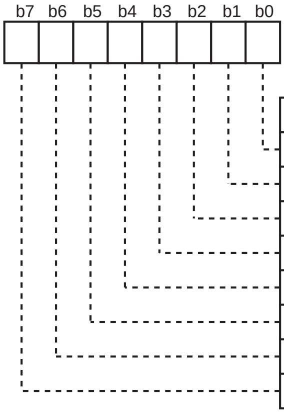
Port P0 direction register



Symbol PD0
Address $03E2_{16}$
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
PD0_0	Port $P0_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD0_1	Port $P0_1$ direction register		○	○
PD0_2	Port $P0_2$ direction register		○	○
PD0_3	Port $P0_3$ direction register		○	○
PD0_4	Port $P0_4$ direction register		○	○
PD0_5	Port $P0_5$ direction register		○	○
PD0_6	Port $P0_6$ direction register		○	○
PD0_7	Port $P0_7$ direction register		○	○

Pull-up control register 0



Symbol PUR0
Address $03FC_{16}$
When reset 00_{16}

Bit symbol	Bit name	Function	R	W
PU00	$P0_0$ to $P0_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
PU01	$P0_4$ to $P0_7$ pull-up		○	○
PU02	$P1_0$ to $P1_3$ pull-up		○	○
PU03	$P1_4$ to $P1_7$ pull-up		○	○
—	—		○	○
—	—		○	○
PU06	$P3_0$ to $P3_3$ pull-up		○	○
PU07	$P3_4$ to $P3_5$ pull-up		○	○

Figure 2.10.2. key-input interrupt-related registers

2.10.2 Operation of Key-Input Interrupt

The following is an operation of key-input interrupt. Figure 2.10.3 shows an example of a circuit that uses the key-input interrupt, Figure 2.10.4 shows an example of operation of key-input interrupt, and Figure 2.10.5 shows the setting procedure of key-input interrupt.

Operation (1) Set the direction register of the ports to be changed to key-input interrupt pins to input, and set the pull-up function.

(2) Setting the key-input interrupt control register and setting the interrupt enable flag makes the interrupt-enabled state ready.

(3) If a falling edge is input to either $\overline{KI_0}$ through $\overline{KI_7}$, the key-input interrupt request bit goes to “1”.

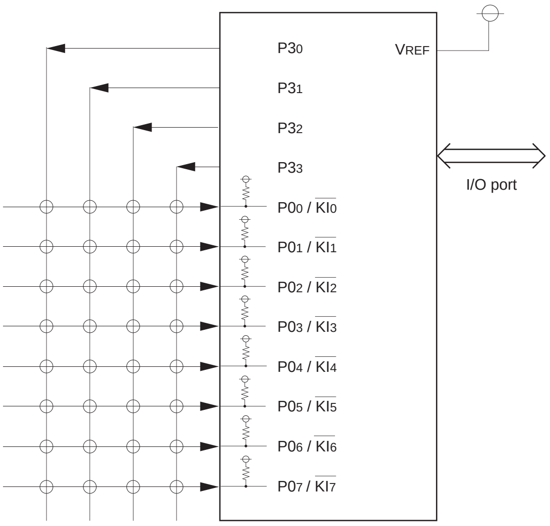


Figure 2.10.3. Example of circuit using the key-input interrupt

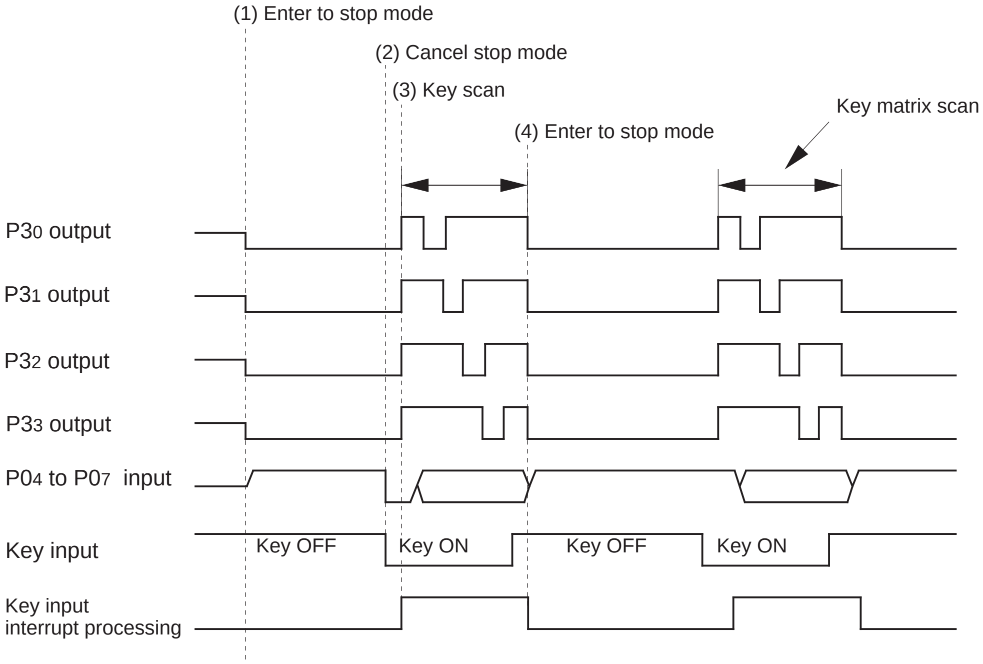


Figure 2.10.4. Example of operation of key-input interrupt

Under development

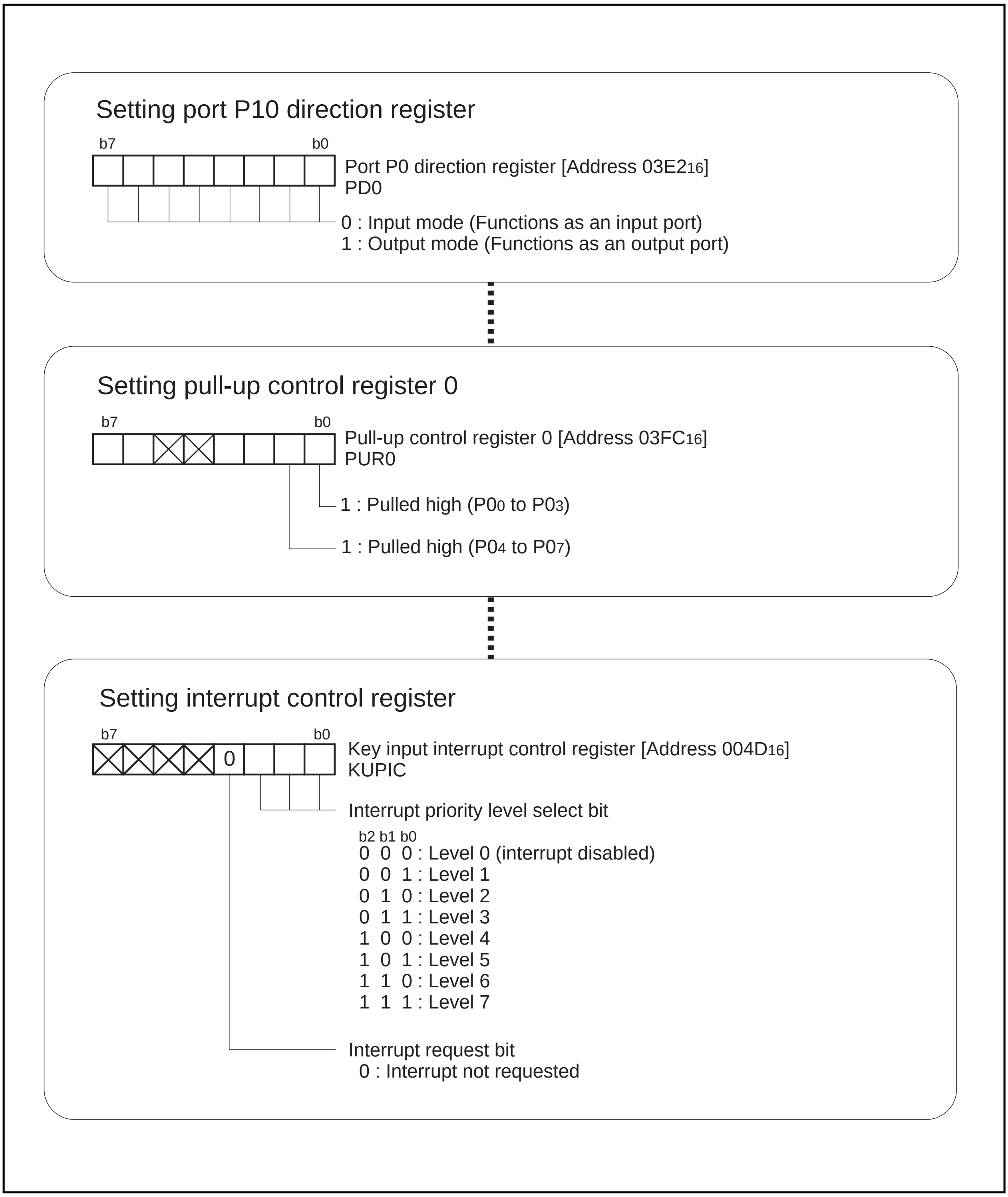

Setting port P10 direction register
b7
b0
Port P0 direction register [Address $03E2_{16}$]
PD0
0 : Input mode (Functions as an input port)
1 : Output mode (Functions as an output port)
Setting pull-up control register 0
b7
b0
Pull-up control register 0 [Address $03FC_{16}$]
PUR0
1 : Pulled high (P0_0 to P0_3)
1 : Pulled high (P0_4 to P0_7)
Setting interrupt control register
b7
b0
0
Key input interrupt control register [Address $004D_{16}$]
KUPIC
Interrupt priority level select bit
b2 b1 b0
0 0 0 : Level 0 (interrupt disabled)
0 0 1 : Level 1
0 1 0 : Level 2
0 1 1 : Level 3
1 0 0 : Level 4
1 0 1 : Level 5
1 1 0 : Level 6
1 1 1 : Level 7
Interrupt request bit
0 : Interrupt not requested

Figure 2.10.5. Set-up procedure of key-input interrupt

2.11 Power Control

2.11.1 Overview

'Power Control' refers to the reduction of CPU power consumption by stopping the CPU and oscillators, or decreasing the operation clock. The following is a description of the three available power control modes:

(1) Modes

Power control is available in three modes.

(a) Normal operation mode

• High-speed mode

Divide-by-1 frequency of the main clock becomes the BCLK. The CPU operates with the BCLK selected. Each peripheral function operates according to its assigned clock.

• Medium-speed mode

Divide-by-2, divide-by-4, divide-by-8, or divide-by-16 frequency of the main clock becomes the BCLK. The CPU operates according to the BCLK selected. Each peripheral function operates according to its assigned clock.

• Low-speed mode

fc becomes the BCLK. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. Each peripheral function operates according to its assigned clock.

• Low power consumption mode

The main clock operating in low-speed mode is stopped. The CPU operates according to the fc clock. The fc clock is supplied by the secondary clock. The only peripheral functions that operate are those with the sub-clock selected as the count source.

(b) Wait mode

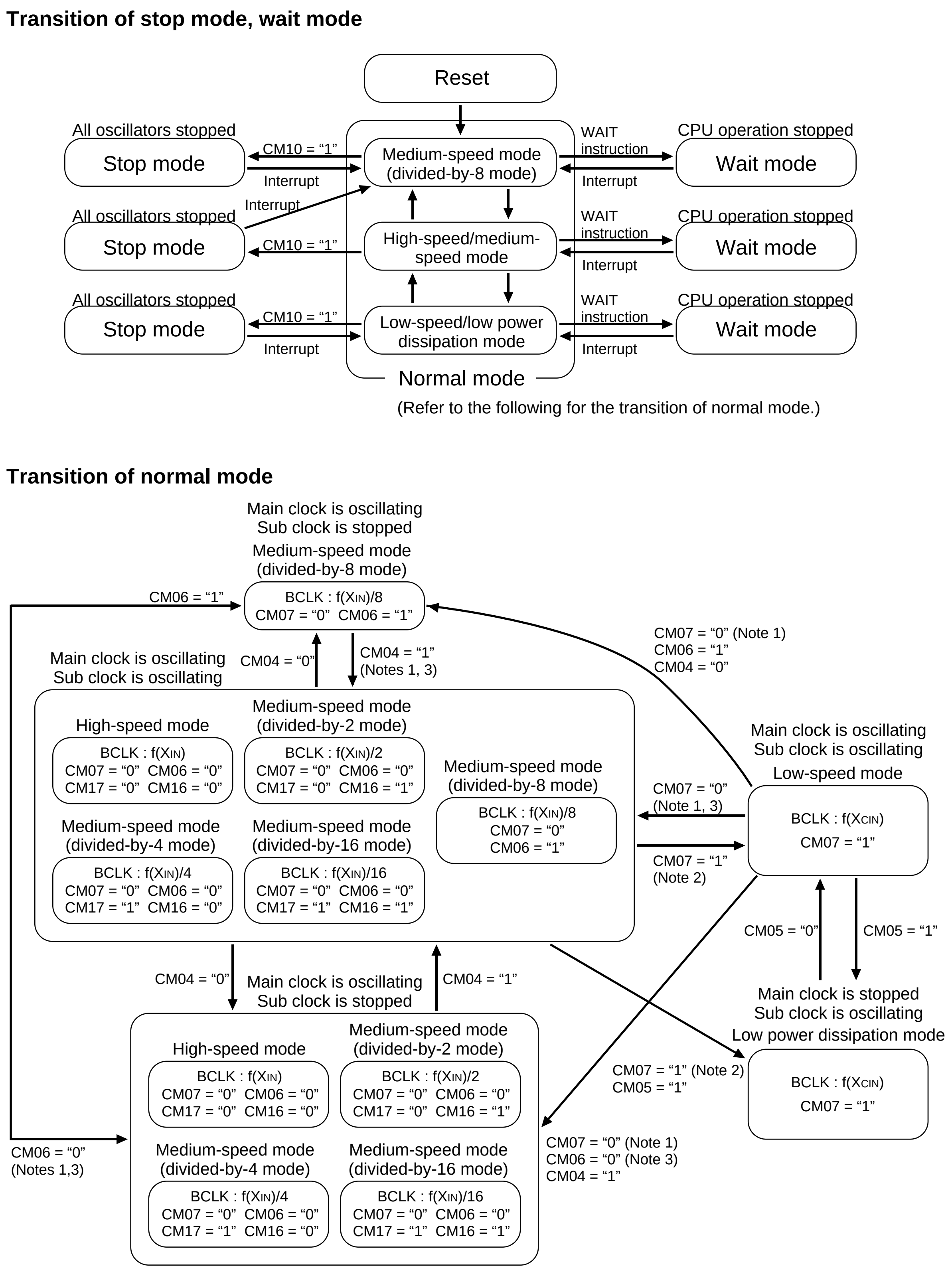
The CPU operation is stopped. The oscillators do not stop.

(c) Stop mode

All oscillators stop. The CPU and all built-in peripheral functions stop. This mode, among the three modes listed here, is the most effective in decreasing power consumption.

Figure 2.11.1 is the state transition diagram of the above modes.

Under development



Note 1: Switch clock after oscillation of main clock is sufficiently stable.
Note 2: Switch clock after oscillation of sub clock is sufficiently stable.
Note 3: Change CM06 after changing CM17 and CM16.
Note 4: Transit in accordance with arrow.

Figure 2.11.1. State transition diagram of power control mode

(2) Switching the driving capacity of the oscillation circuit

Both the main clock and the secondary clock have the ability to switch the driving capacity. Reducing the driving capacity after the oscillation stabilizes allows for further reduction in power consumption.

(3) Clearing stop mode and wait mode

The stop mode and wait mode can be cleared by generating an interrupt request, or by resetting hardware. Set the priority level of the interrupt to be used for clearing, higher than the processor interrupt priority level (IPL), and enable the interrupt enable flag (I flag). When an interrupt clears a mode, that interrupt is processed. Table 2.11.1 shows the interrupts that can be used for clearing a stop mode and wait mode.

(4) BCLK in returning from wait mode or stop mode

(a) Returning from wait mode

The processor immediately returns to the BCLK, which was in use before entering wait mode.

(b) Returning from stop mode

If operation was performed in the high speed mode or medium speed mode prior to engaging the stop mode, CM06 will change to “1” when operation shifts to the stop mode. CM17, CM16 and CM07 do not change. Accordingly, when operation is restored from the stop mode, operation starts in the 8 division mode.

Also, if operation was performed in the low speed mode prior to engaging the stop mode, CM06, CM17, CM16 and CM07 do not change. When operation is restored from the stop mode, operation starts in the low speed mode.

Table 2.11.1. Interrupts available for clearing stop mode and wait mode

Interrupt for clearing	Wait mode		Stop mode
	CM02 = 0	CM02 = 1	
Key input interrupt	Possible	Possible	Possible
A-D interrupt	Note 3	Impossible	Impossible
UART0 transmit interrupt	Possible	Note 1	Note 1
UART0 receive interrupt	Possible	Note 1	Note 1
UART1 transmit interrupt	Possible	Impossible	Impossible
UART1 receive interrupt	Possible	Impossible	Impossible
Timer A0 interrupt	Possible	Note 2	Note 2
Timer B0 interrupt	Possible	Note 2	Note 2
Timer B1 interrupt	Possible	Note 2	Note 2
Timer X0 interrupt	Possible	Note 2	Note 2
Timer X1 interrupt	Possible	Note 2	Note 2
Timer X2 interrupt	Possible	Note 2	Note 2
$\overline{\text{INT0}}$ interrupt	Possible	Possible	Possible
$\overline{\text{INT1}}$ interrupt	Possible	Possible	Possible

Note 1: Can be used when an external clock in clock synchronous serial I/O mode is selected.
Note 2: Can be used when the external signal is being counted in event counter mode.
Note 3: Can be used in one-shot mode and one-shot sweep mode.

(5) Sequence of returning from stop mode

Sequence of returning from stop mode is oscillation start-up time and interrupt sequence.
When interrupt is generated in stop mode, CM10 becomes “0” and clearing stop mode.
Starting oscillation and supplying BCLK execute the interrupt sequence as follow:

In the interrupt sequence, the processor carries out the following in sequence given:

(a) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000_{16}. The interrupt request bit of the interrupt written in address 00000_{16} will then be set to “0”.
(b) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
(c) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer assignment flag (U flag) to “0” (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
(d) Saves the content of the temporary register (Note) within the CPU in the stack area.
(e) Saves the content of the program counter (PC) in the stack area.
(f) Sets the interrupt priority level of the accepted instruction in the IPL.

Note: This register cannot be utilized by the user.
After the interrupt sequence is completed, the processor resumes executing instructions from the first address of the interrupt routine.

Figure 2.11.2 shows the sequence of returning from stop mode.

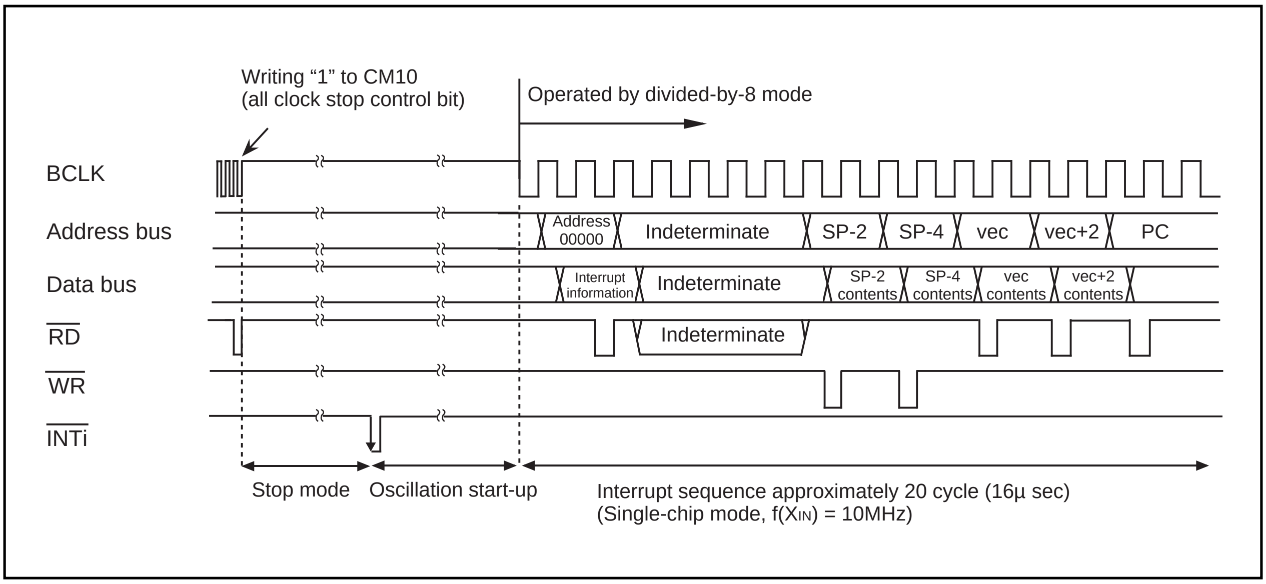


Figure 2.11.2. Sequence of returning from stop mode

(6) Registers related to power control

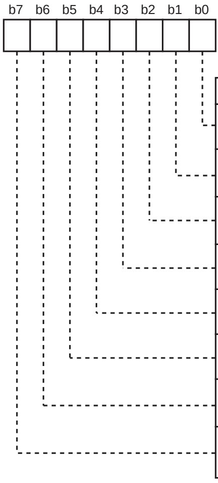
Figure 2.11.3 shows the memory map of power control-related registers, and Figure 2.11.4 shows power control-related registers.

Under development

0006$_{16}$	System clock control register 0 (CM0)
0007$_{16}$	System clock control register 1 (CM1)

Figure 2.11.3. Memory map of power control-related registers

System clock control register 0 (Note 1)

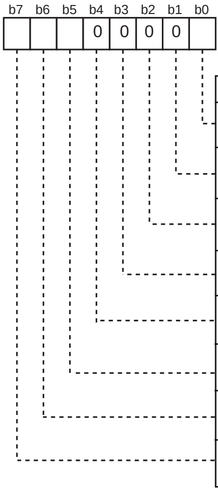


Symbol	Address	When reset
CM0	0006$_{16}$	48$_{16}$

Bit symbol	Bit name	Function	R	W
CM00	Clock output function select bit	b1 b0 0 0 : I/O port P5$_4$ 0 1 : fc output 1 0 : f$_8$ output 1 1 : Clock divide counter output	○	○
CM01			○	○
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral function clock in wait mode 1 : Stop peripheral function clock in wait mode (Note 8)	○	○
CM03	X$_{CIN}$-X$_{COUT}$ drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM04	Port X$_C$ select bit	0 : I/O port 1 : X$_{CIN}$-X$_{COUT}$ generation	○	○
CM05	Main clock (X$_{IN}$-X$_{OUT}$) stop bit (Note 3,4,5)	0 : On 1 : Off	○	○
CM06	Main clock division select bit 0 (Note 7)	0 : CM16 and CM17 valid 1 : Division by 8 mode	○	○
CM07	System clock select bit (Note 6)	0 : X$_{IN}$, X$_{OUT}$ 1 : X$_{CIN}$, X$_{COUT}$	○	○

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shifting to stop mode and at a reset.
Note 3: This bit is used to stop the main clock when placing the device in a low-power mode. If you want to operate with X$_{IN}$ after exiting from the stop mode, set this bit to “0”. When operating with a self-excited oscillator, set the system clock select bit (CM07) to “1” before setting this bit to “1”.
Note 4: When inputting external clock, only clock oscillation buffer is stopped and clock input is acceptable.
Note 5: If this bit is set to “1”, X$_{OUT}$ turns “H”. The built-in feedback resistor remains being connected, so X$_{IN}$ turns pulled up to X$_{OUT}$ (“H”) via the feedback resistor.
Note 6: Set port Xc select bit (CM04) to “1” and stabilize the sub-clock oscillating before setting to this bit from “0” to “1”. Do not write to both bits at the same time. And also, set the main clock stop bit (CM05) to “0” and stabilize the main clock oscillating before setting this bit from “1” to “0”.
Note 7: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 8: fc$_{32}$ is not included.

System clock control register 1 (Note 1)



Symbol	Address	When reset
CM1	0007$_{16}$	20$_{16}$

Bit symbol	Bit name	Function	R	W
CM10	All clock stop control bit (Note 4)	0 : Clock on 1 : All clocks off (stop mode)	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
Reserved bit		Always set to “0”	○	○
CM15	X$_{IN}$-X$_{OUT}$ drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
CM16	Main clock division select bit 1 (Note 3)	b7 b6 0 0 : No division mode 0 1 : Division by 2 mode 1 0 : Division by 4 mode 1 1 : Division by 16 mode	○	○
CM17				

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: This bit changes to “1” when shifting from high-speed/medium-speed mode to stop mode and at a reset. When shifting from low-speed/low power dissipation mode to stop mode, the value before stop mode is retained.
Note 3: Can be selected when bit 6 of the system clock control register 0 (address 000616) is “0”. If “1”, division mode is fixed at 8.
Note 4: If this bit is set to “1”, X$_{OUT}$ turns “H”, and the built-in feedback resistor is cut off. X$_{CIN}$ and X$_{COUT}$ turn high-impedance state.

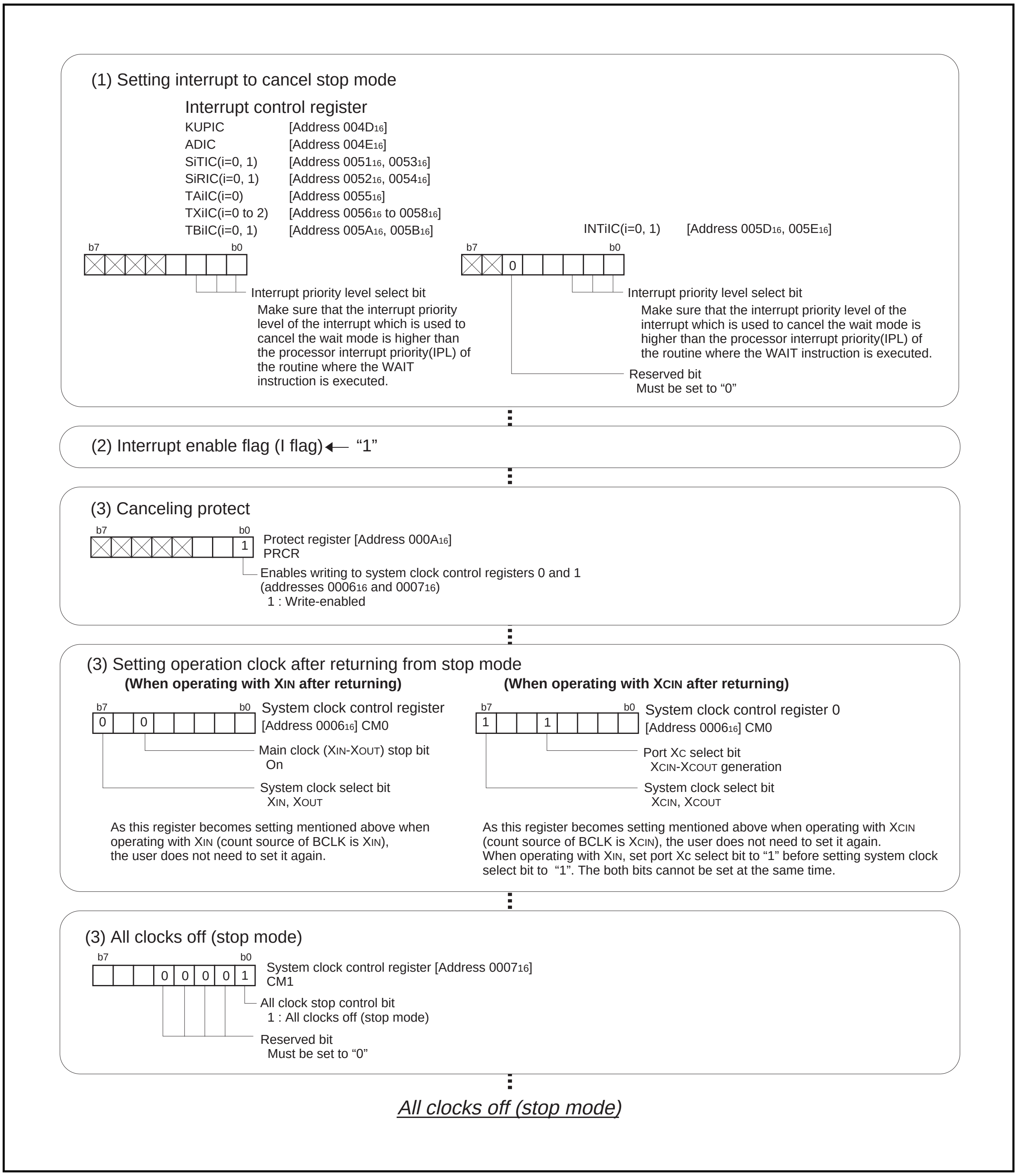
Figure 2.11.4. Power control-related registers

Under development

2.11.2 Stop Mode Set-Up

Settings and operation for entering stop mode are described here.

Operation (1) Enables the interrupt used for returning from stop mode.
(2) Sets the interrupt enable flag (I flag) to “1”.
(3) Clearing the protection and setting every-clock stop bit to “1” stops oscillation and causes the processor to go into stop mode.



(1) Setting interrupt to cancel stop mode

Interrupt control register

KUPIC	[Address 004D16]
ADIC	[Address 004E16]
SiTIC(i=0, 1)	[Address 005116, 005316]
SiRIC(i=0, 1)	[Address 005216, 005416]
TAiIC(i=0)	[Address 005516]
TXiIC(i=0 to 2)	[Address 005616 to 005816]
TBiIC(i=0, 1)	[Address 005A16, 005B16]

Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority(IPL) of the routine where the WAIT instruction is executed.

INTiIC(i=0, 1) [Address 005D16, 005E16]

Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority(IPL) of the routine where the WAIT instruction is executed.

Reserved bit
Must be set to “0”

(2) Interrupt enable flag (I flag) ← “1”

(3) Canceling protect

Protect register [Address 000A16]
PRCR

Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716)
1 : Write-enabled

(3) Setting operation clock after returning from stop mode

(When operating with XIN after returning)

System clock control register
[Address 000616] CM0

Main clock (XIN-XOUT) stop bit
On

System clock select bit
XIN, XOUT

As this register becomes setting mentioned above when operating with XIN (count source of BCLK is XIN), the user does not need to set it again.

(When operating with XCIN after returning)

System clock control register 0
[Address 000616] CM0

Port XC select bit
XCIN-XCOUT generation

System clock select bit
XCIN, XCOUT

As this register becomes setting mentioned above when operating with XCIN (count source of BCLK is XCIN), the user does not need to set it again.
When operating with XIN, set port Xc select bit to “1” before setting system clock select bit to “1”. The both bits cannot be set at the same time.

(3) All clocks off (stop mode)

System clock control register [Address 000716]
CM1

All clock stop control bit
1 : All clocks off (stop mode)

Reserved bit
Must be set to “0”

All clocks off (stop mode)

Figure 2.11.5. Example of stop mode set-up

2.11.3 Wait Mode Set-Up

Settings and operation for entering wait mode are described here.

Operation (1) Enables the interrupt used for returning from wait mode.
(2) Sets the interrupt enable flag (I flag) to “1”.
(3) Clears the protection and changes the content of the system clock control register.
(4) Executes the WAIT instruction.

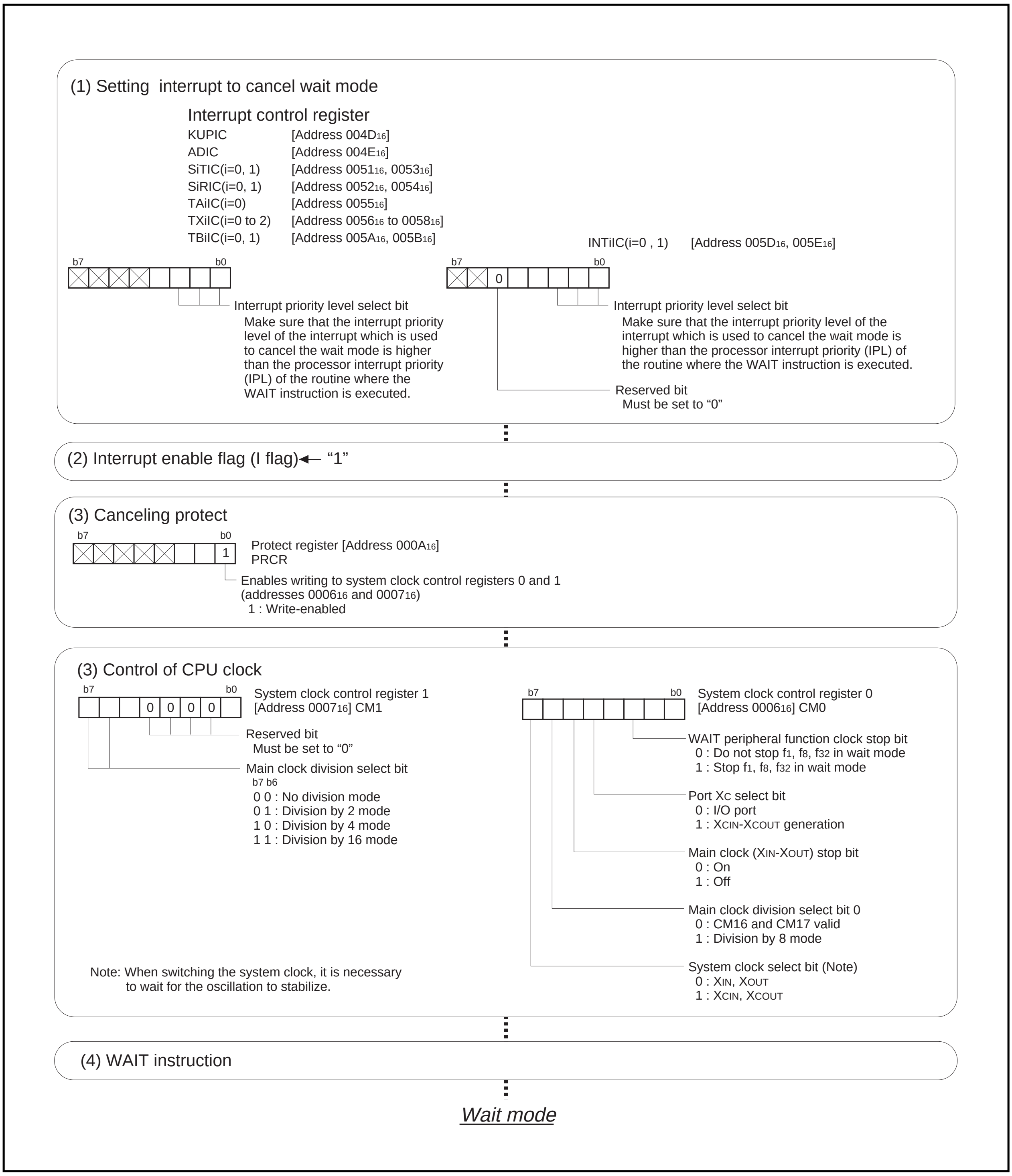

(1) Setting interrupt to cancel wait mode
Interrupt control register
KUPIC [Address 004D16]
ADIC [Address 004E16]
SiTIC(i=0, 1) [Address 005116, 005316]
SiRIC(i=0, 1) [Address 005216, 005416]
TAiIC(i=0) [Address 005516]
TXiIC(i=0 to 2) [Address 005616 to 005816]
TBiIC(i=0, 1) [Address 005A16, 005B16]
INTiIC(i=0 , 1) [Address 005D16, 005E16]
b7
b0
0
Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority (IPL) of the routine where the WAIT instruction is executed.
Interrupt priority level select bit
Make sure that the interrupt priority level of the interrupt which is used to cancel the wait mode is higher than the processor interrupt priority (IPL) of the routine where the WAIT instruction is executed.
Reserved bit
Must be set to “0”
(2) Interrupt enable flag (I flag) ← “1”
(3) Canceling protect
1
Protect register [Address 000A16]
PRCR
Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716)
1 : Write-enabled
(3) Control of CPU clock
0 0 0 0
System clock control register 1
[Address 000716] CM1
Reserved bit
Must be set to “0”
Main clock division select bit
b7 b6
0 0 : No division mode
0 1 : Division by 2 mode
1 0 : Division by 4 mode
1 1 : Division by 16 mode
System clock control register 0
[Address 000616] CM0
WAIT peripheral function clock stop bit
0 : Do not stop f1, f8, f32 in wait mode
1 : Stop f1, f8, f32 in wait mode
Port XC select bit
0 : I/O port
1 : XCIN-XCOUT generation
Main clock (XIN-XOUT) stop bit
0 : On
1 : Off
Main clock division select bit 0
0 : CM16 and CM17 valid
1 : Division by 8 mode
System clock select bit (Note)
0 : XIN, XOUT
1 : XCIN, XCOUT
Note: When switching the system clock, it is necessary to wait for the oscillation to stabilize.
(4) WAIT instruction
Wait mode

Figure 2.11.6. Example of wait mode set-up

2.11.4 Precautions in Power Control

(1) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to “L” level until main clock oscillation is stabilized.

(2) When switching to either wait mode or stop mode, instructions occupying four bytes either from the WAIT instruction or from the instruction that sets the every-clock stop bit to “1” within the instruction queue are prefetched and then the program stops. So put at least four NOPs in succession either to the WAIT instruction or to the instruction that sets the every-clock stop bit to “1”.

(3) Suggestions to reduce power consumption

- **Ports**

  The processor retains the state of each programmable I/O port even when it goes to wait mode or to stop mode. A current flows in active I/O ports. A pass current flows in input ports that float. When entering wait mode or stop mode, set non-used ports to input and stabilize the potential.

(a) A-D converter

A current always flows in the VREF pin. When entering wait mode or stop mode, set the Vref connection bit to “0” so that no current flows into the VREF pin.

(b) Stopping peripheral functions

In wait mode, stop non-used wait peripheral functions using the peripheral function clock stop bit.

(c) Switching the oscillation-driving capacity

Set the driving capacity to “LOW” when oscillation is stable.

(d) External clock

When using an external clock input for the CPU clock, set the main clock stop bit to “1”. Setting the main clock stop bit to “1” causes the XOUT pin not to operate and the power consumption goes down (when using an external clock input, the clock signal is input regardless of the content of the main clock stop bit).

2.12 Programmable I/O Ports

2.12.1 Overview

Fourty-three programmable I/O ports. I/O pins also serve as I/O pins for built-in peripheral functions. Each port has a direction register that defines the I/O direction and also has a port register for I/O data. In addition, each port has a pull-up control register that defines pull-up in terms of 4 bits. Port P1 can be set to N-channel output transistor drive capacity.

The following is an overview of the programmable I/O ports:

(1) Writing to a port register

With the direction register set to output, the level of the written values from each relevant pin is output by writing to a port register. The output level conforms to CMOS output. Writing to the port register, with the direction register set to input, inputs a value to the port register, but nothing is output to the relevant pins. The output level remains floating.

(2) Reading a port register

With the direction register set to output, reading a port register takes out the content of the port register, not the content of the pin. With the direction register set to input, reading the port register takes out the content of the pin.

(3) Effect of the protection register

Data written to the direction register of P4 is affected by the protection register. The direction register of P4 cannot be easily rewritten.

(4) Setting pull-up

The pull-up control bit allows setting of the pull-up, in terms of 4 bits, either in use or not in use. For the four bits chosen, pull-up is effective only in the ports whose direction register is set to input. Pull-up is not effective in ports whose direction register is set to output.

Do not set pull-up of corresponding pin when X_{CIN}/X_{COUT} is set or a port is used as A-D input.

(5) Drive capacity control

The drive capacity of the N channel output transistor on P1 can be set between “LOW” and “HIGH” in units of 1 bit. One bit corresponds to one pin.

(6) I/O functions of built-in peripheral devices

Table 2.12.1 shows relation between ports and I/O functions of built-in peripheral devices.

Table 2.12.1. Relation between ports and I/O functions of built-in peripheral devices

Port	Internal peripheral device I/O pins
P0	key-input interrupt function input pins
$P4_0$	I/O pin for serial I/O communication/Timer A input pin
$P4_1$	Timer A output pin
$P4_2$	Serial I/O input pin
$P4_3$, $P4_4$	Input pins for external interrupt/Timer X I/O pins
$P4_5$	Timer X I/O pin
$P5_0$ to $P5_4$	I/O pins for serial I/O communication/A-D converter input pins
P6	A-D converter input pins
$P7_0$, $P7_1$	Timer B input pins

(7) Examples of working on non-used pins

Table 2.12.2 contains examples of working on non-used pins. There are shown here for mere examples. In practical use, make suitable changes and perform sufficient evaluation in compliance with you application.

Table 2.12.2. Examples of working on unused pins in single-chip mode

Pin name	Connection
Ports P0, P1, P3 to P7	After setting for input mode, connect every pin to V_{SS} or V_{CC} via a resistor; or after setting for output mode, leave these pins open. (Note 1)
X_{OUT} (Note 2)	Open
AV_{CC}	Connect to V_{CC}
AV_{SS}, V_{REF}, BYTE	Connect to V_{SS}

Note 1: If setting these pins in output mode and opening them, ports are in input mode until switched into output mode by use of software after reset. Thus the voltage levels of the pins become unstable, and there can be instances in which the power source current increases while the ports are in input mode.
In view of an instance in which the contents of the direction registers change due to a runaway generated by noise or other causes, setting the contents of the direction registers periodically by use of software increases program reliability.

Note 2: When an external clock is input to the X_{IN} pin.

(8) Registers related to the programmable I/O ports

Figure 2.12.1 shows the memory map of programmable I/O ports-related registers, and Figures 2.12.2 to 2.12.4 show programmable I/O ports-related registers.

$03E0_{16}$	Port P0 (P0)
$03E1_{16}$	Port P1 (P1)
$03E2_{16}$	Port P0 direction register (PD0)
$03E3_{16}$	Port P1 direction register (PD1)
$03E4_{16}$	
$03E5_{16}$	Port P3 (P3)
$03E6_{16}$	
$03E7_{16}$	Port P3 direction register (PD3)
$03E8_{16}$	Port P4 (P4)
$03E9_{16}$	Port P5 (P5)
$03EA_{16}$	Port P4 direction register (PD4)
$03EB_{16}$	Port P5 direction register (PD5)
$03EC_{16}$	Port P6 (P6)
$03ED_{16}$	Port P7 (P7)
$03EE_{16}$	Port P6 direction register (PD6)
$03EF_{16}$	Port P7 direction register (PD7)
≈	≈
$03FC_{16}$	Pull-up control register 0 (PUR0)
$03FD_{16}$	Pull-up control register 1 (PUR1)
$03FE_{16}$	Port P1 drive control register (DRR)

Figure 2.12.1. Memory map of programmable I/O ports-related registers

Under development

Port Pi direction register (Note 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PDi (i = 0 to 7)	03E2$_{16}$, 03E3$_{16}$, 03E7$_{16}$, 03EA$_{16}$,	00$_{16}$
	03EB$_{16}$, 03EE$_{16}$, 03EF$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 7 except 2)	○	○
PDi_1	Port Pi$_1$ direction register		○	○
PDi_2	Port Pi$_2$ direction register		○	○
PDi_3	Port Pi$_3$ direction register		○	○
PDi_4	Port Pi$_4$ direction register		○	○
PDi_5	Port Pi$_5$ direction register		○	○
PDi_6	Port Pi$_6$ direction register		○	○
PDi_7	Port Pi$_7$ direction register		○	○

Note 1: Set bit 2 of protect register (address 000A$_{16}$) to “1” before rewriting to the port P4 direction register.

Note 2: Nothing is assigned in direction register of P3$_6$, P3$_7$, P4$_6$, P4$_7$, P5$_5$ to p5$_7$, P7$_2$ to P7$_7$. These bits can either be set nor reset. When read, its contents are indeterminate.

Port Pi register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
Pi (i = 0 to 7)	03E0$_{16}$, 03E1$_{16}$, 03E5$_{16}$, 03E8$_{16}$,	Indeterminate
	03E9$_{16}$, 03EC$_{16}$, 03ED$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
Pi_0	Port Pi$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : “L” level data 1 : “H” level data (i = 0 to 7 except 2)	○	○
Pi_1	Port Pi$_1$ register		○	○
Pi_2	Port Pi$_2$ register		○	○
Pi_3	Port Pi$_3$ register		○	○
Pi_4	Port Pi$_4$ register		○	○
Pi_5	Port Pi$_5$ register		○	○
Pi_6	Port Pi$_6$ register		○	○
Pi_7	Port Pi$_7$ register		○	○

Note: Nothing is assigned in direction register of P3$_6$, P3$_7$, P4$_6$, P4$_7$, P5$_5$ to p5$_7$, P7$_2$ to P7$_7$. This bit can either be set nor reset. When read, its content is indeterminate.

Figure 2.12.2. Programmable I/O ports-related registers (1)

Pull-up control register 0

Symbol	Address	When reset
PUR0	03FC$_{16}$	00$_{16}$

b7 b6 b5 b4 b3 b2 b1 b0	Bit symbol	Bit name	Function	R	W
b0	PU00	P0$_0$ to P0$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
b1	PU01	P0$_4$ to P0$_7$ pull-up		○	○
b2	PU02	P1$_0$ to P1$_3$ pull-up		○	○
b3	PU03	P1$_4$ to P1$_7$ pull-up		○	○
b4	—	—		○	○
b5	—	—		○	○
b6	PU06	P3$_0$ to P3$_3$ pull-up		○	○
b7	PU07	P3$_4$ to P3$_5$ pull-up		○	○

Pull-up control register 1

Symbol	Address	When reset
PUR1	03FD$_{16}$	00$_{16}$

b7 b6 b5 b4 b3 b2 b1 b0	Bit symbol	Bit name	Function	R	W
b0	PU10	P4$_0$ to P4$_3$ pull-up	The corresponding port is pulled high with a pull-up resistor 0 : Not pulled high 1 : Pulled high	○	○
b1	PU11	P4$_4$ to P4$_7$ pull-up		○	○
b2	PU12	P5$_0$ to P5$_3$ pull-up		○	○
b3	PU13	P5$_4$ pull-up		○	○
b4	PU14	P6$_0$ to P6$_3$ pull-up		○	○
b5	PU15	P6$_4$ to P6$_7$ pull-up		○	○
b6	PU16	P7$_0$ to P7$_1$ pull-up		○	○
b7	—	—		○	○

Port P1 drive capacity control register

Symbol	Address	When reset
DRR	03FE$_{16}$	00$_{16}$

b7 b6 b5 b4 b3 b2 b1 b0	Bit symbol	Bit name	Function	R	W
b0	DRR0	Port P1$_0$ drive capacity	Set P1 N-channel output transistor drive capacity 0 : LOW 1 : HIGH	○	○
b1	DRR1	Port P1$_1$ drive capacuty		○	○
b2	DRR2	Port P1$_2$ drive capacity		○	○
b3	DRR3	Port P1$_3$ drive capacity		○	○
b4	DRR4	Port P1$_4$ drive capacity		○	○
b5	DRR5	Port P1$_5$ drive capacity		○	○
b6	DRR6	Port P1$_6$ drive capacity		○	○
b7	DRR7	Port P1$_7$ drive capacity		○	○

Figure 2.12.3. Programmable I/O ports-related registers (2)

Chapter 3

Examples of Peripheral functions Applications

This chapter presents applications in which peripheral functions built in the M16C/20 are used. They are shown here as examples. In practical use, make suitable changes and perform sufficient evaluation. For basic use, see Chapter 2 How to Use Peripheral Functions.

Here follows the list of applications that appear in this chapter.

3.1 Long-Period Timers

Overview In this process, Timer X0 and Timer X1 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 X
- Event counter mode of timer X

Specifications

(1) Set timer X0 to timer mode, and set timer X1 to event counter mode.

(2) Perform a count on count source f_1 using timer X0 to count for 1 ms, and perform a count on timer X0 using timer X1 to count for 1 second.

(3) Connect a 10-MHz oscillator to X_{IN}.

Operation (1) Setting the count start flag to “1” causes the counter to begin counting. The counter of timer X0 performs a down count on count source f_1.

(2) If the counter of timer X0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer X0 interrupt request bit goes to “1”. The counter of timer X1 performs a down count on underflows in timer X0.

(3) If the counter of timer X1 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer X1 interrupt request bit goes to “1”.

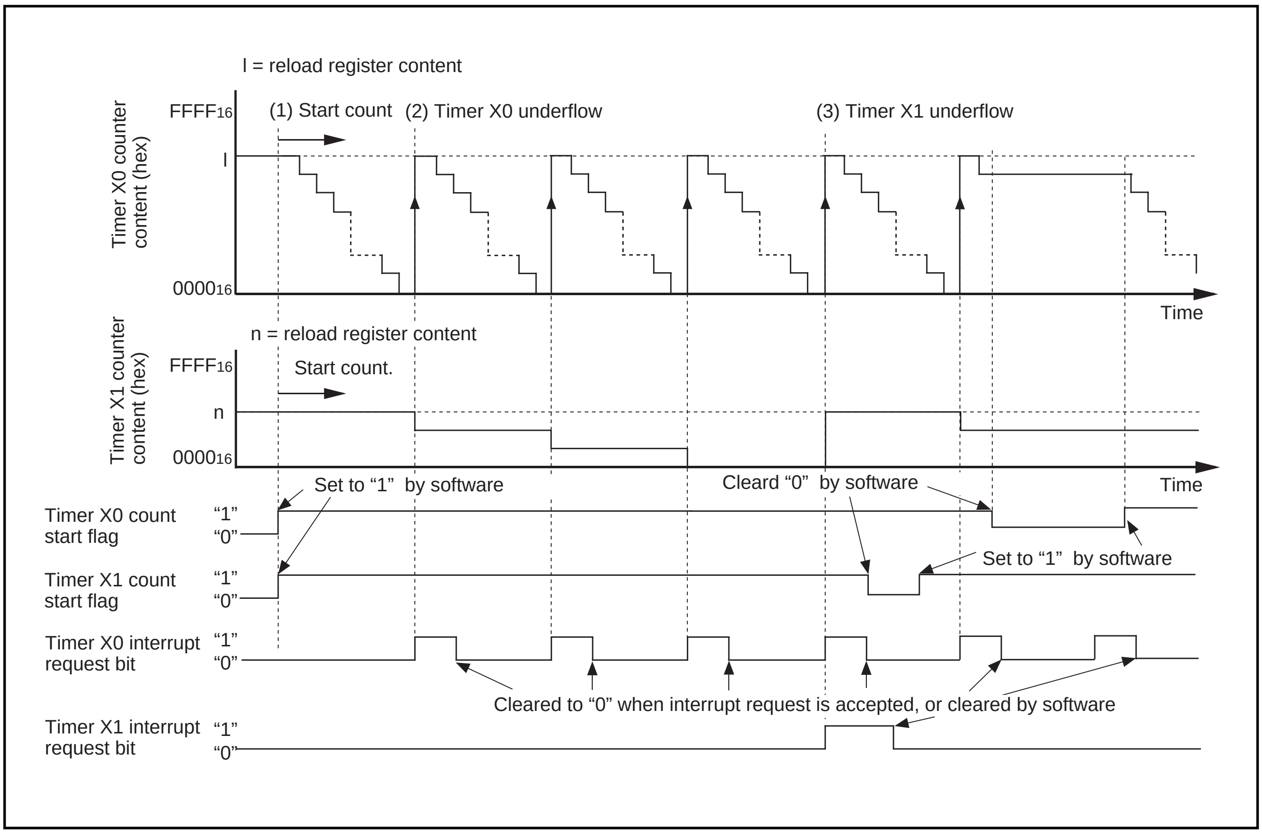

l = reload register content
Timer X0 counter content (hex)
FFFF16
l
000016
(1) Start count
(2) Timer X0 underflow
(3) Timer X1 underflow
Time
n = reload register content
Timer X1 counter content (hex)
FFFF16
Start count.
n
000016
Time
Set to “1” by software
Cleard “0” by software
Set to “1” by software
Timer X0 count start flag
Timer X1 count start flag
Timer X0 interrupt request bit
Timer X1 interrupt request bit
“1”
“0”
Cleared to “0” when interrupt request is accepted, or cleared by software

Figure 3.1.1. Operation timing of long-period timers

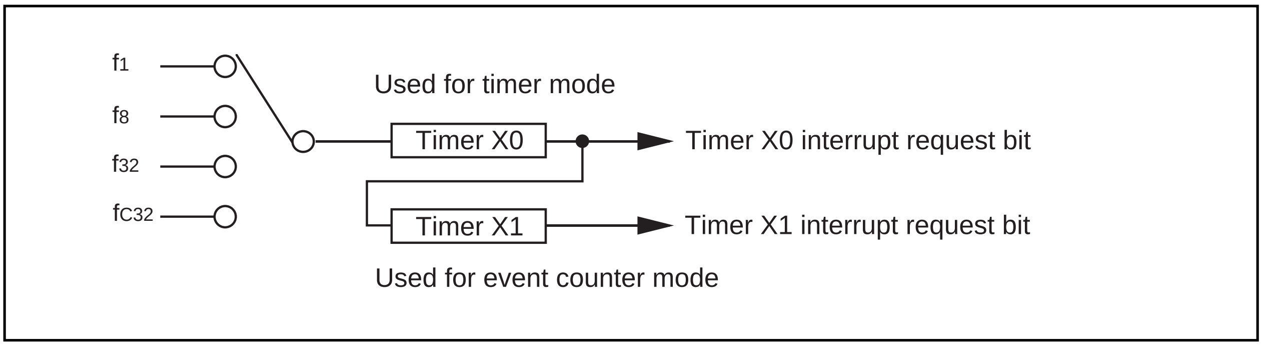
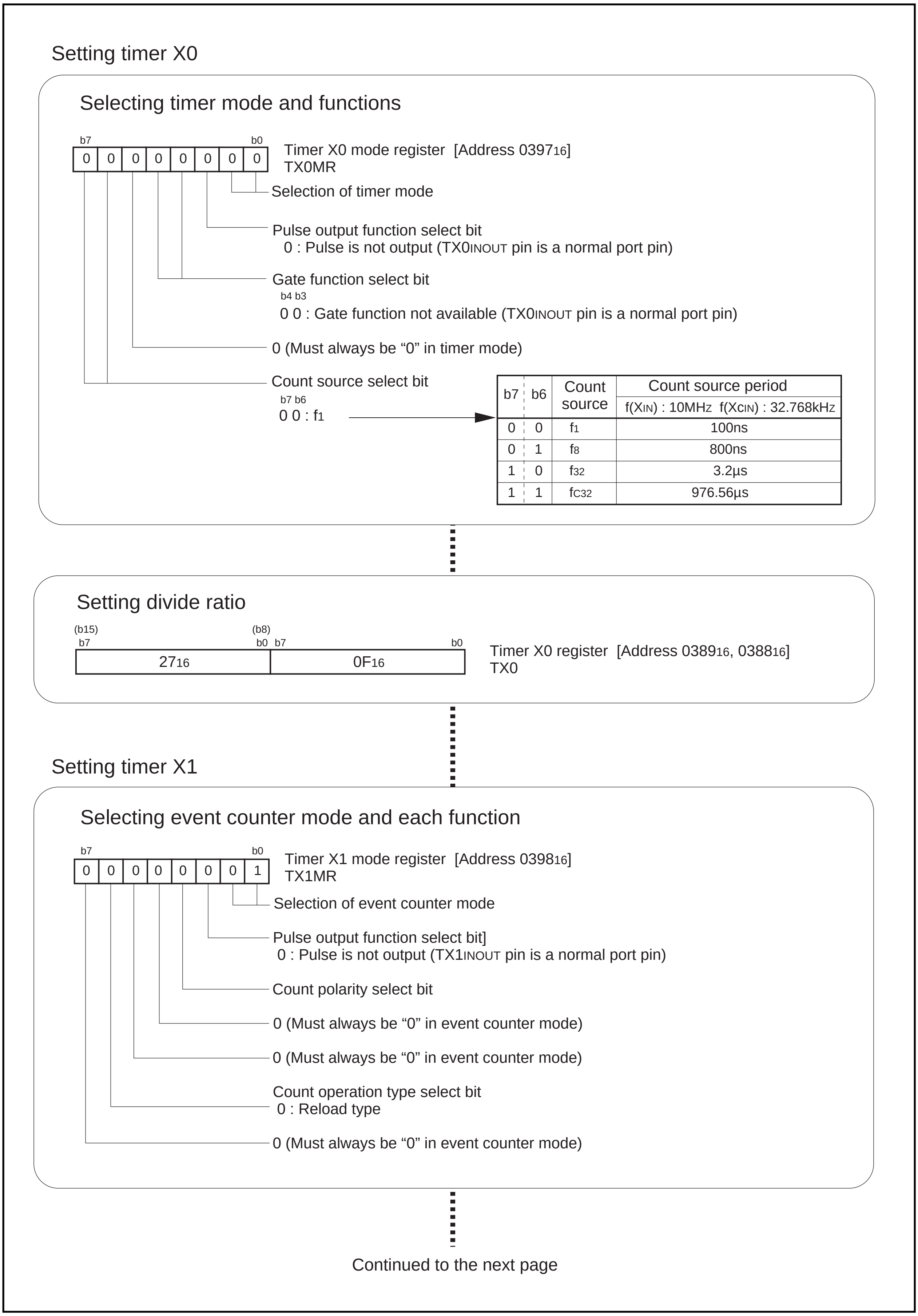

f1
f8
f32
fC32
Used for timer mode
Timer X0
Timer X0 interrupt request bit
Timer X1
Timer X1 interrupt request bit
Used for event counter mode

Figure 3.1.2. Connection diagram of long-period timers



Setting timer X0

Selecting timer mode and functions

Timer X0 mode register [Address 0397$_{16}$]
TX0MR

Selection of timer mode

Pulse output function select bit
0 : Pulse is not output (TX0$_{INOUT}$ pin is a normal port pin)

Gate function select bit
b4 b3
0 0 : Gate function not available (TX0$_{INOUT}$ pin is a normal port pin)

0 (Must always be “0” in timer mode)

Count source select bit
b7 b6
0 0 : f_1

b7	b6	Count source	Count source period
			f(X$_{IN}$) : 10MHz f(X$_{CIN}$) : 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Setting divide ratio

Timer X0 register [Address 0389$_{16}$, 0388$_{16}$]
TX0

Setting timer X1

Selecting event counter mode and each function

Timer X1 mode register [Address 0398$_{16}$]
TX1MR

Selection of event counter mode

Pulse output function select bit]
0 : Pulse is not output (TX1$_{INOUT}$ pin is a normal port pin)

Count polarity select bit

0 (Must always be “0” in event counter mode)

0 (Must always be “0” in event counter mode)

Count operation type select bit
0 : Reload type

0 (Must always be “0” in event counter mode)

Continued to the next page

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

Under development

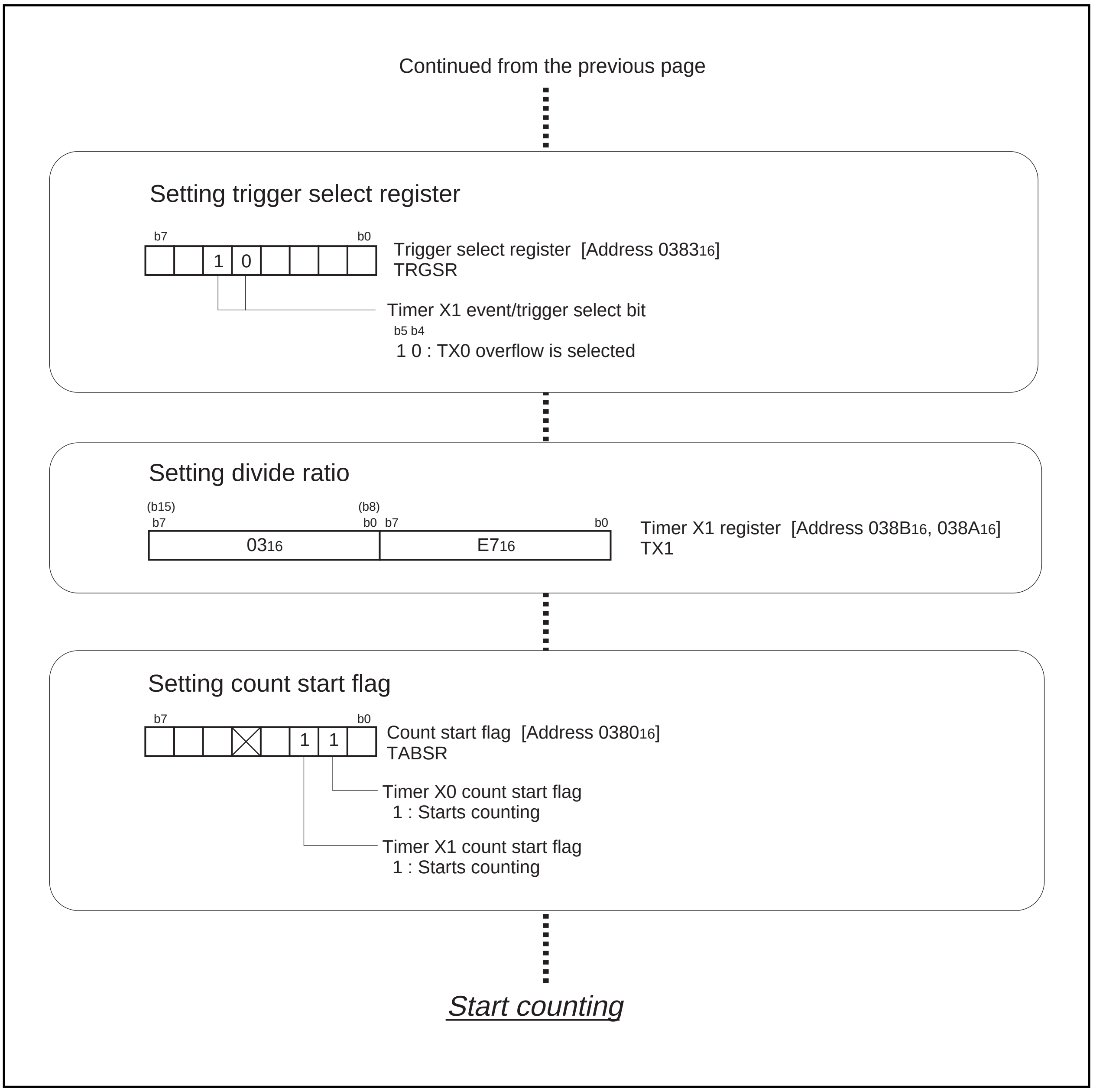

Continued from the previous page
Setting trigger select register
b7
b0
1
0
Trigger select register [Address 0383₁₆]
TRGSR
Timer X1 event/trigger select bit
b5 b4
1 0 : TX0 overflow is selected
Setting divide ratio
(b15)
b7
(b8)
b0
b7
b0
03₁₆
E7₁₆
Timer X1 register [Address 038B₁₆, 038A₁₆]
TX1
Setting count start flag
b7
b0
1
1
Count start flag [Address 0380₁₆]
TABSR
Timer X0 count start flag
1 : Starts counting
Timer X1 count start flag
1 : Starts counting
Start counting

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 X0 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 X
- One-shot timer mode of timer X

Specifications

(1) Set timer X0 in timer mode, and set timer X1 in one-shot timer mode with pulse-output function.

(2) Set 1 ms, the PWM period, to timer X0. Set 500 µs, the width of PWM “H” pulse, to timer X1. Both timer X0 and timer X1 use f_1 for the count source.

(3) Connect a 10-MHz oscillator to X_{IN}.

Operation

(1) Setting the count start flag to “1” causes the counter of timer X0 to begin counting. The counter of timer X0 performs a down count on count source f_1.

(2) If the counter of timer X0 underflows, the counter reloads the content of the reload register and continues counting. At this time, the timer X0 interrupt request bit gose to “1”.

(3) An underflow in timer X0 triggers the counter of timer X1 and causes it to begin counting. When the counter of timer X1 begins counting, the output level of the $TX1_{INOUT}$ pin gose to “H”.

(4) As soon as the count of the counter of timer X1 becomes “0000_{16}”, the output level of $TX1_{INOUT}$ pin gose to “L”, and the counter reloads the content of the reload register and stops counting. At the same time, the timer X1 interrupt request bit gose to “1”.

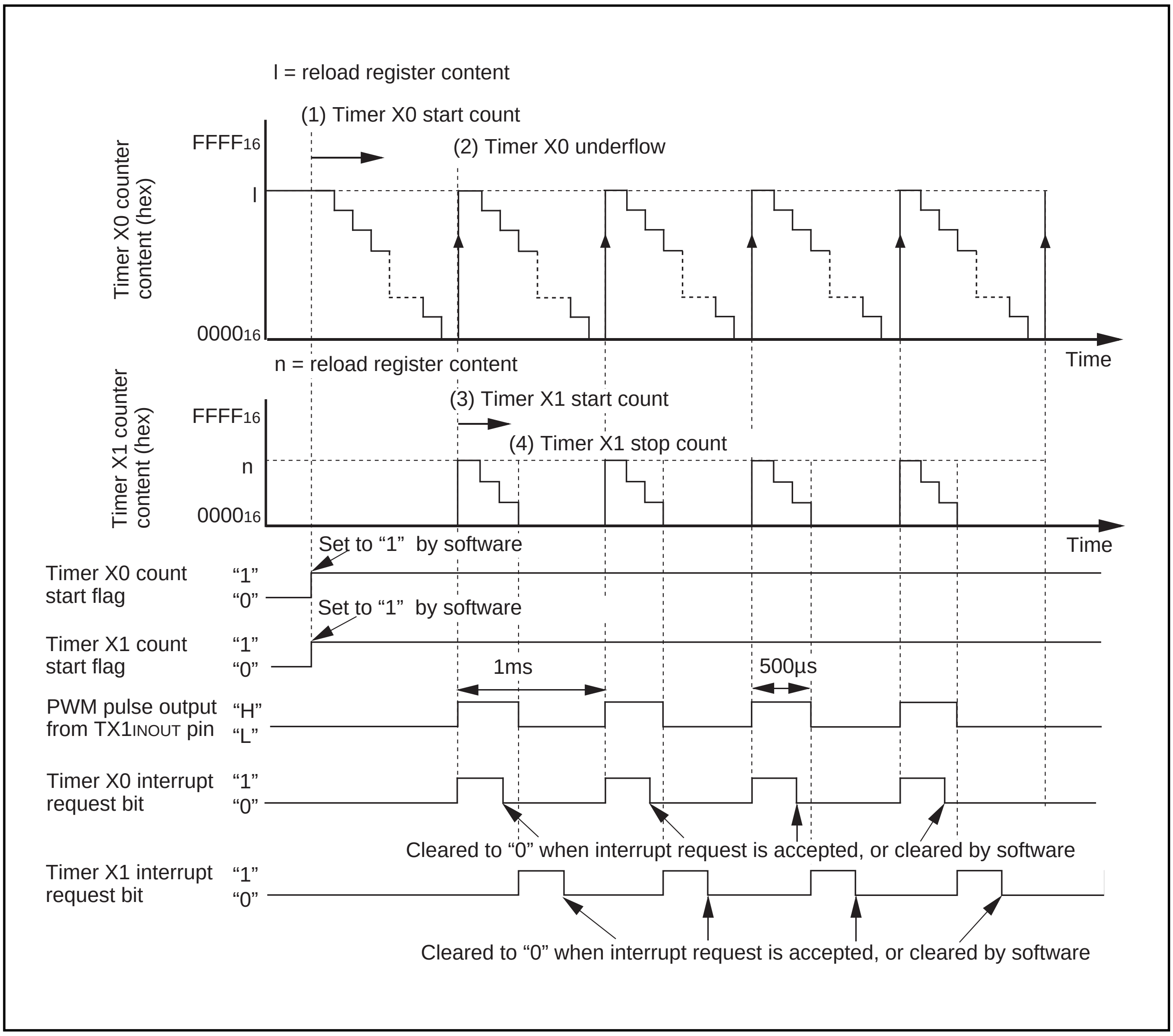

l = reload register content
(1) Timer X0 start count
(2) Timer X0 underflow
Timer X0 counter content (hex)
FFFF16
l
000016
Time
n = reload register content
(3) Timer X1 start count
(4) Timer X1 stop count
Timer X1 counter content (hex)
FFFF16
n
000016
Time
Set to "1" by software
Timer X0 count start flag
"1"
"0"
Set to "1" by software
Timer X1 count start flag
"1"
"0"
1ms
500µs
PWM pulse output from TX1INOUT pin
"H"
"L"
Timer X0 interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software
Timer X1 interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

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

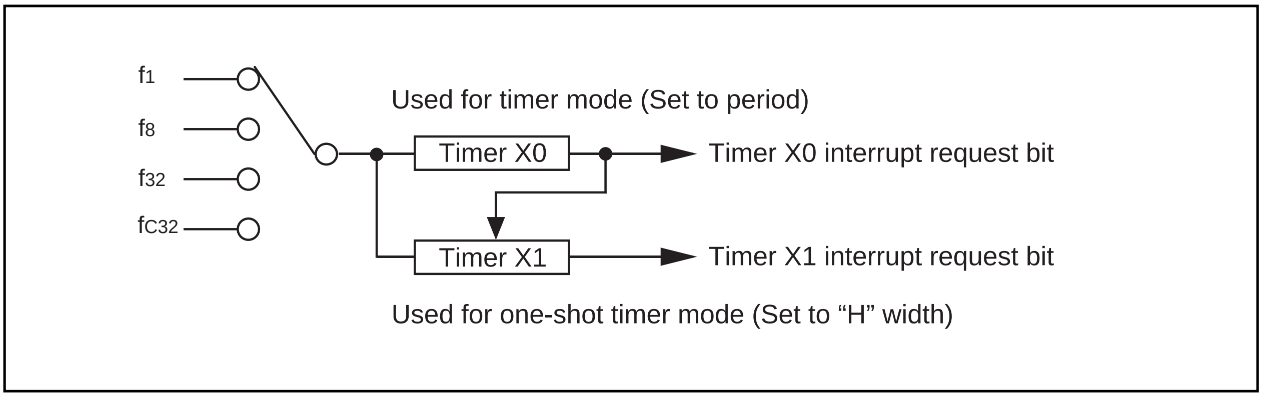
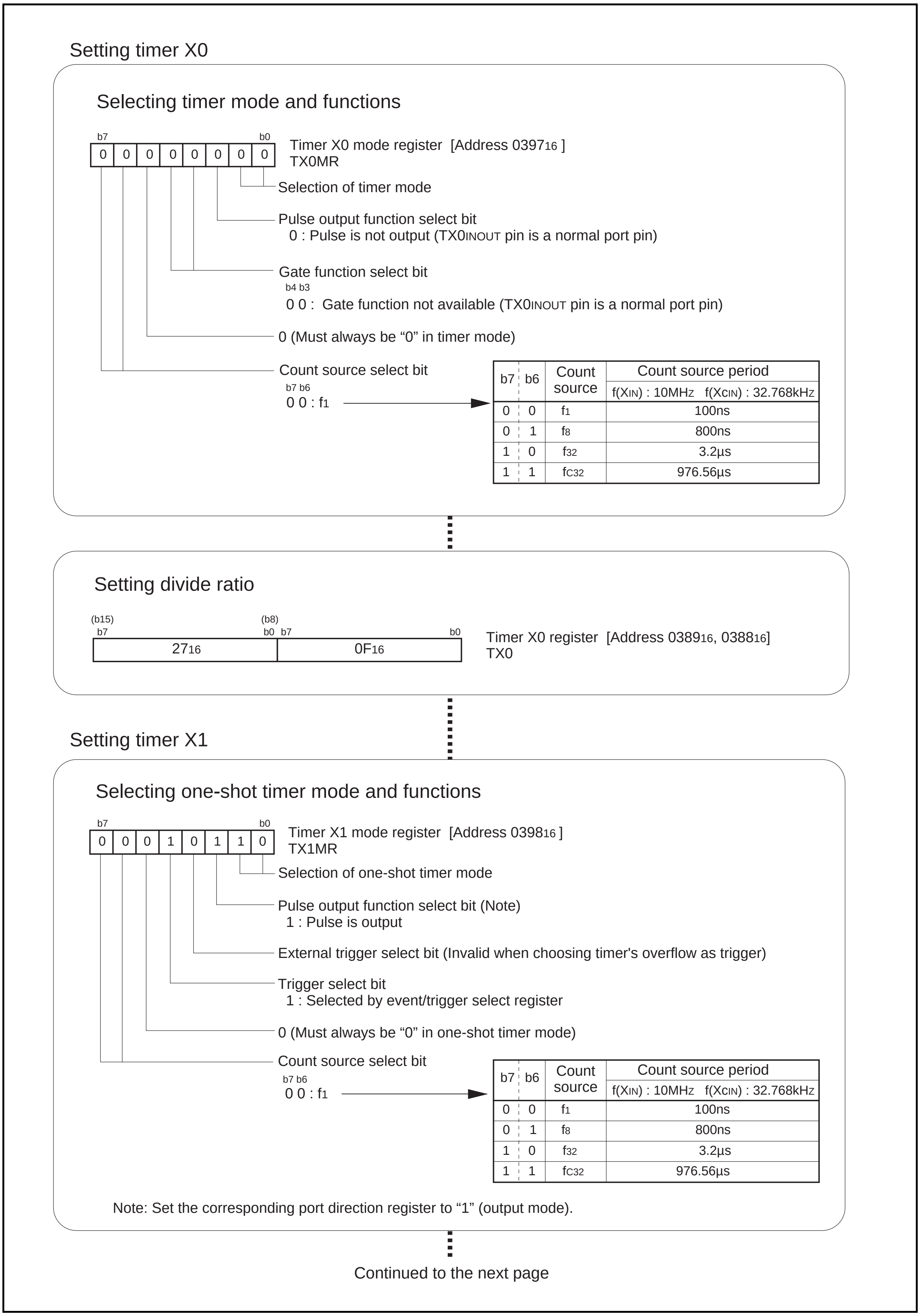

f1
f8
f32
fC32
Used for timer mode (Set to period)
Timer X0
Timer X0 interrupt request bit
Timer X1
Timer X1 interrupt request bit
Used for one-shot timer mode (Set to "H" width)

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



Setting timer X0

Selecting timer mode and functions

Timer X0 mode register [Address 0397_{16}]
TX0MR

Selection of timer mode

Pulse output function select bit
0 : Pulse is not output (TX0INOUT pin is a normal port pin)

Gate function select bit
b4 b3
0 0 : Gate function not available (TX0INOUT pin is a normal port pin)

0 (Must always be “0” in timer mode)

Count source select bit
b7 b6
0 0 : f_1

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Setting divide ratio

Timer X0 register [Address 0389_{16}, 0388_{16}]
TX0

Setting timer X1

Selecting one-shot timer mode and functions

Timer X1 mode register [Address 0398_{16}]
TX1MR

Selection of one-shot timer mode

Pulse output function select bit (Note)
1 : Pulse is output

External trigger select bit (Invalid when choosing timer's overflow as trigger)

Trigger select bit
1 : Selected by event/trigger select register

0 (Must always be “0” in one-shot timer mode)

Count source select bit
b7 b6
0 0 : f_1

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Note: Set the corresponding port direction register to “1” (output mode).

Continued to the next page

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

Under development

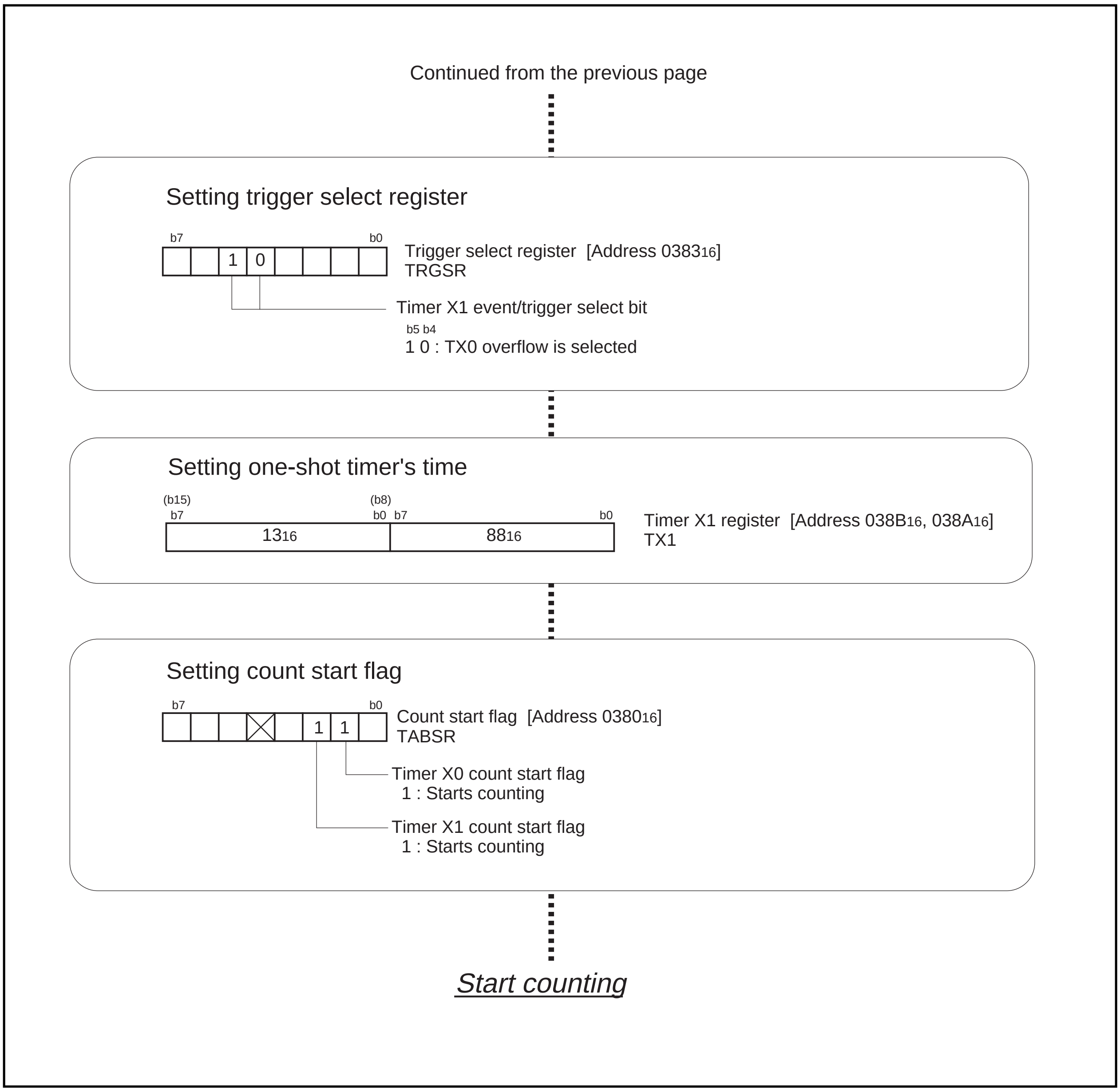

Continued from the previous page
Setting trigger select register
b7
b0
1
0
Trigger select register [Address 0383₁₆]
TRGSR
Timer X1 event/trigger select bit
b5 b4
1 0 : TX0 overflow is selected
Setting one-shot timer's time
(b15)
b7
(b8)
b0
b7
b0
13₁₆
88₁₆
Timer X1 register [Address 038B₁₆, 038A₁₆]
TX1
Setting count start flag
b7
b0
1
1
Count start flag [Address 0380₁₆]
TABSR
Timer X0 count start flag
1 : Starts counting
Timer X1 count start flag
1 : Starts counting
Start counting

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 X

Specifications

(1) Set timer X0 in one-shot timer mode, and set timer X1 in one-shot timer mode with pulse-output function.

(2) Set 1 ms, an interval before a pulse is output, in timer X0; and set 50 µs, a pulse width, in timer X1. Both timer X0 and timer X1 use f_1 for the count source.

(3) Connect a 10-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 X0 to count.

(2) If an effective edge, selected by use of the external trigger select bit, is input to the $TX0_{INOUT}$ pin, the counter begins a down count. The counter of timer X0 performs a down count on count source f_1.

(3) As soon as the counter of timer X0 becomes “0000_{16}”, the counter reloads the content of the reload register and stops counting. At this time, the timer X0 interrupt request bit gose to “1”.

(4) An underflow in timer X0 triggers the counter of timer X1 and causes it to begin counting. When timer X1 begins counting, the output level of the $TX1_{INOUT}$ pin gose to “H”.

(5) As soon as the counter of timer X1 becomes “0000_{16}”, the output level of the $TX1_{INOUT}$ pin gose to “L”, the counter reloads the content of the reload register, and stops counting. At this time, timer X1 interrupt request bit gose to “1”.

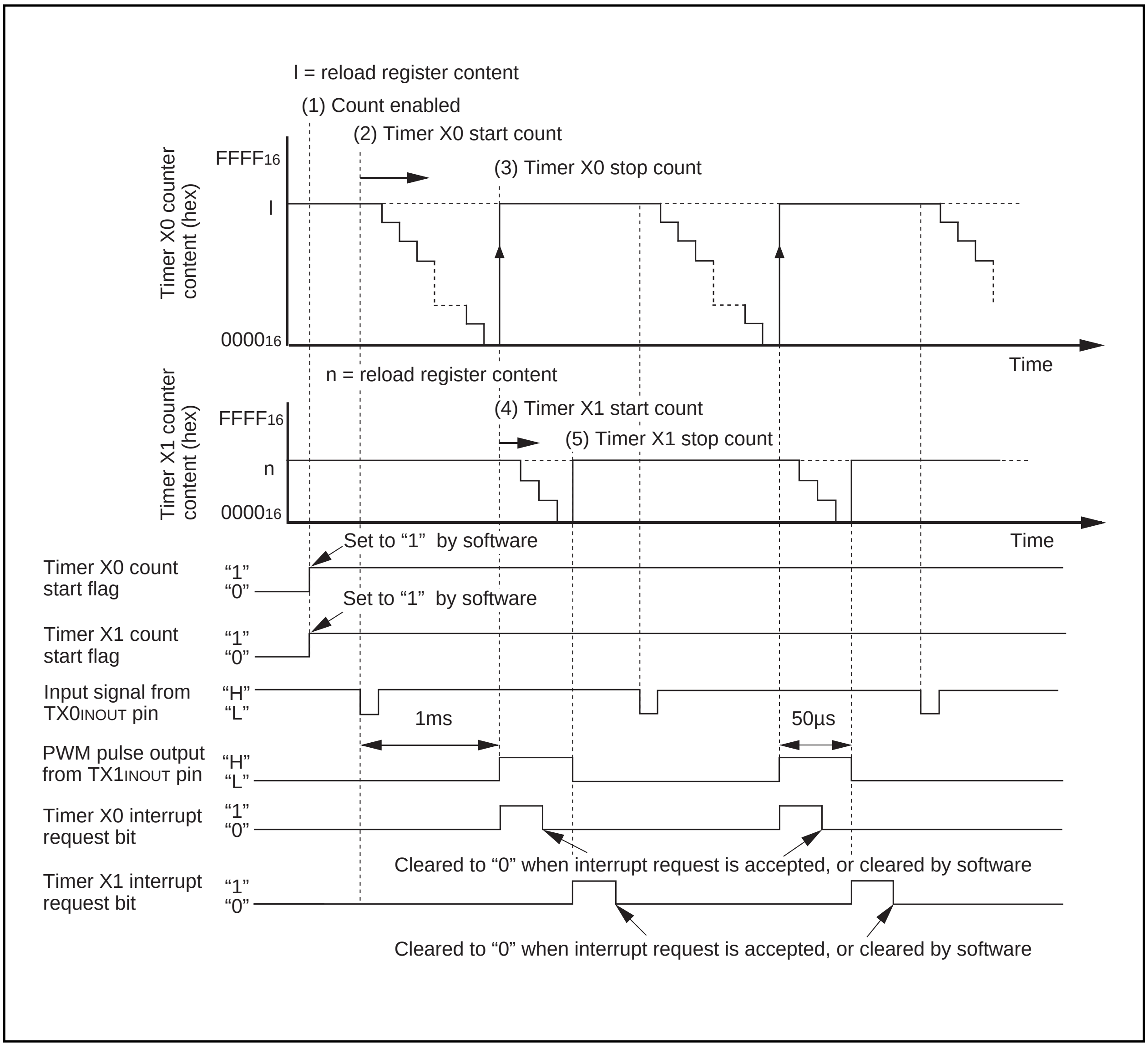

l = reload register content
(1) Count enabled
(2) Timer X0 start count
(3) Timer X0 stop count
Timer X0 counter content (hex)
FFFF16
l
000016
Time
n = reload register content
(4) Timer X1 start count
(5) Timer X1 stop count
Timer X1 counter content (hex)
FFFF16
n
000016
Time
Set to "1" by software
Timer X0 count start flag
"1"
"0"
Set to "1" by software
Timer X1 count start flag
"1"
"0"
Input signal from TX0INOUT pin
"H"
"L"
1ms
50µs
PWM pulse output from TX1INOUT pin
"H"
"L"
Timer X0 interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software
Timer X1 interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Figure 3.3.1. Operation timing of delayed one-shot output

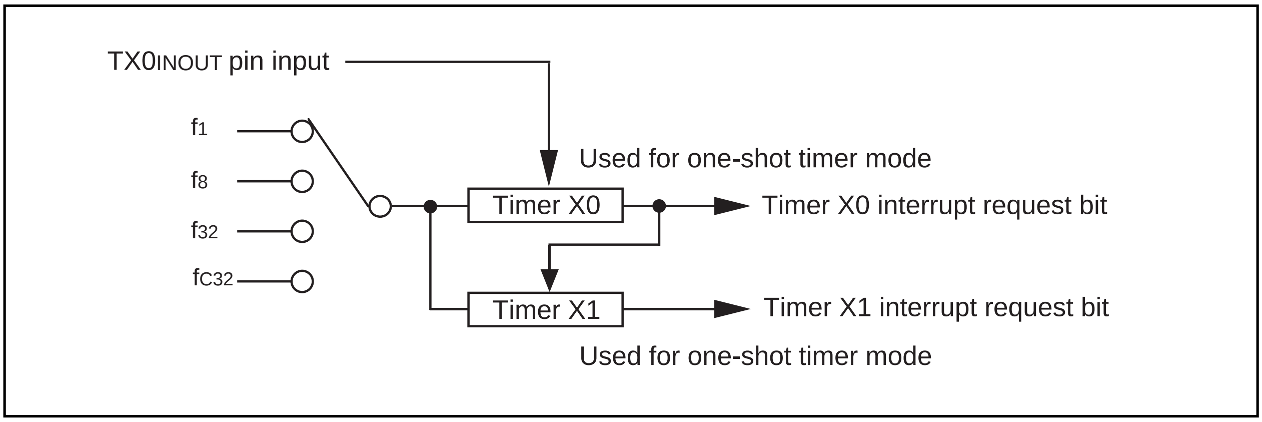
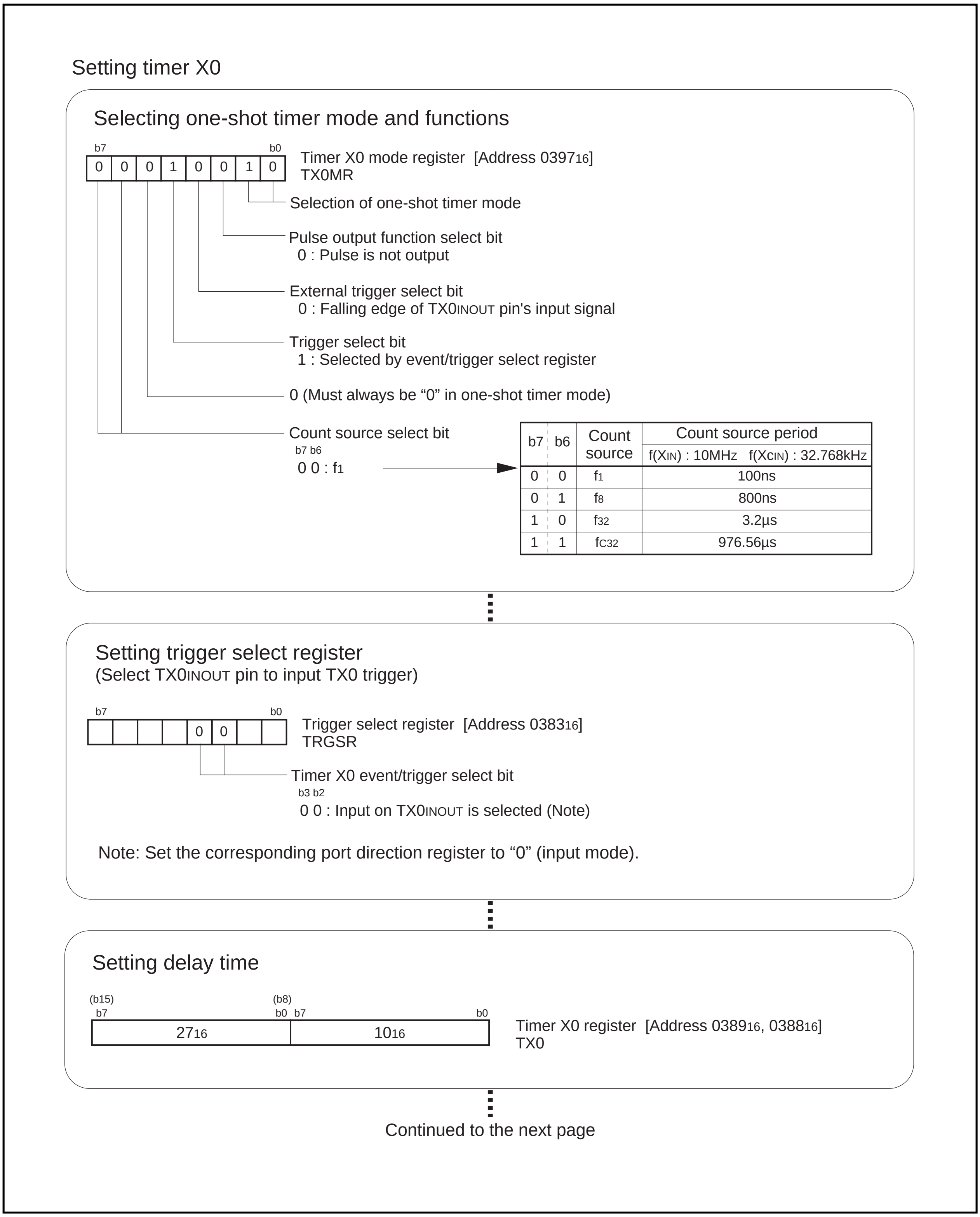

TX0INOUT pin input
f1
f8
f32
fC32
Used for one-shot timer mode
Timer X0
Timer X0 interrupt request bit
Timer X1
Timer X1 interrupt request bit
Used for one-shot timer mode

Figure 3.3.2. Connection diagram of delayed one-shot output



Setting timer X0

Selecting one-shot timer mode and functions

b7 b0
0 0 0 1 0 0 1 0
Timer X0 mode register [Address 0397_{16}]
TX0MR

Selection of one-shot timer mode

Pulse output function select bit
0 : Pulse is not output

External trigger select bit
0 : Falling edge of $\mathrm{TX0_{INOUT}}$ pin's input signal

Trigger select bit
1 : Selected by event/trigger select register

0 (Must always be “0” in one-shot timer mode)

Count source select bit
b7 b6
0 0 : f_1

b7	b6	Count source	Count source period
			$f(X_{IN})$: 10MHz $f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Setting trigger select register
(Select $\mathrm{TX0_{INOUT}}$ pin to input TX0 trigger)

b7 b0
0 0
Trigger select register [Address 0383_{16}]
TRGSR

Timer X0 event/trigger select bit
b3 b2
0 0 : Input on $\mathrm{TX0_{INOUT}}$ is selected (Note)

Note: Set the corresponding port direction register to “0” (input mode).

Setting delay time

(b15) (b8)
b7 b0 b7 b0
27_{16} 10_{16}
Timer X0 register [Address 0389_{16}, 0388_{16}]
TX0

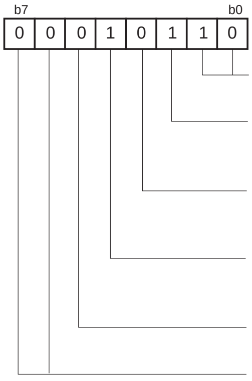
Continued to the next page

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

Continued from the previous page

Setting timer X1

Selecting one-shot timer mode and functions



Timer X1 mode register [Address 0398_{16}]
TX1MR

Selection of one-shot timer mode

Pulse output function select bit (Note)
1 : Pulse is output ($TX1_{INOUT}$ pin is pulse output pin)

External trigger select bit
Invalid when choosing timer's overflow

Trigger select bit
1 : Selected by event/trigger select register

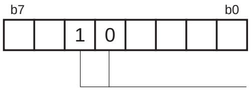
0 (Must always be “0” in one-shot timer mode)

Count source select bit
b7 b6
0 0 : f_1 →

b7	b6	Count source	Count source period
			$f(X_{IN})$: 10MHz $f(X_{CIN})$: 32.768kHz
0	0	f_1	100ns
0	1	f_8	800ns
1	0	f_{32}	3.2µs
1	1	f_{C32}	976.56µs

Note: Set the corresponding port direction register to “1” (output mode).

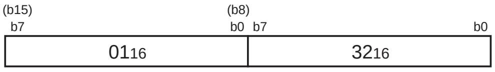
Setting trigger select register
(Set timer X0 to trigger timer X1)



Trigger select register [Address 0383_{16}]
TRGSR

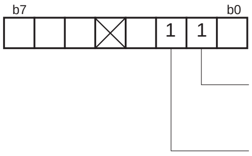
Timer X1 event/trigger select bit
b5 b4
1 0 : TX0 overflow is selected

Setting one-shot timer's time



Timer X1 register [Address $038B_{16}$, $038A_{16}$]
TX1

Setting count start flag



Count start flag [Address 0380_{16}]
TABSR

Timer X0 count start flag
1 : Starts counting

Timer X1 count start flag
1 : Starts counting

Start counting

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 X.

Specifications

(1) Sound a 2-kHz buzz beep by use of timer X0.

(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 10-MHz oscillator to XIN.

Operation (1) The microcomputer begins performing a count on timer X0. Timer X0 has disabled interrupts.

(2) $P4_3$ is $TX0_{INOUT}$ pin. Setting the port $P4_3$ direction register to “1” (output mode) and outputs 2-kHz pulses.

(3) The microcomputer stops outputting pulses by setting the port $P4_3$ direction register to “0” (input mode). $P4_3$ goes to an input pin, and the output from the pin becomes high-impedance.

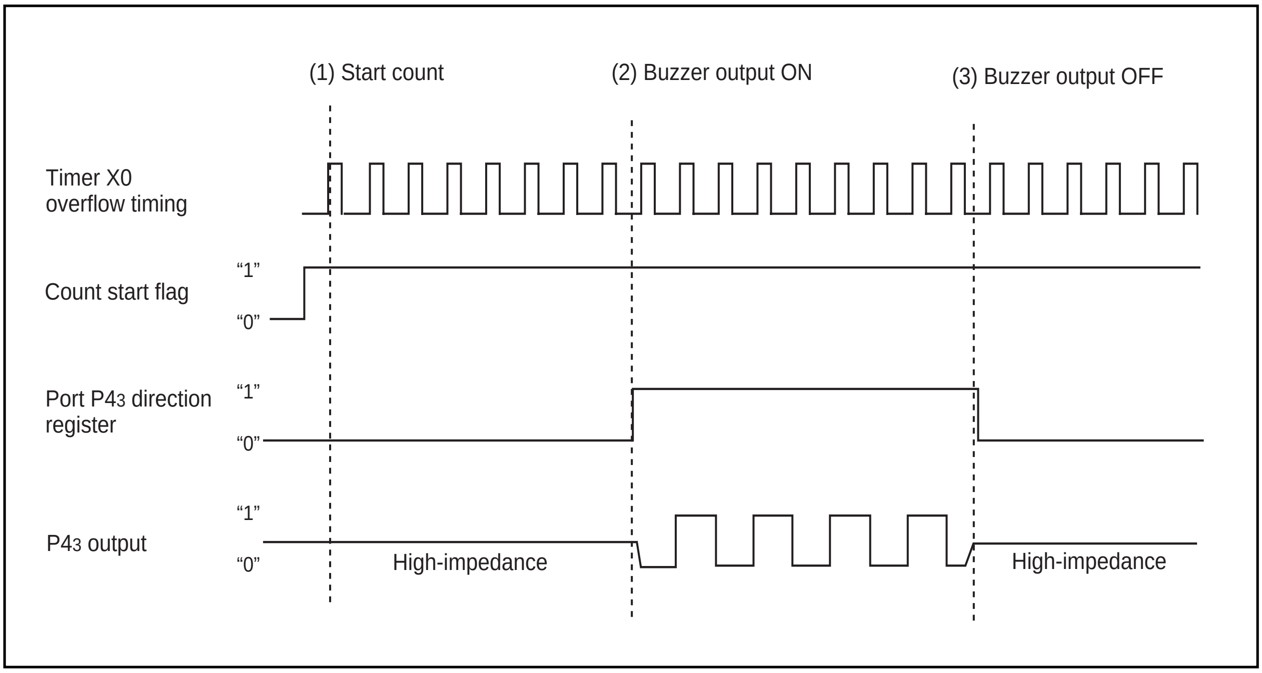
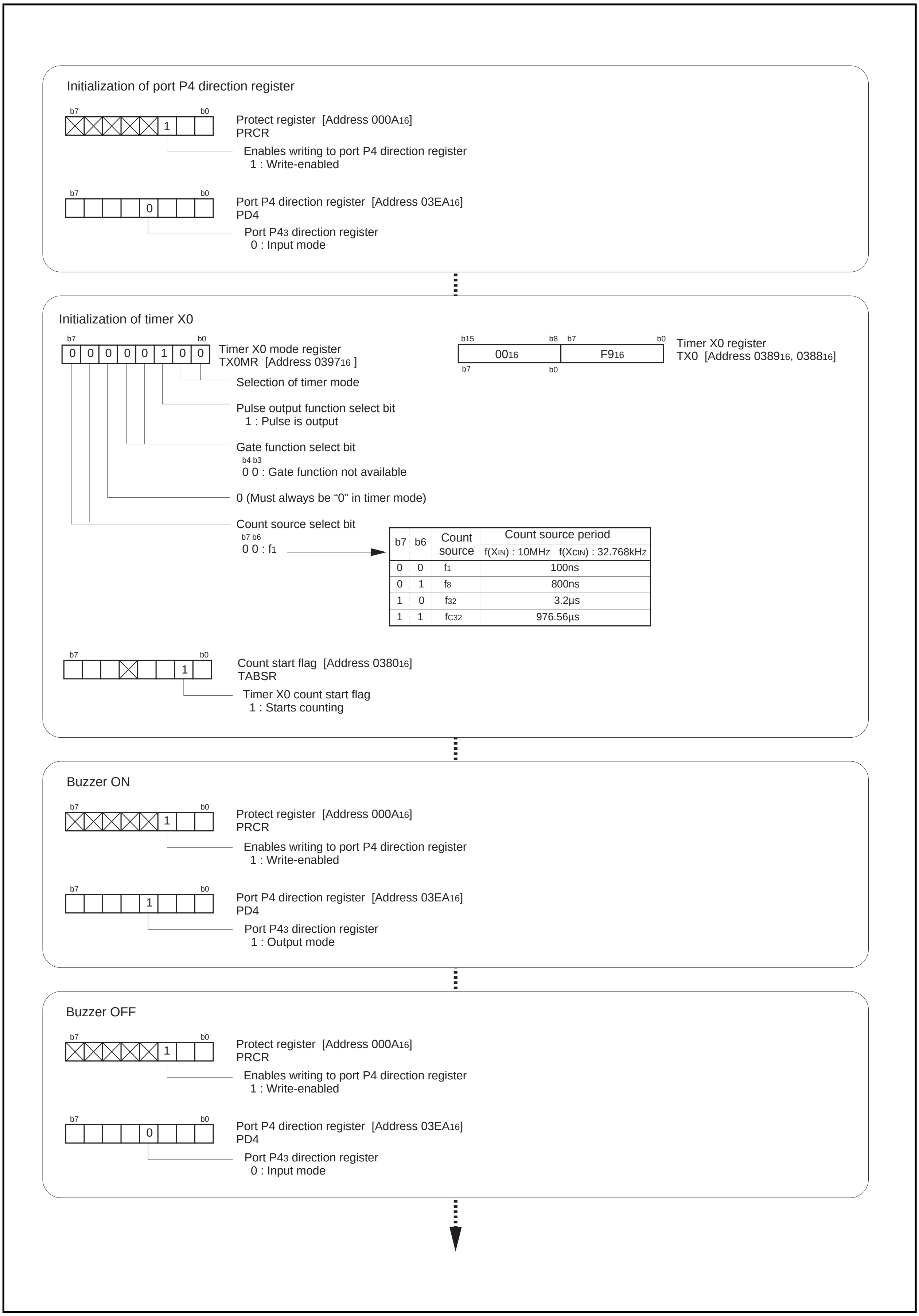


Figure 3.4.1. Operation timing of buzzer output

Under development


Initialization of port P4 direction register
b7 b0
1
Protect register [Address 000A16]
PRCR
Enables writing to port P4 direction register
1 : Write-enabled
b7 b0
0
Port P4 direction register [Address 03EA16]
PD4
Port P43 direction register
0 : Input mode
Initialization of timer X0
b7 b0
0 0 0 0 0 1 0 0
Timer X0 mode register
TX0MR [Address 039716]
Selection of timer mode
Pulse output function select bit
1 : Pulse is output
Gate function select bit
b4 b3
0 0 : Gate function not available
0 (Must always be "0" in timer mode)
Count source select bit
b7 b6
0 0 : f1
b15 b8 b7 b0
0016
F916
b7 b0
Timer X0 register
TX0 [Address 038916, 038816]
b7 b0
1
Count start flag [Address 038016]
TABSR
Timer X0 count start flag
1 : Starts counting
Buzzer ON
b7 b0
1
Protect register [Address 000A16]
PRCR
Enables writing to port P4 direction register
1 : Write-enabled
b7 b0
1
Port P4 direction register [Address 03EA16]
PD4
Port P43 direction register
1 : Output mode
Buzzer OFF
b7 b0
1
Protect register [Address 000A16]
PRCR
Enables writing to port P4 direction register
1 : Write-enabled
b7 b0
0
Port P4 direction register [Address 03EA16]
PD4
Port P43 direction register
0 : Input mode

b7	b6	Count source	Count source period
			f(XIN) : 10MHz f(XCIN) : 32.768kHz
0	0	f1	100ns
0	1	f8	800ns
1	0	f32	3.2µs
1	1	fC32	976.56µs

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 X

Specifications

(1) Inputting a falling edge to the $TX0_{INOUT}$ pin generates a timer X0 interrupt.

Operation (1) Set timer X0 to event counter mode, set timer to “0”, and set interrupt priority levels in timer X0.

(2) Inputting a falling edge to the $TX0_{INOUT}$ pin generates a timer X0 interrupt.

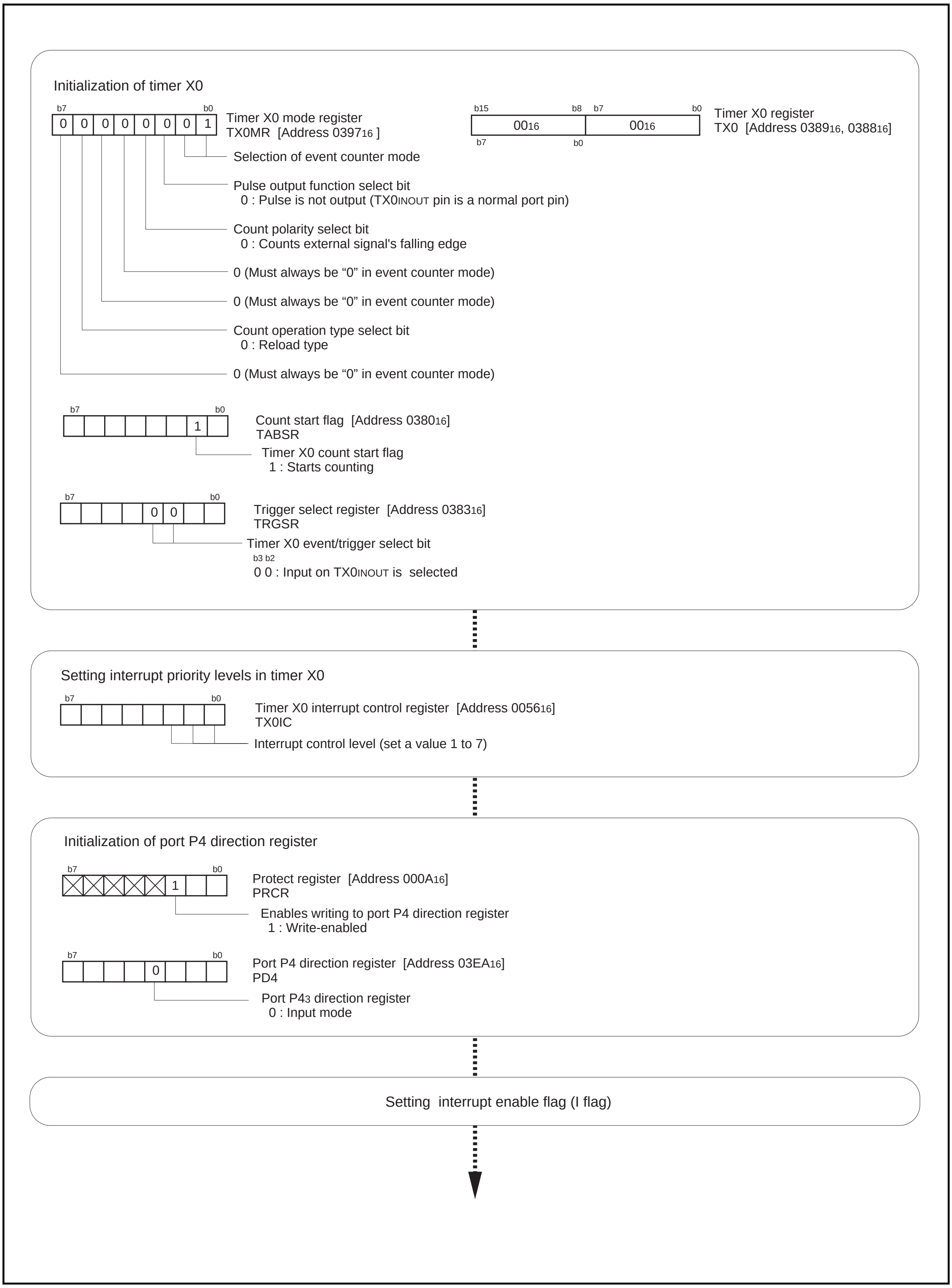

Initialization of timer X0
b7 b0
0 0 0 0 0 0 0 1
Timer X0 mode register
TX0MR [Address 0397₁₆]
Selection of event counter mode
Pulse output function select bit
0 : Pulse is not output (TX0INOUT pin is a normal port pin)
Count polarity select bit
0 : Counts external signal's falling edge
0 (Must always be “0” in event counter mode)
0 (Must always be “0” in event counter mode)
Count operation type select bit
0 : Reload type
0 (Must always be “0” in event counter mode)
b15 b8 b7 b0
00₁₆ 00₁₆
b7 b0
Timer X0 register
TX0 [Address 0389₁₆, 0388₁₆]
b7 b0
1
Count start flag [Address 0380₁₆]
TABSR
Timer X0 count start flag
1 : Starts counting
b7 b0
0 0
Trigger select register [Address 0383₁₆]
TRGSR
Timer X0 event/trigger select bit
b3 b2
0 0 : Input on TX0INOUT is selected
Setting interrupt priority levels in timer X0
b7 b0
Timer X0 interrupt control register [Address 0056₁₆]
TX0IC
Interrupt control level (set a value 1 to 7)
Initialization of port P4 direction register
b7 b0
1
Protect register [Address 000A₁₆]
PRCR
Enables writing to port P4 direction register
1 : Write-enabled
b7 b0
0
Port P4 direction register [Address 03EA₁₆]
PD4
Port P4₃ direction register
0 : Input mode
Setting interrupt enable flag (I flag)

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

Under development

3.6 Controlling Power Using Stop Mode

Overview The following are steps for controlling power using stop mode. Figure 3.6.1 shows the operation timing, Figure 3.6.2 shows an example of circuit, and Figures 3.6.3 and 3.6.4 show the set-up procedure.

Use the following peripheral functions:

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

Specifications

(1) Use $P3_0$ through $P3_3$ for the scan output pins of a key matrix. Use the input pins ($\overline{KI_0}$ through $\overline{KI_7}$) 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 $\overline{KI_7}$. Change the output of $P3_0$ through $P3_3$ to “L” and enter stop mode.

(2) If a key is pressed, “L” is input to one of pins $\overline{KI_0}$ through $\overline{KI_7}$ to clear stop mode. A key-input interrupt occurs to execute the key-input interrupt handling routine.

(3) Sequentially set $P3_0$ through $P3_3$ to “L” to determine which key was pressed.

(4) When the process to determine the key pressed is completed, change the output from $P3_0$ through $P3_3$ to “L” again and enter stop mode.

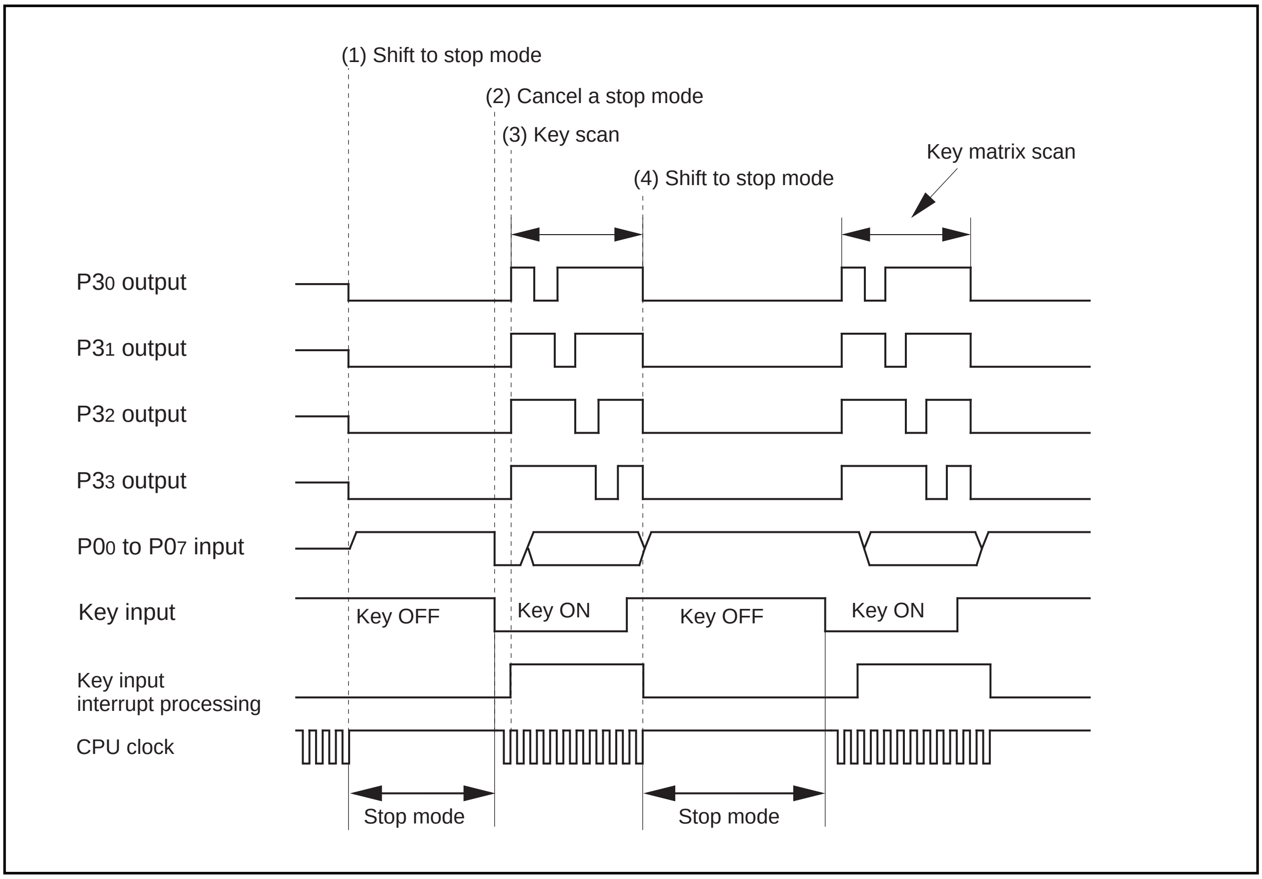


Figure 3.6.1. Operation timing of controlling power using stop mode

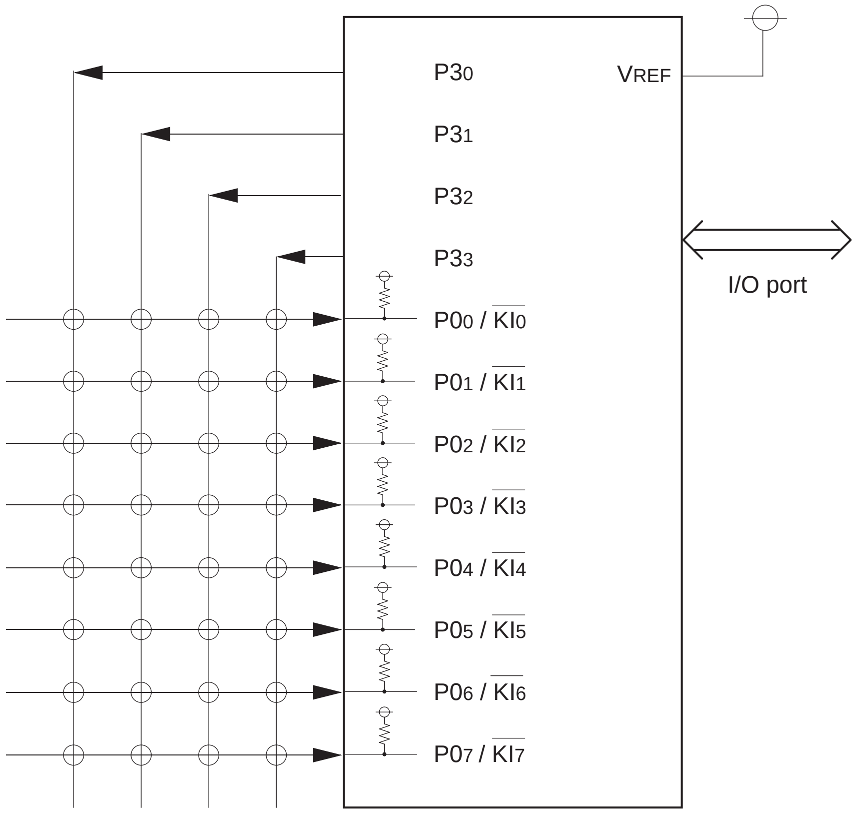
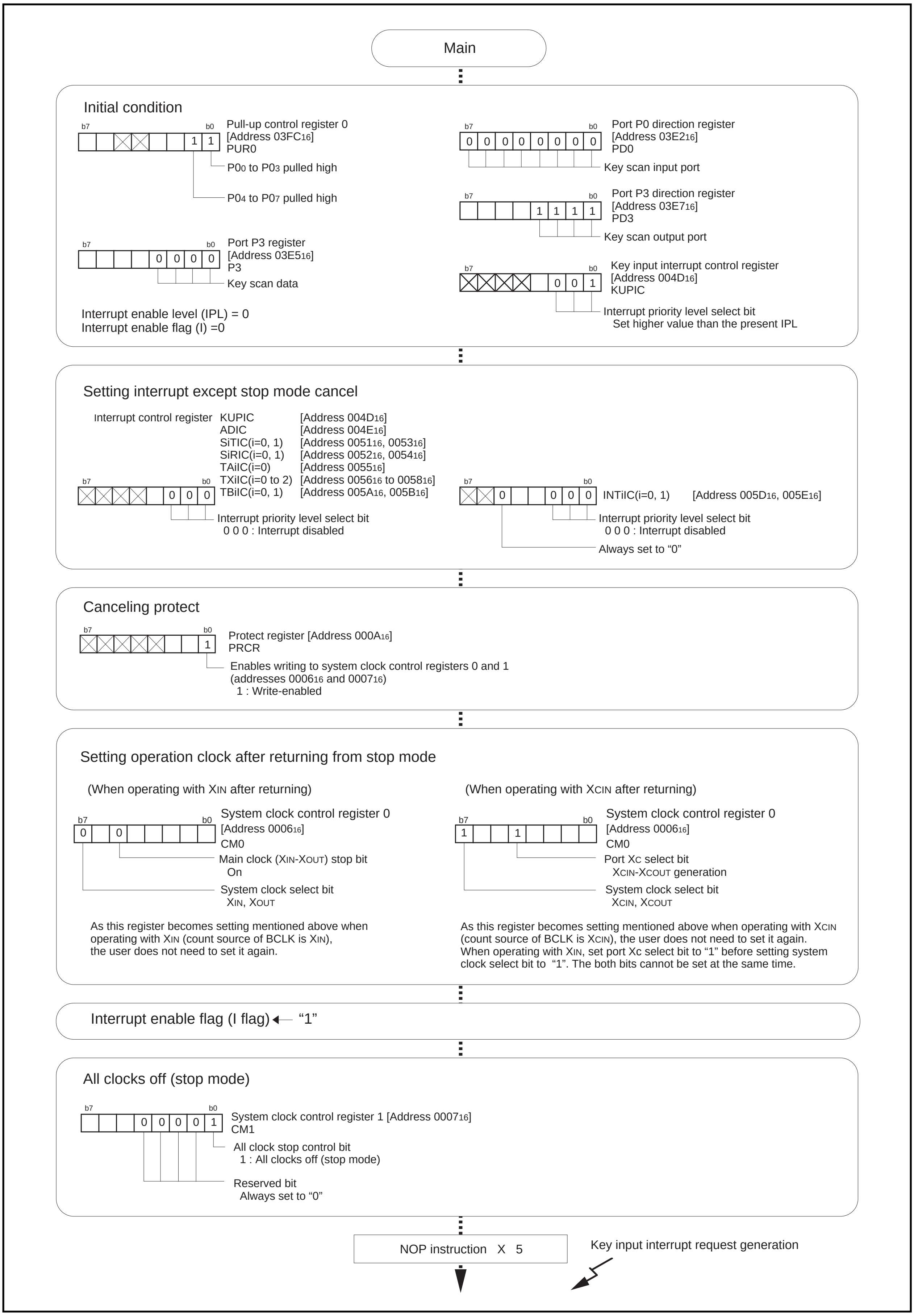

P3$_0$
P3$_1$
P3$_2$
P3$_3$
VREF
I/O port
P0$_0$ / $\overline{KI_0}$
P0$_1$ / $\overline{KI_1}$
P0$_2$ / $\overline{KI_2}$
P0$_3$ / $\overline{KI_3}$
P0$_4$ / $\overline{KI_4}$
P0$_5$ / $\overline{KI_5}$
P0$_6$ / $\overline{KI_6}$
P0$_7$ / $\overline{KI_7}$

Figure 3.6.2. Example of circuit of controling power using stop mode

Under development


Main
Initial condition
b7 b0
1 1
Pull-up control register 0 [Address 03FC16] PUR0
P00 to P03 pulled high
P04 to P07 pulled high
0 0 0 0
Port P3 register [Address 03E516] P3
Key scan data
Interrupt enable level (IPL) = 0
Interrupt enable flag (I) =0
0 0 0 0 0 0 0 0
Port P0 direction register [Address 03E216] PD0
Key scan input port
1 1 1 1
Port P3 direction register [Address 03E716] PD3
Key scan output port
0 0 1
Key input interrupt control register [Address 004D16] KUPIC
Interrupt priority level select bit
Set higher value than the present IPL
Setting interrupt except stop mode cancel
Interrupt control register KUPIC [Address 004D16]
ADIC [Address 004E16]
SiTIC(i=0, 1) [Address 005116, 005316]
SiRIC(i=0, 1) [Address 005216, 005416]
TAiIC(i=0) [Address 005516]
TXiIC(i=0 to 2) [Address 005616 to 005816]
TBiIC(i=0, 1) [Address 005A16, 005B16]
0 0 0
Interrupt priority level select bit
0 0 0 : Interrupt disabled
0 0 0 0
INTiIC(i=0, 1) [Address 005D16, 005E16]
Interrupt priority level select bit
0 0 0 : Interrupt disabled
Always set to “0”
Canceling protect
1
Protect register [Address 000A16] PRCR
Enables writing to system clock control registers 0 and 1 (addresses 000616 and 000716)
1 : Write-enabled
Setting operation clock after returning from stop mode
(When operating with XIN after returning)
0 0
System clock control register 0 [Address 000616] CM0
Main clock (XIN-XOUT) stop bit
On
System clock select bit
XIN, XOUT
(When operating with XCIN after returning)
1 1
System clock control register 0 [Address 000616] CM0
Port XC select bit
XCIN-XCOUT generation
System clock select bit
XCIN, XCOUT
Interrupt enable flag (I flag) ← “1”
All clocks off (stop mode)
0 0 0 0 1
System clock control register 1 [Address 000716] CM1
All clock stop control bit
1 : All clocks off (stop mode)
Reserved bit
Always set to “0”
NOP instruction X 5
Key input interrupt request generation

As this register becomes setting mentioned above when operating with XIN (count source of BCLK is XIN), the user does not need to set it again.

As this register becomes setting mentioned above when operating with XCIN (count source of BCLK is XCIN), the user does not need to set it again. When operating with XIN, set port Xc select bit to “1” before setting system clock select bit to “1”. The both bits cannot be set at the same time.

Figure 3.6.3. Set-up procedure of controlling power using stop mode (1)

Under development

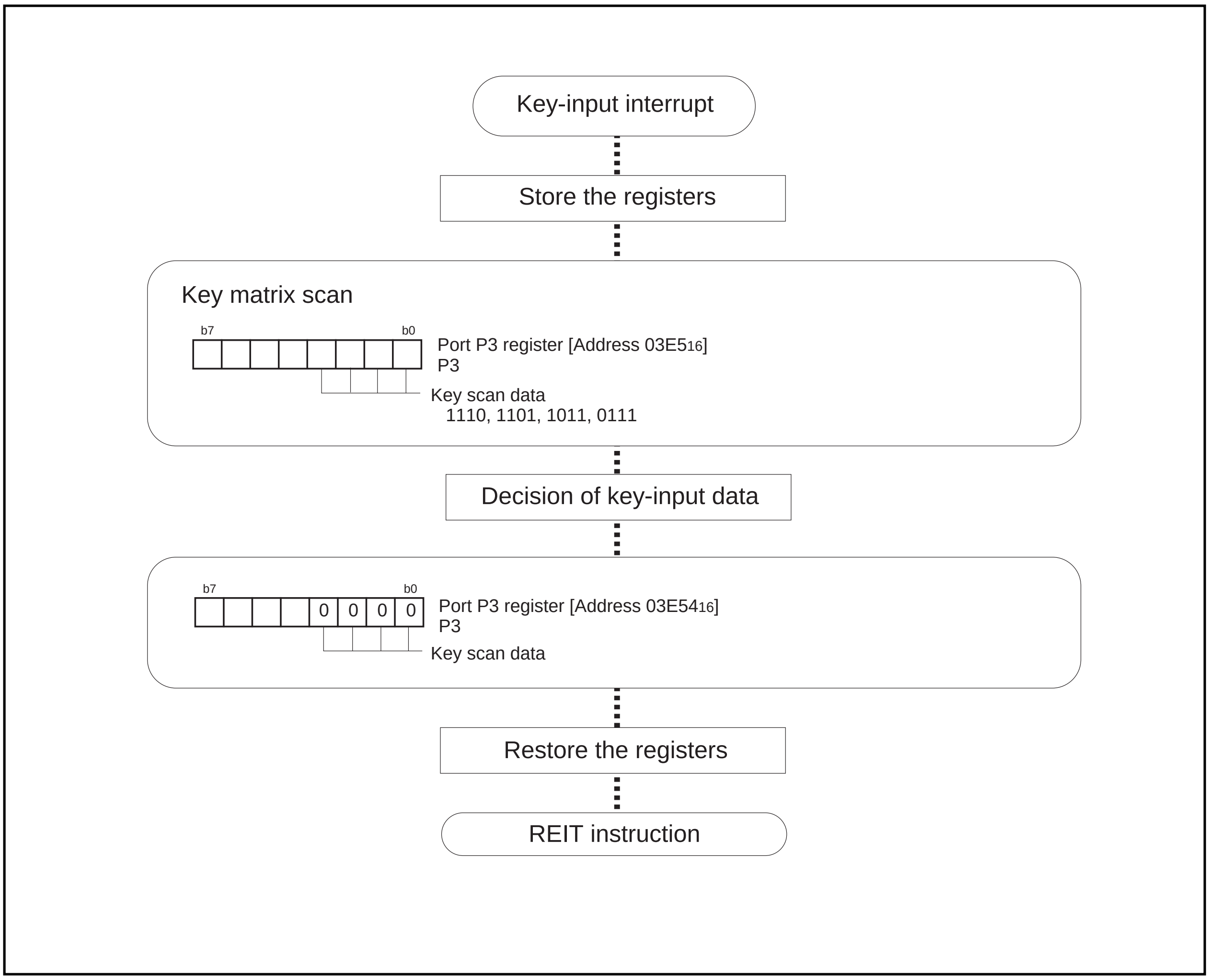


Figure 3.6.4. Set-up procedure of controlling power using stop mode (2)

Under development

3.7 Controling Power Using Wait Mode

Overview The following are steps for controling power using wait mode. Figure 3.7.1 shows the operation timing, and Figures 3.7.2 to 3.7.4 show the set-up procedure.

Use the following peripheral functions:

- Timer mode of timer B
- 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 B0 interrupt and the $\overline{\text{INT0}}$ interrupt.

(3) When a timer B0 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 B0 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.

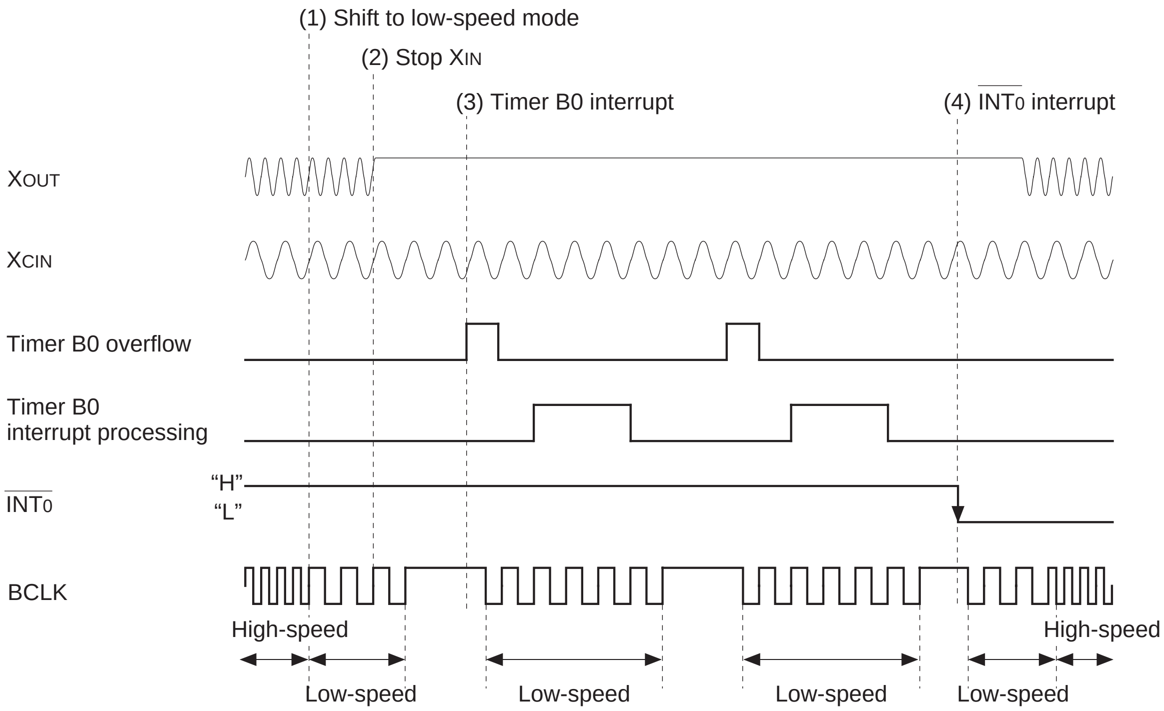
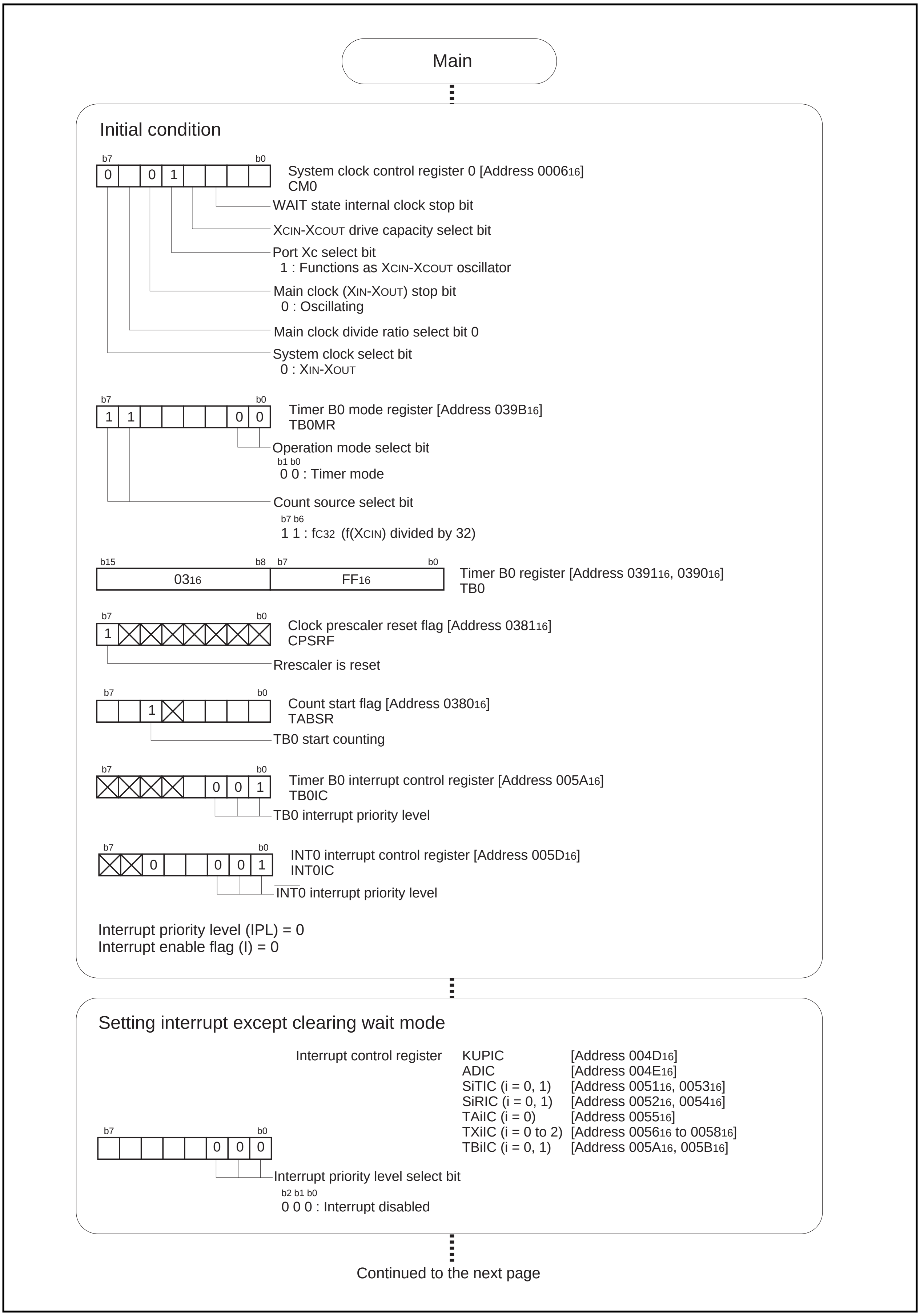


Figure 3.7.1. Operation timing of controling power using wait mode


Main
Initial condition
b7 b0
0 0 1
System clock control register 0 [Address 0006₁₆]
CM0
WAIT state internal clock stop bit
XCIN-XCOUT drive capacity select bit
Port Xc select bit
1 : Functions as XCIN-XCOUT oscillator
Main clock (XIN-XOUT) stop bit
0 : Oscillating
Main clock divide ratio select bit 0
System clock select bit
0 : XIN-XOUT
b7 b0
1 1 0 0
Timer B0 mode register [Address 039B₁₆]
TB0MR
Operation mode select bit
b1 b0
0 0 : Timer mode
Count source select bit
b7 b6
1 1 : fC32 (f(XCIN) divided by 32)
b15 b8 b7 b0
03₁₆
FF₁₆
Timer B0 register [Address 0391₁₆, 0390₁₆]
TB0
b7 b0
1
Clock prescaler reset flag [Address 0381₁₆]
CPSRF
Rrescaler is reset
b7 b0
1
Count start flag [Address 0380₁₆]
TABSR
TB0 start counting
b7 b0
0 0 1
Timer B0 interrupt control register [Address 005A₁₆]
TB0IC
TB0 interrupt priority level
b7 b0
0 0 0 1
INT0 interrupt control register [Address 005D₁₆]
INT0IC
INT0 interrupt priority level
Interrupt priority level (IPL) = 0
Interrupt enable flag (I) = 0
Setting interrupt except clearing wait mode
b7 b0
0 0 0
Interrupt priority level select bit
b2 b1 b0
0 0 0 : Interrupt disabled
Continued to the next page

Interrupt control register		
	KUPIC	[Address 004D₁₆]
	ADIC	[Address 004E₁₆]
	SiTIC (i = 0, 1)	[Address 0051₁₆, 0053₁₆]
	SiRIC (i = 0, 1)	[Address 0052₁₆, 0054₁₆]
	TAiIC (i = 0)	[Address 0055₁₆]
	TXiIC (i = 0 to 2)	[Address 0056₁₆ to 0058₁₆]
	TBiIC (i = 0, 1)	[Address 005A₁₆, 005B₁₆]

Figure 3.7.2. Set-up procedure of controlling power using wait mode (1)

Under development

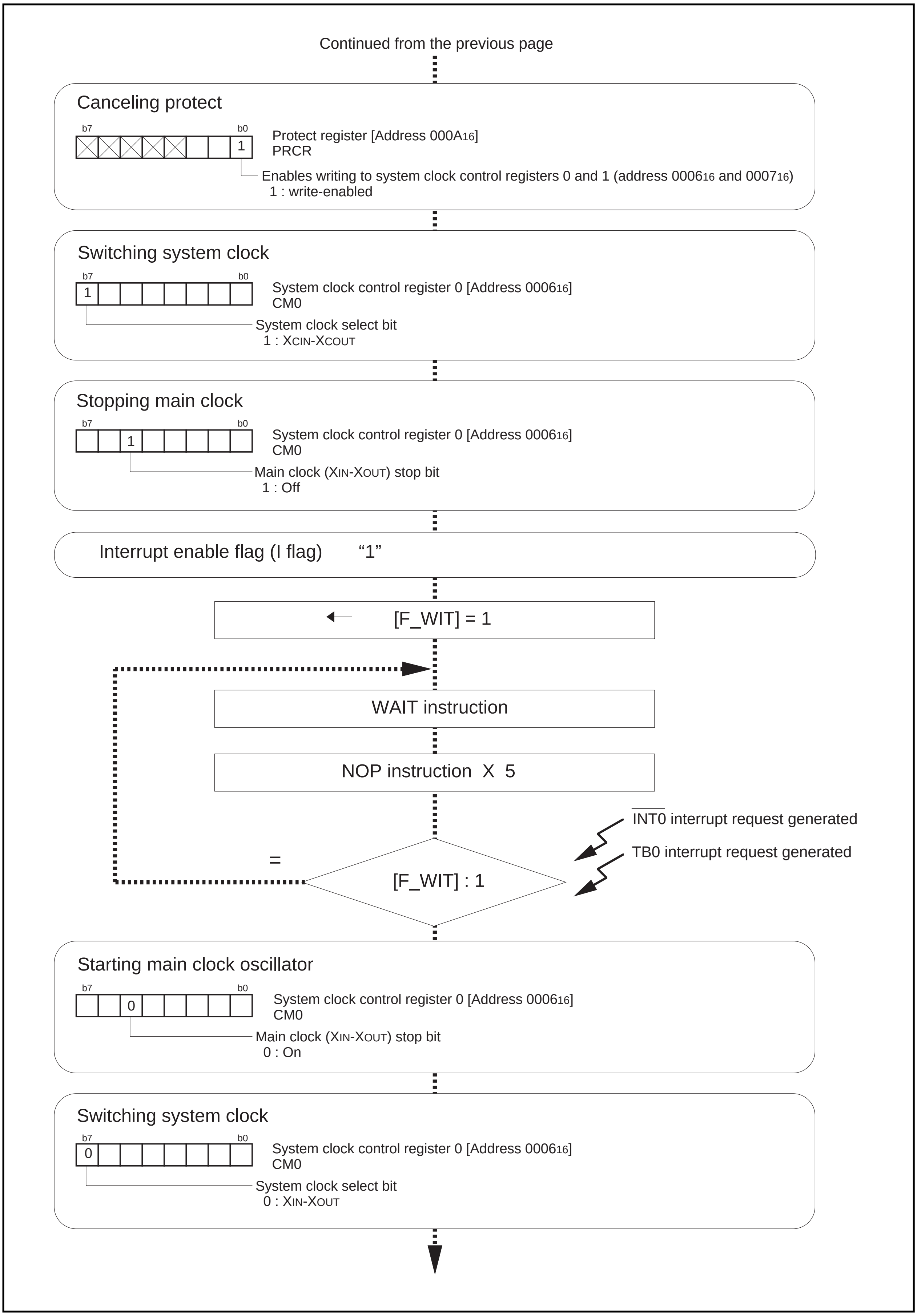

Continued from the previous page
Canceling protect
b7
b0
1
Protect register [Address $000A_{16}$]
PRCR
Enables writing to system clock control registers 0 and 1 (address 0006_{16} and 0007_{16})
1 : write-enabled
Switching system clock
b7
b0
1
System clock control register 0 [Address 0006_{16}]
CM0
System clock select bit
1 : XCIN-XCOUT
Stopping main clock
b7
b0
1
System clock control register 0 [Address 0006_{16}]
CM0
Main clock (XIN-XOUT) stop bit
1 : Off
Interrupt enable flag (I flag) “1”
← [F_WIT] = 1
WAIT instruction
NOP instruction X 5
$\overline{\text{INT0}}$ interrupt request generated
TB0 interrupt request generated
=
[F_WIT] : 1
Starting main clock oscillator
b7
b0
0
System clock control register 0 [Address 0006_{16}]
CM0
Main clock (XIN-XOUT) stop bit
0 : On
Switching system clock
b7
b0
0
System clock control register 0 [Address 0006_{16}]
CM0
System clock select bit
0 : XIN-XOUT

Figure 3.7.3. Set-up procedure of controlling power using wait mode (2)

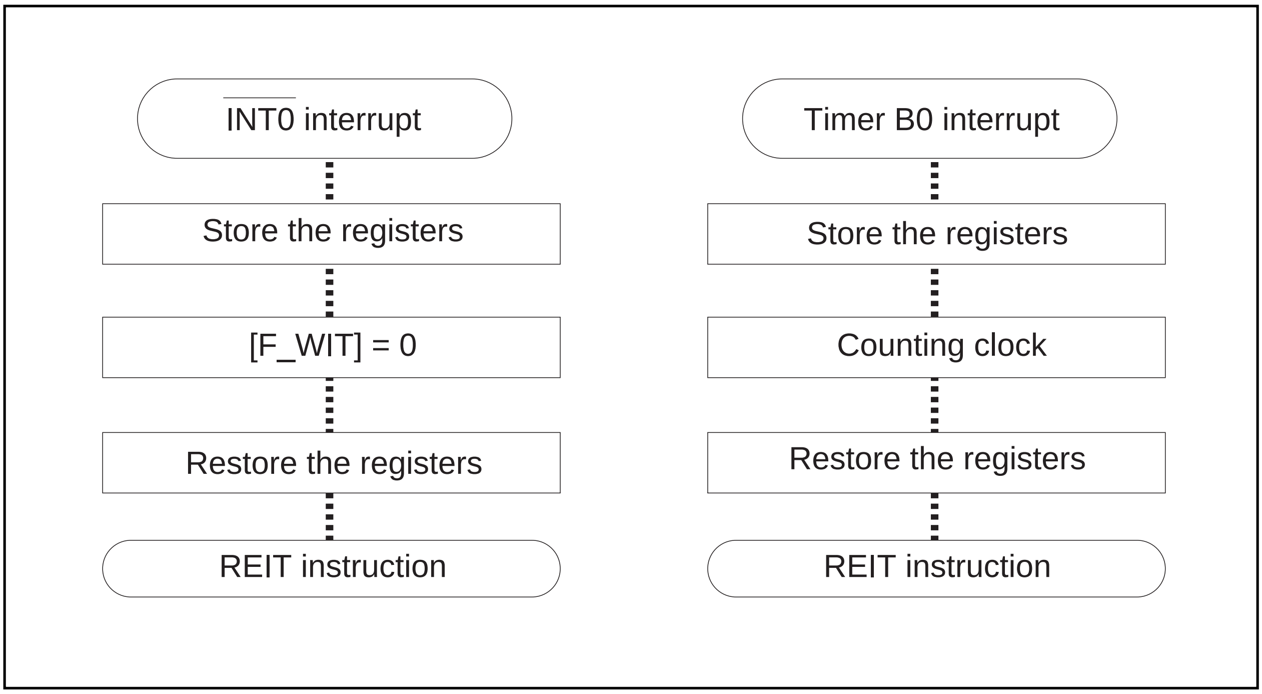


Figure 3.7.4. Set-up procedure of controlling power using wait mode (3)

Chapter 4

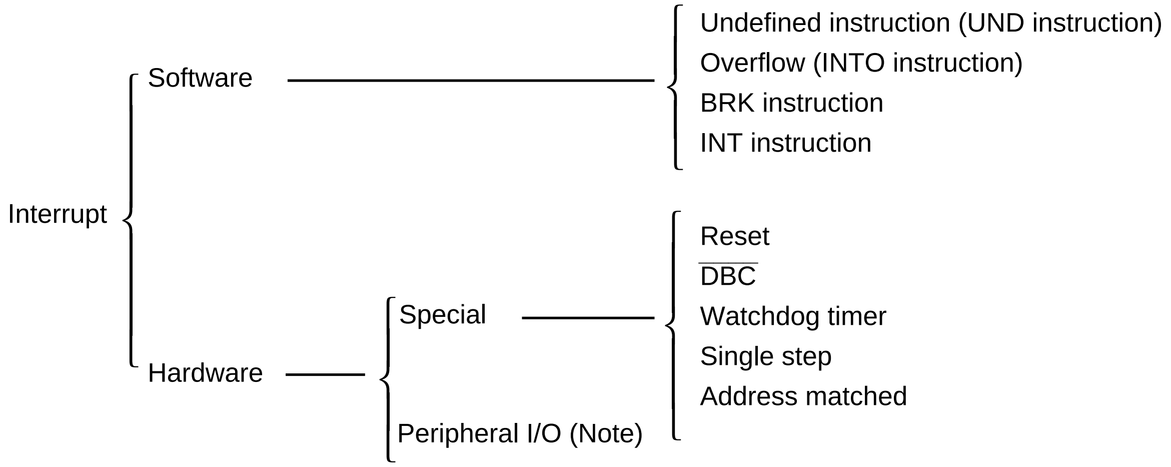
Interrupt

Under development

4.1 Overview of Interrupt

4.1.1 Type of Interrupts

Figure 4.1.1 lists the types of interrupts.



Note: Peripheral I/O interrupts are generated by the peripheral functions built into the microcomputer system.

Figure 4.1.1. Classification of interrupts

- Maskable interrupt : An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable interrupt : An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

4.1.2 Software Interrupts

A software interrupt occurs when executing certain instructions. Software interrupts are non-maskable interrupts.

- **Undefined instruction interrupt**
  An undefined instruction interrupt occurs when executing the UND instruction.
- **Overflow interrupt**
  An overflow interrupt occurs when executing the INTO instruction with the overflow flag (O flag) set to “1”. The following are instructions whose O flag changes by arithmetic:
  ABS, ADC, ADCF, ADD, CMP, DIV, DIVU, DIVX, NEG, RMPA, SBB, SHA, SUB
- **BRK interrupt**
  A BRK interrupt occurs when executing the BRK instruction.
- **INT interrupt**
  An INT interrupt occurs when assiging one of software interrupt numbers 0 through 63 and executing the INT instruction. Software interrupt numbers 0 through 31 are assigned to peripheral I/O interrupts, so executing the INT instruction allows executing the same interrupt routine that a peripheral I/O interrupt does.
  The stack pointer (SP) used for the INT interrupt is dependent on which software interrupt number is involved.
  So far as software interrupt numbers 0 through 31 are concerned, the microcomputer saves the stack pointer assignment flag (U flag) when it accepts an interrupt request. If change the U flag to “0” and select the interrupt stack pointer (ISP), and then execute an interrupt sequence. When returning from the interrupt routine, the U flag is returned to the state it was before the acceptance of interrupt request. So far as software numbers 32 through 63 are concerned, the stack pointer does not make a shift.

4.1.3 Hardware Interrupts

Hardware interrupts are classified into two types — special interrupts and peripheral I/O interrupts.

(1) Special interrupts

Special interrupts are non-maskable interrupts.

- **Reset**

  Reset occurs if an “L” is input to the $\overline{\text{RESET}}$ pin.

- **$\overline{\text{DBC}}$ interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances.

- **Watchdog timer interrupt**

  Generated by the watchdog timer.

- **Single-step interrupt**

  This interrupt is exclusively for the debugger, do not use it in other circumstances. With the debug flag (D flag) set to “1”, a single-step interrupt occurs after one instruction is executed.

- **Address match interrupt**

  An address match interrupt occurs immediately before the instruction held in the address indicated by the address match interrupt register is executed with the address match interrupt enable bit set to “1”. If an address other than the first address of the instruction in the address match interrupt register is set, no address match interrupt occurs. For address match interrupt, see 2.9 Address match Interrupt.

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 31 the INI instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Key-input interrupt**

  A key-input interrupt occurs if an “L” is input to the $\overline{\text{KI}}$ pin.

- **A-D conversion interrupt**

  This is an interrupt that the A-D converter generates.

- **UART0 and UART1 transmission interrupt**

  These are interrupts that the serial I/O transmission generates.

- **UART0 and UART1 reception interrupt**

  These are interrupts that the serial I/O reception generates.

- **Timer A0 interrupt**

  This is an interrupt that timer A generates.

- **Timer B0 interrupt and timer B1 interrupt**

  These are interrupts that timer B generates.

- **Timer X0 interrupt through timer X2 interrupt**
- **$\overline{\text{INT0}}$ interrupt and $\overline{\text{INT1}}$ interrupt**

  An $\overline{\text{INT}}$ interrupt occurs if either a rising edge or a falling edge is input to the $\overline{\text{INT}}$ pin.

4.1.4 Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.

- **Fixed vector tables**

The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from $FFFDC_{16}$ to $FFFFF_{16}$. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 4.1.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 4.1.1. Interrupts assigned to the fixed vector tables and addresses of vector tables

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	$FFFDC_{16}$ to $FFFDF_{16}$	Interrupt on UND instruction
Overflow	$FFFE0_{16}$ to $FFFE3_{16}$	Interrupt on INTO instruction
BRK instruction	$FFFE4_{16}$ to $FFFE7_{16}$	If the vector contains FF_{16}, program execution starts from the address shown by the vector in the variable vector table
Address match	$FFFE8_{16}$ to $FFFEB_{16}$	There is an address-matching interrupt enable bit
Single step (Note)	$FFFEC_{16}$ to $FFFEF_{16}$	Do not use
Watchdog timer	$FFFF0_{16}$ to $FFFF3_{16}$	
$\overline{\text{DBC}}$ (Note)	$FFFF4_{16}$ to $FFFF7_{16}$	Do not use
——	$FFFF8_{16}$ to $FFFFB_{16}$	——
Reset	$FFFFC_{16}$ to $FFFFF_{16}$	

Note: Interrupts used for debugging purposes only.

Under development

- **Variable vector tables**
  The addresses in the variable vector table can be modified, according to the user's settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 4.1.2 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

Table 4.1.2. Interrupts assigned to the variable vector tables and addresses of vector tables

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note)	BRK instruction	Cannot be masked by I flag
——		——	
Software interrupt number 11	+44 to +47 (Note)	——	
Software interrupt number 12	+48 to +51 (Note)	——	
Software interrupt number 13	+52 to +55 (Note)	Key input interrupt	
Software interrupt number 14	+56 to +59 (Note)	A-D	
——		——	
Software interrupt number 17	+68 to +71 (Note)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note)	Timer A0	
Software interrupt number 22	+88 to +91 (Note)	Timer X0	
Software interrupt number 23	+92 to +95 (Note)	Timer X1	
Software interrupt number 24	+96 to +99 (Note)	Timer X2	
Software interrupt number 25	+100 to +103 (Note)	——	
Software interrupt number 26	+104 to +107 (Note)	Timer B0	
Software interrupt number 27	+108 to +111 (Note)	Timer B1	
Software interrupt number 28	+112 to +115 (Note)	——	
Software interrupt number 29	+116 to +119 (Note)	$\overline{\text{INT0}}$	
Software interrupt number 30	+120 to +123 (Note)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note)	——	
Software interrupt number 32 to Software interrupt number 63	+128 to +131 (Note) to +252 to +255 (Note)	Software interrupt	Cannot be masked by I flag

Note : Address relative to address in interrupt table register (INTB).

Under development

4.2 Interrupt Control

Descriptions are given here regarding how to enable or disable maskable interrupts and how to set the priority to be accepted. What is described here does not apply to non-maskable interrupts.

Enable or disable a non-maskable interrupt using the interrupt enable flag (I flag), interrupt priority level selection bit, or processor interrupt priority level (IPL). Whether an interrupt request is present or absent is indicated by the interrupt request bit. The interrupt request bit and the interrupt priority level selection bit are located in the interrupt control register of each interrupt. Also, the interrupt enable flag (I flag) and the IPL are located in the flag register (FLG).

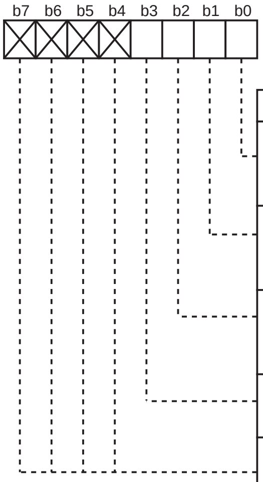
Table 4.2.1 shows the memory map of the interrupt control registers, and Table 4.2.2 shows the interrupt control registers.

$004D_{16}$	Key input interrupt control register(KUPIC)
$004E_{16}$	A-D conversion interrupt control register (ADIC)
$004F_{16}$	
0050_{16}	
0051_{16}	UART0 transmit interrupt control register (S0TIC)
0052_{16}	UART0 receive interrupt control register (S0RIC)
0053_{16}	UART1 transmit interrupt control regster(S1TIC)
0054_{16}	UART1 receive interrupt control register(S1RIC)
0055_{16}	Timer A0 interrupt control register (TA0IC)
0056_{16}	Timer X0 interrupt control register (TX0IC)
0057_{16}	Timer X1 interrupt control register (TX1IC)
0058_{16}	Timer X2 interrupt control register (TX2IC)
0059_{16}	
$005A_{16}$	Timer B0 interrupt control register (TB0IC)
$005B_{16}$	Timer B1 interrupt control register (TB1IC)
$005C_{16}$	
$005D_{16}$	INT0 interrupt control register (INT0IC)
$005E_{16}$	INT1 interrupt control register (INT1IC)

Table 4.2.1. Memory map of the interrupt control registers

Interrupt control register

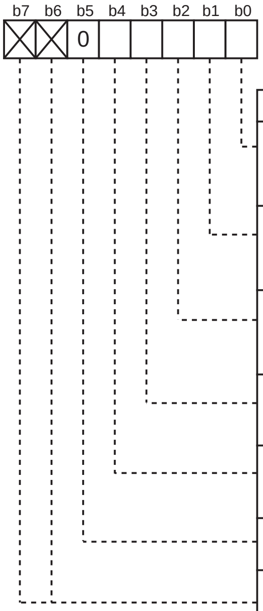
Symbol	Address	When reset
KUPIC	$004D_{16}$	$XXXXX000_2$
ADIC	$004E_{16}$	$XXXXX000_2$
SiTIC(i=0, 1)	0051_{16}, 0053_{16}	$XXXXX000_2$
SiRIC(i=0, 1)	0052_{16}, 0054_{16}	$XXXXX000_2$
TAiIC(i=0)	0055_{16}	$XXXXX000_2$
TXiIC(i=0 to 2)	0056_{16} to 0058_{16}	$XXXXX000_2$
TBiIC(i=0, 1)	$005A_{16}$, $005B_{16}$	$XXXXX000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note)
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Symbol	Address	When reset
INTiIC(i=0, 1)	$005D_{16}$, $005E_{16}$	$XX00X000_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note)
POL	Polarity select bit	0 : Selects falling edge 1 : Selects rising edge	○	○
Reserved bit		Always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Figure 4.2.2. Interrupt control registers

4.2.1 Interrupt Enable Flag

The interrupt enable flag (I flag) controls the enabling and disabling of maskable interrupts. Setting this flag to “1” enables all maskable interrupts; setting it to “0” disables all maskable interrupts. This flag is set to “0” after reset.

The content is changed when the I flag is changed causes the acceptance of the interrupt request in the following timing:

- When changing the I flag using the REIT instruction, the acceptance of the interrupt takes effect as the REIT instruction is executed.
- When changing the I flag using one of the FCLR, FSET, POPC, and LDC instructions, the acceptance of the interrupt is effective as the next instruction is executed.

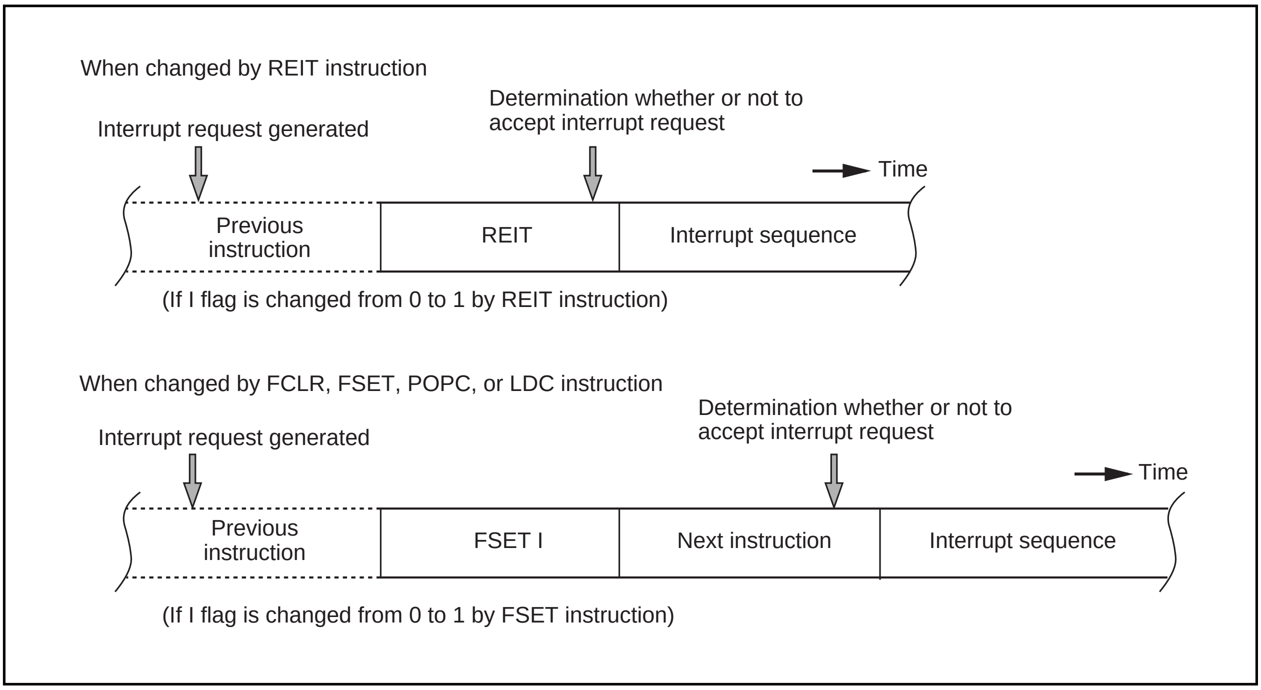

When changed by REIT instruction
Interrupt request generated
Determination whether or not to accept interrupt request
Time
Previous instruction
REIT
Interrupt sequence
(If I flag is changed from 0 to 1 by REIT instruction)
When changed by FCLR, FSET, POPC, or LDC instruction
Interrupt request generated
Determination whether or not to accept interrupt request
Time
Previous instruction
FSET I
Next instruction
Interrupt sequence
(If I flag is changed from 0 to 1 by FSET instruction)

Figure 4.2.3. The timing of reflecting the change in the I flag to the interrupt

4.2.2 Interrupt Request Bit

The interrupt request bit is set to "1" by hardware when an interrupt is requested. After the interrupt is accepted and jumps to the corresponding interrupt vector, the request bit is set to "0" by hardware. The interrupt request bit can also be set to "0" by software. (Do not set this bit to "1").

Under development

4.2.3 Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Set the interrupt priority level using the interrupt priority level select bit, which is one of the component bits of the interrupt control register. When an interrupt request occurs, the interrupt priority level is compared with the IPL. The interrupt is enabled only when the priority level of the interrupt is higher than the IPL. Therefore, setting the interrupt priority level to “0” disables the interrupt.
Table 4.2.1 shows the settings of interrupt priority levels and Table 4.2.2 shows the interrupt levels enabled, according to the consist of the IPL.

The following are conditions under which an interrupt is accepted:
· interrupt enable flag (I flag) = 1
· interrupt request bit = 1
· interrupt priority level > IPL

The interrupt enable flag (I flag), the interrupt request bit, the interrupt priority select bit, and the IPL are independent, and they are not affected by one another.

Table 4.2.1. Settings of interrupt priority levels

Interrupt priority level select bit	Interrupt priority level	Priority order
b2 b1 b0 0 0 0	Level 0 (interrupt disabled)	———
0 0 1	Level 1	Low
0 1 0	Level 2	
0 1 1	Level 3	
1 0 0	Level 4	
1 0 1	Level 5	
1 1 0	Level 6	
1 1 1	Level 7	High

Table 4.2.2. Interrupt levels enabled according to the contents of the IPL

IPL	Enabled interrupt priority levels
IPL_2 IPL_1 IPL_0 0 0 0	Interrupt levels 1 and above are enabled
0 0 1	Interrupt levels 2 and above are enabled
0 1 0	Interrupt levels 3 and above are enabled
0 1 1	Interrupt levels 4 and above are enabled
1 0 0	Interrupt levels 5 and above are enabled
1 0 1	Interrupt levels 6 and above are enabled
1 1 0	Interrupt levels 7 and above are enabled
1 1 1	All maskable interrupts are disabled

When either the IPL or the interrupt priority level is changed, the new level is reflected to the interrupt in the following timing:

- When changing the IPL using the REIT instruction, the reflection takes effect as of the instruction that is executed in 2 clock cycles after the last clock cycle in volved in the REIT instruction.
- When changing the IPL using either the POPC, LDC or LDIPL instruction, the reflection takes effect as of the instruction that is executed in 3 cycles after the last clock cycle involved in the instruction used.
- When changing the interrupt priority level using the MOV or similar instruction, the reflection takes effect as of the instruction that is executed in 2 clock cycles after the last clock cycle involved in the instruction used.

4.2.4 Rewrite the interrupt control register

To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
      FCLR     I              ; Disable interrupts.
      AND.B    #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
      NOP                     ; Four NOP instructions are required when using HOLD function.
      NOP
      FSET     I              ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
      FCLR     I              ; Disable interrupts.
      AND.B    #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
      MOV.W    MEM, R0        ; Dummy read.
      FSET     I              ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
      PUSHC    FLG            ; Push Flag register onto stack
      FCLR     I              ; Disable interrupts.
      AND.B    #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
      POPC     FLG            ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

4.3 Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SMOVB, SMOVF, SSTR or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

In the interrupt sequence, the processor carries out the following in sequence given:

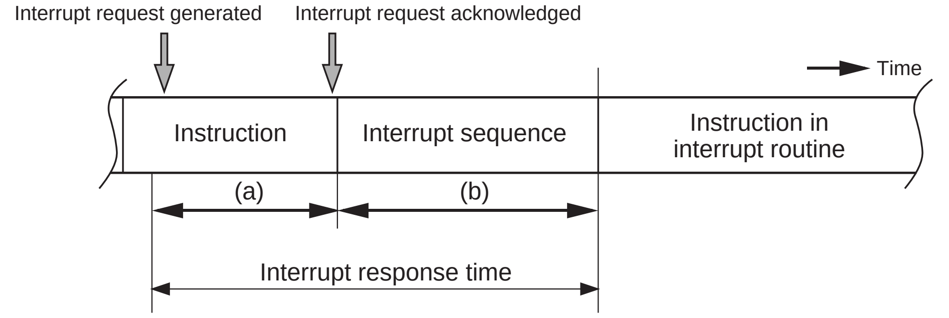
(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 00000_{16}.
(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.
(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to “0” (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)
(4) Saves the content of the temporary register (Note 1) within the CPU in the stack area.
(5) Saves the content of the program counter (PC) in the stack area.
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

After the interrupt sequence is completed, the processor resumes executing instructions from the first address of the interrupt routine.

Note: This register cannot be utilized by the user.

4.3.1 Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 4.3.1 shows the interrupt response time.


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
(b)
Interrupt response time

(a) Time from interrupt request is generated to when the instruction then under execution is completed.
(b) Time in which the instruction sequence is executed.

Figure 4.3.1. Interrupt response time

Under development

Time (a) is dependent on the instruction under execution. Thirty cycles is the maximum required for the DIVX instruction (without wait).

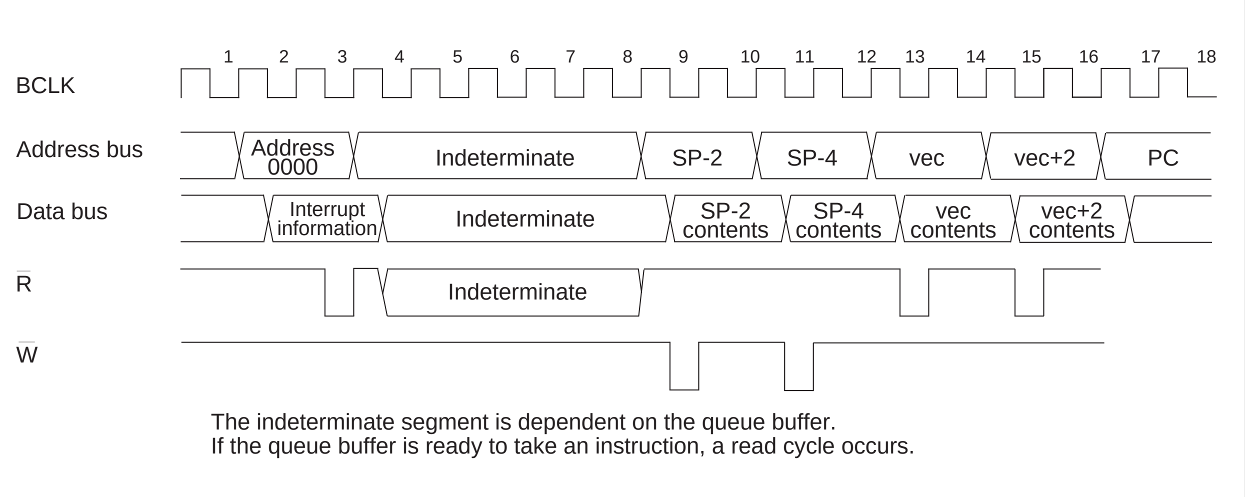
Time (b) is as shown in Table 4.3.1.

Table 4.3.1. Time required for executing the interrupt sequence

Interrupt vector address	Stack pointer (SP) value	16-Bit bus, without wait	8-Bit bus, without wait
Even	Even	18 cycles (Note 1)	20 cycles (Note 1)
Even	Odd	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Even	19 cycles (Note 1)	20 cycles (Note 1)
Odd (Note 2)	Odd	20 cycles (Note 1)	20 cycles (Note 1)

Note 1: Add 2 cycles in the case of a $\overline{\text{DBC}}$ interrupt; add 1 cycle in the case either of an address coincidence interrupt or of a single-step interrupt.

Note 2: Locate an interrupt vector address in an even address, if possible.



The indeterminate segment is dependent on the queue buffer.
If the queue buffer is ready to take an instruction, a read cycle occurs.

Figure 4.3.2. Time required for executing the interrupt sequence

4.3.2 Variation of IPL when Interrupt Request is Accepted

If an interrupt request is accepted, the interrupt priority level of the accepted interrupt is set in the IPL.

If an interrupt request, that does not have an interrupt priority level, is accepted, one of the values shown in Table 4.3.2 is set in the IPL.

Table 4.3.2. Relationship between interrupts without interrupt priority levels and IPL

Interrupt sources without priority levels	Value set in the IPL
Watchdog timer	7
Reset	0
Other	Not changed

4.3.3 Saving Registers

In the interrupt sequence, only the contents of the flag register (FLG) and that of the program counter (PC) are saved in the stack area.

First, the processor saves the four higher-order bits of the program counter, and 4 upper-order bits and 8 lower-order bits of the FLG register, 16 bits in total, in the stack area, then saves 16 lower-order bits of the program counter. Figure 4.3.3 shows the state of the stack as it was before the acceptance of the interrupt request, and the state the stack after the acceptance of the interrupt request.

Save other necessary registers at the beginning of the interrupt routine using software. Using the PUSHM instruction alone can save all the registers except the stack pointer (SP).

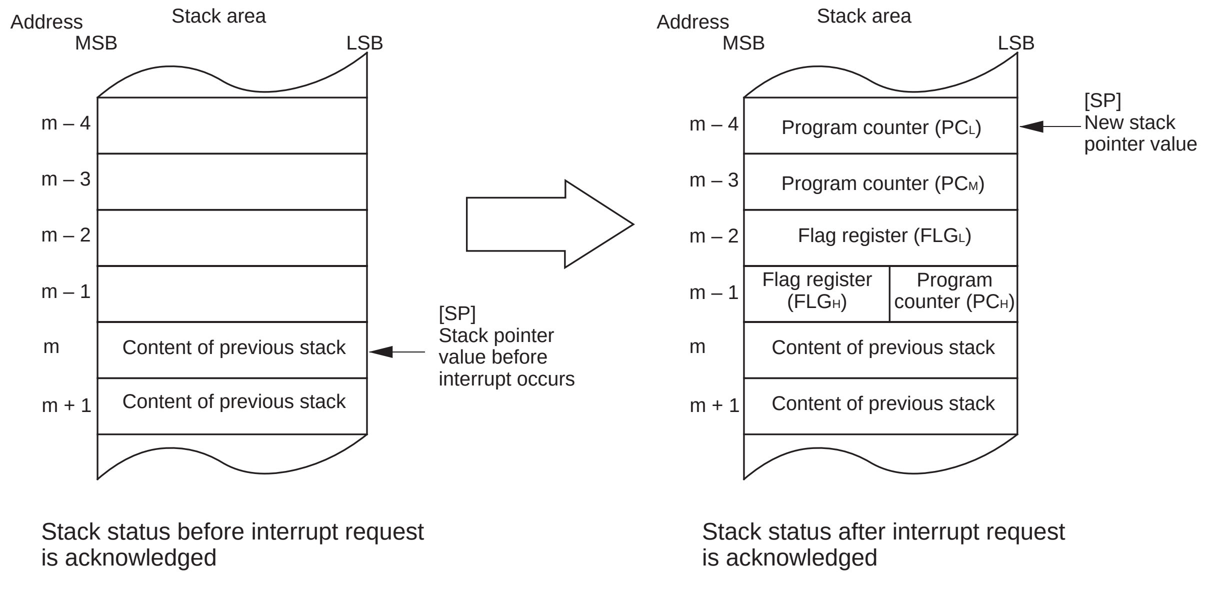
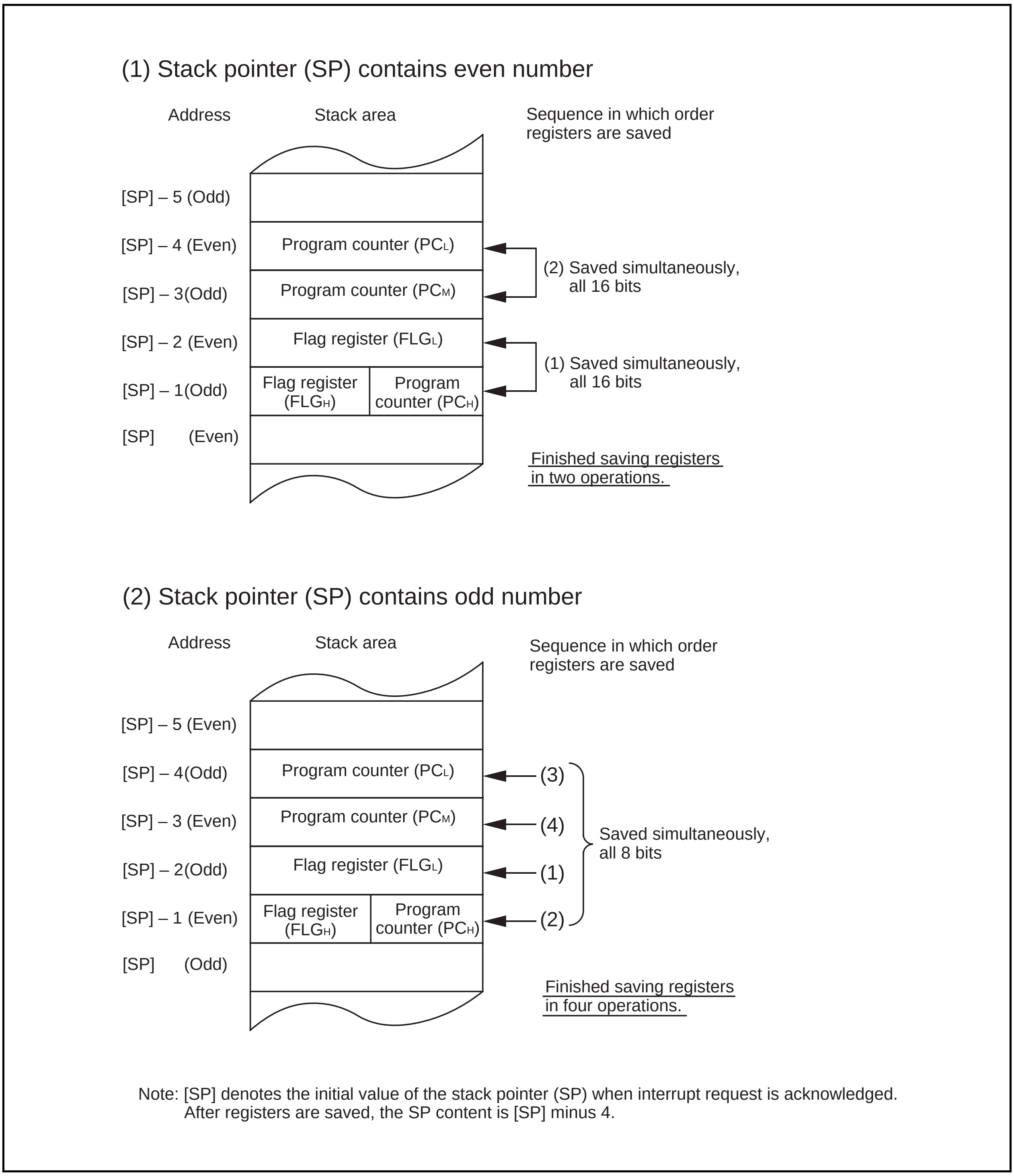

Address
Stack area
MSB
LSB
m – 4
m – 3
m – 2
m – 1
m
m + 1
Content of previous stack
Content of previous stack
[SP]
Stack pointer value before interrupt occurs
Stack status before interrupt request is acknowledged
Address
Stack area
MSB
LSB
m – 4
m – 3
m – 2
m – 1
m
m + 1
Program counter (PCL)
Program counter (PCM)
Flag register (FLGL)
Flag register (FLGH)
Program counter (PCH)
Content of previous stack
Content of previous stack
[SP]
New stack pointer value
Stack status after interrupt request is acknowledged

Figure 4.3.3. State of stack before and after acceptance of interrupt request

The operation of saving registers carried out in the interrupt sequence is dependent on whether the content of the stack pointer, at the time of acceptance of an interrupt request, is even or odd. If the content of the stack pointer (Note) is even, the content of the flag register (FLG) and the content of the program counter (PC) are saved, 16 bits at a time. If odd, their contents are saved in two steps, 8 bits at a time. Figure 4.3.4 shows the operation of the saving registers.
Note: Stack pointer indicated by U flag.



Note: [SP] denotes the initial value of the stack pointer (SP) when interrupt request is acknowledged. After registers are saved, the SP content is [SP] minus 4.

Figure 4.3.4. Operation of saving registers

4.4 Returning from an Interrupt Routine

Executing the REIT instruction at the end of an interrupt routine returns the contents of the flag register (FLG) as it was immediately before the start of interrupt sequence and the contents of the program counter (PC), both of which have been saved in the stack area. Then control returns to the program that was being executed before the acceptance of the interrupt request, so that the suspended process resumes.

Return the other registers saved by software within the interrupt routine using the POPM or similar instruction before executing the REIT instruction.

4.5 Interrupt Priority

If there are two or more interrupt requests occurring at a point in time within a single sampling (checking whether interrupt requests are made), the interrupt assigned a higher priority is accepted.

Assign an arbitrary priority to maskable interrupts (peripheral I/O interrupts) using the interrupt priority level select bit. If the same interrupt priority level is assigned, however, the interrupt assigned a higher hardware priority is accepted (see Figure 4.5.1).

Priorities of the special interrupts, such as Reset (dealt with as an interrupt assigned the highest priority), watchdog timer interrupt, etc. are regulated by hardware.

Figure 4.5.2 shows the priorities of hardware interrupts.

Software interrupts are not affected by the interrupt priority. If an instruction is executed, control branches invariably to the interrupt routine.

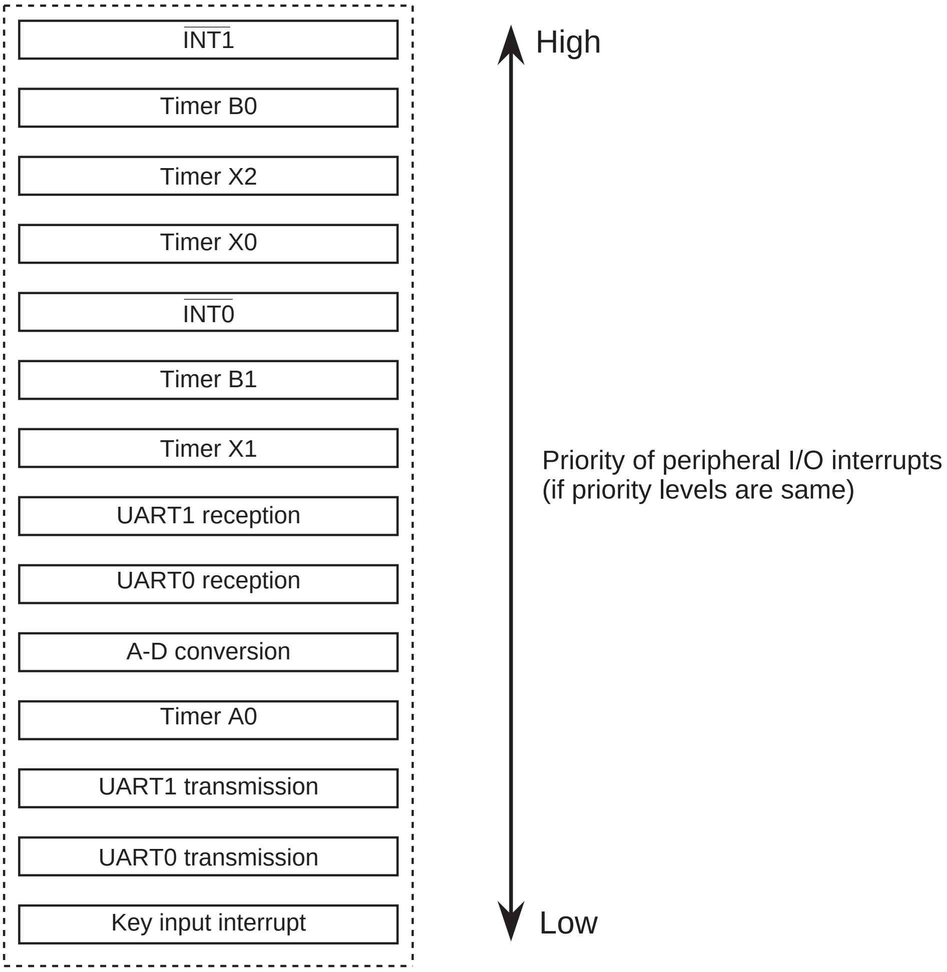


Figure 4.5.1. Maskable interrupts priorities (peripheral I/O interrupts)

Reset > $\overline{\text{DBC}}$ > Watchdog timer > Peripheral I/O > Single step > Address match

Figure 4.5.2. Hardware interrupts priorities

4.6 Multiple Interrupts

The state when control branched to an interrupt routine is described below:

- The interrupt enable flag (I flag) is set to “0” (the interrupt is disabled).
- The interrupt request bit of the accepted interrupt is set to “0”.
- The processor interrupt priority level (IPL) is assigned to the same interrupt priority level as assigned to the accepted interrupt.

Setting the interrupt enable flag (I flag) to “1” within an interrupt routine allows an interrupt request assigned a priority higher than the IPL to be accepted. Figure 4.6.1 shows the scheme of multiple interrupts.
An interrupt request that is not accepted because of low priority will be held. If the condition following is met when the REIT instruction returns the IPL and the interrupt priority is determined, then the interrupt request being held is accepted.

Interrupt priority level of the interrupt request being held > Returned the IPL

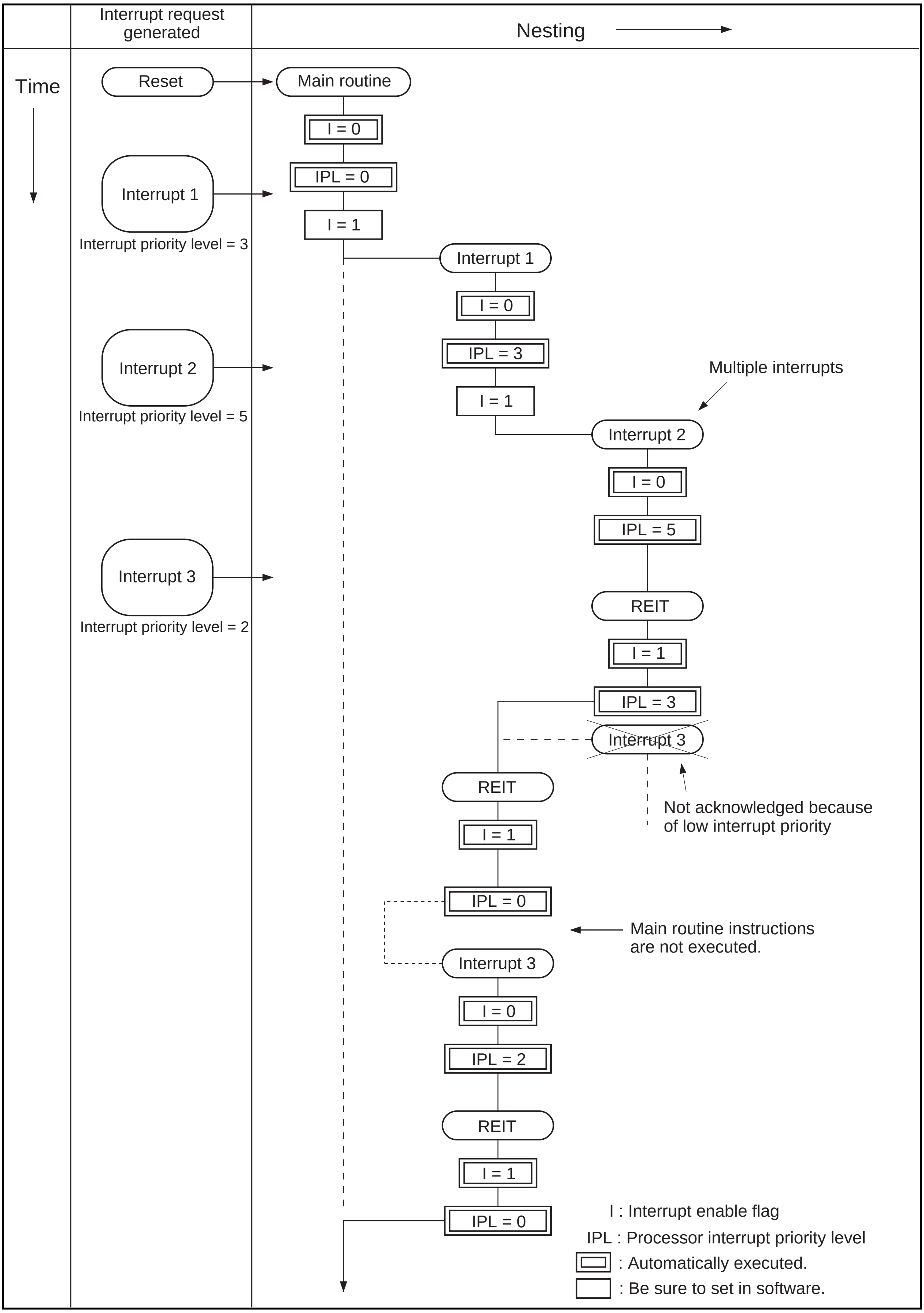

Interrupt request generated
Nesting
Time
Reset
Main routine
I = 0
IPL = 0
I = 1
Interrupt 1
Interrupt priority level = 3
Interrupt 1
I = 0
IPL = 3
I = 1
Interrupt 2
Interrupt priority level = 5
Multiple interrupts
Interrupt 2
I = 0
IPL = 5
Interrupt 3
Interrupt priority level = 2
REIT
I = 1
IPL = 3
Interrupt 3
Not acknowledged because of low interrupt priority
REIT
I = 1
IPL = 0
Main routine instructions are not executed.
Interrupt 3
I = 0
IPL = 2
REIT
I = 1
IPL = 0
I : Interrupt enable flag
IPL : Processor interrupt priority level
: Automatically executed.
: Be sure to set in software.

Figure 4.6.1. Multiple interrupts

Under development

4.7 Precautions for Interrupts

(1) Reading address 0000_{16}

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence.
  The interrupt request bit of the certain interrupt written in address 0000_{16} will then be set to “0”.
  Reading address 0000_{16} by software sets enabled highest priority interrupt source request bit to “0”.
  Though the interrupt is generated, the interrupt routine may not be executed.
  Do not read address 0000_{16} by software.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 0000_{16}. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. Concerning the first instruction immediately after reset, generating any interrupts is prohibited.

(3) External interrupt

- Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to pins $\overline{\mathrm{INT}_0}$ and $\overline{\mathrm{INT}_1}$ regardless of the CPU operation clock.
- When the polarity of the $\overline{\mathrm{INT}_0}$ and $\overline{\mathrm{INT}_1}$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 4.7.1 shows the procedure for changing the $\overline{\mathrm{INT}}$ interrupt generate factor.

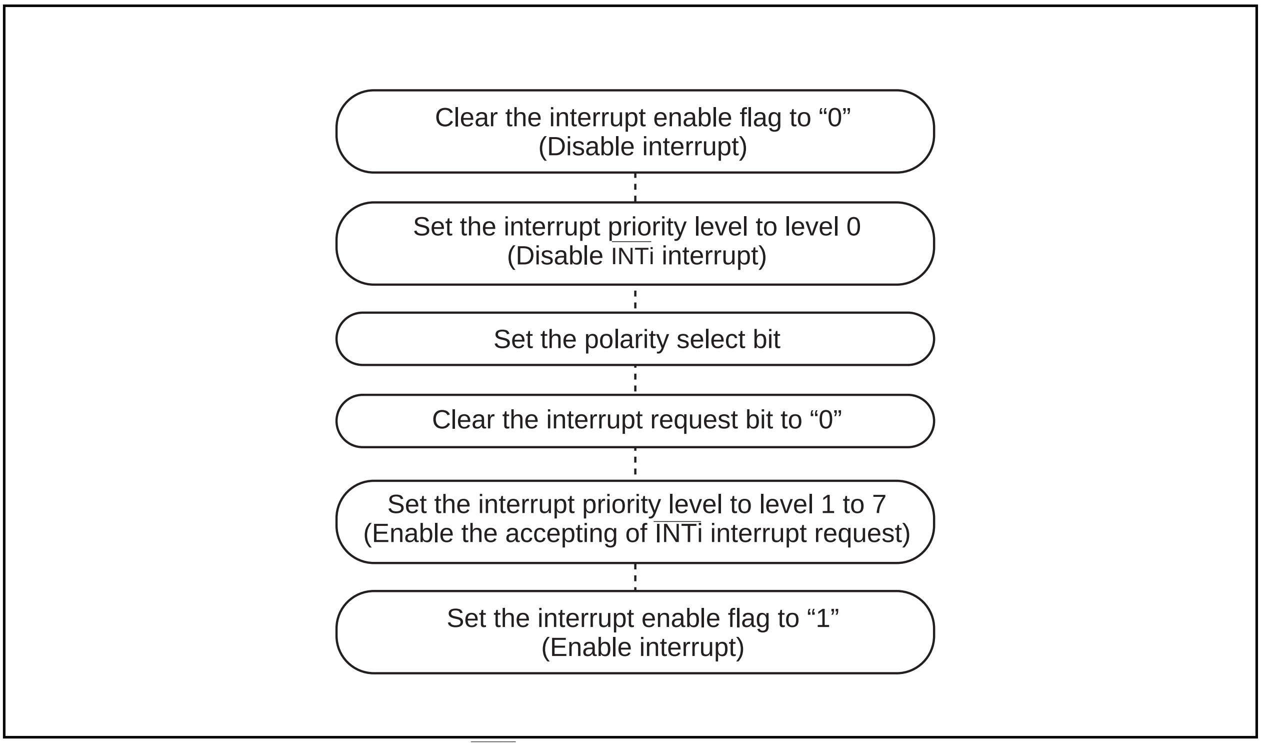

Clear the interrupt enable flag to “0”
(Disable interrupt)
Set the interrupt priority level to level 0
(Disable INTi interrupt)
Set the polarity select bit
Clear the interrupt request bit to “0”
Set the interrupt priority level to level 1 to 7
(Enable the accepting of INTi interrupt request)
Set the interrupt enable flag to “1”
(Enable interrupt)

Figure 4.7.1. Switching condition of $\overline{\mathrm{INT}}$ interrupt request

(4) Rewrite the interrupt control register

- To rewrite the interrupt control register, do so at a point that does not generate the interrupt request for that register. If there is possibility of the interrupt request occur, rewrite the interrupt control register after the interrupt is disabled. The program examples are described as follow:

Example 1:

```
INT_SWITCH1:
      FCLR     I              ; Disable interrupts.
      AND.B    #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
      NOP                     ; Four NOP instructions are required when using HOLD function.
      NOP
      FSET     I              ; Enable interrupts.
```

Example 2:

```
INT_SWITCH2:
      FCLR     I              ; Disable interrupts.
      AND.B    #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
      MOV.W    MEM, R0        ; Dummy read.
      FSET     I              ; Enable interrupts.
```

Example 3:

```
INT_SWITCH3:
      PUSHC    FLG            ; Push Flag register onto stack
      FCLR     I              ; Disable interrupts.
      AND.B    #00h, 0055h    ; Clear TA0IC int. priority level and int. request bit.
      POPC     FLG            ; Enable interrupts.
```

The reason why two NOP instructions (four when using the HOLD function) or dummy read are inserted before FSET I in Examples 1 and 2 is to prevent the interrupt enable flag I from being set before the interrupt control register is rewritten due to effects of the instruction queue.

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.
  Instructions : AND, OR, BCLR, BSET

Chapter 5

Standard Characteristics

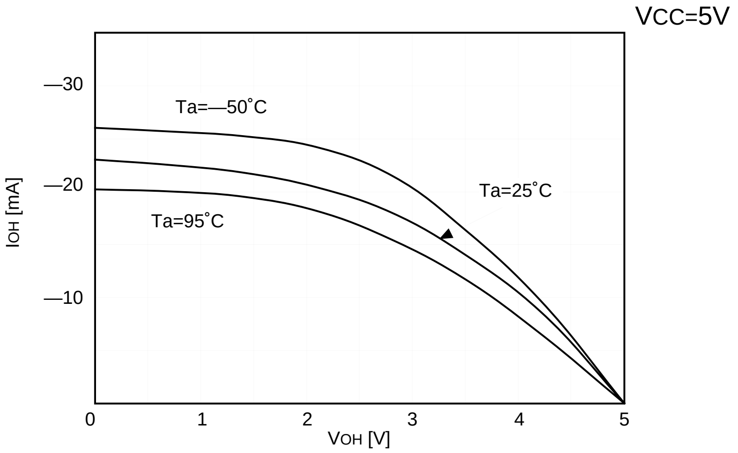
5.1 Standard DC Characteristics

The standard characteristics given in this section are examples of M30201M4-XXXFP. The contents of these examples cannot be guaranteed. For standardized values, see “Electric characteristics”.

5.1.1 Standard Ports Characteristics

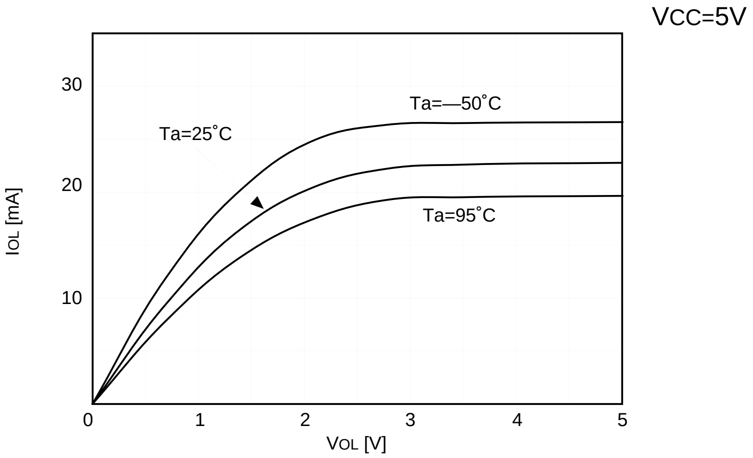
Figures 5.1.1 through 5.1.6 show the standard ports characteristics.

Under
development



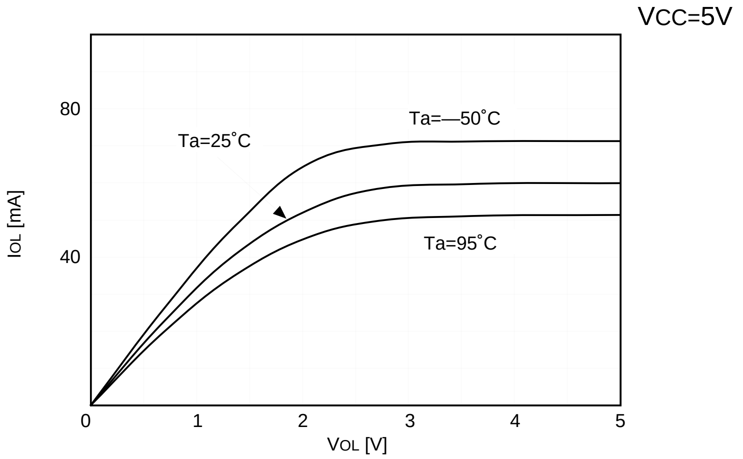
Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figure 5.1.1. I_{OH} - V_{OH} standard characteristics of ports P0 to P7 (V_{CC} = 5V)



Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

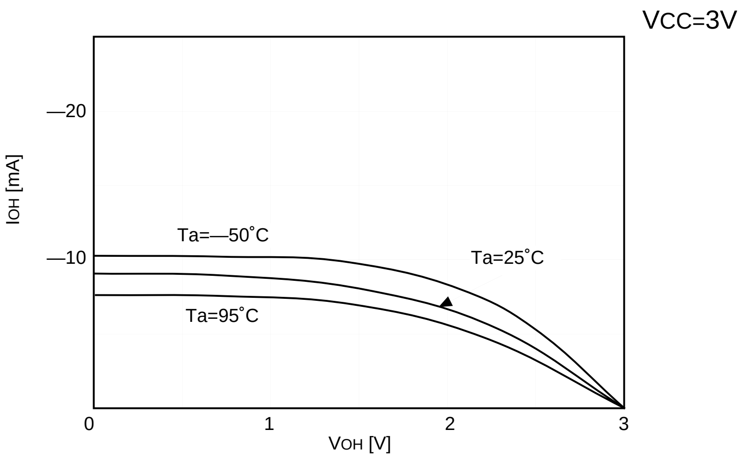
Figure 5.1.2. I_{OL} - V_{OL} standard characteristics of ports P0 to P7 (V_{CC} = 5V)



Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

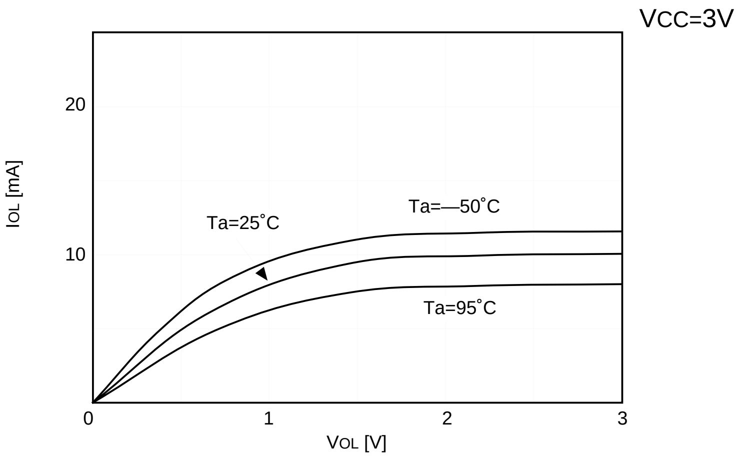
Figure 5.1.3. I_{OL} - V_{OL} standard characteristics of port P1 (V_{CC} = 5V, HIGH POWER)

Under development



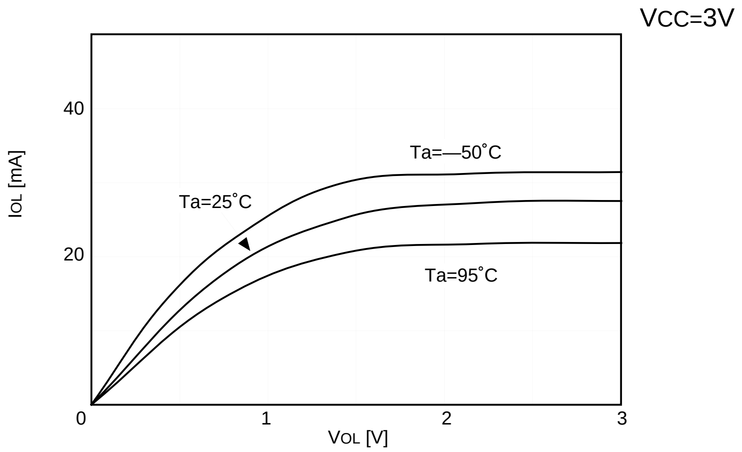
Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figure 5.1.4. I_{OH} - V_{OH} standard characteristics of ports P0 to P7 (V_{CC} = 3V)



Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figure 5.1.5. I_{OL} - V_{OL} standard characteristics of ports P0 to P7 (V_{CC} = 3V)



Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

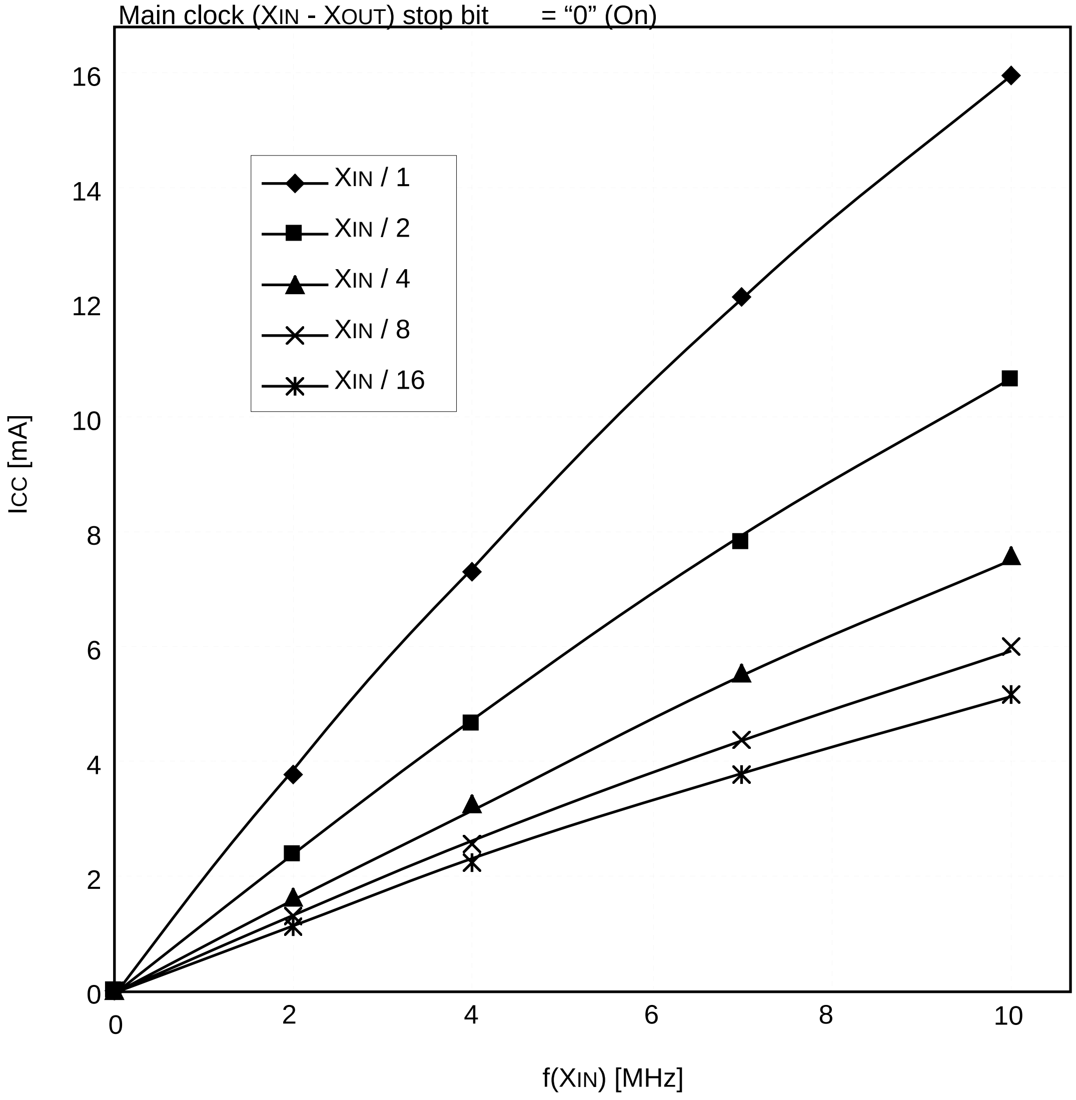
Figure 5.1.6. I_{OL} - V_{OL} standard characteristics of port P1 (V_{CC} = 3V, HIGH POWER)

5.1.2 Characteristics of ICC-f(XIN)

Figures 5.1.7 and 5.1.8 show the Characteristics of ICC-f(XIN).

VCC=5V

- Measurement conditions :
  VCC = 5V, Ta = 25˚C, f(XIN) : square waveform input, single-chip mode
  When access to ROM and RAM without wait
- Register setting condition
  XIN - XOUT drive capacity select bit = “1” (HIGH)
  Main clock (XIN - XOUT) stop bit = “0” (On)

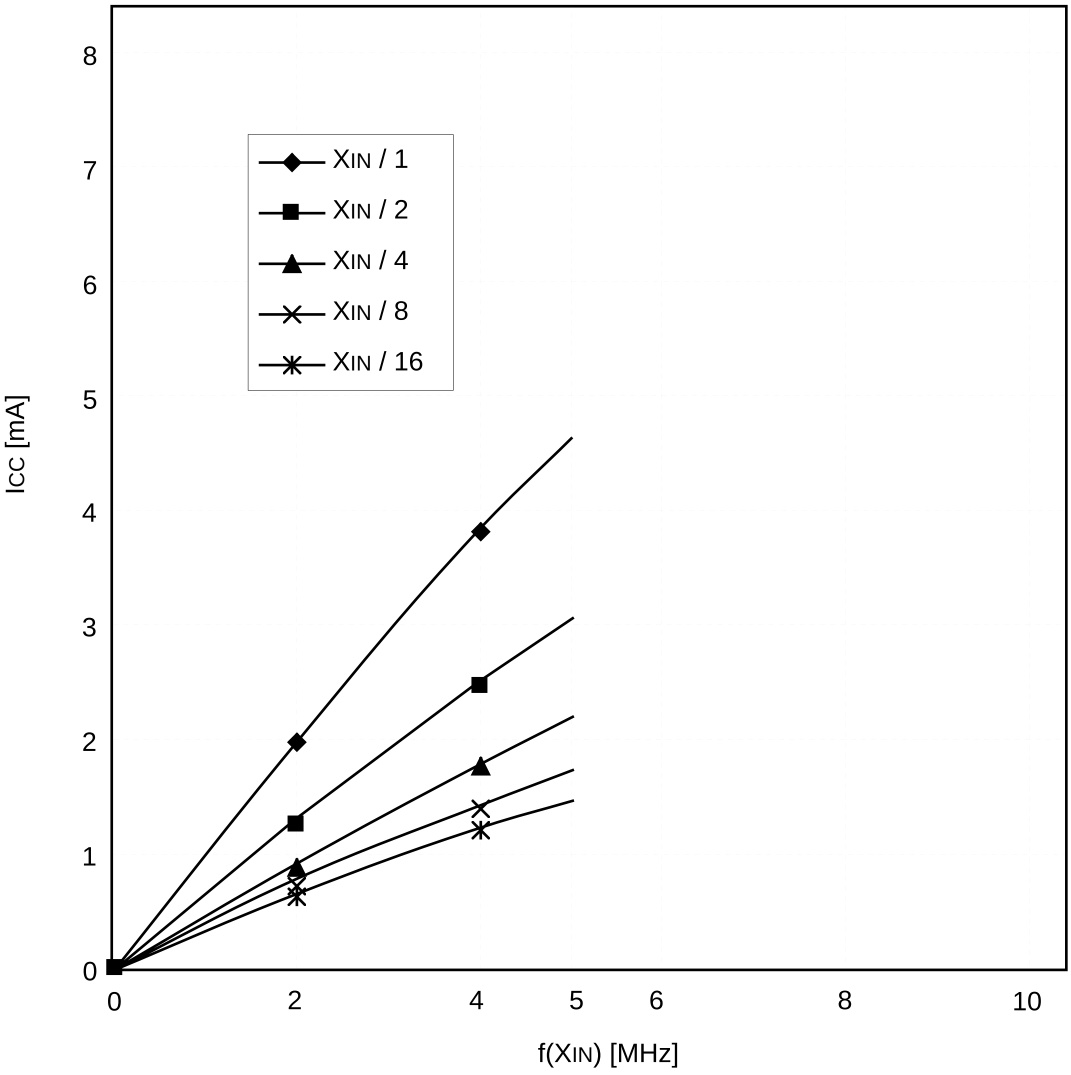


Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figures 5.1.7. Characteristics of ICC-f(XIN) (VCC = 5V)

VCC=3V

- Measurement conditions :
  VCC = 3V, Ta = 25˚C, f(XIN) : square waveform input, single-chip mode
  When access to ROM and RAM without wait
- Register setting condition
  XIN - XOUT drive capacity select bit = “1” (HIGH)
  Main clock (XIN - XOUT) stop bit = “0” (On)

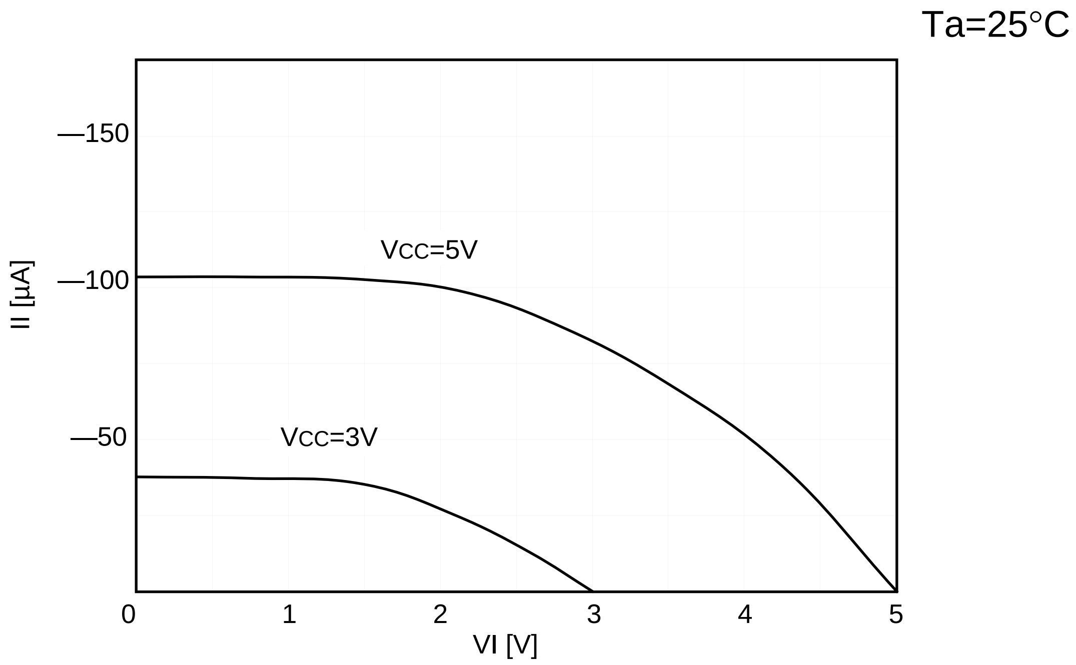


Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figures 5.1.8. Characteristics of ICC-f(XIN) (VCC = 3V)

5.2 Standard Characteristics of Pull-Up Resistor

Figure 5.2.1 shows an example of the standard characteristics of the pull-up resistor.



Note: Data described here are characteristic examples. The data values are not guaranteed.

Figure 5.2.1. Example of the standard characteristics of the pull-up resistor

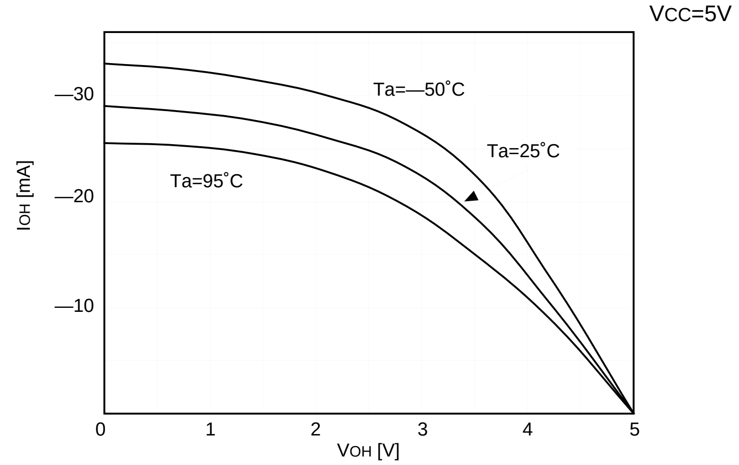
5.3 Standard DC Characteristics (Flash memory version)

The standard characteristics given in this section are examples of M30201F6FP. The contents of these examples cannot be guaranteed. For standardized values, see “Electric characteristics”.

5.3.1 Standard Ports Characteristics

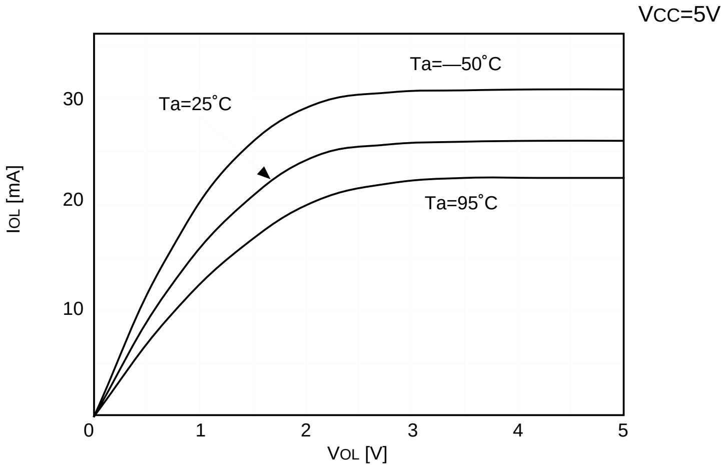
Figures 5.3.1 through 5.3.3 show the standard ports characteristics.

Under development



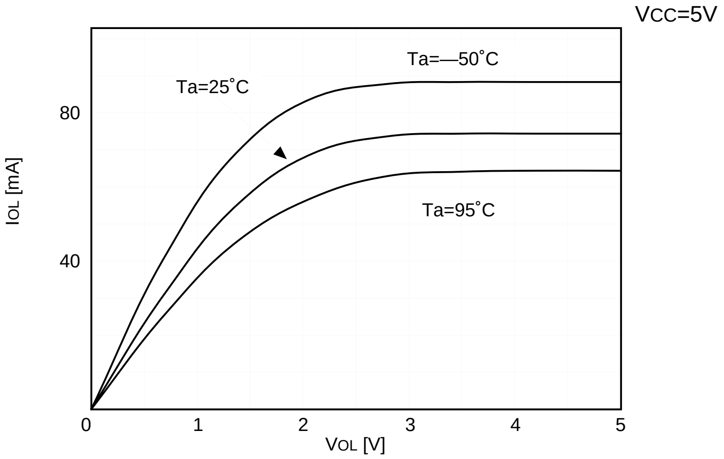
Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figure 5.3.1. I_{OH} - V_{OH} standard characteristics of ports P0 to P7 (V_{CC} = 5V)



Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figure 5.3.2. I_{OL} - V_{OL} standard characteristics of ports P0 to P7 (V_{CC} = 5V)



Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

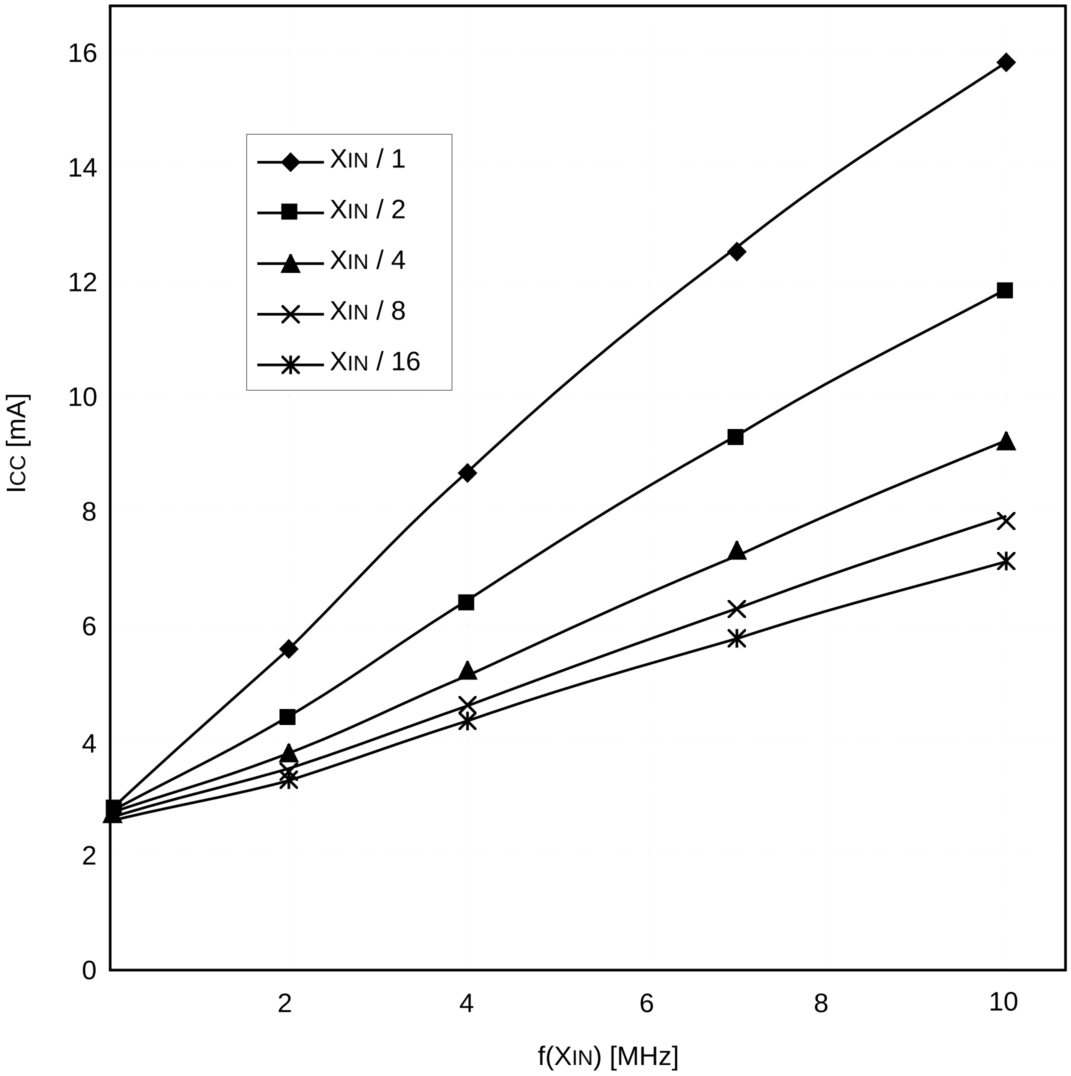
Figure 5.3.3. I_{OL} - V_{OL} standard characteristics of port P1 (V_{CC} = 5V, HIGH POWER)

5.3.2 Characteristics of ICC-f(XIN)

Figure 5.3.4 shows the Characteristics of ICC-f(XIN).

VCC=5V

- Measurement conditions :
  VCC = 5V, Ta = 25˚C, f(XIN) : square waveform input, single-chip mode
  When access to ROM and RAM without wait
- Register setting condition
  XIN - XOUT drive capacity select bit = “1” (HIGH)
  Main clock (XIN - XOUT) stop bit = “0” (On)

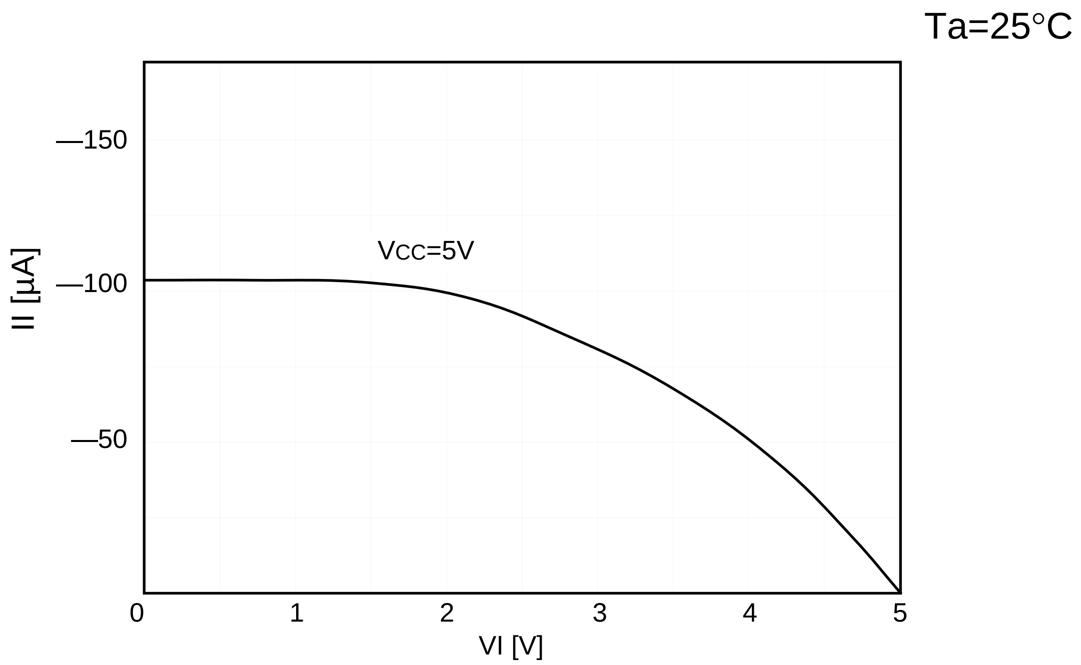


Note: Data described here are characteristic examples. The data values are not guaranteed.
Refer to section “Electrical characteristics” for rated values.

Figures 5.3.4. Characteristics of ICC-f(XIN) (VCC = 5V)

5.4 Standard Characteristics of Pull-Up Resistor

Figure 5.4.1 shows an example of the standard characteristics of the pull-up resistor.



Note: Data described here are characteristic examples. The data values are not guaranteed.

Figure 5.4.1. Example of the standard characteristics of the pull-up resistor

Appendix 1 Check Sheet

The following check sheet was created based on items which had been the source of problems in the past. We recommend you refer to the check sheet when troubleshooting.

Checks regarding register initial settings

☐ Has the initial setting been made in the interrupt stack pointer (ISP) at the top of the program?

☐ Has the initial setting been made in the user stack pointer (USP)? (Only if using the USP)

☐ Does the USP overlap the ISP area? (Only if using the USP)

☐ Is interrupt enabled after setting the ISP and USP?

☐ Is the top address of the variable interrupt vector table set in the interrupt table register (INTB)?

☐ Is interrupt enabled after setting the INTB?

☐ Has the initial setting been made in the frame base register (FB)? (Only if using the FB)

☐ Has the initial setting been made in the stack base register (SB)? (Only if using the SB)

Checks regarding the internal memory

☐ Does the RAM capacity used in the program exceed the RAM capacity of the microcomputer?

☐ Does the ROM capacity used in the program exceed the ROM capacity of the microcomputer?

Checks regarding the protect register

☐ Is writing enabled in the protect register (address $000A_{16}$) before writing in the system clock control register (addresses 0006_{16} and 0007_{16})?

☐ Is writing enabled in the protect register before writing in the processor mode register (addresses 0004_{16} and 0005_{16})?

☐ Is writing enabled in the protect register before writing in the port P4 direction register (address $03EA_{16}$)?

☐ Is writing effectuated in the port P4 direction register by the next instruction after writing is enabled in the protect register?

☐ Does not an interrupt generate between the instruction writing is enabled in the protect register and the instruction writing in the port P4 direction register?

Checks regarding the timer

☐ Is the timer started after a value is set in the timer register?

Checks regarding low power consumption

☐ In the low power consumption mode, does not current flow from Vref when the Vref connection bit (bit 5 in address $03D7_{16}$) is set?

☐ Is not voltage level of port floating in the low power consumption mode?

Checks regarding Interrupt

☐ When rewrite the interrupt register, do so at a point that does not generate the interruput request?

Checks regarding low voltage

☐ When using at low voltage, have you checked recommended operating conditions and changed the wait bit (address 0005_{16}, bit 7) to “1”?

Checks regarding A-D converter

☐ Have you selected other than f_{AD} (no dividing) for $ø_{AD}$ when using the A-D converter at V_{CC} = 2.7 - 4.0V?
☐ Have you selected no sample & hold function when using the A-D converter at V_{CC} = 2.7 - 4.0V?
☐ Have you selected 8-bit mode when using the A-D converter at V_{CC} = 2.7 - 4.0V?

Appendix 2 Hexadecimal instruction CODE table

	D7 to D4	0000	0001	0010	0011	0100	0101	0110	0111
D3 to D0		0	1	2	3	4	5	6	7
0000	0	BRK	AND.B:S R0H,R0L	ADD.B:S R0H,R0L	MOV.B:S R0H,A0	BCLR:S 0,11[SB]	BNOT:S 0,11[SB]	JMP.S label	MULU.B src,dest
0001	1	MOV.B:S R0L,dsp:8[SB]	AND.B:S dsp:8[SB],R0L	ADD.B:S dsp:8[SB],R0L	MOV.B:S dsp:8[SB],A0	BCLR:S 1,11[SB]	BNOT:S 1,11[SB]	JMP.S label	MULU.W src,dest
0010	2	MOV.B:S R0L,dsp:8[FB]	AND.B:S dsp:8[FB],R0L	ADD.B:S dsp:8[FB],R0L	MOV.B:S dsp:8[FB],A0	BCLR:S 2,11[SB]	BNOT:S 2,11[SB]	JMP.S label	MOV.B:G src,dest
0011	3	MOV.B:S R0L,abs16	AND.B:S abs16,R0L	ADD.B:S abs16,R0L	MOV.B:S abs16,A0	BCLR:S 3,11[SB]	BNOT:S 3,11[SB]	JMP.S label	MOV.W:G src,dest
0100	4	NOP	AND.B:S R0L,R0H	ADD.B:S R0L,R0H	MOV.B:S R0Çk,A1	BCLR:S 4,11[SB]	BNOT:S 4,11[SB]	JMP.S label	CODE_74
0101	5	MOV.B:S R0H,dsp:8[SB]	AND.B:S dsp:8[SB],R0H	ADD.B:S dsp:8[SB],R0H	MOV.B:S dsp:8[SB],A1	BCLR:S 5,11[SB]	BNOT:S 5,11[SB]	JMP.S label	CODE_75
0110	6	MOV.B:S R0H,dsp:8[FB]	AND.B:S dsp:8[FB],R0H	ADD.B:S dsp:8[FB],R0H	MOV.B:S dsp:8[FB],A1	BCLR:S 6,11[SB]	BNOT:S 6,11[SB]	JMP.S label	CODE_76
0111	7	MOV.B:S R0H,abs16	AND.B:S abs16,R0H	ADD.B:S abs16,R0H	MOV.B:S abs16,A1	BCLR:S 7,11[SB]	BNOT:S 7,11[SB]	JMP.S label	CODE_77
1000	8	MOV.B:S R0H,R0L	OR.B:S R0H,R0L	SUB.B:S R0H,R0L	CMP.B:S R0H,R0L	BSET:S 0,11[SB]	BTST:S 0,11[SB]	JGEU/C label	MUL.B src,dest
1001	9	MOV.B:S dsp:8[SB],R0L	OR.B:S dsp:8[SB],R0L	SUB.B:S dsp:8[SB],R0L	CMP.B:S dsp:8[SB],R0L	BSET:S 1,11[SB]	BTST:S 1,11[SB]	JGTU label	MUL.W src,dest
1010	A	MOV.B:S dsp:8[FB],R0L	OR.B:S dsp:8[FB],R0L	SUB.B:S dsp:8[FB],R0L	CMP.B:S dsp:8[FB],R0L	BSET:S 2,11[SB]	BTST:S 2,11[SB]	JEQ/Z label	CODE_7A
1011	B	MOV.B:S abs16,R0L	OR.B:S abs16,R0L	SUB.B:S abs16,R0L	CMP.B:S abs16,R0L	BSET:S 3,11[SB]	BTST:S 3,11[SB]	JN label	CODE_7B
1100	C	MOV.B:S R0L,R0H	OR.B:S R0L,R0H	SUB.B:S R0L,R0H	CMP.B:S R0L,R0H	BSET:S 4,11[SB]	BTST:S 4,11[SB]	JLTU/NC label	CODE_7C
1101	D	MOV.B:S dsp:8[SB],R0H	OR.B:S dsp:8[SB],R0H	SUB.B:S dsp:8[SB],R0H	CMP.B:S dsp:8[SB],R0H	BSET:S 5,11[SB]	BTST:S 5,11[SB]	JLEU label	CODE_7D
1110	E	MOV.B:S dsp:8[FB],R0H	OR.B:S dsp:8[FB],R0H	SUB.B:S dsp:8[FB],R0H	CMP.B:S dsp:8[FB],R0H	BSET:S 6,11[SB]	BTST:S 6,11[SB]	JNE/JNZ label	CODE_7E
1111	F	MOV.B:S abs16,R0H	OR.B:S abs16,R0H	SUB.B:S abs16,R0H	CMP.B:S abs16,R0H	BSET:S 7,11[SB]	BTST:S 7,11[SB]	JPZ label	

The next instruction is arranged in each CODE.

CODE_74:STE,MOV,PUSH,NEG,ROT,NOT,LDE,POP,SHL,SHA
CODE_75:STE,MOV,PUSH,NEG,ROT,NOT,LDE,POP,SHL,SHA
CODE_76:TST,XOR,AND,OR,ADD,SUB,ADC,SBB,CMP,DIVX,ROLC,RORC,DIVU,DIV,ADCF,ABS
CODE_77:TST,XOR,AND,OR,ADD,SUB,ADC,SBB,CMP,DIVX,ROLC,RORC,DIVU,DIV,ADCF,ABS
CODE_7A:XCHG,LDC
CODE_7B:XCHG,STC
CODE_7C:MOV *Dir* ,MULU,MUL,EXTS,STC,DIVU,DIV,PUSH,DIVX,DADD,DSUB,DADC,DSBB,SMOVF,SMOVB,SSTR,ADD,LDCTX,RMPA,ENTER
CODE_7D:JMPI,JSRI,MULU,MUL,PUSHA,LDIPL,ADD,J *Cnd* ,BM*Cnd* ,DIVU,DIV,PUSH,DIVX,DADD,DSUB,DADC,DSBB,SMOVF,SMOVB,SSTR, STCTX,RMPA,EXITD,WAIT
CODE_7E:BTSTC,BM *Cnd* ,BNTST,BAND,BNAND,BOR,BNOR,BCLR,BSET,BNOT,BTST,BXOR,BNXOR
CODE_EB:SHL,FSET,FCLR,MOVA,LDC,SHA,PUSHC,POPC,INT

Appendix 2 Hexadecimal instruction CODE table

	D7 to D4	1000	1001	1010	1011	1100	1101	1110	1111
D3 to D0		8	9	A	B	C	D	E	F
0000	0	TST.B src,dest	AND.B:G src,dest	ADD.B:G src,dest	ADC.B src,dest	CMP.B:G src,dest	CMP.B:Q #IMM,dest	ROT.B #IMM,dest	SHA.B #IMM,dest
0001	1	TST.W src,dest	AND.W:G src,dest	ADD.W:G src,dest	ADC.w src,dest	CMP.W:G src,dest	CMP.W:Q #IMM,dest	ROT.W #IMM,dest	SHA.W #IMM,dest
0010	2	PUSH.B:S R0L	POP.B:S R0L	MOV.W:S #IMM,A0	INC.W A0	PUSH.W:S A0	POP.W:S A0	MOV.B:S #IMM,A0	DEC.W A0
0011	3	ADD.B:S #IMM8,R0H	AND.B:S #IMM8,R0H	INC.B R0H	MOV.B:Z #0,R0H	MOV.B:S #IMM8,R0H	STNZ #IMM8,R0H	CMP.B:S #IMM8,R0H	RTS
0100	4	ADD.B:S #IMM8,R0L	AND.B:S #IMM8,R0L	INC.B R0L	MOV.B:Z #0,R0L	MOV.B:S #IMM8,R0L	STNZ #IMM8,R0L	CMP.B:S #IMM8,R0L	JMP.W label
0101	5	ADD.B:S #IMM8,dsp:8[SB]	AND.B:S #IMM8,dsp:8[SB]	INC.B dsp:8[SB]	MOV.B:Z #0,dsp:8[SB]	MOV.B:S #IMM8,dsp:8[SB]	STNZ #IMM8,dsp:8[SB]	CMP.B:S #IMM8,dsp:8[SB]	JSR.W label
0110	6	ADD.B:S #IMM8,dsp:8[FB]	AND.B:S #IMM8,dsp:8[FB]	INC.B dsp:8[FB]	MOV.B:Z #0,dsp:8[FB]	MOV.B:S #IMM8,dsp:8[FB]	STNZ #IMM8,dsp:8[FB]	CMP.B:S #IMM8,dsp:8[FB]	INTO
0111	7	ADD.B:S #IMM8,abs16	AND.B:S #IMM8,abs16	INC.B abs16	MOV.B:Z #0,abs16	MOV.B:S #IMM8,abs16	STNZ #IMM8,abs16	CMP.B:S #IMM8,abs16	
1000	8	XOR.B src,dest	OR.B:G src,dest	SUB.B:G src,dest	SBB.B src,dest	ADD.B:Q #IMM,dest	MOV.B:Q #IMM,dest	SHL.B #IMM,dest	ADJNZ.B #IMM,dest,label
1001	9	XOR.W src,dest	OR.W:G src,dest	SUB.W:G src,dest	SBB.W src,dest	ADD.W:Q #IMM,dest	MOV.W:Q #IMM,dest	SHL.W #IMM,dest	ADJNZ.W #IMM,dest,label
1010	A	PUSH.B:S R0H	POP.B:S R0H	MOV.W:S #IMM,A1	INC.W A1	PUSH.W:S A1	POP.W:S A1	MOV.B:S #IMM,A1	DEC.W A1
1011	B	SUB.B:S #IMM8,R0H	OR.B:S #IMM8,R0H	DEC.B R0H	NOT.B:S R0H	STZ #IMM8,R0H	STZX #IMM8,#IMM8,R0H	CODE_EB	REIT
1100	C	SUB.B:S #IMM8,R0L	OR.B:S #IMM8,R0L	DEC.B R0L	NOT.B:S R0L	STZ #IMM8,R0L	STZX #IMM8,#IMM8,R0L	PUSHM src	JMP.A label
1101	D	SUB.B:S #IMM8,dsp:8[SB]	OR.B:S #IMM8,dsp:8[SB]	DEC.B dsp:8[SB]	NOT.B:S dsp:8[SB]	STZ #IMM8,dsp:8[SB]	STZX #IMM8,#IMM8,dsp:8[SB]	POPM dest	JSR.A label
1110	E	SUB.B:S #IMM8,dsp:8[FB]	OR.B:S #IMM8,dsp:8[FB]	DEC.B dsp:8[FB]	NOT.B:S dsp:8[FB]	STZ #IMM8,dsp:8[FB]	STZX #IMM8,#IMM8,dsp:8[FB]	JMPS #IMM8	JMP.B label
1111	F	SUB.B:S #IMM8,abs16	OR.B:S #IMM8,abs16	DEC.B abs16	NOT.B:S abs16	STZ #IMM8,abs16	STZX #IMM8,#IMM8,abs16	JSRS #IMM8	UND

MITSUBISHI Single-Chip Microcomputer
User's Manual
M30201 Group

Feb. First Edition 1999
REV.A
Editioned by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Kitaitami Works

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.
©1999 MITSUBISHI ELECTRIC CORPORATION