

To all our customers

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

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

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

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

MITSUBISHI 8-BIT SINGLE-CHIP MICROCOMPUTER
740 FAMILY / 7600 SERIES

7630
Group

User's Manual

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.

REVISION DESCRIPTION LIST

7630 GROUP USER'S MANUAL

REVISION	DATE	PAGE		MODIFICATIONS
		NEW	OLD	
SEP-98	09-98			First Edition
JAN-99	01-99	1-13 1-14 1-21 1-28 1-38 1-41 3-4	1-13 1-14 1-21 1-28 1-38 1-41 3-4	"CAN controller" is replaced by "CAN module" in whole documents. Schematics (1) is modified. Schematics (8) and (11) are corrected. Replaced:"PUPDJ" with "PUP4J". Replaced:"URxD" with "SOUT" . Replaced:"URxD" with "SIN". Fig.41 is modified. Replaced:"FFFBH" with "FFFB16". Replaced:"FFFAH" with "FFFA16". Values changed:lih (35,113) to (20,200) and lih (-122,-70) to (-200,-20);typical values are removed.

Preface

This user's manual describes Mitsubishi's CMOS 8-bit microcomputers 7630 Group.

After reading this manual, the user should have a thorough knowledge of the functions and features of the 7630 Group, and should be able to fully utilize the product. The manual starts with specifications and ends with application examples.

For details of software, refer to the "SERIES MELPS 7600 <SOFTWARE> USER'S MANUAL."

For details of development support tools, refer to the "DEVELOPMENT SUPPORT TOOLS FOR MICROCOMPUTERS" data book.

BEFORE USING THIS USER'S MANUAL

This user's manual consists of the following three chapters. Refer to the chapter appropriate to your conditions, such as hardware design or software development. Chapter 3 also includes necessary information for systems development. Be sure to refer to this chapter.

1. Organization

● CHAPTER 1 HARDWARE

This chapter describes features of the microcomputer and operation of each peripheral function.

● CHAPTER 2 APPLICATION

This chapter describes usage and application examples of peripheral functions, based mainly on setting examples of related registers.

● CHAPTER 3 APPENDIX

This chapter includes necessary information for systems development using the microcomputer, electric characteristics, a list of registers, the masking confirmation (mask ROM version), and mark specifications which are to be submitted when ordering.

2. Structure of register

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

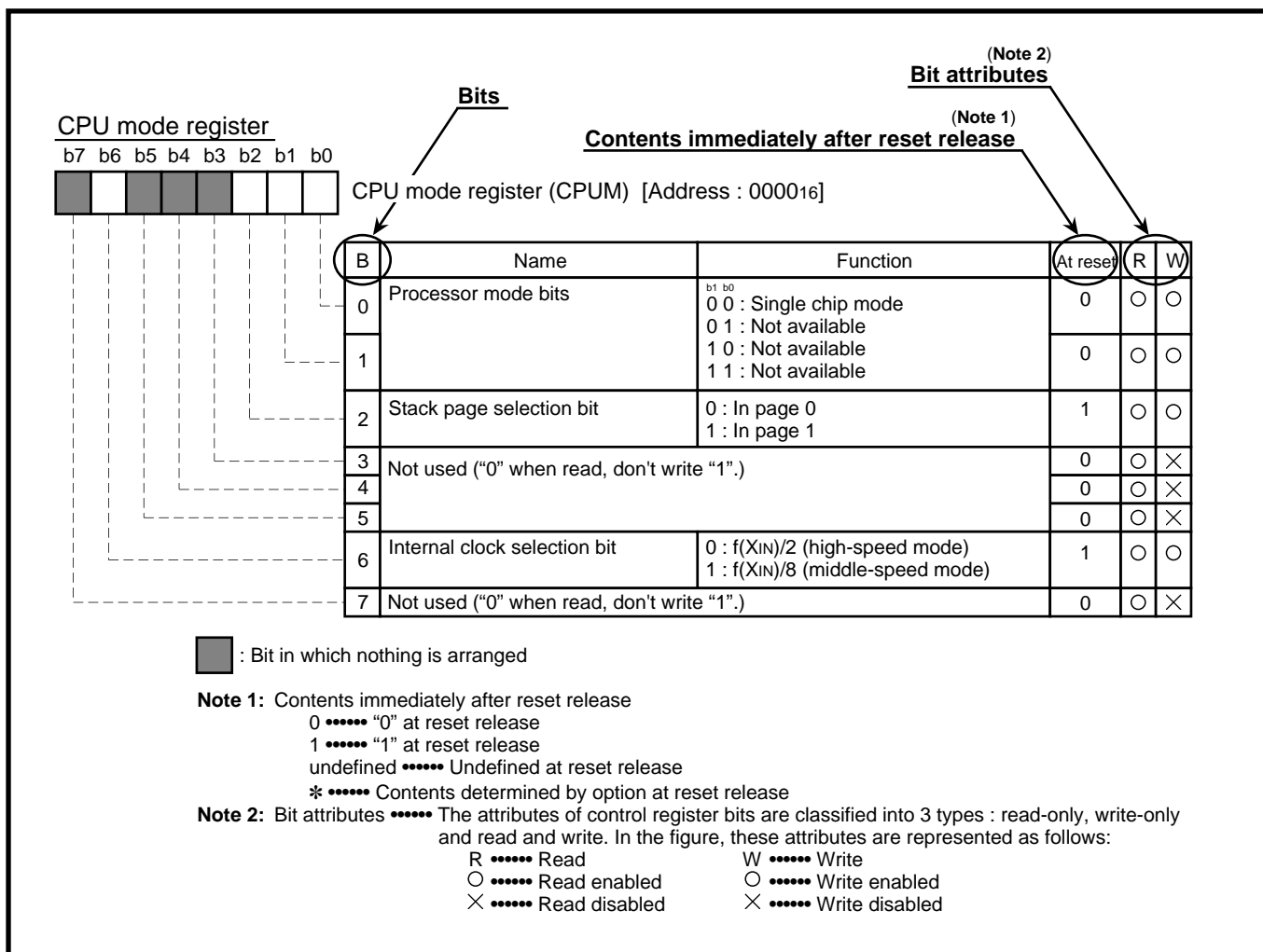


Table of contents

CHAPTER 1. HARDWARE

DESCRIPTION	1-2
FEATURES	1-2
APPLICATION	1-2
PIN CONFIGURATION	1-2
FUNCTIONAL BLOCK DIAGRAM	1-3
PIN DESCRIPTION	1-4
PART NUMBERING	1-5
GROUP EXPANSION	1-6
Memory Type	1-6
Memory Size	1-6
Package	1-6
FUNCTIONAL DESCRIPTION	1-7
Central Processing Unit (CPU)	1-7
Memory	1-11
I/O Ports	1-13
Interrupts	1-17
Key-on Wake-up	1-21
Timers	1-22
Serial I/Os	1-28
CAN Module	1-33
A-D Converter	1-38
Watchdog Timer	1-40
Reset Circuit	1-41
Clock Generating Circuit	1-43
Data Required for Mask Orders	1-44
Absolute Maximum Rating	1-45
Electrical Characteristics	1-46
A-D Converter Characteristics	1-47
Timing Requirements	1-48
Switching Characteristics	1-49
Timing Diagram	1-50

CHAPTER 2. APPLICATION

2.1 I/O ports	2-2
2.1.1 Memory map of I/O ports	2-2
2.1.2 Related registers	2-3
2.1.3 Overvoltage conditions at digital input ports	2-7
2.1.4 Handling examples of unused pins	2-8
2.2 Interrupts	2-9
2.2.1 Memory map of interrupt related registers	2-9
2.2.2 Related registers	2-10
2.2.3 Interrupt setting method	2-14
2.2.4 Key-on wake-up interrupt	2-16
2.3 Timers	2-18
2.3.1 Memory map of timer	2-18
2.3.2 Related registers	2-19
2.3.3 Timer application examples	2-22

Table of contents

2.4 Controller Area Network (CAN) module	2-34
2.4.1 Description	2-34
2.4.2 Special function register map	2-35
2.4.3 Related registers	2-37
2.4.4 Operational modes	2-48
2.4.5 Module initialization	2-50
2.4.6 Module reset	2-53
2.4.7 Acceptance filtering	2-54
2.4.8 Message reception	2-55
2.4.9 Message transmission	2-60
2.4.10 Abort transmission	2-62
2.4.11 CAN interrupts	2-65
2.4.12 Error condition	2-65
2.4.13 Wake-up via CAN	2-66
2.5 Serial I/O	2-67
2.5.1 Memory map of serial I/O	2-67
2.5.2 Related registers	2-68
2.5.3 Serial I/O connection examples	2-72
2.5.4 Setting of serial I/O transfer data format	2-74
2.5.5 Serial I/O application examples	2-75
2.6 A-D converter	2-90
2.6.1 Memory map of A-D conversion	2-90
2.6.2 Related registers	2-91
2.6.3 A-D conversion application example	2-92
2.6.4 Conversion time	2-93
2.6.5 Notes on use	2-93
2.7 Watchdog timer	2-94
2.7.1 Related register	2-94
2.7.2 Watchdog timer cycle	2-95
2.7.3 Watchdog timer procedure	2-95
2.8 Reset	2-96
2.9 Oscillation Circuit	2-97
2.9.1 Memory map of oscillation circuit related registers	2-97
2.9.2 Related registers	2-98
2.9.3 Application examples	2-99
2.10 Development support tools (M37630T-RFS)	2-100
2.11 Built-in PROM version	2-101
2.11.1 Product expansion	2-101
2.11.2 Pin configuration	2-102
2.11.3 Programming adapter	2-103
2.11.4 Notes on use	2-104

CHAPTER 3. APPENDIX

3.1 Electrical characteristics	3-2
3.1.1 Absolute maximum ratings	3-2
3.1.2 Recommended operating conditions	3-3
3.1.3 Electrical characteristics	3-4
3.1.4 A-D converter characteristics	3-5
3.1.5 Timing requirements	3-5
3.1.6 Switching characteristics	3-6
3.2 Standard characteristics	3-8
3.2.1 Power source current standard characteristics	3-8

3.2.2 Output current standard characteristics	3-10
3.2.3 Input current standard characteristics	3-11
3.2.4 A-D conversion standard characteristics	3-12
3.3 Notes on use	3-13
3.3.1 Notes on interrupts	3-13
3.3.2 Notes on A-D converter	3-13
3.3.3 Notes on RESET pin	3-14
3.3.4 Notes on input and output pins	3-14
3.3.5 Notes on programming	3-15
3.4 Countermeasures against noise	3-17
3.4.1 Shortest wiring length	3-17
3.4.2 Connection of a bypass capacitor across the Vss line and the Vcc line	3-18
3.4.3 Wiring to analog input pins	3-18
3.4.4 Consideration for oscillator	3-19
3.4.5 Setup for I/O ports	3-20
3.4.6 Providing of watchdog timer function by software	3-21
3.5 List of registers	3-22
3.6 Mask ROM ordering method	3-49
3.7 Mark specification form	3-51
3.8 Package outline	3-52
3.9 List of instruction codes	3-53
3.10 Machine instructions	3-54
3.11 SFR memory map	3-64
3.12 Pin configuration	3-65

List of figures

CHAPTER 1. HARDWARE

Fig. 1 Pin configuration of M37630M4T-XXXXFP	1-2
Fig. 2 Functional block diagram	1-3
Fig. 3 Part numbering	1-5
Fig. 4 Memory expansion plan	1-6
Fig. 5 740 Family CPU register structure	1-7
Fig. 6 Register push and pop at interrupt generation and subroutine call	1-8
Fig. 7 Structure of CPU mode register	1-10
Fig. 8 Memory map diagram	1-11
Fig. 9 Memory map of special register (SFR)	1-12
Fig. 10 Structure of Port- and Port direction registers	1-13
Fig. 11 Structure of Port I/Os (1)	1-13
Fig. 12 Structure of Port I/Os (2)	1-14
Fig. 13 Structure of Port pull-up/down control registers	1-15
Fig. 14 Structure of Polarity control register	1-15
Fig. 15 Interrupt control	1-19
Fig. 16 Structure of Interrupt polarity selection register	1-19
Fig. 17 Structure of Interrupt request and control registers A, B and C	1-20
Fig. 18 Block diagram of key-on wake-up circuit	1-21
Fig. 19 Block diagram of timers X and Y	1-22
Fig. 20 Structure of Timer X mode register	1-23
Fig. 21 Structure of Timer Y mode register	1-24
Fig. 22 Timer X bi-phase counter mode operation	1-25
Fig. 23 Block diagram of timers 1 to 3	1-26
Fig. 24 Timer 123 mode register configuration	1-27
Fig. 25 Block diagram of clock synchronous SI/O	1-28
Fig. 26 Timing of clock synchronous SI/O function	1-28
Fig. 27 Structure of Serial I/O control register	1-29
Fig. 28 Block diagram of UART	1-30
Fig. 29 Structure of UART mode register	1-31
Fig. 30 Structure of UART control register	1-31
Fig. 31 Structure of UART status register	1-32
Fig. 32 Bit time of CAN module	1-33
Fig. 33 Block diagram of CAN module	1-33
Fig. 34 Structure of CAN transmit control register	1-34
Fig. 35 Structure of CAN receive control register	1-34
Fig. 36 Structure of CAN transmit abort request register	1-35
Fig. 37 Structure of CAN bus timing control register 1	1-35
Fig. 38 Structure of CAN bus timing control register 2	1-36
Fig. 39 Structure of CAN mask and code registers	1-36
Fig. 40 Structure of CAN transmission and reception buffer registers	1-37
Fig. 41 Block diagram of A-D converter	1-38
Fig. 42 Structure of A-D control register	1-39
Fig. 43 Block diagram of watchdog timer	1-40
Fig. 44 Structure of watchdog timer register	1-40
Fig. 45 Example of reset circuit	1-41
Fig. 46 Reset sequence	1-41
Fig. 47 Internal status of microcomputer after reset	1-42

Fig. 48 Ceramic resonator circuit	1-43
Fig. 49 Block diagram of clock generating circuit	1-43
Fig. 50 Programming and testing of One Time PROM version	1-44
Fig. 51 Circuit for measuring output switching characteristics	1-49
Fig. 52 Timing diagram	1-50

CHAPTER 2. APPLICATION

Fig. 2.1.1 Memory map of I/O port related registers	2-2
Fig. 2.1.2 Structure of Port Pi register (i = 0, 1, 2, 3, 4)	2-3
Fig. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)	2-3
Fig. 2.1.4 Structure of Port Pi pull-up register (i = 0, 2)	2-4
Fig. 2.1.5 Structure of Port P1 pull-up register	2-4
Fig. 2.1.6 Structure of Port P3 pull-up control register	2-5
Fig. 2.1.7 Structure of Port P4 pull-up/down control register	2-5
Fig. 2.1.8 Structure of Polarity control register	2-6
Fig. 2.1.9 External circuit example applying overvoltage to digital inputs	2-7
Fig. 2.2.1 Memory map of interrupt related registers	2-9
Fig. 2.2.2 Structure of Interrupt request register A	2-10
Fig. 2.2.3 Structure of Interrupt request register B	2-10
Fig. 2.2.4 Structure of Interrupt request register C	2-11
Fig. 2.2.5 Structure of Interrupt control register A	2-11
Fig. 2.2.6 Structure of Interrupt control register B	2-12
Fig. 2.2.7 Structure of Interrupt control register C	2-12
Fig. 2.2.8 Structure of Interrupt polarity selection register	2-13
Fig. 2.2.9 Structure of Polarity control register	2-13
Fig. 2.2.10 Interrupt setting method (1)	2-14
Fig. 2.2.11 Interrupt setting method (2)	2-15
Fig. 2.2.12 Setting method for registers related to key-on wake-up interrupt (1)	2-16
Fig. 2.2.13 Setting method for registers related to key-on wake-up interrupt (2)	2-17
Fig. 2.3.1 Memory map of timer related registers	2-18
Fig. 2.3.2 Structure of Timer 1, Timer 3	2-19
Fig. 2.3.3 Structure of Timer 2	2-19
Fig. 2.3.4 Structure of Timer 123 mode register	2-20
Fig. 2.3.5 Structure of Timer XL, Timer XH, Timer YL, Timer YH	2-20
Fig. 2.3.6 Structure of Timer X mode register	2-21
Fig. 2.3.7 Structure of Timer Y mode register	2-21
Fig. 2.3.8 Timers connection and division ratios [Clock function]	2-23
Fig. 2.3.9 Setting of related registers [Clock function]	2-24
Fig. 2.3.10 Control procedure [Clock function]	2-25
Fig. 2.3.11 A method for judging if input pulse exists	2-26
Fig. 2.3.12 Setting of related registers [Measurement of frequency] (1)	2-27
Fig. 2.3.13 Setting of related registers [Measurement of frequency] (2)	2-28
Fig. 2.3.14 Control procedure [Measurement of frequency]	2-29
Fig. 2.3.15 Timer connection and division ratios [Measurement of pulse width]	2-30
Fig. 2.3.16 Setting of related registers [Measurement of pulse width]	2-31
Fig. 2.3.17 Control procedure [Measurement of pulse width] (1)	2-32
Fig. 2.3.18 Control procedure [Measurement of pulse width] (2)	2-33
Fig. 2.4.1 Block diagram of CAN module	2-35
Fig. 2.4.2 Memory map of CAN related registers	2-36
Fig. 2.4.3 Structure of CAN transmit control register	2-37
Fig. 2.4.4 Structure of CAN bus timing control register 1	2-37
Fig. 2.4.5 Structure of CAN bus timing control register 2	2-38

List of figures

Fig. 2.4.6 Structure of CAN acceptance code register 0	2-38
Fig. 2.4.7 Structure of CAN acceptance code register 1	2-39
Fig. 2.4.8 Structure of CAN acceptance code register 2	2-39
Fig. 2.4.9 Structure of CAN acceptance code register 3	2-40
Fig. 2.4.10 Structure of CAN acceptance code register 4	2-40
Fig. 2.4.11 Structure of CAN acceptance mask register 0	2-41
Fig. 2.4.12 Structure of CAN acceptance mask register 1	2-41
Fig. 2.4.13 Structure of CAN acceptance mask register 2	2-42
Fig. 2.4.14 Structure of CAN acceptance mask register 3	2-42
Fig. 2.4.15 Structure of CAN acceptance mask register 4	2-43
Fig. 2.4.16 Structure of CAN receive control register	2-43
Fig. 2.4.17 Structure of CAN transmit abort register	2-44
Fig. 2.4.18 Structure of CAN transmit/receive buffer registers 0	2-44
Fig. 2.4.19 Structure of CAN transmit/receive buffer registers 1	2-45
Fig. 2.4.20 Structure of CAN transmit/receive buffer registers 2	2-45
Fig. 2.4.21 Structure of CAN transmit/receive buffer registers 3	2-46
Fig. 2.4.22 Structure of CAN transmit/receive buffer registers 4	2-46
Fig. 2.4.23 Structure of CAN transmit/receive buffer registers 5	2-47
Fig. 2.4.24 Structure of CAN transmit/receive buffer registers 6 to D	2-47
Fig. 2.4.25 Transitions between operational modes	2-48
Fig. 2.4.26 Transitions among module sub-modes	2-49
Fig. 2.4.27 Segmentation of bit-time	2-50
Fig. 2.4.28 Module initialization sequence	2-52
Fig. 2.4.29 Module reset sequence	2-53
Fig. 2.4.30 Structure of acceptance mask/code registers	2-54
Fig. 2.4.31 Acceptance filter logic	2-55
Fig. 2.4.32 Receive buffer handling	2-56
Fig. 2.4.33 Flowchart of the receive process	2-57
Fig. 2.4.34 Receive sequence timing	2-58
Fig. 2.4.35 Receive sequence timing (overrun condition)	2-59
Fig. 2.4.36 Transmit buffer organization	2-61
Fig. 2.4.37 Transmit sequence timing (arbitration win)	2-62
Fig. 2.4.38 Flowchart of transmit process	2-64
Fig. 2.4.39 Error state diagram	2-65
Fig. 2.5.1 Memory map of serial I/O related registers	2-67
Fig. 2.5.2 Structure of Serial I/O shift register	2-68
Fig. 2.5.3 Structure of Serial I/O control register	2-68
Fig. 2.5.4 Structure of UART mode register	2-69
Fig. 2.5.5 Structure of UART baud rate generator	2-69
Fig. 2.5.6 Structure of UART control register	2-70
Fig. 2.5.7 Structure of UART status register	2-70
Fig. 2.5.8 Structure of UART transmit buffer register 1, 2	2-71
Fig. 2.5.9 Structure of UART receive buffer register 1, 2	2-71
Fig. 2.5.10 Serial I/O connection examples (1)	2-72
Fig. 2.5.11 Serial I/O connection examples (2)	2-73
Fig. 2.5.12 Setting of serial I/O transfer data format	2-74
Fig. 2.5.13 Connection diagram [Output of serial data]	2-75
Fig. 2.5.14 Timing chart [Output of serial data]	2-75
Fig. 2.5.15 Setting of serial I/O related registers [Output of serial data]	2-76
Fig. 2.5.16 Setting of serial I/O transmission data [Output of serial data]	2-77
Fig. 2.5.17 Control procedure of clock synchronous serial I/O [Output of serial data] ...	2-78
Fig. 2.5.18 Connection diagram [Communication using UART]	2-79
Fig. 2.5.19 Timing chart [Communication using UART]	2-79

Fig. 2.5.20	Setting of related registers on transmitting side [Communication using UART]	2-81
Fig. 2.5.21	Setting of related registers on receiving side [Communication using UART]	2-82
Fig. 2.5.22	Control procedure on transmitting side [Communication using UART]	2-83
Fig. 2.5.23	Control procedure on receiving side [Communication using UART]	2-84
Fig. 2.5.24	Connection diagram [Communication using UART]	2-85
Fig. 2.5.25	Timing chart [Communication using UART]	2-85
Fig. 2.5.26	Setting of related registers on transmitting side [Communication using UART]	2-86
Fig. 2.5.27	Setting of related registers on receiving side [Communication using UART]	2-87
Fig. 2.5.28	Control procedure on transmitting side [Communication using UART]	2-88
Fig. 2.5.29	Control procedure on receiving side [Communication using UART]	2-89
Fig. 2.6.1	Memory map of A-D conversion related registers	2-90
Fig. 2.6.2	Structure of A-D conversion register	2-91
Fig. 2.6.3	Structure of A-D control register	2-91
Fig. 2.6.4	Connection diagram [Measurement of analog signals]	2-92
Fig. 2.6.5	Setting of related registers [Measurement of analog signals]	2-92
Fig. 2.6.6	Control procedure [Measurement of analog signals]	2-93
Fig. 2.7.1	Structure of Watchdog timer register	2-94
Fig. 2.7.2	Set-up procedure of watchdog timer	2-95
Fig. 2.8.1	Example of Power on reset circuit	2-96
Fig. 2.8.2	RAM back-up system	2-96
Fig. 2.9.1	Memory map of oscillation circuit related registers	2-97
Fig. 2.9.2	Structure of CPU mode register	2-98
Fig. 2.9.3	Structure of Watchdog timer register	2-98
Fig. 2.9.4	Switching procedure to Stop mode	2-99
Fig. 2.9.5	Switching procedure to Wait mode	2-99
Fig. 2.10.1	Configuration example of using M37630T-RFS	2-100
Fig. 2.11.1	Pin configuration of 7630 group's built-in PROM versions	2-102
Fig. 2.11.2	Programming and testing of One Time PROM version	2-104

CHAPTER 3. APPENDIX

Fig. 3.1.1	Circuit for measuring output switching characteristics	3-6
Fig. 3.1.2	Timing diagram	3-7
Fig. 3.2.1	Icc-Vcc standard characteristics (in high-speed mode)	3-8
Fig. 3.2.2	Icc-Vcc standard characteristics (in middle-speed mode)	3-8
Fig. 3.2.3	Power source current standard characteristics (in high-speed mode)	3-9
Fig. 3.2.4	Power source current standard characteristics (in middle-speed mode)	3-9
Fig. 3.2.5	Output current standard characteristics (P-channel)	3-10
Fig. 3.2.6	Output current standard characteristics (N-channel)	3-10
Fig. 3.2.7	Pull-up transistor standard characteristics IIL-VI	3-11
Fig. 3.2.8	Pull-down transistor standard characteristics IIL-VI	3-11
Fig. 3.2.9	A-D conversion standard characteristics	3-12
Fig. 3.3.1	Stack memory contents after instruction execution	3-15
Fig. 3.3.2	Interrupt routine	3-16
Fig. 3.4.1	Wiring for the RESET input pin	3-17
Fig. 3.4.2	Wiring for clock I/O pins	3-17
Fig. 3.4.3	Wiring for the VPP pin of the One Time PROM and the EPROM version	3-18
Fig. 3.4.4	Bypass capacitor across the Vss line and the Vcc line	3-18
Fig. 3.4.5	Analog signal line and a resistor and a capacitor	3-18
Fig. 3.4.6	Wiring for a large current signal line	3-19
Fig. 3.4.7	Wiring to a signal line where potential levels change frequently	3-19
Fig. 3.4.8	Vss pattern on the underside of an oscillator	3-20
Fig. 3.4.9	Setup for I/O ports	3-20

List of figures

Fig. 3.4.10 Watchdog timer by software	3-21
Fig. 3.5.1 Structure of CPU mode register	3-22
Fig. 3.5.2 Structure of Interrupt request register A	3-22
Fig. 3.5.3 Structure of Interrupt request register B	3-23
Fig. 3.5.4 Structure of Interrupt request register C	3-23
Fig. 3.5.5 Structure of Interrupt control register A	3-24
Fig. 3.5.6 Structure of Interrupt control register B	3-24
Fig. 3.5.7 Structure of Interrupt control register C	3-25
Fig. 3.5.8 Structure of Port Pi register (i = 0, 1, 2, 3, 4)	3-25
Fig. 3.5.9 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)	3-26
Fig. 3.5.10 Structure of Serial I/O shift register	3-26
Fig. 3.5.11 Structure of Serial I/O control register	3-27
Fig. 3.5.12 Structure of A-D conversion register	3-27
Fig. 3.5.13 Structure of A-D control register	3-28
Fig. 3.5.14 Structure of Timer 1, Timer 3	3-28
Fig. 3.5.15 Structure of Timer 2	3-29
Fig. 3.5.16 Structure of Timer 123 mode register	3-29
Fig. 3.5.17 Structure of Timer XL, Timer XH, Timer YL, Timer YH	3-30
Fig. 3.5.18 Structure of Timer X mode register	3-30
Fig. 3.5.19 Structure of Timer Y mode register	3-31
Fig. 3.5.20 Structure of UART mode register	3-31
Fig. 3.5.21 Structure of UART baud rate generator	3-32
Fig. 3.5.22 Structure of UART control register	3-32
Fig. 3.5.23 Structure of UART status register	3-33
Fig. 3.5.24 Structure of UART transmit buffer register 1, 2	3-33
Fig. 3.5.25 Structure of UART receive buffer register 1, 2	3-34
Fig. 3.5.26 Structure of Port Pi pull-up control register (i = 0, 2)	3-34
Fig. 3.5.27 Structure of Port P1 pull-up control register	3-35
Fig. 3.5.28 Structure of Port P3 pull-up control register	3-35
Fig. 3.5.29 Structure of Port P4 pull-up/down control register	3-36
Fig. 3.5.30 Structure of Interrupt polarity selection register	3-36
Fig. 3.5.31 Structure of Watchdog timer register	3-37
Fig. 3.5.32 Structure of Polarity control register	3-37
Fig. 3.5.33 Structure of CAN transmit control register	3-38
Fig. 3.5.34 Structure of CAN bus timing control register 1	3-38
Fig. 3.5.35 Structure of CAN bus timing control register 2	3-39
Fig. 3.5.36 Structure of CAN acceptance code register 0	3-39
Fig. 3.5.37 Structure of CAN acceptance code register 1	3-40
Fig. 3.5.38 Structure of CAN acceptance code register 2	3-40
Fig. 3.5.39 Structure of CAN acceptance code register 3	3-41
Fig. 3.5.40 Structure of CAN acceptance code register 4	3-41
Fig. 3.5.41 Structure of CAN acceptance mask register 0	3-42
Fig. 3.5.42 Structure of CAN acceptance mask register 1	3-42
Fig. 3.5.43 Structure of CAN acceptance mask register 2	3-43
Fig. 3.5.44 Structure of CAN acceptance mask register 3	3-43
Fig. 3.5.45 Structure of CAN acceptance mask register 4	3-44
Fig. 3.5.46 Structure of CAN receive control register	3-44
Fig. 3.5.47 Structure of CAN transmit abort register	3-45
Fig. 3.5.48 Structure of CAN transmit/receive buffer registers 0	3-45
Fig. 3.5.49 Structure of CAN transmit/receive buffer registers 1	3-46
Fig. 3.5.50 Structure of CAN transmit/receive buffer registers 2	3-46
Fig. 3.5.51 Structure of CAN transmit/receive buffer registers 3	3-47
Fig. 3.5.52 Structure of CAN transmit/receive buffer registers 4	3-47

Fig. 3.5.53 Structure of CAN transmit/receive buffer registers 5 3-48
Fig. 3.5.54 Structure of CAN transmit/receive buffer registers 6 to D 3-48

List of tables

CHAPTER 1. HARDWARE

Table 1	Pin description	1-4
Table 2	List of supported products	1-6
Table 3	Push and pop instructions of accumulator of processor status register	1-8
Table 4	Set and clear instructions of each bit of processor status register	1-9
Table 5	Interrupt vector addresses and priority	1-18
Table 6	Timer X count direction in bi-phase counter mode	1-24
Table 7	Output control conditions	1-30
Table 8	Programming adapter name	1-44
Table 9	Absolute maximum ratings	1-45
Table 10	Recommended operating conditions	1-45
Table 11	Electrical characteristics	1-46
Table 12	A-D converter characteristic	1-47
Table 13	Timing requirements	1-48
Table 14	Switching characteristics	1-49

CHAPTER 2. APPLICATION

Table 2.1.1	Handling of unused pins	2-8
Table 2.5.1	Setting examples of Baud rate generator values and transfer bit rate values	2-80
Table 2.5.2	Clock divider selection for serial I/O	2-80
Table 2.7.1	Watchdog timer cycle	2-95
Table 2.11.1	7630 group's built-in PROM version supporting products	2-101
Table 2.11.2	Programming adapter	2-103
Table 2.11.3	Setting of programming adapter switch	2-103
Table 2.11.4	Setting of PROM programmer address	2-103

CHAPTER 3. APPENDIX

Table 3.1.1	Absolute maximum ratings	3-2
Table 3.1.2	Recommended operating conditions	3-3
Table 3.1.3	Electrical characteristics	3-4
Table 3.1.4	A-D converter characteristics	3-5
Table 3.1.5	Timing requirements	3-5
Table 3.1.6	Switching characteristics	3-6



CHAPTER 1

HARDWARE

DESCRIPTION
FEARURES
APPLICATION
FUNCTION BLOCK DIAGRAM
PIN DESCRIPTION
PART NUMBERING
GROUP EXPANSION
FUNCTIONAL DESCRIPTION

HARDWARE

DESCRIPTION

DESCRIPTION

The 7630 group is a single chip 8-bit microcomputer designed with CMOS silicon gate technology.

Being equipped with a CAN (Controller Area Network) module circuit, the microcomputer is suited to drive automotive equipments. The CAN module complies with CAN specification version 2.0, part B and allows priority-based message management.

In addition to the microcomputers simple instruction set, the ROM, RAM and I/O addresses are placed in the same memory map to enable easy programming.

The built-in ROM is available as mask ROM or One Time PROM. For development purposes, emulator- and EPROM-type microcomputers are available as well.

FEATURES

- Basic machine-language instructions 71
- Minimum instruction execution time (at 10 MHz oscillation frequency) 0.2 μ s
- Memory size
ROM 16252 bytes (M37630M4T-XXXFP)
RAM 512 bytes (M37630M4T-XXXFP)
- I/O ports
Programmable I/O ports 35
Input 1

- Interrupts 24 sources, 24 vectors
- Timers
16-bit Timers 2 channels
8-bit Timers 3 channels
- Serial I/Os
Clock synchronous 1 channel
UART 1 channel
- CAN module
(CAN specification version 2.0, part B) 1 channel
- A-D converter 8-bits x 8 channels
- Watchdog timer 1
- Clock Generating Circuit 1
Built-in with internal feedback resistor
- Power source voltage (at 10 MHz oscillation frequency) 4.0 to 5.5 V
- Power dissipation
In high-speed mode 55 mW
(at 8 MHz oscillation frequency, at 5 V power source voltage)
- Operating temperature range -40 to 85 °C
- Package 44QFP (44P6N-A)

APPLICATION

Automotive controls

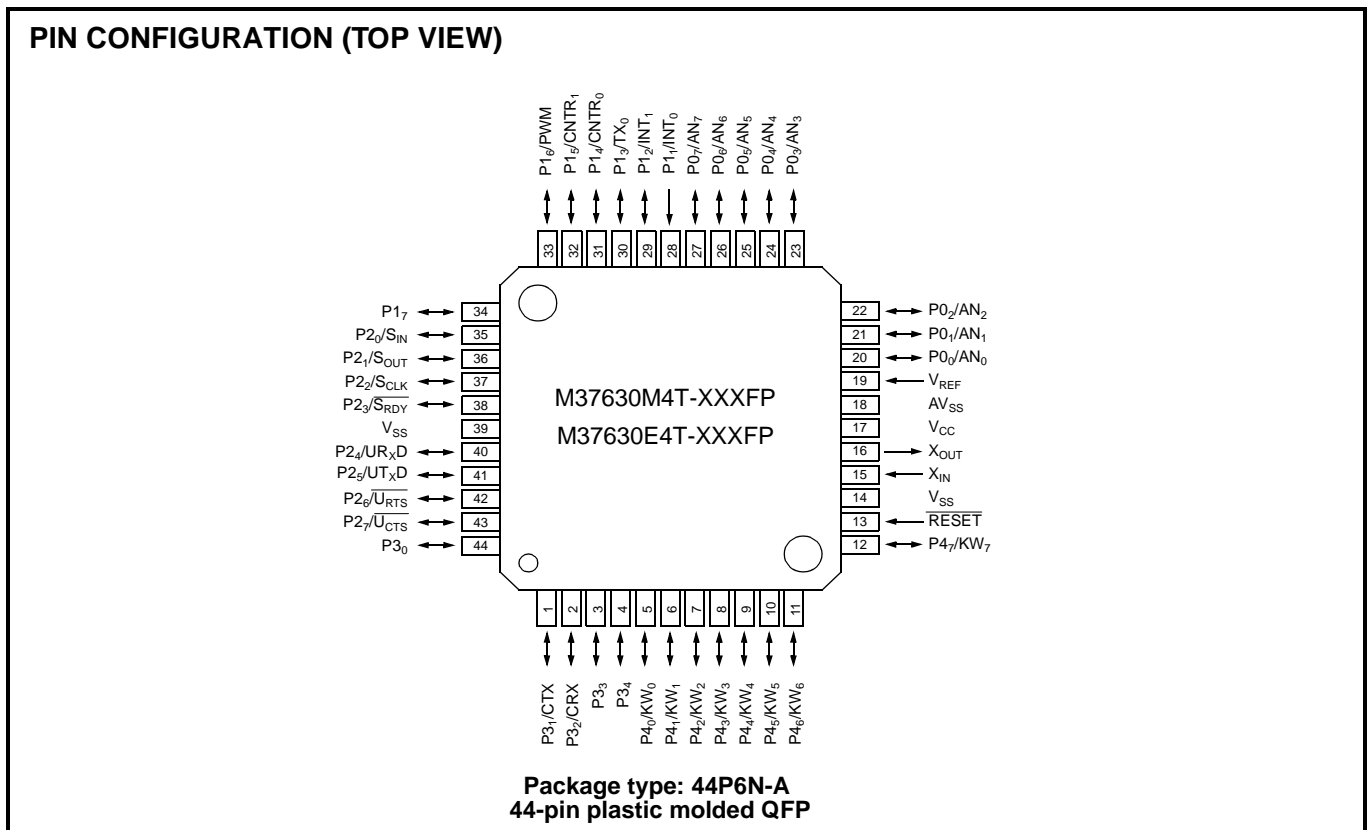


Fig. 1 Pin configuration of M37630M4T-XXXFP

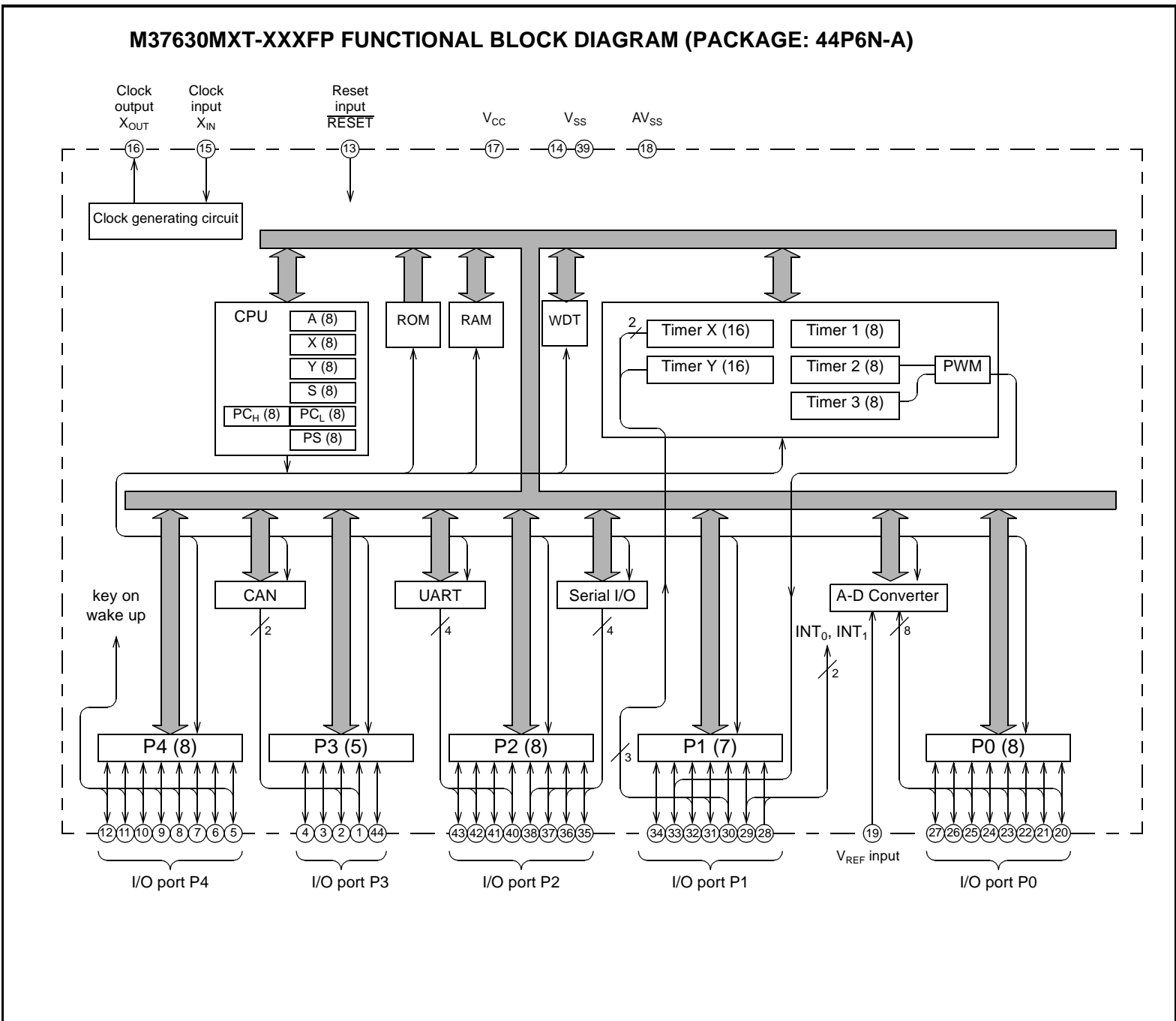


Fig. 2 Functional block diagram

HARDWARE

PIN DESCRIPTION

PIN DESCRIPTION

Table 1: Pin description

Pin	Name	Input/Output	Description
V _{CC} , V _{SS}	Power source voltage		Power supply pins; apply 4.0 to 5.5 V to V _{CC} and 0 V to V _{SS}
AV _{SS}	Analog power source voltage		Ground pin for A-D converter. Connect to V _{SS}
RESET	Reset input	Input	Reset pin. This pin must be kept at "L" level for more than 2 μs, to enter the reset state. If the crystal or ceramic resonator requires more time to stabilize, extend the "L" level period.
X _{IN}	Clock input	Input	Input and output pins of the internal clock generating circuit. Connect a ceramic or quartz-crystal resonator between the X _{IN} and X _{OUT} pins. When an external clock source is used, connect it to X _{IN} and leave X _{OUT} open.
X _{OUT}	Clock output	Output	
V _{REF}	Reference voltage input	Input	Reference voltage input pin for A-D converter
P0 ₀ /AN ₀ — P0 ₇ /AN ₇	I/O port P0	I/O	CMOS I/O ports or analog input ports
P1 ₁ /INT ₀	I/O port P1	Input	CMOS input port or external interrupt input port. The active edge (rising or falling) of external interrupts can be selected. This pin will be used as V _{PP} pin during PROM programming of One Time PROM Versions.
P1 ₂ /INT ₁			CMOS I/O port or external interrupt input port. The active edge (rising or falling) of external interrupts can be selected.
P1 ₃ /TX ₀		I/O	CMOS I/O port or input pin used in the bi-phase counter mode
P1 ₄ /CNTR ₀			CMOS I/O port or timer X input pin used for the event counter, pulse width measurement and bi-phase counter mode
P1 ₅ /CNTR ₁			CMOS I/O port or timer Y input pin used for the event counter, pulse width and pulse period measurement mode
P1 ₆ /PWM			CMOS I/O port or PWM output pin used in the PWM mode of timers 2 and 3
P1 ₇			CMOS I/O port
P2 ₀ /S _{IN} P2 ₁ /S _{OUT} P2 ₂ /S _{CLK} P2 ₃ /S _{RDY}	I/O port P2	I/O	CMOS I/O ports or clock synchronous serial I/O pins
P2 ₄ /UR _X D P2 ₅ /UT _X D P2 ₆ /U _{RTS} P2 ₇ /U _{CTS}			CMOS I/O ports or asynchronous serial I/O pins
P3 ₀	I/O port P3	I/O	CMOS I/O port
P3 ₁ /CTX			CMOS I/O port or CAN transmit data pin
P3 ₂ /CRX			CMOS I/O port or CAN receive data pin
P3 ₃ —P3 ₄			CMOS I/O port
P4 ₀ /KW ₀ — P4 ₇ /KW ₇	I/O port P4	I/O	CMOS I/O ports. These ports can be used for key-on wake-up when configured as inputs.

PART NUMBERING

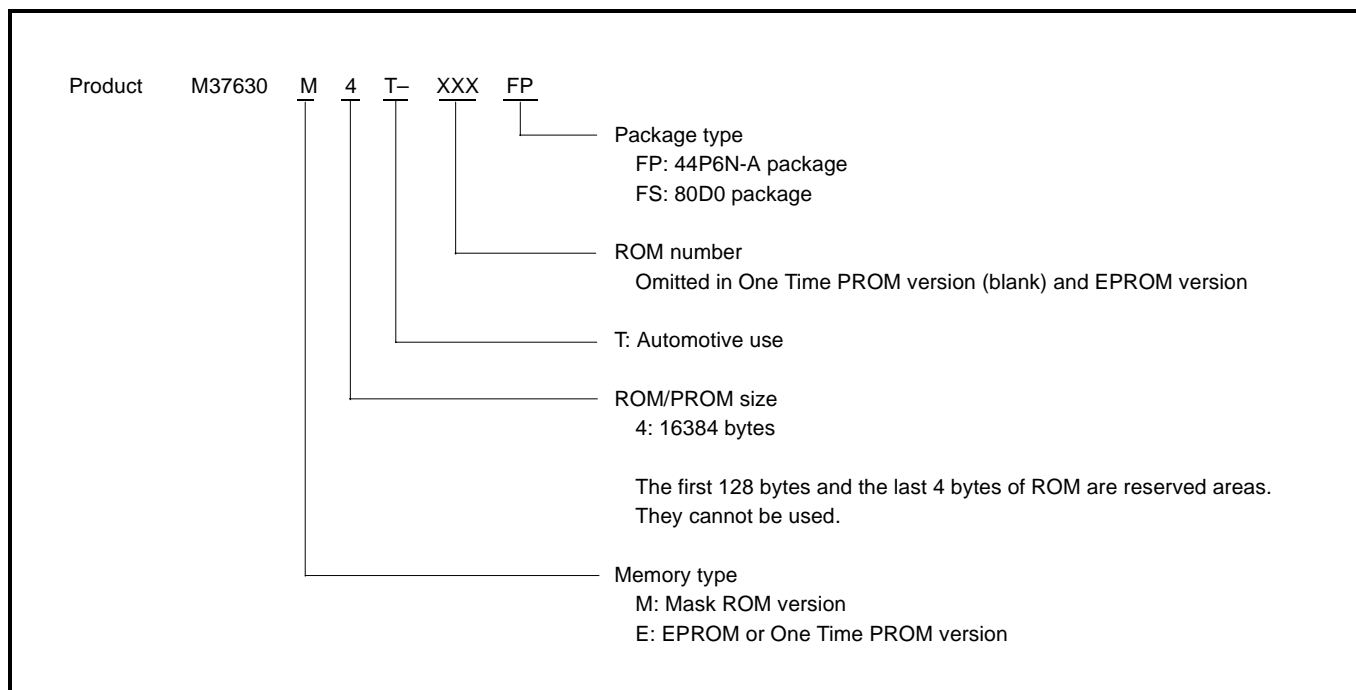


Fig. 3 Part numbering

HARDWARE

GROUP EXPANSION

GROUP EXPANSION

Mitsubishi plans to expand the 7630 group as follows:

Memory Type

Support mask ROM, One Time PROM and EPROM versions.

Memory Size

ROM/PROM size 16 Kbytes
 RAM size 512 bytes

Package

44P6N-A 0.8mm-pitch plastic molded QFP
 80D0 0.8mm-pitch ceramic LCC (EPROM version)

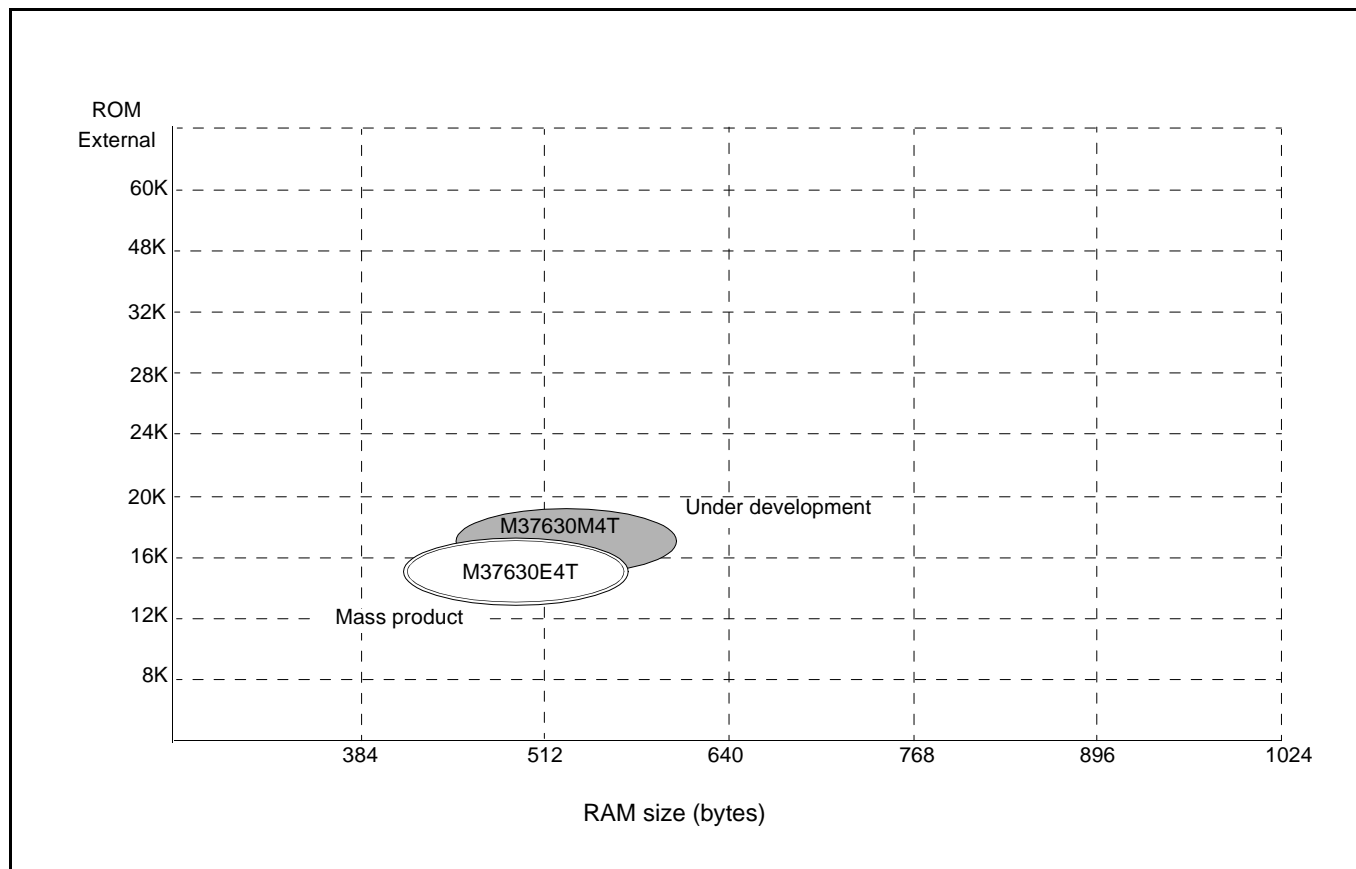


Fig. 4 Memory expansion plan

Currently supported products are listed below:

Table 2: List of supported products

As of March 1998

Product	(P)ROM size (bytes) ROM size for User ()	RAM size (bytes)	Package	Remarks
M37630M4T-XXXFP	16384 (16252)	512	44P6N-A	Mask ROM version
M37630E4T-XXXFP				One Time PROM version
M37630E4FP				One Time PROM version (blank)
M37630E4FS			80D0	EPROM version

FUNCTIONAL DESCRIPTION Central Processing Unit (CPU)

The 7630 group uses the standard 740 family instruction set. Refer to the table of 7600 series addressing modes and machine instructions or the 7600 series Software Manual for details on the instruction set.

Machine-resident 7600 series instructions are as follows:

The MUL, DIV, WIT and STP instruction can be used.

The central processing unit (CPU) has the six registers.

Accumulator (A)

The accumulator is an 8-bit register. Data operations such as data transfer, etc., are executed mainly through the accumulator.

Index register X (X), Index register Y (Y)

Both index register X and index register Y are 8-bit registers. In the index addressing modes, the value of the OPERAND is added to the contents of register X or register Y and specifies the real address.

When the T flag in the processor status register is set to "1", the value contained in index register X becomes the address for the second OPERAND.

Stack pointer (S)

The stack pointer is an 8-bit register used during sub-routine calls and interrupts. The stack is used to store the current address data and processor status when branching to subroutines or interrupt routines.

The lower eight bits of the stack address are determined by the contents of the stack pointer. The upper eight bits of the stack address are determined by the Stack Page Selection Bit. If the Stack Page Selection Bit is "0", then the RAM in the zero page is used as the stack area. If the Stack Page Selection Bit is "1", then RAM in page 1 is used as the stack area.

The Stack Page Selection Bit is located in the SFR area in the zero page. Note that the initial value of the Stack Page Selection Bit varies with each microcomputer type. Also some microcomputer types have no Stack Page Selection Bit and the upper eight bits of the stack address are fixed. The operations of pushing register contents onto the stack and popping them from the stack are shown in Fig.7.

Program counter (PC)

The program counter is a 16-bit counter consisting of two 8-bit registers PCH and PCL. It is used to indicate the address of the next instruction to be executed.

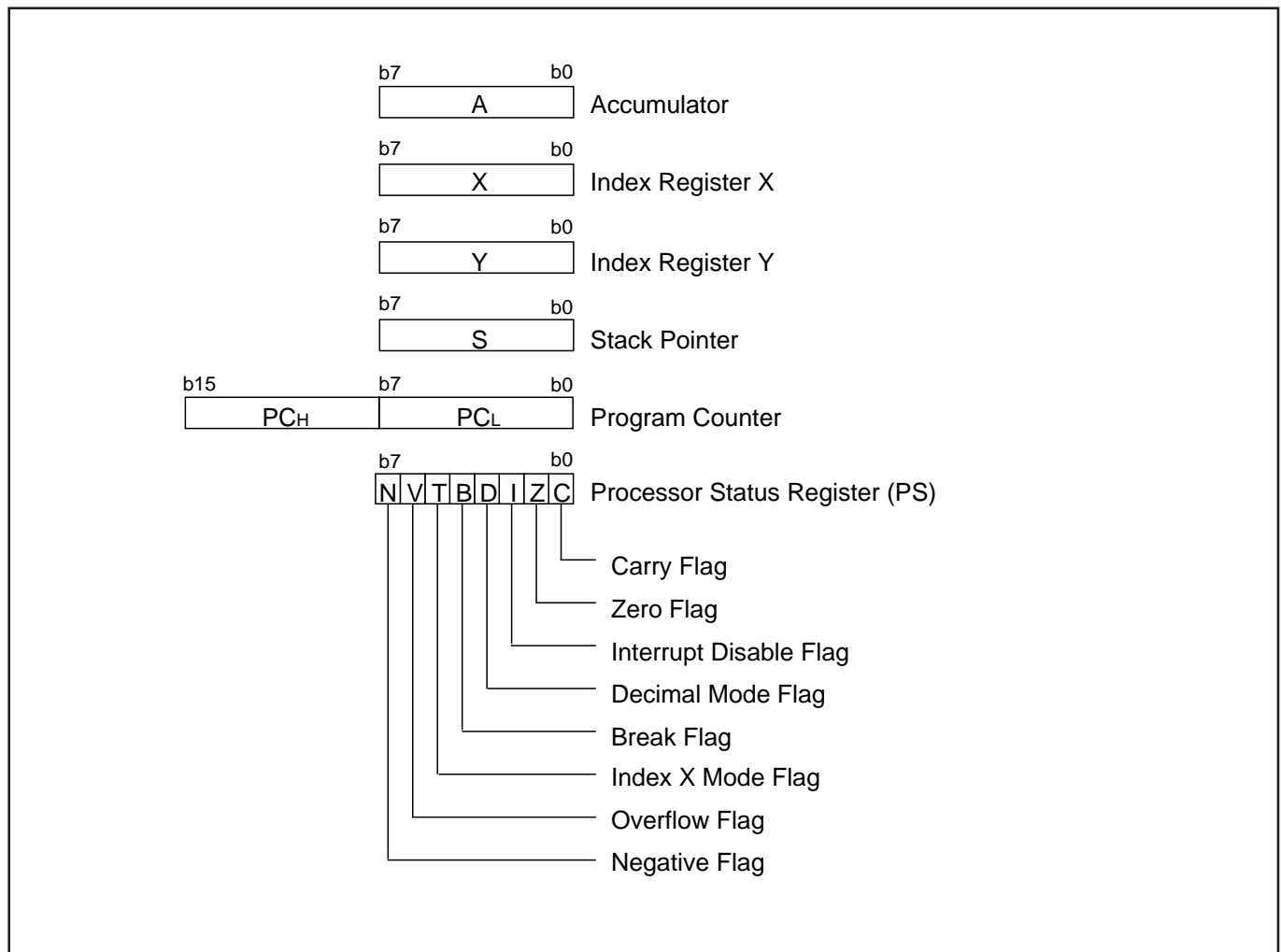


Fig. 5 740 Family CPU register structure

HARDWARE

FUNCTIONAL DESCRIPTION

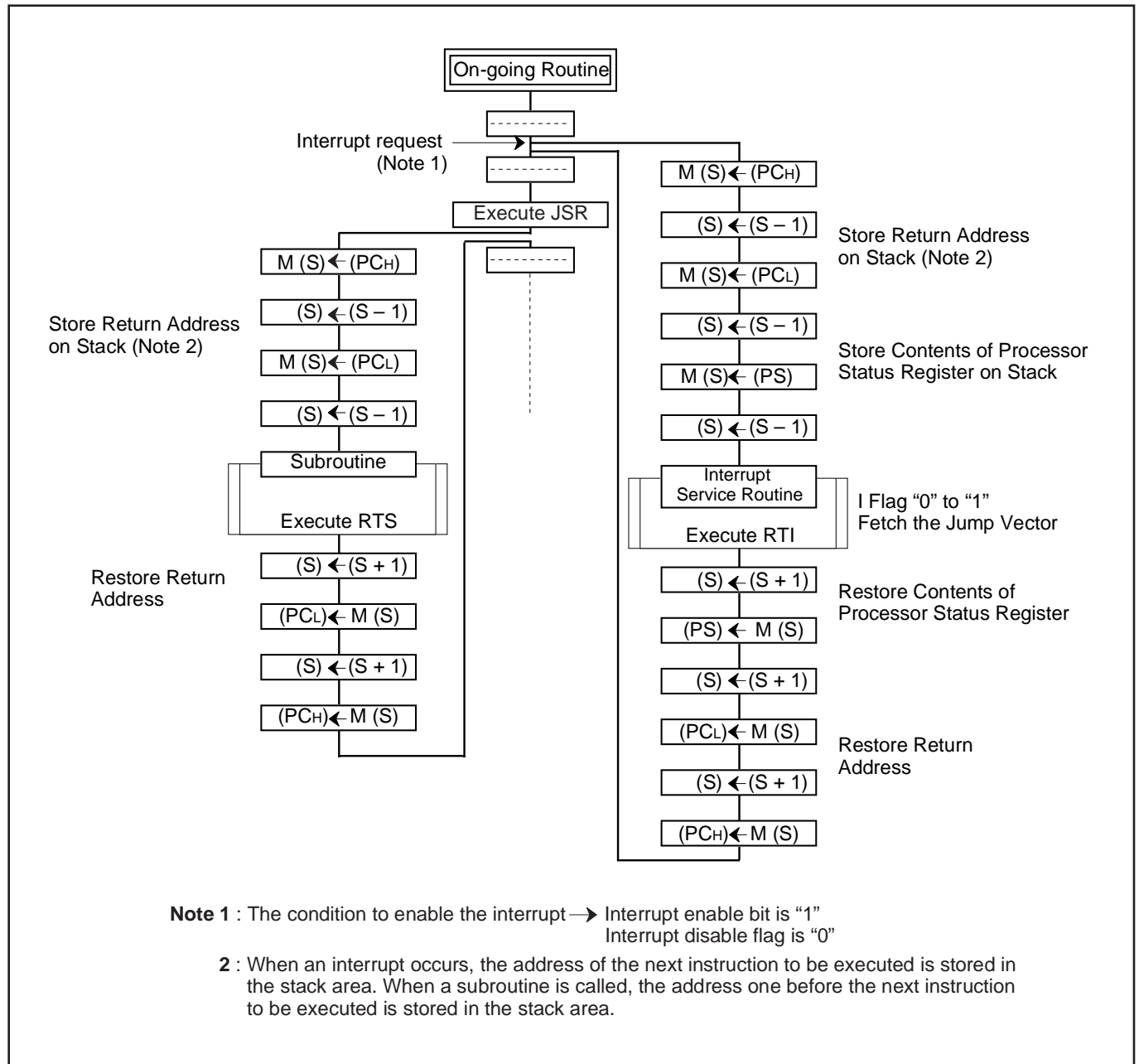


Fig. 6 Register push and pop at interrupt generation and subroutine call

Table 3 Push and pop instructions of accumulator or processor status register

	Push instruction to stack	Pop instruction from stack
Accumulator	PHA	PLA
Processor status register	PHP	PLP

Processor status register (PS)

The processor status register is an 8-bit register consisting of flags which indicate the status of the processor after an arithmetic operation. Branch operations can be performed by testing the Carry (C) flag, Zero (Z) flag, Overflow (V) flag, or the Negative (N) flag. In decimal mode, the Z, V, N flags are not valid.

After reset, the Interrupt disable (I) flag is set to "1", but all other flags are undefined. Since the Index X mode (T) and Decimal mode (D) flags directly affect arithmetic operations, they should be initialized in the beginning of a program.

(1) Carry flag (C)

The C flag contains a carry or borrow generated by the arithmetic logic unit (ALU) immediately after an arithmetic operation. It can also be changed by a shift or rotate instruction.

(2) Zero flag (Z)

The Z flag is set if the result of an immediate arithmetic operation or a data transfer is "0", and cleared if the result is anything other than "0".

(3) Interrupt disable flag (I)

The I flag disables all interrupts except for the interrupt generated by the BRK instruction.

Interrupts are disabled when the I flag is "1".

When an interrupt occurs, this flag is automatically set to "1" to prevent other interrupts from interfering until the current interrupt is serviced.

(4) Decimal mode flag (D)

The D flag determines whether additions and subtractions are executed in binary or decimal. Binary arithmetic is executed when this flag is "0"; decimal arithmetic is executed when it is "1".

Decimal correction is automatic in decimal mode. Only the ADC and SBC instructions can be used for decimal arithmetic.

(5) Break flag (B)

The B flag is used to indicate that the current interrupt was generated by the BRK instruction. The BRK flag in the processor status register is always "0". When the BRK instruction is used to generate an interrupt, the processor status register is pushed onto the stack with the break flag set to "1". The saved processor status is the only place where the break flag is ever set.

(6) Index X mode flag (T)

When the T flag is "0", arithmetic operations are performed between accumulator and memory, e.g. the results of an operation between two memory locations is stored in the accumulator. When the T flag is "1", direct arithmetic operations and direct data transfers are enabled between memory locations, i.e. between memory and memory, memory and I/O, and I/O and I/O. In this case, the result of an arithmetic operation performed on data in memory location 1 and memory location 2 is stored in memory location 1. The address of memory location 1 is specified by index register X, and the address of memory location 2 is specified by normal addressing modes.

(7) Overflow flag (V)

The V flag is used during the addition or subtraction of one byte of signed data. It is set if the result exceeds +127 to -128. When the BIT instruction is executed, bit 6 of the memory location operated on by the BIT instruction is stored in the overflow flag.

(8) Negative flag (N)

The N flag is set if the result of an arithmetic operation or data transfer is negative. When the BIT instruction is executed, bit 7 of the memory location operated on by the BIT instruction is stored in the negative flag.

Table 4 Set and clear instructions of each bit of processor status register

	C flag	Z flag	I flag	D flag	B flag	T flag	V flag	N flag
Set instruction	SEC	–	SEI	SED	–	SET	–	–
Clear instruction	CLC	–	CLI	CLD	–	CLT	CLV	–

HARDWARE

FUNCTIONAL DESCRIPTION

FUNCTIONAL DESCRIPTION

Central Processing Unit (CPU)

The core of 7630 group microcomputers is the 7600 series CPU. This core is based on the standard instruction set of 740 series; however the performance is improved by allowing to execute the same instructions as that of the 740 series in less cycles. Refer to the 7600 Series Software Manual for details of the instruction set.

CPU Mode Register CPUM

The CPU mode register contains the stack page selection bit and internal system clock selection bit. The CPU mode register is allocated to address 0000_{16} .

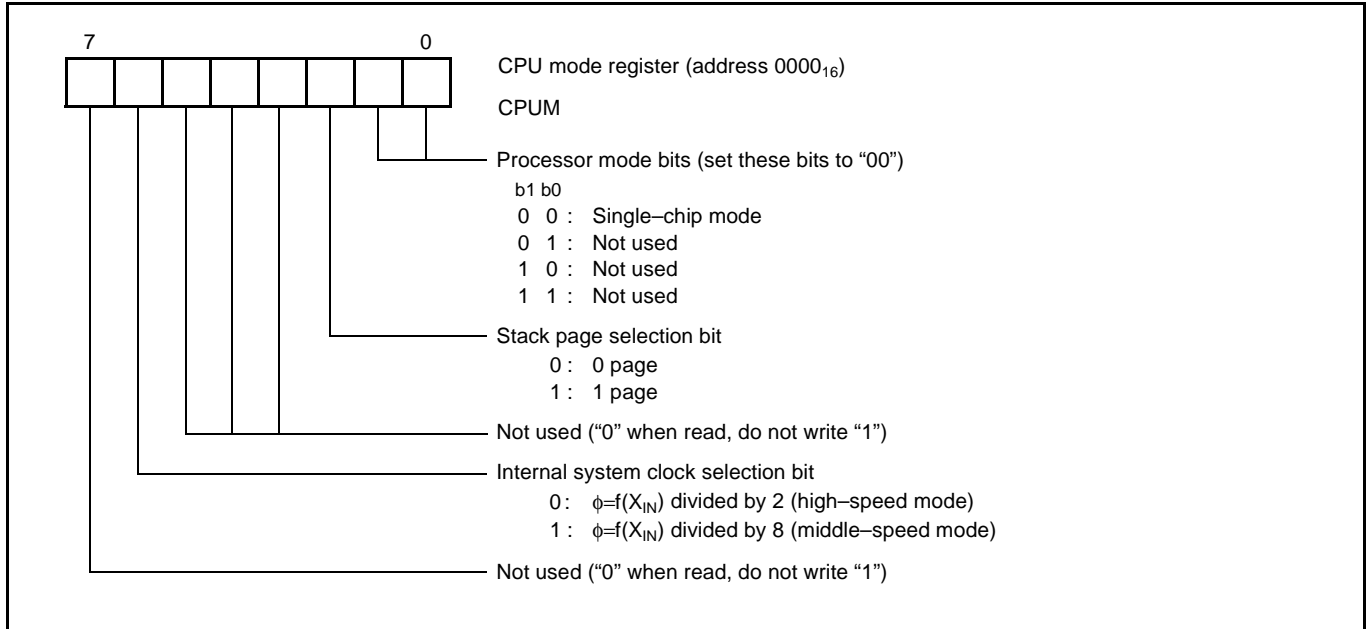


Fig. 7 Structure of CPU mode register

MEMORY

Special Function Register (SFR) Area

The special function register (SFR) area contains the registers relating to functions such as I/O ports and timers.

RAM

RAM is used for data storage and for stack area of subroutine calls and interrupts.

ROM

ROM is used for storing user's program code as well as the interrupt vector area.

Interrupt Vector Area

The interrupt vector area is for storing jump destination addresses used at reset or when an interrupt is generated.

Zero Page

This area can be accessed most efficiently by means of the zero page addressing mode.

Special Page

This area can be accessed most efficiently by means of the special page addressing mode.

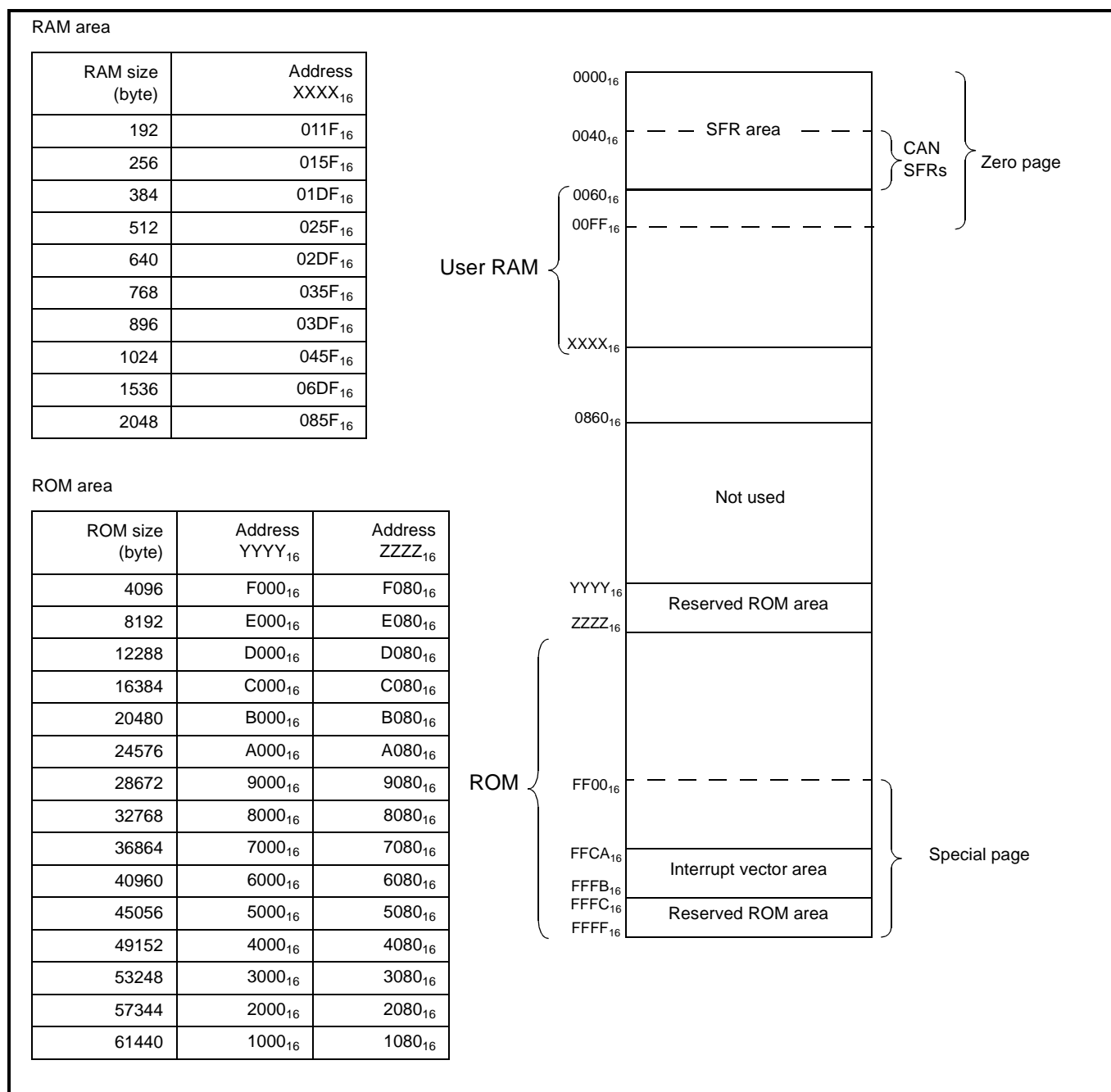


Fig. 8 Memory map diagram

HARDWARE

FUNCTIONAL DESCRIPTION

SPECIAL FUNCTION REGISTERS (SFR)

0000 ₁₆	CPU mode register	CPUM	0030 ₁₆	CAN transmit control register	CTRM
0001 ₁₆	Not used		0031 ₁₆	CAN bus timing control register 1	CBTCON1
0002 ₁₆	Interrupt request register A	IREQA	0032 ₁₆	CAN bus timing control register 2	CBTCON2
0003 ₁₆	Interrupt request register B	IREQB	0033 ₁₆	CAN acceptance code register 0	CAC0
0004 ₁₆	Interrupt request register C	IREQC	0034 ₁₆	CAN acceptance code register 1	CAC1
0005 ₁₆	Interrupt control register A	ICONA	0035 ₁₆	CAN acceptance code register 2	CAC2
0006 ₁₆	Interrupt control register B	ICONB	0036 ₁₆	CAN acceptance code register 3	CAC3
0007 ₁₆	Interrupt control register C	ICONC	0037 ₁₆	CAN acceptance code register 4	CAC4
0008 ₁₆	Port P0 register	P0	0038 ₁₆	CAN acceptance mask register 0	CAM0
0009 ₁₆	Port P0 direction register	P0D	0039 ₁₆	CAN acceptance mask register 1	CAM1
000A ₁₆	Port P1 register	P1	003A ₁₆	CAN acceptance mask register 2	CAM2
000B ₁₆	Port P1 direction register	P1D	003B ₁₆	CAN acceptance mask register 3	CAM3
000C ₁₆	Port P2 register	P2	003C ₁₆	CAN acceptance mask register 4	CAM4
000D ₁₆	Port P2 direction register	P2D	003D ₁₆	CAN receive control register	CREC
000E ₁₆	Port P3 register	P3	003E ₁₆	CAN transmit abort register	CABORT
000F ₁₆	Port P3 direction register	P3D	003F ₁₆	Reserved	
0010 ₁₆	Port P4 register	P4	0040 ₁₆	CAN transmit buffer register 0	CTB0
0011 ₁₆	Port P4 direction register	P4D	0041 ₁₆	CAN transmit buffer register 1	CTB1
0012 ₁₆	Serial I/O shift register	SIO	0042 ₁₆	CAN transmit buffer register 2	CTB2
0013 ₁₆	Serial I/O control register	SIOCON	0043 ₁₆	CAN transmit buffer register 3	CTB3
0014 ₁₆	A-D conversion register	AD	0044 ₁₆	CAN transmit buffer register 4	CTB4
0015 ₁₆	A-D control register	ADCON	0045 ₁₆	CAN transmit buffer register 5	CTB5
0016 ₁₆	Timer 1	T1	0046 ₁₆	CAN transmit buffer register 6	CTB6
0017 ₁₆	Timer 2	T2	0047 ₁₆	CAN transmit buffer register 7	CTB7
0018 ₁₆	Timer 3	T3	0048 ₁₆	CAN transmit buffer register 8	CTB8
0019 ₁₆	Timer 123 mode register	T123M	0049 ₁₆	CAN transmit buffer register 9	CTB9
001A ₁₆	Timer XL	TXL	004A ₁₆	CAN transmit buffer register A	CTBA
001B ₁₆	Timer XH	TXH	004B ₁₆	CAN transmit buffer register B	CTBB
001C ₁₆	Timer YL	TYL	004C ₁₆	CAN transmit buffer register C	CTBC
001D ₁₆	Timer YH	TYH	004D ₁₆	CAN transmit buffer register D	CTBD
001E ₁₆	Timer X mode register	TXM	004E ₁₆	Reserved	
001F ₁₆	Timer Y mode register	TYM	004F ₁₆	Reserved	
0020 ₁₆	UART mode register	UMOD	0050 ₁₆	CAN receive buffer register 0	CRB0
0021 ₁₆	UART baud rate generator	UBRG	0051 ₁₆	CAN receive buffer register 1	CRB1
0022 ₁₆	UART control register	UCON	0052 ₁₆	CAN receive buffer register 2	CRB2
0023 ₁₆	UART status register	USTS	0053 ₁₆	CAN receive buffer register 3	CRB3
0024 ₁₆	UART transmit buffer register 1	UTBR1	0054 ₁₆	CAN receive buffer register 4	CRB4
0025 ₁₆	UART transmit buffer register 2	UTBR2	0055 ₁₆	CAN receive buffer register 5	CRB5
0026 ₁₆	UART receive buffer register 1	URBR1	0056 ₁₆	CAN receive buffer register 6	CRB6
0027 ₁₆	UART receive buffer register 2	URBR2	0057 ₁₆	CAN receive buffer register 7	CRB7
0028 ₁₆	Port P0 pull-up control register	PUP0	0058 ₁₆	CAN receive buffer register 8	CRB8
0029 ₁₆	Port P1 pull-up control register	PUP1	0059 ₁₆	CAN receive buffer register 9	CRB9
002A ₁₆	Port P2 pull-up control register	PUP2	005A ₁₆	CAN receive buffer register A	CRBA
002B ₁₆	Port P3 pull-up control register	PUP3	005B ₁₆	CAN receive buffer register B	CRBB
002C ₁₆	Port P4 pull-up/down control register	PUP4	005C ₁₆	CAN receive buffer register C	CRBC
002D ₁₆	Interrupt polarity selection register	IPOL	005D ₁₆	CAN receive buffer register D	CRBD
002E ₁₆	Watchdog timer register	WDT	005E ₁₆	Reserved	
002F ₁₆	Polarity control register	PCON	005F ₁₆	Reserved	

Fig. 9 Memory map of special register (SFR)

I/O PORTS

The 7630 group has 35 programmable I/O pins and one input pin arranged in five I/O ports (ports P0 to P4). The I/O ports are controlled by the corresponding port registers and port direction registers; each I/O pin can be controlled separately.

When data is read from a port configured as an output port, the port latch's contents are read instead of the port level. A port configured

as an input port becomes floating and its level can be read. Data written to this port will affect the port latch only; the port remains floating. Refer to Structure of port- and port direction registers, Structure of port I/Os (1) and Structure of port I/Os (2).

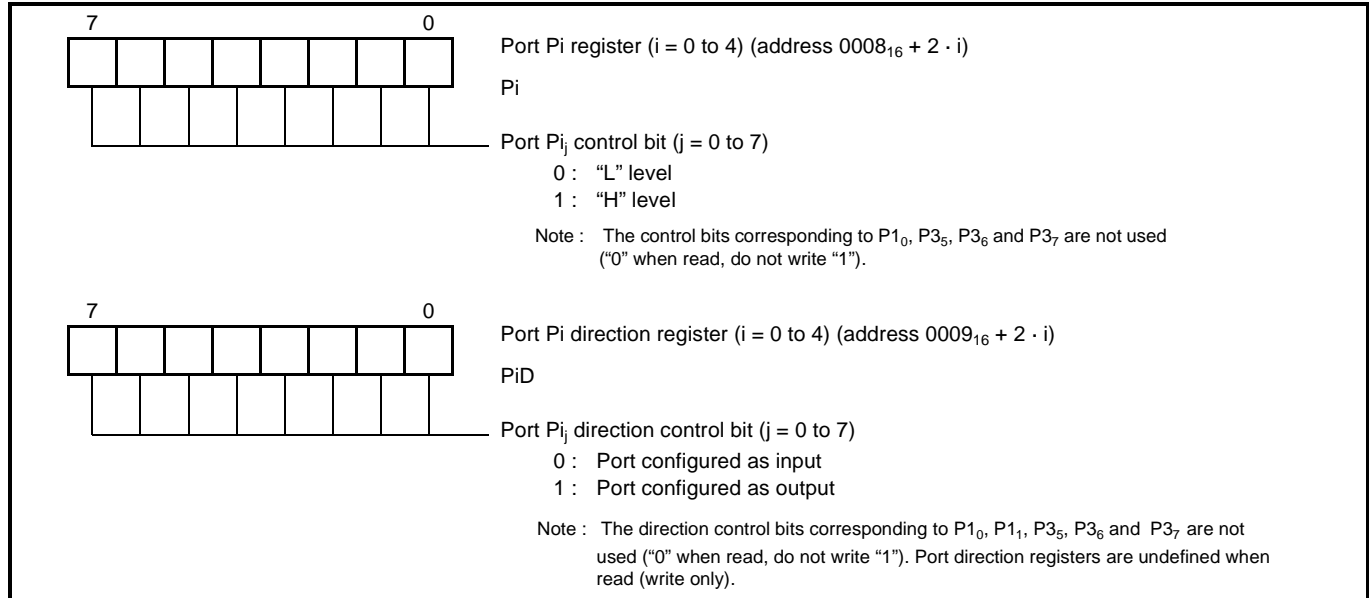


Fig. 10 Structure of port- and port direction registers

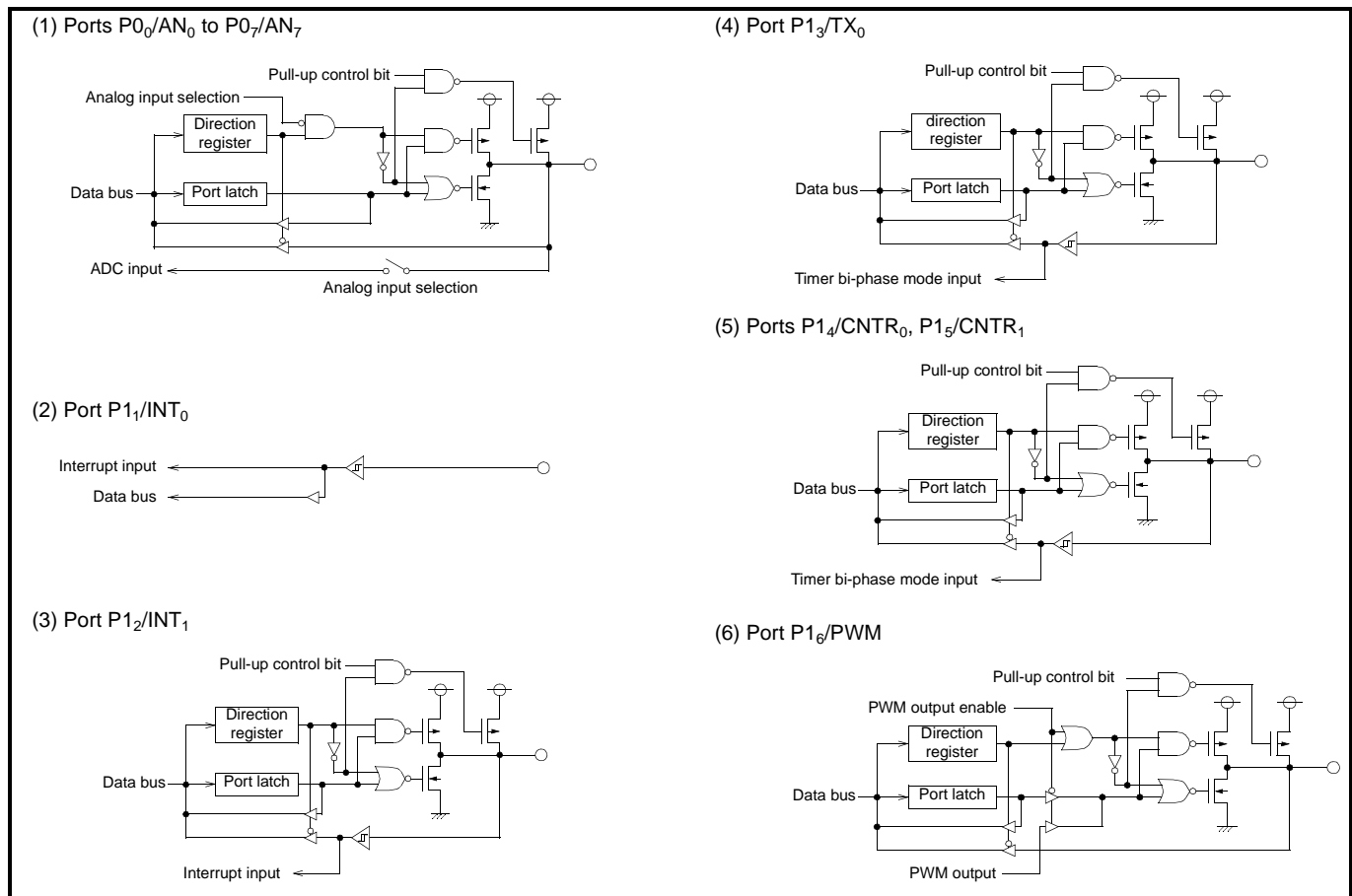


Fig. 11 Structure of port I/Os (1)

HARDWARE

I/O PORTS

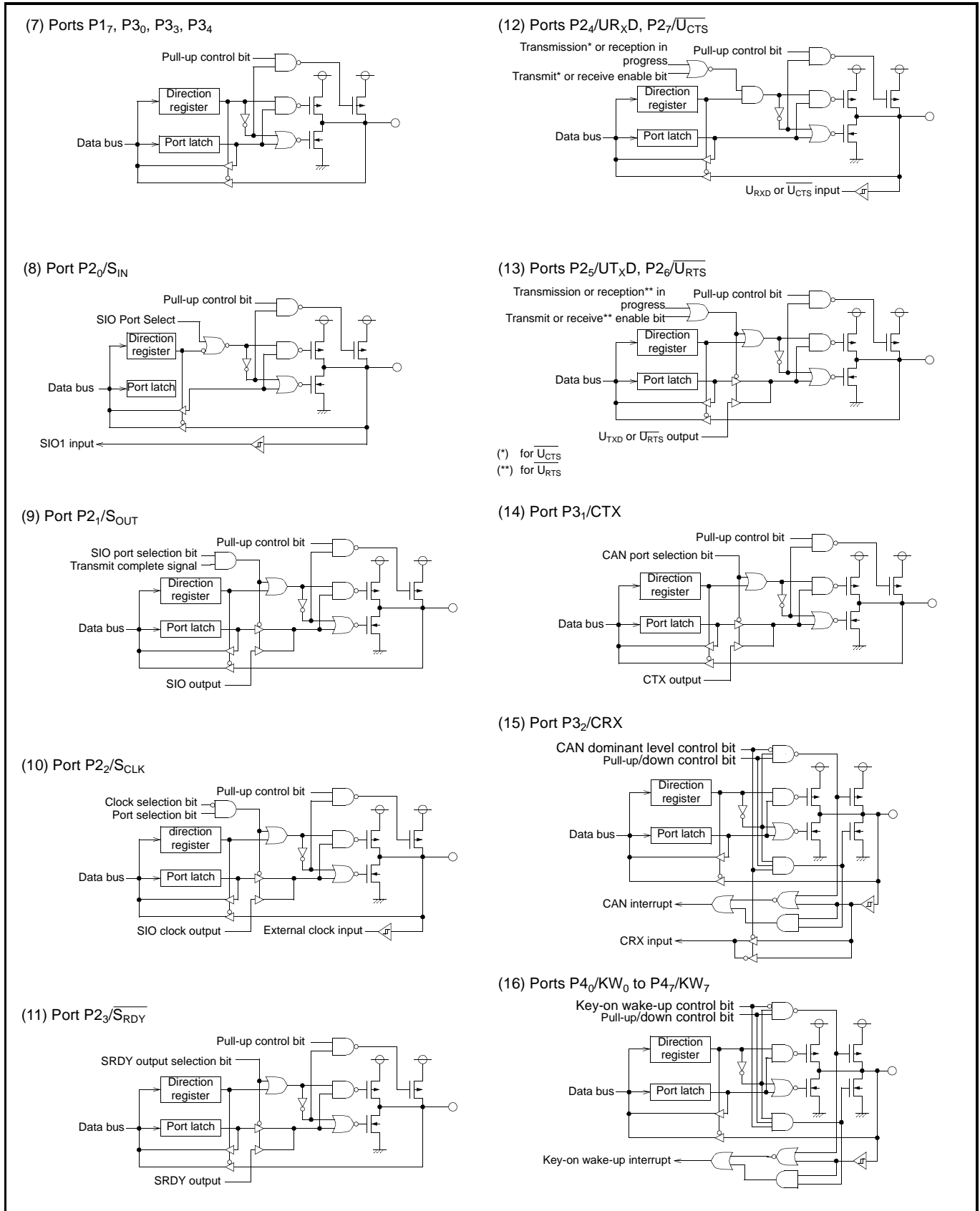


Fig. 12 Structure of port I/Os (2)

Port Pull-up/pull-down Function

Each pin of ports P0 to P4 except P1₁ is equipped with a programmable pull-up transistor. P3₂/CRX and P4₀/KW₀ to P4₇/KW₇ are equipped with programmable pull-down transistors as well. The pull-up function of P0 to P3 can be controlled by the corresponding port

pull-up control registers (see Structure of port pull-up/down control registers). The pull-up/down function of ports P3₂ and P4 can be controlled by the corresponding port pull-up/pull-down registers together with the polarity control register (see Structure of polarity control register).

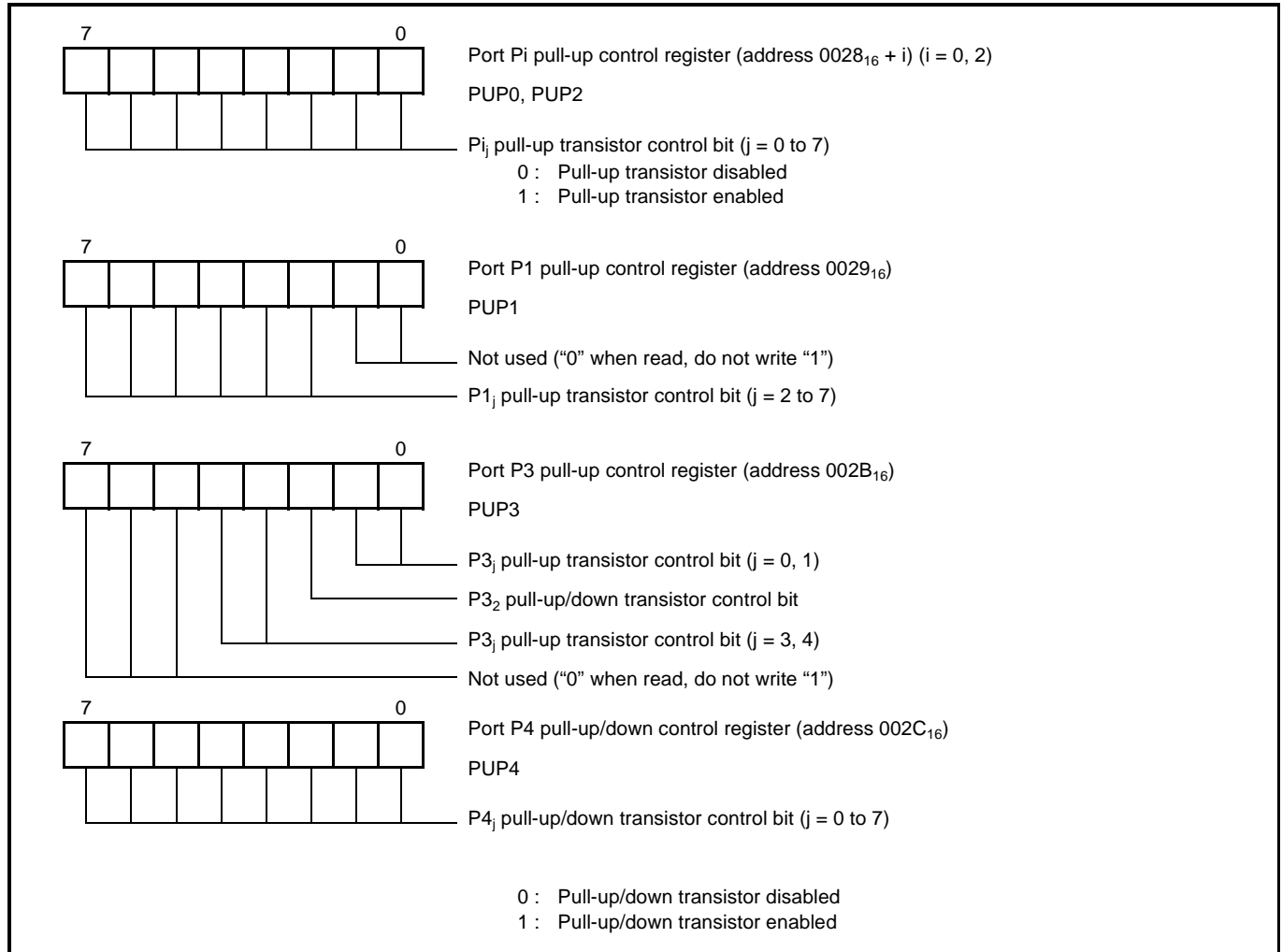


Fig. 13 Structure of port pull-up/down control registers

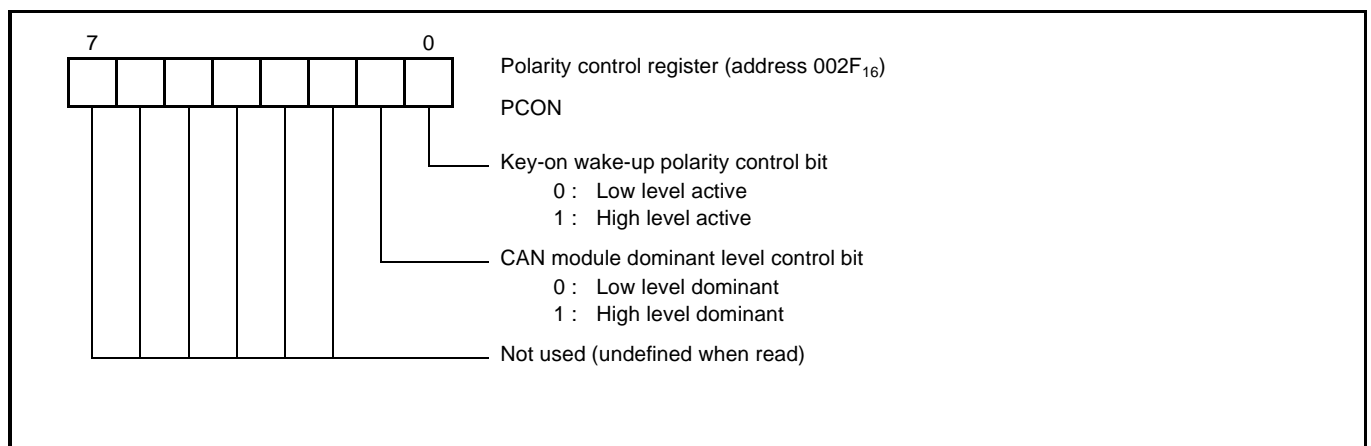


Fig. 14 Structure of polarity control register

HARDWARE

I/O PORTS

Port Overvoltage Application

When configured as input ports, P1 to P4 may be subjected to overvoltage ($V_I > V_{CC}$) if the input current to the applicable port is limited to the specified values (see "Table 10:"). Use a serial resistor of appropriate size to limit the input current. To estimate the resistor value, assume the port voltage to be V_{CC} at overvoltage condition.

Notes:

- Subjecting ports to overvoltage may effect the supply voltage. Assure to keep V_{CC} and V_{SS} within the target limits.
- Avoid to subject ports to overvoltage causing V_{CC} to rise above 5.5 V.
- The overvoltage condition causing input current flowing through the internal port protection circuits has a negative effect on the ports noise immunity. Therefore, careful and intense testing of the target system's noise immunity is required. Refer to the "countermeasures against noise" of the corresponding users manual.
- Port P0 must not be subjected to overvoltage conditions.

INTERRUPTS

There are 24 interrupts: 6 external, 17 internal, and 1 software.

Interrupt Control

Each interrupt except the BRK instruction interrupt has both an interrupt request bit and an interrupt enable bit, and is controlled by the interrupt disable flag. An interrupt occurs when the corresponding interrupt request and enable bits are "1" and the interrupt disable flag is "0". Interrupt enable bits can be cleared or set by software. Interrupt request bits can be cleared by software but cannot be set by software. The BRK instruction interrupt and reset cannot be disabled with any flag or bit. The I flag disables all interrupts except the BRK instruction interrupt and reset. If several interrupt requests occur at the same time, the interrupt with the highest priority is accepted first.

Interrupt Operation

Upon acceptance of an interrupt, the following operations are automatically performed.

1. The processing being executed is stopped.

2. The contents of the program counter and processor status register are automatically pushed onto the stack.
3. Concurrently with the push operation, the interrupt jump destination address is read from the vector table into the program counter.
4. The interrupt disable flag is set and the corresponding interrupt request bit is cleared.

Notes on use

When the active edge of an external interrupt (INT₀, INT₁, CNTR₀, CNTR₁, CWKU or KOI) is changed, the corresponding interrupt request bit may also be set. Therefore, take the following sequence.

- (1) Disable the external interrupt which is selected.
- (2) Change the active edge in interrupt edge selection register.
(in the case of CNTR₀: Timer X mode register; in the case of CNTR₁: Timer Y mode register)
- (3) Clear the interrupt request bit to "0".
- (4) Enable the external interrupt which is selected.

HARDWARE

INTERRUPTS

Table 5: Interrupt vector addresses and priority

Interrupt source	Priority	Vector Address (Note 1)		Interrupt Request Generating Conditions	Remarks
		High	Low		
Reset (Note 2)	1	FFFB ₁₆	FFFA ₁₆	At Reset	Non-maskable
Watchdog timer	2	FFF9 ₁₆	FFF8 ₁₆	At Watchdog timer underflow	Non-maskable
INT0	3	FFF7 ₁₆	FFF6 ₁₆	At detection of either rising or falling edge of INT ₀ interrupt	External Interrupt (active edge selectable)
INT1	4	FFF5 ₁₆	FFF4 ₁₆	At detection of either rising or falling edge of INT ₁ interrupt	External Interrupt (active edge selectable)
CAN successful transmit	5	FFF3 ₁₆	FFF2 ₁₆	At CAN module successful transmission of message	Valid when CAN module is activated and request transmit
CAN successful receive	6	FFF1 ₁₆	FFF0 ₁₆	At CAN module successful reception of message	Valid when CAN module is activated
CAN overrun	7	FFEF ₁₆	FFEE ₁₆	If CAN module receives message when receive buffers are full.	Valid when CAN module is activated
CAN error passive	8	FFED ₁₆	FFEC ₁₆	When CAN module enters into error passive state	Valid when CAN module is active
CAN error bus off	9	FFEB ₁₆	FFEA ₁₆	When CAN module enters into bus off state	Valid when CAN module is active
CAN wake up	10	FFE9 ₁₆	FFE8 ₁₆	When CAN module wakes up via CAN bus	
Timer X	11	FFE7 ₁₆	FFE6 ₁₆	At Timer X underflow or overflow	
Timer Y	12	FFE5 ₁₆	FFE4 ₁₆	At Timer Y underflow	
Timer 1	13	FFE3 ₁₆	FFE2 ₁₆	At Timer 1 underflow	
Timer 2	14	FFE1 ₁₆	FFE0 ₁₆	At Timer 2 underflow	
Timer 3	15	FFDF ₁₆	FFDE ₁₆	At Timer 3 underflow	
CNTR0	16	FFDD ₁₆	FFDC ₁₆	At detection of either rising or falling edge in CNTR ₀ input	External Interrupt (active edge selectable)
CNTR1	17	FFDB ₁₆	FFDA ₁₆	At detection of either rising or falling edge in CNTR ₁ input	External Interrupt (active edge selectable)
UART receive	18	FFD9 ₁₆	FFD8 ₁₆	At completion of UART receive	Valid when UART is selected
UART transmit	19	FFD7 ₁₆	FFD6 ₁₆	At completion of UART transmit	Valid when UART is selected
UART transmit buffer empty	20	FFD5 ₁₆	FFD4 ₁₆	At UART transmit buffer empty	Valid when UART is selected
UART receive error	21	FFD3 ₁₆	FFD2 ₁₆	When UART reception error occurs.	Valid when UART is selected
Serial I/O	22	FFD1 ₁₆	FFD0 ₁₆	At completion of serial I/O data transmit and receive	Valid when serial I/O is selected
A-D conversion	23	FFCF ₁₆	FFCE ₁₆	At completion of A-D conversion	
Key-on wake-up	24	FFCD ₁₆	FFCC ₁₆	At detection of either rising or falling edge of P4 input	External Interrupt (active edge selectable)
BRK instruction	25	FFCB ₁₆	FFCA ₁₆	At BRK instruction execution	Non-maskable

Notes 1: Vector addresses contain interrupt jump destination address

2: Reset function in the same way as an interrupt with the highest priority

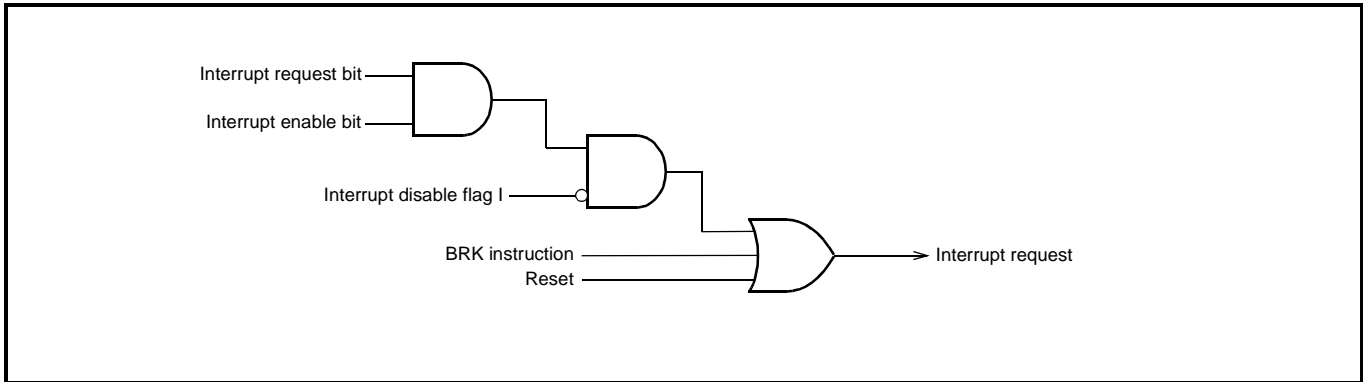


Fig. 15 Interrupt control

For the external interrupts INT0 and INT1, the active edge causing the interrupt request can be selected by the INT0 and INT1 interrupt edge selection bits of the interrupt polarity selection register (IPOL); please refer to Fig. 16 below.

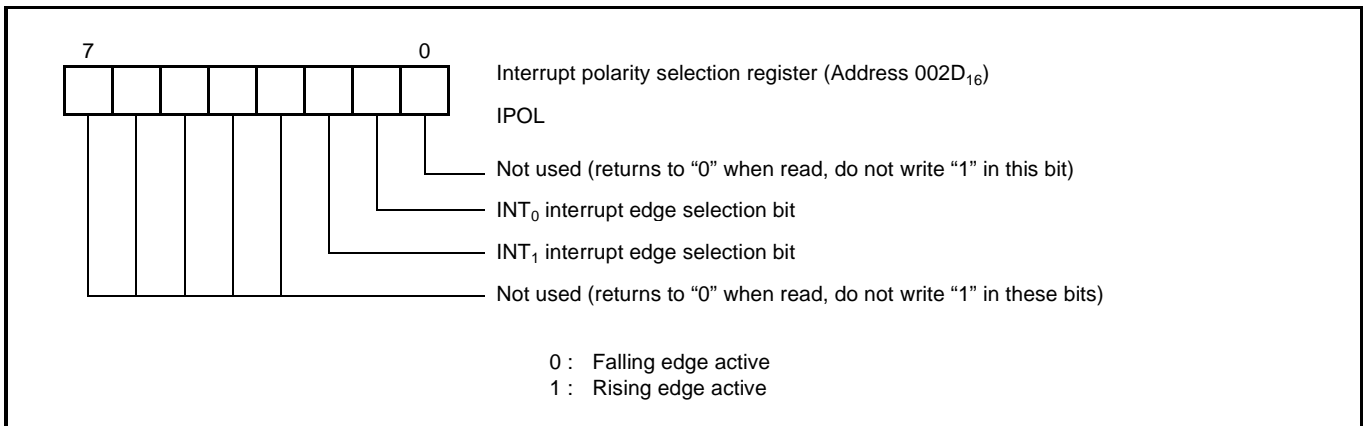


Fig. 16 Structure of interrupt polarity selection register

HARDWARE

INTERRUPTS

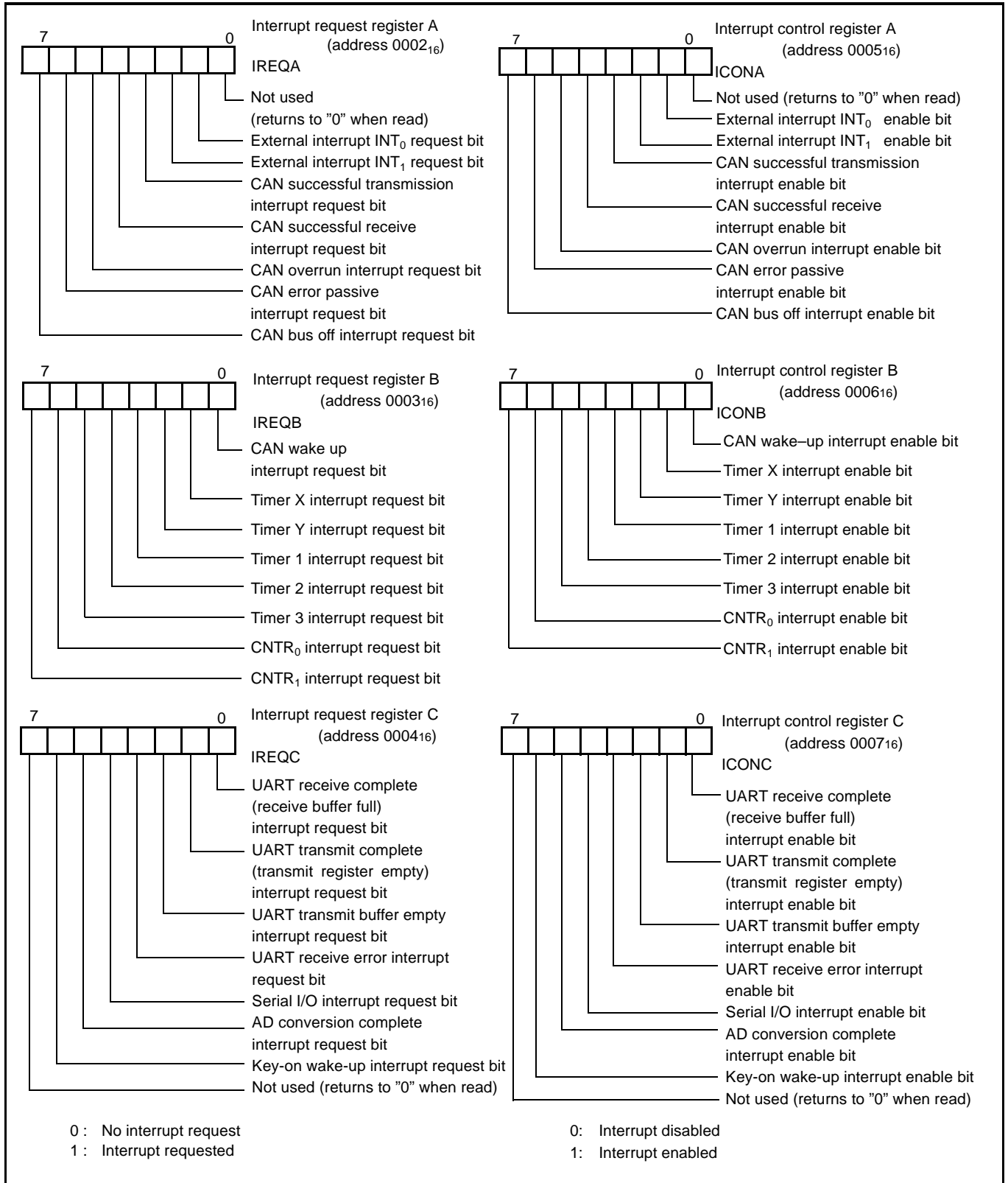


Fig. 17 Structure of interrupt request and control registers A, B and C

KEY-ON WAKE-UP

“Key-on wake-up” is one way of returning from a power-down state caused by the STP or WIT instruction. Any terminal of port P4 can be used to generate the key-on wake-up interrupt request. The active polarity can be selected by the key-on wake-up polarity control bit of

PCON (see Fig. 14). If any pin of port P4 has the selected active level applied, the key-on wake-up interrupt request will be set to “1”. Please refer to Fig. 18.

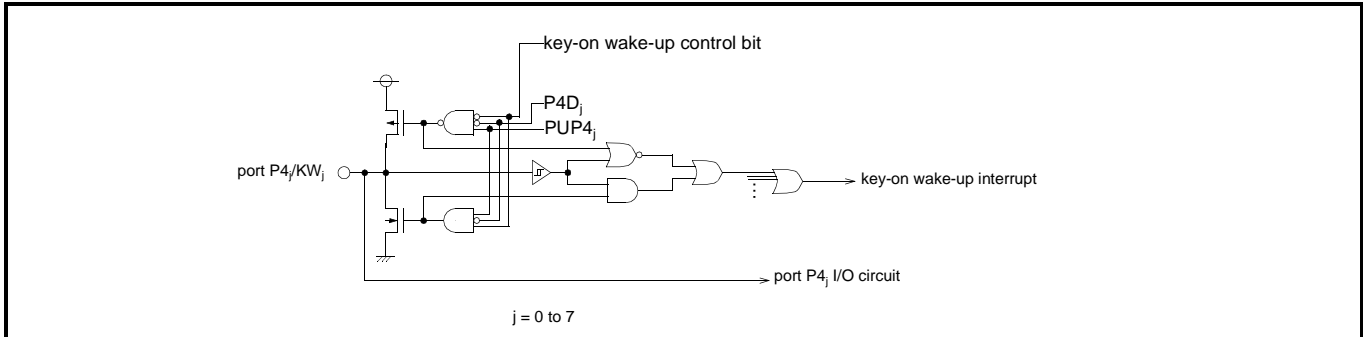


Fig. 18 Block diagram of key-on wake-up circuit

HARDWARE

TIMERS

TIMERS

The 7630 group has five timers: two 16-bit timers and three 8-bit timers. All these timers will be described in detail below.

16-bit Timers

Timers X and Y are 16-bit timers with multiple operating modes. Please refer to Fig. 19.

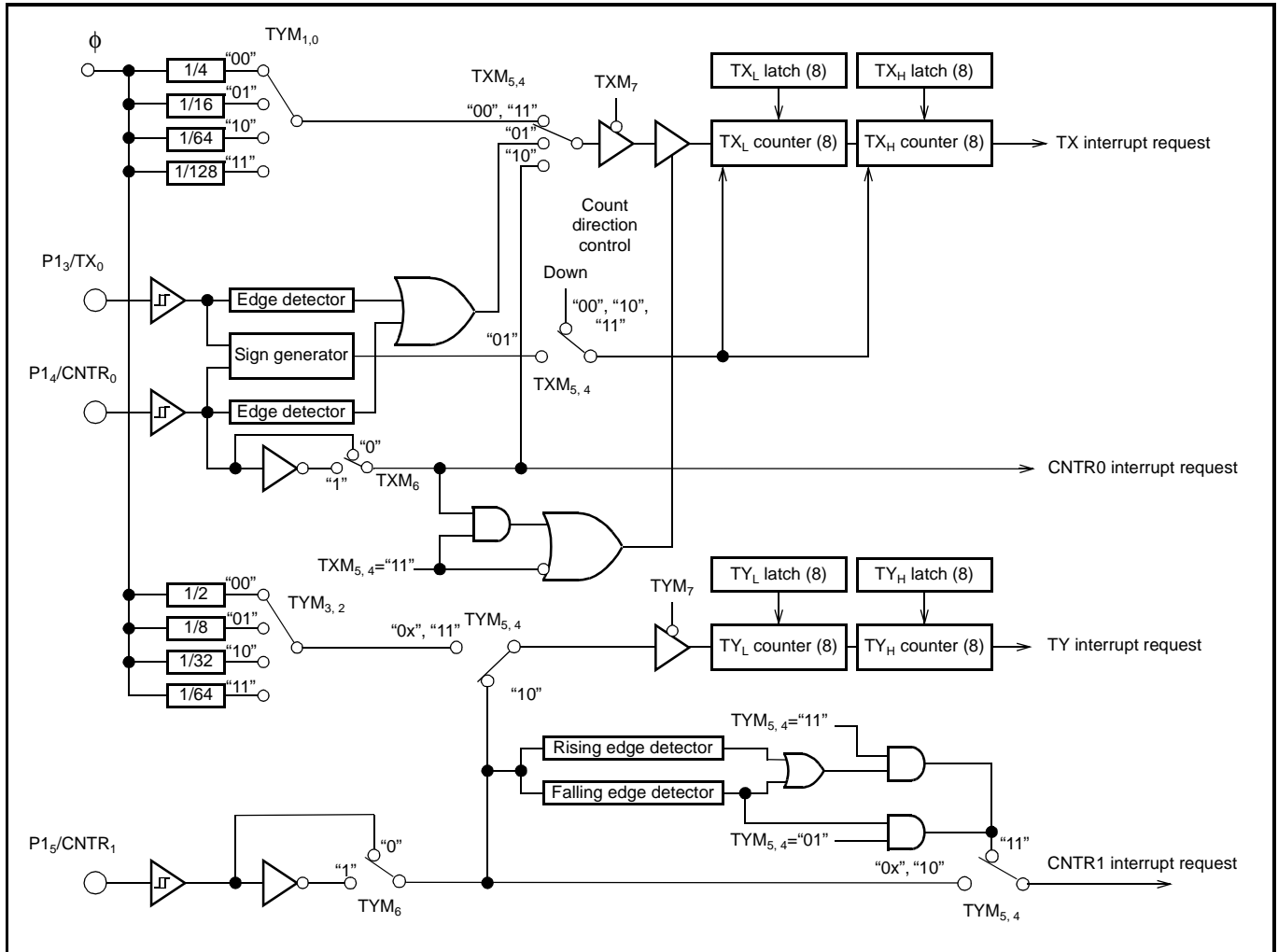


Fig. 19 Block diagram of timers X and Y (ϕ is internal system clock)

Timer X

Timer X is a 16-bit timer with a 16-bit reload latch supporting the following operating modes:

- (1) Timer mode
- (2) Bi-phase counter mode
- (3) Event counter mode
- (4) Pulse width measurement mode

These modes can be selected by the timer X mode register (TXM). In the timer- and pulse width measurement mode, the timer's count source can be selected by the timer X count source selection bits of the timer Y mode register (TYM). Please refer to the Figures below for the TXM and TYM bit assignment.

On read or write access to timer X, note that the high-order and low-order bytes must be accessed in the specific order.

Write method

When writing to the timer X, write the low-order byte first. The data written is stored in a temporary register which is assigned to the

same address as TX_L. Next, write the high-order byte. When this is finished, the data is placed in the timer X high-order reload latch and the low-order byte is transferred from its temporary register to the timer X low-order reload latch. Depending on the timer X write control bit, the latch contents are reloaded to the timer immediately (write control bit = "0") or on the next timer underflow (write control bit = "1").

Read method

When reading the timer X, read the high-order byte first. This causes the timer X high- and low-order bytes to be transferred to temporary registers being assigned to the same addresses as TX_H and TX_L. Next, read the low-order byte which is read from the temporary register. This method assures the correct timer value can be read during the timer count operation.

Timer X count stop control

Regardless of the actual operating mode, timer X can be stopped by setting the timer X count stop bit (bit 7 of the timer X mode register) to "1".

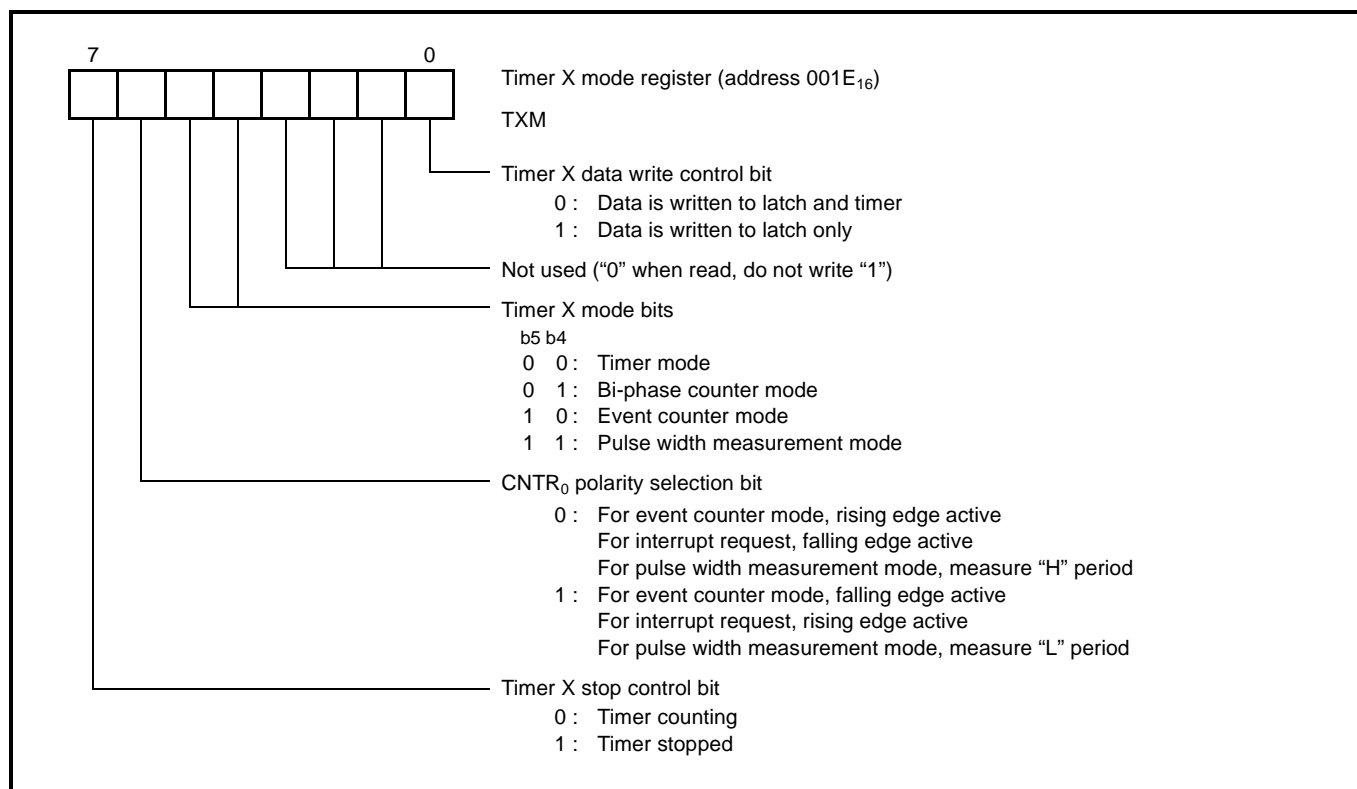


Fig. 20 Structure of Timer X mode register

Timer Y

Timer Y is a 16 bit timer with a 16-bit reload latch supporting the following operating modes:

- (1) Timer mode
- (3) Event counter mode
- (5) Pulse period measurement mode
- (6) H/L pulse width measurement mode

These modes can be selected by the timer Y mode register (TYM). In the timer, pulse period- and pulse width measurement modes' the timer's count source can be selected by the timer Y count source selection bits. Please refer to Fig. 21.

On read or write access to timer Y, note that the high-order and low-order bytes must be accessed in a specific order.

Write method

When writing to timer Y, write the low-order byte first. The data written is stored in a temporary register which is assigned to the same

address as TY_L. Next, write the high-order byte. When this is finished, the data is placed in the timer Y high-order reload latch and the low-order byte is transferred from its temporary register to the timer Y low-order reload latch.

Read method

When reading the timer Y, read the high-order byte first. This causes the timer Y high- and low-order bytes to be transferred to temporary registers being assigned to the same addresses as TY_H and TY_L. Next, read the low-order byte which is read from the temporary register. This method assures the correct timer value can be read during timer count operation.

Timer Y count stop control

Regardless of the actual operating mode, timer Y can be stopped by setting the timer Y count stop bit (bit 7 of the timer Y mode register) to "1".

HARDWARE

TIMERS

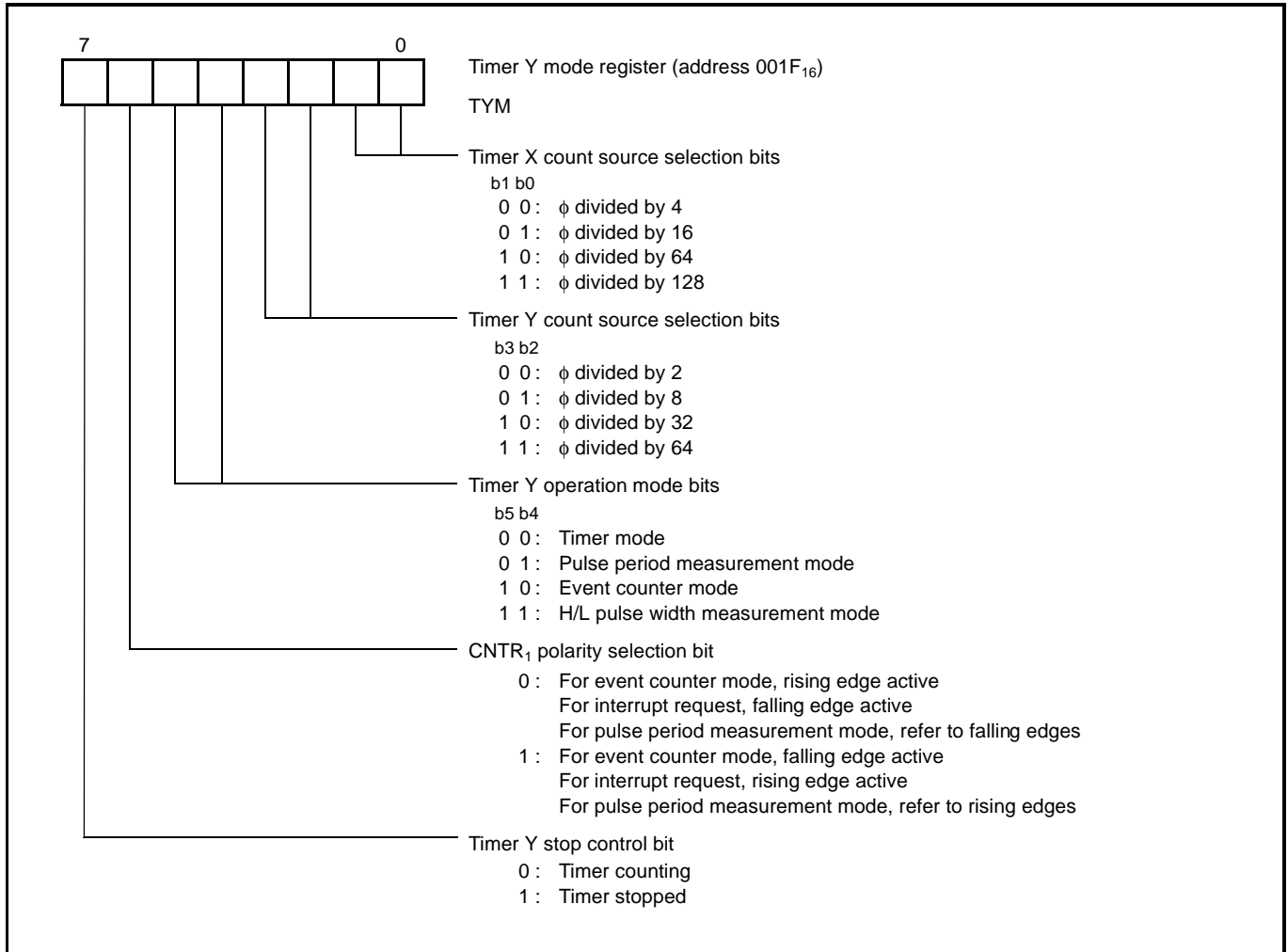


Fig. 21 Structure of timer Y mode register (ϕ is internal system clock)

Operating Modes

(1) Timer mode

This mode is available with timer X and timer Y.

- Count source
The count source for timer X and Y is the output of the corresponding clock divider. The division ratio can be selected by the timer Y mode register.
- Operation
Both timers X and Y are down counters. On a timer underflow, the corresponding timer interrupt request bit will be set to "1", the contents of the corresponding timer latches will be reloaded to the counters and counting continues.

(2) Bi-phase counter mode (quadruplicate)

This mode is available with timer X only.

- Count source
The count sources are P1₄/CNTR₀ and the P1₃/TX₀ pins.
- Operation
Timer X will count both rising and falling edges on both input pins (see above). Refer to Timer X bi-phase counter mode operation for the timing chart of the bi-phase counter mode.

The count direction is determined by the edge polarity and level of count source inputs and may change during the count operation. Refer to the table below.

Table 6: Timer X count direction in Bi-phase counter mode

P1 ₃ /TX ₀	P1 ₄ /CNTR ₀	Count direction
↑ Edge	L	Up
	H	Down
↓ Edge	L	Down
	H	Up
L	↑ Edge	Down
H		Up
L	↓ Edge	Up
H		Down

On a timer over- or underflow, the corresponding interrupt request bit will be set to "1" and counting continues.

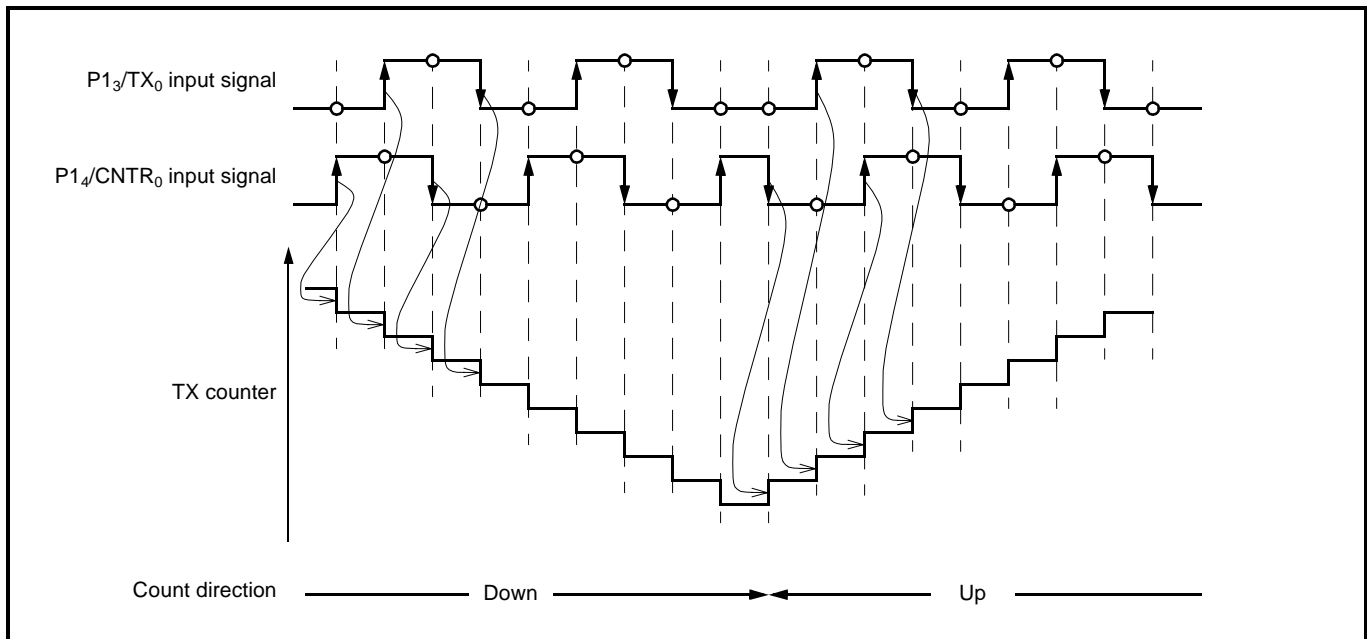


Fig. 22 Timer X bi-phase counter mode operation

(3) Event counter mode

This mode is available with timer X and timer Y.

- **Count source**
The count source for timer X is the input signal to the P1₄/CNTR₀ pin and for timer Y the input signal to P1₅/CNTR₁ pin.
- **Operation**
The timer counts down. On a timer underflow, the corresponding timer interrupt request bit will be set to "1", the contents of the corresponding timer latches will be reloaded to the counters and counting continues. The active edge used for counting can be selected by the polarity selection bit of the corresponding pin P1₄/CNTR₀ or P1₅/CNTR₁. These bits are part of TXM (Structure of Timer X mode register) and TYM (Structure of timer Y mode register (f is internal system clock)) registers.

(4) Pulse width measurement mode

This mode is available with timer X only.

- **Count source**
The count source is the output of timer X clock divider. The division ratio can be selected by the timer Y mode register.
- **Operation**
The timer counts down while the input signal level on P1₄/CNTR₀ matches the active polarity selected by the CNTR₀ polarity selection bit of TXM (Structure of Timer X mode register). On a timer underflow, the timer X interrupt request bit will be set to "1", the contents of the timer latches are reloaded to the counters and counting continues. When the input level changes from active polarity (as selected), the CNTR₀ interrupt request bit will be set to "1." The measurement result may be obtained by reading timer X during interrupt service.

(5) Pulse period measurement mode

This mode is available with timer Y only.

- **Count source**
The count source is the output of timer Y clock divider.

- **Operation**
The active edge of input signal to be measured can be selected by CNTR₁ polarity selection bit (Fig. 20). When this bit is set to "0", the time between two consecutive falling edges of the signal input to P1₅/CNTR₁ pin will be measured, when the polarity bit is set to "1", the time between two consecutive rising edges will be measured.
The timer counts down. On detection of an active edge of input signal, the contents of the TY counters will be transferred to temporary registers assigned to the same addresses as TY. At the same time, the contents of TY latches will be reloaded to the counters and counting continues. The active edge of input signal also causes the CNTR₁ interrupt request bit to be set to "1". The measurement result may be obtained by reading timer Y during interrupt service.

(6) H/L pulse width measurement mode

This mode is available with timer Y only.

- **Count source**
The count source is the output of the timer Y's clock divider.
- **Operation**
This mode measures both the "H" and "L" periods of a signal input to P1₅/CNTR₁ pin continuously. On detection of any edge (rising or falling) of input signal to P1₅/CNTR₁ pin, the contents of timer Y counters are stored to temporary registers which are assigned to the same addresses as timer Y. At the same time, the contents of timer Y latches are reloaded to the counters and counting continues. The detection of an edge causes the CNTR₁ interrupt request bit to be set to "1" as well. The result of measurement may be obtained by reading timer Y during interrupt service. This read access will address the temporary registers. On a timer underflow, the timer Y interrupt request bit will be set to "1", the contents of timer Y latches will be transferred to the counters and counting continues.

HARDWARE

TIMERS

TIMER 1, TIMER 2, TIMER 3

Timers 1 to 3 are 8-bit timers with 8-bit reload latches and one common pre-divider. Timer 1 can operate in the timer mode only, whereas

timers 2 and 3 can be used to generate a PWM output signal timing as well. Timers 1 to 3 are down count timers. See Fig. 23.

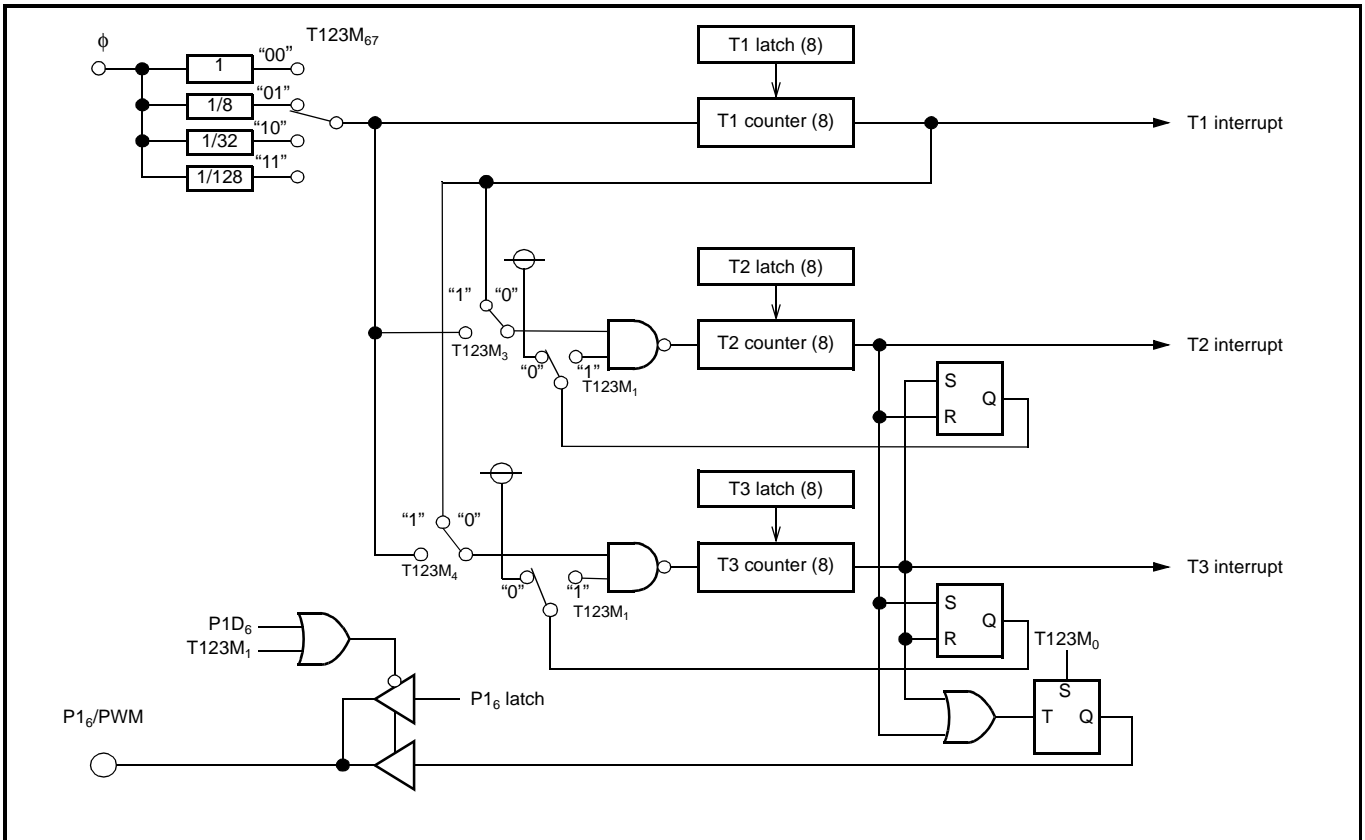


Fig. 23 Block diagram of timers 1 to 3 (ϕ is internal system clock)

Timer 1

The count source of timer 1 is the output of timer 123 pre-divider. The division ratio of the pre-divider can be selected by the pre-divider division ratio bits of timer 123 mode register (T123M). Refer to Timer 123 mode register configuration (f is internal system clock).

On a timer 1 underflow, the timer 1 interrupt request bit will be set to "1".

Writing to timer 1 initializes the latch and counter.

Timers 2 and 3

The count source of timers 2 and 3 can be either the output of the timer 123 pre-divider or the timer 1 underflow. The count source can

be selected by the timer count source selection bits of timer 123 mode register (T123M).

Writing to timer 2 register affects the reload latch only or both of the reload latch and counter depending on the timer 2 write control bit of T123M. When the timer write control bit is set to "0", both latch and counter will be initialized simultaneously; when set to "1" only the reload latch will be initialized, on an underflow, the counter will be set to the modified reload value. Writing to timer 3 initializes latch and counter both.

Timer 2 or 3 underflow causes the timer 2 or 3 interrupt request bit to be set to "1".

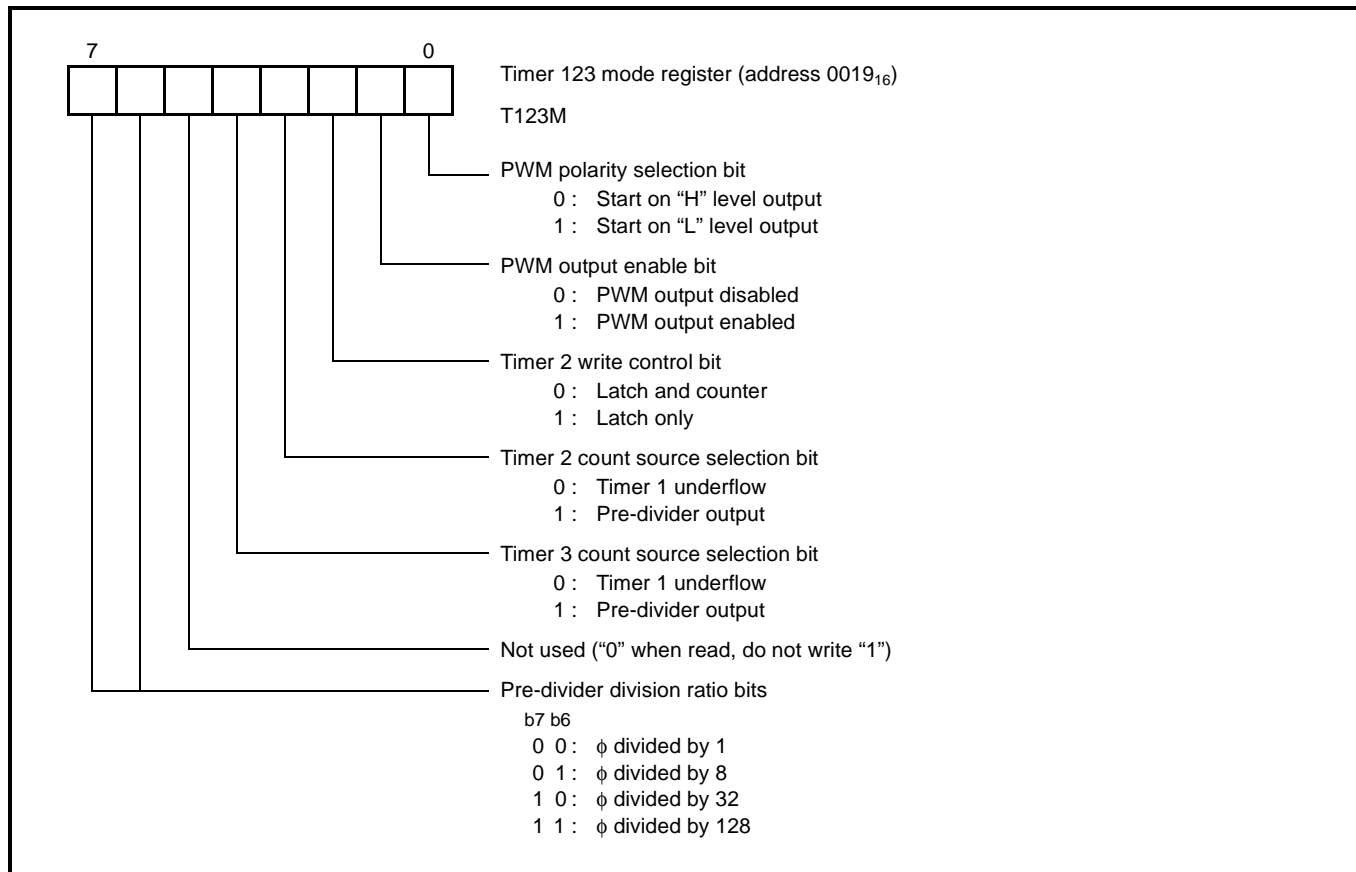


Fig. 24 Timer 123 mode register configuration (ϕ is internal system clock)

Operating Modes

(1) Timer Mode

This mode is available with timers 1 to 3.

- Count source
For timer 1, the count source is the output of the corresponding pre-divider. For timers 2 and 3, the count source can be separately selected to be either the pre-divider output or timer 1 underflow.
- Operation
The timer counts down. On a timer underflow, the corresponding timer interrupt request bit will be set to "1", the contents of the corresponding timer latch will be reloaded to the counter and counting continues.

(2) PWM Mode

This mode is available with timer 2 and 3.

- Count source
The count source can be separately selected to be either the pre-divider output or timer 1 underflow.
- Operation
When the PWM-mode is enabled, timer 2 starts counting. As soon as timer 2 underflows, timer 2 stops and timer 3 starts counting. If bit 0 is set, timer 2 determines the low duration and the initial output level is low. Timer 3 determines the high duration. If bit 0 is zero timer 2 determines the high duration and the initial output level is high. In this case timer 3 determines the low duration.
- Note: Be sure to configure the P1₆/PWM pin as an output port before using PWM mode.

HARDWARE

SERIAL I/Os

SERIAL I/Os

The serial I/O section of 7630 group consists of one clock synchronous and one asynchronous (UART) interface.

Clock Synchronous Serial I/O (SIO)

The clock synchronous interface allows full duplex communication based on 8 bit word length. The transfer clock can be selected from an internal or external clock. When an internal clock is selected, a programmable clock divider allows eight different transmission

speeds. Refer to Block diagram of clock synchronous I/O (f is internal system clock). The operation of the clock synchronous serial I/O can be configured by the serial I/O control register SIOCON; refer to Fig. 27.

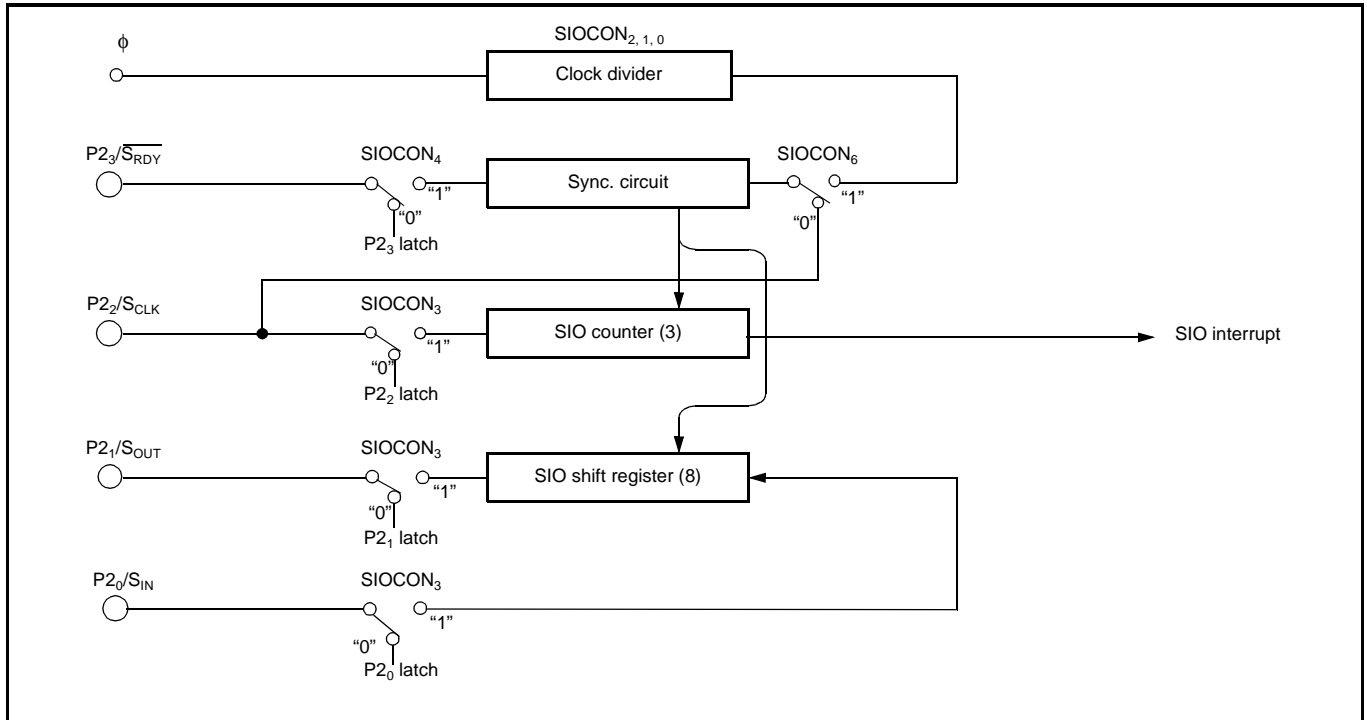


Fig. 25 Block diagram of clock synchronous I/O (ϕ is internal system clock)

(1) Clock synchronous serial I/O operation

Either an internal or external transfer clock can be selected by bit 6 of SIOCON. The internal clock divider can be programmed by bits 0 to 2 of SIOCON. Bit 3 of SIOCON determines whether the double function pins P20 to P22 will act as I/O ports or serve as SIO pins. Bit 4 of SIOCON allows the same selection for pin P23.

When an internal transfer clock is selected, transmission can be triggered by writing data to the SIO shift register (SIO, address 0012₁₆).

After an 8-bit transmission has been completed, the S_{OUT} pin will change to high impedance and the SIO interrupt request bit will be set to "1".

When an external transfer clock is selected, the SIO interrupt request bit will be set to "1" after 8 cycles but the contents of the SIO shift register continue to be shifted while the transfer clock is being input. Therefore, the clock needs to be controlled externally; the S_{OUT} pin will not change to high impedance automatically.

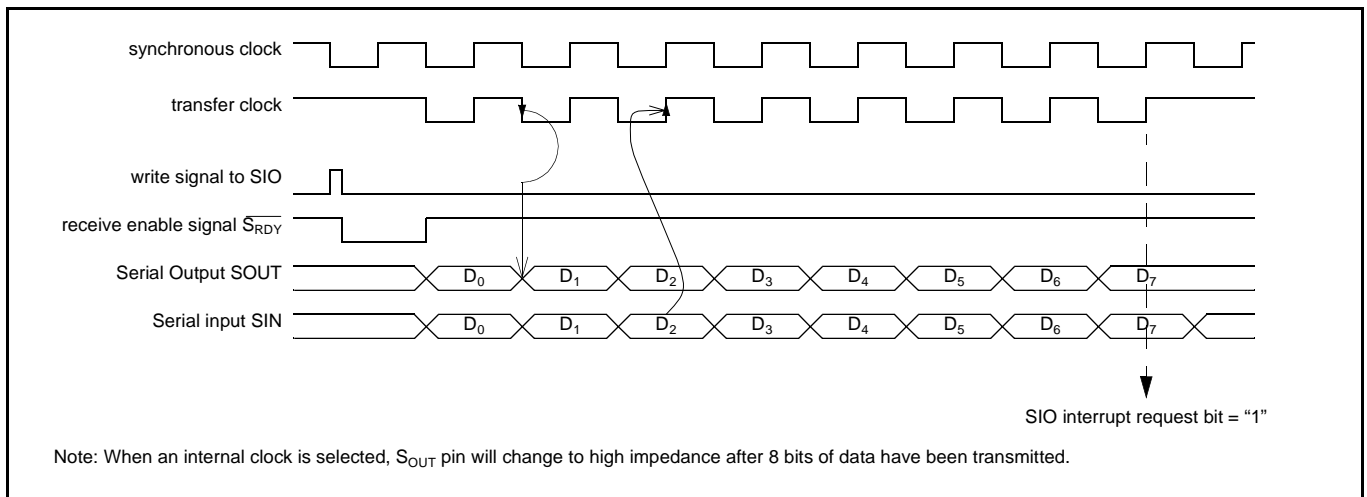


Fig. 26 Timing of clock synchronous SIO function (LSB first selected)

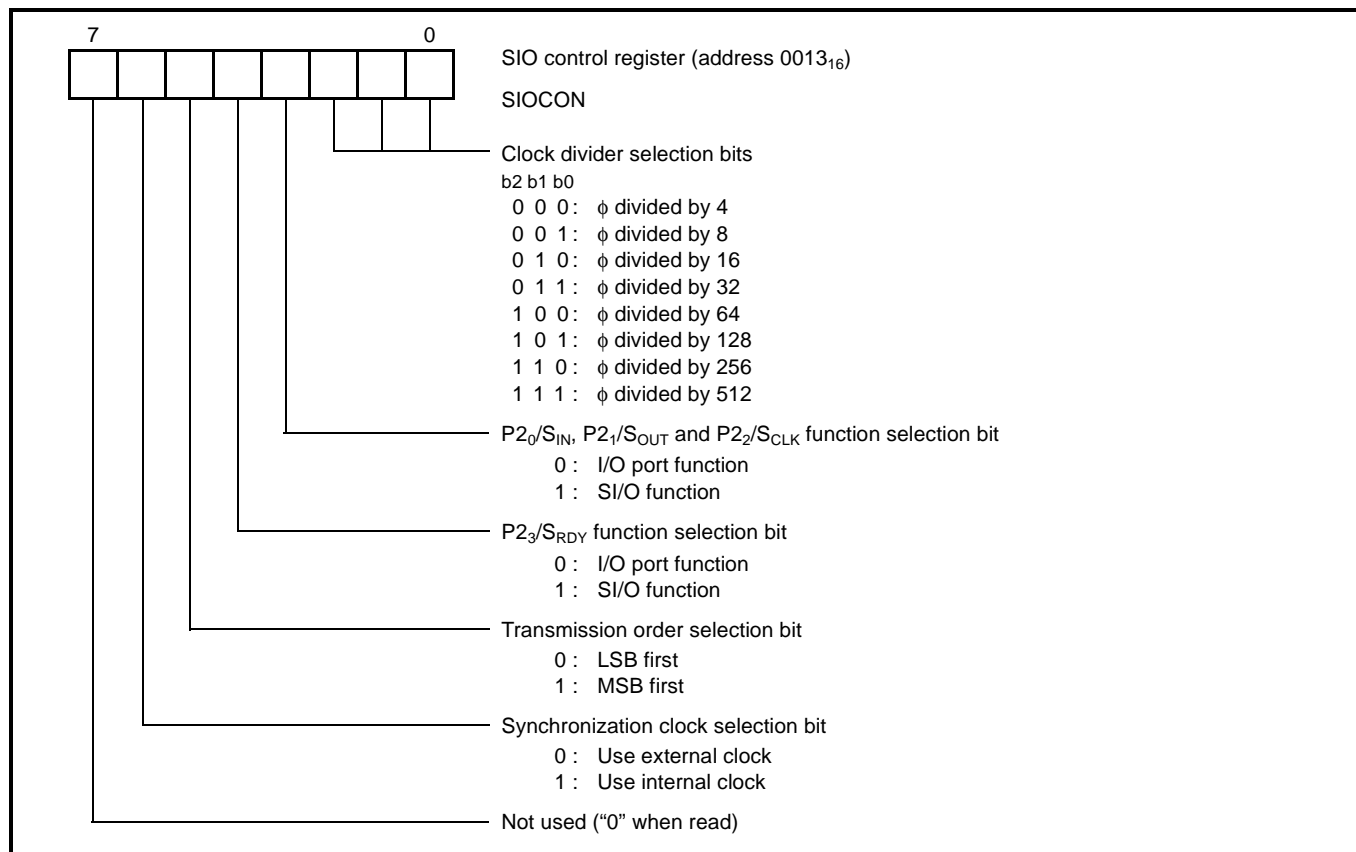


Fig. 27 Structure of serial I/O control register (ϕ is internal system clock)

Clock Asynchronous Serial I/O (UART)

The UART is a full duplex asynchronous transmit/receive unit. The built-in clock divider and baud rate generator enable a broad range of transmission speeds. Please refer to Block diagram of UART.

(1) Description

The transmit and receive shift registers have a buffer (consisting of high and low order byte) each. Since the shift registers cannot be written to or read from directly, transmit data is written to the transmit buffer and receive data is read from the receive buffer. A transmit or receive operation will be triggered by the transmit enable bit and receive enable bit of the UART control register UCON (see Structure of UART control register). The double function terminals P2₅/UT_xD, P2₆/URT_S and P2₄/UR_xD, P2₇/UCT_S will be switched to the UART pins automatically.

(2) Baud rate selection

The baud rate of transmission and reception is determined by the setting of the prescaler and the contents of the UART baud rate generator register. It is calculated by:

$$b = \frac{\phi}{16 \cdot p \cdot (n + 1)}$$

where p is the division ratio of the prescaler and n is the contents of the UART baud rate generator register. The prescalers division ration can be selected by the UART mode register (see page 1-31).

UART mode register (UMOD, Structure of UART mode register)

The UART mode register allows to select the transmission and reception format with the following options:

- word length: 7, 8 or 9 bits
- parity: none, odd or even
- stop bits: 1 or 2

It allows to select the prescalers division ratio as well.

UART baud rate generator (UBRG)

This 8 bit register allows to select the baud rate of the UART (see above). Set this register to the desired value before enabling reception or transmission.

UART control register (UCON, Structure of UART control register)

The UART control register consists of four control bits (bit 0 to bit 3) which allow to control reception and transmission.

UART status register (USTS, Structure of UART status register)

The read-only UART status register consists of 7 bits (bit 0 to bit 6) which indicate the operating status of the UART function and various errors.

HARDWARE

SERIAL I/Os

(3) Handshaking signals

When used as transmitter, the UART will recognize the clear-to-send signal via $P2_7/\overline{UCTS}$ terminal for handshaking. When used as receiver it will issue a request-to-send signal through $P2_6/\overline{URTS}$ pin.

Clear-to-send input

When used as a transmitter (transmit enable bit set to "1"), the UART starts transmission after recognizing "L" level on $P2_7/\overline{UCTS}$. After started, the UART will continue to transmit regardless of the actual level of $P2_7/\overline{UCTS}$ or status of the transmit enable bit.

Request-to-send output

The UART controls the $P2_6/\overline{URTS}$ output according to the following conditions.

Table 7: Output control conditions

Condition	$P2_6/\overline{URTS}$
Receive enable bit is set to "1"	"L"
Reception completed during receive enable bit set to "1"	
Start bit (falling edge) detected	"H"
Receive enable bit is set to "0" before reception started	
Hardware reset	
Receive initialization bit is set to "1"	

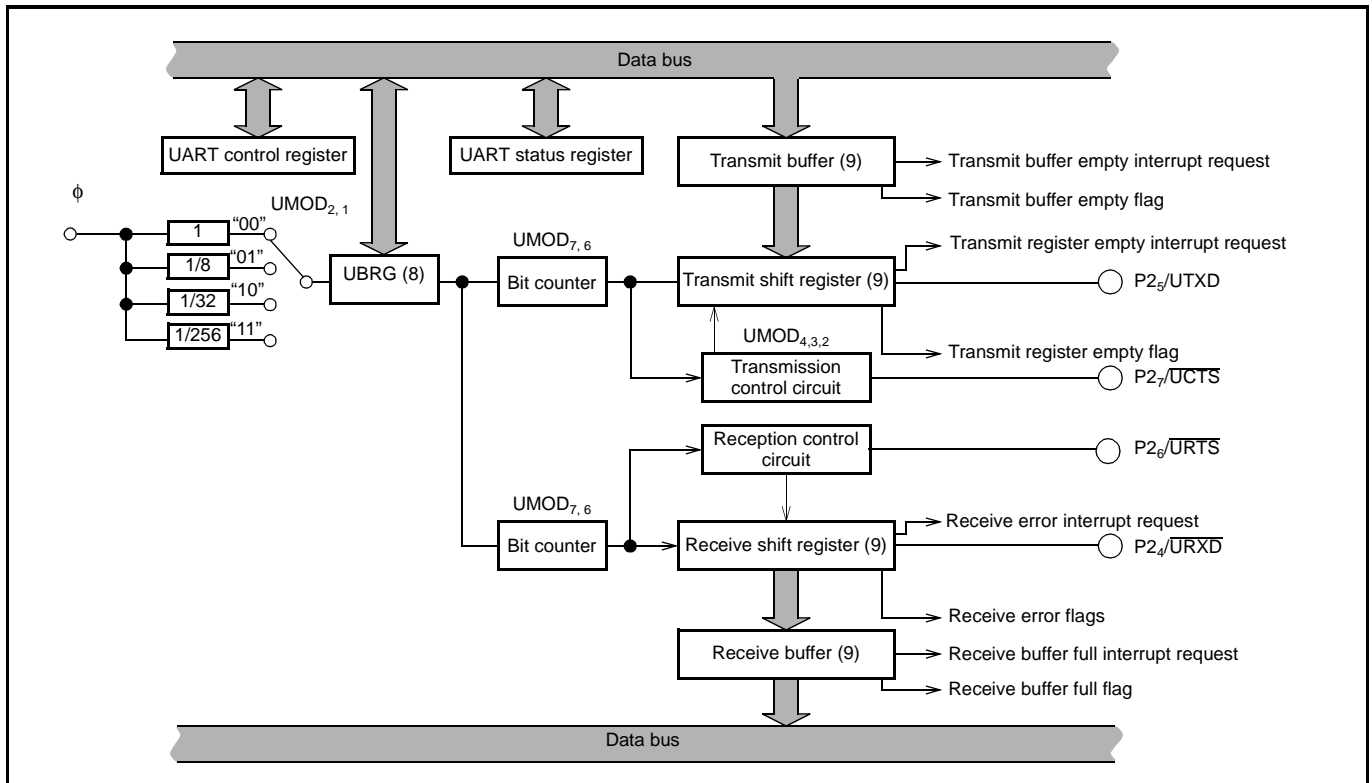


Fig. 28 Block diagram of UART

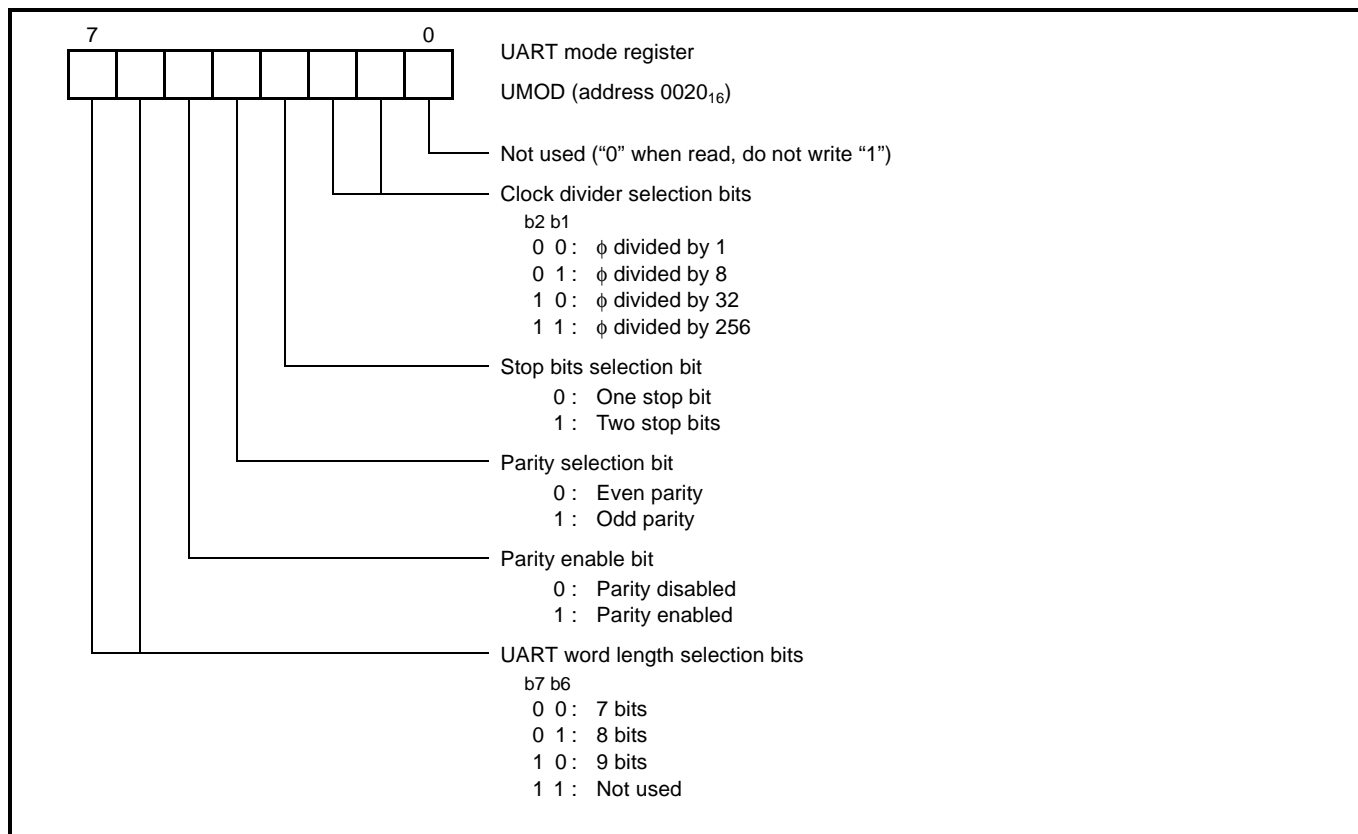


Fig. 29 Structure of UART mode register

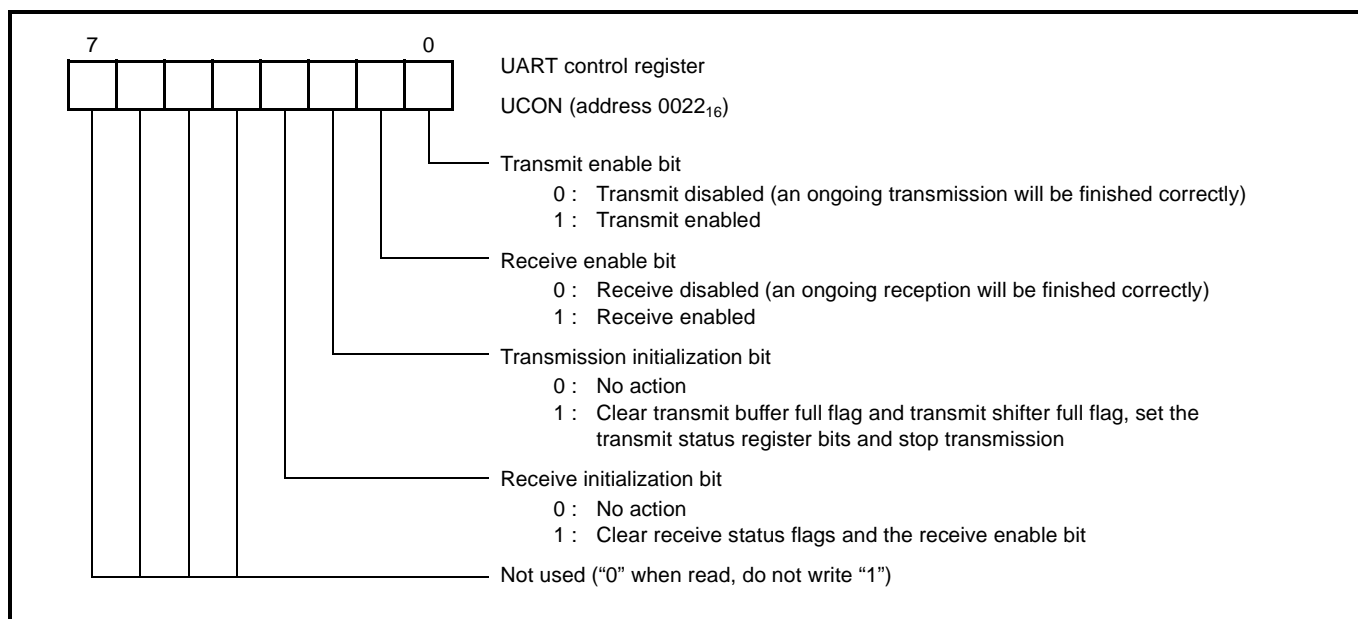


Fig. 30 Structure of UART control register

HARDWARE

SERIAL I/Os

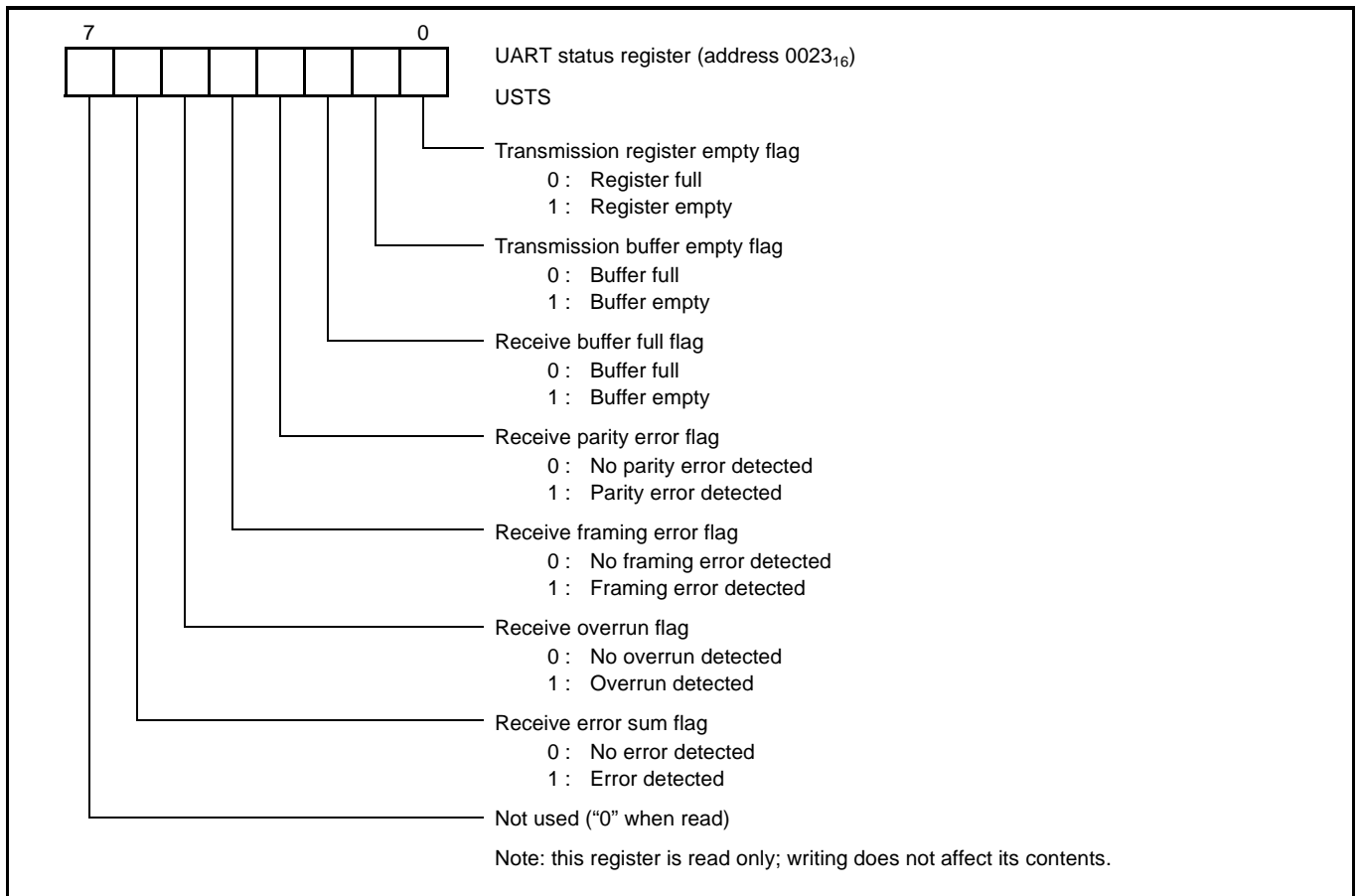


Fig. 31 Structure of UART status register

CAN MODULE

The CAN (Controller Area Network) interface of the 7630 group complies with the 2.0B specification, enabling reception and transmission of frames with either 11- or 29- bit identifier length. Refer to Fig. 33 for a block diagram of the CAN interface.

The programmer's interface to the CAN module is formed by three status/control registers (Fig. 34, Fig. 35, Fig. 36), two bus timing control registers (Fig. 37 Fig. 38), several registers for acceptance filtering (Fig. 39), the transmit and receive buffer registers (Fig. 40) and one dominant level control bit (Fig. 24).

Baud Rate Selection

A programmable clock prescaler is used to derive the CAN module's basic clock from the internal system clock frequency (ϕ). Bit 0 to bit 3 of the CAN bus timing control register represent the prescaler allowing a division ratio from 1 to 1/16 to be selected. So the CAN module basic clock frequency f_{CANB} can be calculated as follows:

$$f_{CANB} = \frac{\phi}{p + 1}$$

where p is the value of the prescaler (selectable from 1 to 15). The effective baud rate of the CAN bus communication depends on the CAN bus timing control parameters and will be explained below.

CAN Bus Timing Control

Each bit-time consists of four different segments (see Fig. 32):

- Synchronization segment (SS),
- Propagation time segment (PTS),
- Phase buffer segment 1 (PBS1) and
- Phase buffer segment 2 (PBS2).

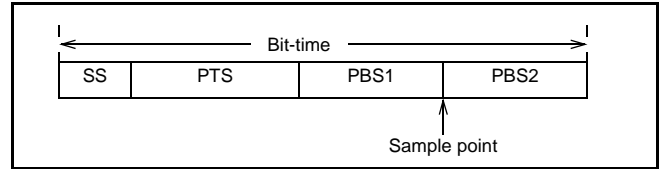


Fig. 32 Bit time of CAN module

The first of these segments is of fixed length (one Time Quantum) and the latter three can be programmed to be 1 to 8 Time Quanta by the CAN bus timing control register 1 and 2 (see Fig. 37 and Fig. 38). The whole bit-time has to consist of minimum 8 and maximum 25 Time Quanta. The duration of one Time Quantum is the cycle time of f_{CANB} . For example, assuming $\phi = 5$ MHz, $p = 0$, one Time Quantum will be 200 ns long. This allows the maximum transmission rate of 625 kb/s to be reached (assuming 8 Time Quanta per bit-time).

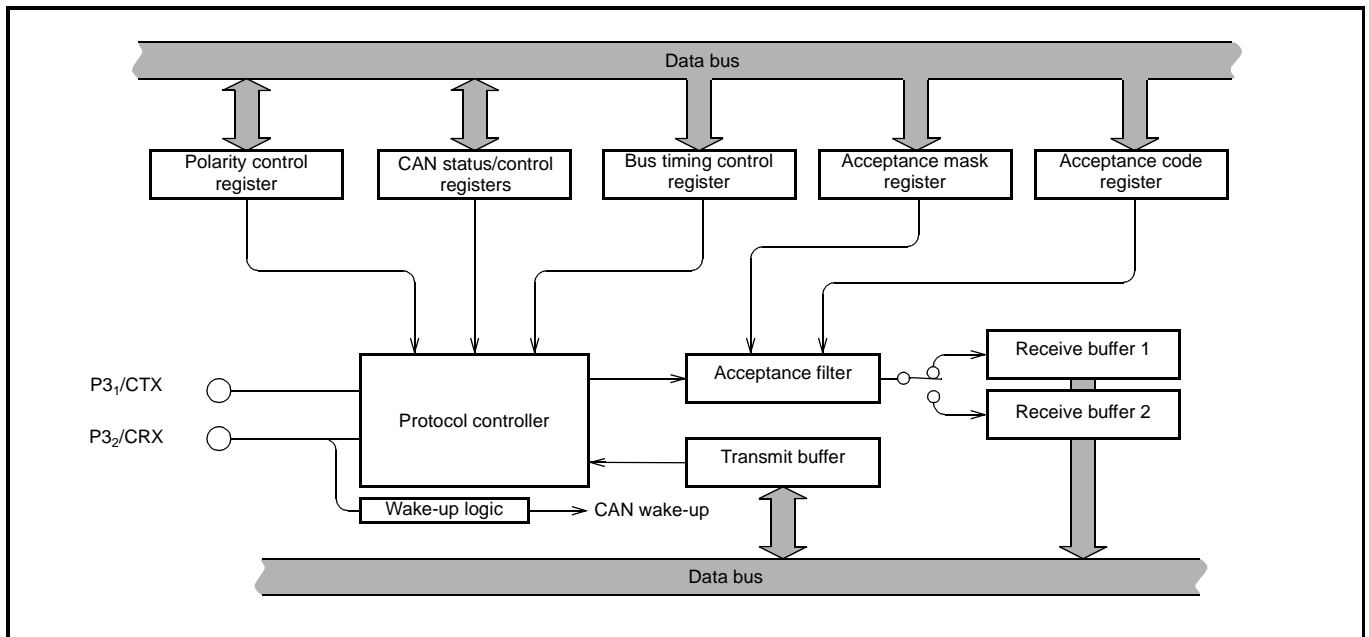


Fig. 33 Block diagram of CAN module

HARDWARE

CAN MODULE

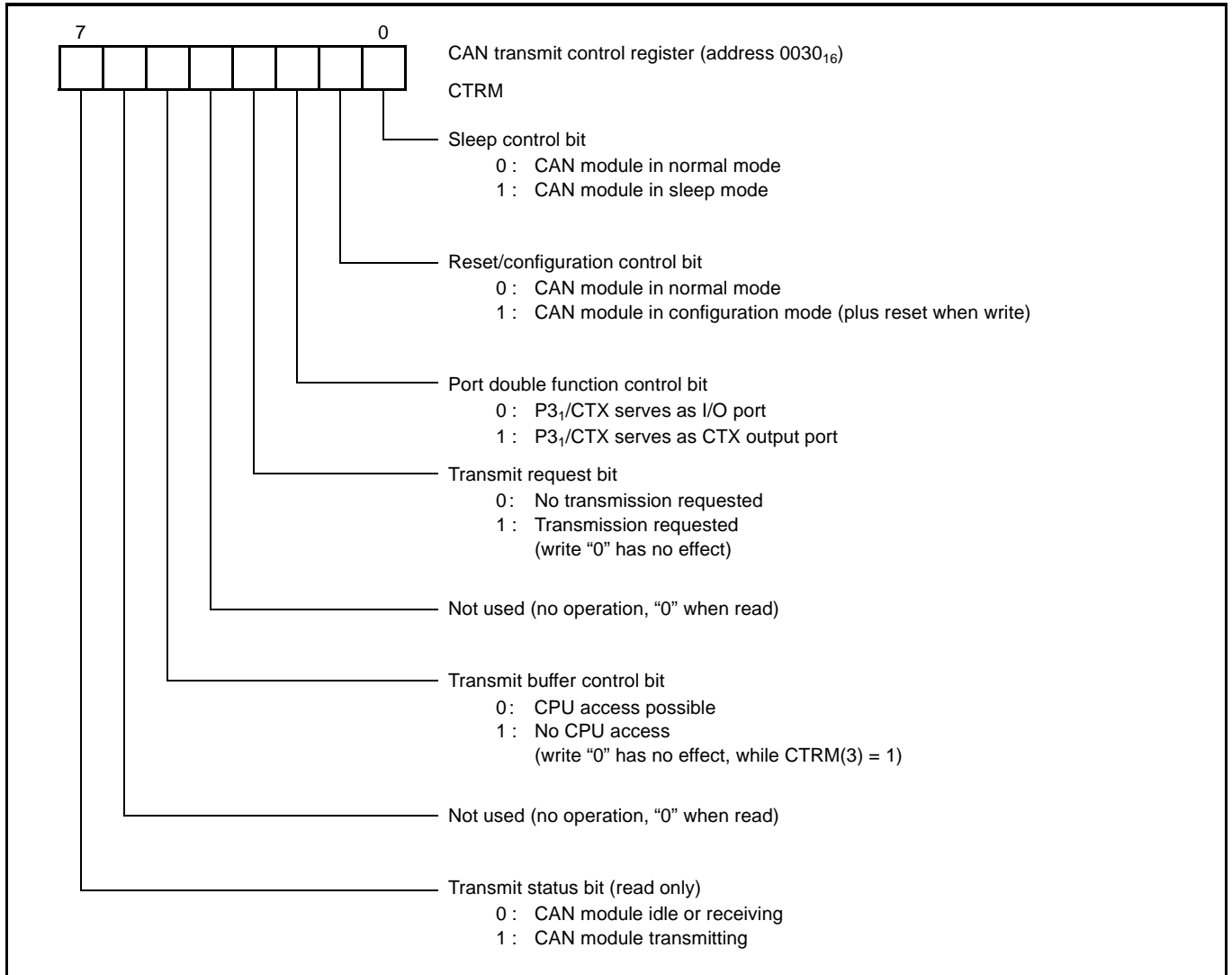


Fig. 34 Structure of CAN transmit control register

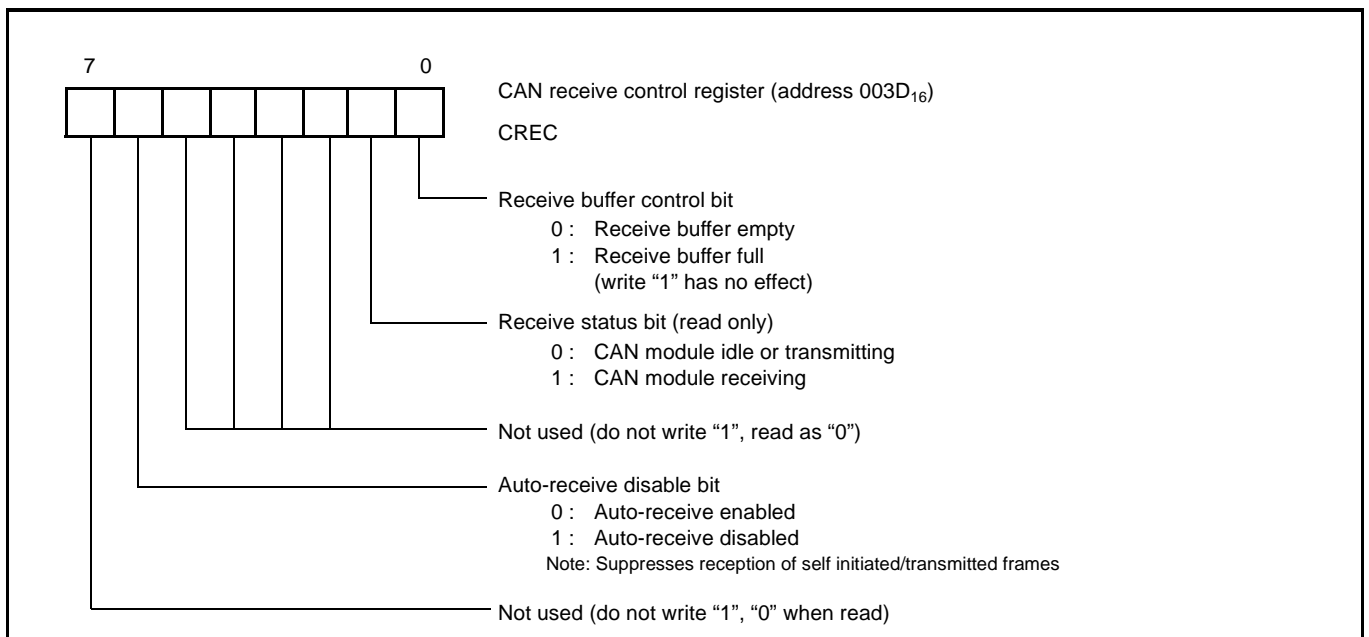


Fig. 35 Structure of CAN receive control register

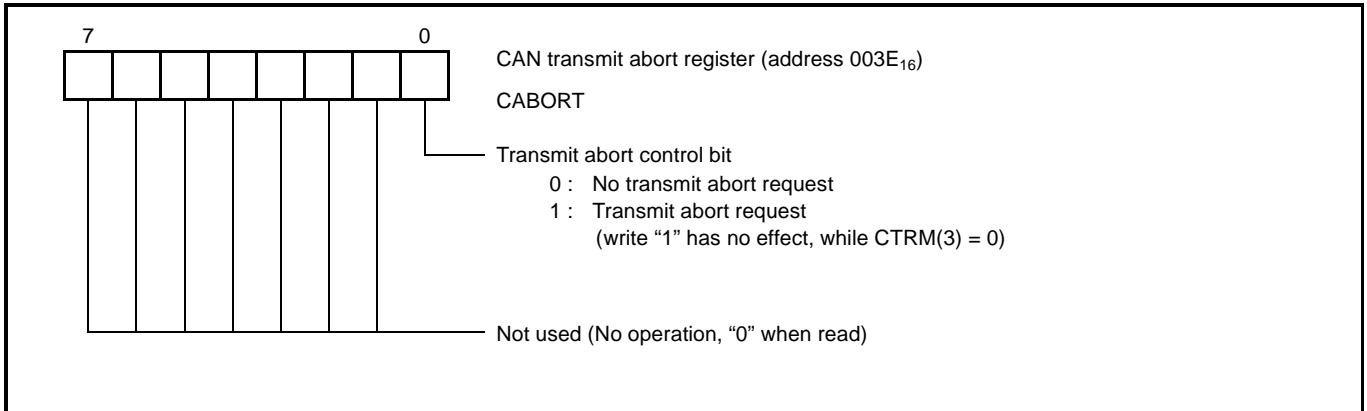


Fig. 36 Structure of CAN transmit abort register

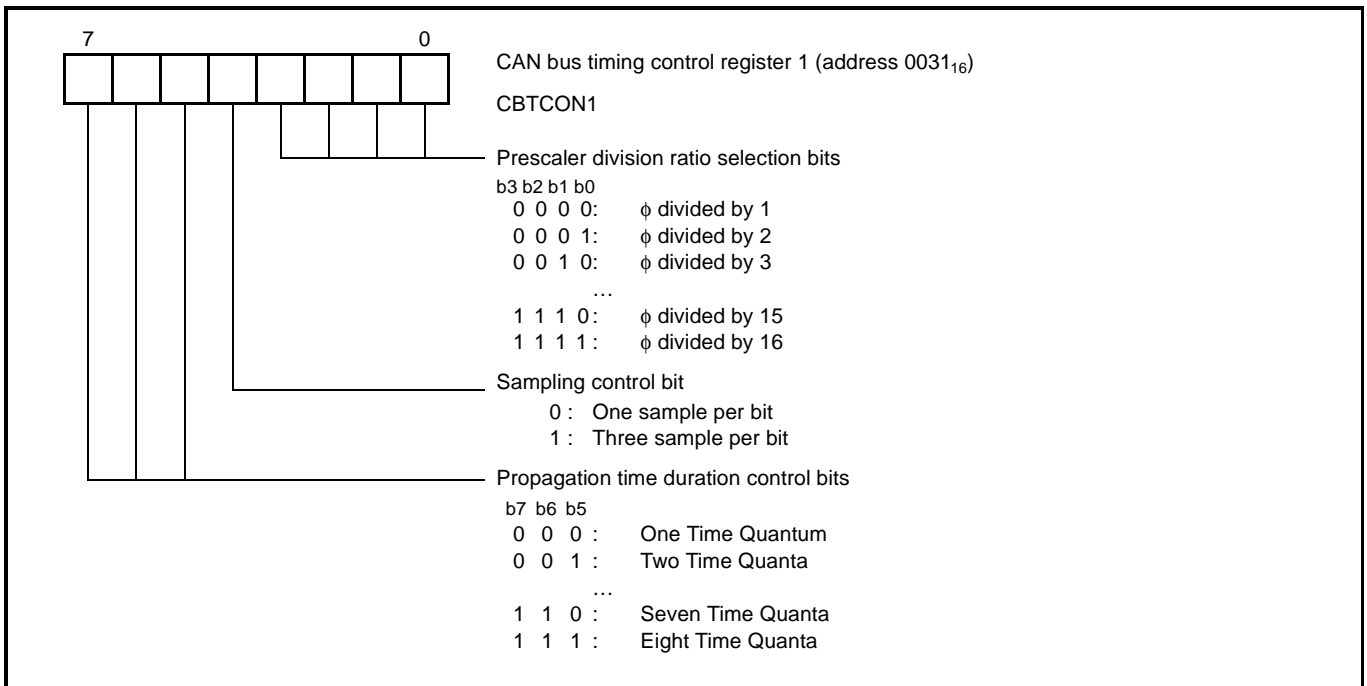


Fig. 37 Structure of CAN bus timing control register 1

HARDWARE

CAN MODULE

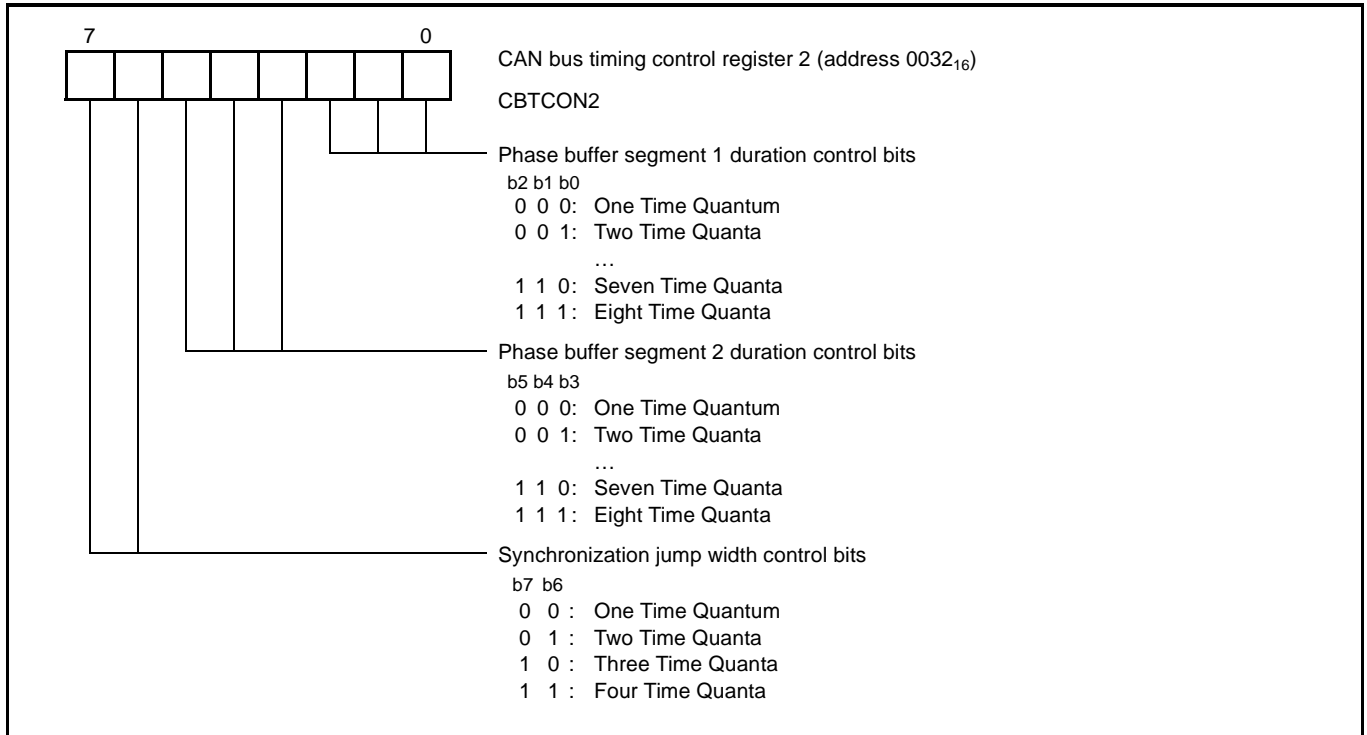


Fig. 38 Structure of CAN bus timing control register 2

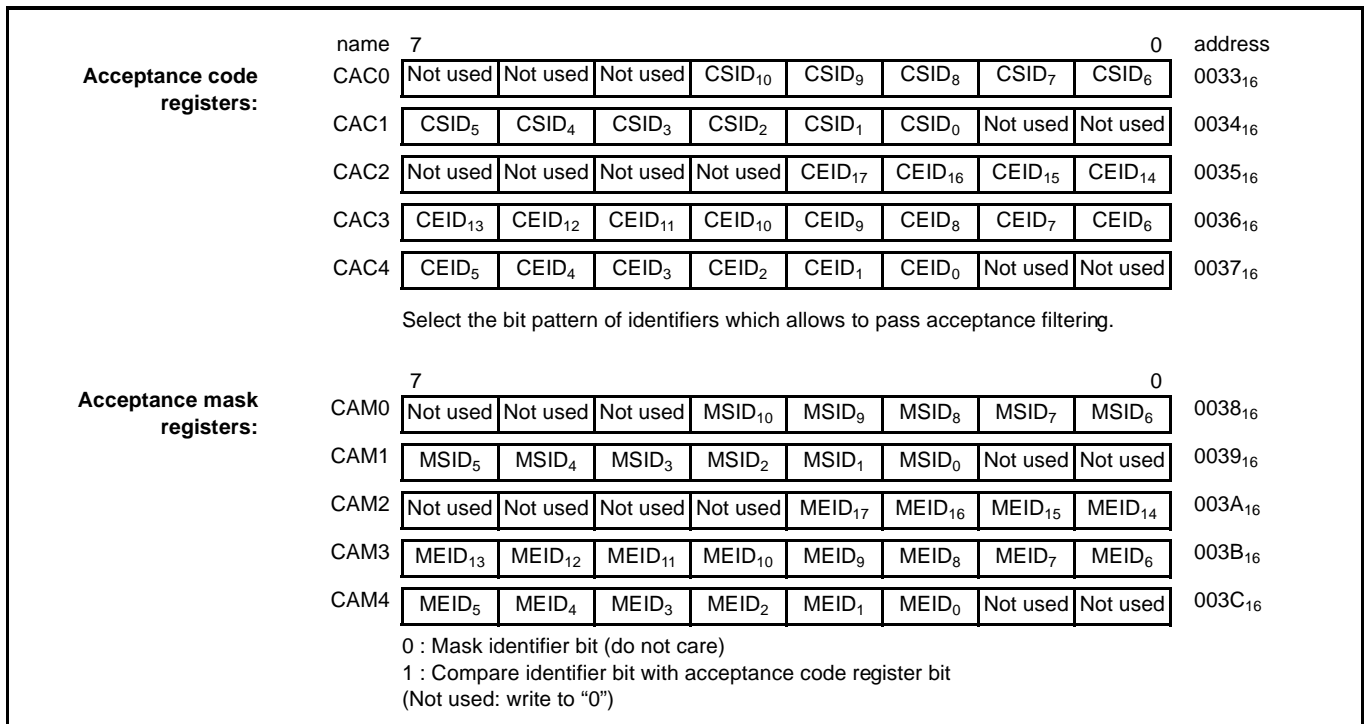


Fig. 39 Structure of CAN mask and code registers

name	7							0	offset
CTB0, CRB0	Not used	Not used	Not used	SID ₁₀	SID ₉	SID ₈	SID ₇	SID ₆	0000 ₁₆
CTB1, CRB1	SID ₅	SID ₄	SID ₃	SID ₂	SID ₁	SID ₀	RTR/SRR	IDE	0001 ₁₆
CTB2, CRB2	Not used	Not used	Not used	Not used	EID ₁₇	EID ₁₆	EID ₁₅	EID ₁₄	0002 ₁₆
CTB3, CRB3	EID ₁₃	EID ₁₂	EID ₁₁	EID ₁₀	EID ₉	EID ₈	EID ₇	EID ₆	0003 ₁₆
CTB4, CRB4	EID ₅	EID ₄	EID ₃	EID ₂	EID ₁	EID ₀	RTR	r1	0004 ₁₆
CTB5, CRB5	Not used	Not used	Not used	r ₀	DLC ₃	DLC ₂	DLC ₁	DLC ₀	0005 ₁₆
CTB6, CRB6	Data byte 0								0006 ₁₆
CTB7, CRB7	Data byte 1								0007 ₁₆
CTB8, CRB8	Data byte 2								0008 ₁₆
CTB9, CRB9	Data byte 3								0009 ₁₆
CTBA, CRBA	Data byte 4								000A ₁₆
CTBB, CRBB	Data byte 5								000B ₁₆
CTBC, CRBC	Data byte 6								000C ₁₆
CTBD, CRBD	Data byte 7								000D ₁₆

Calculate the actual address as follows:
 TxD buffer address = 0040₁₆ + offset
 RxD buffer address = 0050₁₆ + offset
 (Not used: write to "0")

Fig. 40 Structure of CAN transmission and reception buffer registers

Note 1: All CAN related SFRs must not be written in "CAN sleep" mode.

HARDWARE

A-D CONVERTER

A-D CONVERTER

The A-D converter uses the successive approximation method with 8 bit resolution. The functional blocks of the A-D converter are described below. Refer to Block diagram of A-D converter.

Comparison Voltage Generator

The comparison voltage generator divides the voltage between AV_{SS} and V_{REF} by 256, and outputs the divided voltage.

Channel Selector

The channel selector selects one of ports PO_0/AN_0 to PO_7/AN_7 , and inputs its voltage to the comparator.

A-D conversion register AD

The A-D conversion register is a read-only register that stores the result of an A-D conversion. This register must not be read during an A-D conversion.

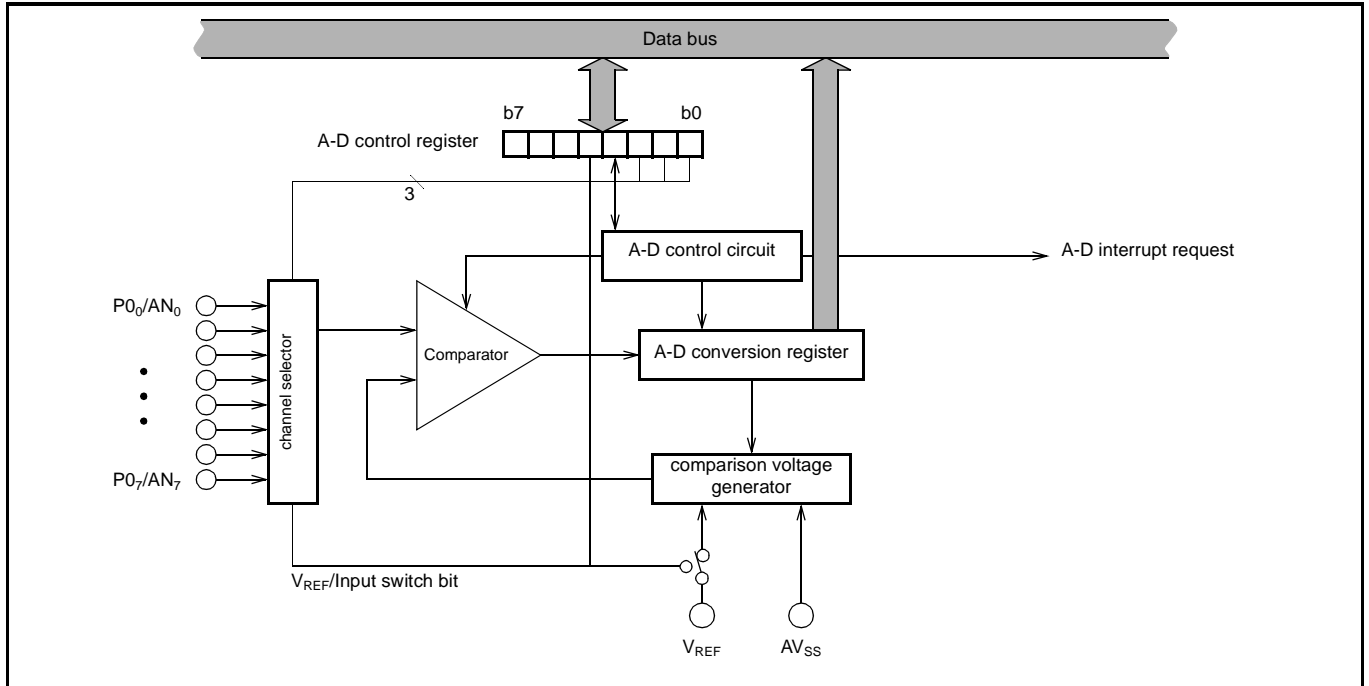


Fig. 41 Block diagram of A-D converter

A-D control register (Structure of A-D control register)

The A-D control register controls the A-D conversion process. Bits 0 to 2 select a specific analog input pin. Bit 3 signals the completion of an A-D conversion. The value of this bit remains "0" during an A-D

conversion, and changes to "1" when an A-D conversion ends. Writing "0" to this bit starts the A-D conversion. Bit 4 is the V_{REF} /Input switch bit.

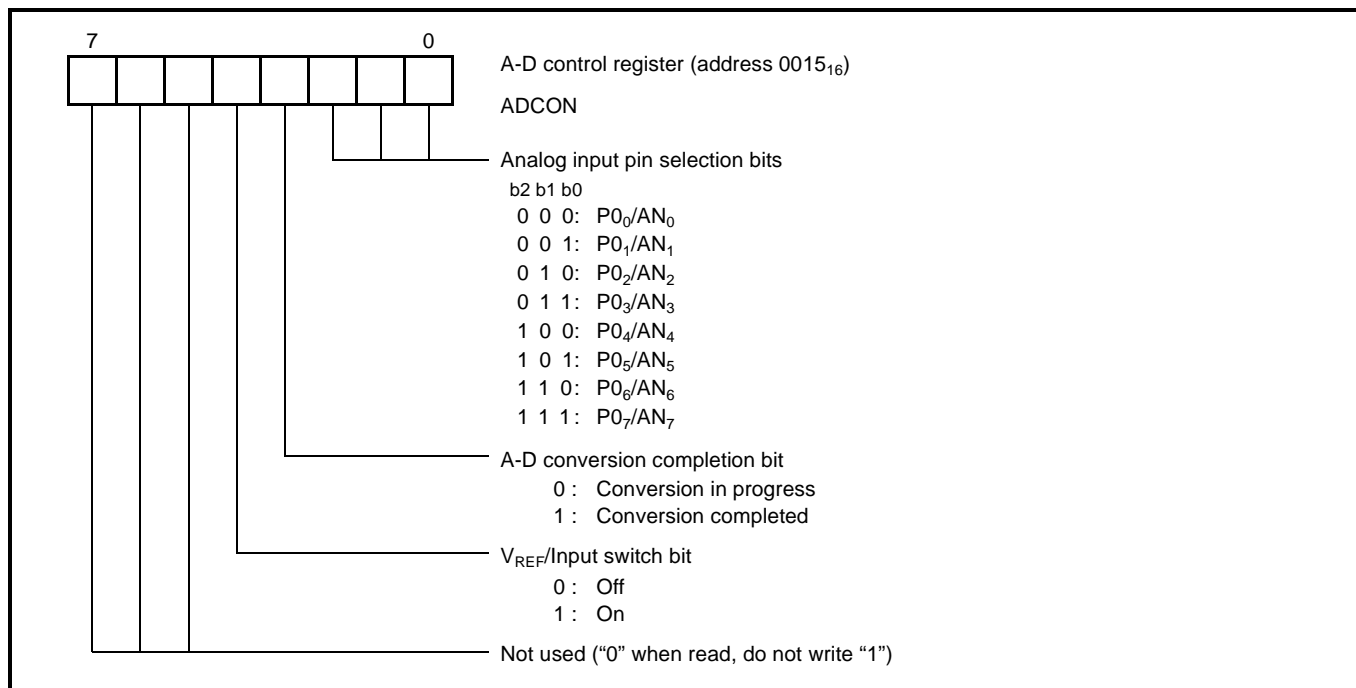


Fig. 42 Structure of A-D control register

A-D Converter Operation

The comparator and control circuit reference an analog input voltage with the reference voltage, then stores the result in the A-D conversion register. When an A-D conversion is complete, the control circuit sets the A-D conversion completion bit and the A-D interrupt request

bit to "1". The result of A-D conversion can be obtained from the A-D conversion register, AD (address 0014₁₆).

Note that the comparator is linked to a capacitor, so set $f(X_{IN})$ to 500 kHz or higher during A-D conversion.

HARDWARE

WATCHDOG TIMER

WATCHDOG TIMER

The watchdog timer consists of two separate counters: one 7-bit counter (WD_H) and one 4-bit counter (WD_L). Cascading both counters or using the high-order counter allows only to select the time-out from either 524288 or 32768 cycles of the internal clock ϕ . Refer to Fig. 43 and Fig. 44. Both counters are addressed by the same watchdog timer register (WDT). When writing to this register, both counters will be set to the following default values:

- the high-order counter will be set to address $7F_{16}$
- the low-order counter will be set to address F_{16}

regardless of the data written to the WDT register. Reading the watchdog timer register will return the corresponding control bit status, not the counter contents.

Once the WDT register is written to, the watchdog timer starts counting down and the watchdog timer interrupt is enabled. Once it is running, the watchdog timer cannot be disabled or stopped except by reset. On a watchdog timer underflow, a non-maskable watchdog timer interrupt will be requested.

To prevent the system being stopped by STP instruction, this instruction can be disabled by the STP instruction disable bit of WDT register. Once the STP instruction is disabled, it cannot be enabled again except by RESET.

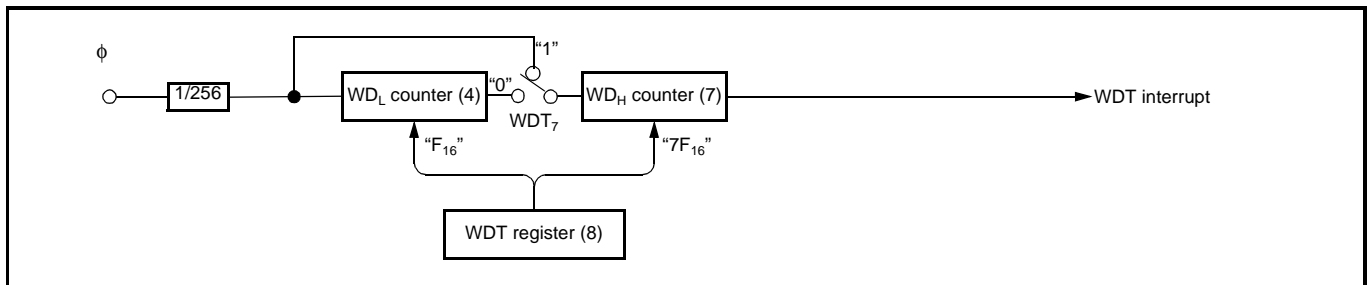


Fig. 43 Block diagram of watchdog timer

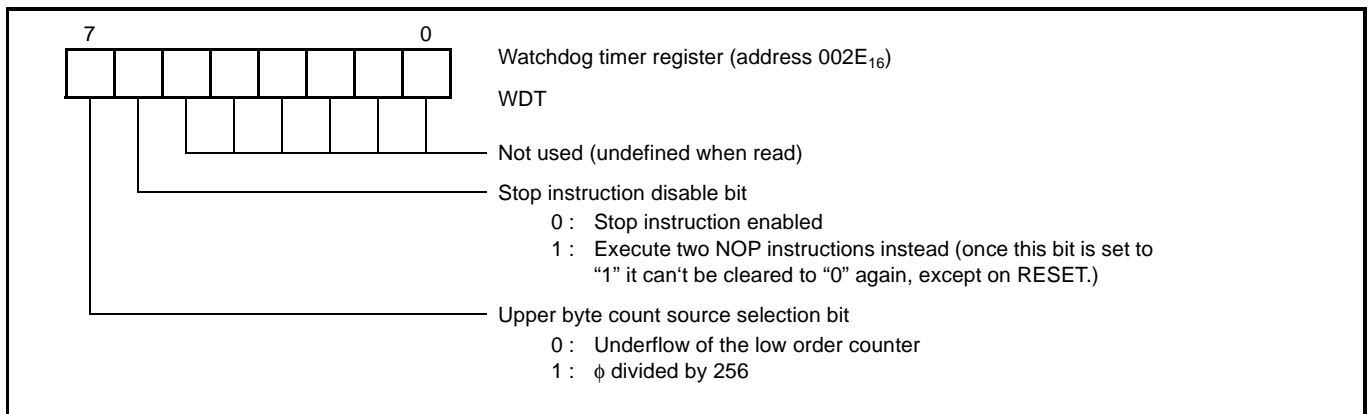


Fig. 44 Structure of watchdog timer register (ϕ is internal clock system)

RESET CIRCUIT

The 7630 group is reset according to the sequence shown in Fig. 46. It starts program execution from the address formed by the contents of the addresses $FFFB_{16}$ and $FFFA_{16}$, when the \overline{RESET} pin is held at

"L" level for more than $2\ \mu\text{s}$ while the power supply voltage is in the recommended operating condition and then returned to "H" level. Refer to Fig. 45 for an example of the reset circuit.

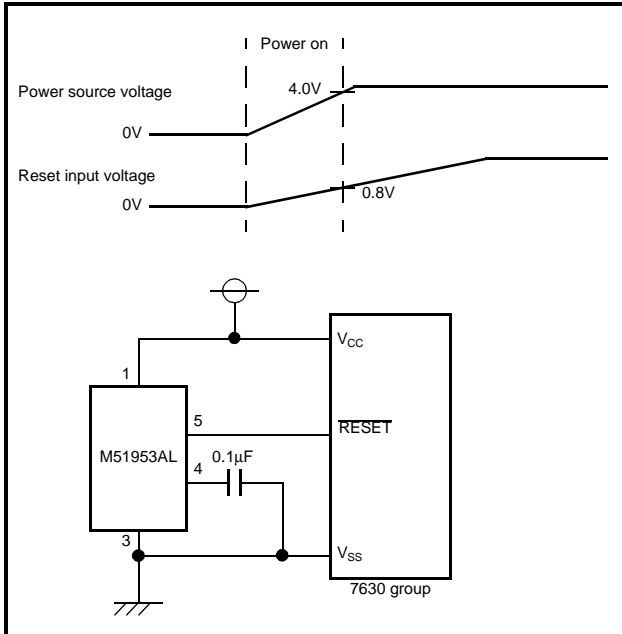


Fig. 45 Example of reset circuit

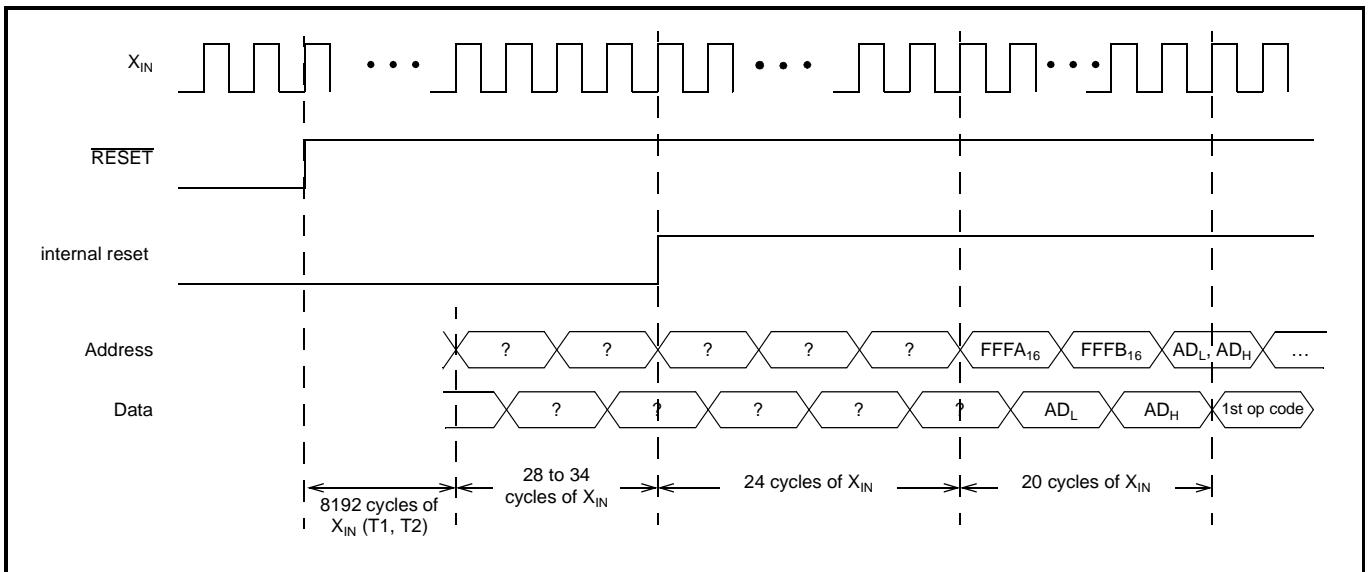


Fig. 46 Reset sequence

HARDWARE

RESET CIRCUIT

Register	Address	Register contents	Register	Address	Register contents
CPU mode reg.	0000 ₁₆	48 ₁₆	Timer XH	001B ₁₆	FF ₁₆
Interrupt request reg. A	0002 ₁₆	00 ₁₆	Timer YL	001C ₁₆	FF ₁₆
Interrupt request reg. B	0003 ₁₆	00 ₁₆	Timer YH	001D ₁₆	FF ₁₆
Interrupt request reg. C	0004 ₁₆	00 ₁₆	Timer X mode reg.	001E ₁₆	00 ₁₆
Interrupt control reg. A	0005 ₁₆	00 ₁₆	Timer Y mode reg.	001F ₁₆	00 ₁₆
Interrupt control reg. B	0006 ₁₆	00 ₁₆	UART mode reg.	0020 ₁₆	00 ₁₀
Interrupt control reg. C	0007 ₁₆	00 ₁₆	UART control reg.	0022 ₁₆	00 ₁₆
Port P0 reg.	0008 ₁₆	00 ₁₆	UART status reg.	0023 ₁₆	07 ₁₀
Port P0 direction reg.	0009 ₁₆	00 ₁₆	Port P0 pull-up control reg.	0028 ₁₆	00 ₁₆
Port P1 reg.	000A ₁₆	00 ₁₆	Port P1 pull-up control reg.	0029 ₁₆	00 ₁₆
Port P1 direction reg.	000B ₁₆	00 ₁₆	Port P2 pull-up control reg.	002A ₁₆	00 ₁₆
Port P2 reg.	000C ₁₆	00 ₁₆	Port P3 pull-up control reg.	002B ₁₆	00 ₁₆
Port P2 direction reg.	000D ₁₆	00 ₁₆	Port P4 pull-up/down control reg.	002C ₁₆	00 ₁₆
Port P3 reg.	000E ₁₆	00 ₁₆	Interrupt polarity selection reg.	002D ₁₆	00 ₁₆
Port P3 direction reg.	000F ₁₆	00 ₁₆	Watchdog timer reg.	002E ₁₆	3F ₁₆
Port P4 reg.	0010 ₁₆	00 ₁₆	Polarity control reg.	002F ₁₆	00 ₁₆
Port P4 direction reg.	0011 ₁₆	00 ₁₆	CAN transmit control reg.	0030 ₁₆	02 ₁₆
Serial I/O control reg.	0013 ₁₆	00 ₁₆	CAN bus timing control reg. 1	0031 ₁₆	00 ₁₆
A-D control reg.	0015 ₁₆	08 ₁₆	CAN bus timing control reg. 2	0032 ₁₆	00 ₁₆
Timer 1	0016 ₁₆	FF ₁₆	CAN receive control reg.	003D ₁₆	00 ₁₆
Timer 2	0017 ₁₆	01 ₁₆	CAN transmit abort reg.	003E ₁₆	00 ₁₆
Timer 3	0018 ₁₆	FF ₁₆	Processor status reg.	(PS)	04 ₁₆
Timer 123 mode reg.	0019 ₁₆	40 ₁₆	Program counter (high-order byte)	(PCH)	contents of FFFB ₁₆
Timer XL	001A ₁₆	FF ₁₆	Program counter (low-order byte)	(PCL)	contents of FFFA ₁₆

Note: The contents of RAM and registers other than the above registers are undefined after reset; thus software initialization is required.

Fig. 47 Internal status of microcomputer after reset

CLOCK GENERATING CIRCUIT

The 7630 group is equipped with an internal clock generating circuit. Please refer to Fig. 48 for a circuit example using a ceramic resonator or quartz crystal oscillator. For the capacitor values, refer to the manufacturers recommended parameters which depend on each oscillators characteristics. When using an external clock, input it to the X_{IN} pin and leave X_{OUT} open.

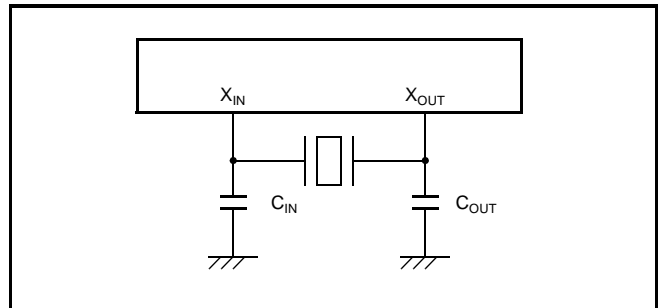


Fig. 48 Ceramic resonator circuit

Oscillation Control

The 7630 group has two low power modes: the stop and the wait mode.

Stop mode

The microcomputer enters the stop mode by executing the STP instruction. The oscillator stops with the internal clock ϕ at "H" level. Timers 1 and 2 will be cascaded and initialized by their reload latches contents. The count source for timer 1 will be set to $f(X_{IN})/16$.

Oscillation is restarted if an external interrupt is accepted or at reset. When using an external interrupt, the internal clock ϕ remains at "H" level until timer 2 underflows allowing a time-out until the clock oscil-

lation becomes stable. When using reset, a fixed time-out will be generated allowing oscillation to stabilize.

Wait mode

The microcomputer enters the wait mode by executing the WIT instruction. The internal clock ϕ stops at "H" level while the oscillator keeps running.

Recovery from wait mode can be done in the same way as from stop mode. However, the time-out period mentioned above is not required to return from wait-mode, thus no such time-out mechanism has been implemented.

Note: Set the interrupt enable bit of the interrupt source to be used to return from stop or wait mode to "1" before executing STP or WIT instruction.

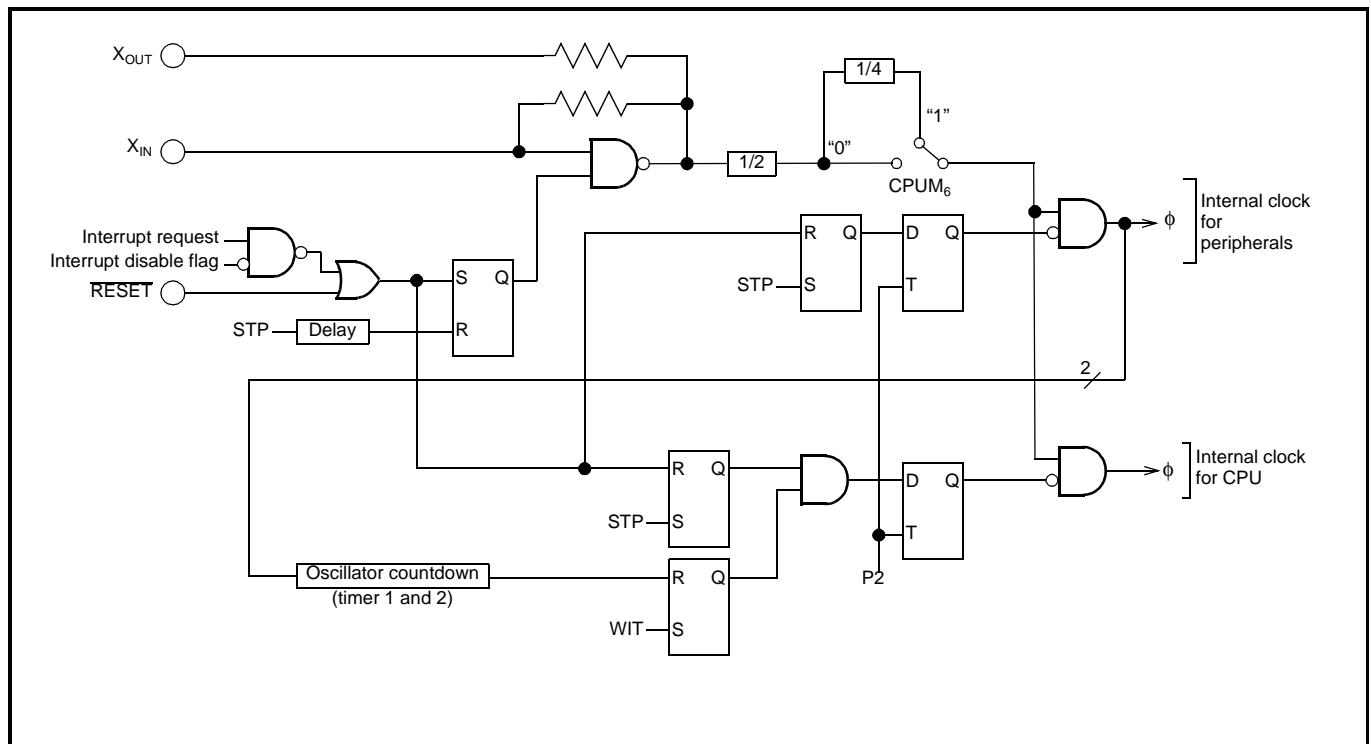


Fig. 49 Block diagram of clock generating circuit

HARDWARE

DATA REQUIRED FOR MASK ORDERS

DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

- 1 Mask ROM Order Confirmation Form
- 2 Mask Specification Form
- 3 Contents of Mask ROM, in EPROM form (three identical copies)

PROM PROGRAMMING METHOD

The built-in PROM of the blank One Time PROM version and built-in EPROM version can be read or programmed with a general purpose PROM programmer using a special programming adapter. Set the address of PROM programmer to the user ROM area.

For the programming adapter type name, please refer to the following table:

Table 8: Programming adapter name

MCU type	Package	Programming adapter type
One Time PROM	44P6N-A	PCA7430
EPROM	80D0	PCA7431

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Fig. 50 is recommended to verify programming.

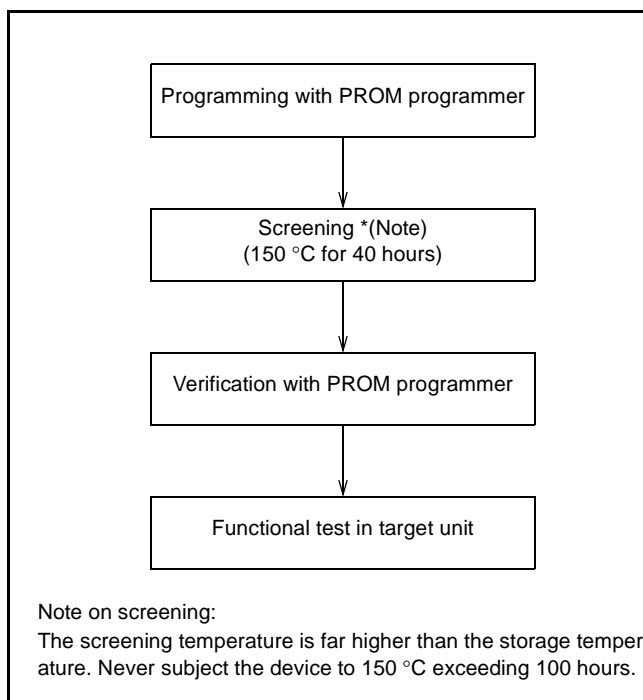


Fig. 50 Programming and testing of One Time PROM version

Table 9: ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Conditions	Ratings	Unit
V_{CC}	Power source voltage	All voltages with respect to V_{SS} and output transistors are "off".	-0.3 to 7.0	V
V_I	Input voltage P0 ₀ —P0 ₇ , P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , RESET, X _{IN}		-0.3 to $V_{CC} + 0.3$	V
V_O	Output voltage P0 ₀ —P0 ₇ , P1 ₂ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , X _{OUT}		-0.3 to $V_{CC} + 0.3$	V
P_d	Power dissipation	T _a = 25 °C	500	mW
T _{opr}	Operating temperature		-40 to 85	°C
T _{stg}	Storage temperature		-60 to 150	°C

Table 10: RECOMMENDED OPERATING CONDITIONS

 ($V_{CC} = 4.0$ to 5.5 V, $V_{SS} = AV_{SS} = 0$ V, T_a = -40 to 85 °C unless otherwise noted)

Symbol	Parameter	Limits			Unit
		min.	typ.	max.	
V_{CC}	Power source voltage	4.0	5.0	5.5	V
V_{SS}			0		V
V_{IH}	"H" Input voltage P0 ₀ —P0 ₇ , P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , RESET, X _{IN}	$0.8 \cdot V_{CC}$		V_{CC}	V
V_{IL}	"L" Input voltage P0 ₀ —P0 ₇ , P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , RESET, X _{IN}	0		$0.2 \cdot V_{CC}$	V
$\sum I_{OH}$ (peak)	"H" sum peak output current P0 ₀ —P0 ₇ , P1 ₂ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇			-80	mA
$\sum I_{OH}$ (avg)	"H" sum average output current			-40	mA
$\sum I_{OL}$ (peak)	"L" sum peak output current			80	mA
$\sum I_{OL}$ (avg)	"L" sum average output current			40	mA
I_{OH} (peak)	"H" peak output current			-10	mA
I_{OH} (avg)	"H" average output current			-5	mA
I_{OL} (peak)	"L" peak output current			10	mA
I_{OL} (avg)	"L" average output current			5	mA
I_{IO}	input current at overvoltage condition ($V_I > V_{CC}$) P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇			1	mA
$\sum I_{IO}$	total input current at overvoltage condition ($V_I > V_{CC}$) P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇			16	mA
f(CNTR)	Timer input frequency (based on 50 % duty)	P1 ₄ /CNTR ₀ , P1 ₅ /CNTR ₁ (except bi-phase counter mode)		f(X _{IN})/16	MHz
		P1 ₃ /TX ₀ , P1 ₄ /CNTR ₀ (bi-phase counter mode)		f(X _{IN})/32	MHz
f(X _{IN})	Clock input oscillation frequency			10	MHz

HARDWARE

ELECTRICAL CHARACTERISTICS

Table 11: ELECTRICAL CHARACTERISTICS

($V_{CC} = 4.0$ to 5.5 V, $V_{SS} = AV_{SS} = 0$ V, $T_a = -40$ to 85 °C unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			min.	typ.	max.	
V_{OH}	"H" output voltage P0 ₀ —P0 ₇ , P1 ₂ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇	$I_{OH} = -5$ mA	$0.8 \cdot V_{CC}$			V
V_{OL}	"L" output voltage P0 ₀ —P0 ₇ , P1 ₂ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇	$I_{OL} = 5$ mA			2.0	V
$V_{T+} - V_{T-}$	Hysteresis P1 ₁ /INT ₀ , P1 ₂ /INT ₁ , P1 ₃ /TX ₀ , P1 ₄ /CNTR ₀ , P1 ₅ /CNTR ₁ , P2 ₀ /S _{IN} , P2 ₂ /S _{CLK} , P2 ₆ /U _{RTS} , P2 ₇ /U _{CTS} , P3 ₂ /CRX, RESET			0.5		V
I_{IH}	"H" input current P0 ₀ —P0 ₇ , P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , RESET	$V_I = V_{CC}$			5	μA
I_{IH}	"H" input current X_{IN}	$V_I = V_{CC}$		4		μA
I_{IL}	"L" input current P0 ₀ —P0 ₇ , P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , RESET	$V_I = V_{SS}$			-5	μA
I_{IL}	"L" input current X_{IN}	$V_I = V_{SS}$		-4		μA
I_{IH}	"H" input current P3 ₂ , P4 ₀ —P4 ₇	$V_I = V_{CC}$ Pull-Down = 'On'	20		200	μA
I_{IL}	"L" input current P0 ₀ —P0 ₇ , P1 ₁ —P1 ₇ , P2 ₀ —P2 ₇ , P3 ₀ —P3 ₄ , P4 ₀ —P4 ₇ , RESET	$V_I = V_{SS}$ Pull-Up = 'On'	-200		-20	μA
V_{RAM}	RAM hold voltage	When clock stopped	2.0			V
I_{CC}	Power source current	high speed mode, $f(X_{IN}) = 8$ MHz, $V_{CC} = 5$ V, output transistors off, CAN module running, ADC running		11.0	18.0	mA
		high speed mode, $f(X_{IN}) = 8$ MHz, $V_{CC} = 5$ V, output transistors off, CAN module stopped, ADC running		9.0	16.0	mA
		middle speed mode, $f(X_{IN}) = 8$ MHz, $V_{CC} = 5$ V, output transistors off, CAN module running, ADC running		6.0	11.0	mA
		middle speed mode, wait mode, $f(X_{IN}) = 8$ MHz, $V_{CC} = 5$ V, output transistors off, CAN module stopped, ADC stopped		2.0		mA
		stop mode, $f(X_{IN}) = 0$ MHz, $V_{CC} = 5$ V, $T_a = 25$ °C		0.1	1.0	μA
		stop mode, $f(X_{IN}) = 0$ MHz, $V_{CC} = 5$ V, $T_a = 85$ °C			10.0	μA

A-D CONVERTER CHARACTERISTICS

Table 12: A-D converter characteristics

($V_{CC} = 4.0$ to 5.5 V, $V_{SS} = AV_{SS} = 0$ V, $T_a = -40$ to 85 °C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			min.	typ.	max.	
—	Resolution				8	Bit
—	Absolute accuracy			±1.0	±2.5	LSB
t_{CONV}	Conversion time	high-speed mode	106		108	$t_C(X_{IN})$
		middle-speed mode	424		432	$t_C(X_{IN})$
V_{REF}	Reference input voltage		2.0		V_{CC}	V
I_{REF}	Reference input current	$V_{CC} = V_{REF} = 5.12$ V		150	200	μA
R_{LADDER}	Ladder resistor value			35		kΩ
I_{IAN}	Analog input current	$V_I = V_{SS}$ to V_{CC}		0.5	5.0	μA

HARDWARE

TIMING REQUIREMENTS

Table 13: Timing requirements

($V_{CC} = 4.0$ to 5.5 V, $V_{SS} = AV_{SS} = 0$ V, $T_a = -40$ to 85 °C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		min.	typ.	max.	
$t_W(\overline{\text{RESET}})$	Reset input "L" pulse width	2			μs
$t_C(X_{IN})$	External clock input cycle time	100			ns
$t_{WH}(X_{IN})$	External clock input "H" pulse width	37			ns
$t_{WL}(X_{IN})$	External clock input "L" pulse width	37			ns
$t_C(\text{CNTR})$	CNTR ₀ , CNTR ₁ input cycle time (except bi-phase counter mode)	1600			ns
	CNTR ₀ input cycle time (bi-phase counter mode)	2000			ns
$t_{WH}(\text{CNTR})$	CNTR ₀ , CNTR ₁ input "H" pulse width (except bi-phase counter mode)	800			ns
	CNTR ₀ input "H" pulse width (bi-phase counter mode)	1000			ns
$t_{WL}(\text{CNTR})$	CNTR ₀ , CNTR ₁ input "L" pulse width (except bi-phase counter mode)	800			ns
	CNTR ₀ input "L" pulse width (bi-phase counter mode)	1000			ns
$t_L(\text{CNTR}_0\text{-TX}_0)$	Lag of CNTR ₀ and TX ₀ input edges (bi-phase counter mode)	500			ns
$t_C(\text{TX}_0)$	TX ₀ input cycle time (bi-phase counter mode)	3200			ns
$t_{WH}(\text{TX}_0)$	TX ₀ input "H" pulse width (bi-phase counter mode)	1600			ns
$t_{WL}(\text{TX}_0)$	TX ₀ input "L" pulse width (bi-phase counter mode)	1600			ns
$t_{WH}(\text{INT})$	INT ₀ , INT ₁ input "H" pulse width	460			ns
$t_{WL}(\text{INT})$	INT ₀ , INT ₁ input "L" pulse width	460			ns
$t_C(\text{S}_{CLK})$	Serial I/O clock input cycle time	$8 \cdot t_C(X_{IN})$			ns
$t_{WH}(\text{S}_{CLK})$	Serial I/O clock input "H" pulse width	$4 \cdot t_C(X_{IN})$			ns
$t_{WL}(\text{S}_{CLK})$	Serial I/O clock input "L" pulse width	$4 \cdot t_C(X_{IN})$			ns
$t_{SU}(\text{S}_{IN}\text{-S}_{CLK})$	Serial I/O input setup time	200			ns
$t_H(\text{S}_{CLK}\text{-S}_{IN})$	Serial I/O input hold time	150			ns

Table 14: Switching characteristics

($V_{CC}=4.0$ to 5.5 V, $V_{SS}=AV_{SS}=0$ V, $T_a=-40$ to 85 °C unless otherwise noted)

Symbol	Parameter	Limits			Unit
		min.	typ.	max.	
$t_{WH}(S_{CLK})$	Serial I/O clock output "H" pulse width	$0.5 \cdot t_C(S_{CLK}) - 50$			ns
$t_{WL}(S_{CLK})$	Serial I/O clock output "L" pulse width	$0.5 \cdot t_C(S_{CLK}) - 50$			ns
$t_D(S_{CLK}-S_{OUT})$	Serial I/O output delay time			50	ns
$t_V(S_{CLK}-S_{OUT})$	Serial I/O output valid time	0		50	ns
$t_R(S_{CLK})$	Serial I/O clock output rise time			50	ns
$t_R(CMOS)$	CMOS output rise time		10	50	ns
$t_F(CMOS)$	CMOS output fall time		10	50	ns

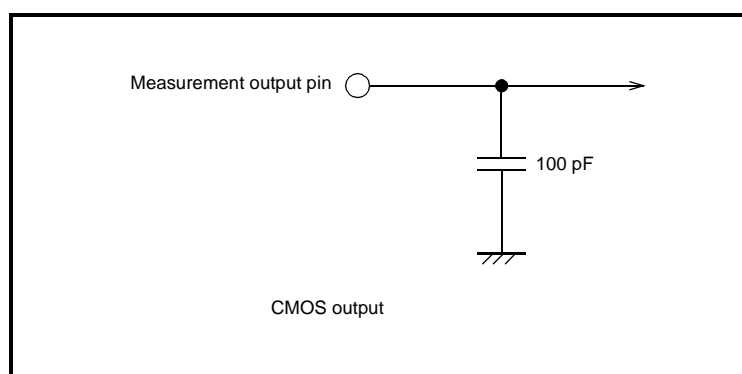


Fig. 51 Circuit for measuring output switching characteristics

HARDWARE

TIMING DIAGRAM

TIMING DIAGRAM

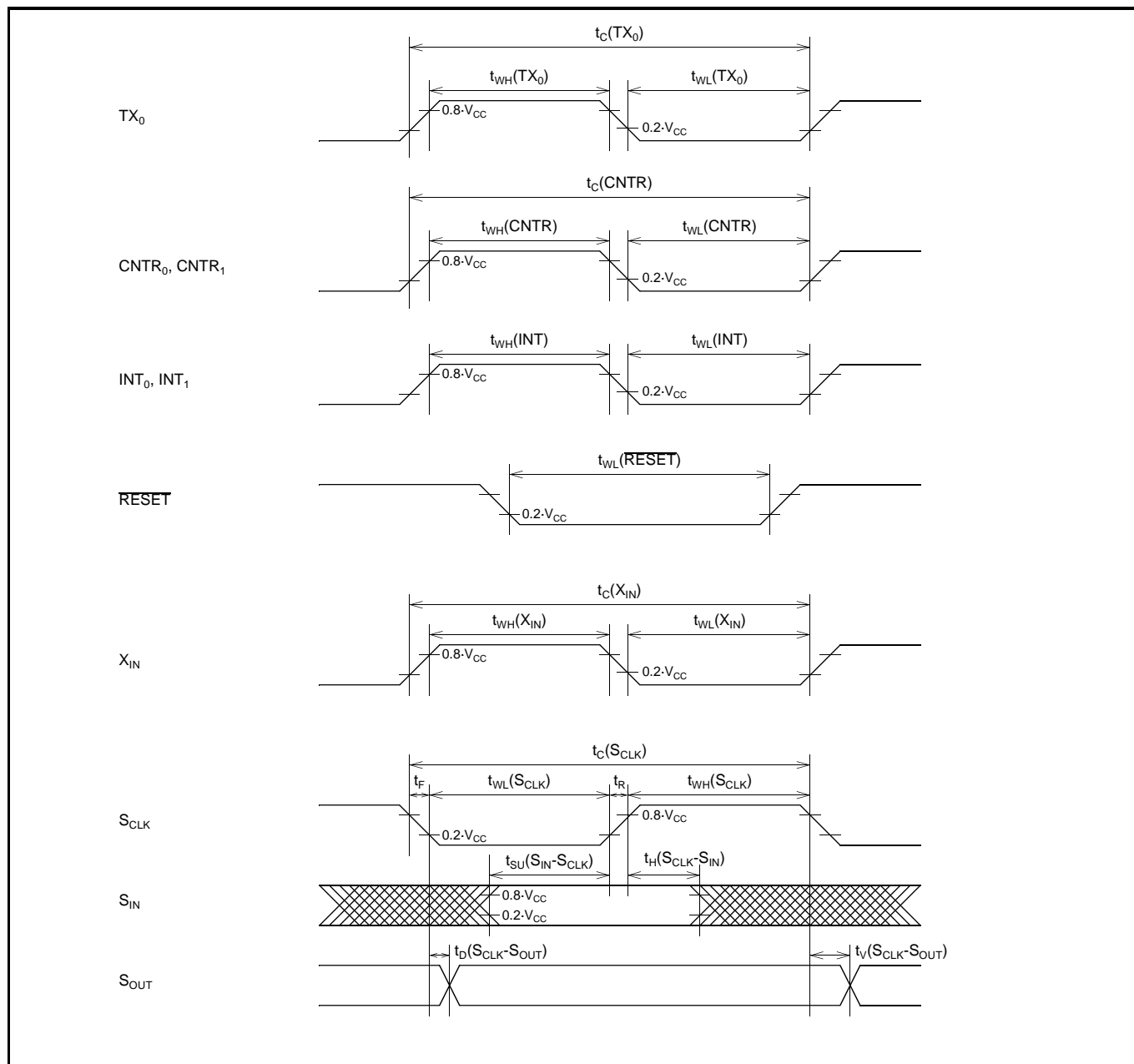


Fig. 52 Timing diagram



CHAPTER 2

APPLICATION

- 2.1 I/O Ports
- 2.2 Interrupts
- 2.3 Timers
- 2.4 Controller Area Network (CAN) module
- 2.5 Serial I/O
- 2.6 A-D converter
- 2.7 Watchdog timer
- 2.8 Reset
- 2.9 Oscillation circuit
- 2.10 Development support tools
- 2.11 Built-in PROM version

APPLICATION

2.1 I/O ports

2.1 I/O ports

2.1.1 Memory map of I/O ports

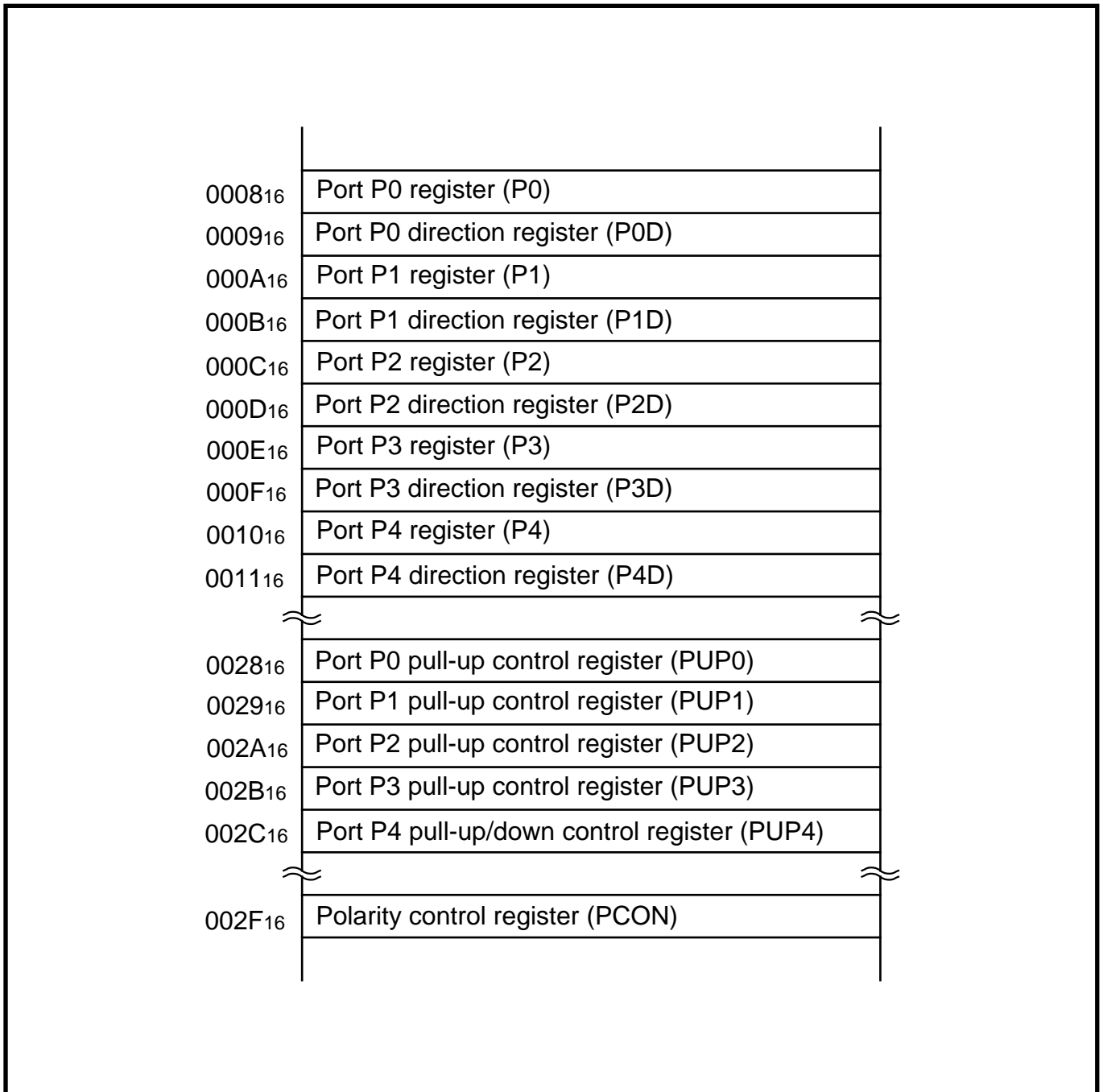


Fig. 2.1.1 Memory map of I/O port related registers

2.1.2 Related registers

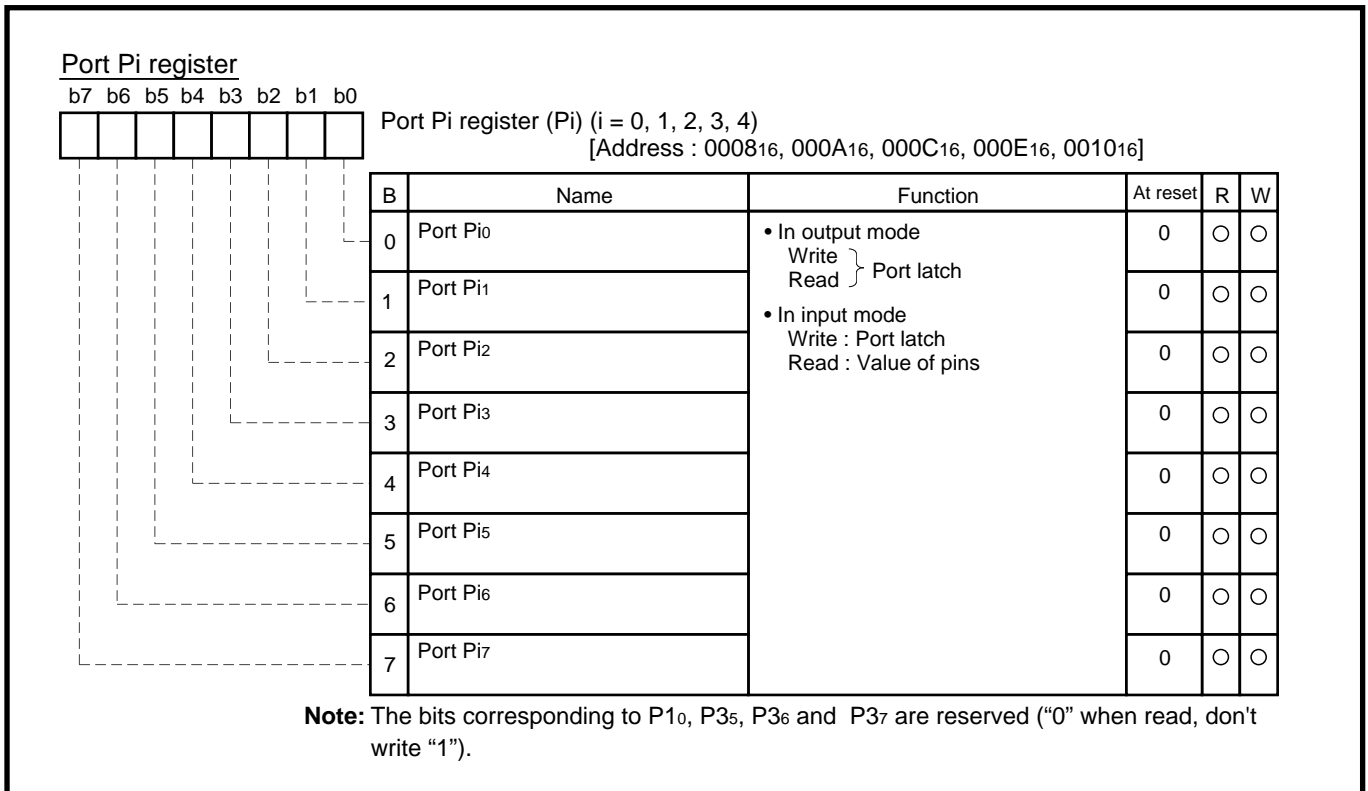


Fig. 2.1.2 Structure of Port Pi register (i = 0, 1, 2, 3, 4)

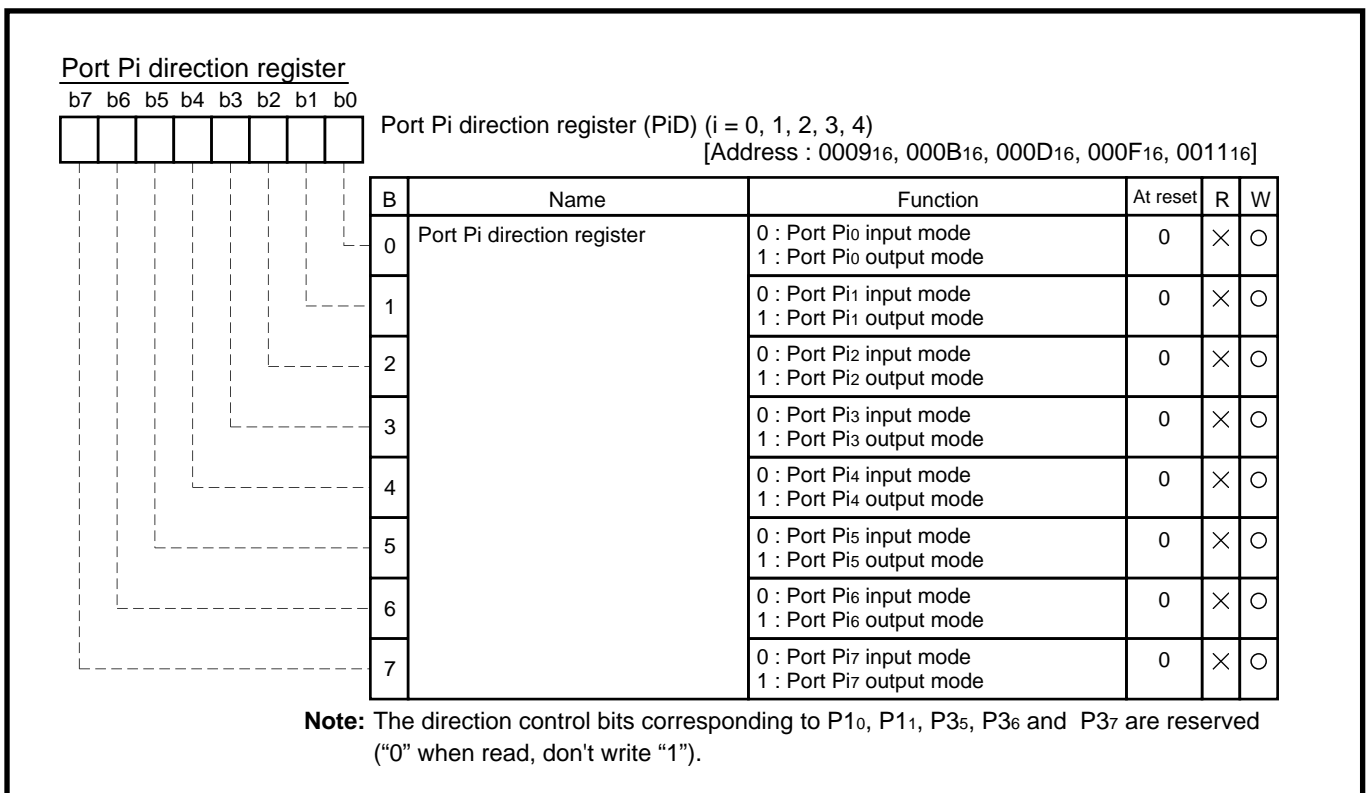


Fig. 2.1.3 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)

APPLICATION

2.1 I/O ports

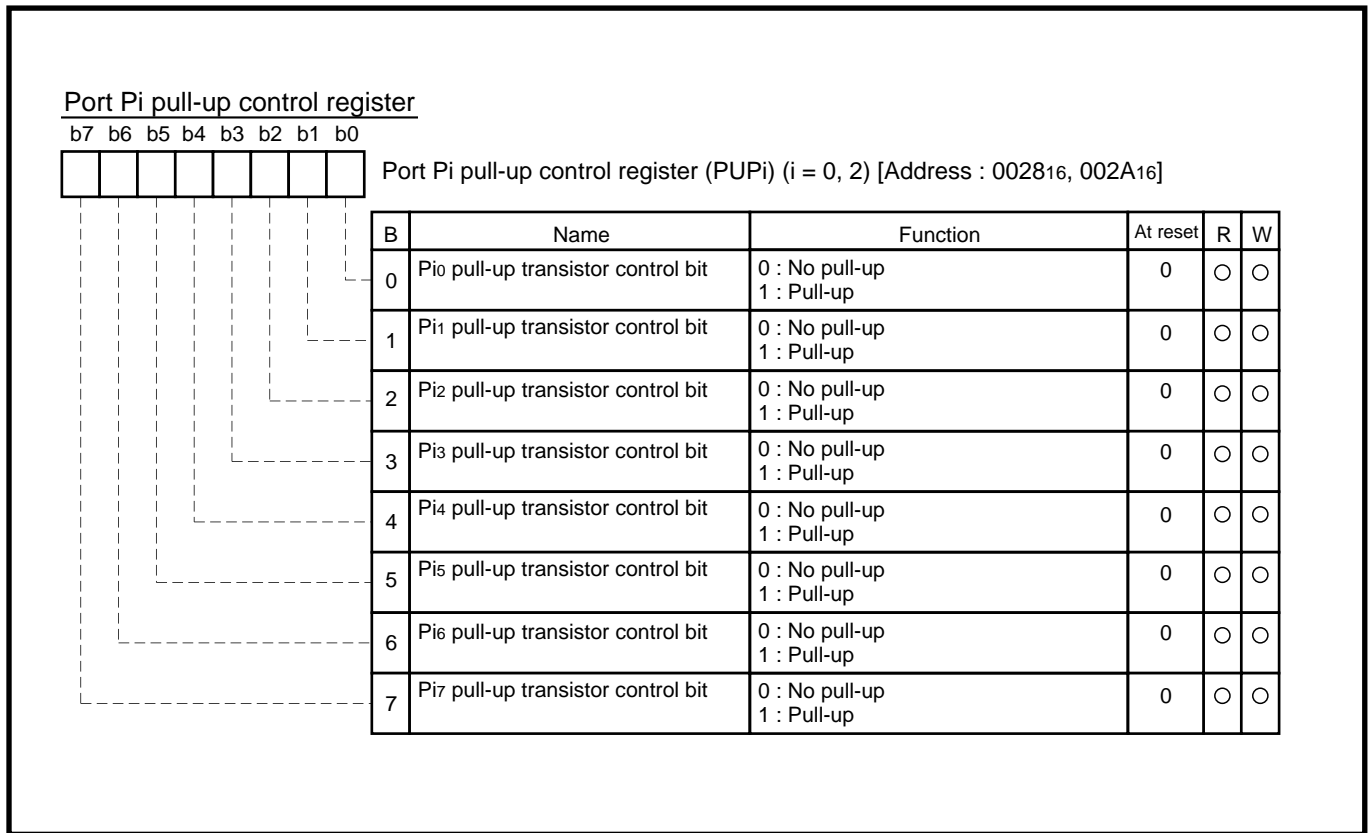


Fig. 2.1.4 Structure of Port Pi pull-up register (i = 0, 2)

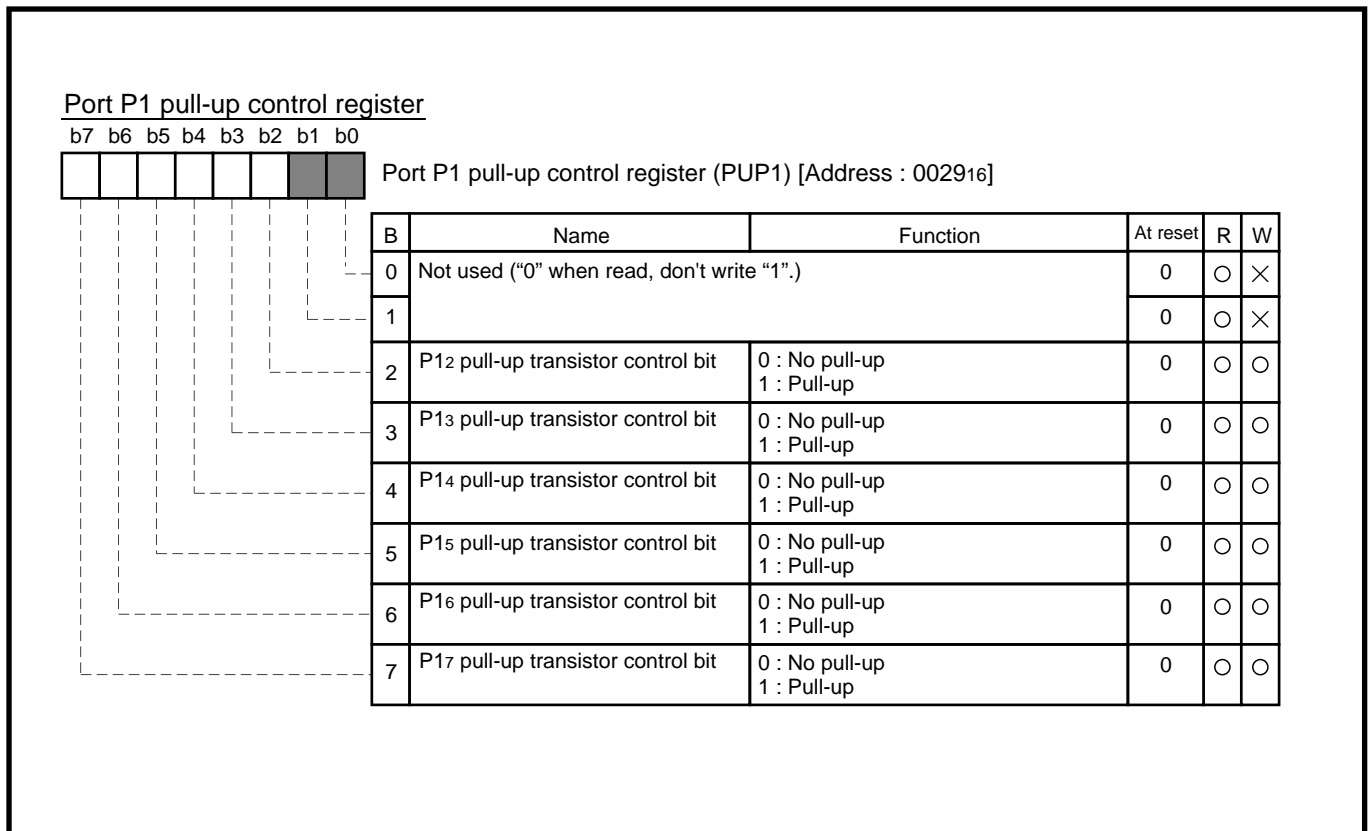


Fig. 2.1.5 Structure of Port P1 pull-up register

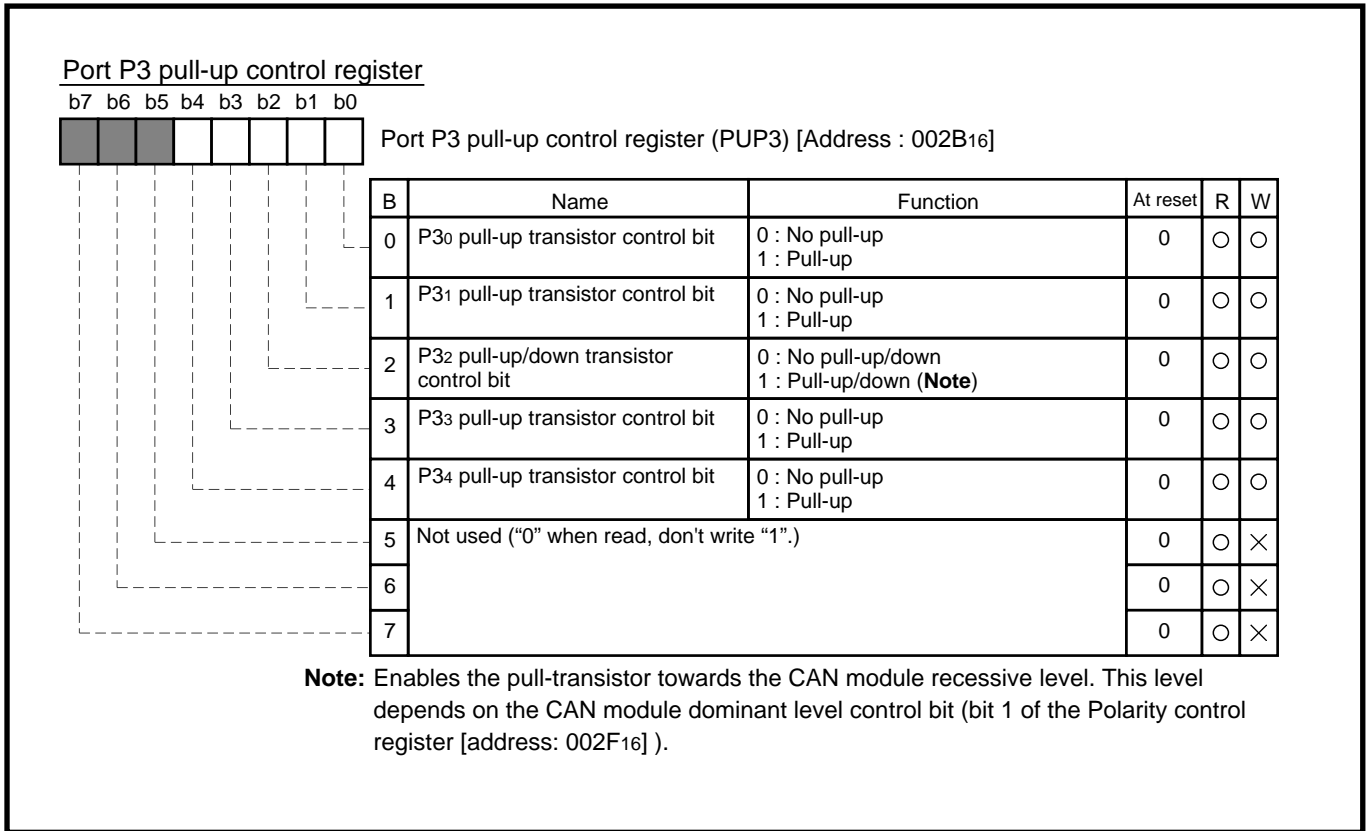


Fig. 2.1.6 Structure of Port P3 pull-up control register

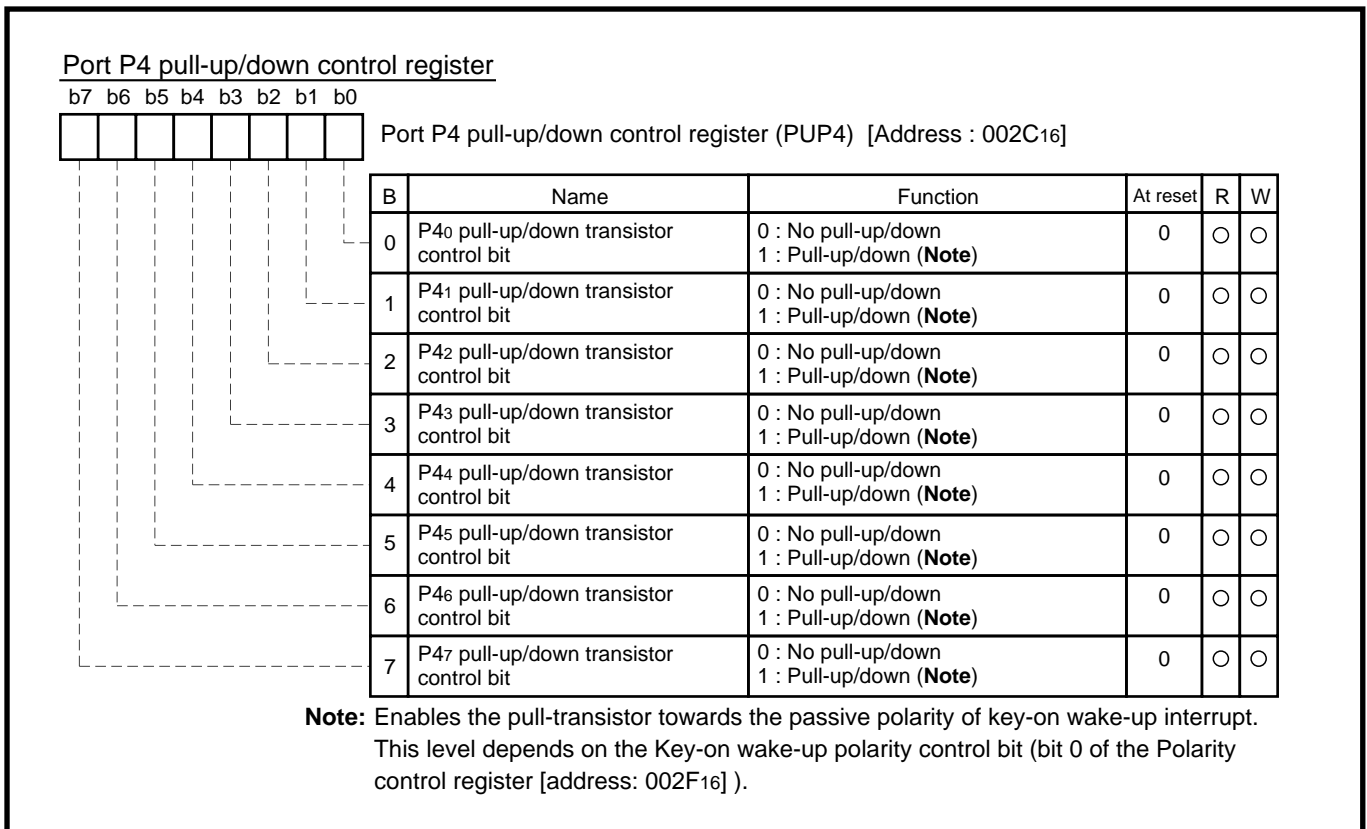


Fig. 2.1.7 Structure of Port P4 pull-up/down control register

APPLICATION

2.1 I/O ports

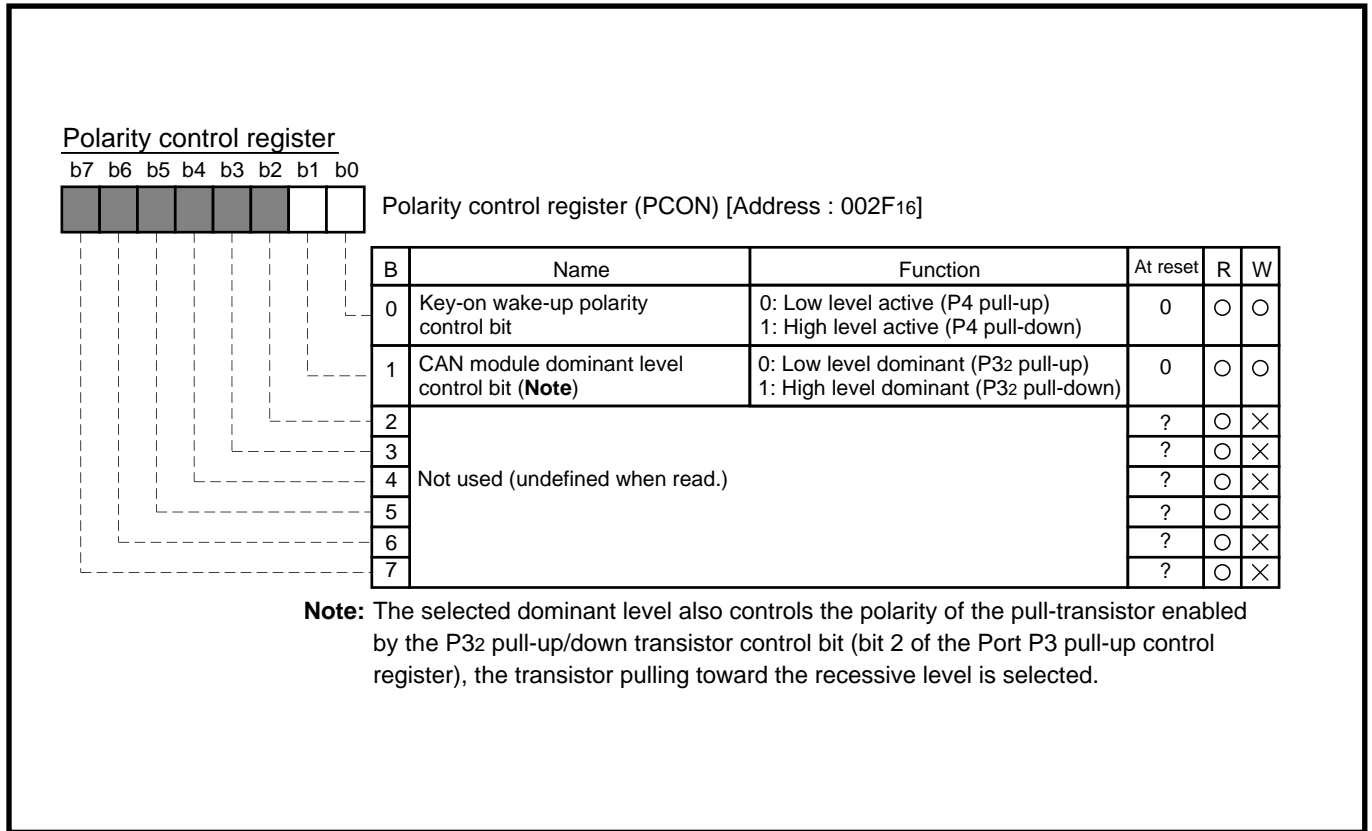


Fig. 2.1.8 Structure of Polarity control register

2.1.3 Overvoltage conditions at digital input ports

This section describes how to use digital input ports of the 7630 group at overvoltages. The terms over voltage refer to voltage levels beyond $V_{CC}+0.3V$. When subjected to such input voltage levels, the built-in protection circuit of the input port attempts to limit the input voltage in order to avoid permanent damage to the device. This condition causes input current to the port. The built-in protection circuit tolerates input currents up to specified limits (refer to “**3.1 Electrical characteristics**”). The input current levels must be limited by appropriate design of the application circuit connected to the corresponding port. Figure 2.1.9 shows an example circuit.

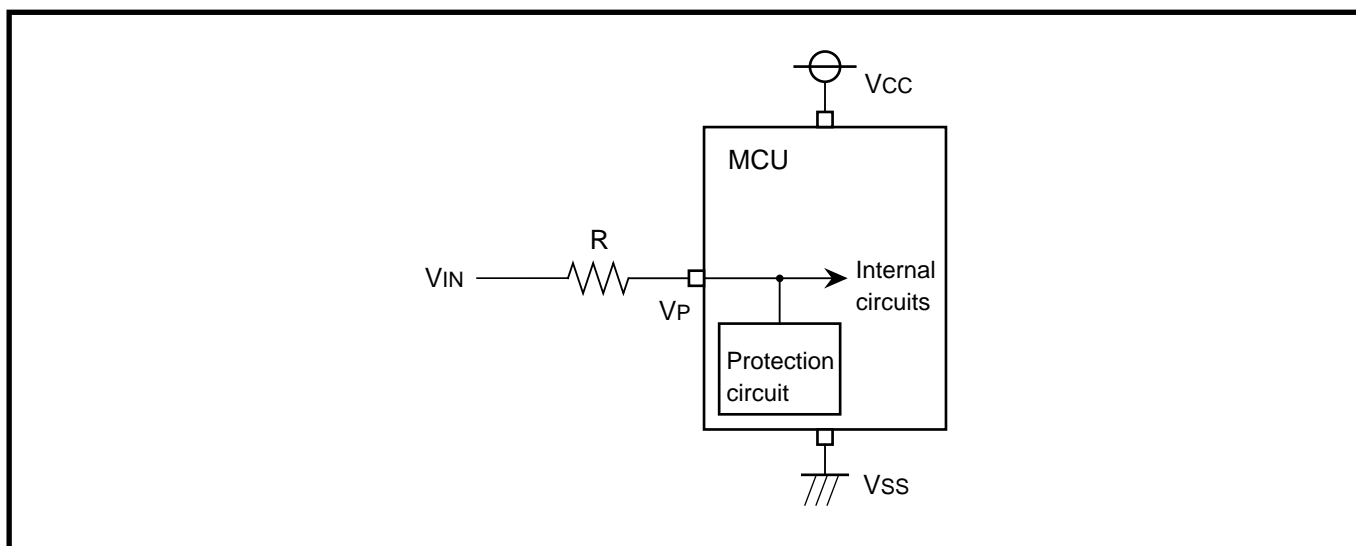


Fig. 2.1.9 External circuit example applying overvoltage to digital inputs

Assume V_{IN} the voltage to be connected to the MCU. The resistor R limits the input current to satisfy the “recommended operating conditions”. For an estimation of the resistor, the port voltage V_P should be assumed to be V_{CC} at overvoltage ($V_{IN} > V_{CC}+0.3V$). To determine the appropriate resistor size refer to the below:

$$R \geq \frac{V_{IN(max.)} - V_{CC}}{I_{IO}} \quad (V_{IN} > V_{CC}+0.3V)$$

- Notes:**
- Subjecting ports to overvoltage may effect the supply voltage and ground levels of the application and the device. Ensure appropriate design (low impedance) of the power and ground supply to keep V_{CC} and V_{SS} within the specified limits. In particular, avoid subjecting ports to overvoltage causing $V_{CC}-V_{SS}$ to rise above 5.5 V.
 - Port P0 must not be subjected to overvoltage conditions.
 - Overvoltages causing input current flowing through the internal port protection circuits have a negative effect on the ports noise immunity. Therefore, careful and intense testing of the target system’s noise immunity is required.
 - Because of the above noise immunity issue, it is not recommended to subjects ports with interrupt functions (such as ports for external interrupt) to overvoltage conditions.
 - Refer to the “**3.4 Countmeasures against noise**”.

APPLICATION

2.1 I/O ports

2.1.4 Handling examples of unused pins

Table 2.1.1 Handling of unused pins

Name of Pins/Ports	Handling
P0, P11(Note) P12 to P17, P2, P3, P4	Configure as inputs and pull to VCC or Vss via a resistor of 1 k Ω to 10 k Ω , or configure as outputs and leave open (except P11).
VREF	Connect to Vss(GND) or leave open.
AVSS	Connect to Vss(GND).
XOUT	Leave open (only when using external clock).

Note: The P11 pin of the built-in programmable ROM version is used in common with the VPP pin, insert a resistor of about 5 k Ω in series and connect by the shortest wiring.

2.2 Interrupts

2.2.1 Memory map of interrupt related registers

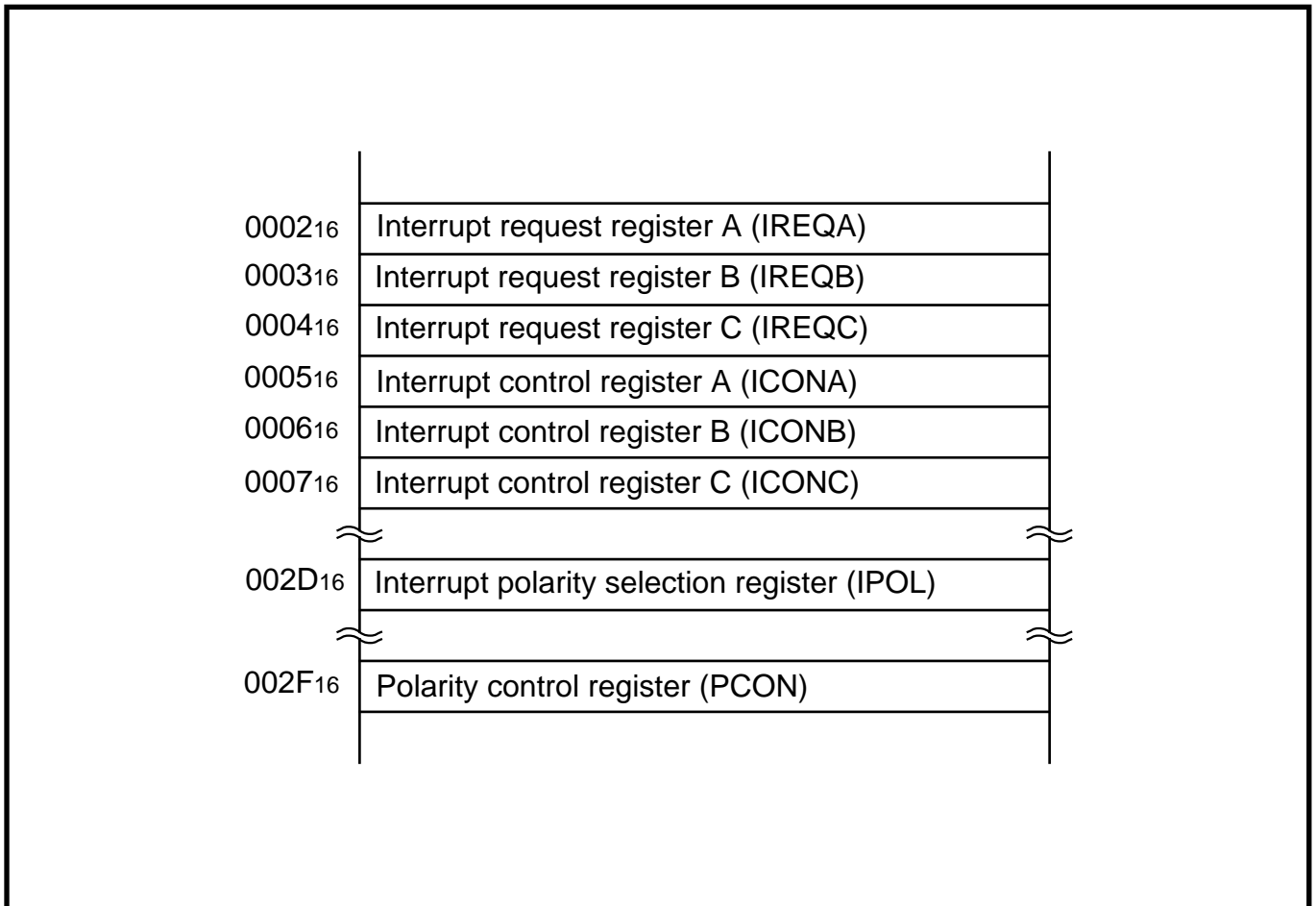


Fig. 2.2.1 Memory map of interrupt related registers

APPLICATION

2.2 Interrupts

2.2.2 Related registers

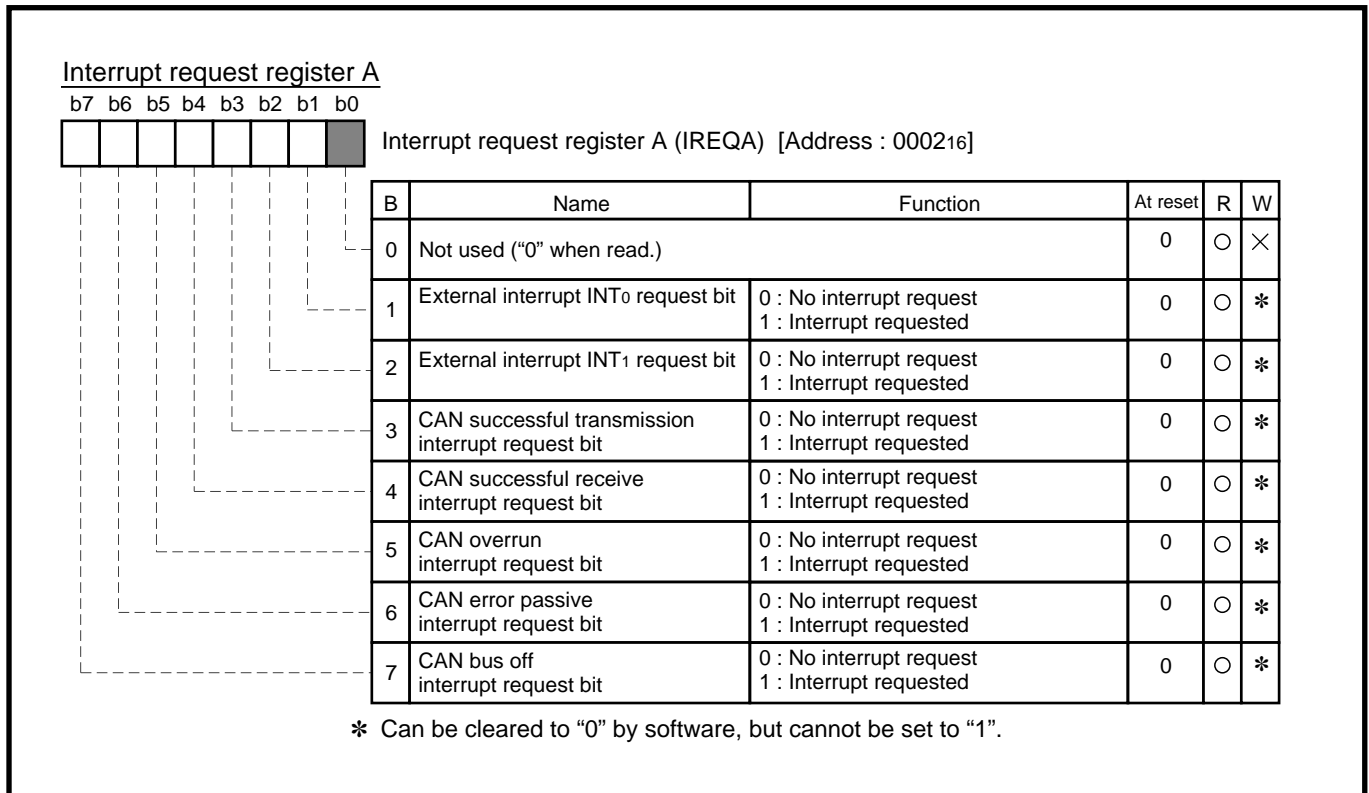


Fig. 2.2.2 Structure of Interrupt request register A

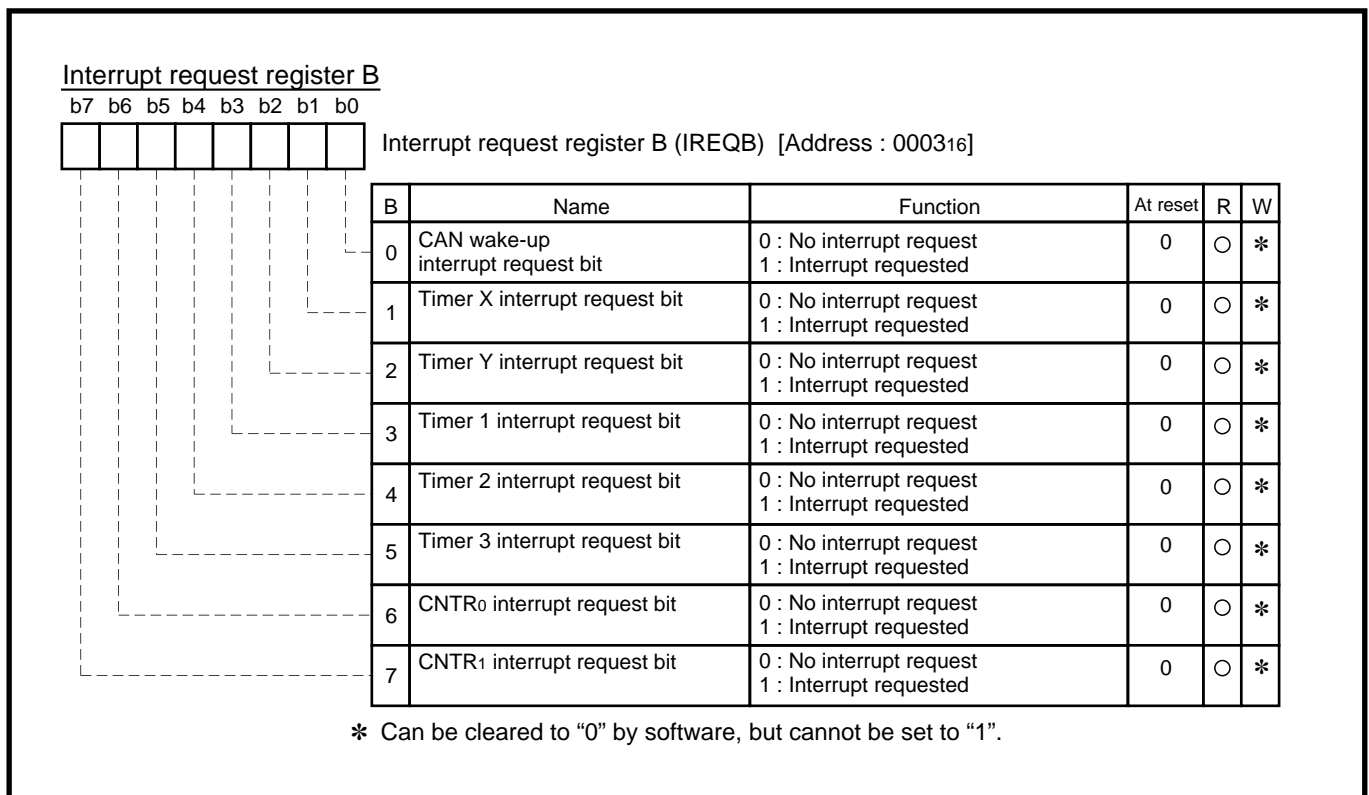


Fig. 2.2.3 Structure of Interrupt request register B

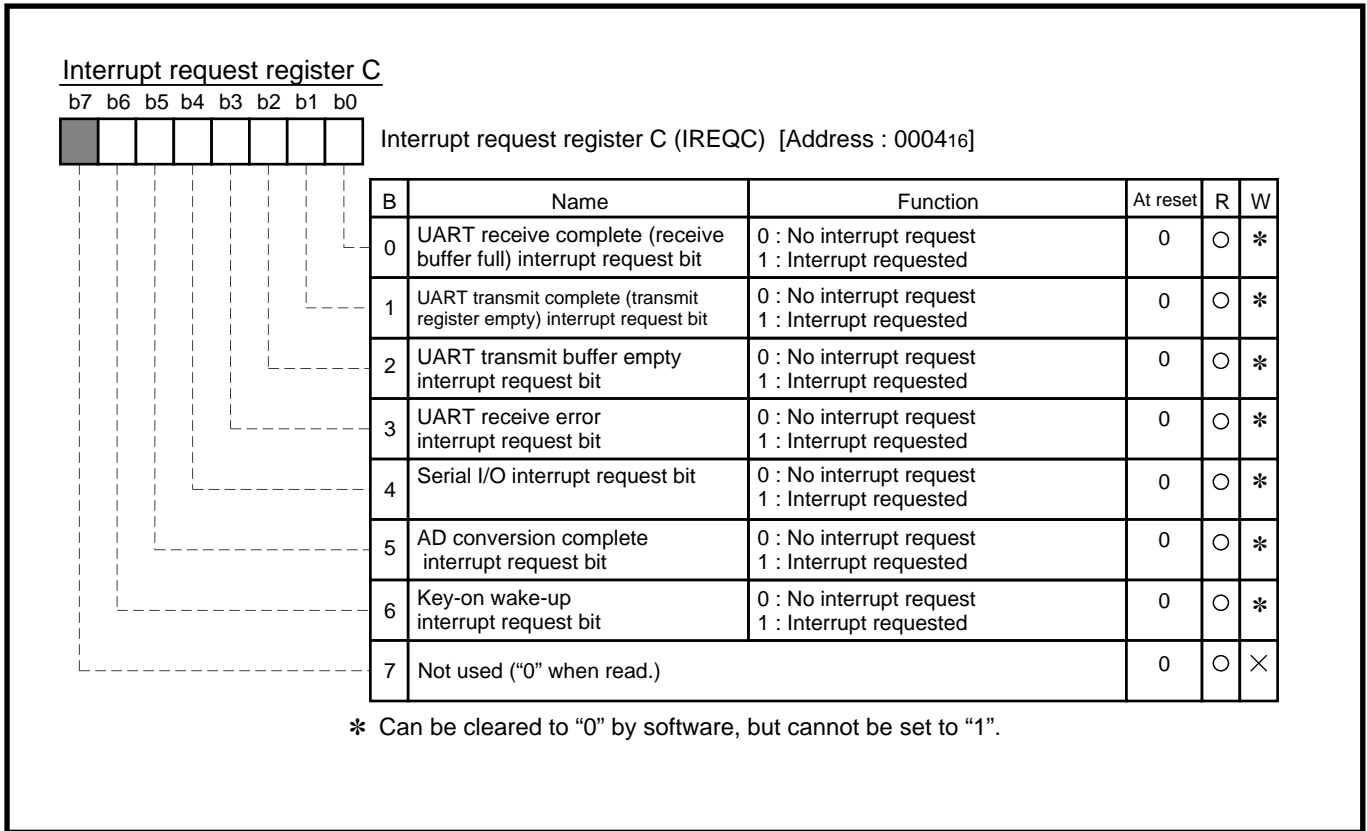


Fig. 2.2.4 Structure of Interrupt request register C

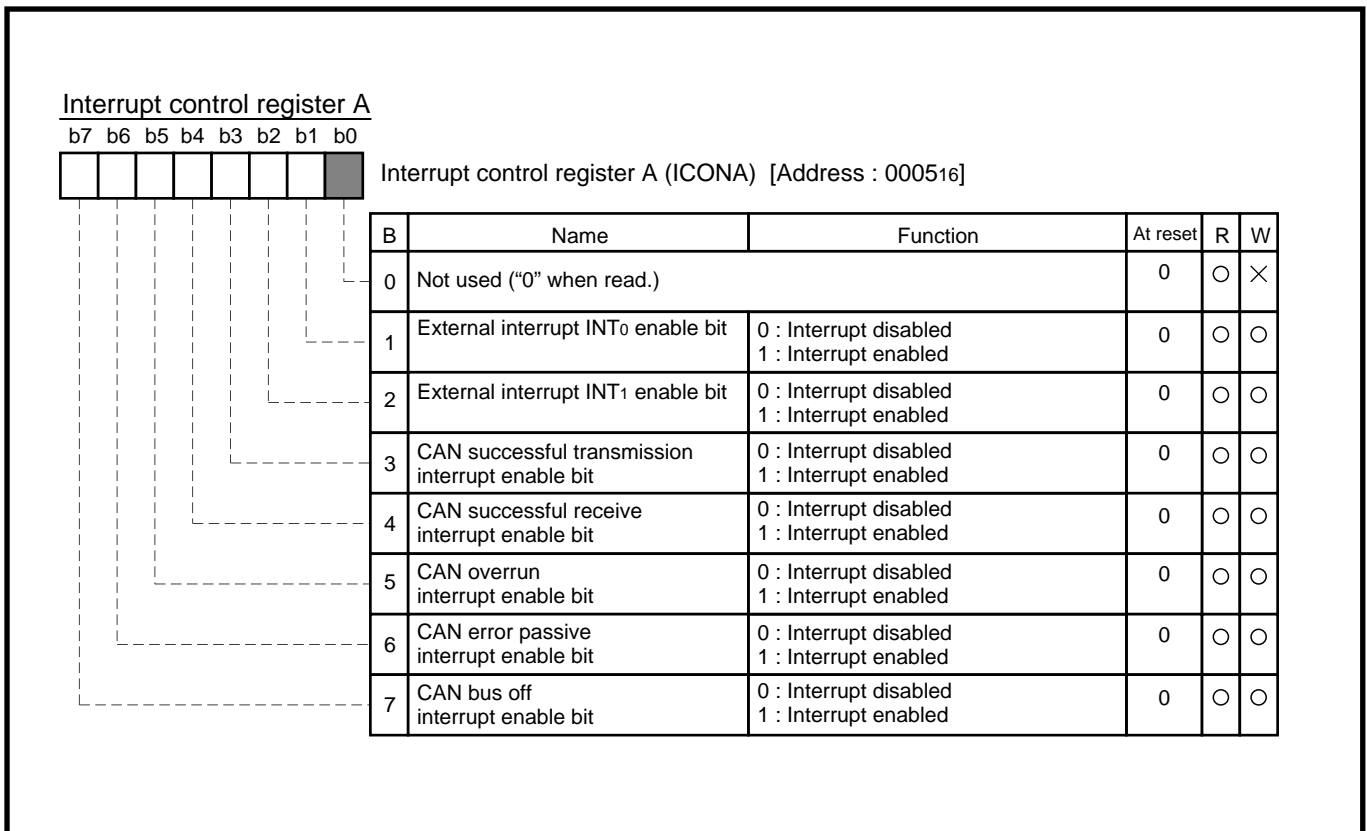


Fig. 2.2.5 Structure of Interrupt control register A

APPLICATION

2.2 Interrupts

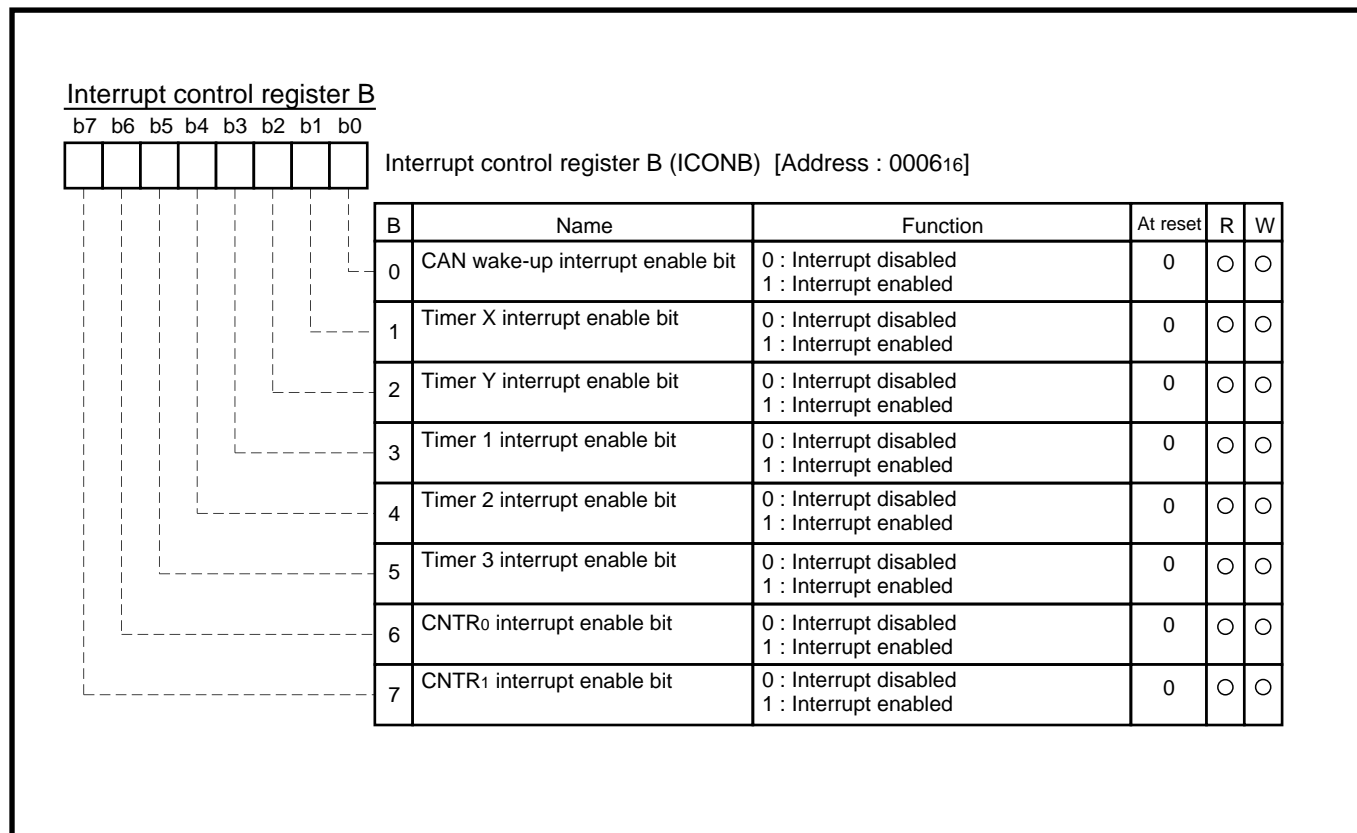


Fig. 2.2.6 Structure of Interrupt control register B

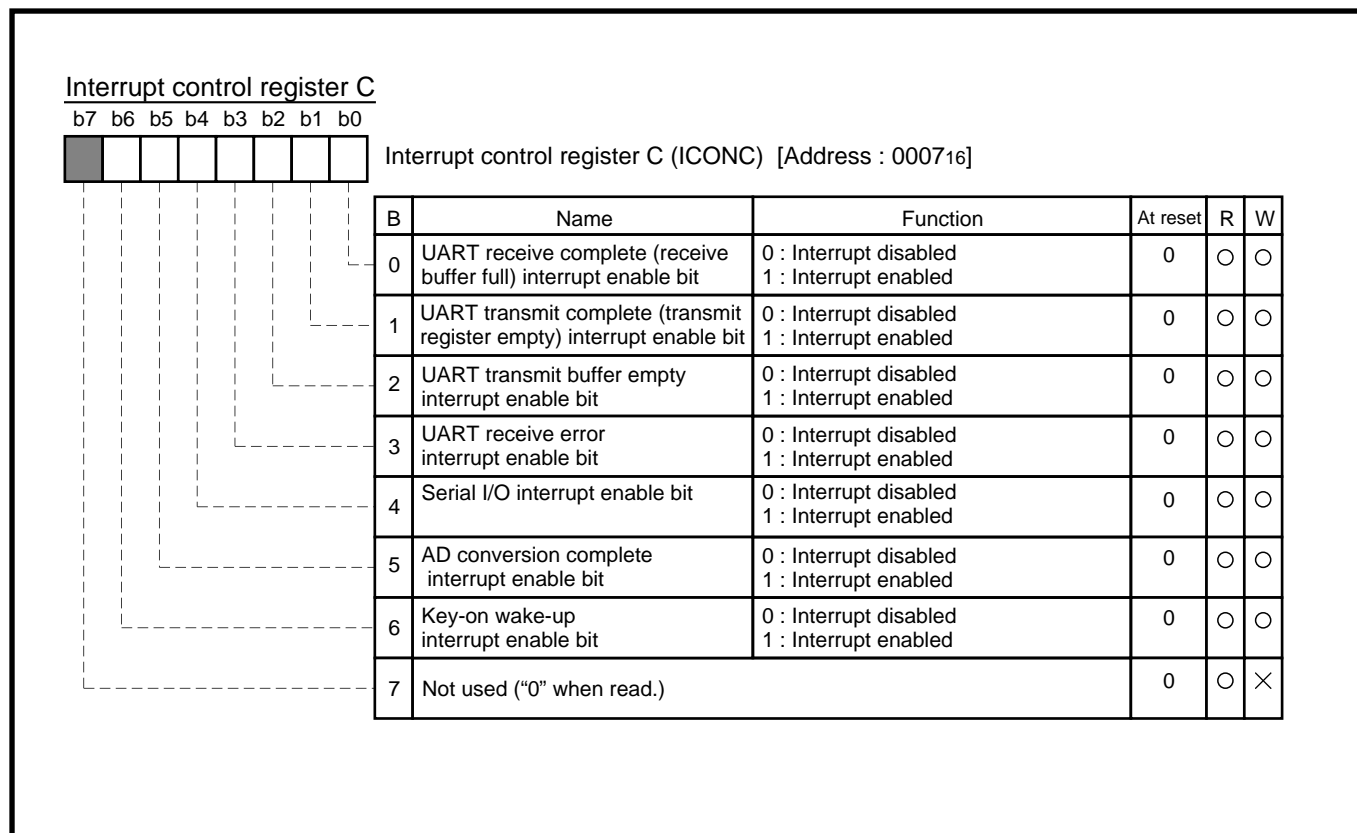


Fig. 2.2.7 Structure of Interrupt control register C

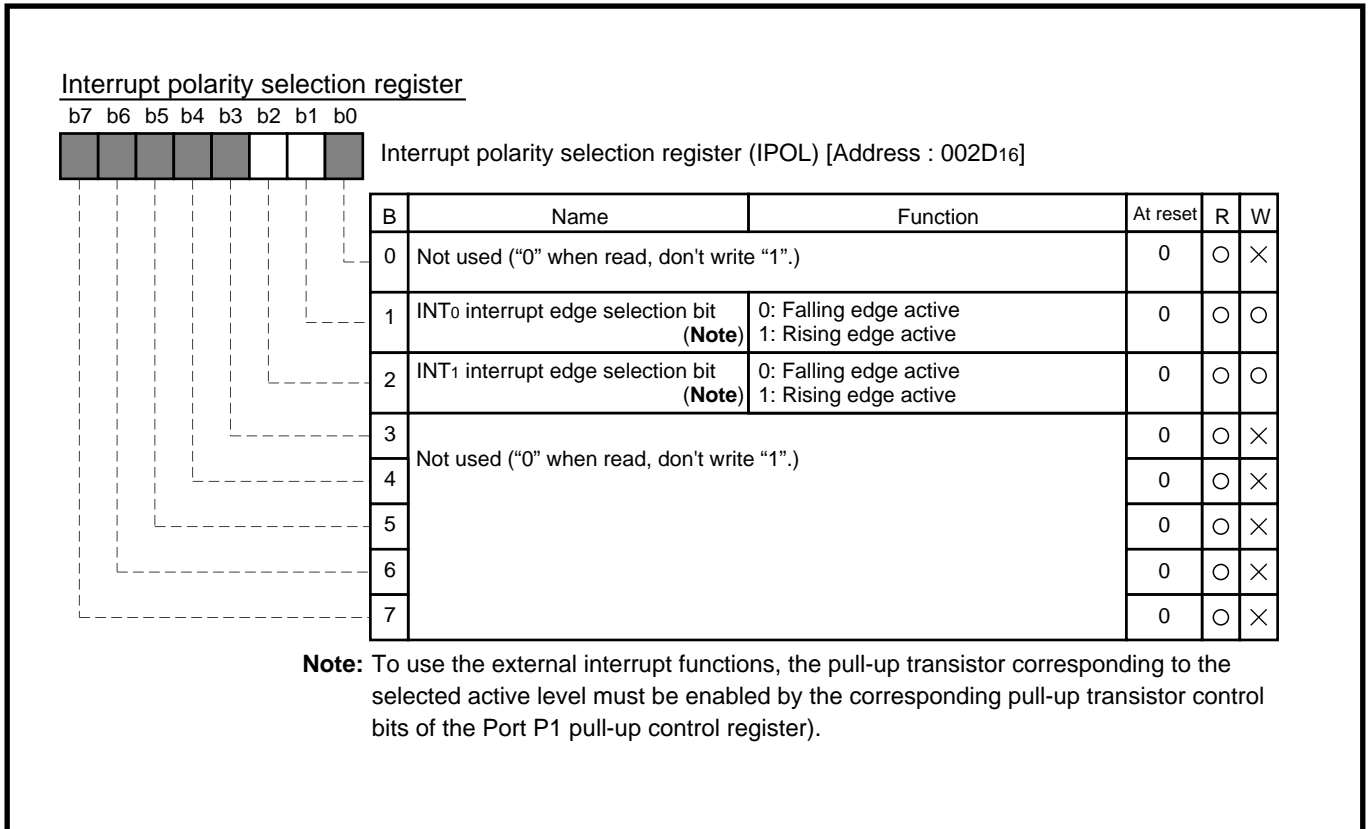


Fig. 2.2.8 Structure of Interrupt polarity selection register

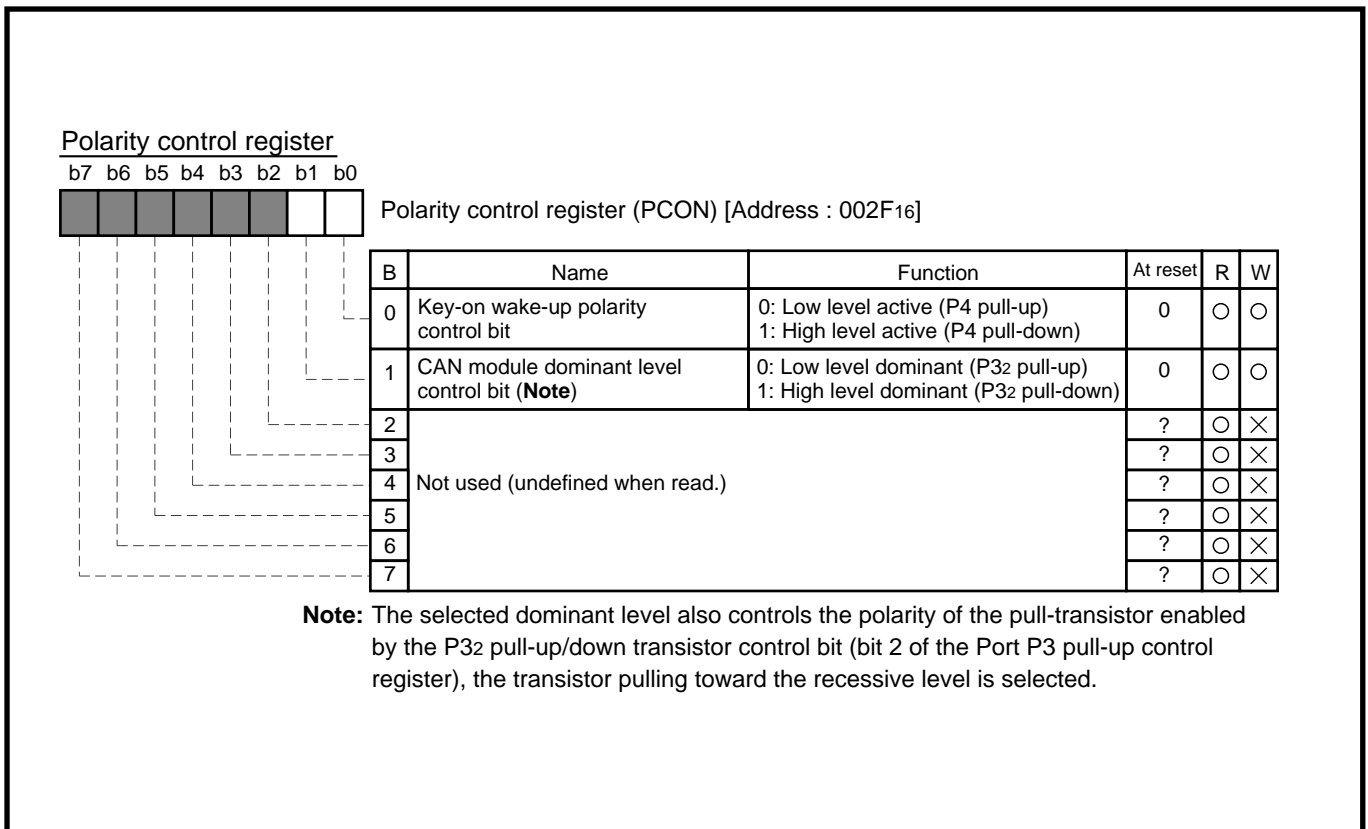


Fig. 2.2.9 Structure of Polarity control register

APPLICATION

2.2 Interrupts

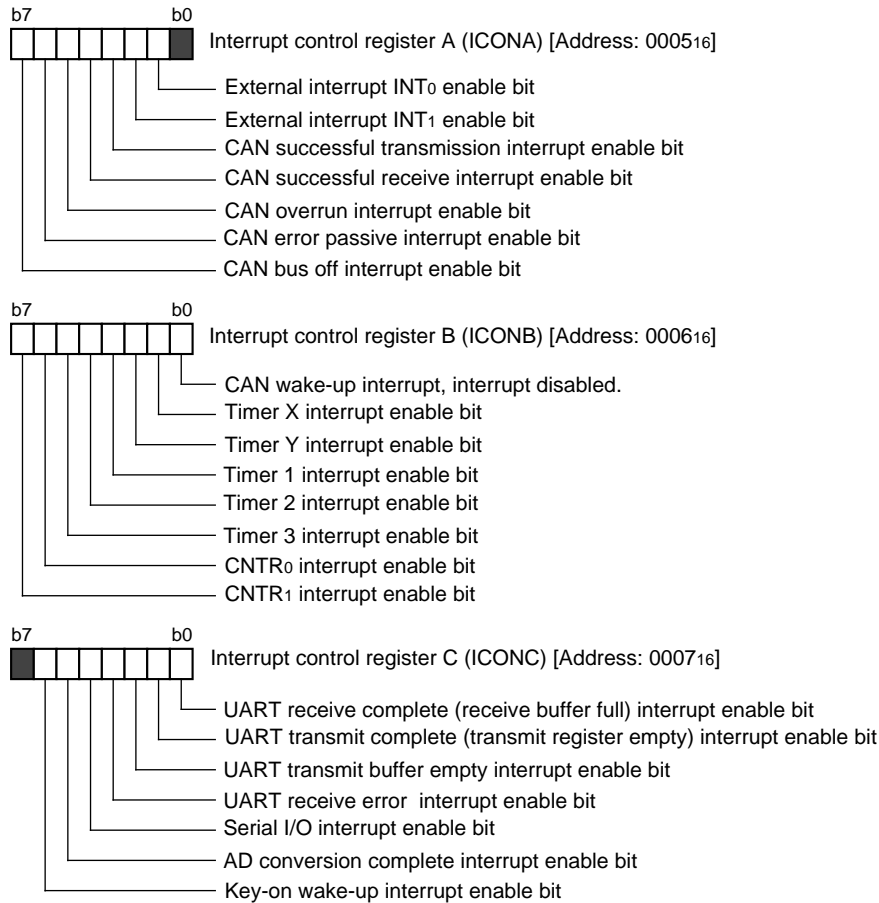
2.2.3 Interrupt setting method

Figure 2.2.10 and Figure 2.2.11 show interrupt setting method.

Step 1: Disable all interrupts or the setting interrupts to prevent unnecessary interrupts occurring during setting.

In the former, set the Interrupt disable flag (I) to "1".

In the latter, clear the corresponding interrupt enable bits to "0".

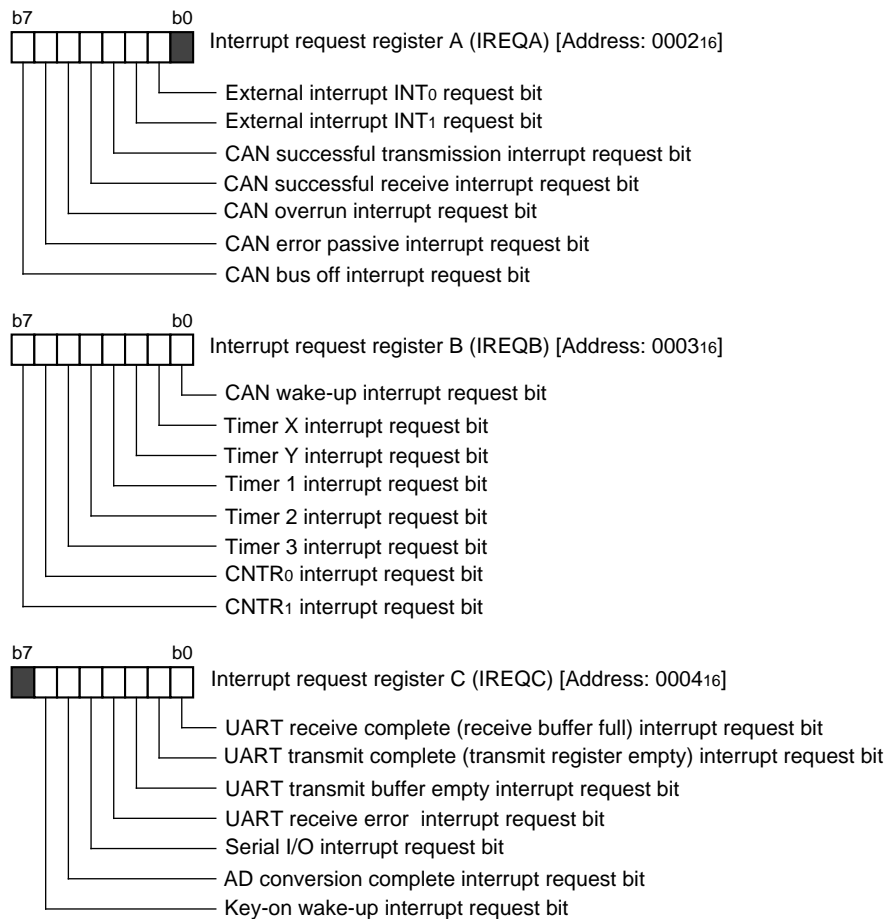


Step 2: Set the each function related the setting interrupts (**Note 1**).

Note 1: For details, refer to setting method of each function.

Fig. 2.2.10 Interrupt setting method (1)

Step 3: Clear the setting interrupt request bits to “0” (no interrupt request).



Step 4: Set the using interrupt enable bits to “1” (interrupt enabled). Refer to **Step 1**.

Step 5: When the Interrupt disable flag (I) is set to “1” in **Step 1**, clear the flag to “0” (interrupt enabled).

Step 6: Operate the each functions related the using interrupts (**Note 2**).

Note 2: For details, refer to setting method of each function.

Fig. 2.2.11 Interrupt setting method (2)

APPLICATION

2.2 Interrupts

2.2.4 Key-on wake-up interrupt

Figure 2.2.12 and Figure 2.2.13 show setting method for registers related to the key-on wake-up interrupt.

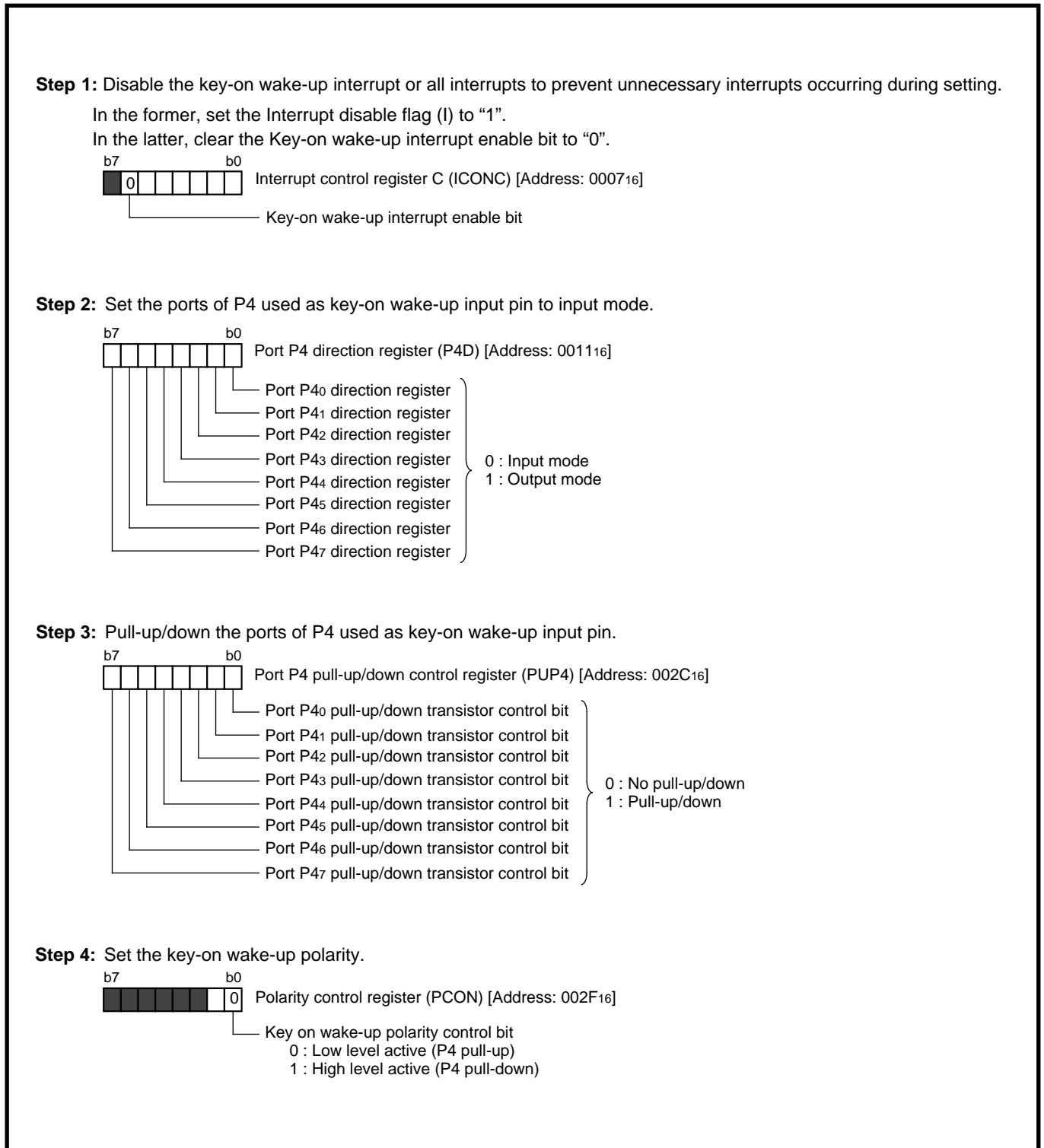
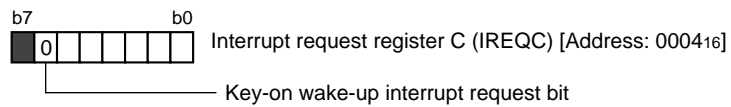
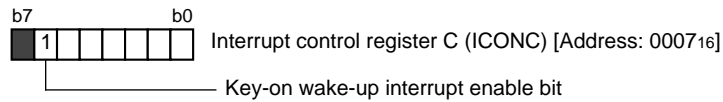


Fig. 2.2.12 Setting method for registers related to key-on wake-up interrupt (1)

Step 5: Clear the Key-on wake-up interrupt request bit to “0” (no interrupt request).



Step 6: Set the Key-on wake-up interrupt enable bit to “1” (interrupt enabled).



Step 7: When the Interrupt disable flag (I) is set to “1” in **Step 1**, clear the flag to “0” (interrupt enabled).

Step 8: Execute the **STP/WIT** instruction to switch procedure to the stop/wait mode.

Fig. 2.2.13 Setting method for registers related to key-on wake-up interrupt (2)

APPLICATION

2.3 Timers

2.3 Timers

2.3.1 Memory map of timer

0016 ₁₆	Timer 1 (T1)
0017 ₁₆	Timer 2 (T2)
0018 ₁₆	Timer 3 (T3)
0019 ₁₆	Timer 123 mode register (T123M)
001A ₁₆	Timer XL (TXL)
001B ₁₆	Timer XH (TXH)
001C ₁₆	Timer YL (TYL)
001D ₁₆	Timer YH (TYH)
001E ₁₆	Timer X mode register (TXM)
001F ₁₆	Timer Y mode register (TYM)

Fig. 2.3.1 Memory map of timer related registers

2.3.2 Related registers

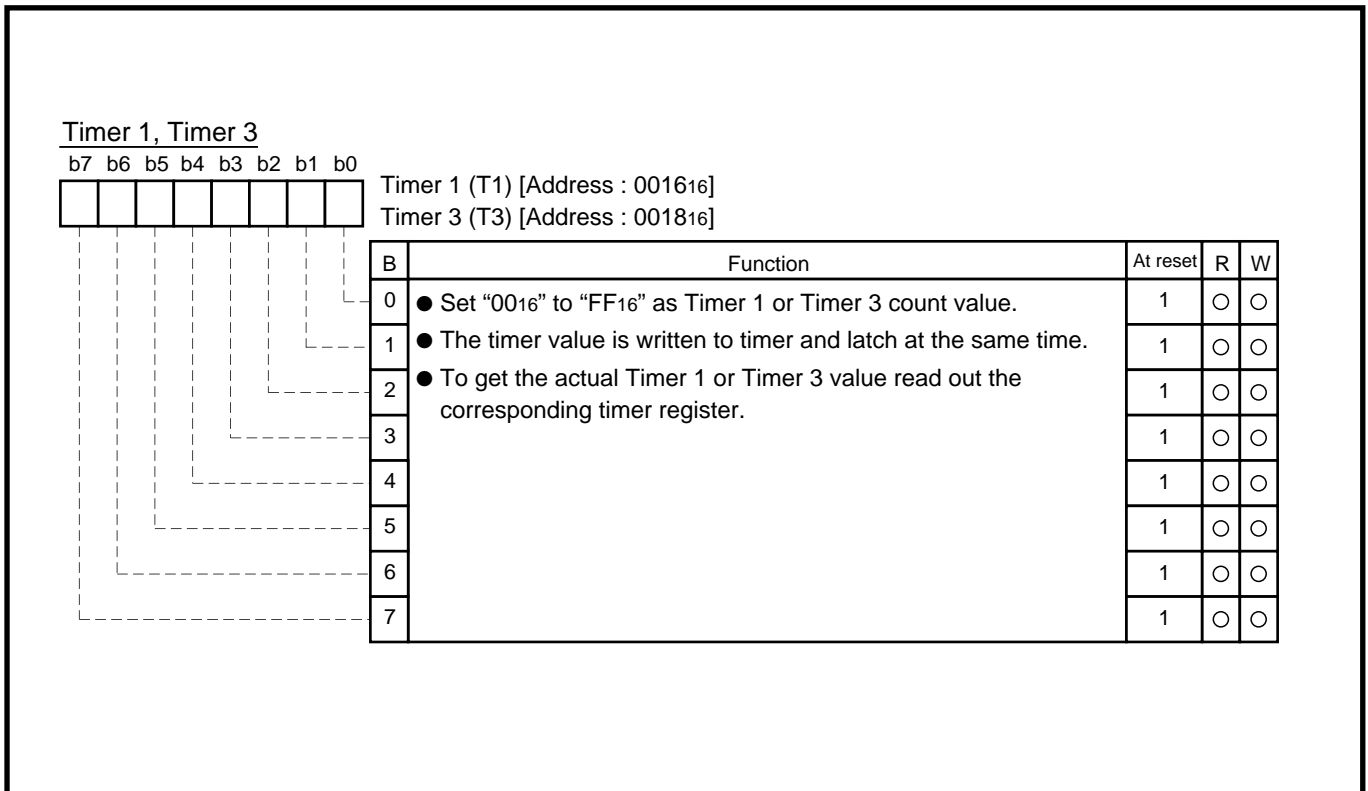


Fig. 2.3.2 Structure of Timer 1, Timer 3

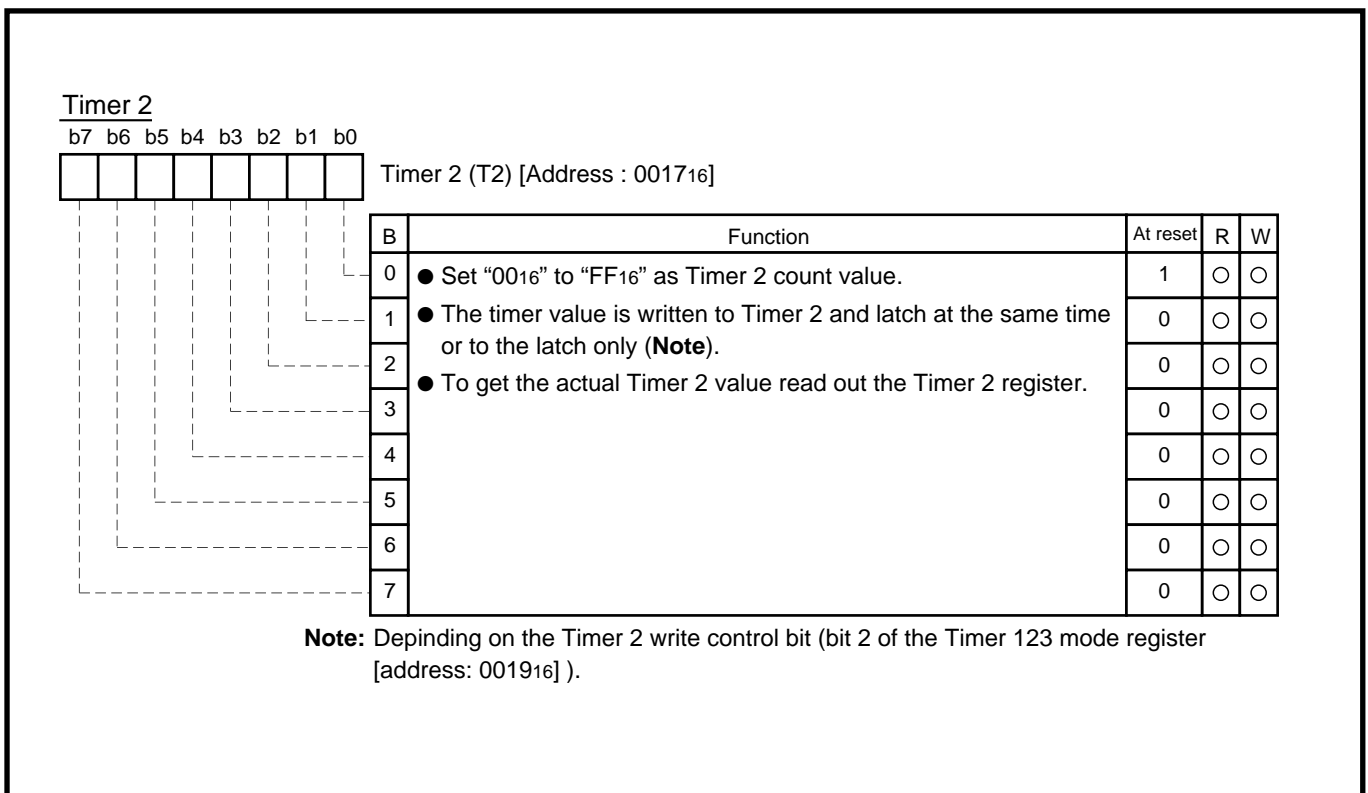


Fig. 2.3.3 Structure of Timer 2

APPLICATION

2.3 Timers

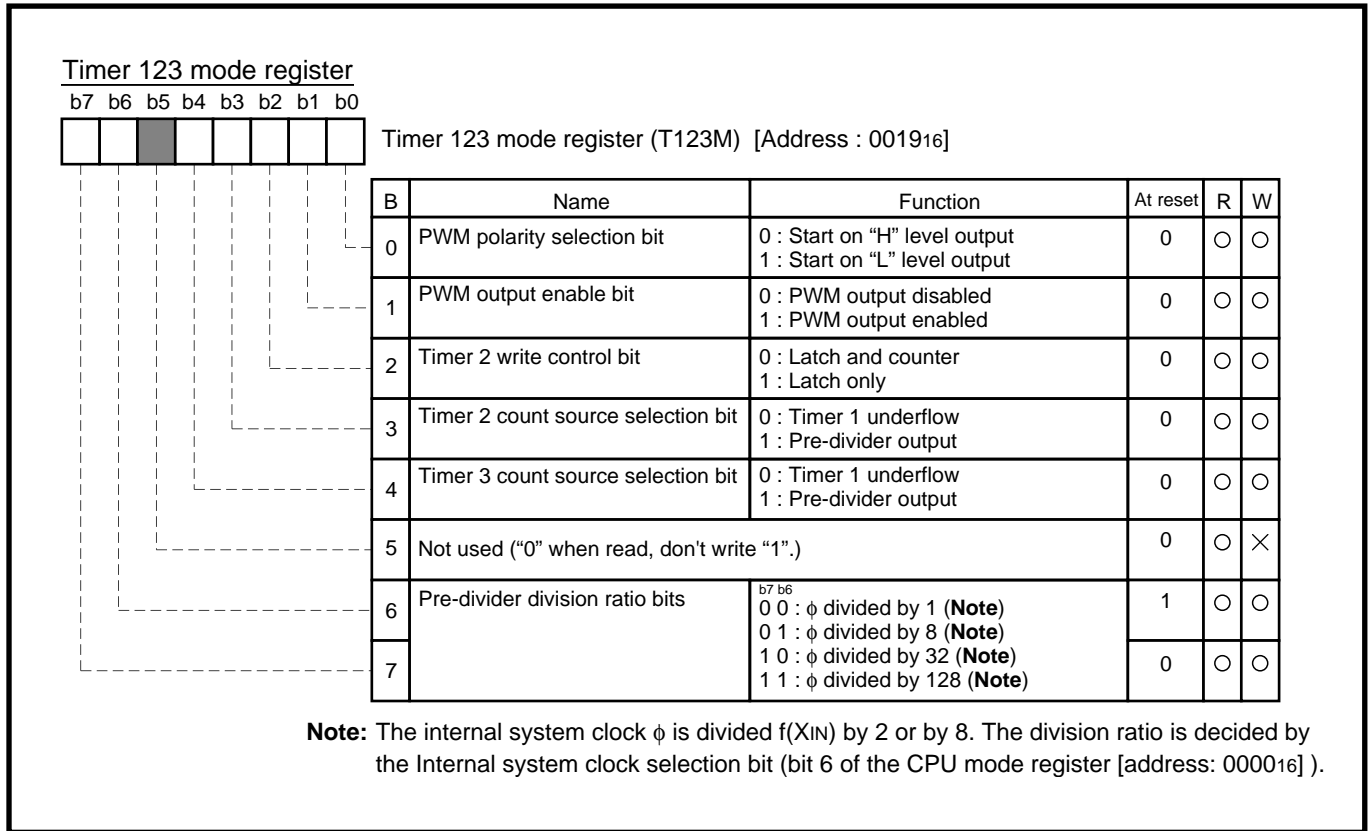


Fig. 2.3.4 Structure of Timer 123 mode register

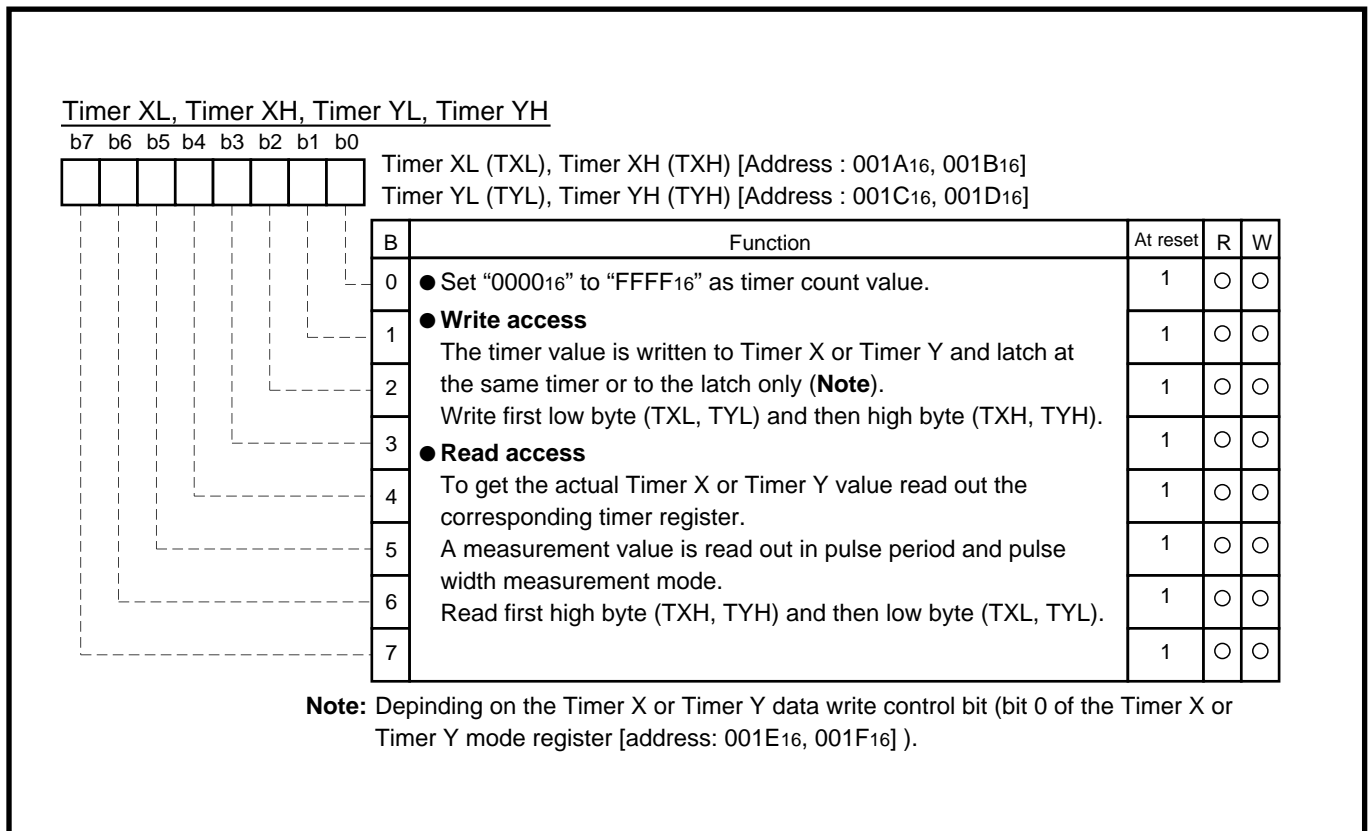


Fig. 2.3.5 Structure of Timer XL, Timer XH, Timer YL, Timer YH

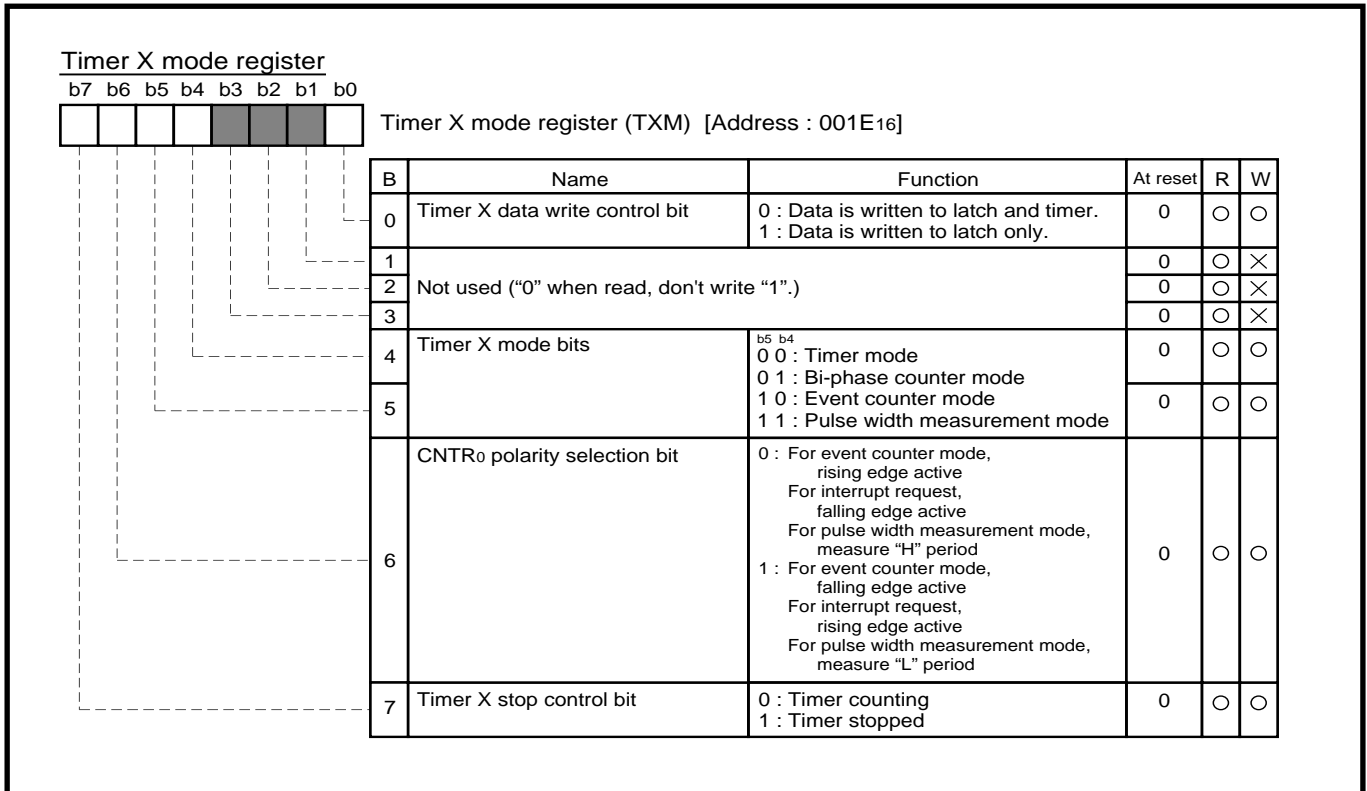


Fig. 2.3.6 Structure of Timer X mode register

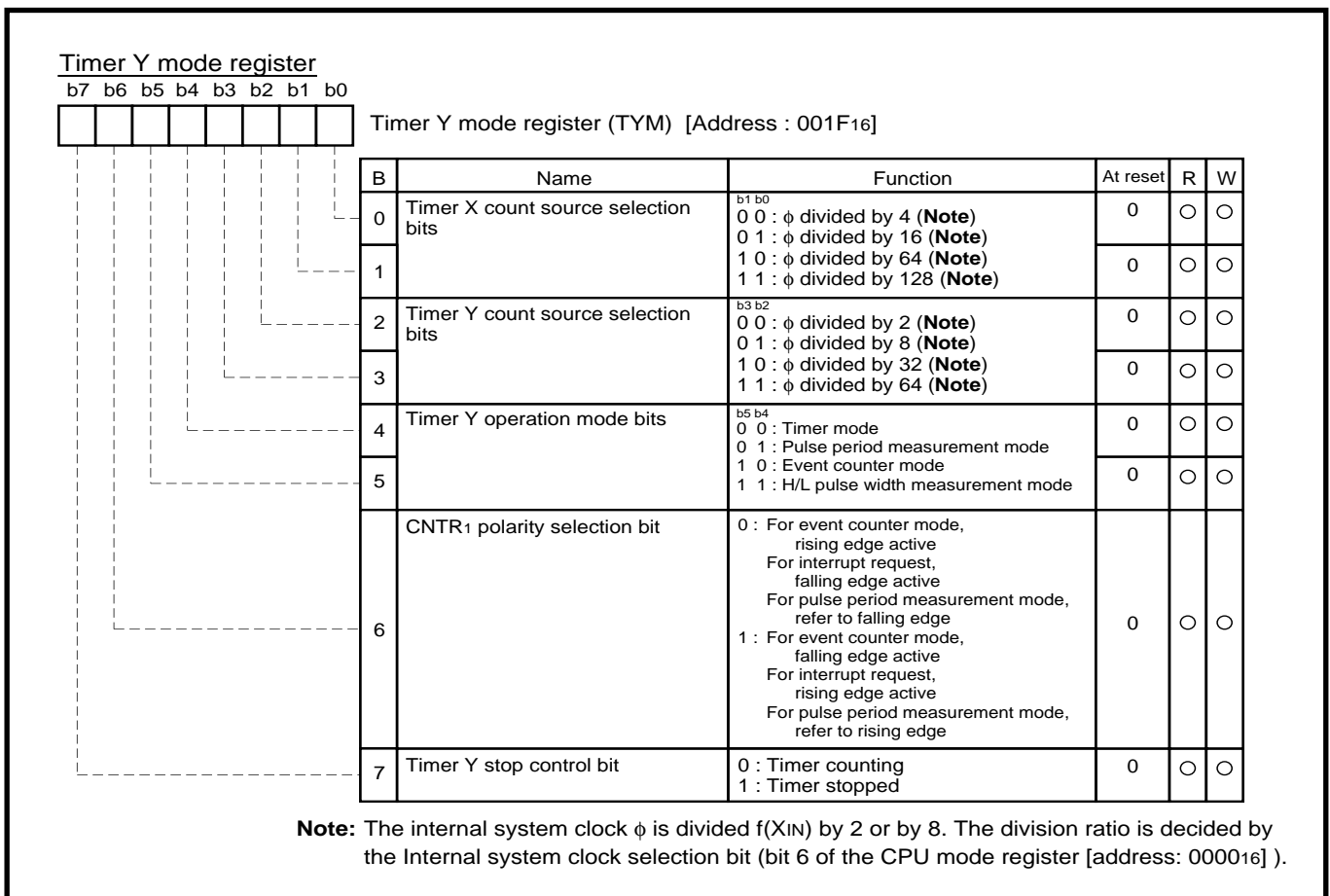


Fig. 2.3.7 Structure of Timer Y mode register

APPLICATION

2.3 Timers

2.3.3 Timer application examples

(1) Basic functions and uses

[Function 1] Control of Event interval (Timer X, Timer Y, Timer 1, Timer 2, Timer 3)

The Timer count stop bit is set to "0" after setting a count value to a timer. Then a timer interrupt request occurs after a certain period (**Timer mode**).

- [Use]
- Generation of an output signal timing
 - Generation of a waiting time

[Function 2] Control of Cyclic operation (Timer X, Timer Y, Timer 1, Timer 2, Timer 3)

The value of a timer latch is automatically written to a corresponding timer every time a timer underflows, and each cyclic timer interrupt request occurs (**Timer mode**).

- [Use]
- Generation of cyclic interrupts
 - Clock function (measurement of 25ms) → Application example 1
 - Control of a main routine cycle

[Function 3] Count of External pulse (Timer X)

External pulses input to the CNTR pin are selected as a timer count source (**Bi-phase mode**).

- [Use]
- Measurement of incremental sensor output signals

[Function 4] Count of External pulse (Timer X, Timer Y)

External pulses input to the CNTR pin are selected as a timer count source (**Event counter mode**).

- [Use]
- Measurement of frequency → Application example 2
 - Division of external pulses.
 - Generation of interrupts in a cycle based on an external pulse.
(count of a reel pulse)

[Function 5] Measurement of External pulse width (Timer X, Timer Y)

The "H" or "L" level width of external pulses input to CNTR pin is measured (**Pulse width measurement mode**).

- [Use]
- Measurement of external pulse frequency (Measurement of pulse width of FG pulse* generated by motor) → Application example 3
 - Measurement of external pulse duty (when the frequency is fixed)

* FG pulse : Pulse used for detecting the motor speed to control the motor speed.

(2) Timer application example 1 : Clock function (measurement of 25 ms)

Outline : The input clock is divided by a timer so that the clock counts up every 25 ms.

Specifications :

- The clock $f(X_{IN}) = 8 \text{ MHz}$ is divided by a timer.
- The clock is counted at intervals of 25 ms by the Timer 3 interrupt.

Figure 2.3.8 shows the timers connection and division ratios, Figures 2.3.9 show a setting of related registers, and Figure 2.3.10 shows a control procedure.

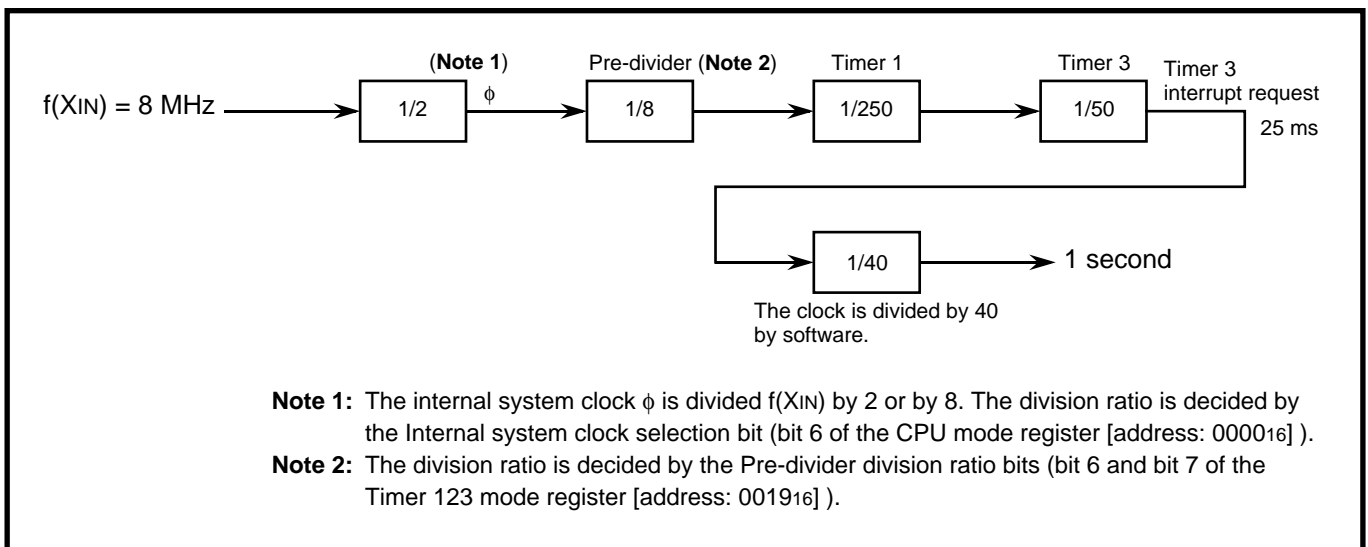


Fig. 2.3.8 Timers connection and division ratios [Clock function]

APPLICATION

2.3 Timers

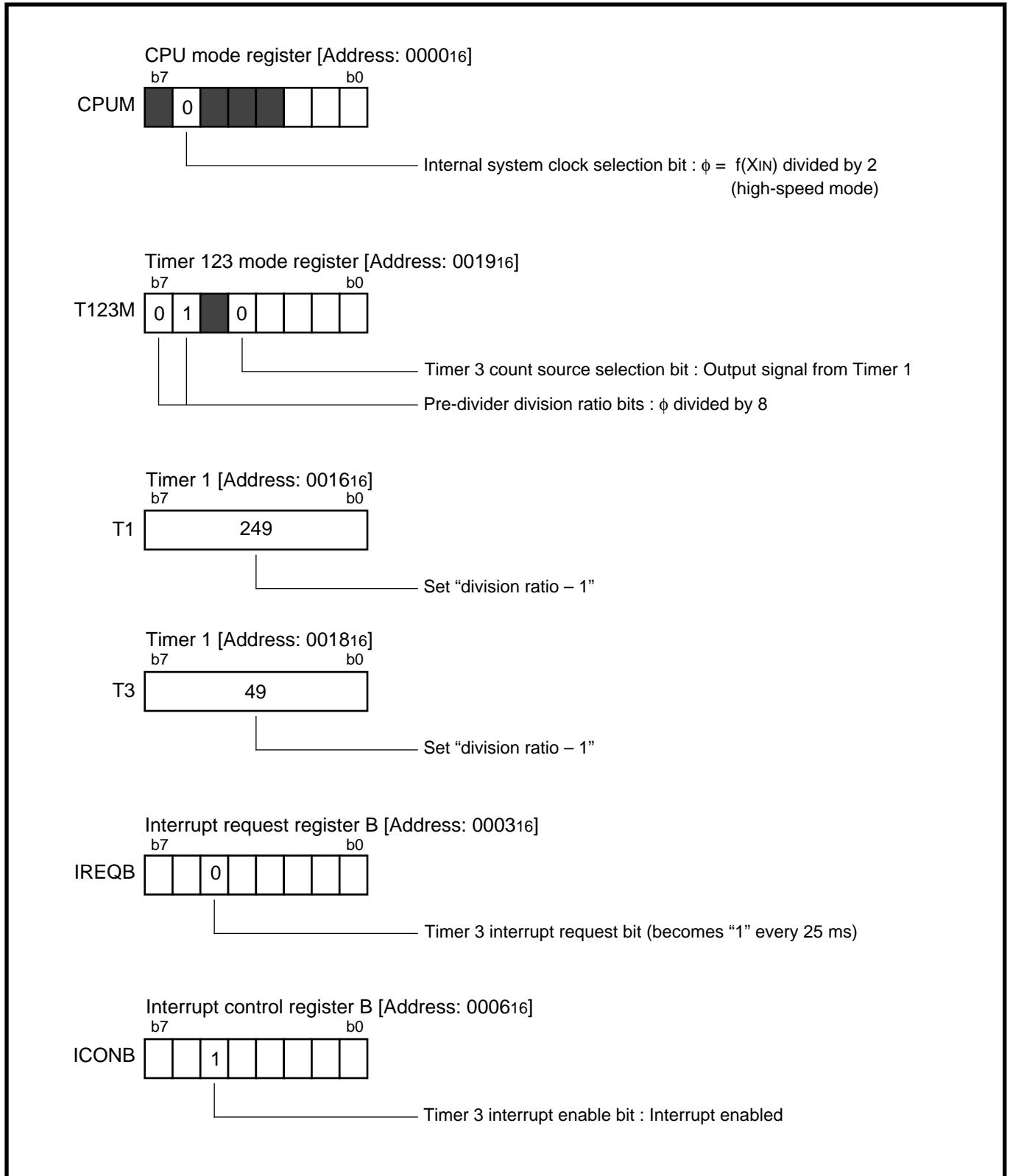


Fig. 2.3.9 Setting of related registers [Clock function]

Control procedure : Figure 2.3.10 shows a control procedure.

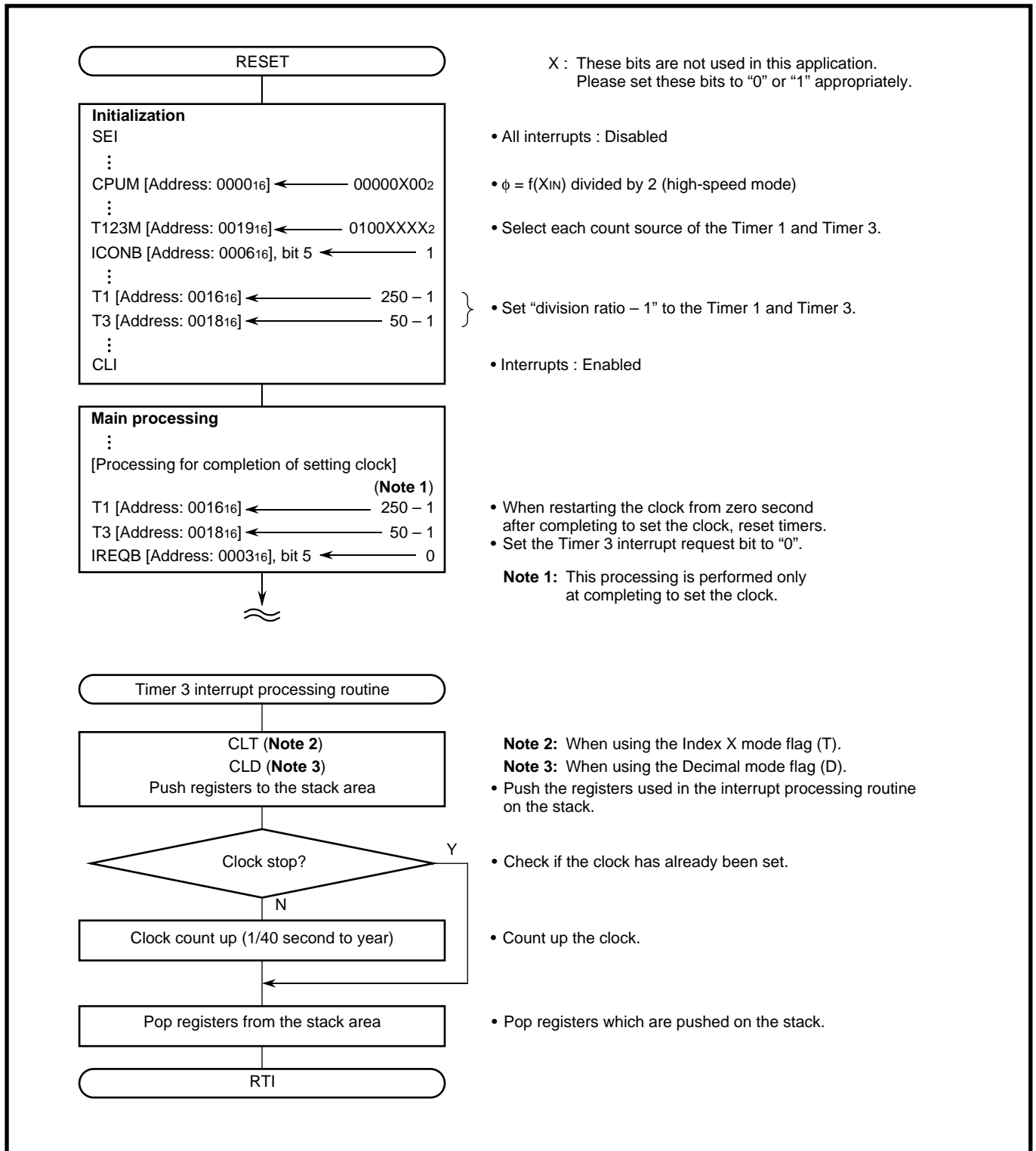


Fig. 2.3.10 Control procedure [Clock function]

APPLICATION

2.3 Timers

(3) Timer application example 2 : Measurement of frequency

Outline : To judge if the frequency is within a given range, the following two values are compared

- Timer value (representing the number of pulses at P14/CNTR0),
- Reference value.

- Specifications :**
- The pulse is input to the P14/CNTR0 pin and counted by the Timer X.
 - A count value is read out at the interval of about 2 ms (Timer Y interrupt interval).
When the count value is between 28 and 40, the input signal is judged valid.
 - Because the timer is a down-counter, the count value is compared with 227 to 215*.
* 227 to 215 = 255 (initialized value of counter) – 28 to 40 (the number of valid value)

Figure 2.3.11 shows a method for judging if input pulse exists, and Figure 2.3.12 and Figure 2.3.13 show a setting of related registers.

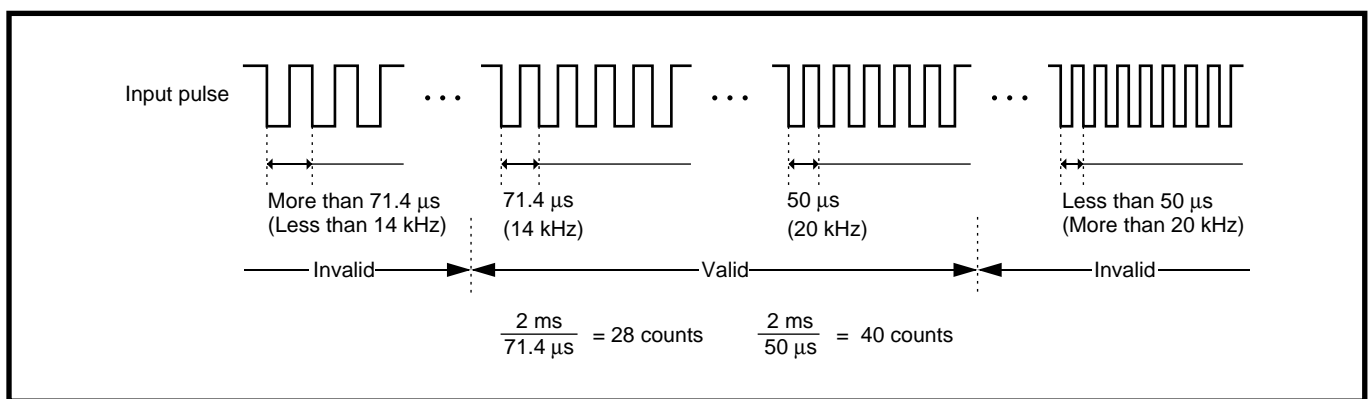


Fig. 2.3.11 A method for judging if input pulse exists

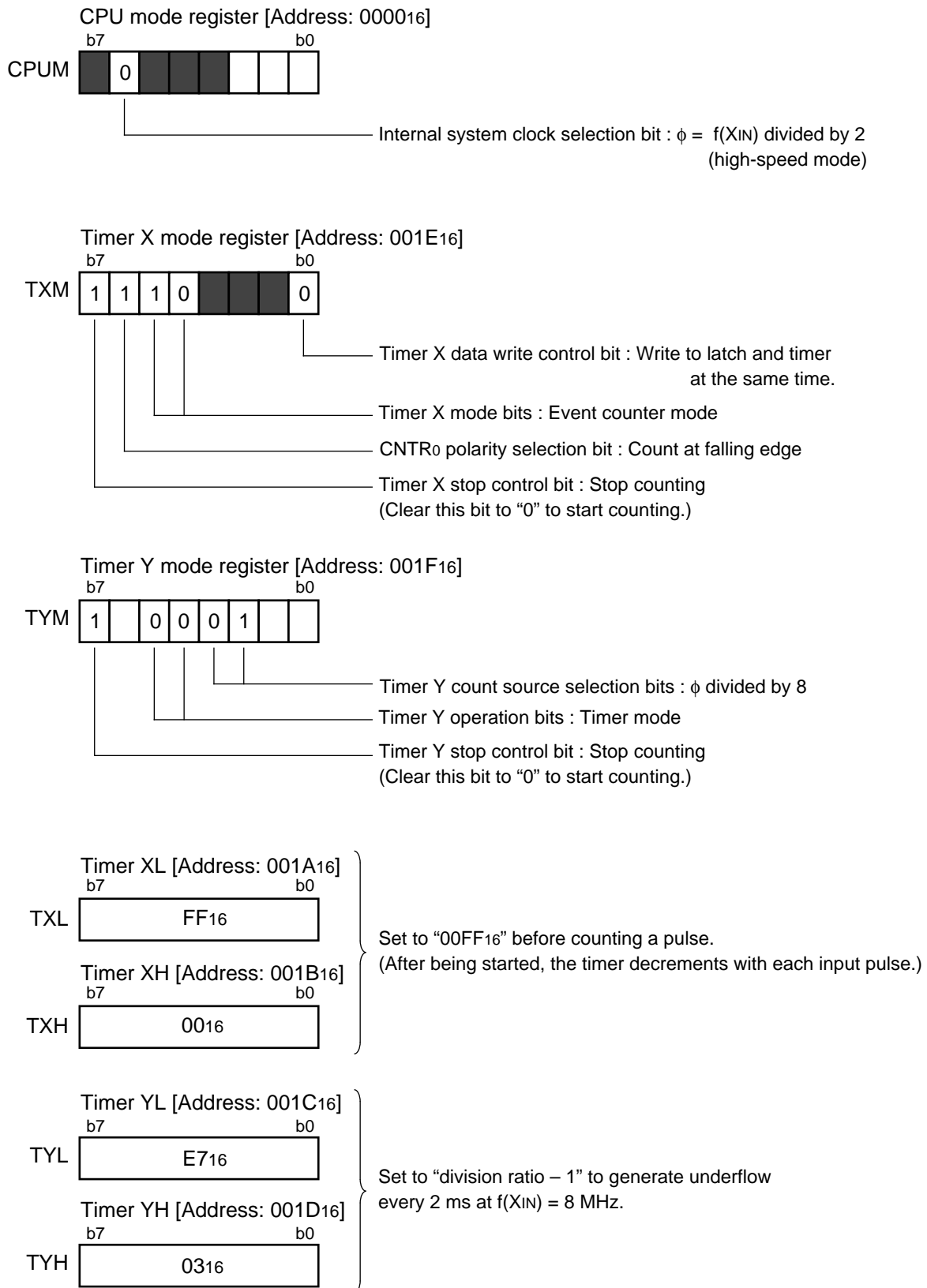


Fig. 2.3.12 Setting of related registers [Measurement of frequency] (1)

APPLICATION

2.3 Timers

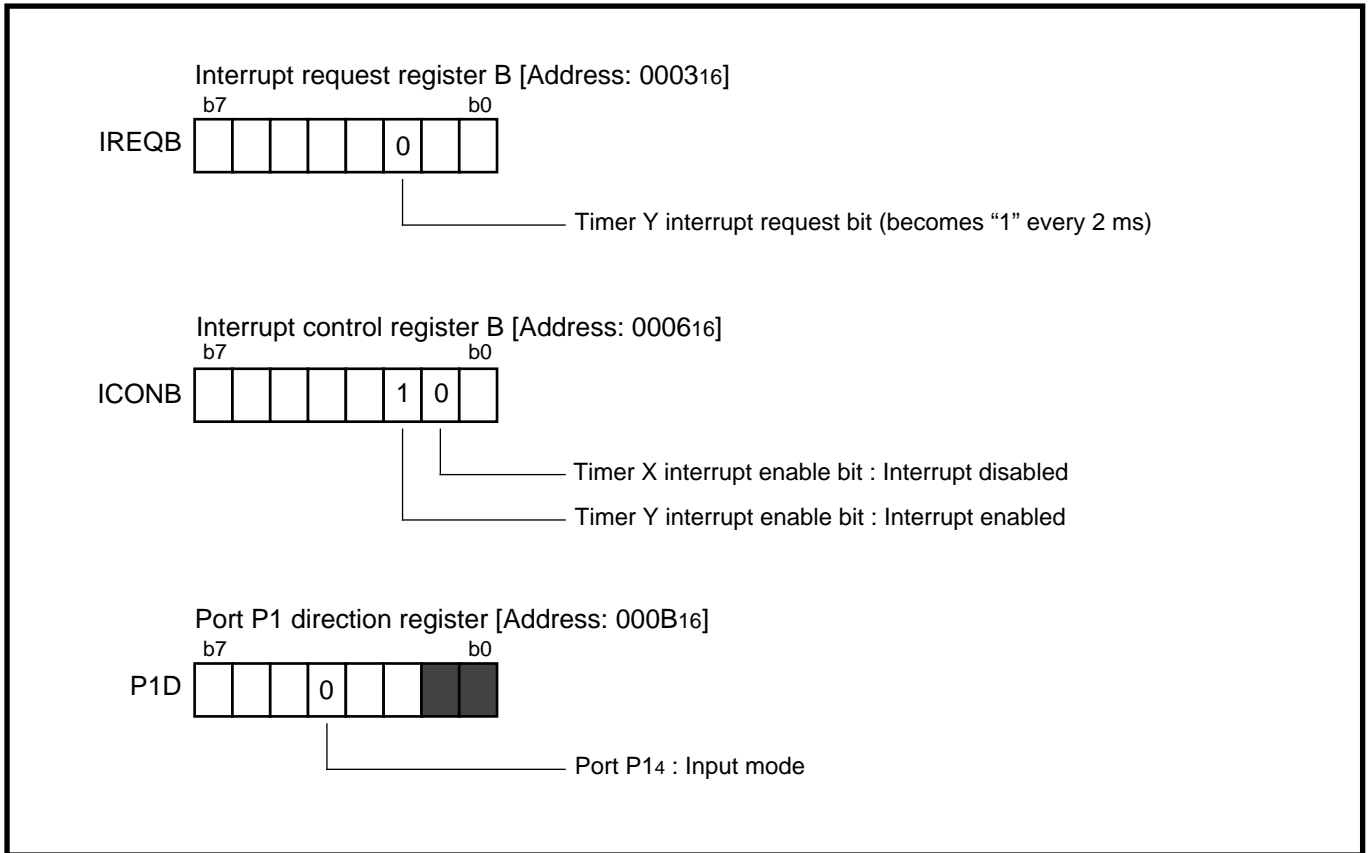


Fig. 2.3.13 Setting of related registers [Measurement of frequency] (2)

Control procedure : Figure 2.3.14 shows a control procedure.

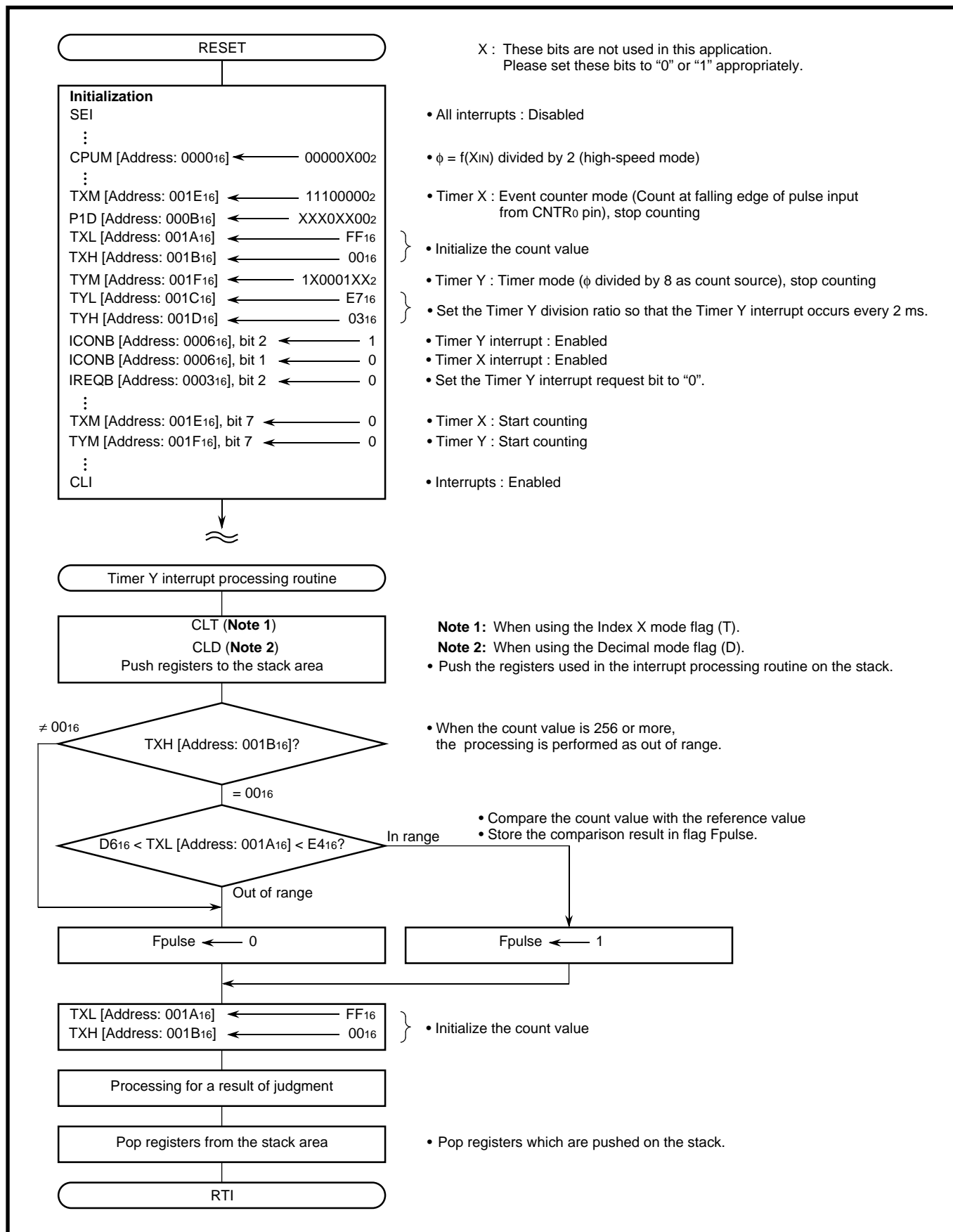


Fig. 2.3.14 Control procedure [Measurement of frequency]

APPLICATION

2.3 Timers

(4) Timer application example 3 : Measurement of pulse width of FG pulse generated by motor

Outline : The “H” level width of a pulse input to the P14/CNTR0 pin is counted by Timer X. An underflow is detected by Timer X interrupt and an end of the input pulse “H” level is detected by a CNTR0 interrupt.

Specifications : The “H” level width of FG pulse input to the P14/CNTR0 pin is counted by Timer X.
(Example : When the clock frequency is 8 MHz, the count source would be 4 μ s that is obtained by dividing the clock frequency by 32. Measurement can be made up to 262.144 ms in the range of FFFF₁₆ to 0000₁₆.)

Figure 2.3.15 shows timer connection and division ratio, and Figure 2.3.16 shows a setting of related registers.

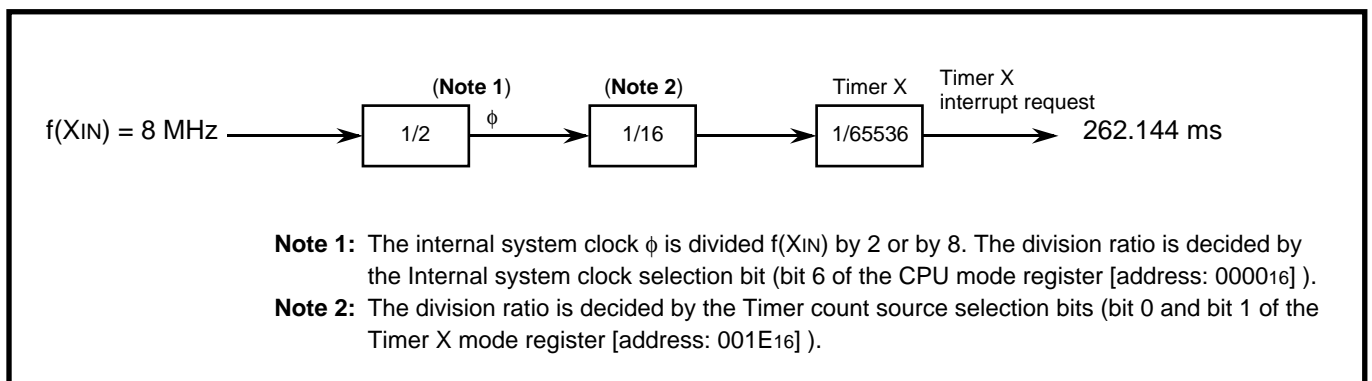


Fig. 2.3.15 Timer connection and division ratio [Measurement of pulse width]

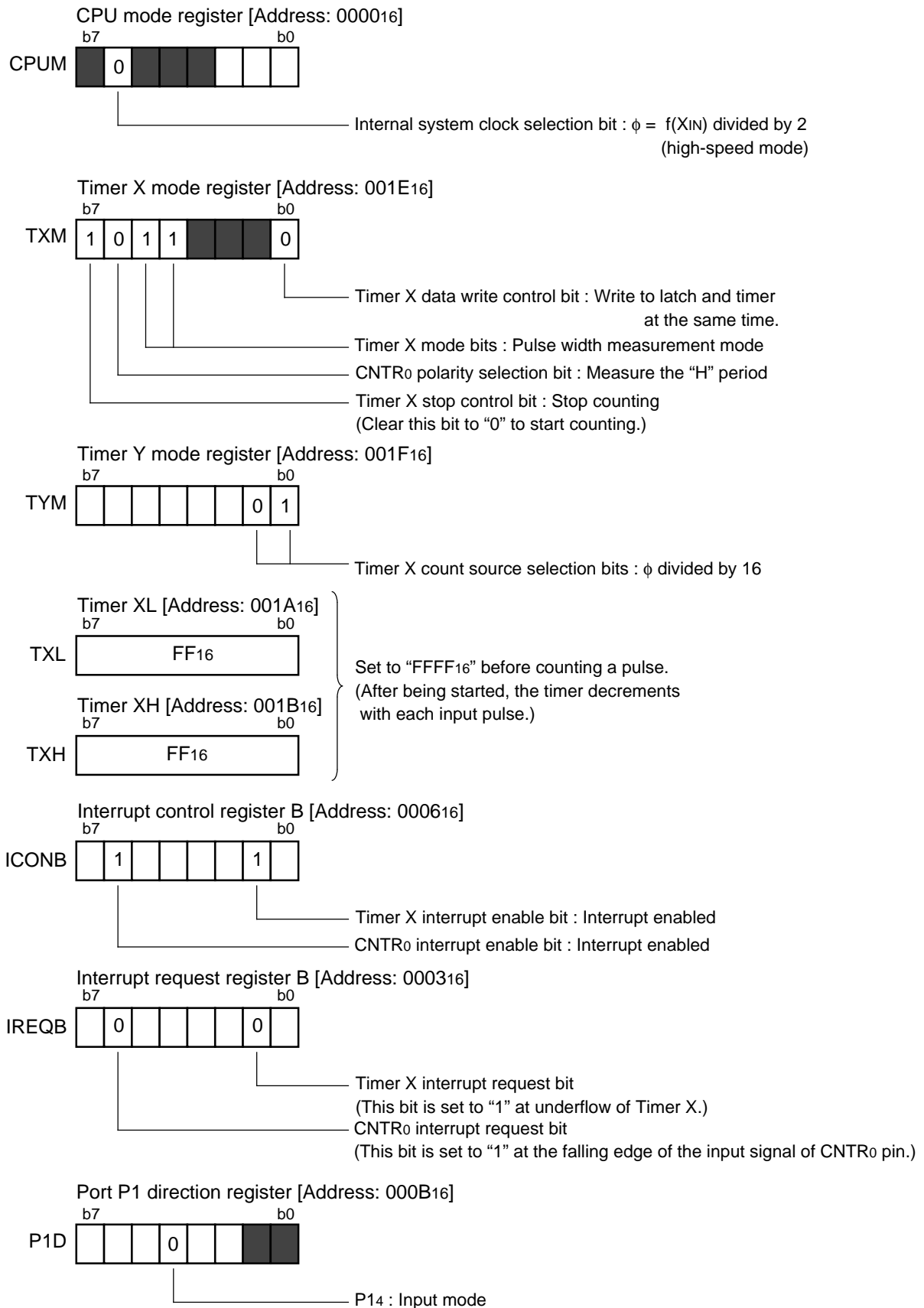


Fig. 2.3.16 Setting of related registers [Measurement of pulse width]

APPLICATION

2.3 Timers

Control procedure : Figure 2.3.17 and Figure 2.3.18 show a control procedure.

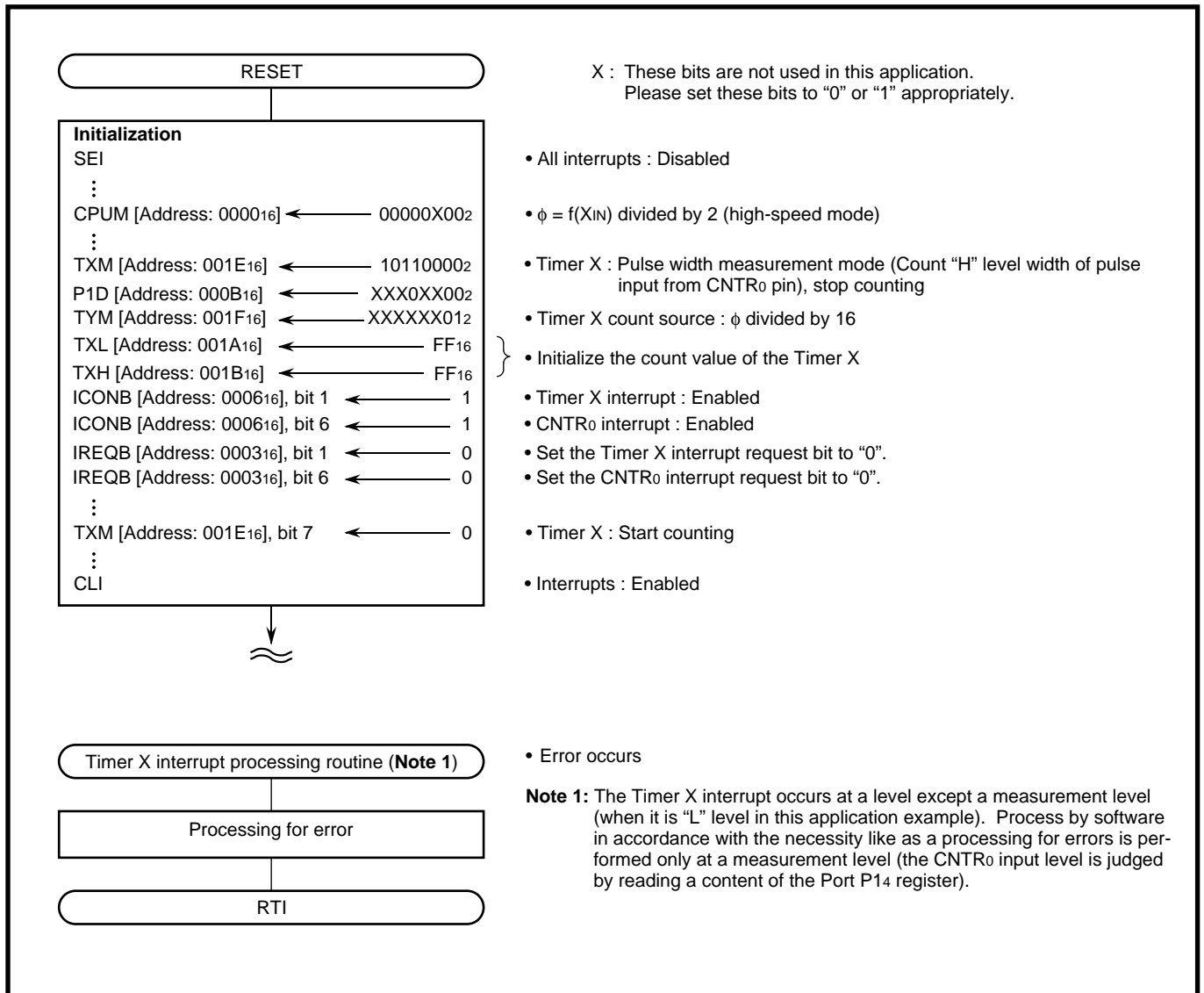
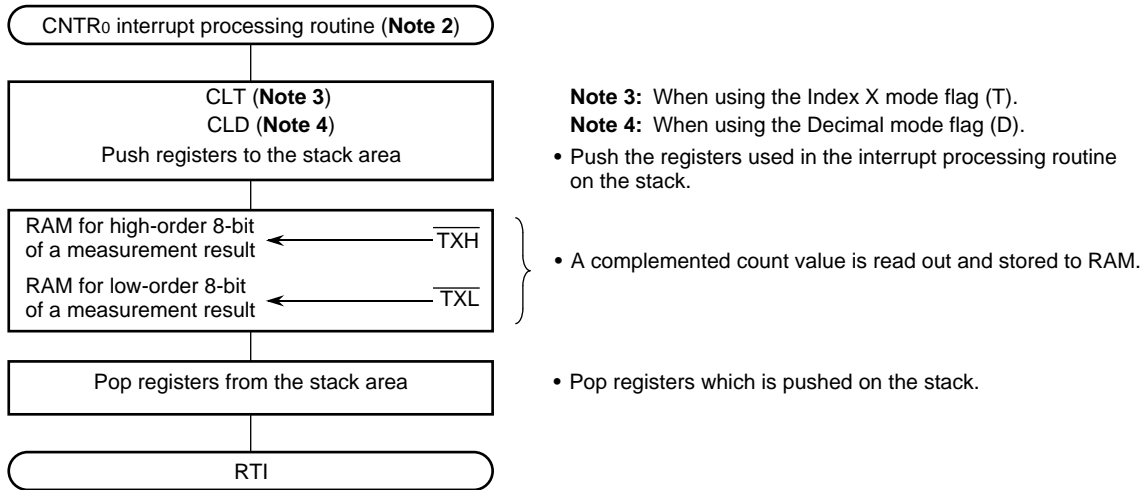


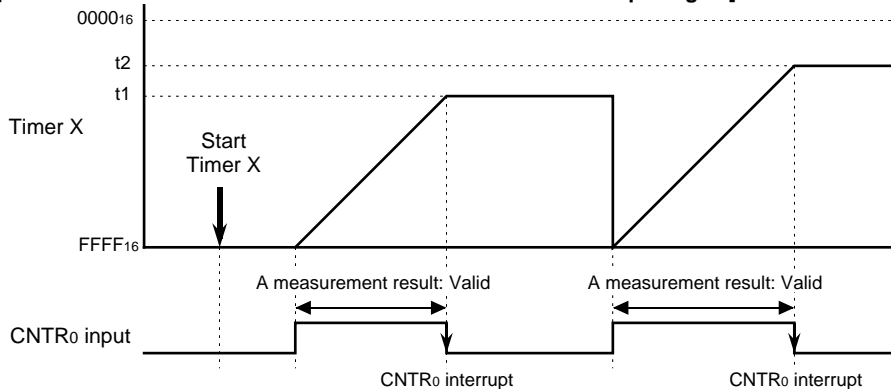
Fig. 2.3.17 Control procedure [Measurement of pulse width] (1)



Note 2: The first measurement with Timer X is invalid as shown in the following figure.

- Example:**
1. Make sure to start the Timer X count at "L" level of the CNTR0 input signal.
(A level of the CNTR0 input signal is judged by reading a content of the Port P14 register.)
 2. Be sure to invalidate the first CNTR0 interrupt after starting the Timer X count.

[When the Timer X count is started at "L" level of the CNTR0 input signal]



[When the Timer X count is started at "H" level of the CNTR0 input signal]

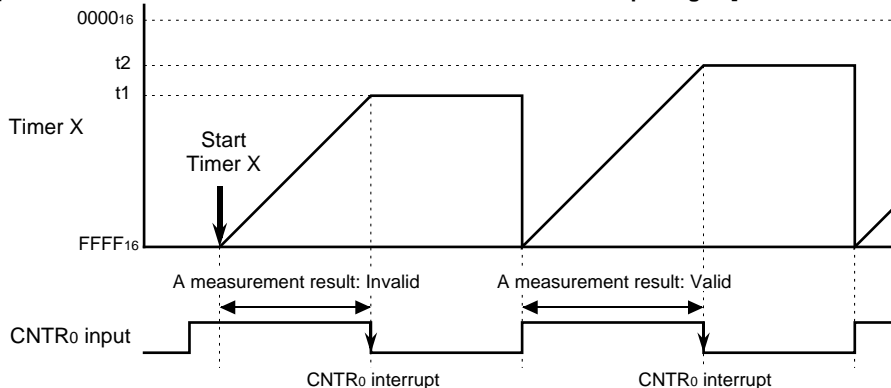


Fig. 2.3.18 Control procedure [Measurement of pulse width] (2)

APPLICATION

2.4 Controller Area Network (CAN) module

2.4 Controller Area Network (CAN) module

This section outlines the Controller Area Network (CAN) module of the MCU. First the module's architecture and the programming interface with its related special function registers are explained. Second, after having defined the fundamental operational modes of the module, the programming sequences to initialize and reset the module are clarified. Third, the module's communication functions, that is acceptance filtering, reception, and transmission are discussed in detail. The closing of the section goes into the interrupt capabilities, CAN error conditions, and the wake-up function.

2.4.1 Description

The CAN module can be characterized as follows:

- *Compatibility*: The module's protocol controller complies with CAN specification version 2.0, part B as defined by Bosch in September 1991 (this document is later on called *CAN specification*). The receive and transmit sections of the module are capable of handling standard (11-bit identifier) as well as extended (29-bit identifier) format frames of either data or remote type.
- *CPU interface*: The module is memory mapped with sixteen control registers, two interrupt control registers, one transmission, and two receive buffer register sets.
- *Acceptance filtering*: Up to 29-bit identifiers can be filtered by using one set of acceptance mask and code registers.
- *Multi-channel interrupt capability*: Separate interrupt vectors for each event (successful transmission, successful reception, overrun, error passive, bus off) allow efficient and rapid interrupt service routine operation.
- *Low power (sleep) mode*: To reduce power consumption, the module can be set to sleep mode; wake-up from CAN traffic is supported by a dedicated interrupt source and vector.
- *Priority based message management support*: To cope with the problem of priority inversion, the contents of the transmission buffer can be released, in order to let another higher priority message to take over.
- *Baud rate prescaler*: This programmable divider provides a flexible baud rate selection up to 625kbps (at $f(XIN) = 10\text{MHz}$).
- *Programmable bit timing*: The durations of propagation time segment (PTS), phase buffer segments 1 (PBS1), and 2 (PBS2) are programmable.
- *Physical interface*: A two terminal CMOS-compatible interface (formed by ports P3₁ and P3₂) allows direct connection to the most popular transceiver devices (e.g. ISO 11898, ISO 11519).

Refer to the block diagram in [Figure 2.4.1](#).

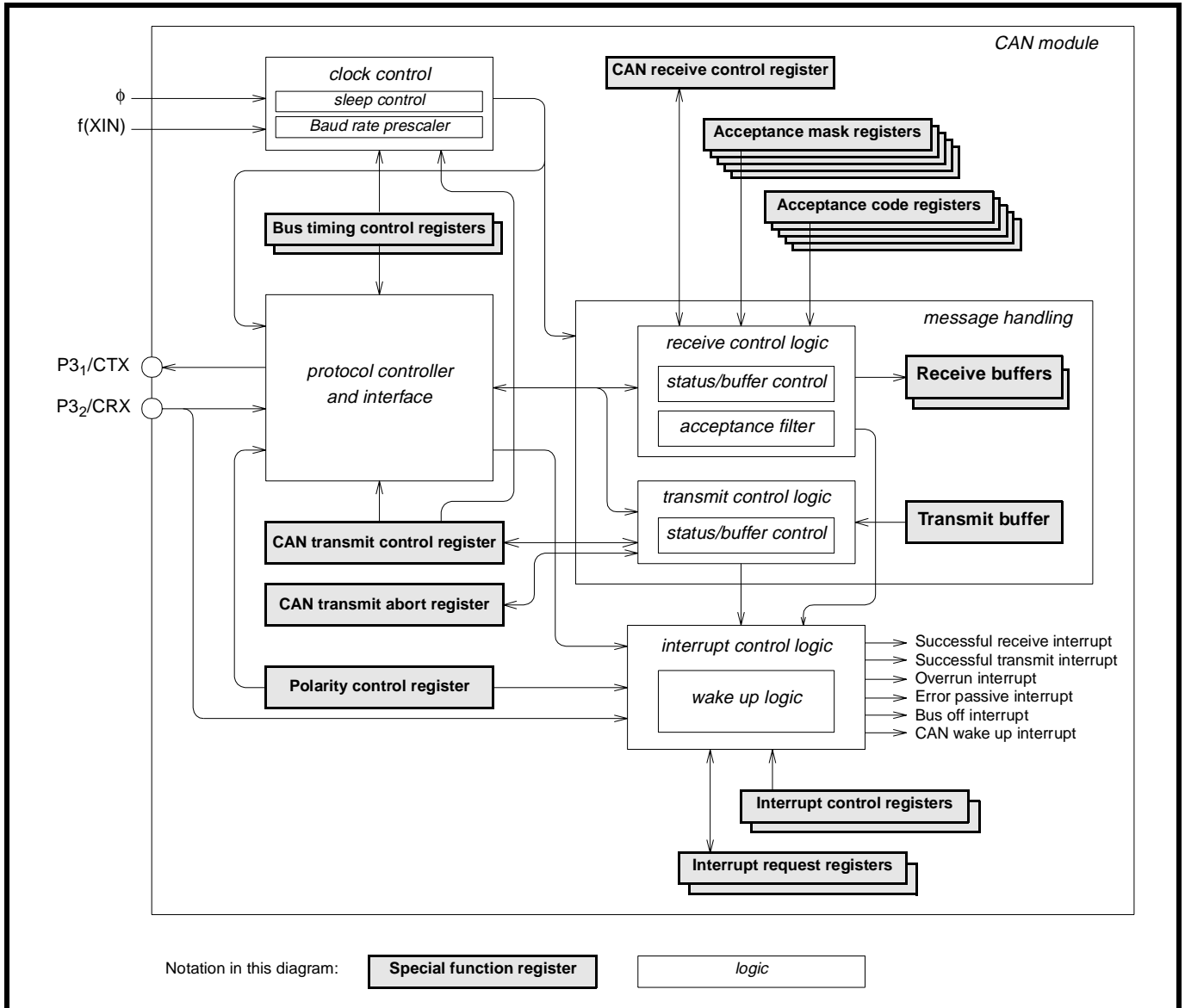


Fig. 2.4.1 Block diagram of CAN module

2.4.2 Special function register map

The CAN module's programming interface consists of the registers listed below:

- Transmit control register
- Bus timing control registers
- Acceptance code and mask registers
- Receive control register
- Transmit abort register
- Transmit/receive buffers
- Polarity control register
- Interrupt request and control registers

Figure 2.4.2 shows the memory map of these registers. The next section explains each register in detail; for the polarity control register and the interrupt registers refer to section 2.2.

APPLICATION

2.4 Controller Area Network (CAN) module

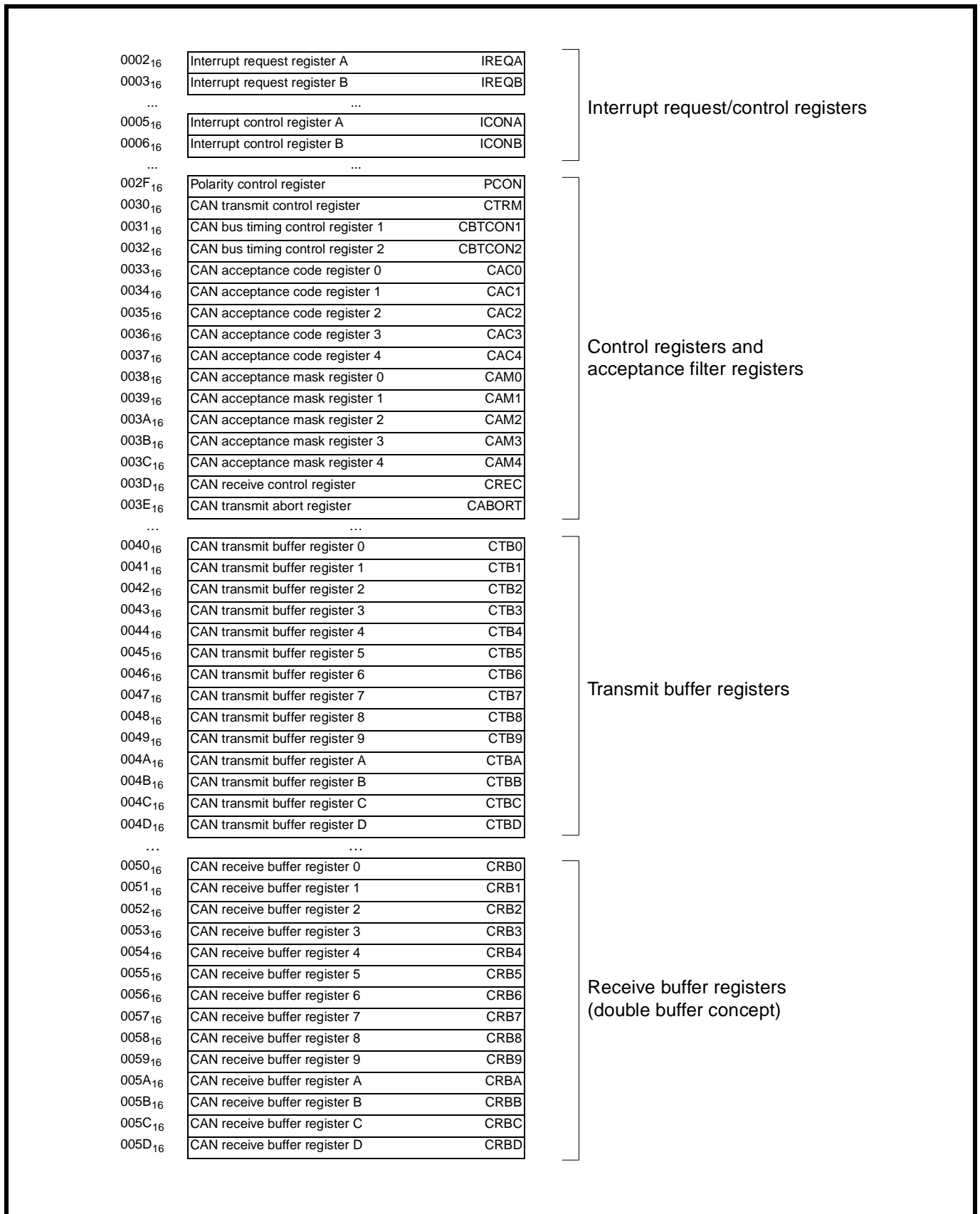


Fig. 2.4.2 Memory map of CAN related registers

2.4.3 Related registers

This section comprises the description of special function registers allocated to the CAN module.

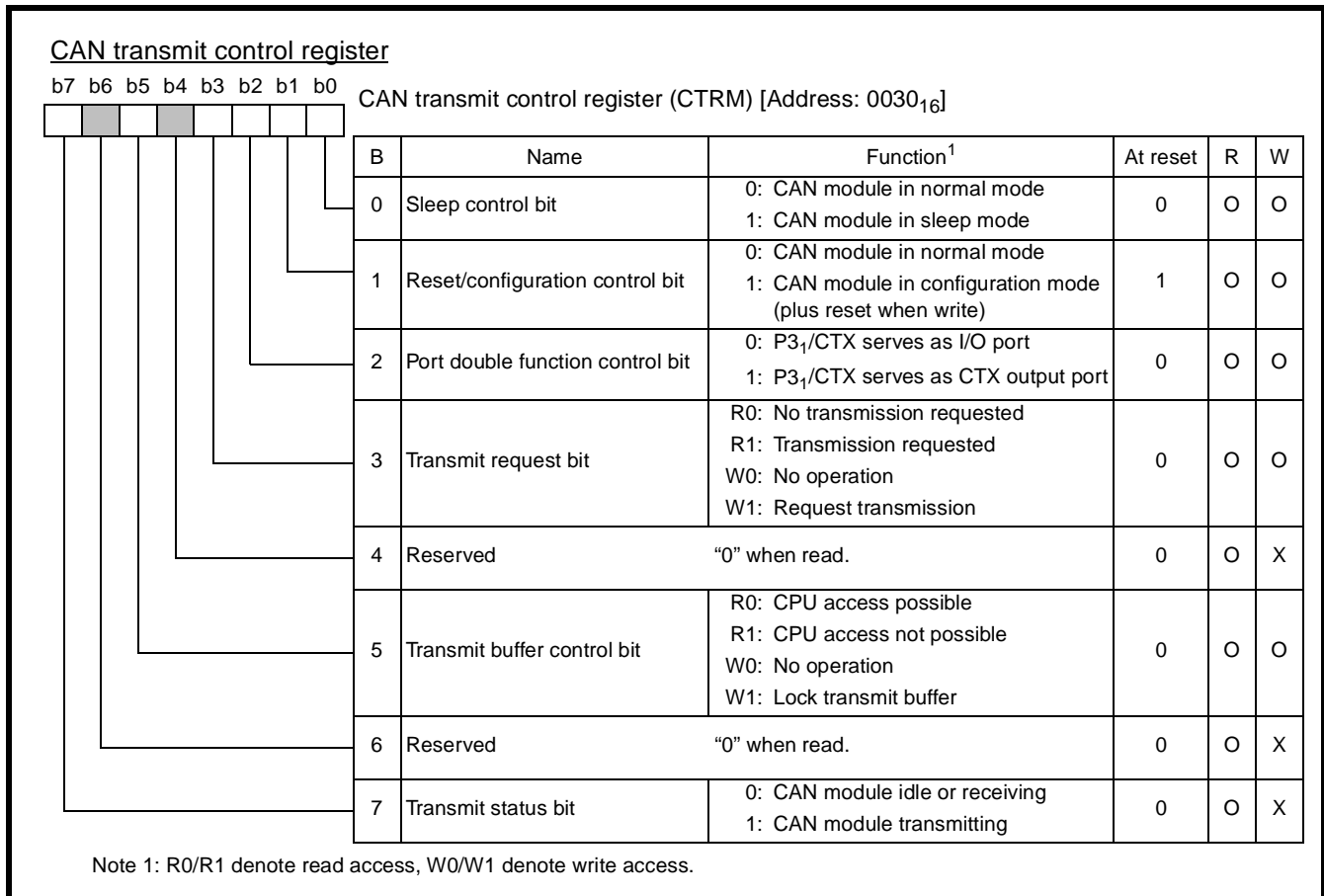


Fig. 2.4.3 Structure of CAN transmit control register (CTRM)

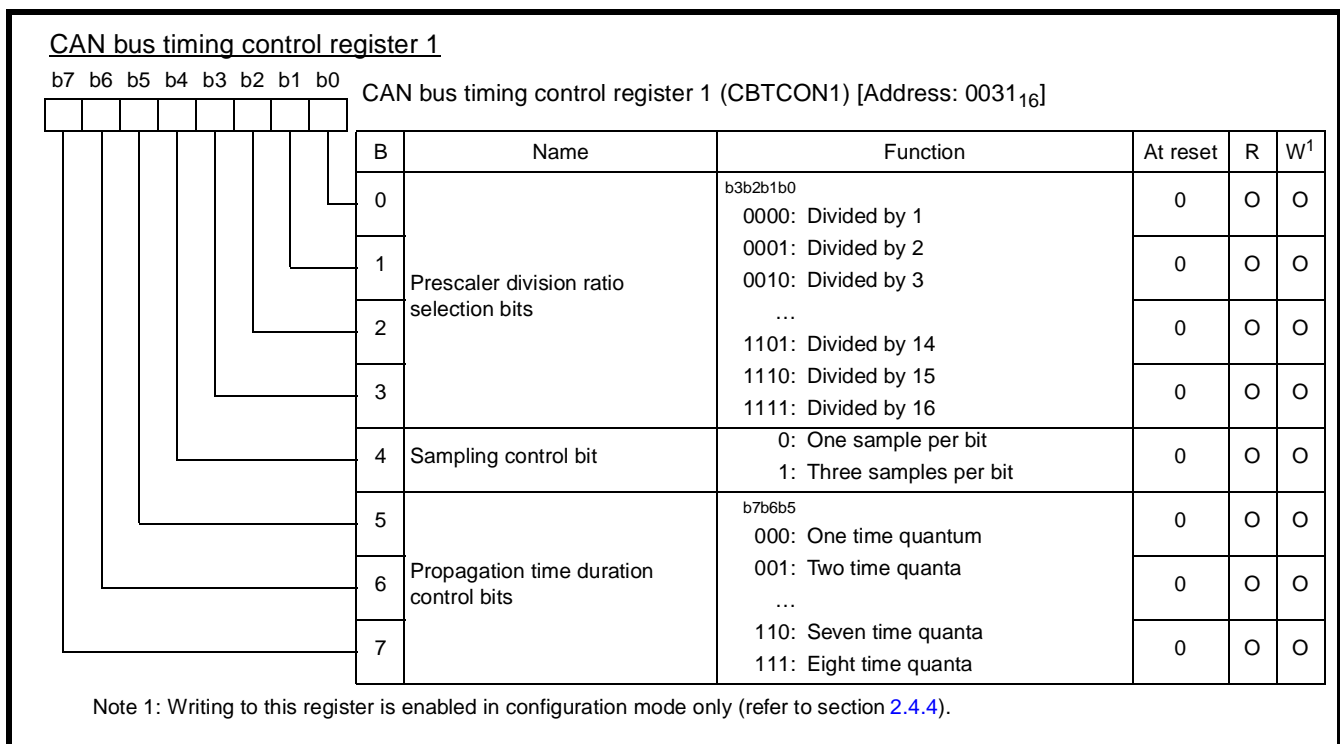


Fig. 2.4.4 Structure of CAN bus timing control register 1 (CBTCON1)

APPLICATION

2.4 Controller Area Network (CAN) module

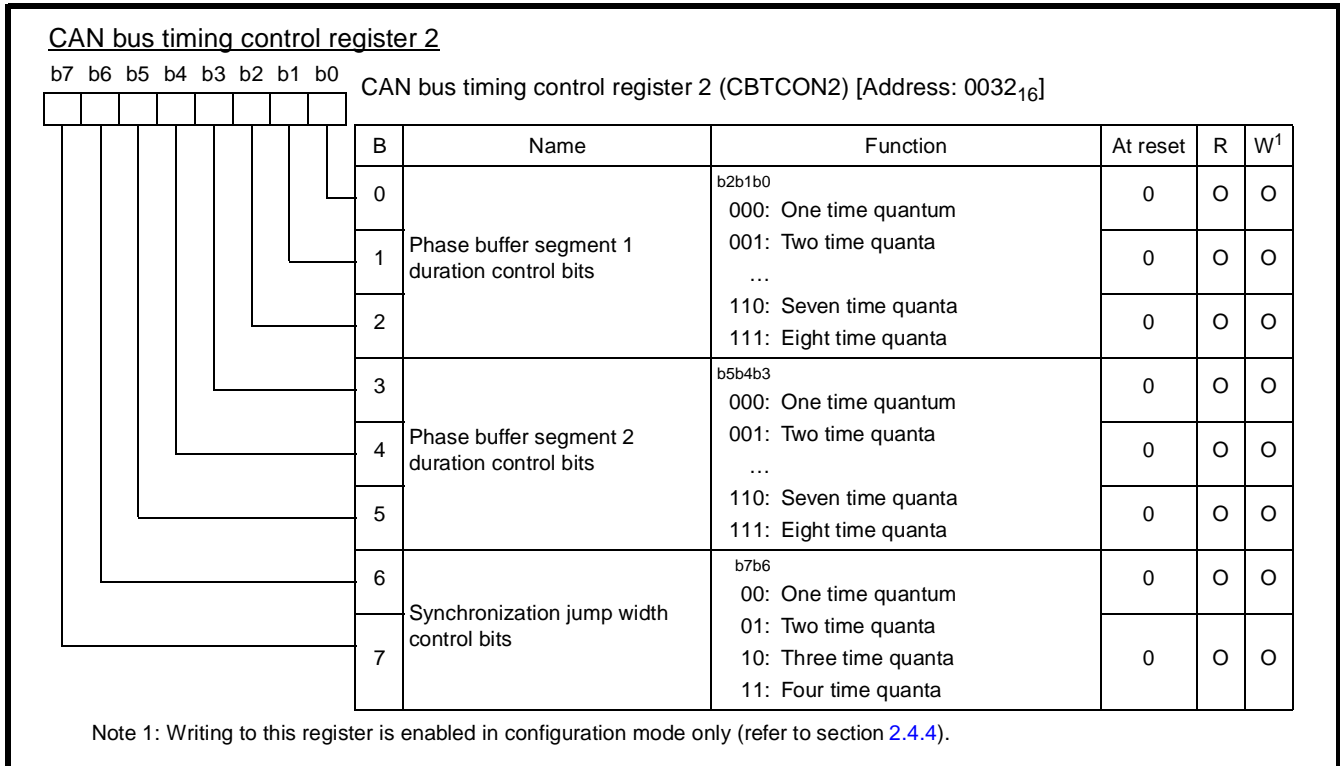


Fig. 2.4.5 Structure of CAN bus timing control register 2 (CBTCON2)

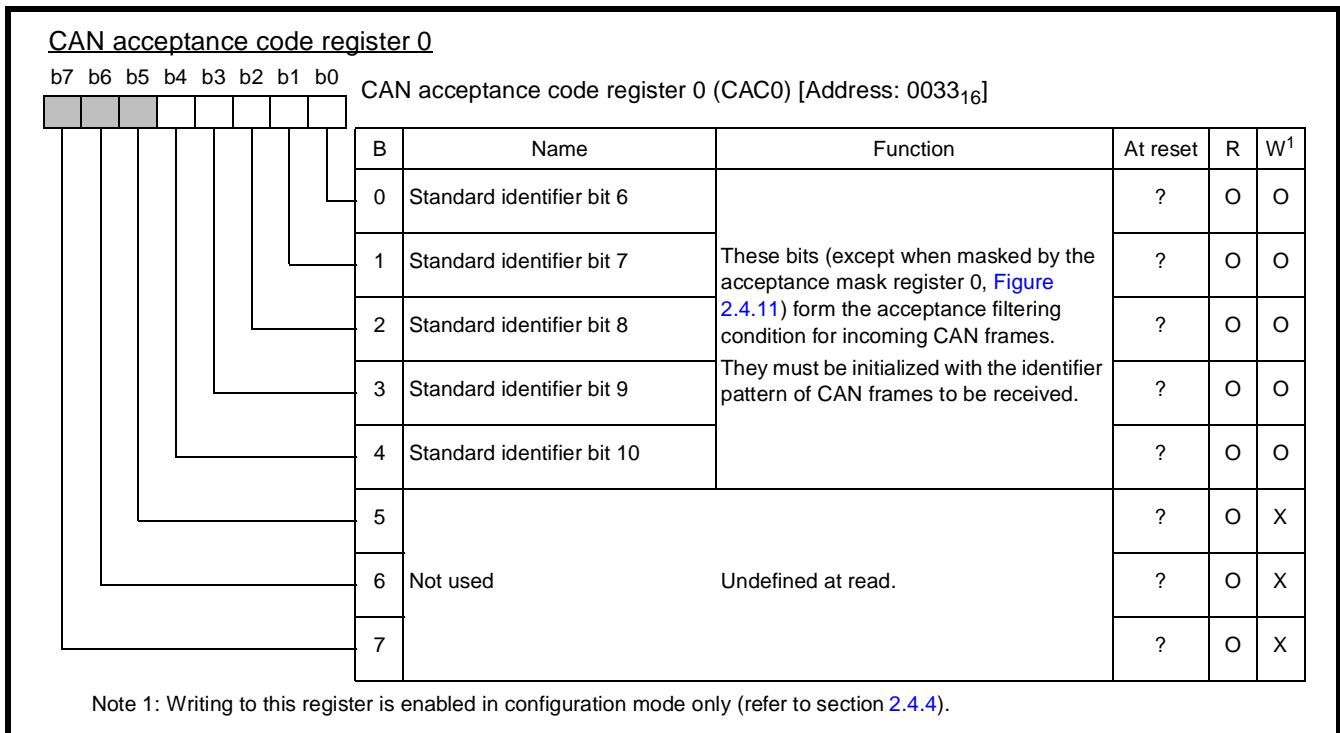


Fig. 2.4.6 Structure of CAN acceptance code register 0 (CAC0)

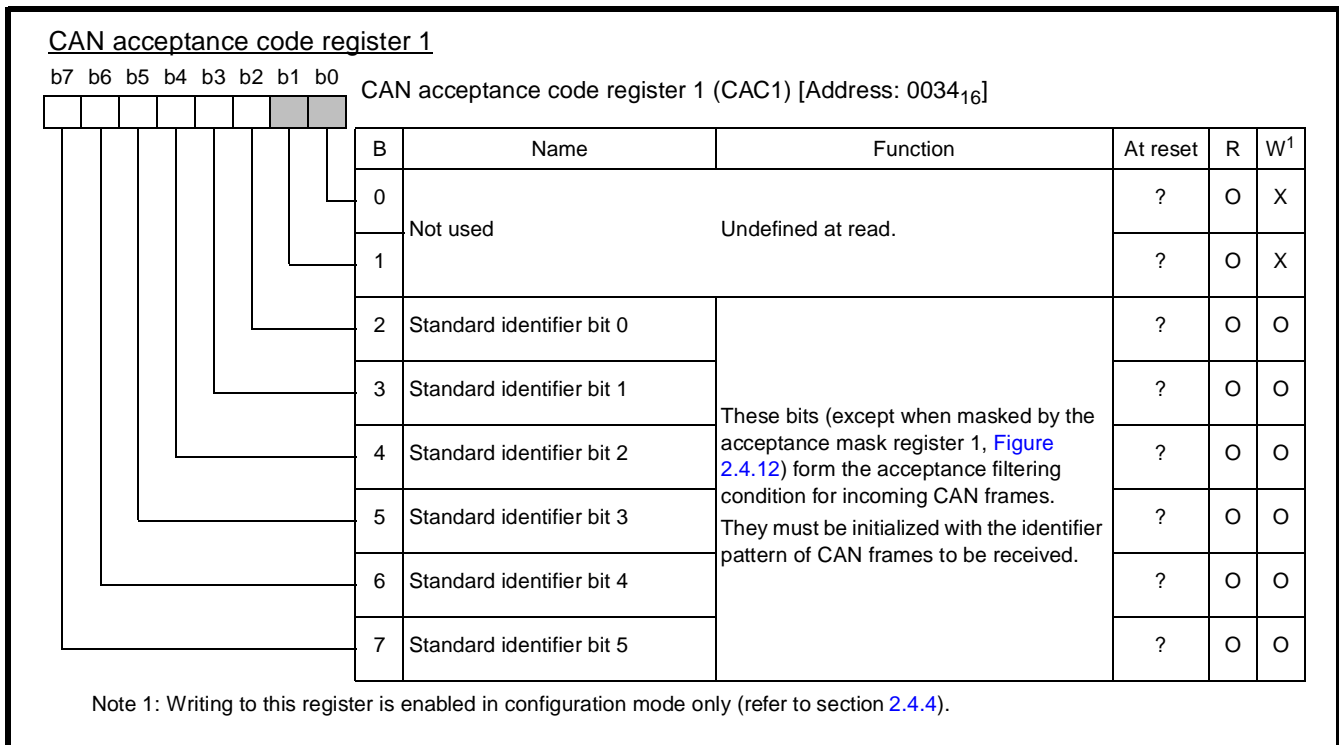


Fig. 2.4.7 Structure of CAN acceptance code register 1 (CAC1)

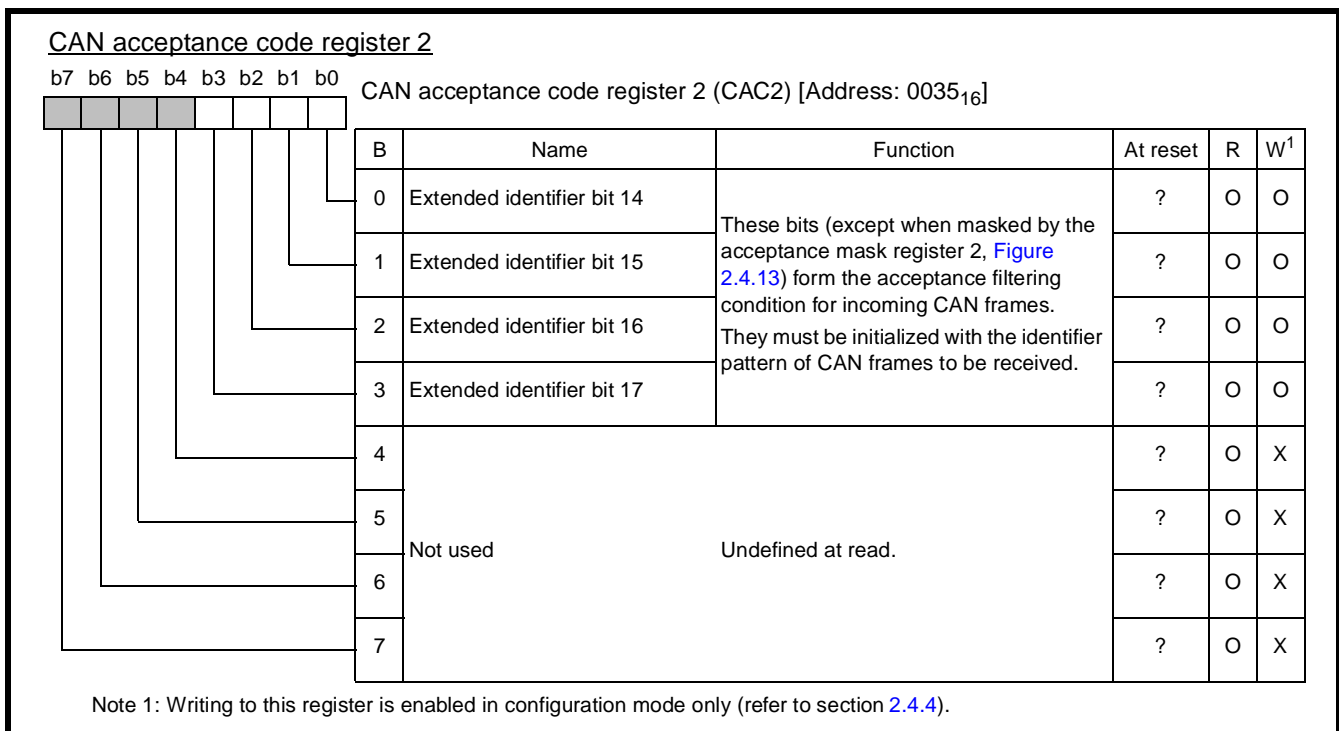


Fig. 2.4.8 Structure of CAN acceptance code register 2 (CAC2)

APPLICATION

2.4 Controller Area Network (CAN) module

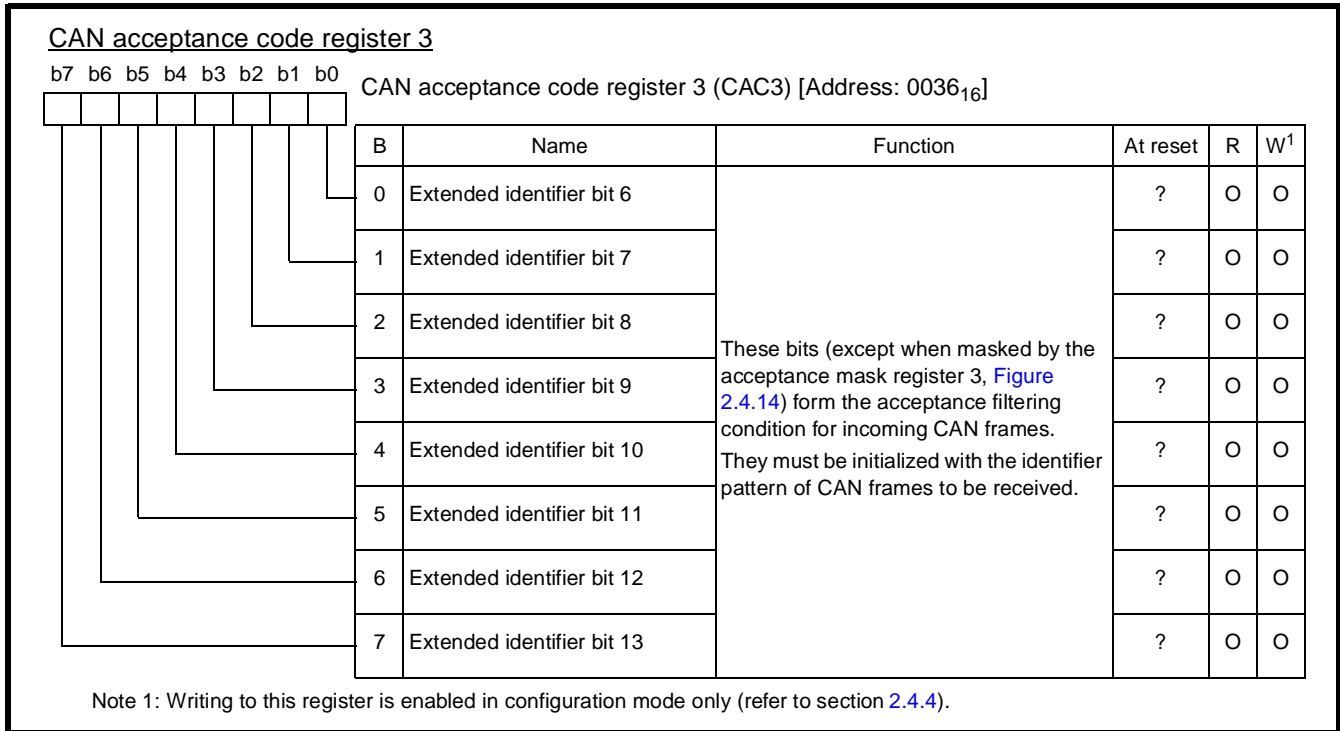


Fig. 2.4.9 Structure of CAN acceptance code register 3 (CAC3)

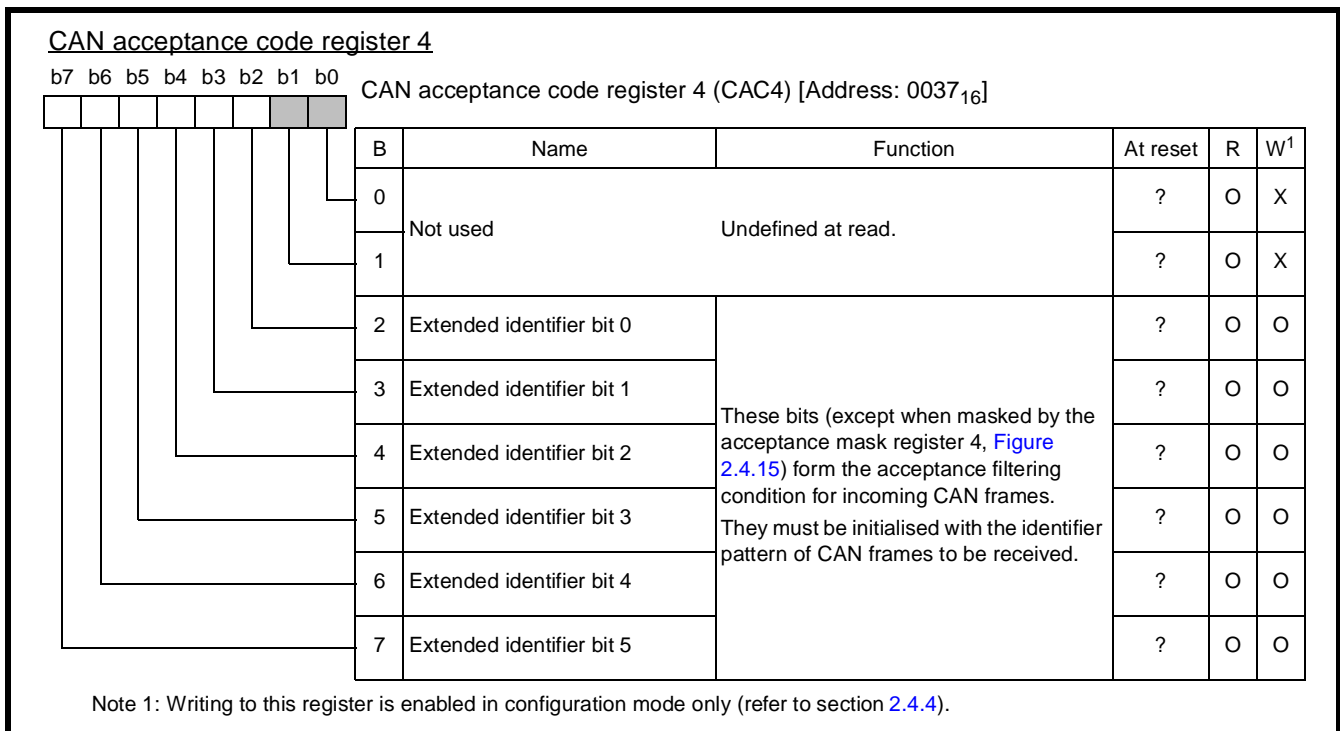


Fig. 2.4.10 Structure of CAN acceptance code register 4 (CAC4)

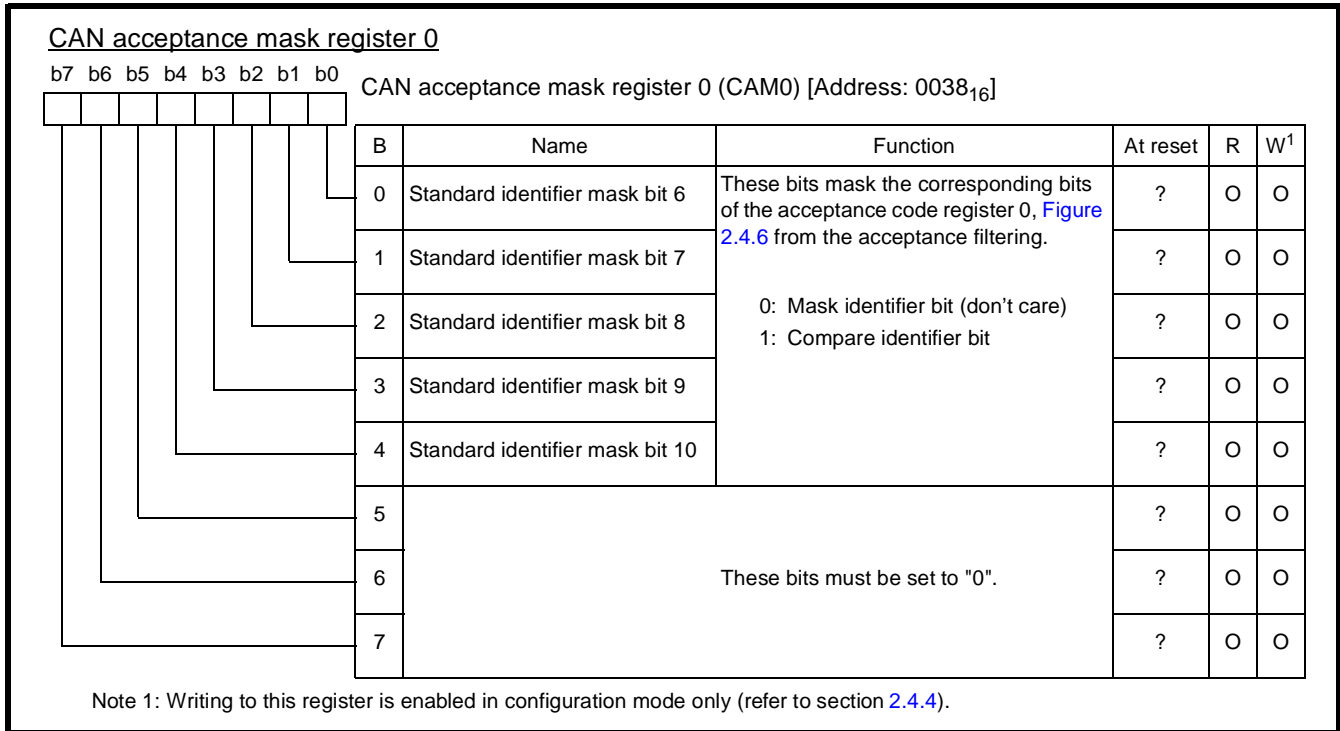


Fig. 2.4.11 Structure of CAN acceptance mask register 0 (CAM0)

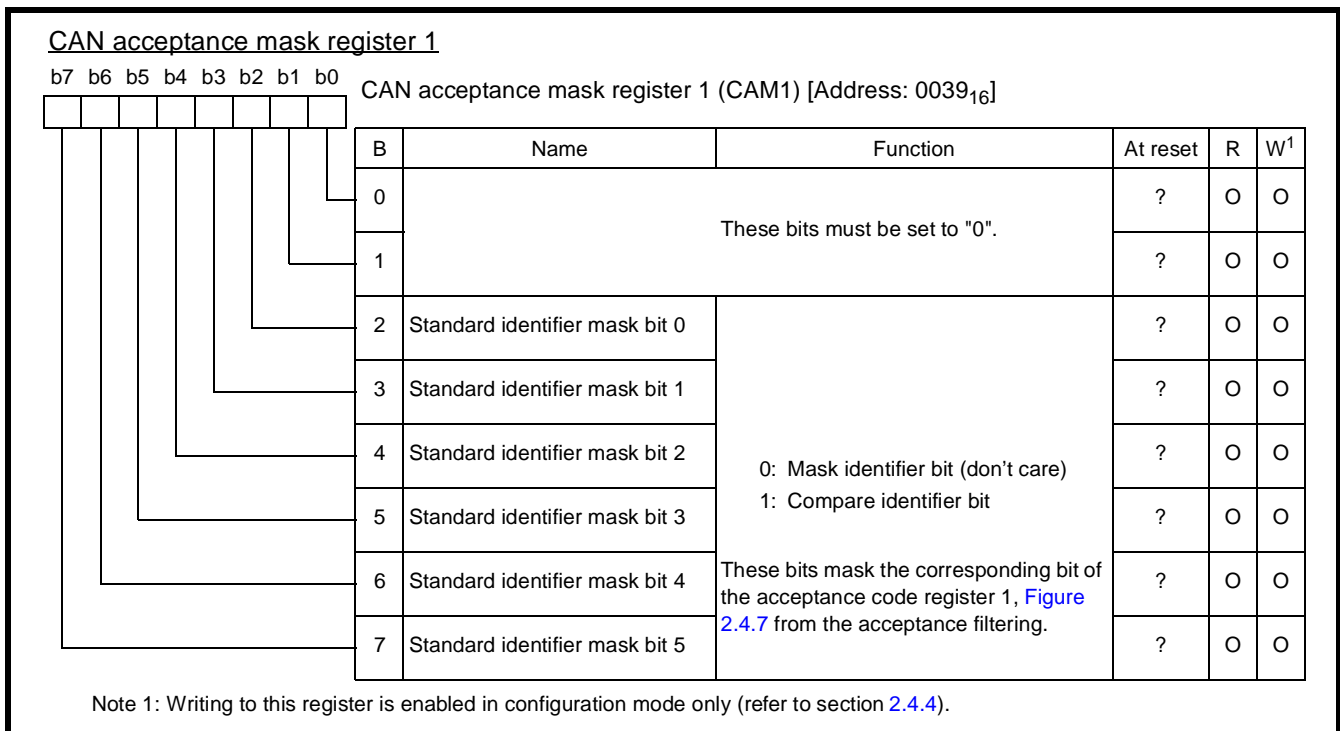


Fig. 2.4.12 Structure of CAN acceptance mask register 1 (CAM1)

APPLICATION

2.4 Controller Area Network (CAN) module

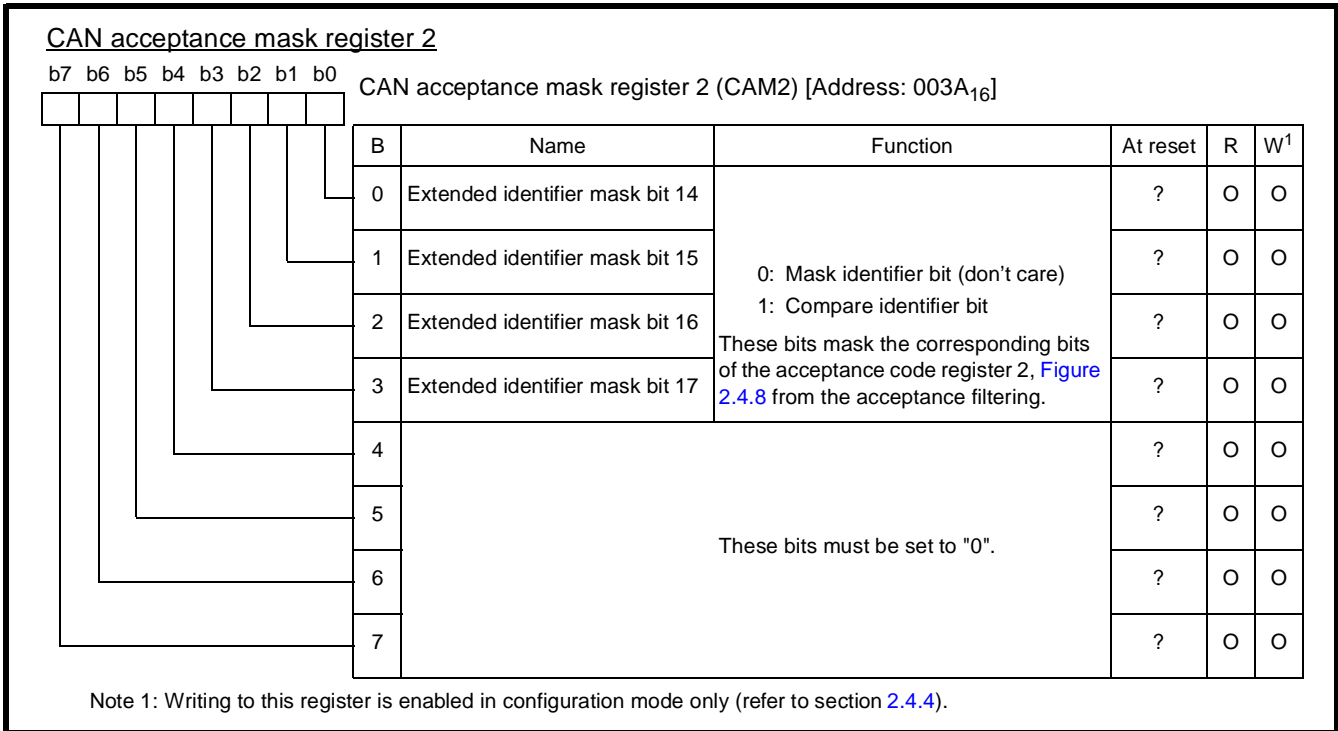


Fig. 2.4.13 Structure of CAN acceptance mask register 2 (CAM2)

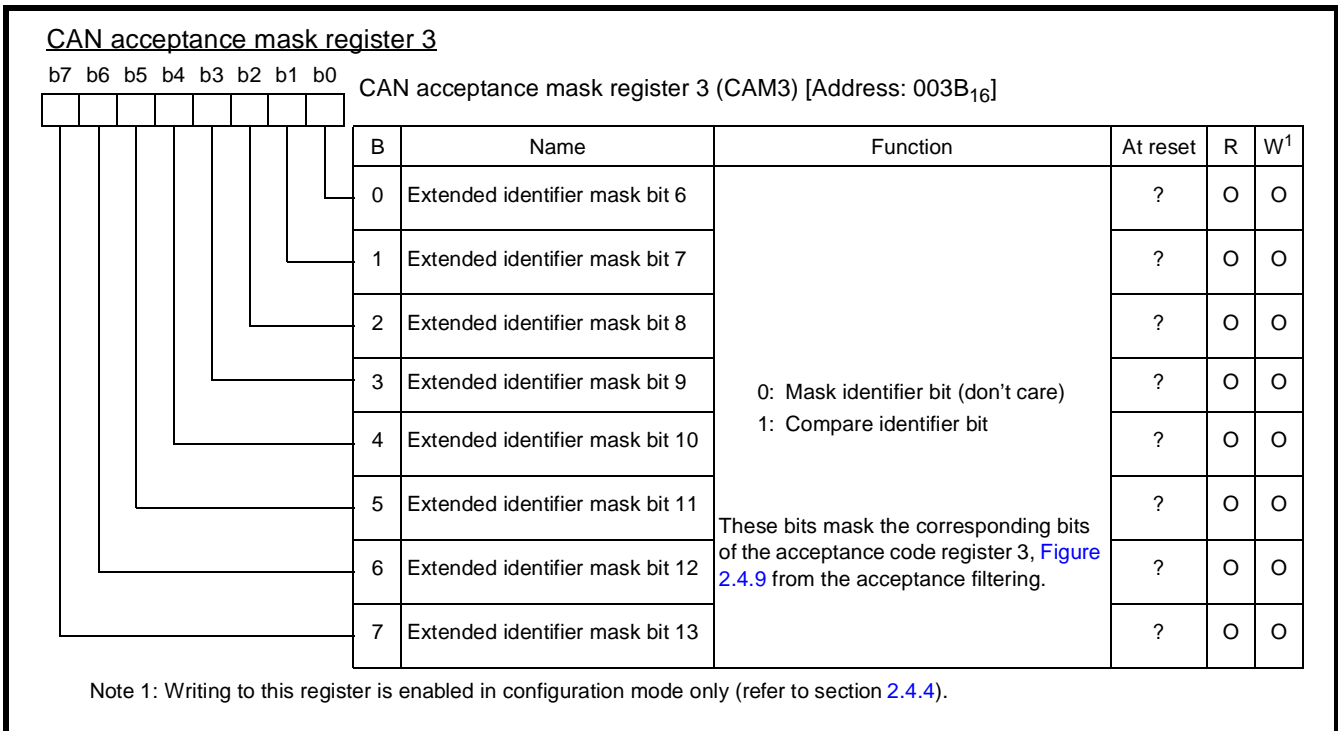


Fig. 2.4.14 Structure of CAN acceptance mask register 3 (CAM3)

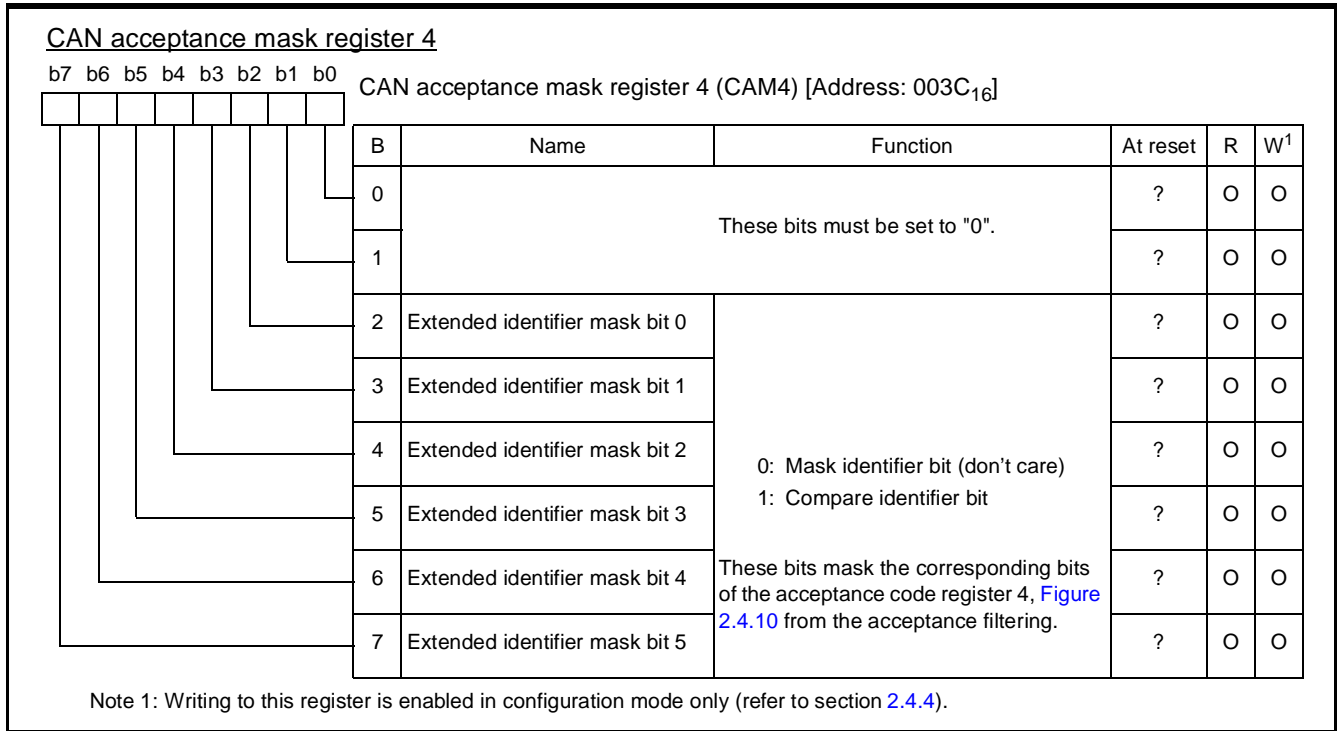


Fig. 2.4.15 Structure of CAN acceptance mask register 4 (CAM4)

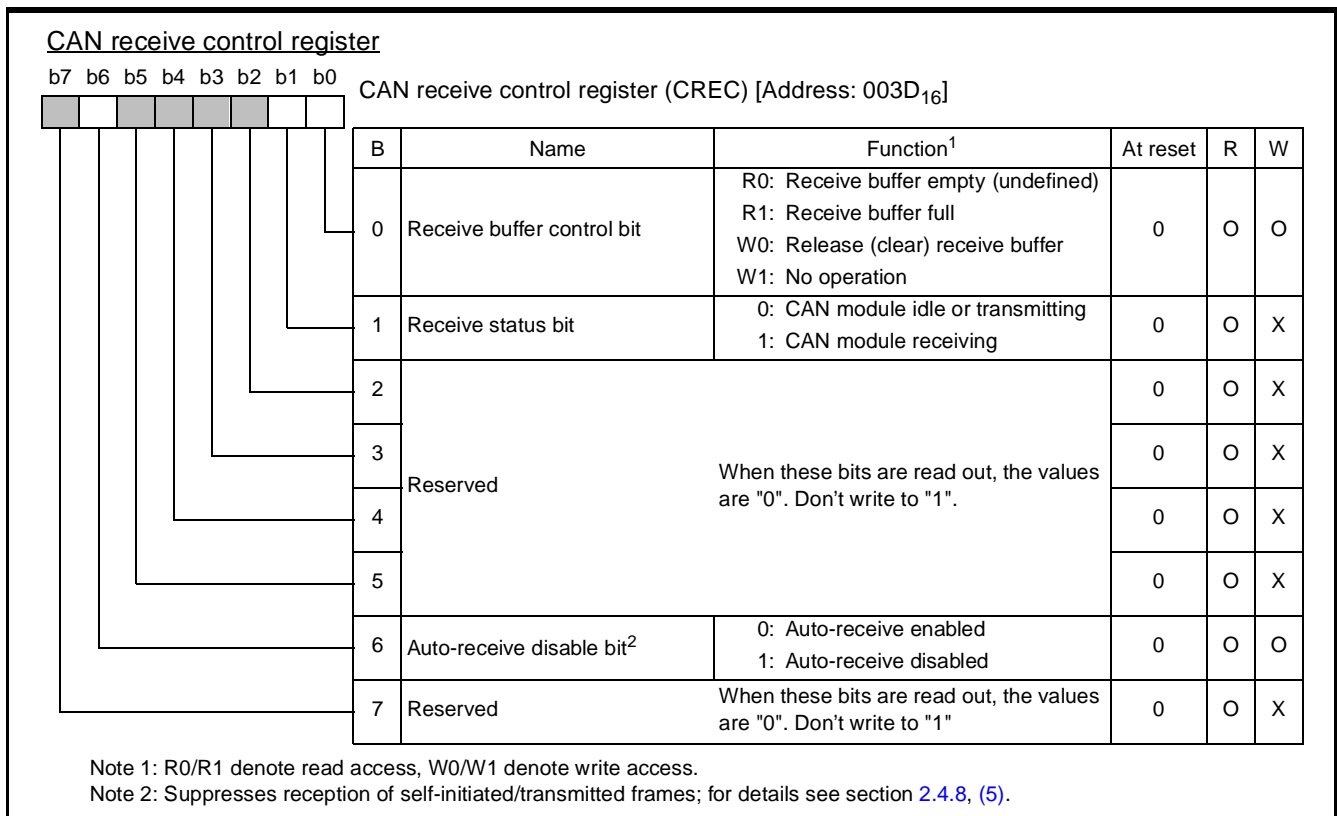


Fig. 2.4.16 Structure of CAN receive control register (CREC)

APPLICATION

2.4 Controller Area Network (CAN) module

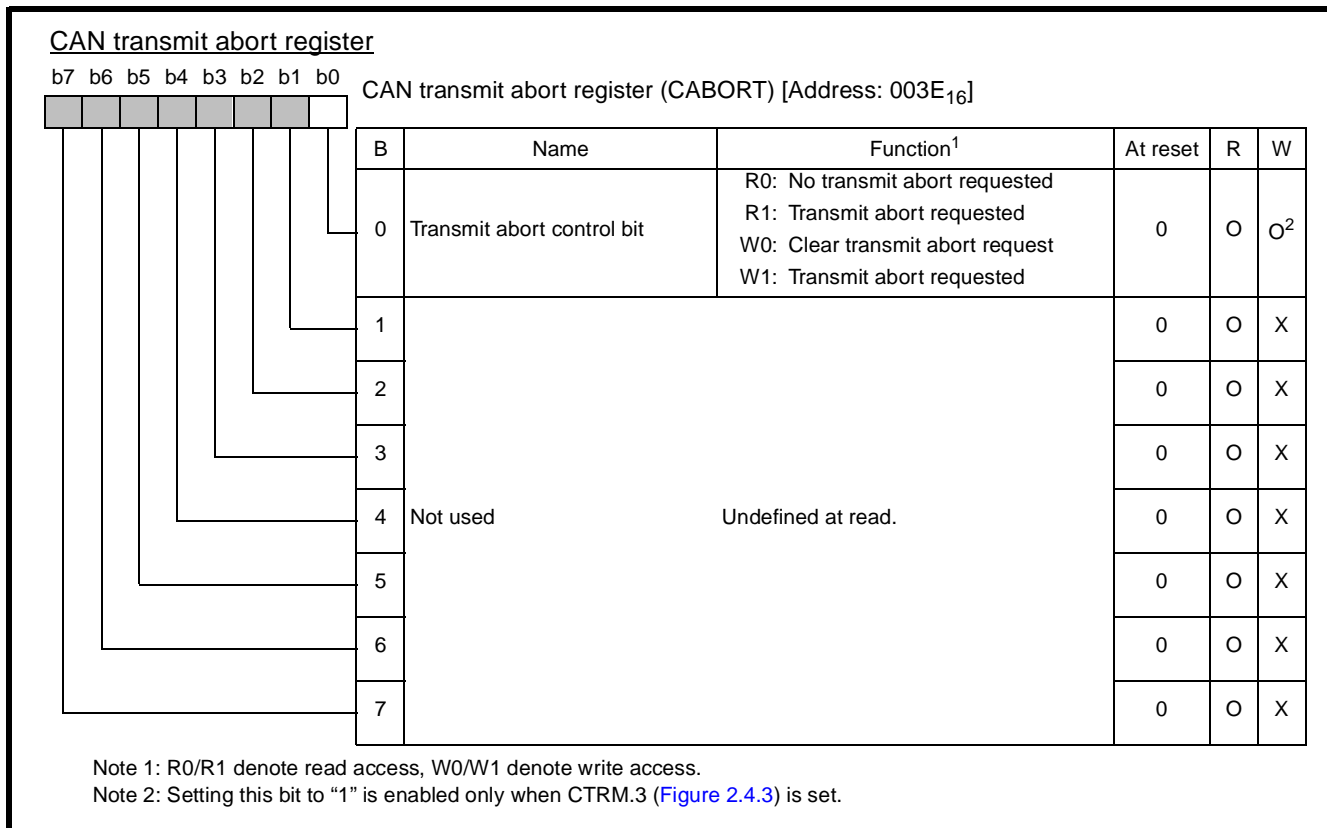


Fig. 2.4.17 Structure of CAN transmit abort register (CABORT)

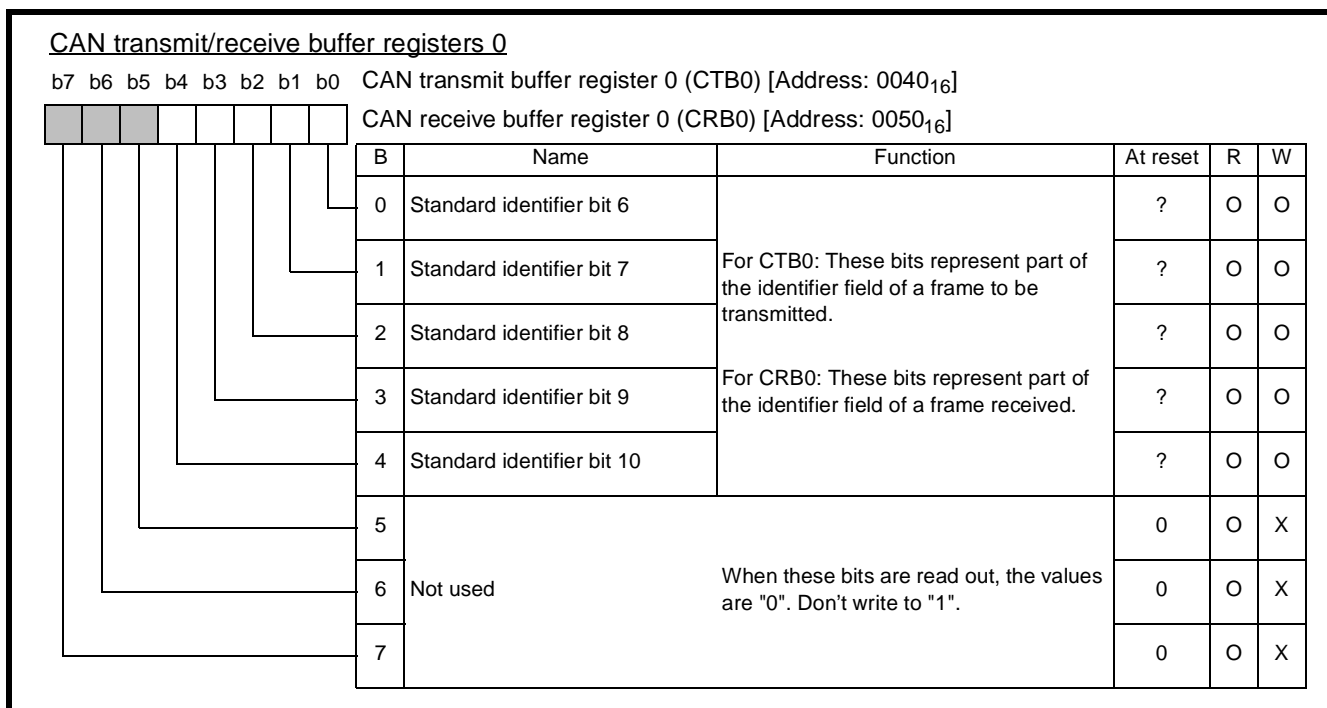


Fig. 2.4.18 Structure of CAN transmit/receive buffer registers 0 (CTB0/CRB0)

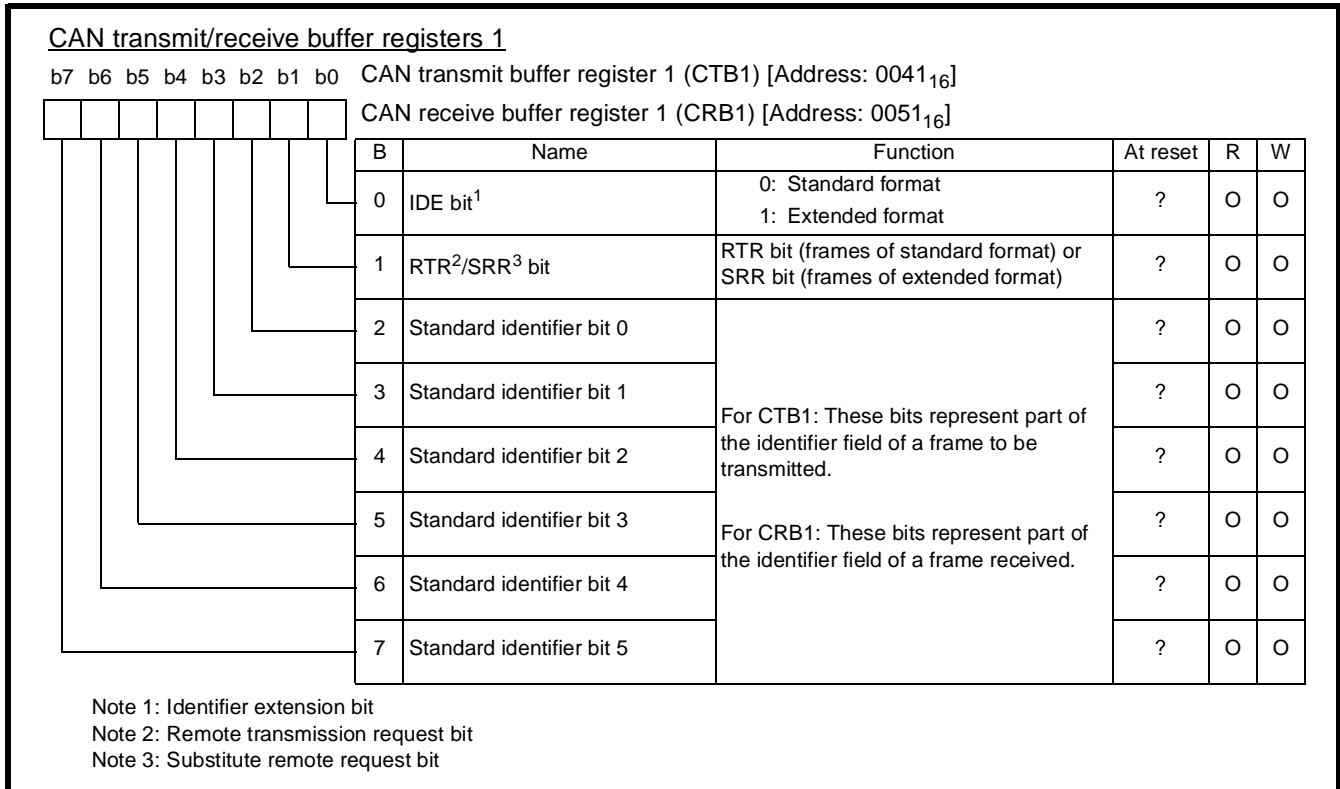


Fig. 2.4.19 Structure of CAN transmit/receive buffer registers 1 (CTB1/CRB1)

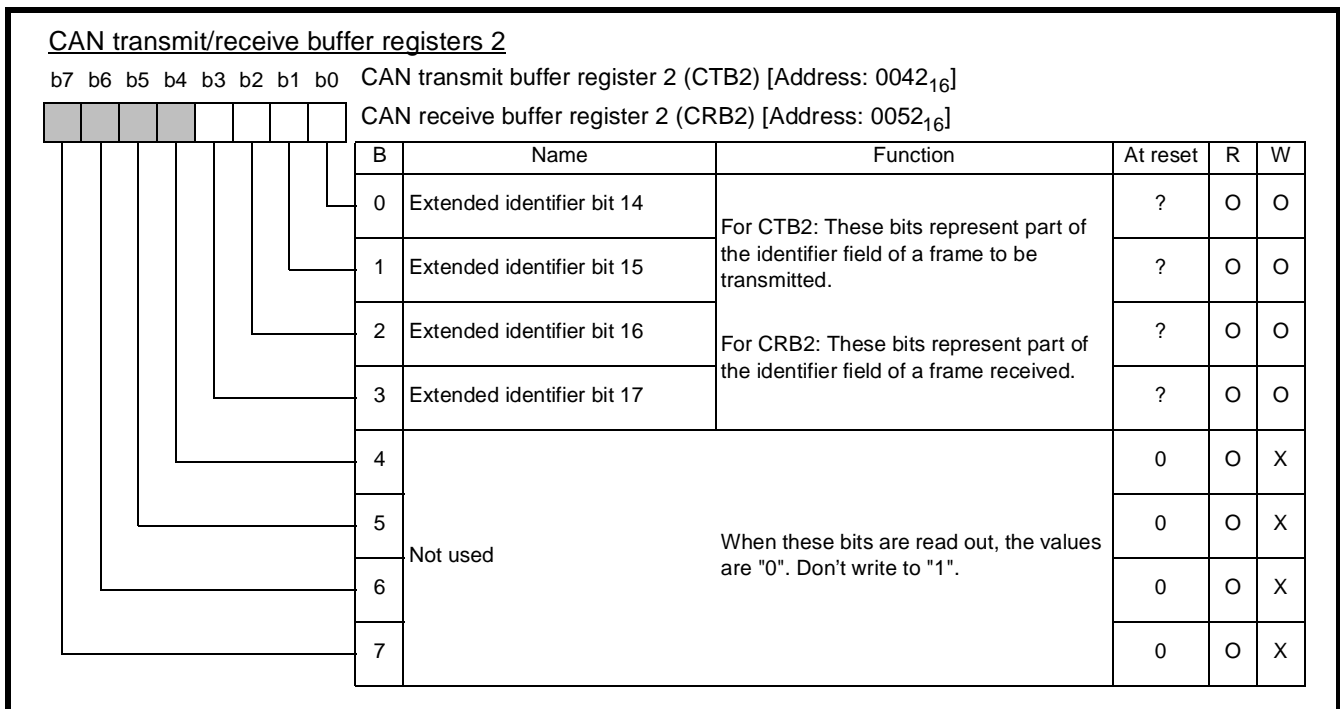


Fig. 2.4.20 Structure of CAN transmit/receive buffer registers 2 (CTB2/CRB2)

APPLICATION

2.4 Controller Area Network (CAN) module

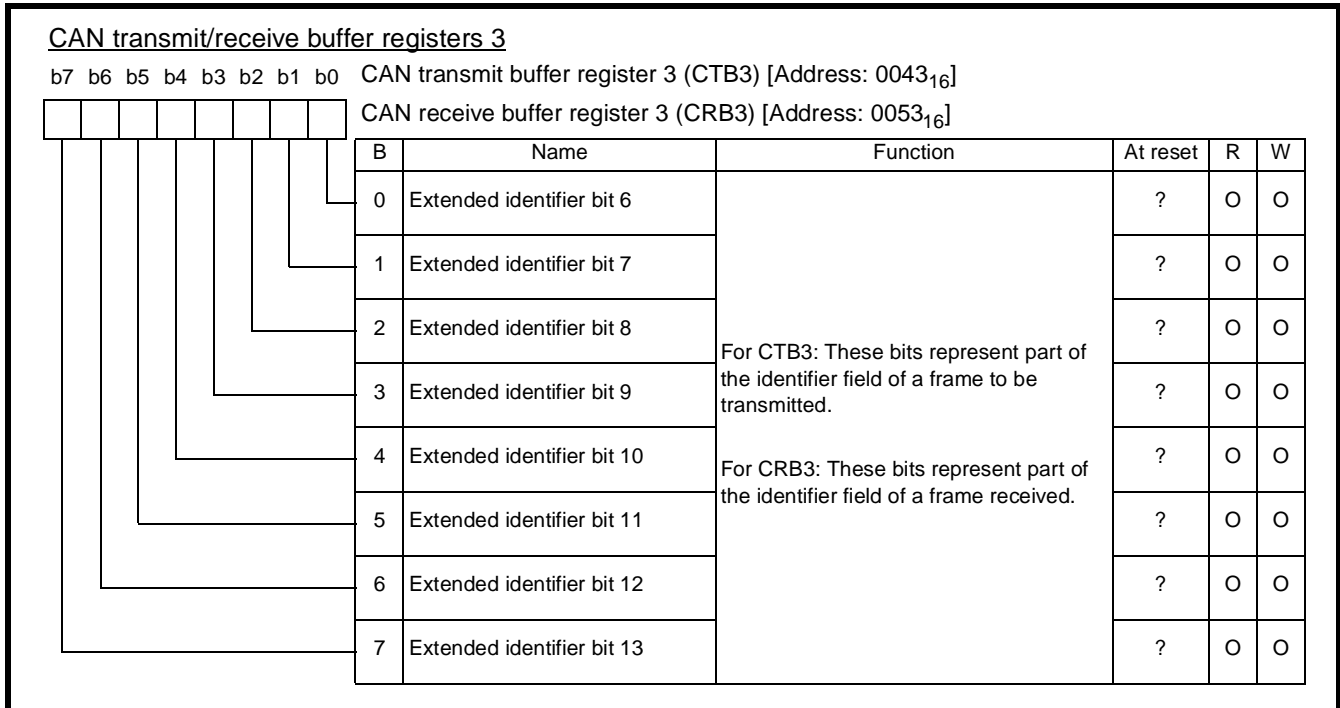


Fig. 2.4.21 Structure of CAN transmit/receive buffer registers 3 (CTB3/CRB3)

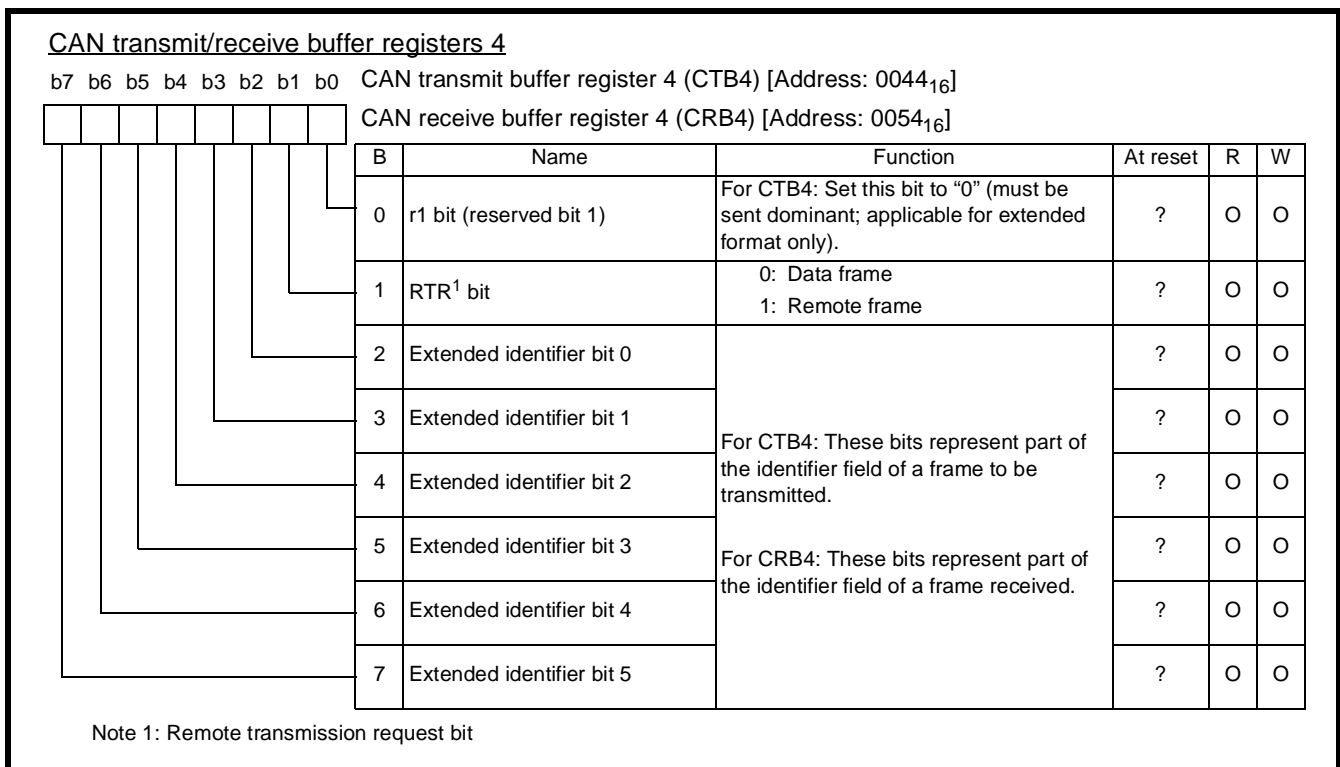


Fig. 2.4.22 Structure of CAN transmit/receive buffer registers 4 (CTB4/CRB4)

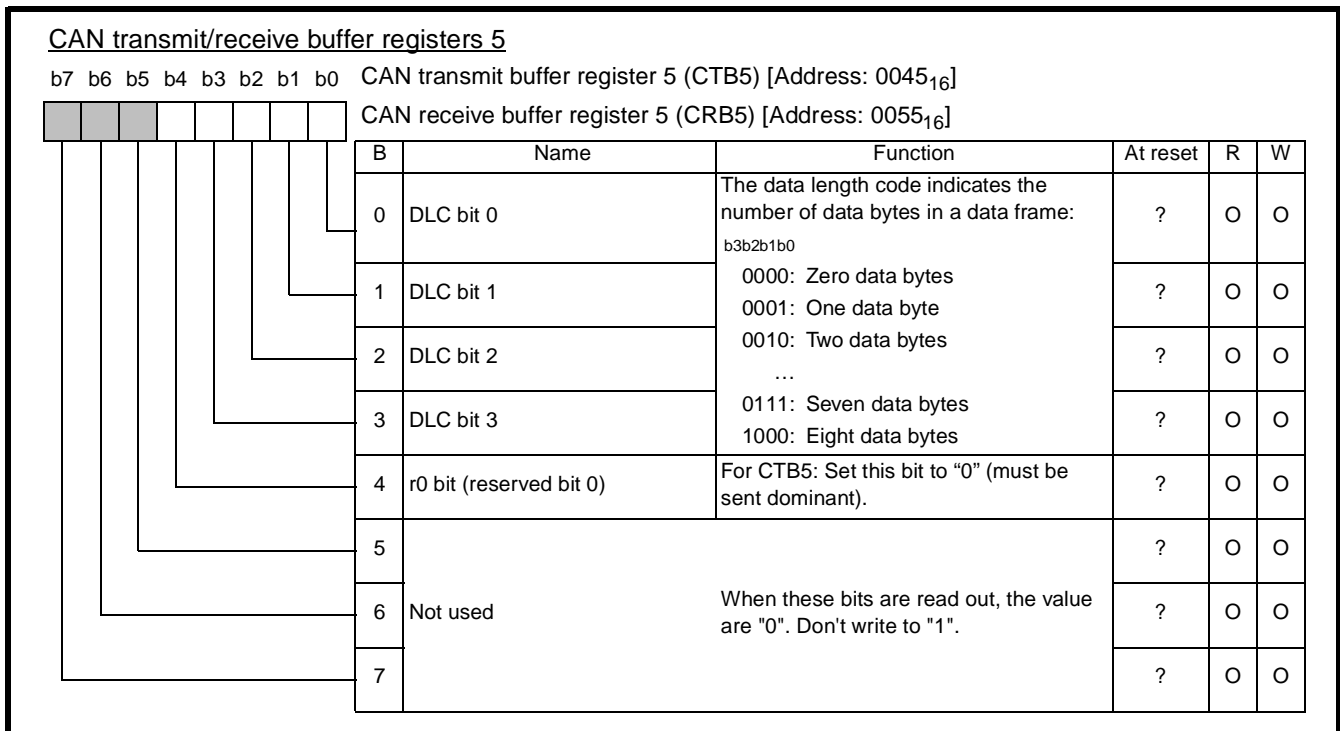


Fig. 2.4.23 Structure of CAN transmit/receive buffer registers 5 (CTB5/CRB5)

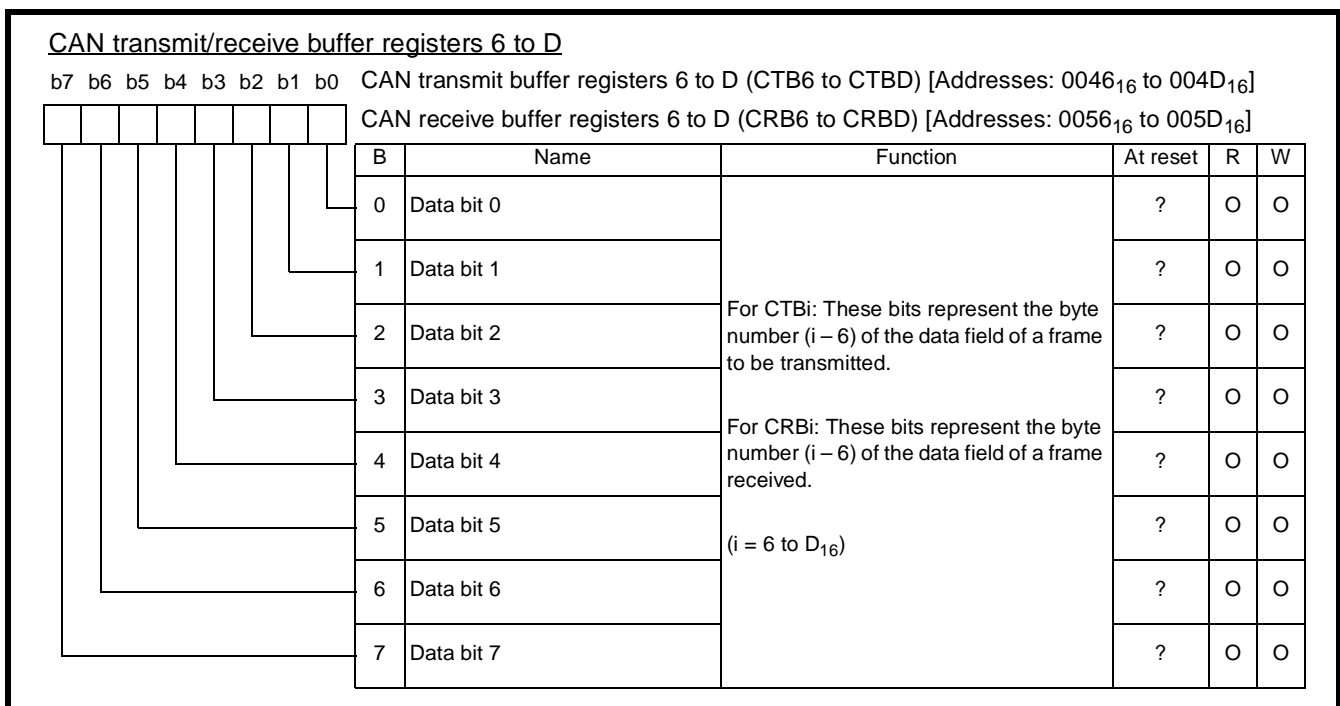


Fig. 2.4.24 Structure of CAN transmit/receive buffer registers 6 to D (CTB6-D/CRB6-D)

APPLICATION

2.4 Controller Area Network (CAN) module

2.4.4 Operational modes

The module features three operational modes which can be selected by the sleep control bit CTRM.0 and the reset/configuration control bit CTRM.1 of the CAN transmit/control register (Figure 2.4.3). Mode transitions may be carried out according to Figure 2.4.25.

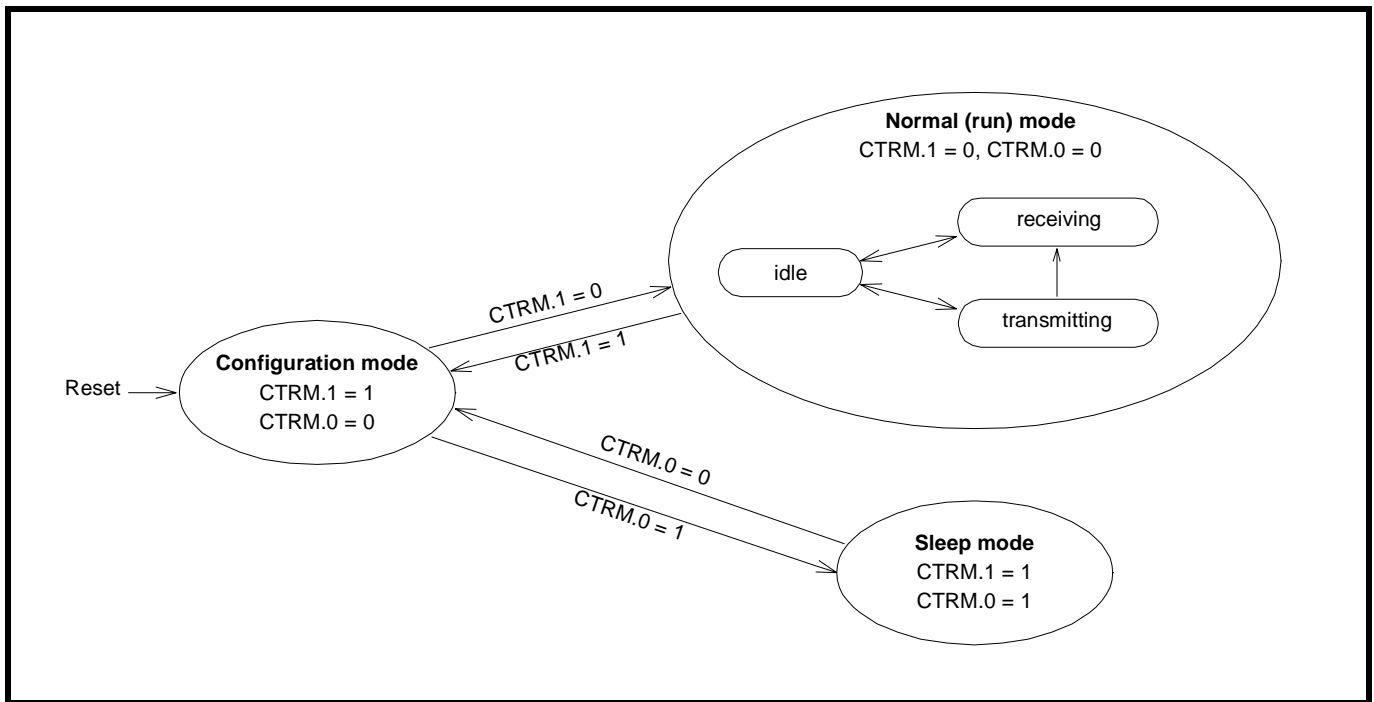


Fig. 2.4.25 Transitions among operational modes

(1) Configuration mode

This mode is used to initialize (refer to section 2.4.5) or reset (refer to section 2.4.6) the module. Entering the configuration mode initiates the following functions by the module:

- Suspend communication functions
- Set P3₁/CTX output to recessive (if P3₁ is configured as CTX output port; see CTRM.2 in Figure 2.4.3)
- Unlock the following configuration register to enable initialization: (1) acceptance code and mask registers (CACi/CAMi, Figures 2.4.6 to 2.4.15), (2) CAN bus timing control registers (CBTCONi, Figures 2.4.4 and 2.4.5)
- Set module to error active state and clear the internal error counters (refer to section 2.4.12)
- Clear transmit request, transmit buffer control and transmit status bits of CTRM (Figure 2.4.3)
- Clear receive buffer control and receive status bits of CREC (Figure 2.4.16)
- Clear transmit abort bit of CABORT (Figure 2.4.17)

The values of the remaining bits of CTRM, CREC and of other CAN module related configuration registers (PCON, CBTCON1, CBTCON2, CAC0 to CAC4, CAM0 to CAM4) retain the values they had before entering the configuration mode.

The contents of the transmit and receive buffer registers CTBi/CRBi (Figures 2.4.19 to 2.4.24) are undefined in configuration mode.

The module is set to configuration mode upon MCU reset.

Note: Switching the module from normal (run) to configuration mode during an ongoing transmission suspends communication immediately; this causes a corrupted frame on the bus. To avoid the corrupt frame, either await the successful transmission (see section 2.4.9) or issue an abort transmission request (see section 2.4.10) before attempting the mode transition.

(2) Normal (run) mode

Entering this mode initiates the following functions by the module:

- Release communication functions; the module becomes an active node on the network and may transmit and receive CAN frames. For details on the transmit and receive operations as well as corresponding interrupt functions see sections 2.4.8 to 2.4.11.

- Lock the following registers to prevent accidental modifications: (1) acceptance code and mask registers (CACi/CAMi, Figures 2.4.6 to 2.4.15), (2) CAN bus timing control registers (CBTCNi, Figures 2.4.4 and 2.4.5)
- Release the internal fault-confinement logic, transmit-, and receive error counters; the module may leave the error active state depending on the error counts (refer to section 2.4.12).

Ensure to initialize the module by using the above mentioned configuration mode before entering normal (run) mode; for details refer to section 2.4.5.

Within normal (run) mode the module can be in three different sub-modes, depending on which type of communication functions are actually performed (see Figure 2.4.26):

- Idle: The module's receive and transmit sections are inactive.
- Receiving: The module is receiving a frame sent by another node.
- Transmitting: The module transmits a frame. Simultaneously, the module may receive its own frame; this is called auto-receive function, for details refer to section 2.4.8, (5).

From CPU side, the sub-modes may be monitored by the receive- and transmit status bits CREC.1 (Figure 2.4.16) and CTRM.7 (Figure 2.4.3).

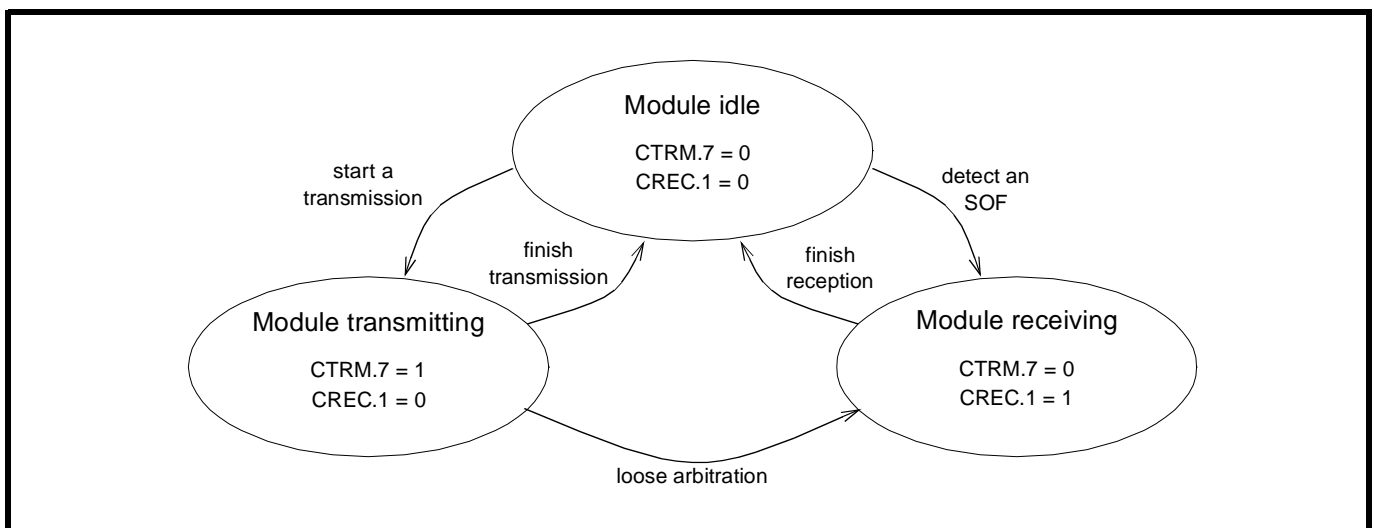


Fig. 2.4.26 Transitions among module sub-modes

(3) Sleep mode

This mode enables reduced power consumption by stopping the clock of the module; consequently all functions (incl. communication) are suspended.

Setting the sleep control bit CTRM.1 (Figure 2.4.3) switches the module to sleep mode. Upon entering this mode, the module's clock supply stops immediately. CAN related registers retain their contents upon entering sleep mode. Enter or leave sleep mode via configuration mode only (refer to Figure 2.4.25).

Warning: Switching the module from normal (run) mode straight to sleep mode (bypassing configuration mode) may cause erroneous frames being sent to the bus or a CTX terminal forcing the bus to dominant level permanently.

APPLICATION

2.4 Controller Area Network (CAN) module

2.4.5 Module initialization

Initializing the module comprises several steps. Before attempting to access the registers involved the following items must be considered:

- **Acceptance filter**

The CAN acceptance code registers CACi (Figures 2.4.6 to 2.4.10) and CAN acceptance mask registers CAMi (Figures 2.4.11 to 2.4.15) need to be initialized. For details on the acceptance filter see section 2.4.7.

- **Bit timing**

The time for the transmission of a single bit consists of four segments:

- Synchronization segment (SS)
- Propagation time segment (PTS)
- Phase buffer segment 1 (PBS1)
- Phase buffer segment 2 (PBS2)

SS is of fixed length (one time-quantum), but the length of PTS, PBS1/2 must be programmed by the bus timing control registers CBTCONi (Figures 2.4.4 and 2.4.5). Figure 2.4.27 shows the segmentation of one bit-time and the possible range for each segment to be programmed.

Note: The CAN specification defines the sum of all time quanta within one bit-time between 8 and 25.

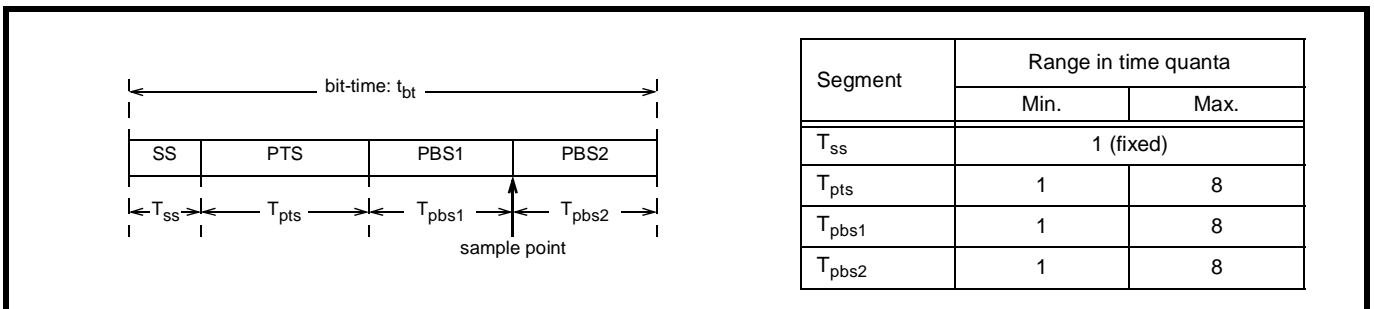


Fig. 2.4.27 Segmentation of bit-time

The sample point is the point within a bit-time where the bus level is known as the value of that respective bit. Its position is between phase buffer segment 1 and phase buffer segment 2. The sample point must be defined in common for all active nodes on the network.

- **Resynchronization jump width**

The resynchronization jump width can be programmed via CBTCON2 (Figure 2.4.5).

Note: The CAN specification defines resynchronization jump width as $\min(4, T_{pbs1})$.

- **Sampling**

The sampling control bit CBTCON1.4 (Figure 2.4.4) allows to decide the bit level based on either a single or three samples. With single sampling, the level is sampled at the defined sample point (refer to Figure 2.4.27). Triple sampling takes two additional samples two and four cycles of $f(XIN)$ before the defined sample point; the bit level is decided on the majority of the three samples. Triple sampling implements a means of digital filtering being appropriate if the bus signal is contaminated by noise.

- **Baud rate**

The module contains a programmable prescaler which is clocked by the MCUs internal clock frequency f_ϕ . This prescaler allows division ratios of 1 to 1/16 (refer to Figure 2.4.4). The baud rate can be calculated as follows, where p is the prescaler division ratio:

$$f_{CAN} = \frac{1}{t_{bt}} = \frac{f_\phi}{p \cdot (1 + T_{pts} + T_{pbs1} + T_{pbs2})}$$

- **Dominant polarity**

The polarity control register PCON (Figure 2.2.9) allows to select the dominant level either “high” or “low”. This setting depends on the transceiver; please refer to the specification of the device/circuit in use.

- **Auto-receive function**

If receiving of CAN frames originating by the module itself are not required, the auto-receive disable bit CREC.6 (Figure 2.4.16) should be set.

- **CAN interrupts**

The module features six interrupts, each of them can be either enabled or disabled by the corresponding control bits of the interrupt control registers ICONA and ICONB; refer to section 2.4.11.

- **Enable CAN transmit pin**

Configure Port P3₁/CTX as CAN transmit output pin by setting the port double function control bit CTRM.2 (Figure 2.4.3) before starting the communication functions of the module.

Example

Figure 2.4.28 shows an example of the initialization sequence required. At first, the module is switched to configuration mode (for details refer to section 2.4.4) to enable altering the special function registers. After that, the related SFRs are initialized with the corresponding parameters (see below). Finally, the module is switched to normal (run) mode to enable the communication and to protect the critical SFRs from being altered accidentally.

The following table shows the conditions being used below:

Item	Setting	Description
Acceptance filtering		Filtering disabled; accept all identifiers.
T _{pts}	1	These settings result in 10 time quanta per bit; the corresponding baudrate is 500kbps at f(XIN) = 10MHz (equivalent to f _φ =5MHz).
T _{pbs1}	4	
T _{pbs2}	4	
Prescaler division ratio	1	
Resynchronisation jump width	4	
Sampling	single	
Dominant polarity	low	
Auto-receive	disabled	
CAN interrupts		not defined here (should be initialized)

These conditions result in CAN bus timing control register values of CBTCON1=00₁₆ and CBTCON2=DB₁₆.

APPLICATION

2.4 Controller Area Network (CAN) module

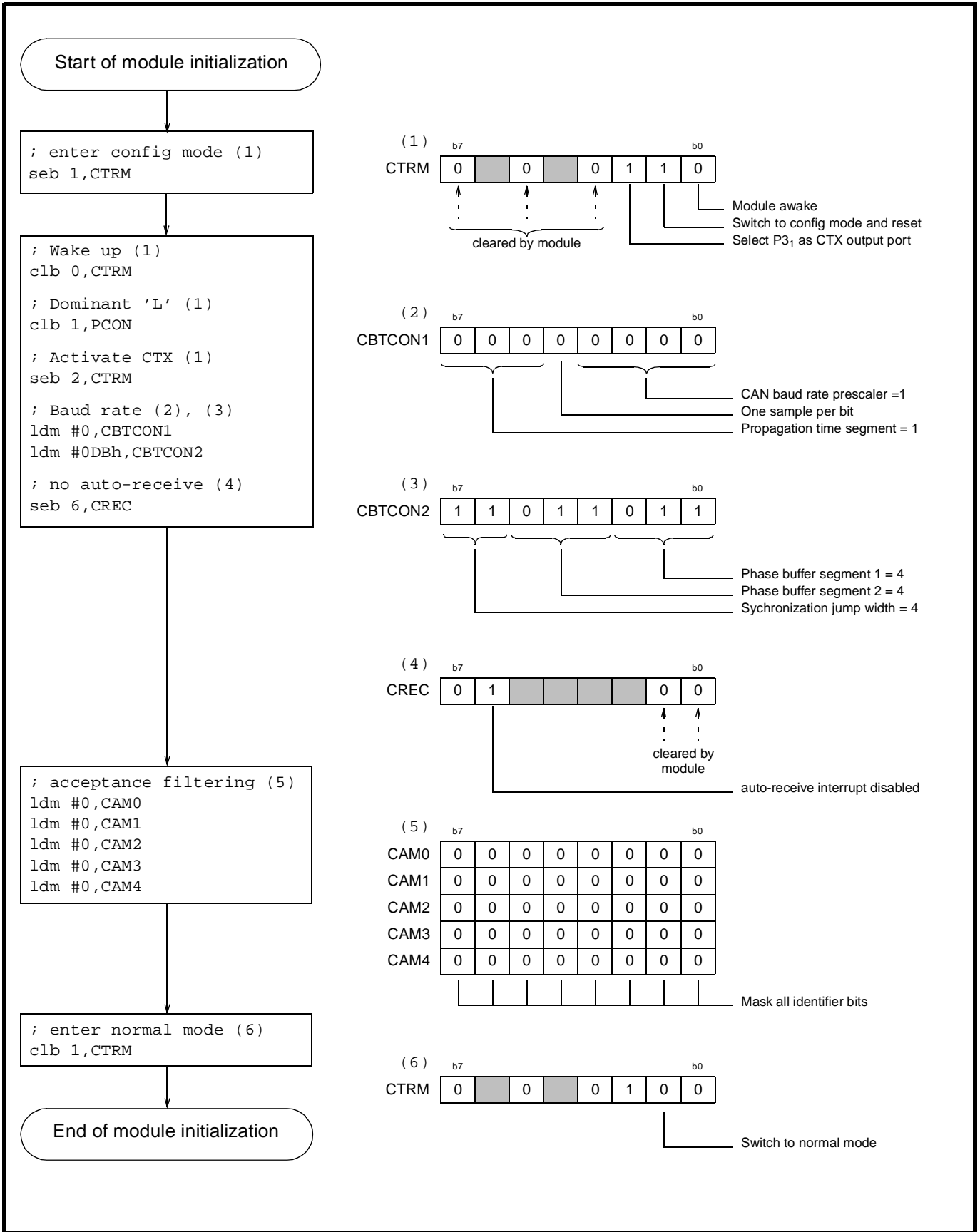


Fig. 2.4.28 Module initialization sequence

2.4.6 Module reset

The sequence shown in [Figure 2.4.29](#) initiates a module reset by switching the module from normal (run) to configuration mode and back to normal (run) mode again. For details on configuration and normal (run) mode refer to section [2.4.4](#).

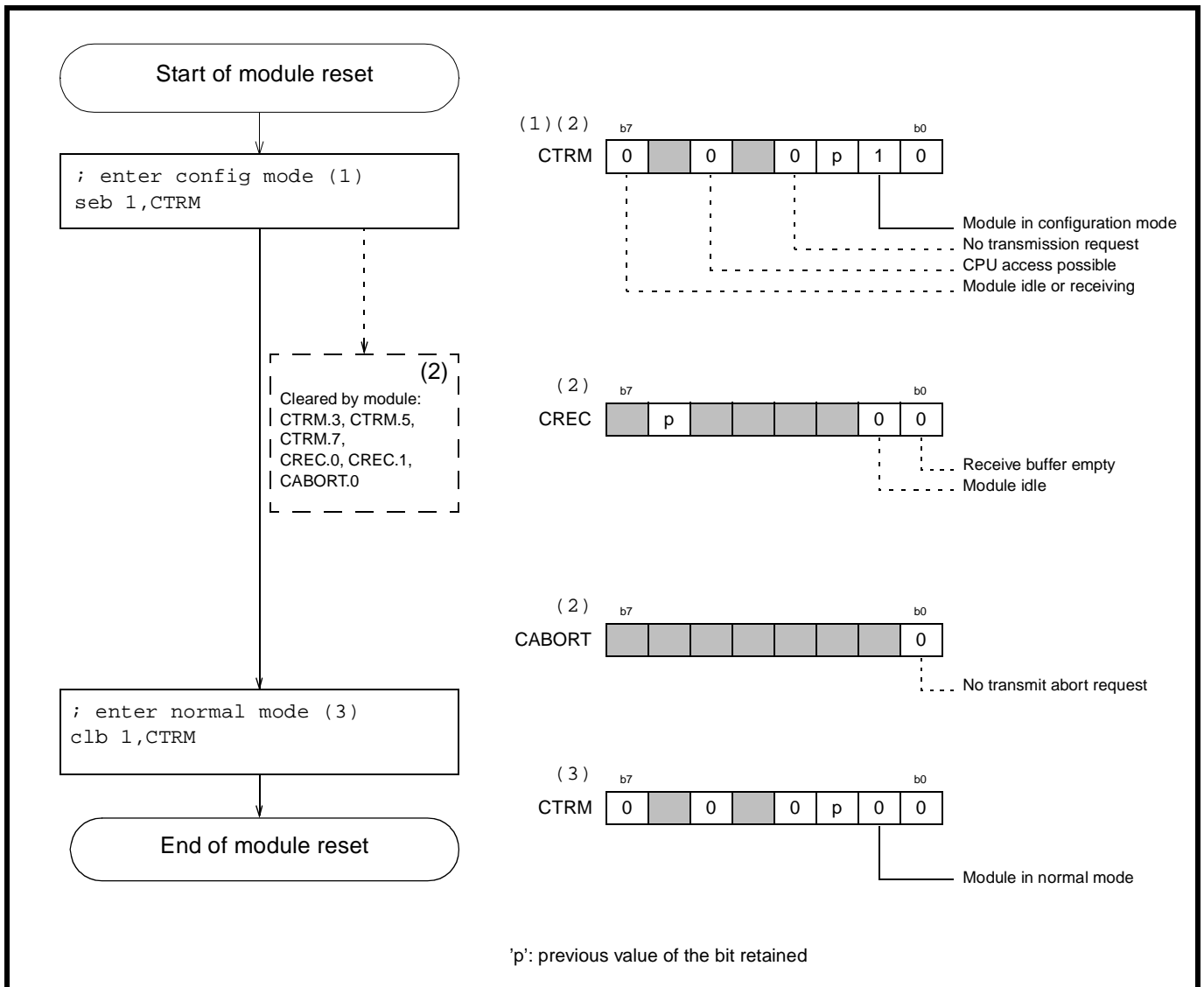


Fig. 2.4.29 Module reset sequence

Note: Do not reset the module starting from sleep mode.

APPLICATION

2.4 Controller Area Network (CAN) module

2.4.7 Acceptance filtering

The module contains a hardware filtering circuit to screen out the useless messages out of the message stream and thereby reduce CPU load. This filter probes the identifier field of any frame on the bus and decides which frames are relevant to the given node and which may be abandoned.

(1) Register structure

The hardware implements a single condition identifier filter by a set of acceptance code CAC_i (Figures 2.4.6 to 2.4.10) and acceptance mask CAM_i (Figures 2.4.11 to 2.4.15) registers. These registers cover the entire 29-bit identifier scale (extended format); however 11-bit identifiers (standard format) can be handled as well. The registers (shown in Figure 2.4.30) can be modified in configuration mode (refer to section 2.4.5) only.

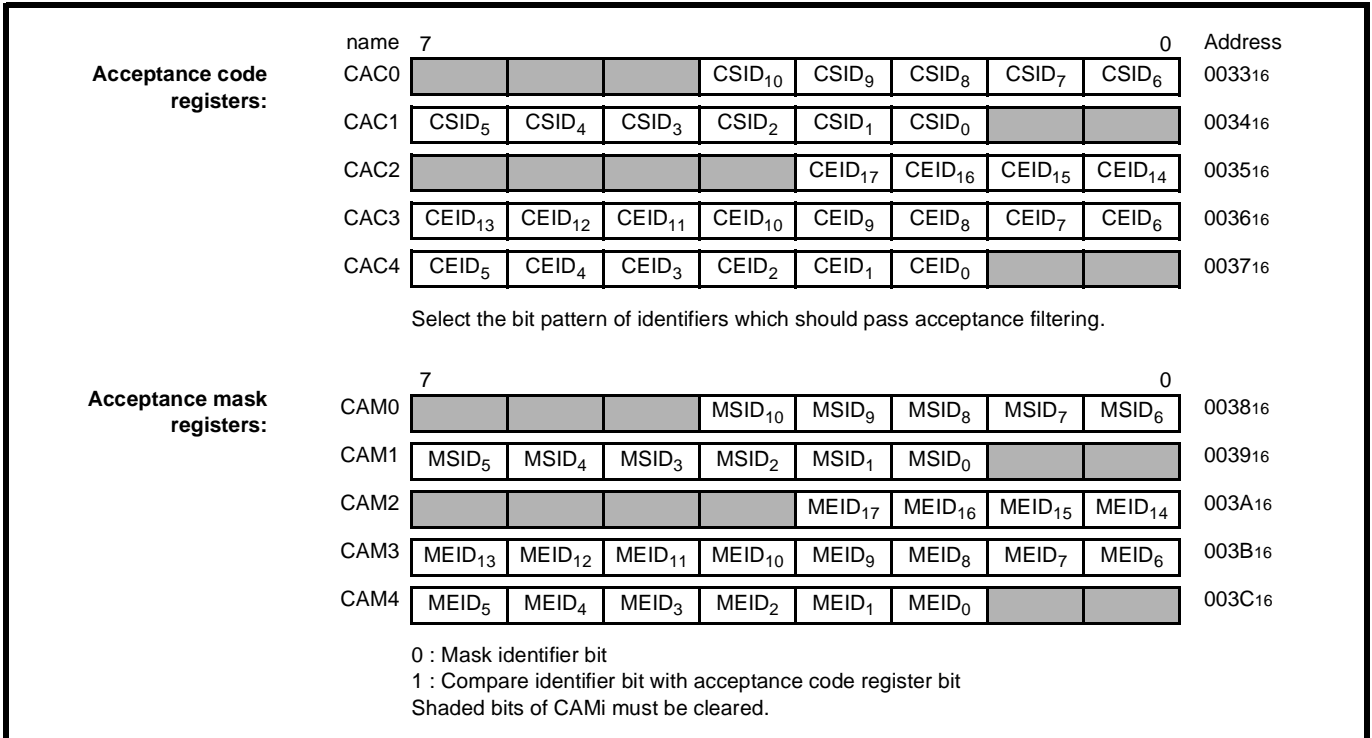


Fig. 2.4.30 Structure of acceptance mask/code registers

(2) Operation

Acceptance filtering starts after detecting the start-of-frame of a CAN message. The content of the acceptance mask registers define which identifier bits have to be subjected to comparison with the corresponding bits of the acceptance code registers.

If the acceptance filter judges an incoming frame relevant, the frame is—depending on the availability of a receive buffer—either stored in a receive buffer or a CAN overrun interrupt (COVR) is issued by setting the corresponding interrupt request bit. For further details on the receive buffer system and the overrun interrupt refer to sections 2.4.8 and 2.4.11.

(3) Schematic of acceptance filter

The acceptance filter mechanism (see Figure 2.4.31) comprises one gate which compares the acceptance code register bit with the corresponding identifier bit of the frame being received, and one gate which tests the relevance of this bit for the acceptance filter process. When all acceptance bits are true, the module rates the frame relevant and attempts to store it to a receive buffer.

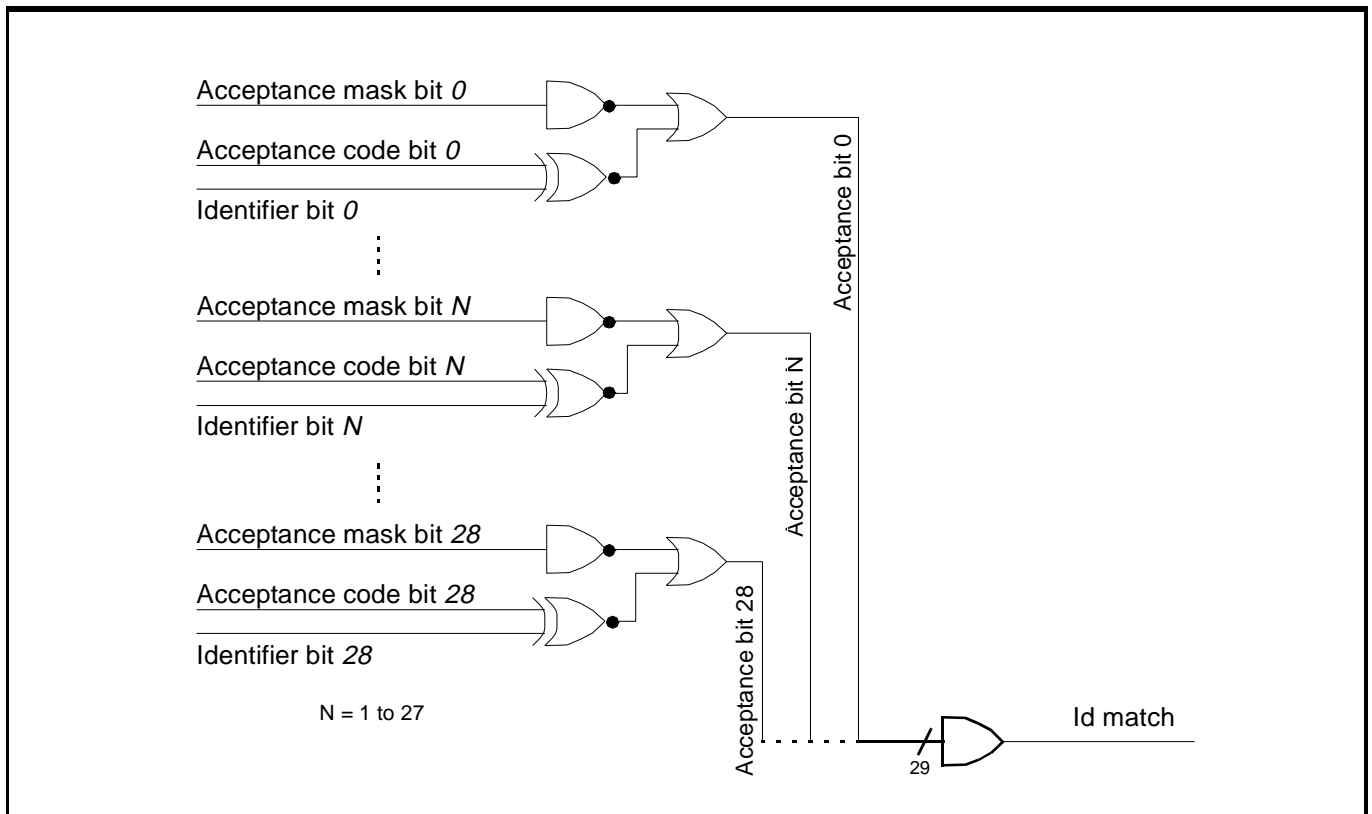


Fig. 2.4.31 Acceptance filter logic

2.4.8 Message reception

The module is equipped with two physical receive buffers. This double buffer architecture allows simultaneous CPU-processing of a previously received frame and reception/storage of a new frame by the module. To simplify the software handling both buffers are memory mapped to the same address range (0050₁₆ to 005D₁₆). The module's internal receive logic controls access to the buffers so that one buffer (later on called *foreground buffer*) can be read out by the CPU while the other buffer (*background buffer*) can be written to by the module.

The user is able to switch back and forth between both buffers by one control bit; however the individual buffers can not be addressed directly. This architecture allows sequential handling of the incoming frames on a one-by-one basis.

The receive buffers support both standard and extended frame formats of data and remote type. The buffers hold the following information on the received frame:

- Identifier (standard or extended),
- Frame type (data frame or remote frame),
- Data length (in case of a data frame), and
- Data bytes (in case of a data frame).

(1) Message storage

As explained above, the module features two receive buffers; after initialization both buffers are empty (undefined). Upon successful reception of the first relevant (refer to section 2.4.7) frame the module will:

1. Store the frame in one of the buffers.
2. Switch the buffer to foreground to permit CPU access.
3. Flag the reception by special function bits and interrupt service request (details in section (2)).

The foreground buffer holding the message received is now under (exclusive) control of the CPU, while the background buffer is still vacant.

After having processed the buffer contents, the CPU should release the buffer and thereby return buffer control to the module. Following to release, the buffer content is not available to the CPU any more and can not be recaptured in any way.

APPLICATION

2.4 Controller Area Network (CAN) module

A second message can be stored to the background buffer while the CPU still deals with the foreground buffer; for an example see [Figure 2.4.34](#): while the CPU still processes frame C, the module simultaneously receives frame D.

The control of the buffers is done via the receive buffer control bit CREC.0 ([Figure 2.4.16](#)) as shown in [Figure 2.4.32](#). In normal (run) mode, CREC.0 can be cleared by the CPU but can only be set by the CAN module.

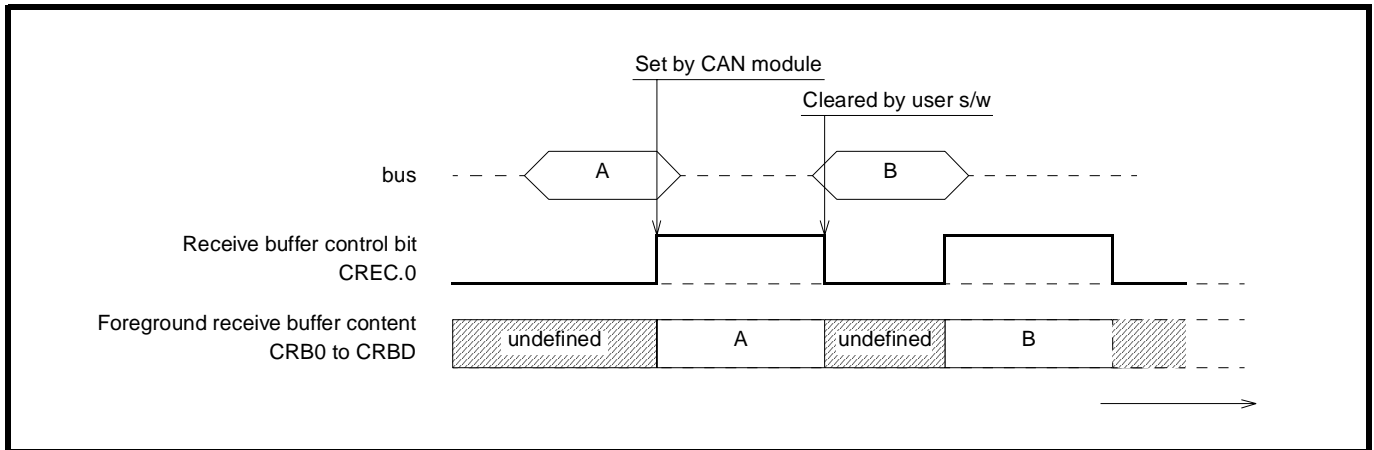


Fig. 2.4.32 Receive buffer handling

(2) Receive process

The receive sequence is initialized by the start-of-frame of a new incoming message. The user software can check this status by the transmit status bit CTRM.7 and the receive status bit CREC.1. The module status changes from idle/transmitting to receiving.

During the reception of a message, the receive status bit CREC.1 is kept high. It is cleared after the end of frame (for details refer to section (3) below). If the message is accepted and received without any errors, the module initiates the following actions:

1. Set CAN successful receive (CSR) interrupt request bit IREQA.4.
2. Enable CPU access to the buffer (i.e., switch it from background to foreground).
3. Set receive buffer control bit CREC.0.

If the CPU still processes a previously received frame in the foreground buffer (and has not released the buffer yet), the actions 2 and 3 are postponed. In this case, the module will continuously monitor CREC.0 and wait for the flag being cleared by user s/w. After that, the module will execute the actions 2 and 3.

The frame reception can be observed from CPU side either by polling of IREQA.4 or CREC.0 or by CSR interrupt service. After having processed the buffer contents, the receive buffer control bit should be cleared by the CPU, in order to give control of the buffer back to the module. This kind of handshaking enables the module to use the buffer to store the next frame to be received; consequently data read from the receive buffer address range is undefined until the next frame reception.

Note: If the reception function is implemented based on CSR interrupt service, reconfirm the CREC.0 status after clearing it. If both buffers are already occupied before entering the CSR interrupt service routine, *both* buffers must be processed in *one* CSR interrupt service. However, it is recommended to optimize the timing of the interrupt system for minimum latency and short execution time to avoid the possibility of CAN overrun situations; see section (4). If the requirements to avoid overrun situations are fulfilled, one CSR interrupt service has to deal with one single frame only (because each frame reception triggers one CSR interrupt service request).

[Figure 2.4.33](#) shows the receive process flowcharts from CAN module and CPU side. Section (3) discusses the timing details on the module's receive processing.

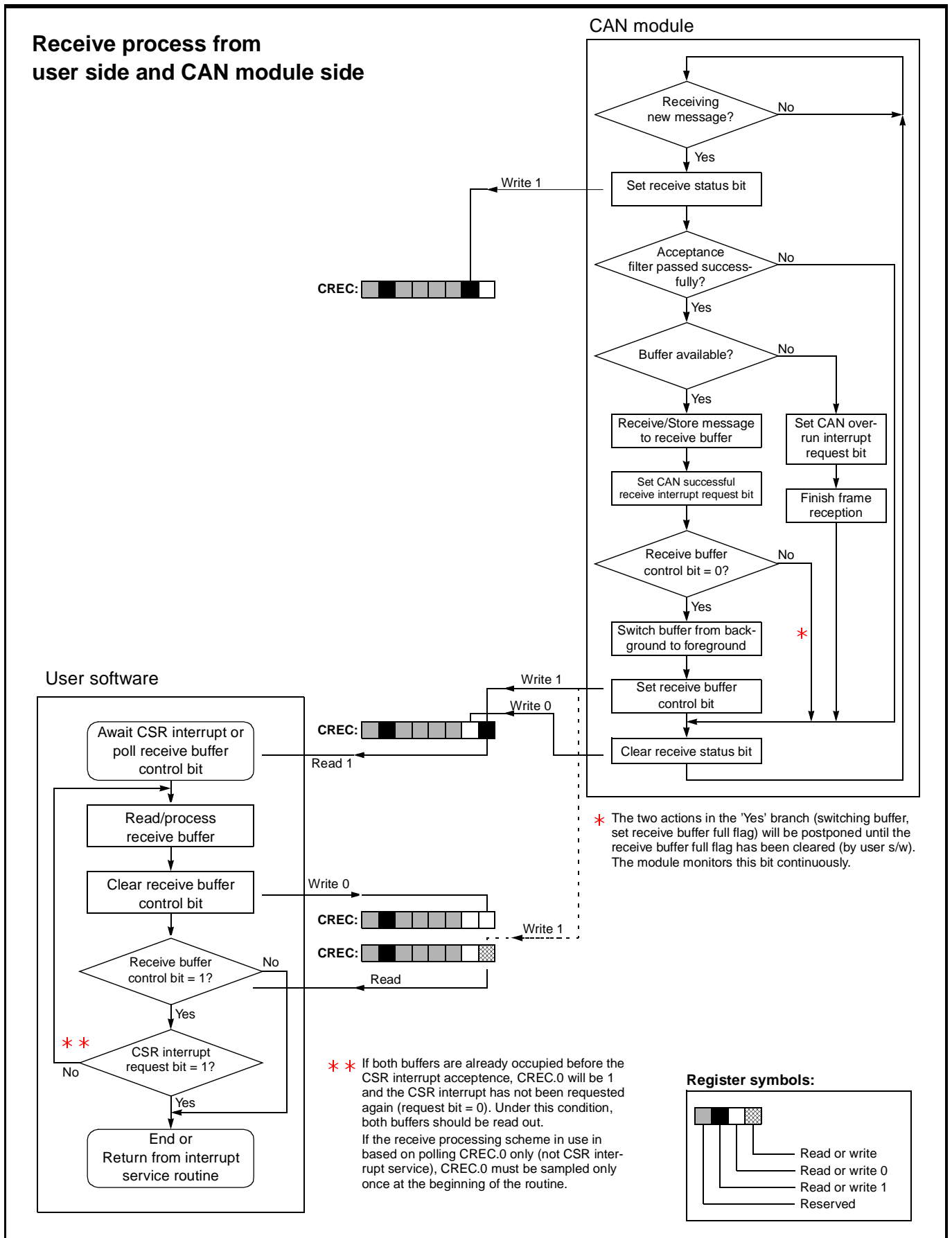


Fig. 2.4.33 Flowchart of the receive process

APPLICATION

2.4 Controller Area Network (CAN) module

(3) Timing of receive sequence

The timing diagrams in Figure 2.4.34 shows the status of the internal special function bits and the contents of the foreground receive buffer during receive sequence. Of course this timing diagram depends on the application software and actual communication model (i.e., scheduling of messages on the bus); therefore the diagram should be considered an example. The time between the occurrence of the receive interrupt request and the clearing of this request bit depends on the interrupt system of the actual application. The latency of the interrupt depends on the execution time of all interrupt service routines of the application unless interrupt nesting is enabled.

The conditions used in the timing diagram are:

- Module in virgin condition (as after module reset)
- Acceptance filtering disabled (mask all identifier bits)
- Buffer content processing by CAN successful receive (CSR) interrupt service
- Execution time of CSR service routine shorter than the frames on the bus
- Interrupt system of the application optimised for minimum latency of interrupts

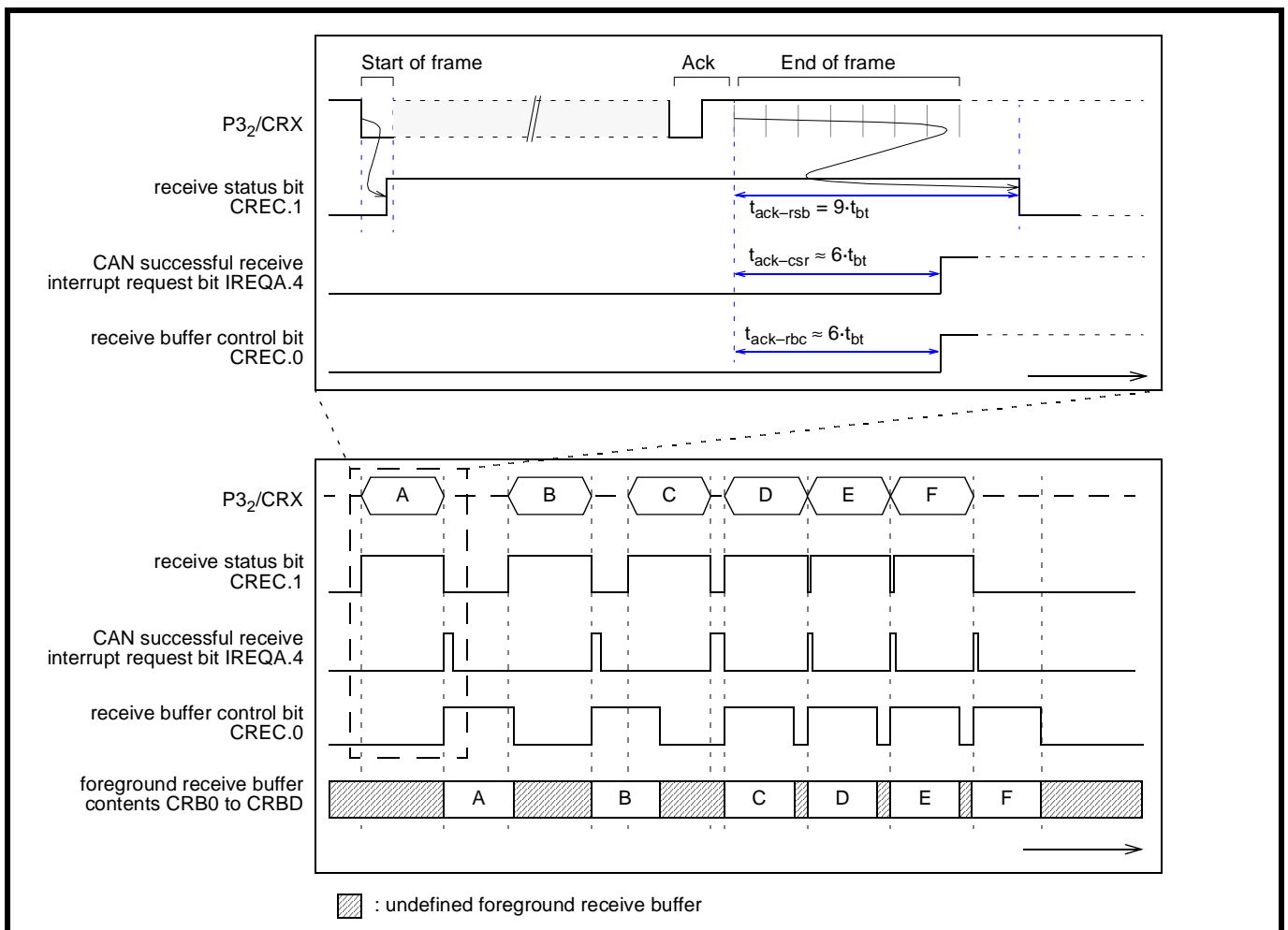


Fig. 2.4.34 Receive sequence timing

During a successful reception the module alters the special function bits CREC.1, IREQA.4 and CREC.0 according to the following sequence:

1. Set CREC.1 within the start-of-frame bit.
2. Set IREQA.4 and CREC.0 within the seventh bit-time of the end of frame field.
3. Clear CREC.1 after the second bit-time of the intermission field.

(4) Avoiding CAN overrun interrupts

The CAN overrun interrupt (COVR) is requested upon reception of a relevant frame when both receive buffers are preoccupied by previously received frames because the CPU has not released a buffer yet. The frame causing the CAN overrun interrupt can not be stored and hence is lost.

The timing diagram in [Figure 2.4.35](#) shows one example for a condition leading to a COVR request. The conditions used in the timing diagram are:

- Module in virgin condition (as after module reset)
- Acceptance filtering disabled (mask all identifier bits)
- Buffer content processing by CAN successful receive (CSR) interrupt service
- Execution time of the CSR interrupt service routine *longer* than the frames on the bus
- Interrupt system of the application optimised for minimum latency of interrupts

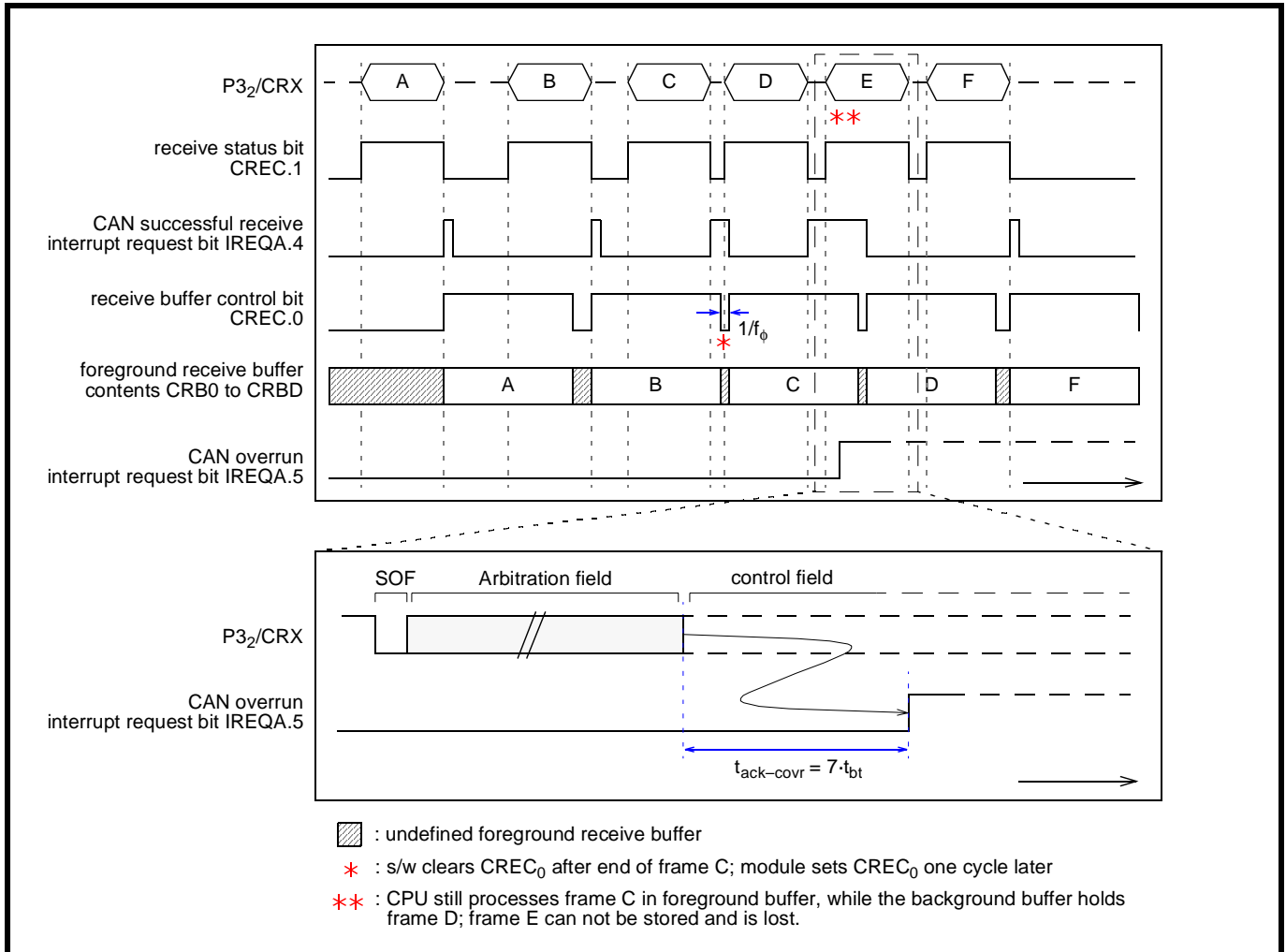


Fig. 2.4.35 Receive sequence timing (overrun condition)

To avoid an overrun condition, the time between a successful reception and the release of receive buffer by the CPU must be shorter than the shortest possible frame in the network. This can be accomplished by optimization of the applications interrupt system towards minimum latency and interrupt execution times.

(5) Auto-receive function

The auto-receive disable bit CREC.6 ([Figure 2.4.16](#)) allows to select two options regarding the handling of frames sent by the module:

- Enabling auto-receive by clearing CREC.6 causes the module to subject a self-generated frame to the reception process described above; however, the following deviations apply:
 - Self-generated frames are not self-acknowledged.
 - CREC.1 remains “0” during the auto-receive process. The corresponding sub-mode according to [Figure 2.4.26](#) is *transmitting*.

A frame rated relevant by the acceptance filtering causes a successful receive interrupt request and possibly an overrun interrupt request (see section (4)).

APPLICATION

2.4 Controller Area Network (CAN) module

- Disabling auto-receive by setting CREC.6 causes the module to suspend all receive functions related to the *receive control logic* block (Figure 2.4.1) during the transmission process. The transmitting/receiving sub-mode of the normal mode shown in Figure 2.4.25 is disabled.

Note: Disabling auto-receive does not affect the receive function related to the *protocol controller* block (required to monitor the bus level while transmitting).

2.4.9 Message transmission

The module is equipped with one transmit buffer. Similar to the receive section, the architecture allows to transmit CAN frames on a one-by-one basis. Both standard and extended frame formats of data and remote type are supported. The programming sequence required to initiate a transmission comprises four steps, each step shall be described in detail below:

1. Check the module status (i.e., the availability of the transmit buffer).
2. Initialize the transmit buffer (the buffer content defines the frame to be transmitted).
3. Lock the transmit buffer (hands the buffer control to the module and protects the buffer from accidental modifications by the CPU).
4. Issue the transmit request (triggers the module to start the transmission).

Once the transmission of a frame has been requested, the module takes care about bus arbitration, error handling and acknowledgement of the frame by other nodes. The frame is considered transmitted successfully if:

- The frame could win arbitration,
- no errors were detected during transmission, and
- the frame gained acknowledgement by another node on the bus.

In case of unsuccessful transmissions the module attempts to re-transmit the frame until transmission can be finished successfully or the transmit request is withdrawn by user software (refer to section 2.4.10). In case of an arbitration loss, the module transits to sub-mode *receiving* (refer to section 2.4.8 and Figure 2.4.26).

The programming sequence and functionality is explained below and by the flow charts in Figure 2.4.38.

(1) Check for availability of the transmit buffer

Before the transmit buffer can be initialized, its availability must be checked; the transmit buffer control bit CTRM.5 must be “0” indicating the availability of the buffer and completion of the previous transmission. The user software can now initialize the transmit buffer.

(2) Initialize the transmit buffer

In the user software the identifier (standard or extended), the frame type (data or remote frame), length code/data bytes (in case of a data frame) have to be written to the transmit buffer (Figures 2.4.18 to 2.4.24). Only the transmit buffer registers relevant for the specific CAN frame need to be initialized (e.g. if only standard CAN frames are concerned, the transmit buffer registers CTB2 and CTB3 can be ignored).

If the transmit buffer has been used for a previous transmission, the contents of the buffer are retained; therefore only the transmit buffer contents which need to be changed versus the previous frame have to be initialized.

Figure 2.4.36 shows the transmit buffer register organisation.

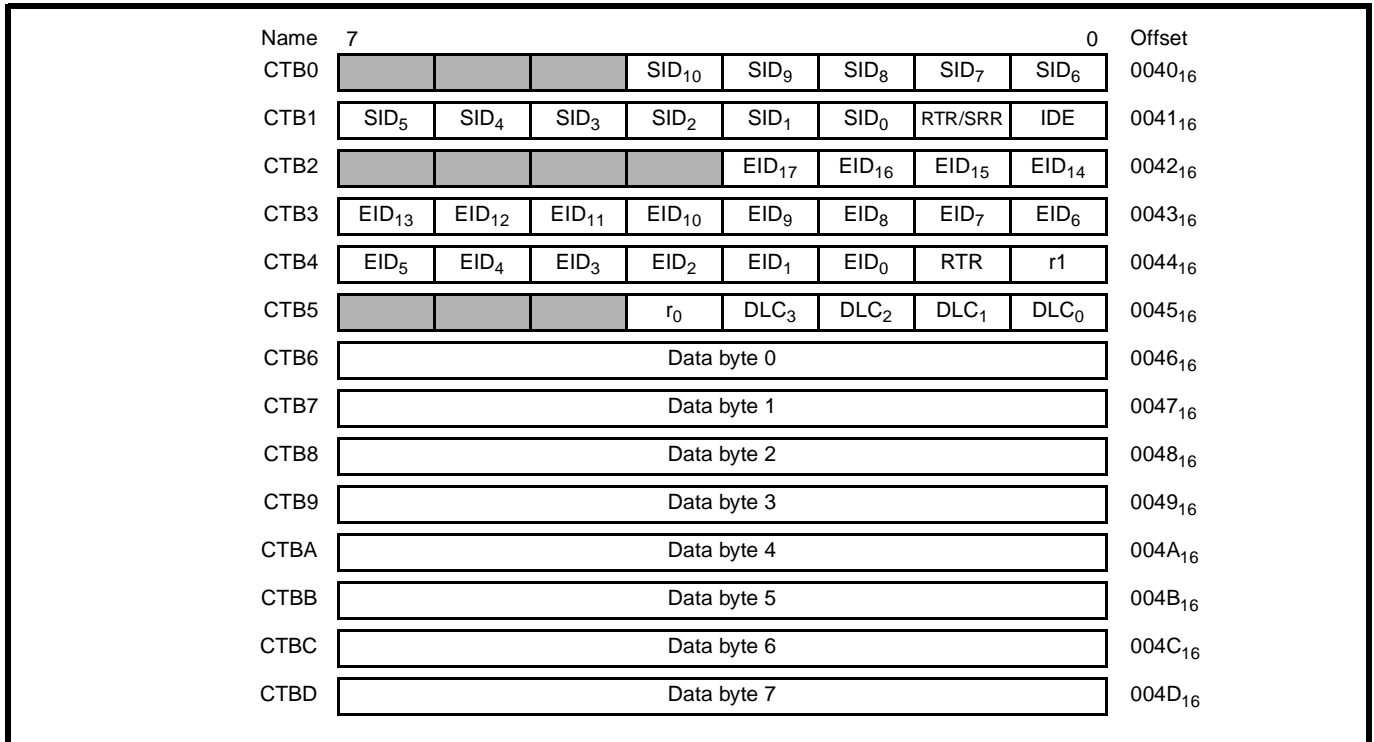


Fig. 2.4.36 Transmit buffer organization

Altogether the following items of the transmit buffer need to be initialized:

- IDE bit CBT1.0: this bit represents the identifier extension (IDE) bit of a frame.
- RTR/SRR bit CBT1.1: for standard format frames, this bit represents the *remote transmission request* (RTR) bit of the arbitration field; for extended format frames it represents the *substitute remote request* (SRR) bit of the arbitration field.
- RTR bit CBT4.1: this bit represents the *remote transmission request* (RTR) bit of the arbitration field of an extended format frame; not relevant for standard format frames.
- Reserved bit r0 CBT5.4: this bit represents the reserved bit r0 of the control field of a standard or extended format frame; should be set to "0"/sent dominant (refer to the CAN specification).
- Reserved bit r1 CBT4.0: this bit represents the reserved bit r0 of the control field of an extended format frame; not relevant for standard format frames; should be set to "0"/sent dominant (refer to the CAN specification).
- Identifier bits of CBT0 to CBT4: these bits represent the identifier of the arbitration field of a frame; CBT2 to CBT4 hold the extended identifier bits and are not relevant for standard format frames.
- DLC bits of CBT5: these bits represent the data length code of the data field of a data frame; not relevant for remote frames.
- Data bits of CBT6 to CBTD: these bits represent the data field of a data frame; not relevant for remote frames.

For settings of the IDE-, RTR/SRR-, RTR-, r0- and r1 bits to be programmed refer to the table below:

Frame format	Frame type	IDE	RTR/SRR	RTR	r0	r1
standard format (11-bit identifier)	data frame	0	0	X	0	X
	remote frame	0	1	X		
extended format (29-bit identifier)	data frame	1	1	0	0	0
	remote frame	1	1	1		

(3) Lock the transmit buffer

Once the transmit buffer is initialized, the transmit buffer control bit of CTRM (Figure 2.4.3) should be set to "1". This locks the transmit buffer thereby protects it from being altered accidentally. After locking, data read from the address range of the transmit buffer is undefined. The buffer remains locked until either the trans-

APPLICATION

2.4 Controller Area Network (CAN) module

mission process could be finished successfully, or a transmit abort is requested (see section 2.4.10); the buffer can not be unlocked by clearing the transmit buffer control bit from CPU side.

(4) Issue transmission request

Finally, the transmission can be started by setting the transmission control bit of CTRM (Figure 2.4.3). Setting this bit gives control of the buffer to the CAN module; the module attempts to transmit the frame defined by the transmit buffer contents following the rules of the CAN specification.

After a successful transmission (without errors) the module will:

- Clear the transmit buffer control bit CTRM.5.
- Clear the transmit request bit CTRM.3.
- Set the CAN successful transmit (CST) interrupt request bit IREQA.3.

Note: The CAN successful receive interrupt (CSR) interrupt may also be requested unless disabled by the auto-receive interrupt disable bit of CREC (Figure 2.4.16); refer to section 2.4.8, • too.

(5) Timing of transmit sequence

The timing diagram in Figure 2.4.37 shows the status of the internal special function bits during transmit sequence if the module wins bus arbitration.

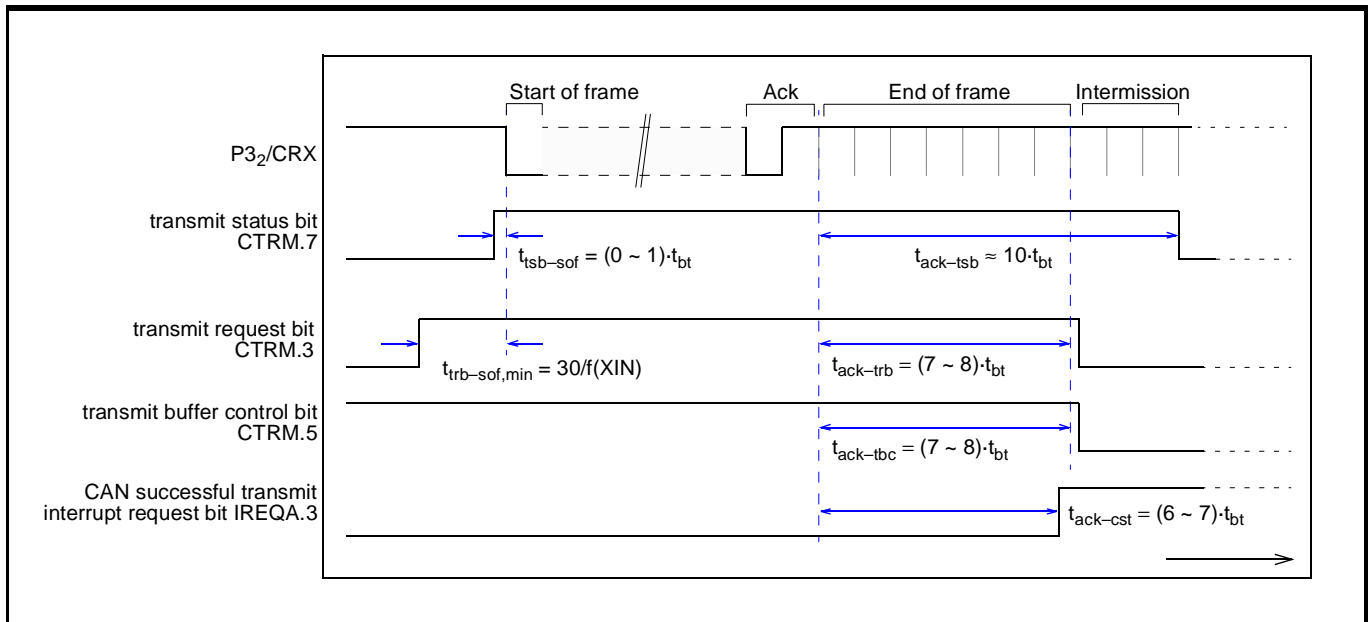


Fig. 2.4.37 Transmit sequence timing (arbitration win)

After a transmission has been requested by setting CTRM.3, the module attempts to start the transmission at the next possible time (depending on the bus condition). During a successful transmission process the module alters the special function bits as follows:

1. Set transmit status bit CTRM.7 shortly before the recessive to dominant edge of the start-of-frame field.
2. Set CAN successful transmit interrupt request bit IREQA.3 within the last bit-time of the end-of-frame field.
3. Clear transmit request bit CTRM.3 and transmit buffer control bit CTRM.5 within the first bit-time after the end-of-frame field.
4. Clear transmit status bit CTRM.7 within the first bit-time after the intermission field.

Upon arbitration loss, the module changes to receiving state beginning with the next bit-time after the occurrence of arbitration loss. The transition from transmitting to receiving state is flagged by clearing of CTRM.7 and setting of CREC.1 (see Figure 2.4.26).

2.4.10 Abort transmission

A low priority frame in the transmit buffer may not gain bus access if the bus carries heavy traffic by medium priority frames. The low priority frame blocks the transmit buffer and causes significant delay in scheduling of further even high priority frames to be sent. This scenario is known as *priority inversion*.

To overcome this situation, the module features an abort transmission request function. This function is controlled by the transmit abort control bit CABORT.0 ([Figure 2.4.17](#)); setting this bit withdraws the transmit request of the frame currently occupying the transmit buffer.

Requesting transmit abort during the transmission process (CTRM.7 = 1) does not interrupt the process to avoid causing erroneous frames being sent.

In result of the abort transmit request, the module will:

1. Clear the transmit buffer control bit CTRM.5 (thereby release the buffer).
2. Clear the transmit request bit CTRM.3.
3. Clear the transmit abort control bit CABORT.0.

As the abort transmission request might fall together with an ongoing transmission, the buffer might not be available immediately after issuing the abort transmission request. Therefore the re-initialization of the buffer should not be started before having confirmed its availability via CTRM.5 (refer to [section 2.4.9](#)).

APPLICATION

2.4 Controller Area Network (CAN) module

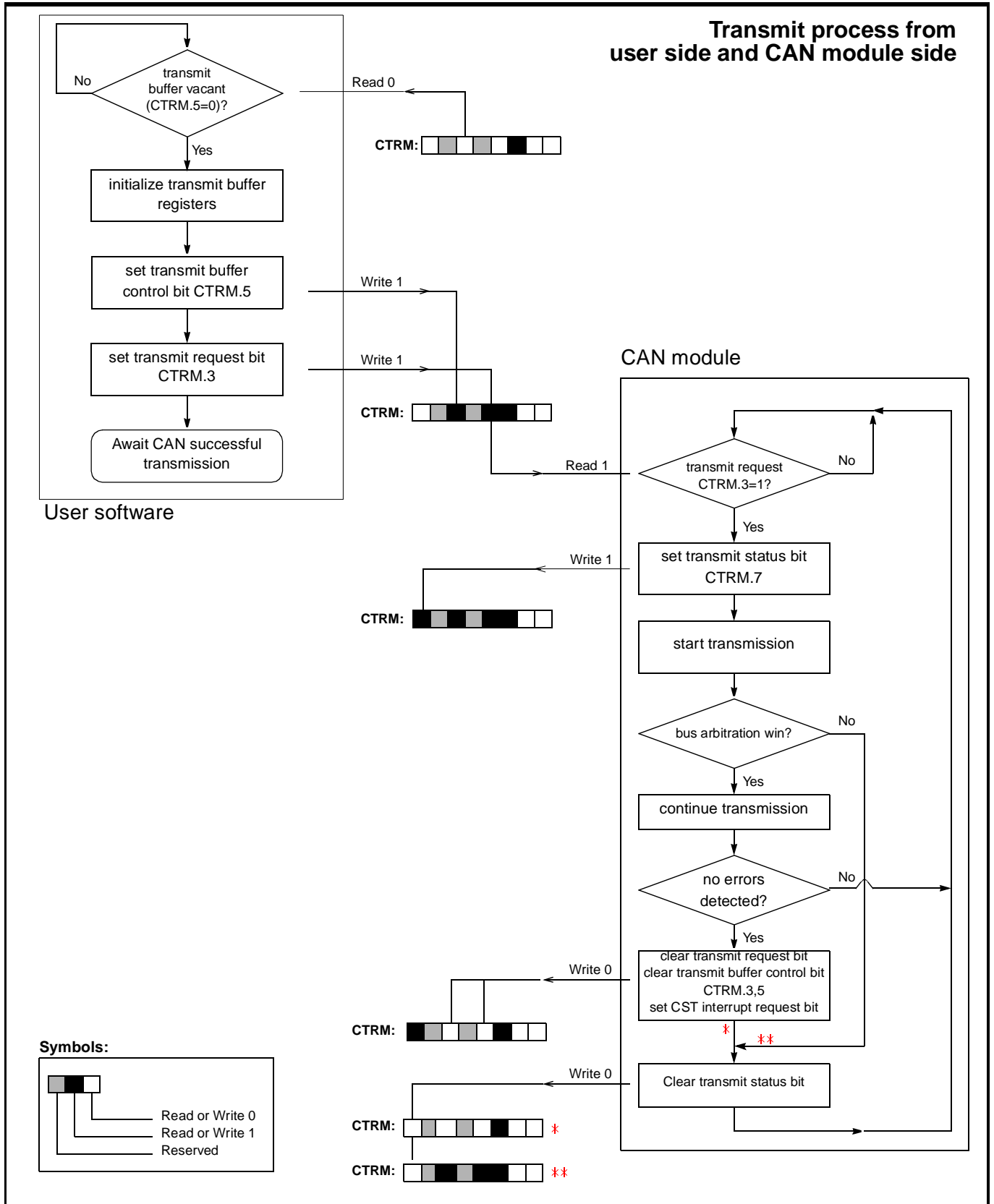


Fig. 2.4.38 Flowchart of transmit process

2.4.11 CAN interrupts

The module provides six interrupt sources with separate interrupt vectors; each interrupt is requested by setting the corresponding interrupt request bit.

Interrupt name	Requested upon	Description	Vector Address		Request bit	Control bit
			High	Low		
CAN successful transmit (CST)	successful transmission of a CAN frame	section 2.4.9	FFF3 ₁₆	FFF2 ₁₆	IREQA.3	ICONA.3
CAN successful receive (CSR)	successful reception of a CAN frame	section 2.4.8	FFF1 ₁₆	FFF0 ₁₆	IREQA.4	ICONA.4
CAN overrun (COVR)	detecting a relevant frame on the bus while no receive buffer is vacant	"Avoiding CAN overrun interrupts", Figure 2.4.35	FFEF ₁₆	FFEE ₁₆	IREQA.5	ICONA.5
CAN error passive (CERP)	state transition from error active to error passive	section 2.4.12	FFED ₁₆	FFEC ₁₆	IREQA.6	ICONA.6
CAN bus off (CBOF)	state transition from error passive to bus off	section 2.4.12	FFEB ₁₆	FFEA ₁₆	IREQA.7	ICONA.7
CAN wake up (CWKU)	detecting a recessive to dominant edge on CRX	section 2.4.13	FFE9 ₁₆	FFE8 ₁₆	IREQB.0	ICONB.0

2.4.12 Error condition

As defined in the CAN specification, the module features internal transmit and receive error counters; these counters serve to define the state of the module between the options *error active*, *error passive* and *bus off*. In normal (run) mode, the error counters are increased upon detection of an error and decreased upon successful transmission or reception of CAN frames following the rules of the CAN specification. These counters are internal registers and not available to the CPU; the transitions from error active to error passive state and from error passive to bus off are flagged by a request of the corresponding error passive (CERP) and bus off (CBOF) interrupts. In normal (run) mode, the receive and transmit error counters are under control of the module and can not be altered or read by the CPU. Upon switching to configuration mode however, the counters are cleared and the module is put to error active state. As defined in the CAN specification, the module takes part in normal bus communication and flags errors detected by sending an active error flag. After the module has transit to error passive state the module continues communication but errors detected are flagged by a passive error flag. In bus off state, the module suspends the communication and does not influence the bus any more; the CTX pin is kept at recessive level. Please refer to the state diagram in Figure 2.4.39 and to the CAN specification for details on the conditions leading to state-transitions.

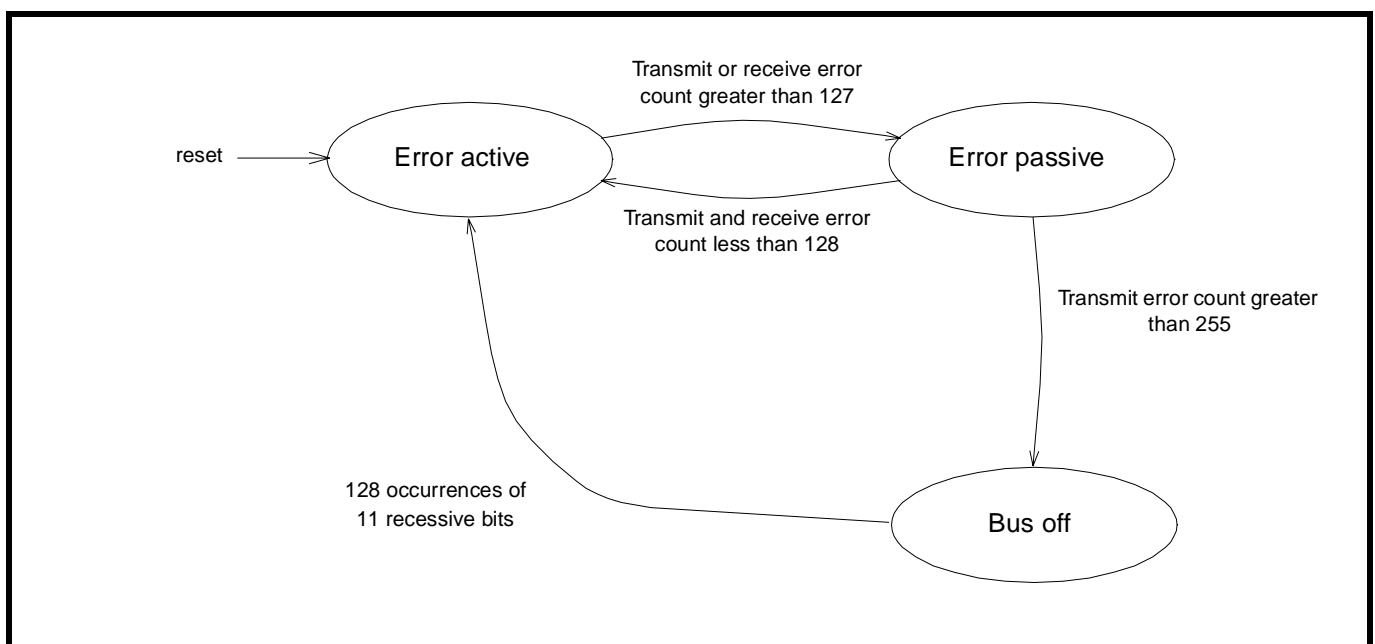


Fig. 2.4.39 Error state diagram

APPLICATION

2.4 Controller Area Network (CAN) module

2.4.13 Wake-up via CAN

The module features a function to wake-up the CPU on CAN traffic. This feature is implemented by an external interrupt function on the P3₂/CRX input port. The wake-up interrupt is requested upon a recessive to dominant edge (by start-of-frame of a message sent by another node) on the P3₂/CRX.

The initialization required to use the wake-up function is:

1. Clear CWKU interrupt request bit IREQB.0.
2. Set CWKU interrupt control bit ICONB.0.
3. Put module to sleep mode by setting the sleep control bit CTRM.0 ([Figure 2.4.3](#)); see section [2.4.4](#).
4. Set the pull transistor enable bit PUP3.2; this activates the pull transistor towards the recessive level (depending on the dominant polarity selected by PCON.1).
5. Put MCU to low-power mode (wait or stop mode).

The first frame sent by another node awakens the CPU by an CWKU interrupt request. The module should be put to normal (run) mode again as part of the CWKU interrupt service; also the pull transistor should be disabled as part of this interrupt service routine.

Note: The frame triggering the wake-up function can not be received and is lost.

2.5 Serial I/O

2.5.1 Memory map of serial I/O

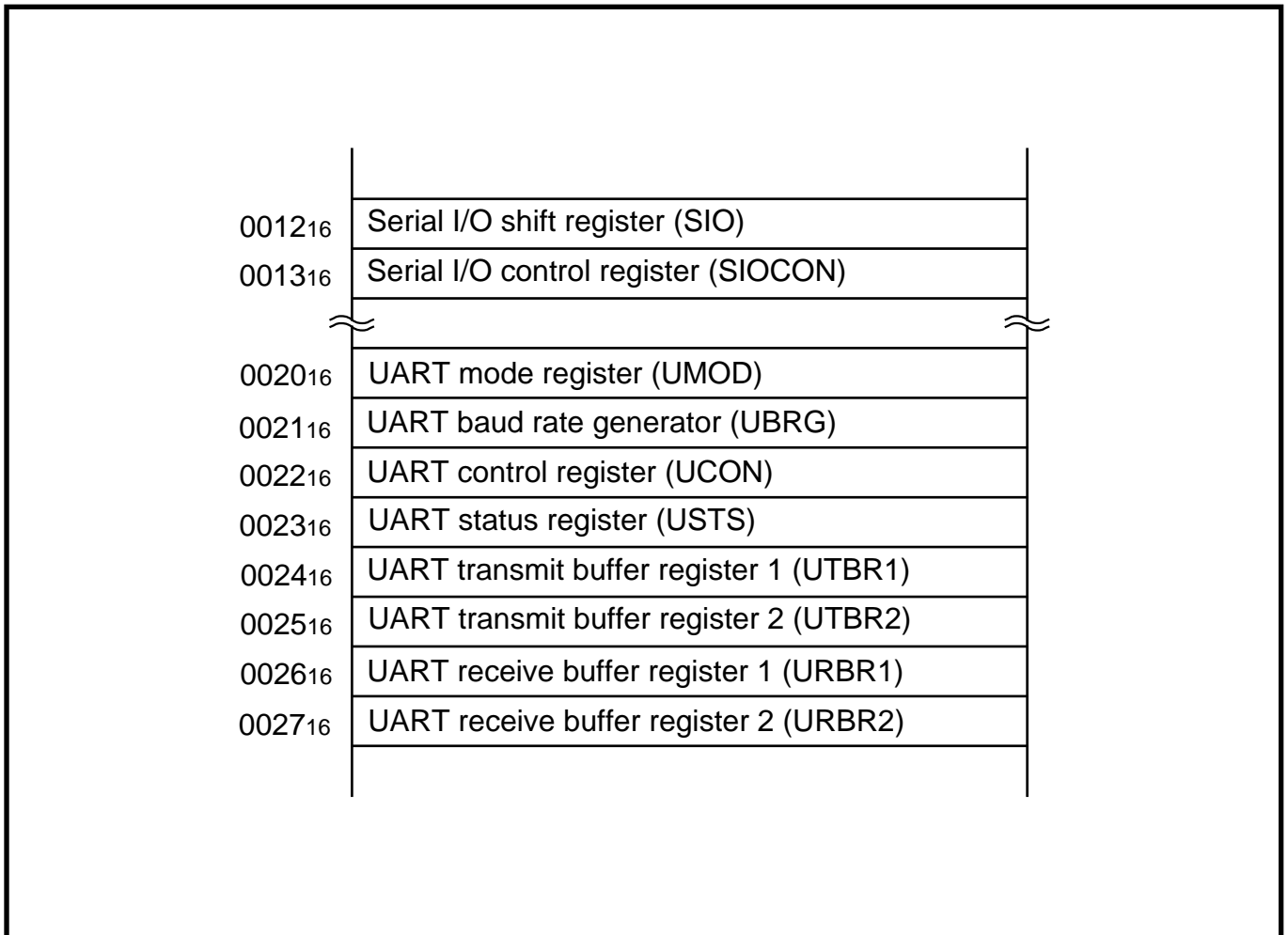


Fig. 2.5.1 Memory map of serial I/O related registers

APPLICATION

2.5 Serial I/O

2.5.2 Related registers

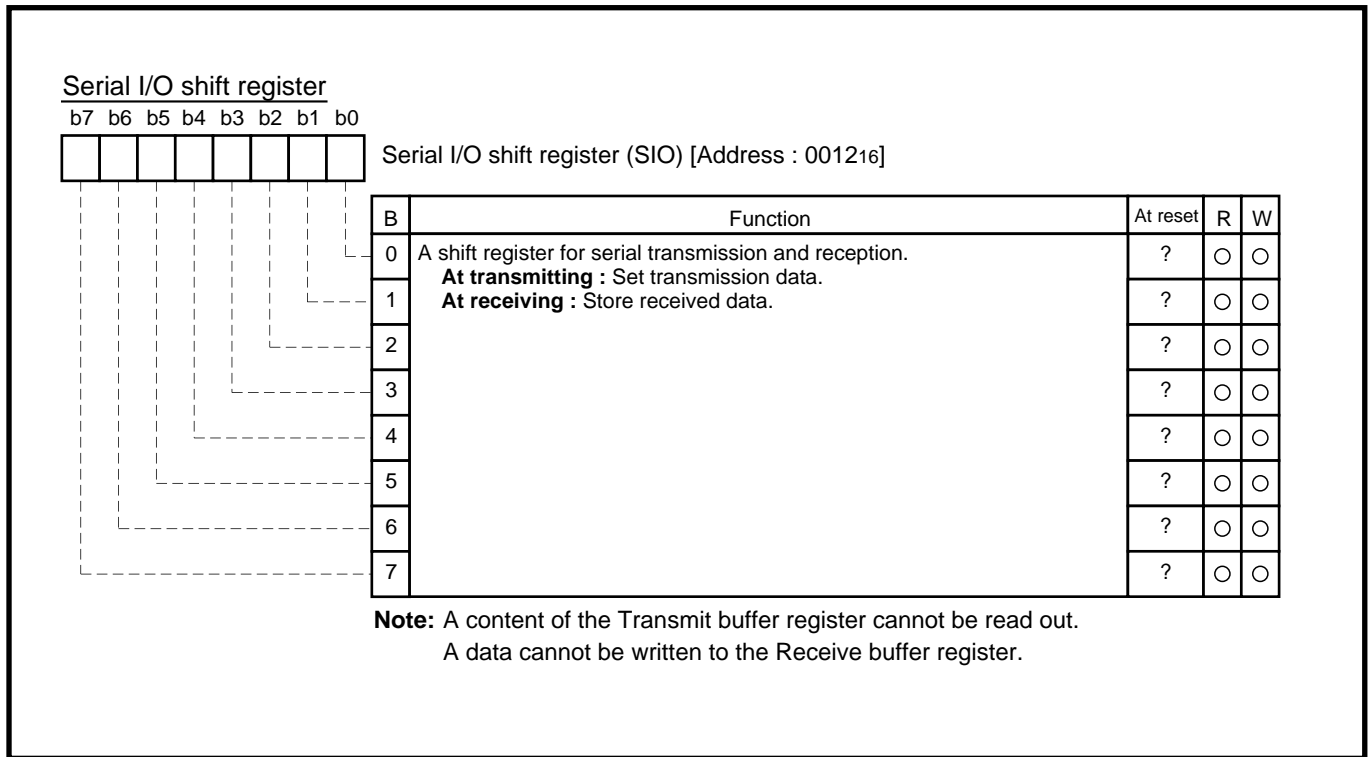


Fig. 2.5.2 Structure of Serial I/O shift register

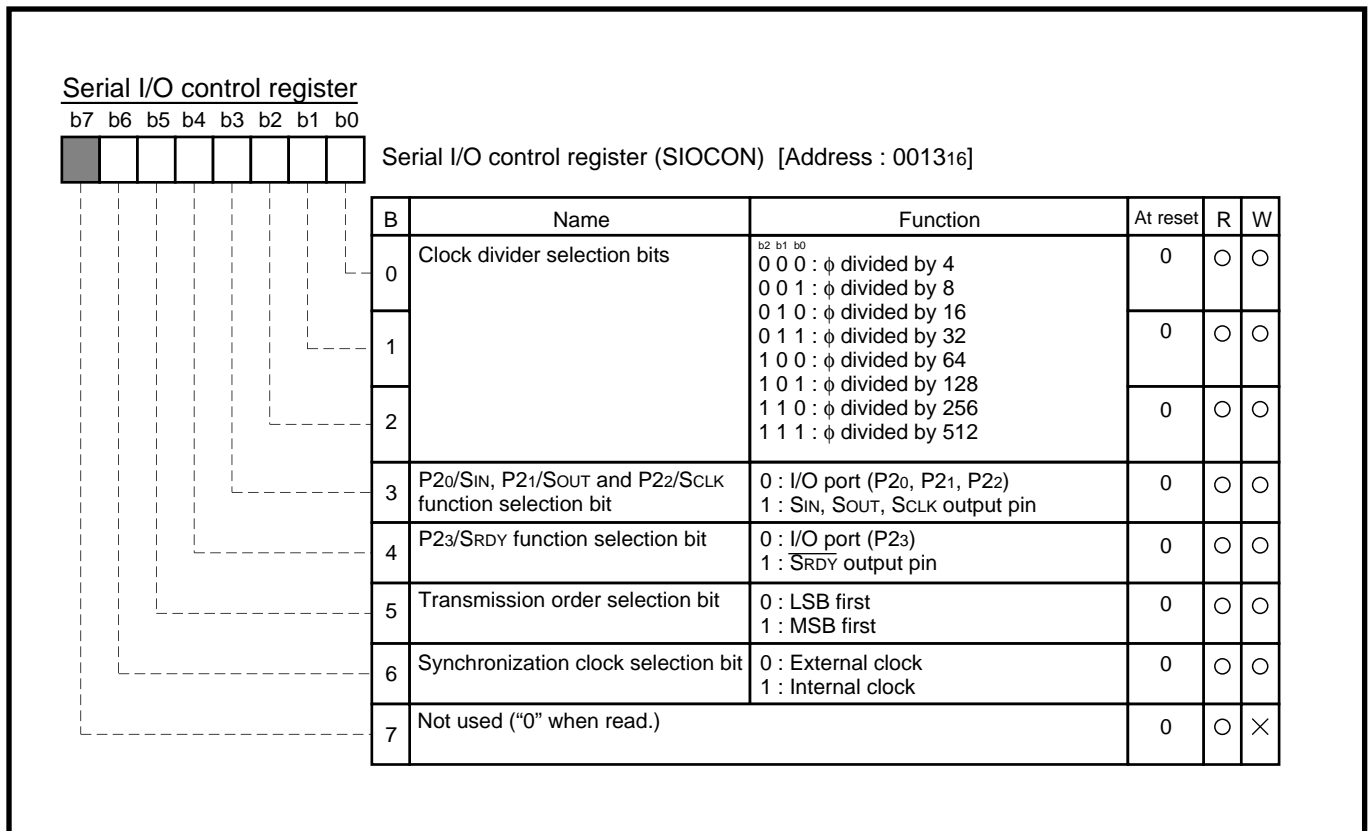


Fig. 2.5.3 Structure of Serial I/O control register

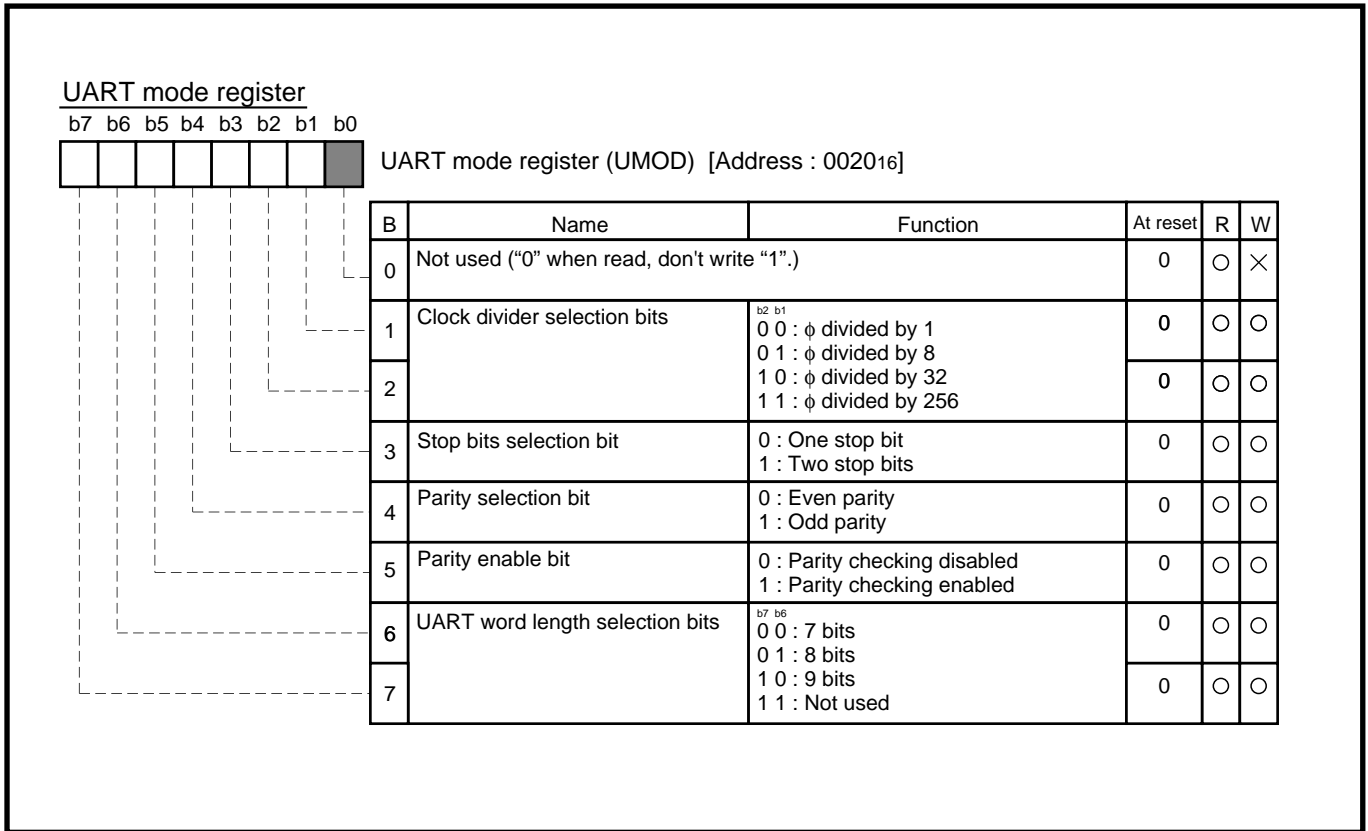


Fig. 2.5.4 Structure of UART mode register

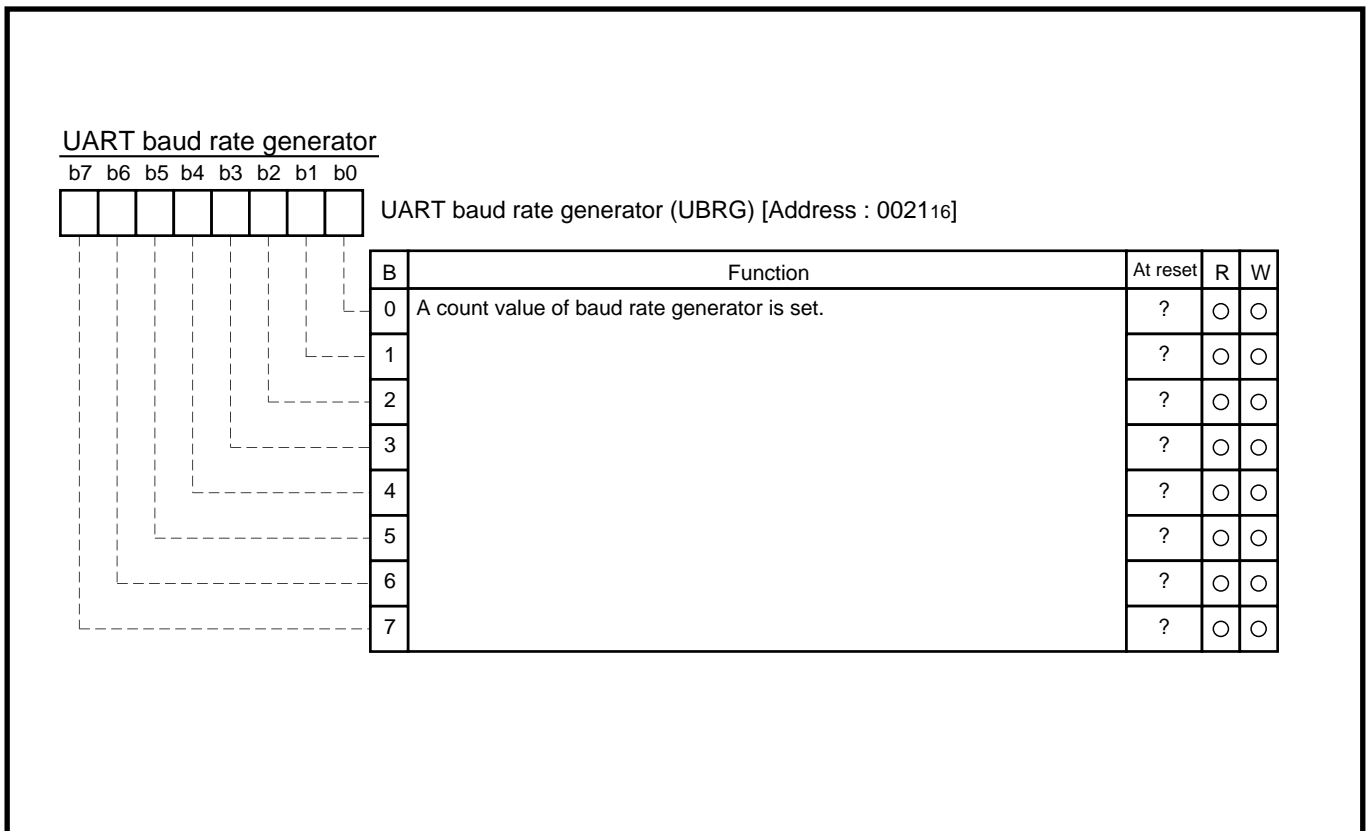


Fig. 2.5.5 Structure of UART baud rate generator

APPLICATION

2.5 Serial I/O

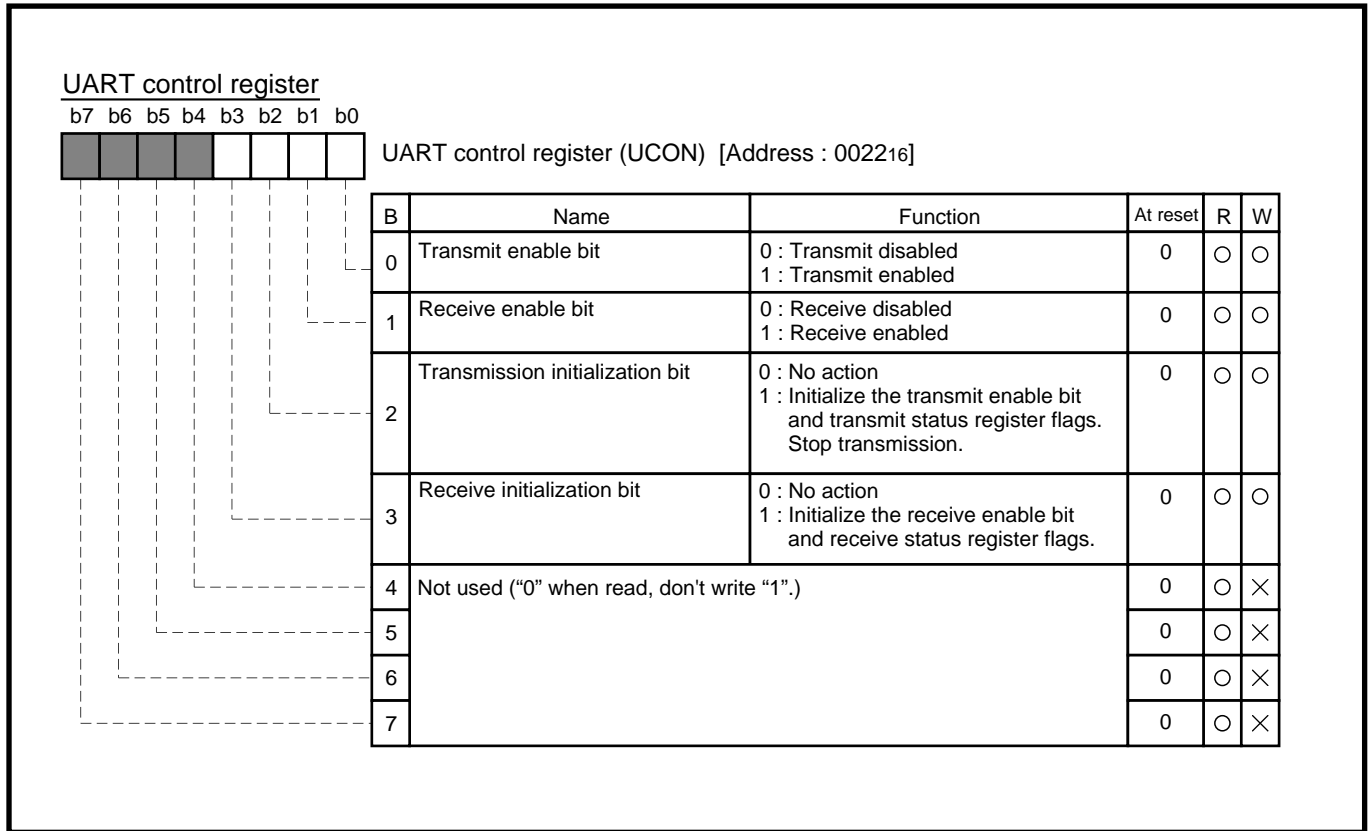


Fig. 2.5.6 Structure of UART control register

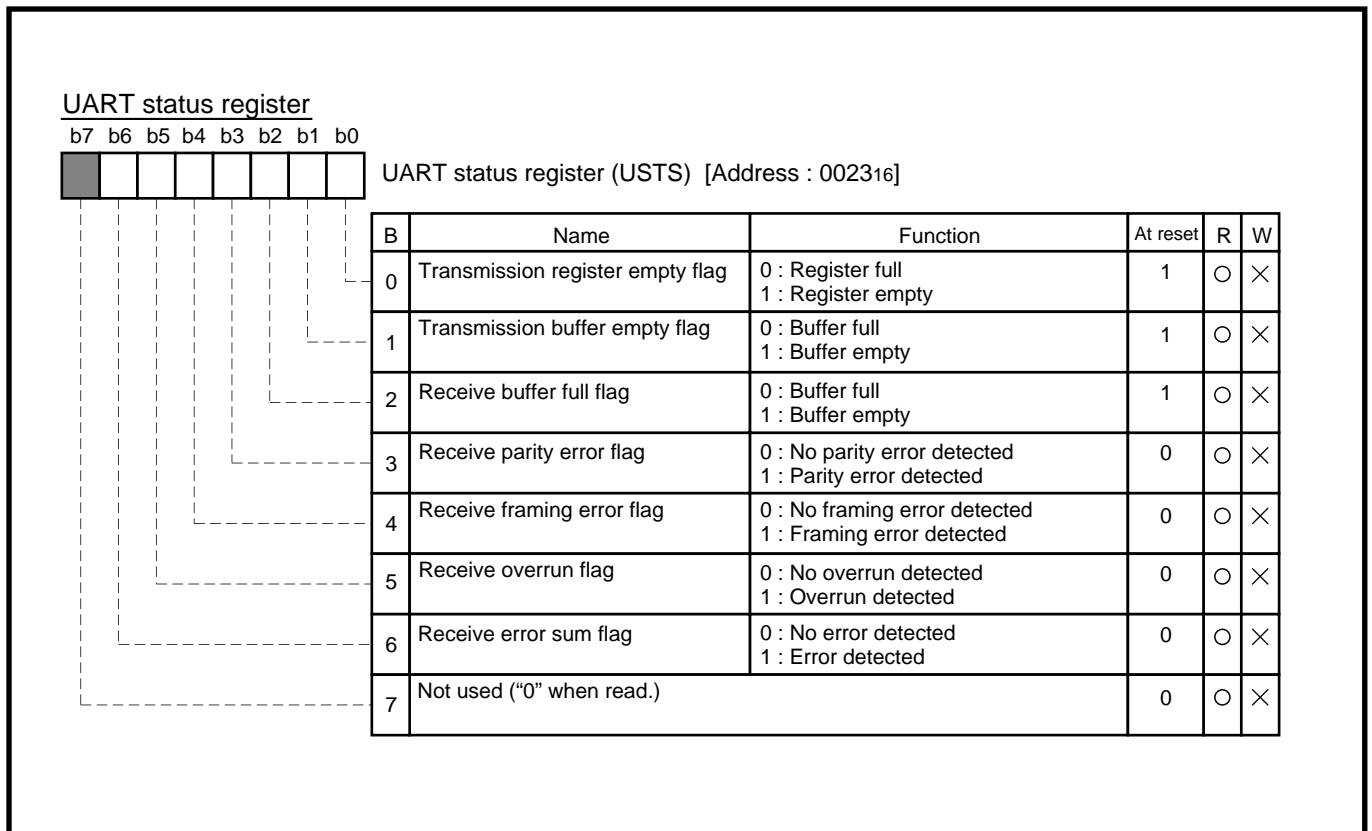


Fig. 2.5.7 Structure of UART status register

UART transmit buffer register 1

UART transmit buffer register 2

b7 b6 b5 b4 b3 b2 b1 b0



UART transmit buffer register 1 (UTBR1) [Address : 002416]

UART transmit buffer register 2 (UTBR2) [Address : 002516]

B	Function	At reset	R	W
0	Transmit data is written to this buffer register (consisting of low-order and high-order byte).	?	×	○
1		?	×	○
2		?	×	○
3		?	×	○
4		?	×	○
5		?	×	○
6		?	×	○
7		?	×	○

Fig. 2.5.8 Structure of UART transmit buffer register 1, 2

UART receive buffer register 1

UART receive buffer register 2

b7 b6 b5 b4 b3 b2 b1 b0



UART receive buffer register 1 (URBR1) [Address : 002616]

UART receive buffer register 2 (URBR2) [Address : 002716]

B	Function	At reset	R	W
0	Receive data is read from this buffer register (consisting of low-order and high-order byte).	?	○	×
1		?	○	×
2		?	○	×
3		?	○	×
4		?	○	×
5		?	○	×
6		?	○	×
7		?	○	×

Fig. 2.5.9 Structure of UART receive buffer register 1, 2

APPLICATION

2.5 Serial I/O

2.5.3 Serial I/O connection examples

(1) Control of peripheral IC equipped with CS pin

Figure 2.5.10 shows connection examples with peripheral ICs using clock synchronous serial I/O mode.

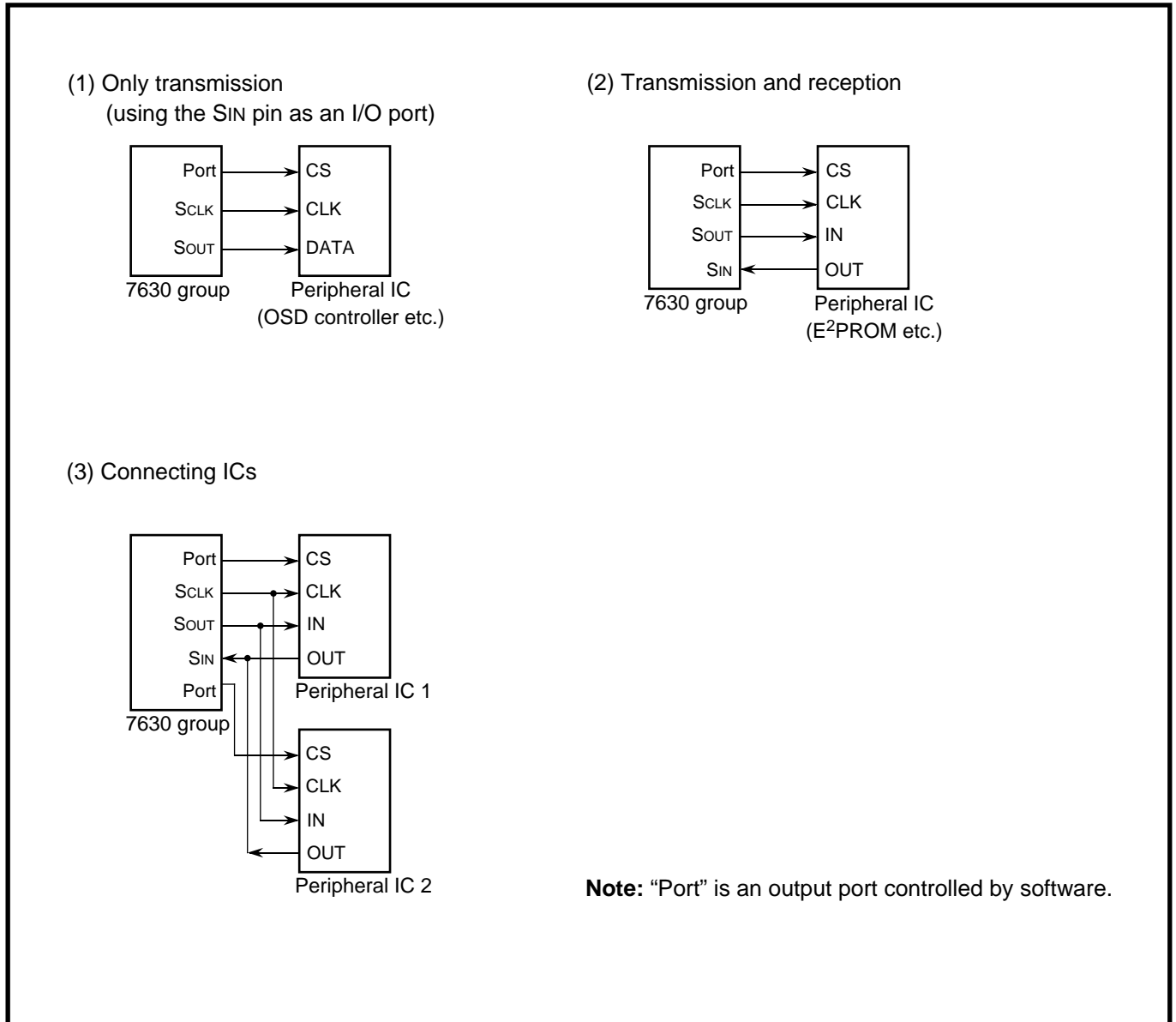


Fig. 2.5.10 Serial I/O connection examples (1)

(2) Connection with microcomputer

Figure 2.5.11 shows connection examples with other microcomputers using serial I/O.

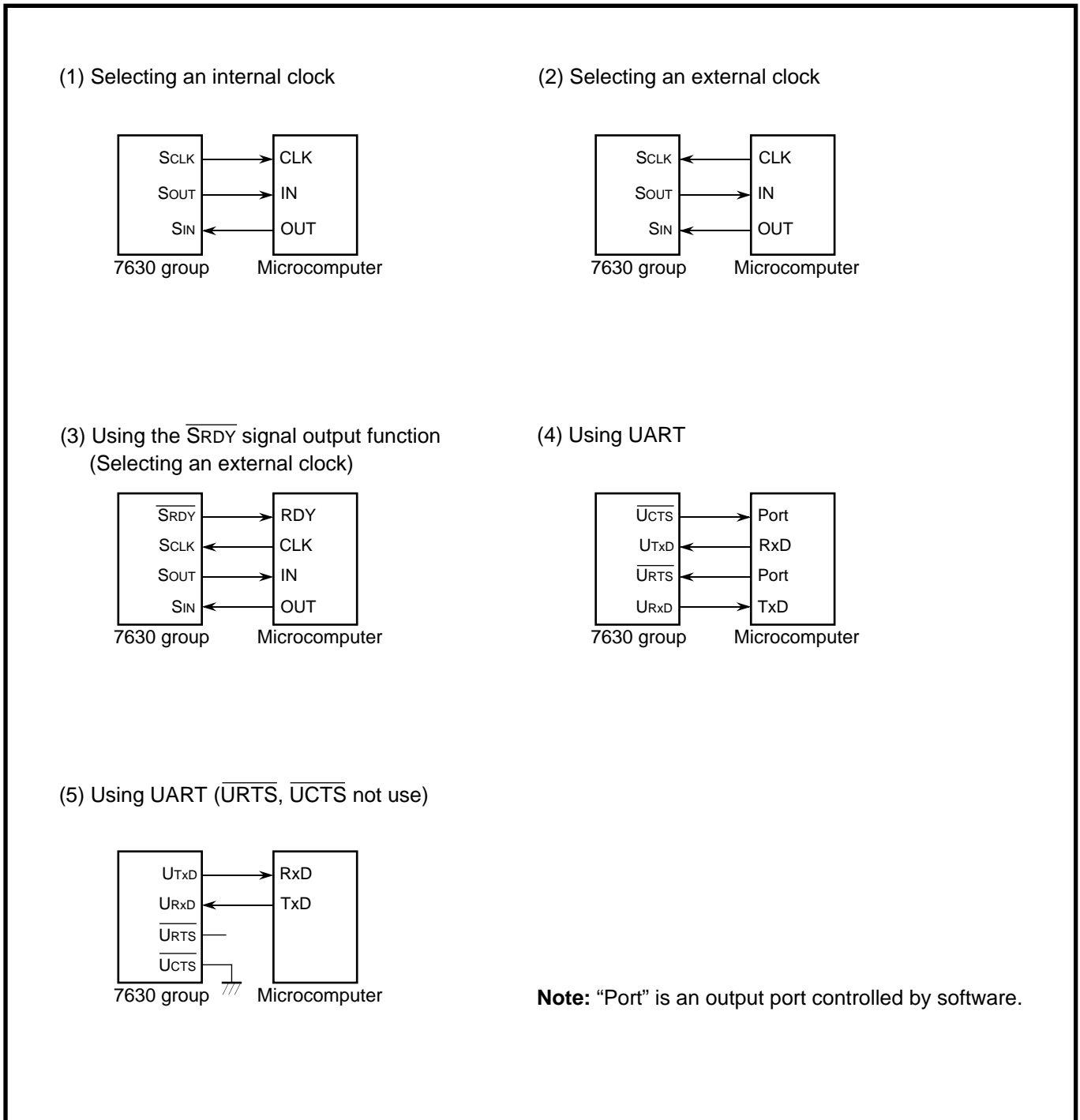


Fig. 2.5.11 Serial I/O connection examples (2)

APPLICATION

2.5 Serial I/O

2.5.4 Setting of serial I/O transfer data format

A clock synchronous or clock asynchronous (UART) is selected as a data format. Figure 2.5.12 shows a setting of serial I/O transfer data format.

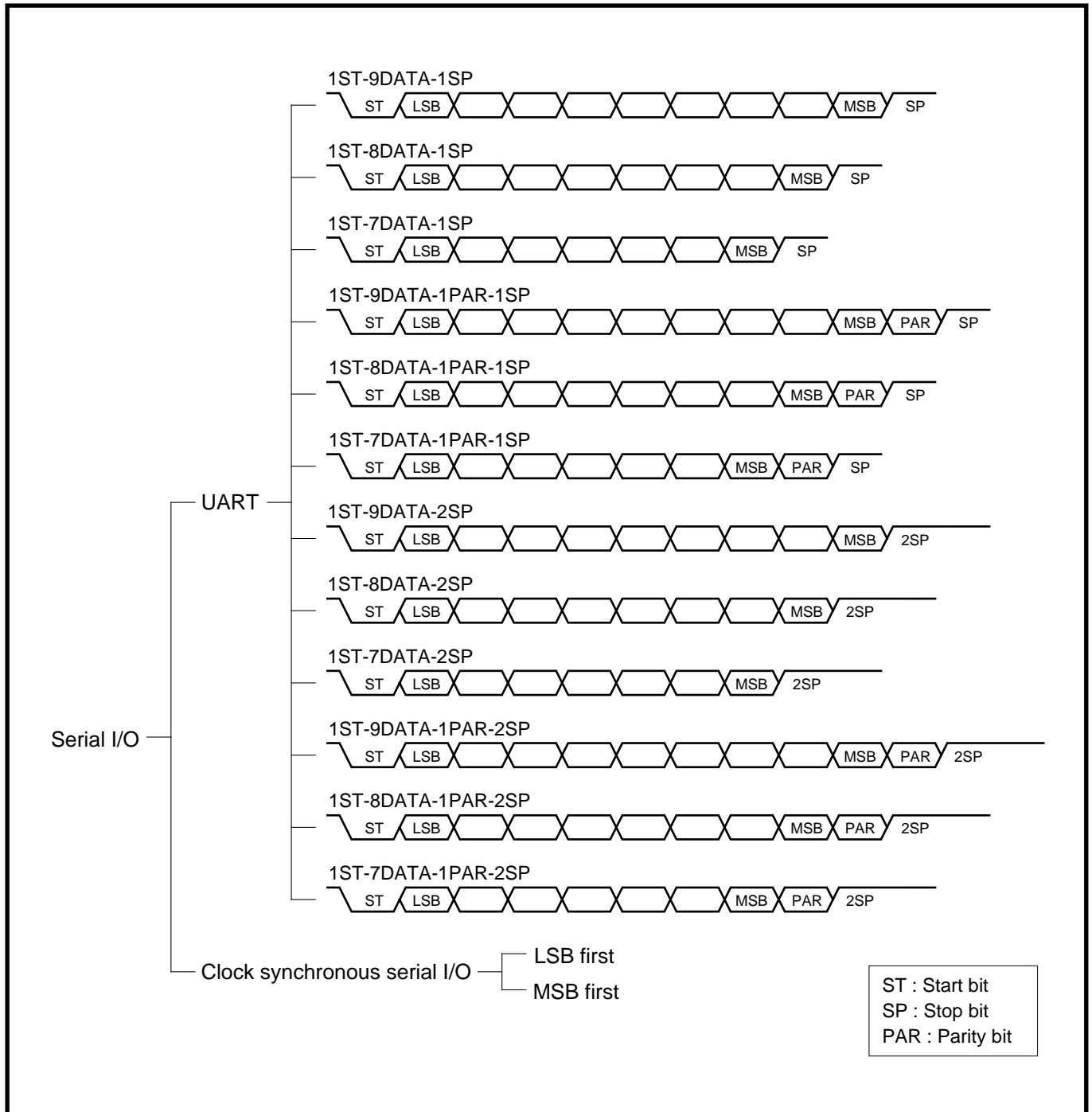


Fig. 2.5.12 Setting of serial I/O transfer data format

2.5.5 Serial I/O application examples

(1) Output of serial data (control of a peripheral IC)

Outline : 4-byte data is transmitted and received using the clock synchronous serial I/O. The CS signal is output to a peripheral IC through the port P33.

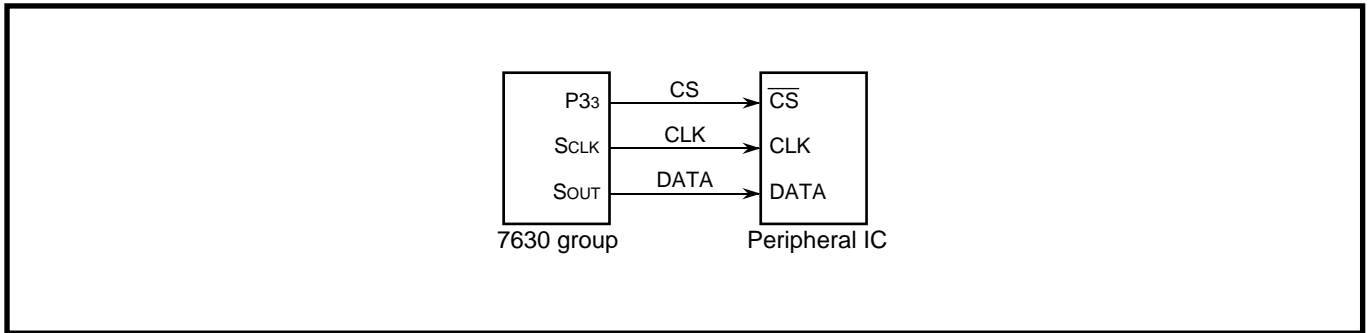
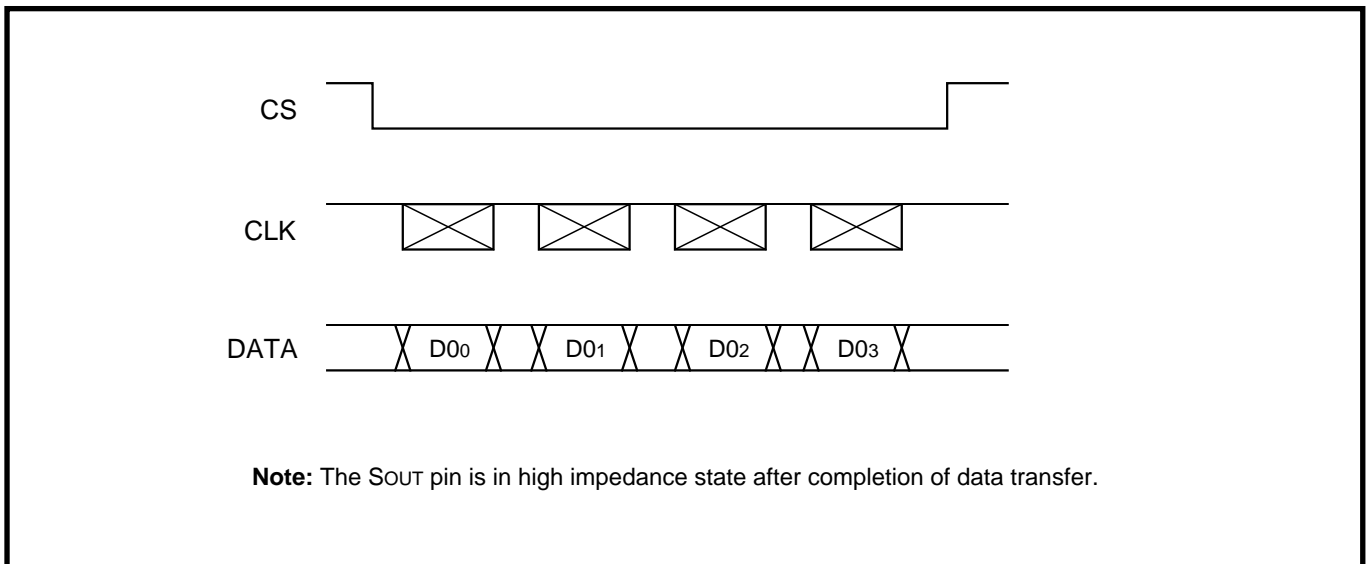


Fig. 2.5.13 Connection diagram [Output of serial data]

- Specifications :**
- The Serial I/O is used(the clock synchronous serial I/O is selected).
 - Synchronous clock frequency : 125 kHz ($f(XIN) = 8 \text{ MHz}$ is divided by 64)
 - Transfer direction : LSB first.
 - The Serial I/O interrupt is not used.
 - Port P33 is connected to the \overline{CS} pin ("L" active) of the peripheral IC for transmission control (the output level of port P33 is controlled by software).

Figure 2.5.14 shows an output timing chart of serial data.



Note: The SOUT pin is in high impedance state after completion of data transfer.

Fig. 2.5.14 Timing chart [Output of serial data]

APPLICATION

2.5 Serial I/O

Figure 2.5.15 shows a setting of serial I/O related registers, and Figure 2.5.16 shows a setting of serial I/O transmission data.

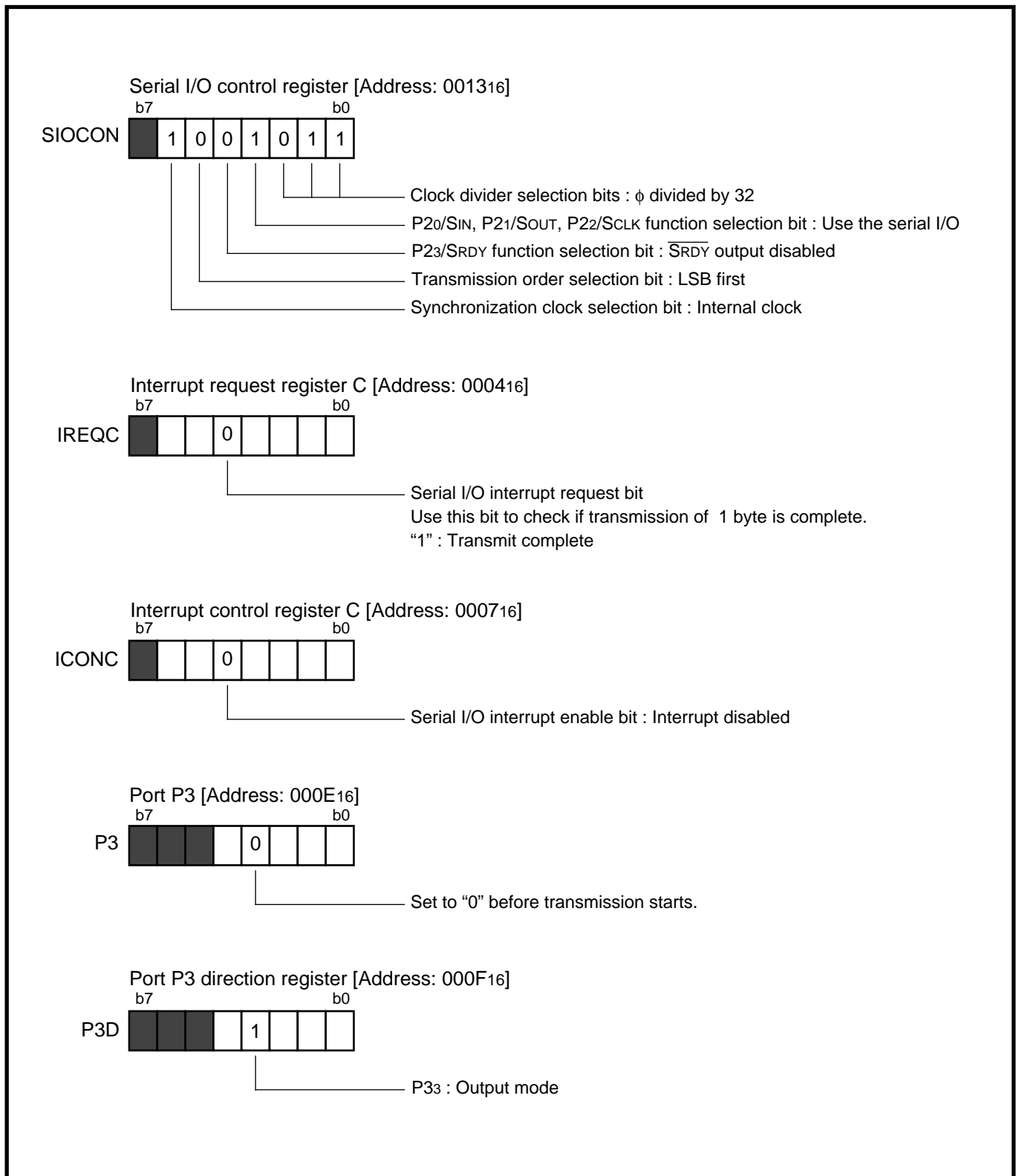


Fig. 2.5.15 Setting of serial I/O related registers [Output of serial data]

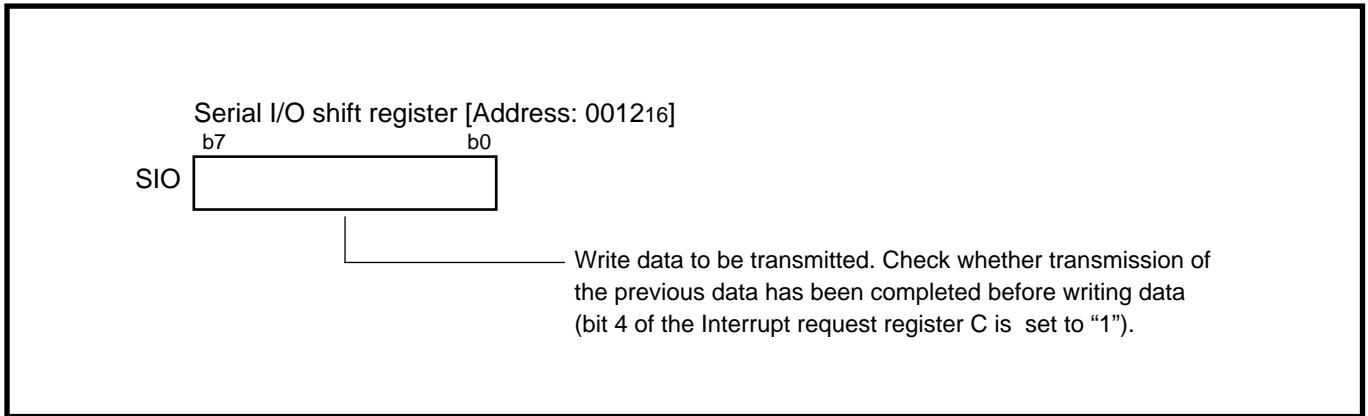


Fig. 2.5.16 Setting of serial I/O transmission data [Output of serial data]

APPLICATION

2.5 Serial I/O

Control procedure : When the registers are set as shown in Figure 2.5.15, the Serial I/O transmits 1-byte data simply by writing data to the Serial I/O shift register. Thus, after setting the CS signal to “L”, write the transmission data to the Serial I/O shift register on 1-byte base, and return the CS signal to “H” when the desired number of bytes have been transmitted. Figure 2.5.17 shows a control procedure of serial I/O(clock synchronous serial I/O).

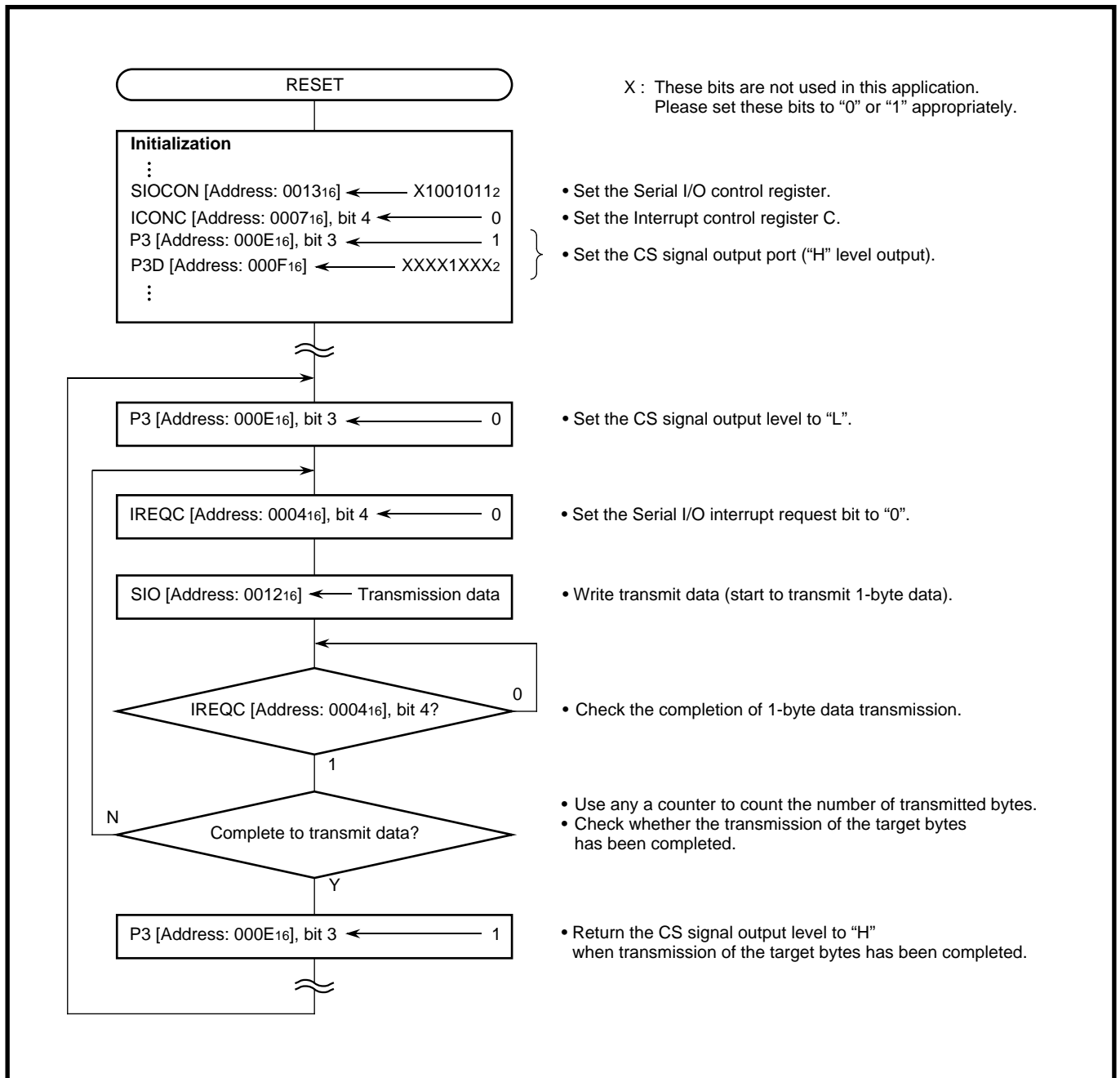


Fig. 2.5.17 Control procedure of clock synchronous serial I/O [Output of serial data]

(2) Communication (transmit/receive) using asynchronous serial I/O (UART) 1

Point : 1-word data is transmitted and received through asynchronous serial I/O.

Figure 2.5.18 shows a connection diagram, and Figure 2.5.19 shows a timing chart.

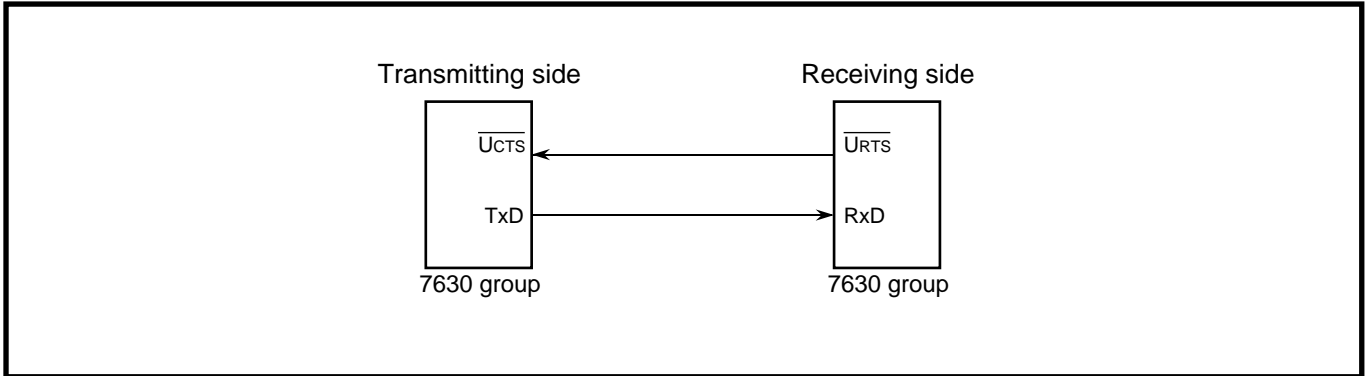


Fig. 2.5.18 Connection diagram [Communication using UART]

- Specifications :**
- The Serial I/O is used (UART is selected).
 - Transfer bit rate : 9600 bps ($f(XIN) = 10.0 \text{ MHz}$ is divided by 1024)
 - Communication control using port \overline{URTS} and \overline{UCTS}

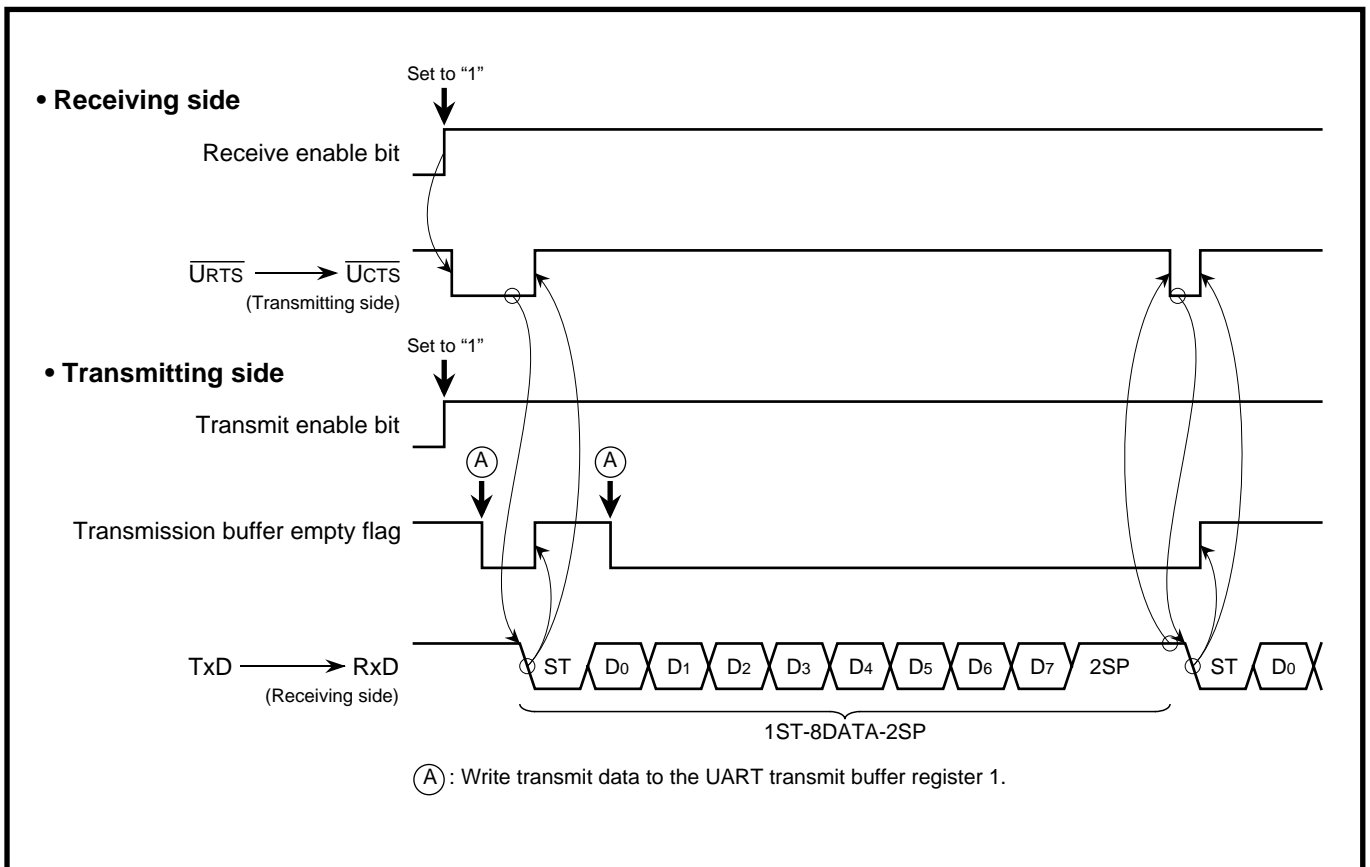


Fig. 2.5.19 Timing chart [Communication using UART]

APPLICATION

2.5 Serial I/O

Table 2.5.1 shows setting examples of UART baud rate generator (UBRG) values and transfer bit rate values, Figure 2.5.20 shows a setting of related registers on the transmitting side, and Figure 2.5.21 shows a setting of related registers on the receiving side.

Table 2.5.1 Setting examples of Baud rate generator values and transfer bit rate values

Transfer bit rate (bps) (Note 1)	BRG count source (Note 2)	at $\phi = 8\text{MHz} / 2$		at $\phi = 10\text{MHz} / 2$	
		BRG setting value	Actual time (bps)	BRG setting value	Actual time (bps)
75	$\phi/256$	12 (0C16)	75.12	15 (0F16)	76.29
150	$\phi/256$	5 (0516)	162.76	7 (0716)	152.58
300	$\phi/32$	25 (1916)	300.48	31 (1F16)	305.17
600	$\phi/32$	12 (0C16)	600.96	15 (0F16)	610.35
1200	$\phi/32$	5 (0516)	1302.08	7 (0716)	1220.70
2400	$\phi/8$	12 (0C16)	2403.84	15 (0F16)	2441.40
4800	$\phi/8$	5 (0516)	5208.33	7 (0716)	4882.81
9600	$\phi/8$	2 (0216)	10416.66	3 (0316)	9765.62
19200	ϕ	12 (0C16)	19230.76	15 (0F16)	19531.25
31250	ϕ	7 (0716)	31250.00	9 (0916)	31250.00
62500	ϕ	3 (0316)	62500.00	4 (0416)	62500.00
76800	ϕ	2 (0216)	83333.33	3 (0316)	78125.00

Notes 1: Equation of transfer bit rate

$$\text{Transfer bit rate (bps)} = \frac{\phi}{(\text{BRG setting value} + 1) \times 16 \times p^*}$$

Notes 2: BRG count source (ϕ/p^*) is selected by the Clock divider selection bits (bit 1 and 2 of the UART mode register [Address: 002016]).

* The Value p is decided by the Clock divider selection bits (bit 1 and 2 of the UART mode register [Address: 002016]). Refer to Table 2.5.2.

Table 2.5.2 Clock divider selection for serial I/O

Clock divider p	Clock divider selection bits	
	bit 2	bit 1
1	0	0
8	0	1
32	1	0
256	1	1

APPLICATION

2.5 Serial I/O

• Receiving side

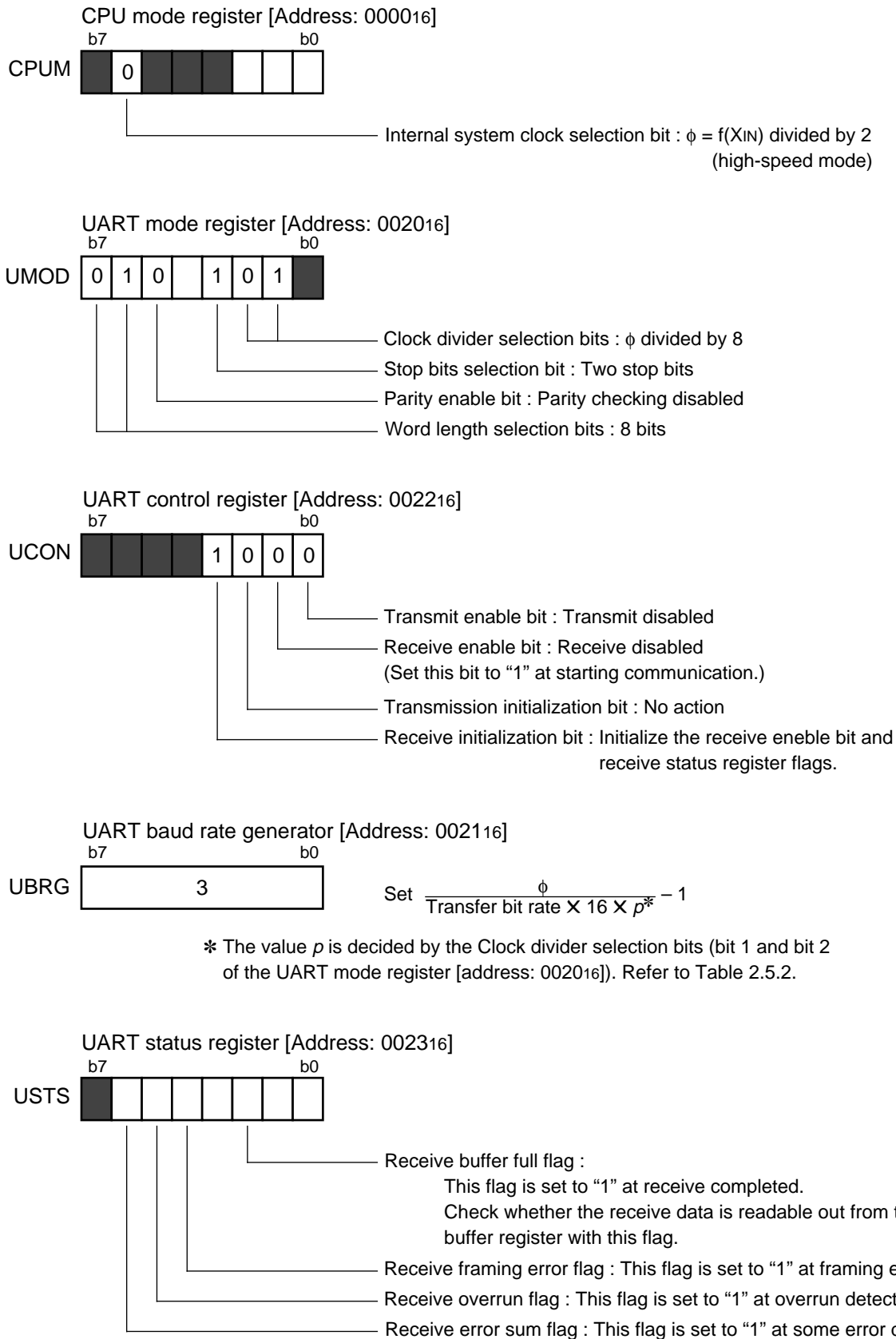


Fig. 2.5.21 Setting of related registers on receiving side [Communication using UART]

Control procedure : Figure 2.5.22 shows a control procedure on the transmitting side, and Figure 2.5.23 shows a control procedure on the receiving side.

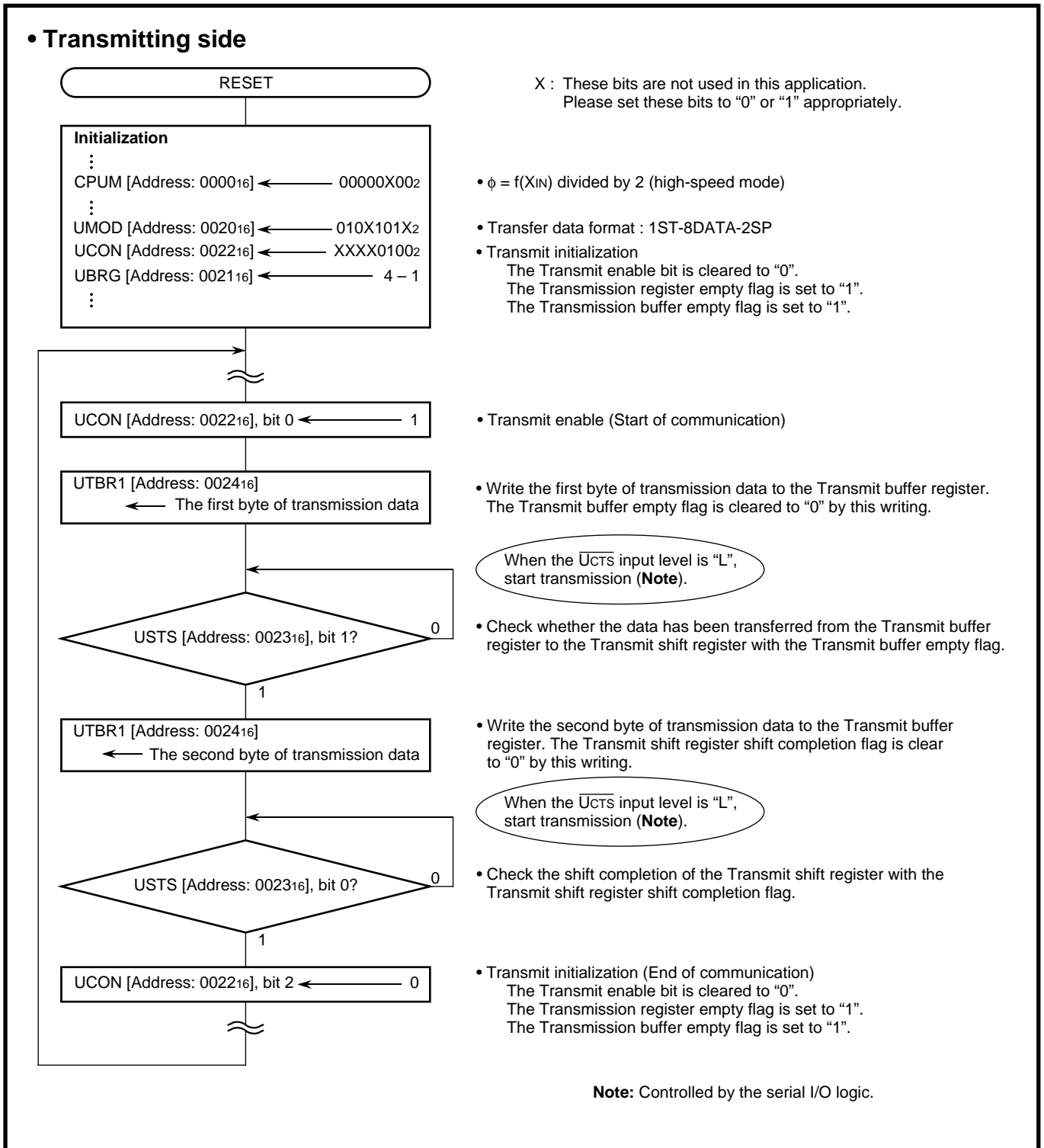


Fig. 2.5.22 Control procedure on transmitting side [Communication using UART]

APPLICATION

2.5 Serial I/O

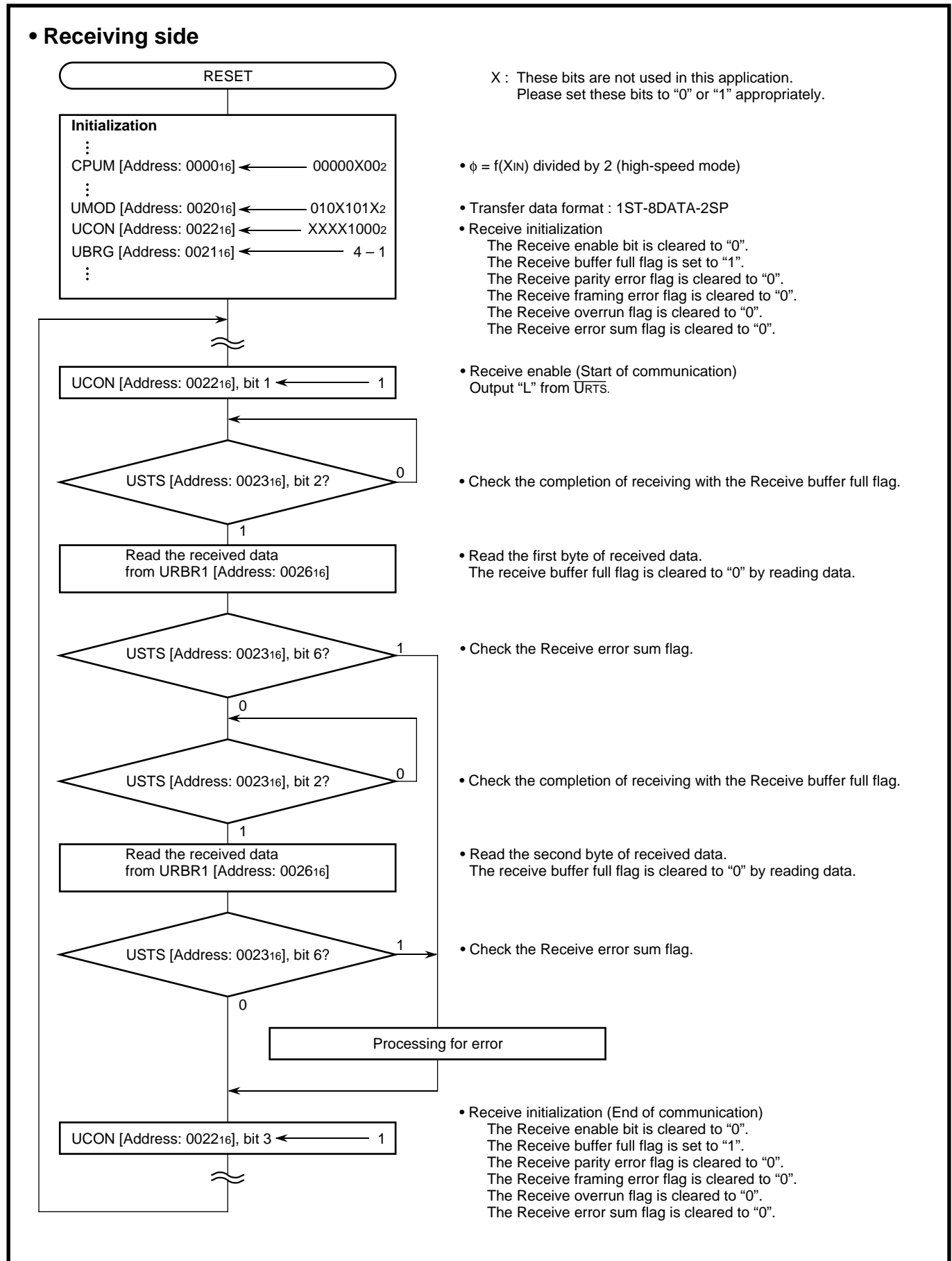


Fig. 2.5.23 Control procedure on receiving side [Communication using UART]

(3) Communication (transmit/receive) using asynchronous serial I/O (UART) 2

Point : 9-bit data is transmitted and received through asynchronous serial I/O.

Figure 2.5.24 shows a connection diagram, and Figure 2.5.25 shows a timing chart.

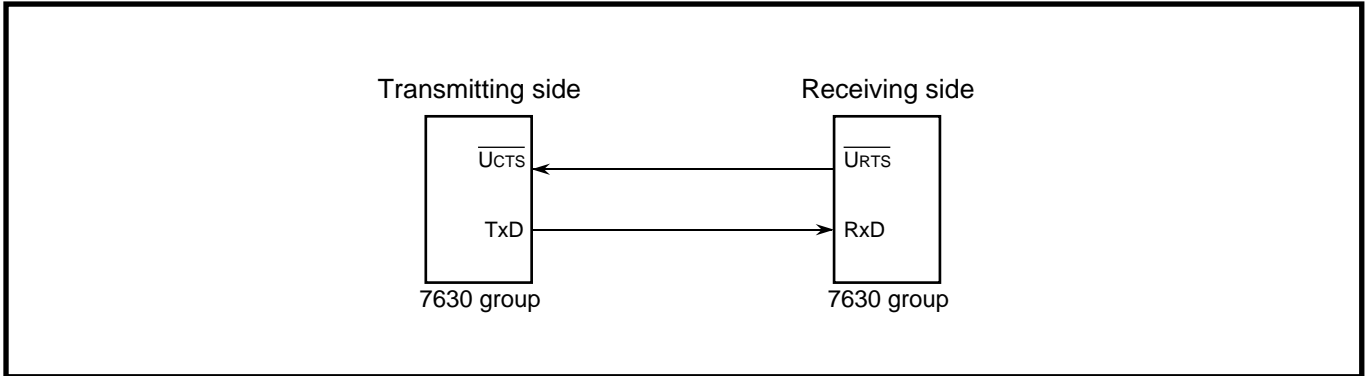


Fig. 2.5.24 Connection diagram [Communication using UART]

- Specifications :**
- The Serial I/O is used (UART is selected).
 - Transfer bit rate : 9600 bps ($f(\text{XIN}) = 10.0 \text{ MHz}$ is divided by 1024)
 - Communication control using port $\overline{\text{URTS}}$ and $\overline{\text{UCTS}}$

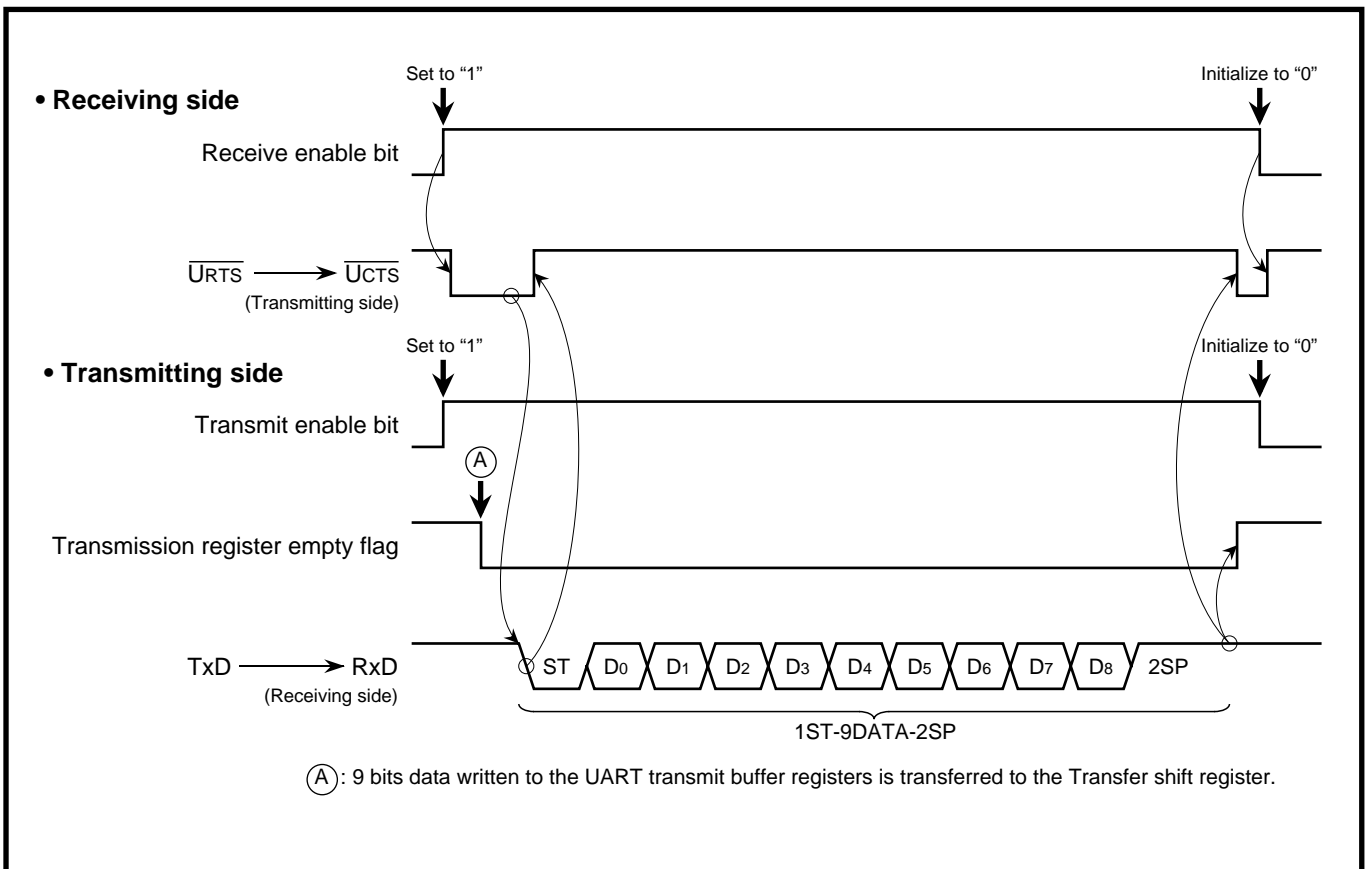


Fig. 2.5.25 Timing chart [Communication using UART]

APPLICATION

2.5 Serial I/O

Figure 2.5.26 shows a setting of related registers at a transmitting side, and Figure 2.5.27 shows a setting of related registers at a receiving side.

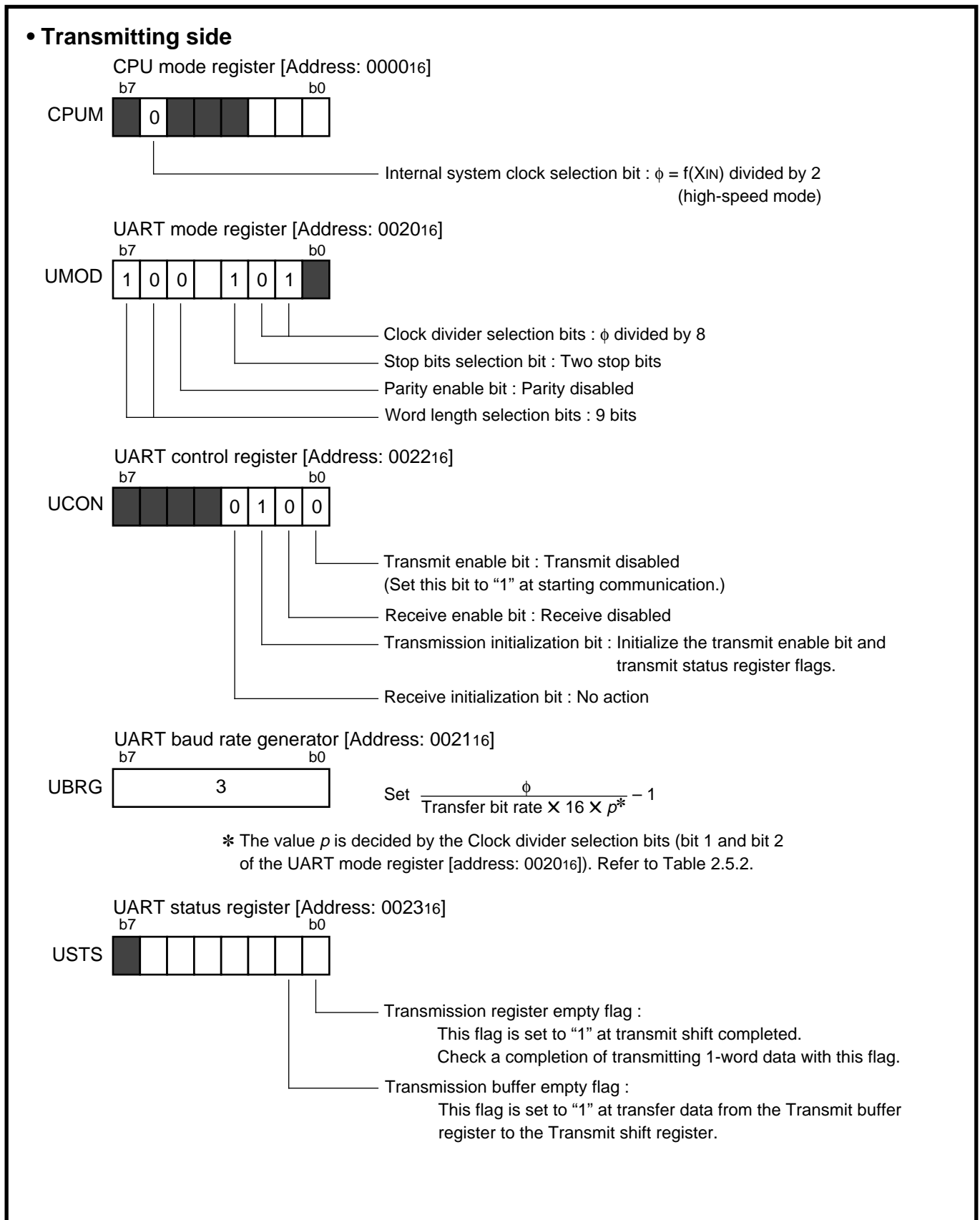


Fig. 2.5.26 Setting of related registers on transmitting side [Communication using UART]

APPLICATION

2.5 Serial I/O

Control procedure : Figure 2.5.28 shows a control procedure on the transmitting side, and Figure 2.5.29 shows a control procedure on the receiving side.

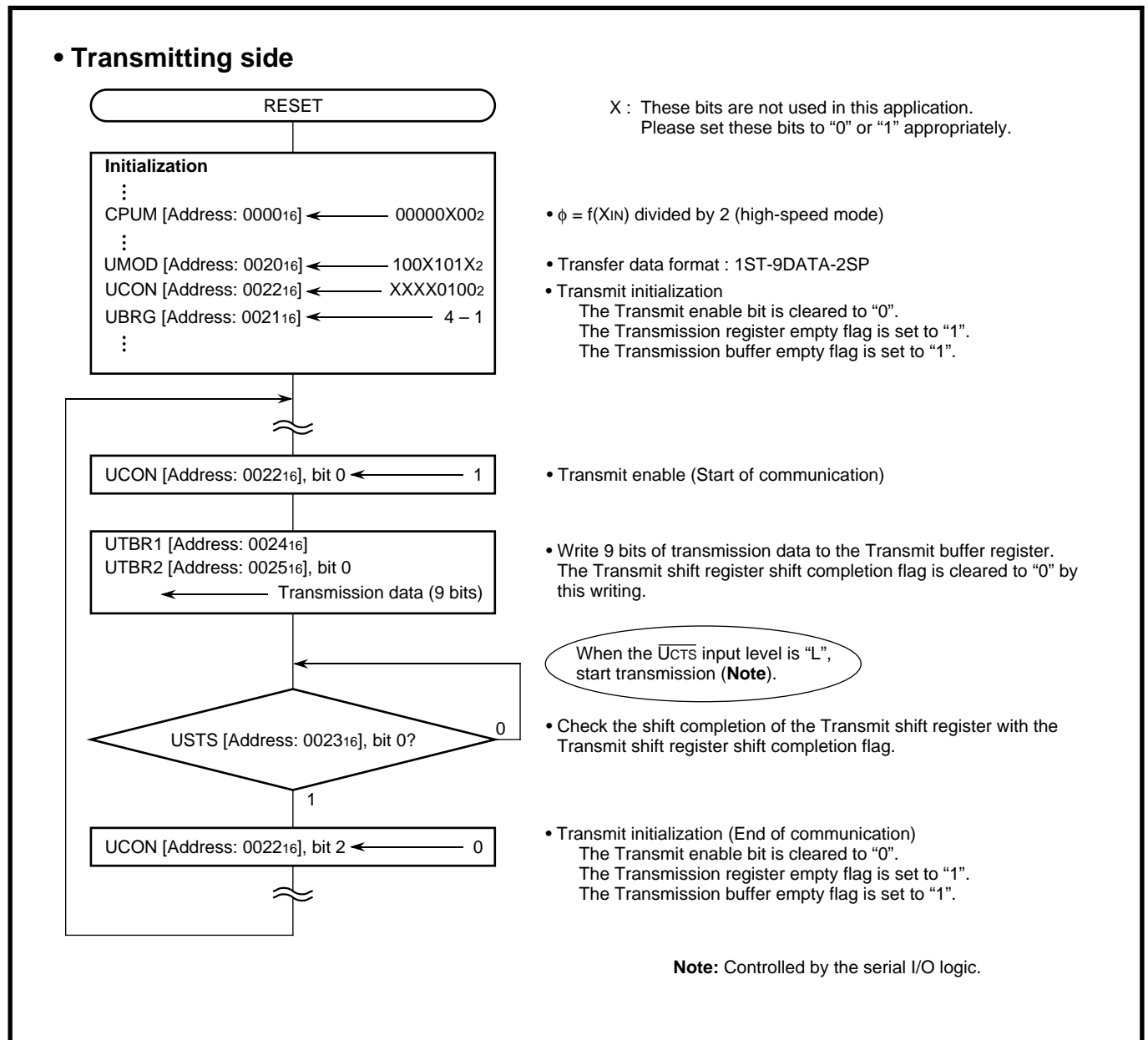


Fig. 2.5.28 Control procedure on transmitting side [Communication using UART]

• Receiving side

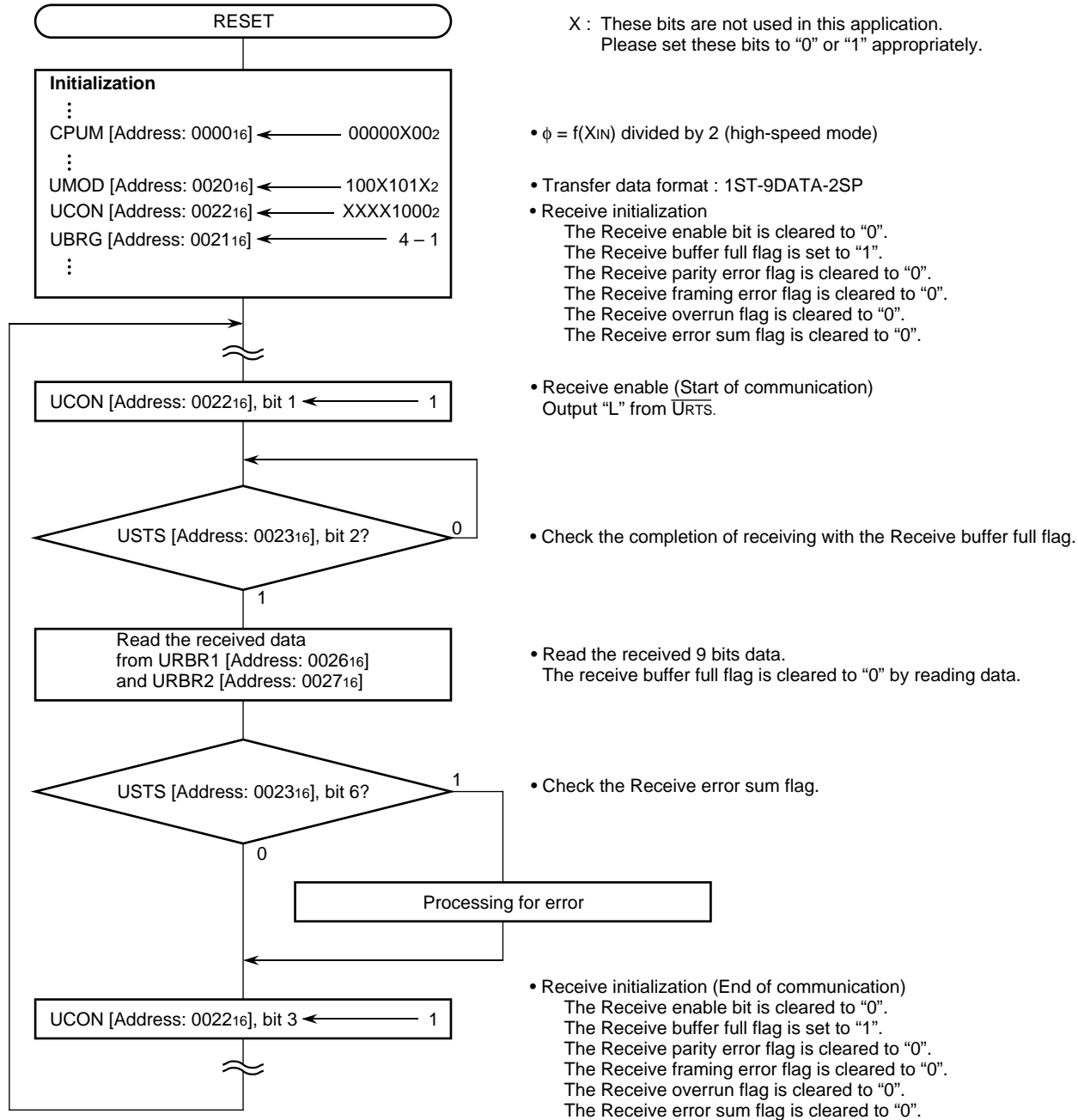


Fig. 2.5.29 Control procedure on receiving side [Communication using UART]

APPLICATION

2.6 A-D converter

2.6 A-D converter

2.6.1 Memory map of A-D conversion

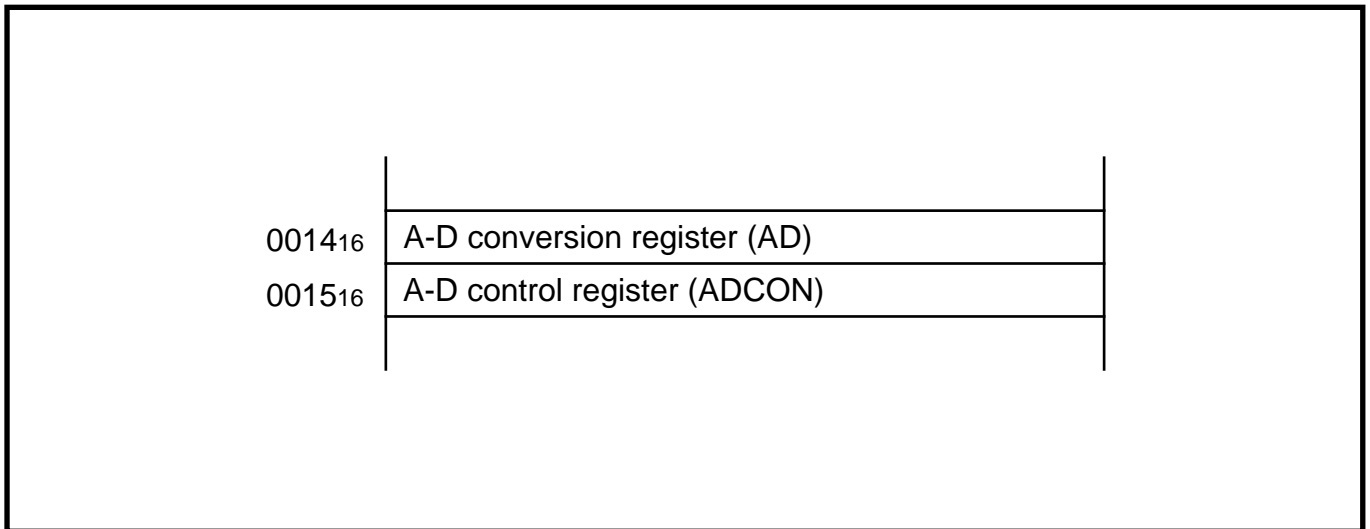


Fig. 2.6.1 Memory map of A-D conversion related registers

2.6.2 Related registers

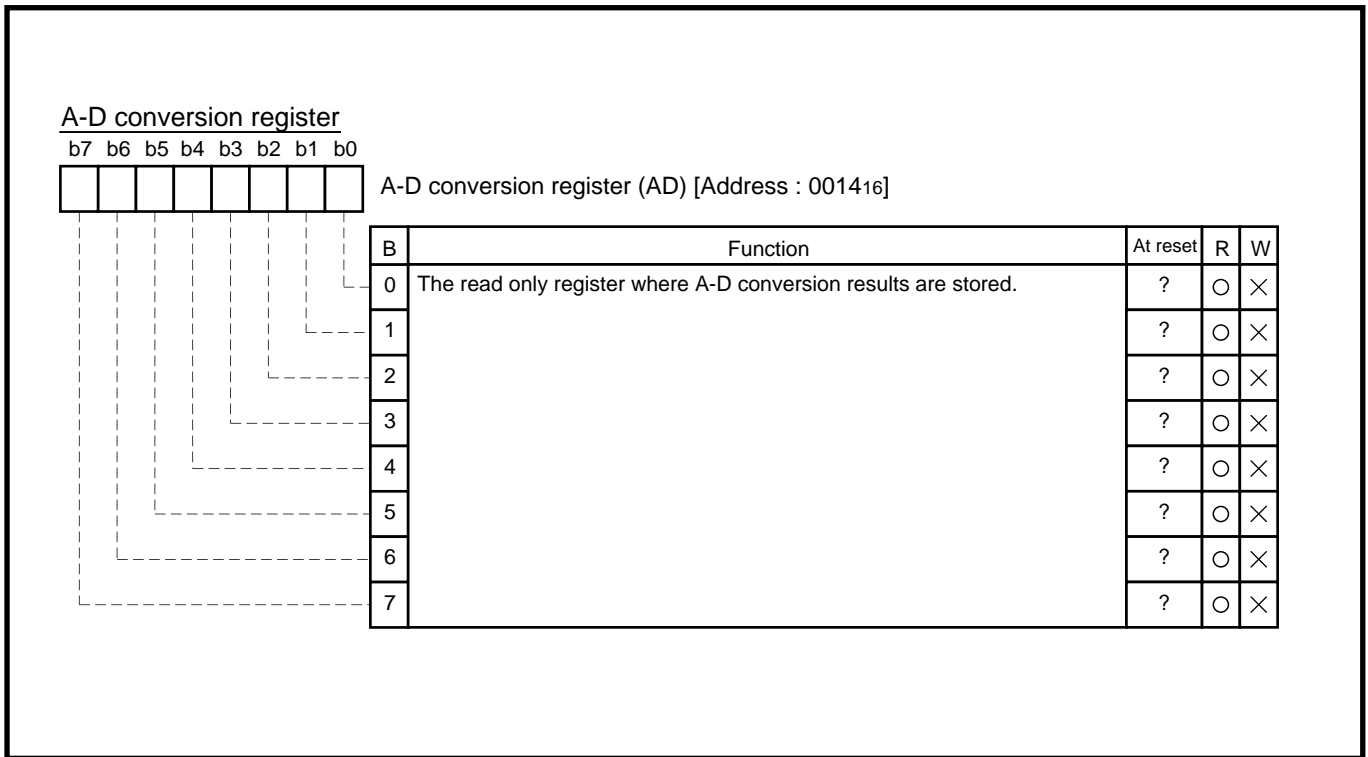


Fig. 2.6.2 Structure of A-D conversion register

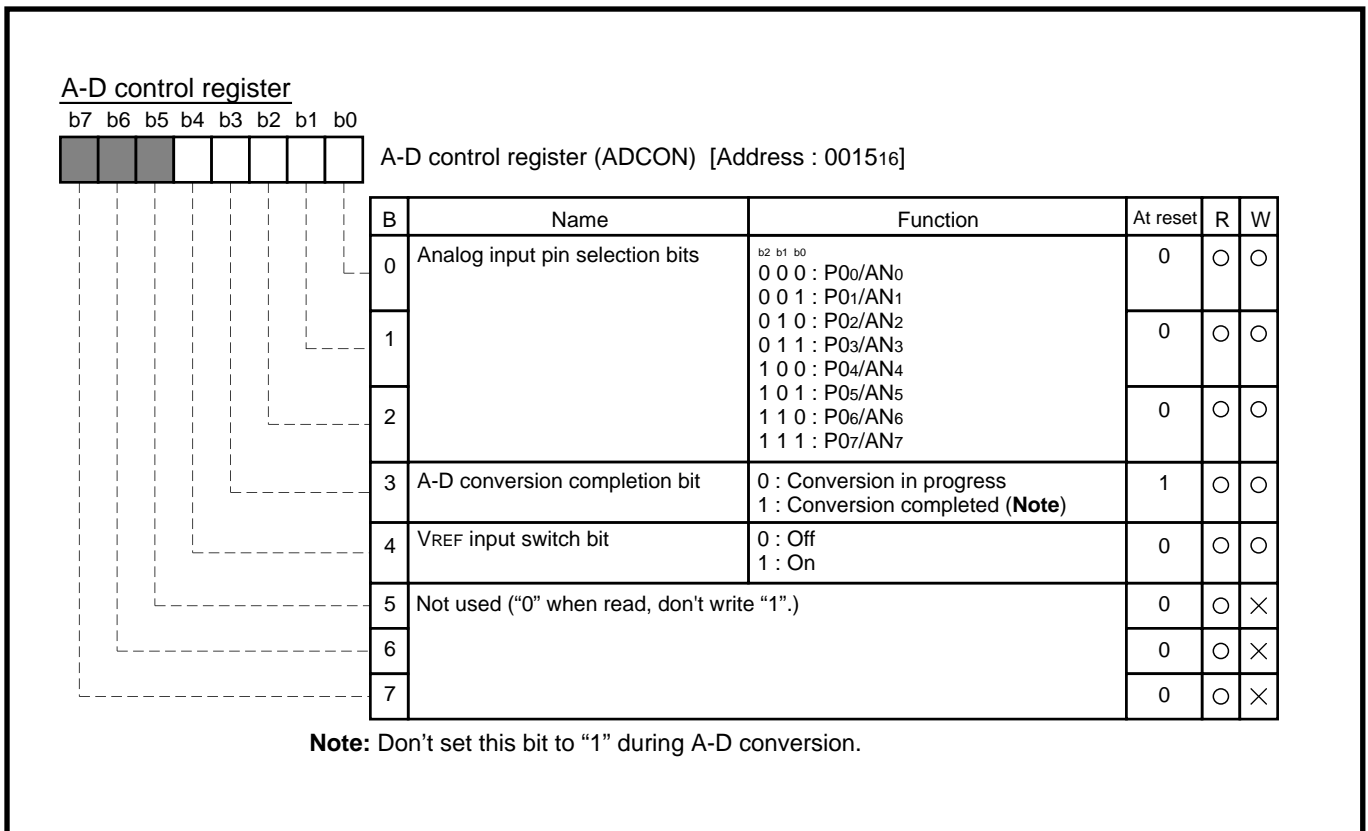


Fig. 2.6.3 Structure of A-D control register

APPLICATION

2.6 A-D converter

2.6.3 A-D conversion application example

[Measurement of analog signals]

Outline : A sensor's analog output voltage is converted to digital values.

Figure 2.6.4 shows a connection related registers.

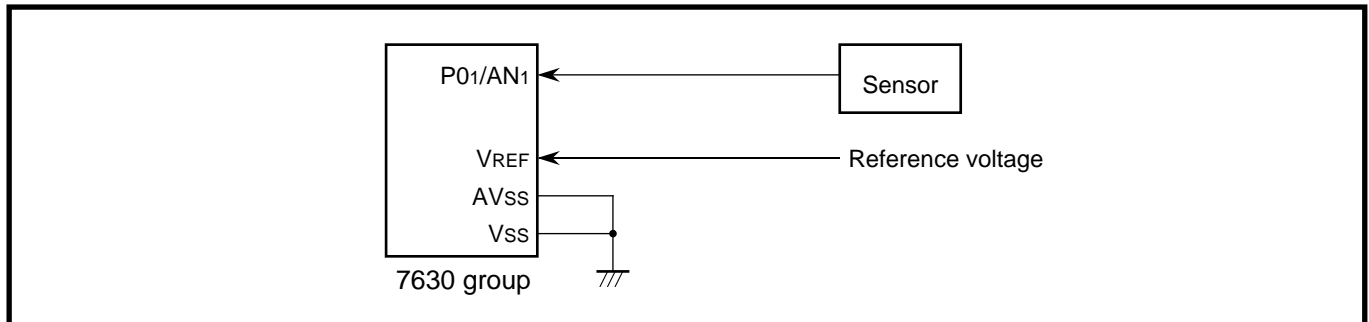


Fig. 2.6.4 Connection diagram [Measurement of analog signals]

- Specifications** :
- The analog input voltage injected from the sensor is converted into digital values (**Note**).
 - The P01/AN1 pin is used as an analog input pin.

Note: [Example] When a reference voltage, 5.12 V is input to the VREF pin and a voltage, 4 V to the P01/AN1 pin, the input voltage is converted to following value.

$$(256 / 5.12 \text{ V}) \times 4 \text{ V} = 200 \text{ (C816)}$$

Figure 2.6.5 shows a setting of related registers.

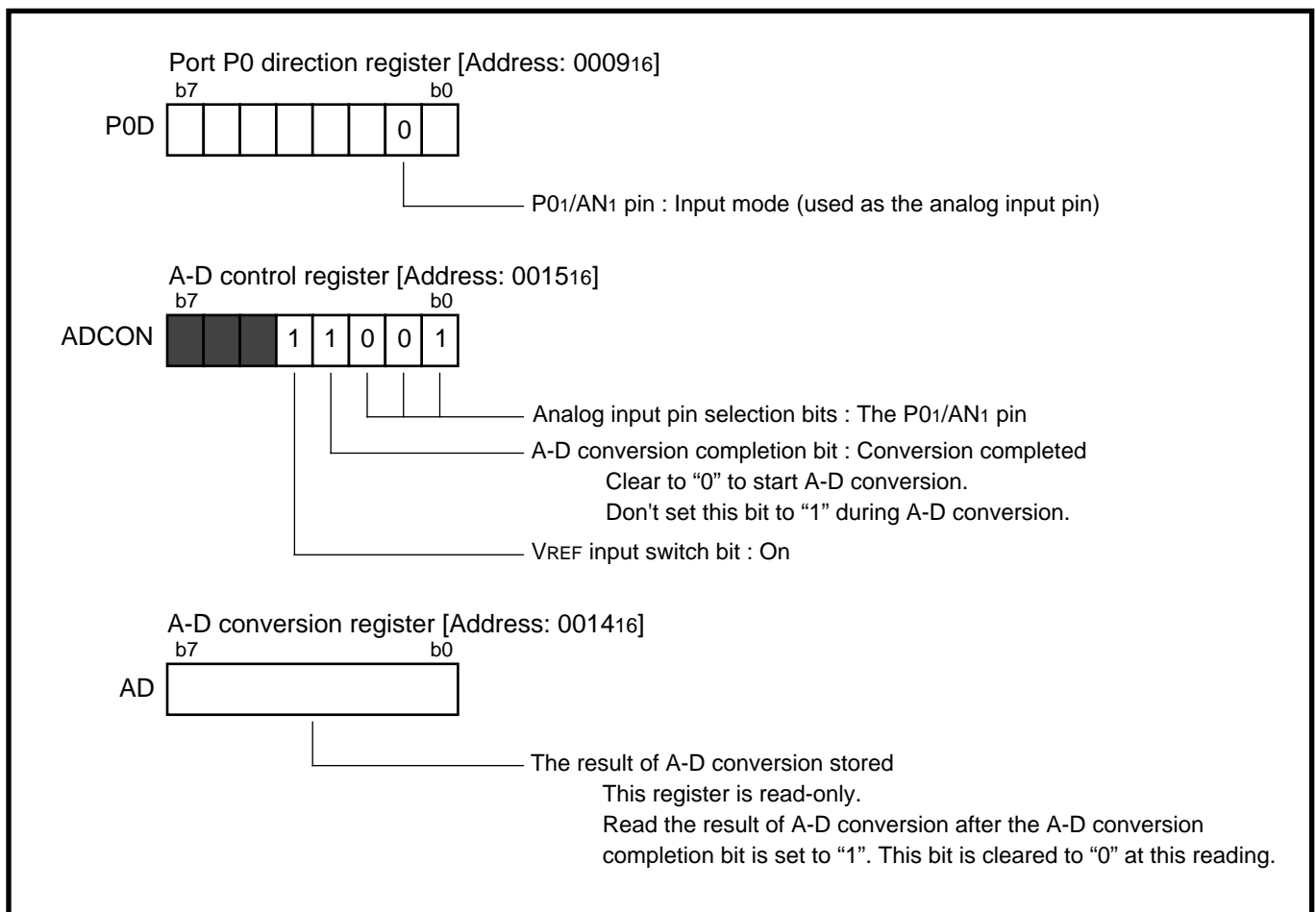


Fig. 2.6.5 Setting of related registers [Measurement of analog signals]

Control procedure : By setting the related registers as shown in Figure 2.6.6, the analog voltage input from the sensor are converted into digital values.

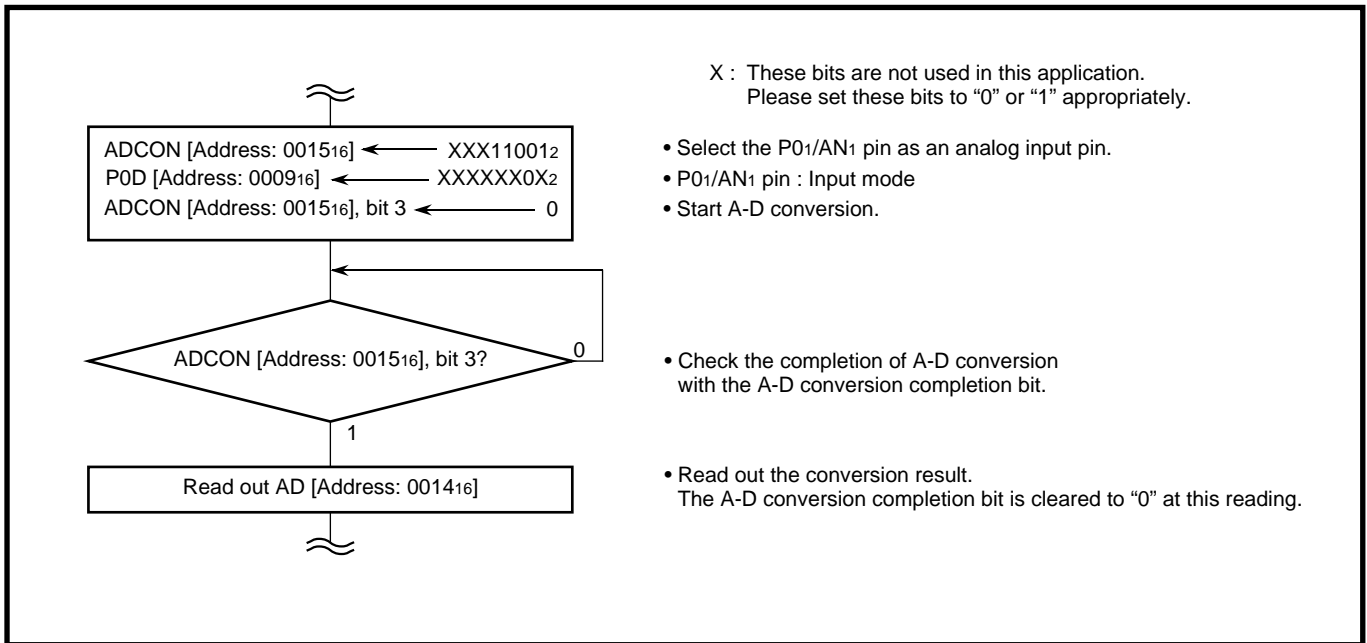


Fig. 2.6.6 Control procedure [Measurement of analog signals]

2.6.4 Conversion time

On A-D conversion process takes 53 to 54 cycles of the internal system clock ϕ .

2.6.5 Notes on use

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of $0.01\mu\text{F}$ to $1\mu\text{F}$. Further, be sure to verify the operation of application products on the user side.

Reason

An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A-D conversion precision to be worse.

(2) Reference voltage input pin (VREF)

Apply a voltage of 2V to VCC to the reference voltage input pin VREF during A-D conversion. Note that if the reference voltage is lowered below the above value, the A-D conversion precision will be degraded.

(3) Oscillation frequency during A-D conversion

The comparator is configured by capacity coupling, so the charge is lost if the clock input oscillation frequency is low. Set $f(\text{XIN})$ at 500kHz or more during A-D conversion.

(4) Set the analog input pin to input mode

Clear the bit of the Port P0 direction register [Address: 000916] which corresponds to the used analog input pin to "0" (input mode).

APPLICATION

2.7 Watchdog timer

2.7 Watchdog timer

The watchdog timer can detect a runaway program using either 7-bit or 11-bit timer prescaler.

2.7.1 Related register

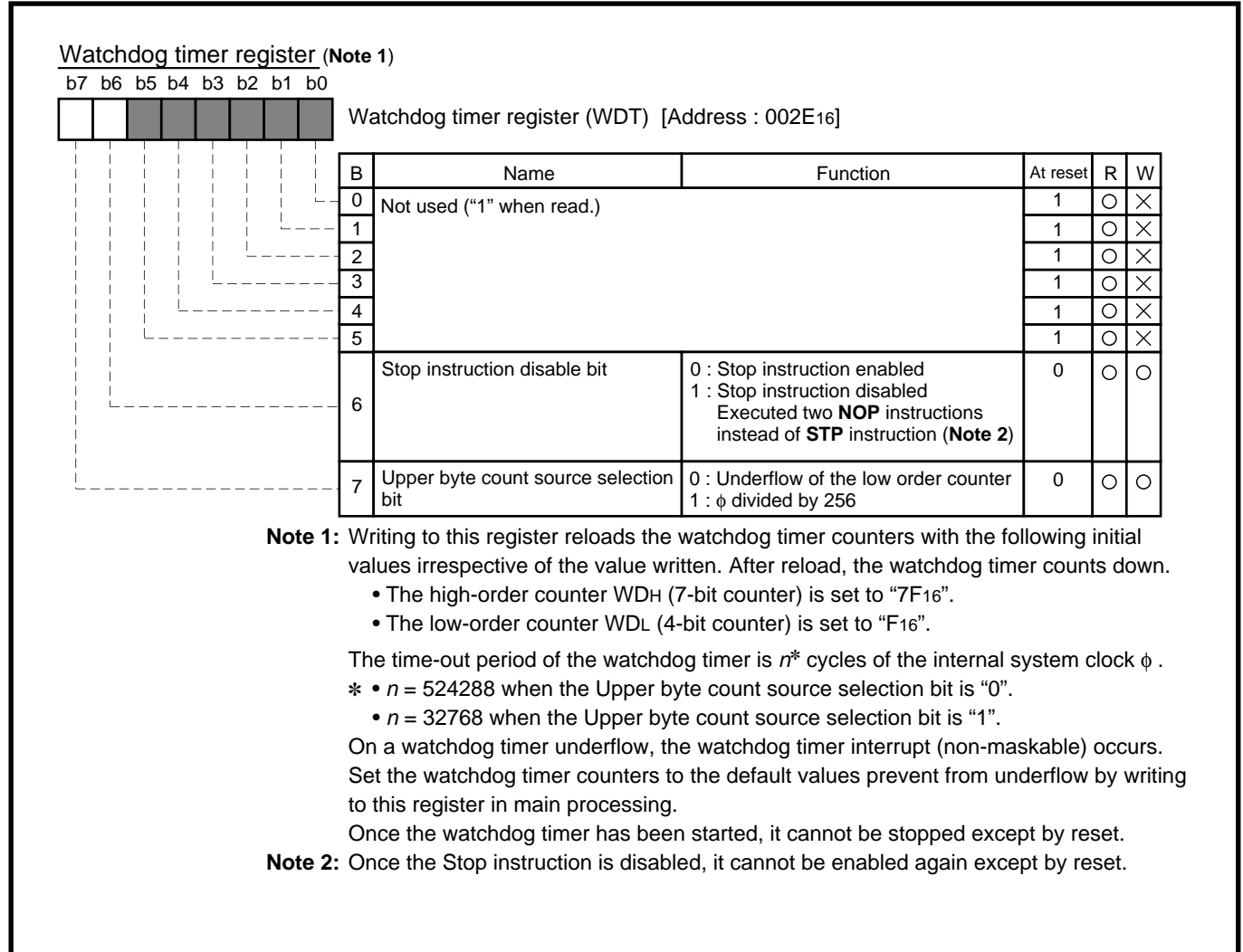


Fig. 2.7.1 Structure of Watchdog timer register

2.7.2 Watchdog timer cycle

The watchdog timer cycle varies depending on the internal clock ϕ and the frequency division ratio of the prescaler selected.

Table 2.7.1 shows the watchdog timer cycle.

Table 2.7.1 Watchdog timer cycle

f(X _{IN})	Internal clock selection bit (bit 6 of CPU mode register[address: 0000 ₁₆])	Upper byte count source selection bit (bit 7 of Watchdog timer register[address: 002E ₁₆])	Period
10 MHz	0 ($\phi = f(X_{IN})/2 = 5$ MHz)	0	Approx. 104.9 ms
		1	Approx. 6.6 ms
	1 ($\phi = f(X_{IN})/8 = 1.25$ MHz)	0	Approx. 419.4 ms
		1	Approx. 26.2 ms
8 MHz	0 ($\phi = f(X_{IN})/2 = 4$ MHz)	0	Approx. 131.1 ms
		1	Approx. 8.2 ms
	1 ($\phi = f(X_{IN})/8 = 1$ MHz)	0	Approx. 524.3 ms
		1	Approx. 32.8 ms

2.7.3 Watchdog timer procedure

Figure 2.7.2 shows the set-up procedure of watchdog timer.

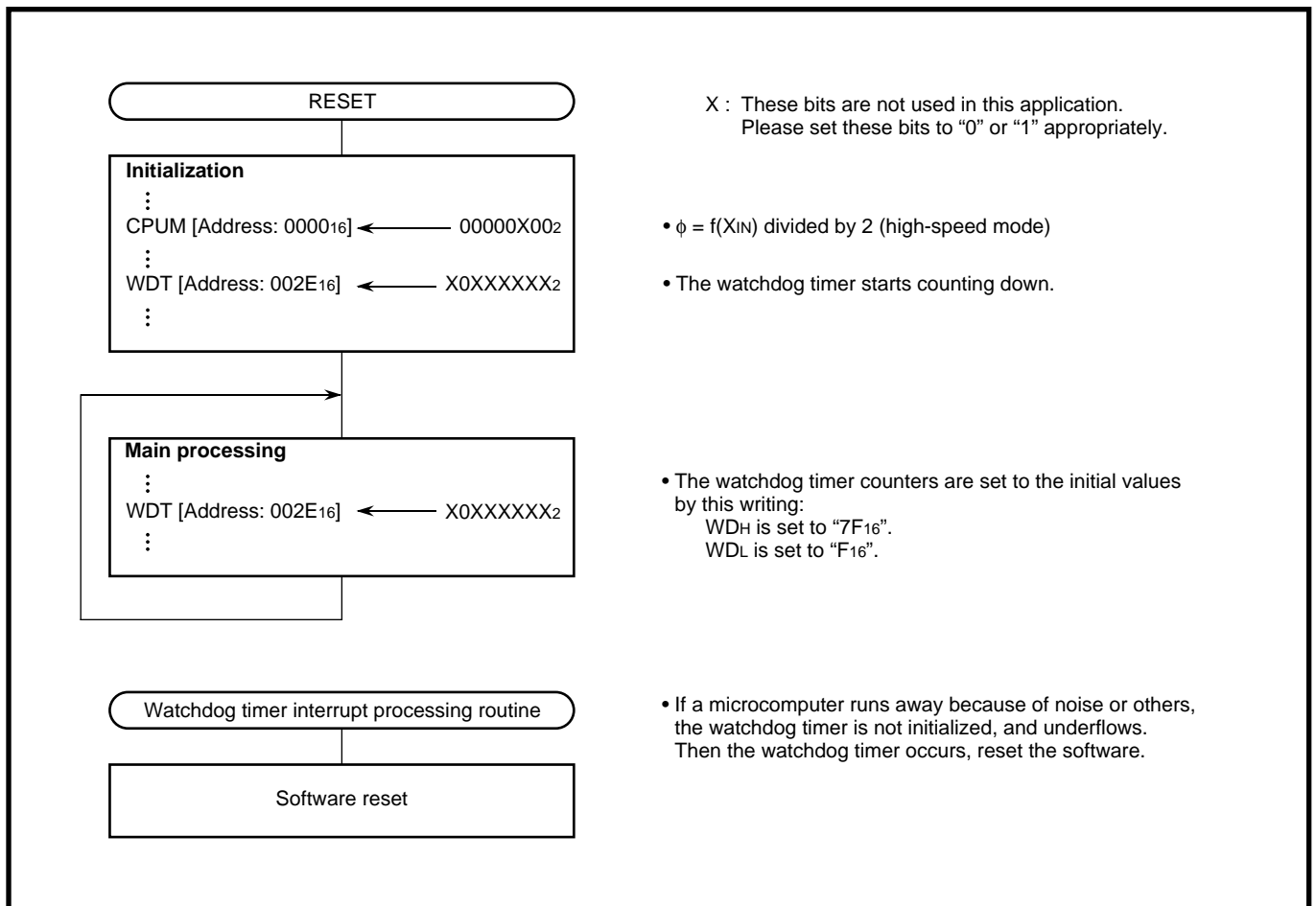


Fig. 2.7.2 Set-up procedure of watchdog timer

APPLICATION

2.8 Reset

2.8 Reset

Figure 2.8.1 shows an example of power on reset circuit.

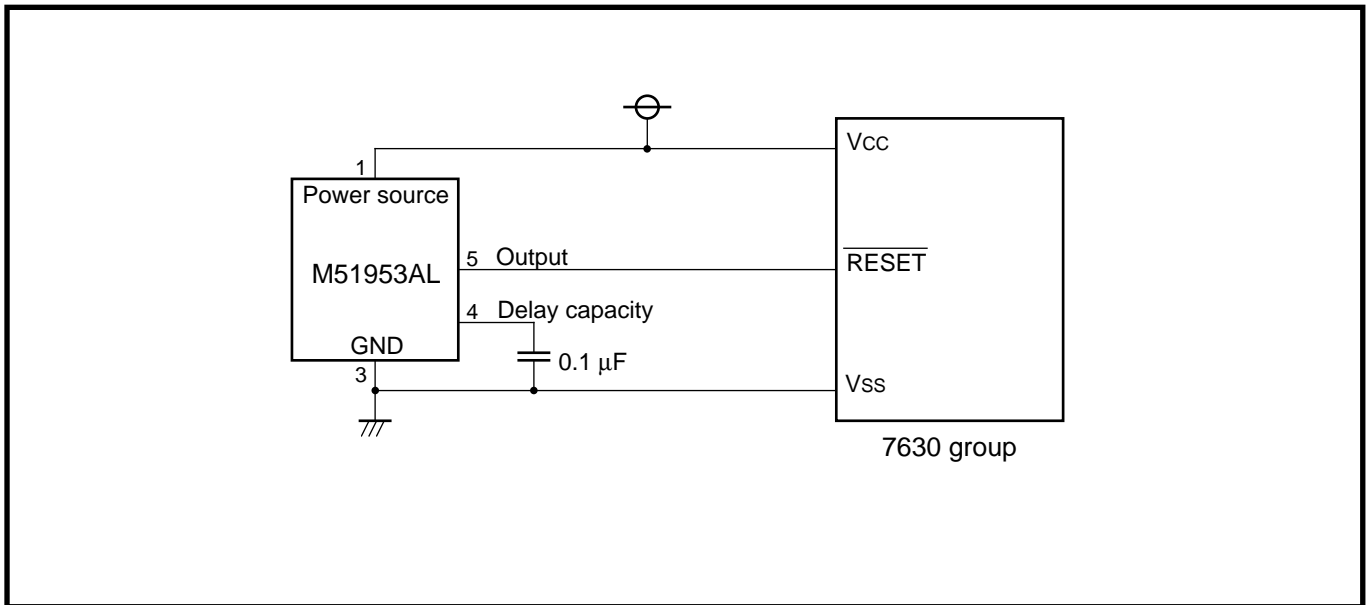


Fig. 2.8.1 Example of Power on reset circuit

Figure 2.8.2 shows system example which switch the microcomputer to the RAM backup mode by detecting a drop of the system power source voltage with the INT interrupt.

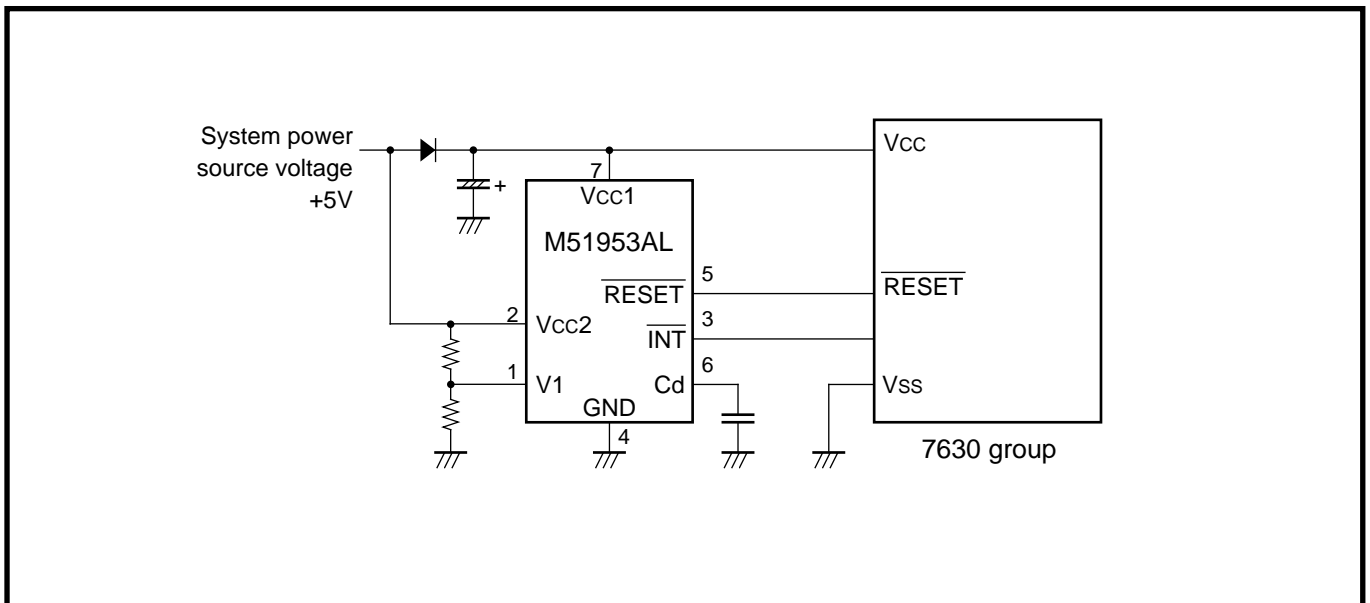


Fig. 2.8.2 RAM back-up system

2.9 Oscillation Circuit

2.9.1 Memory map of oscillation circuit related registers

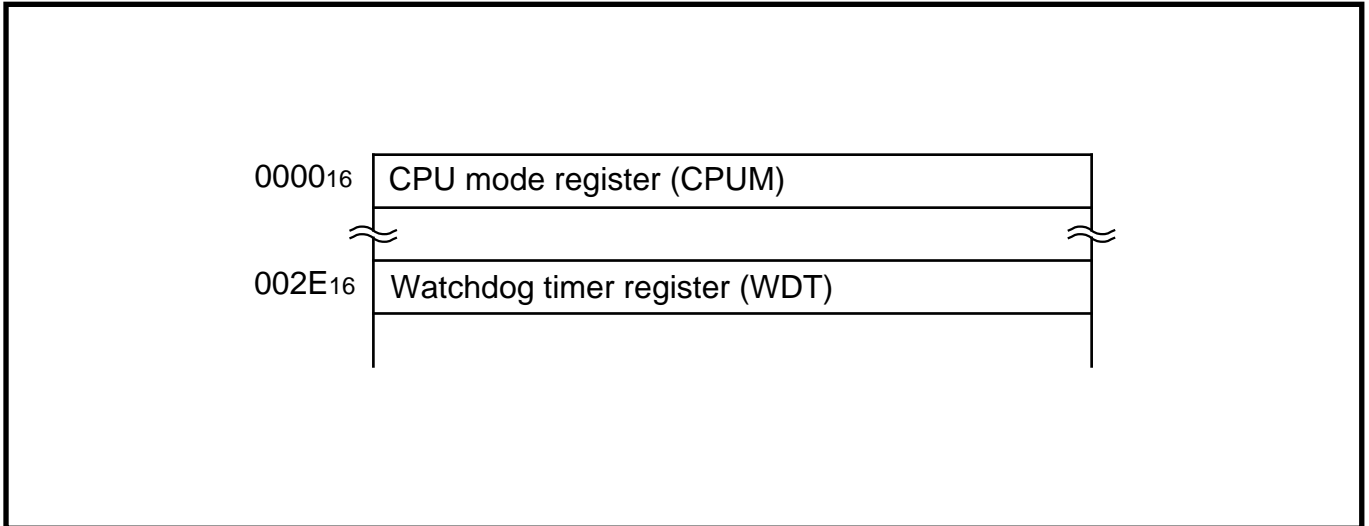


Fig. 2.9.1 Memory map of oscillation circuit related registers

APPLICATION

2.9 Oscillation circuit

2.9.2 Related registers

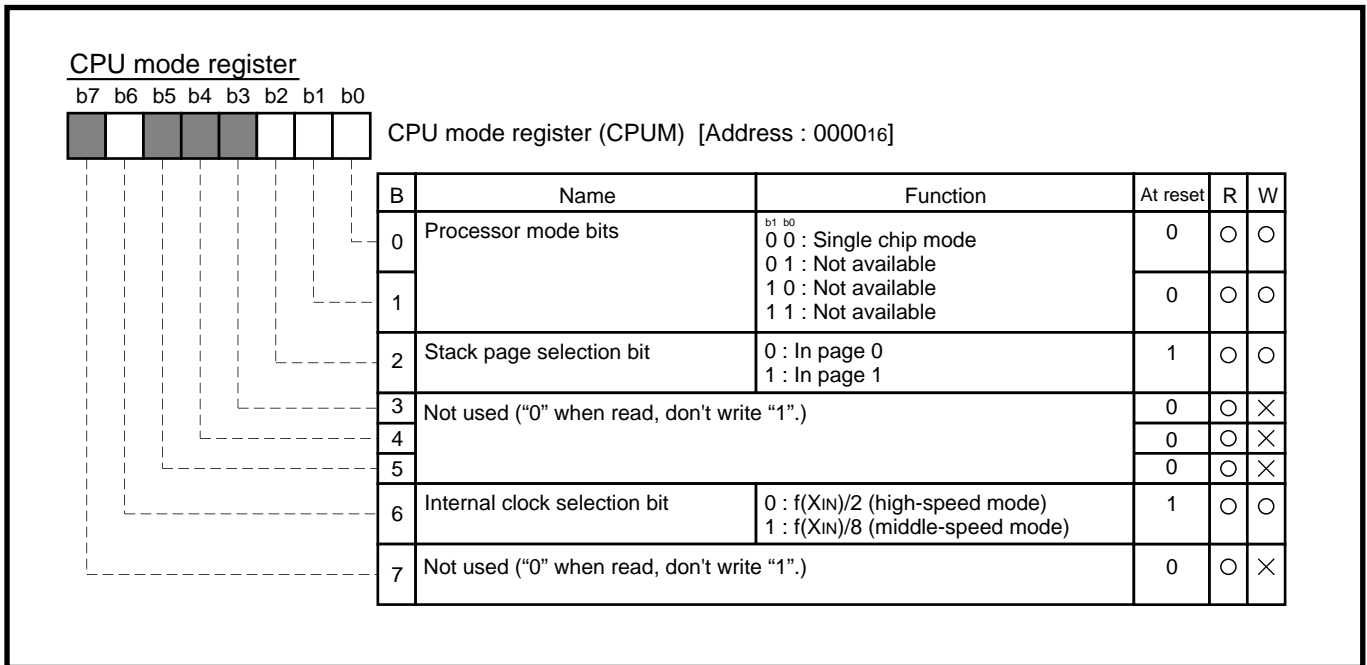


Fig. 2.9.2 Structure of CPU mode register

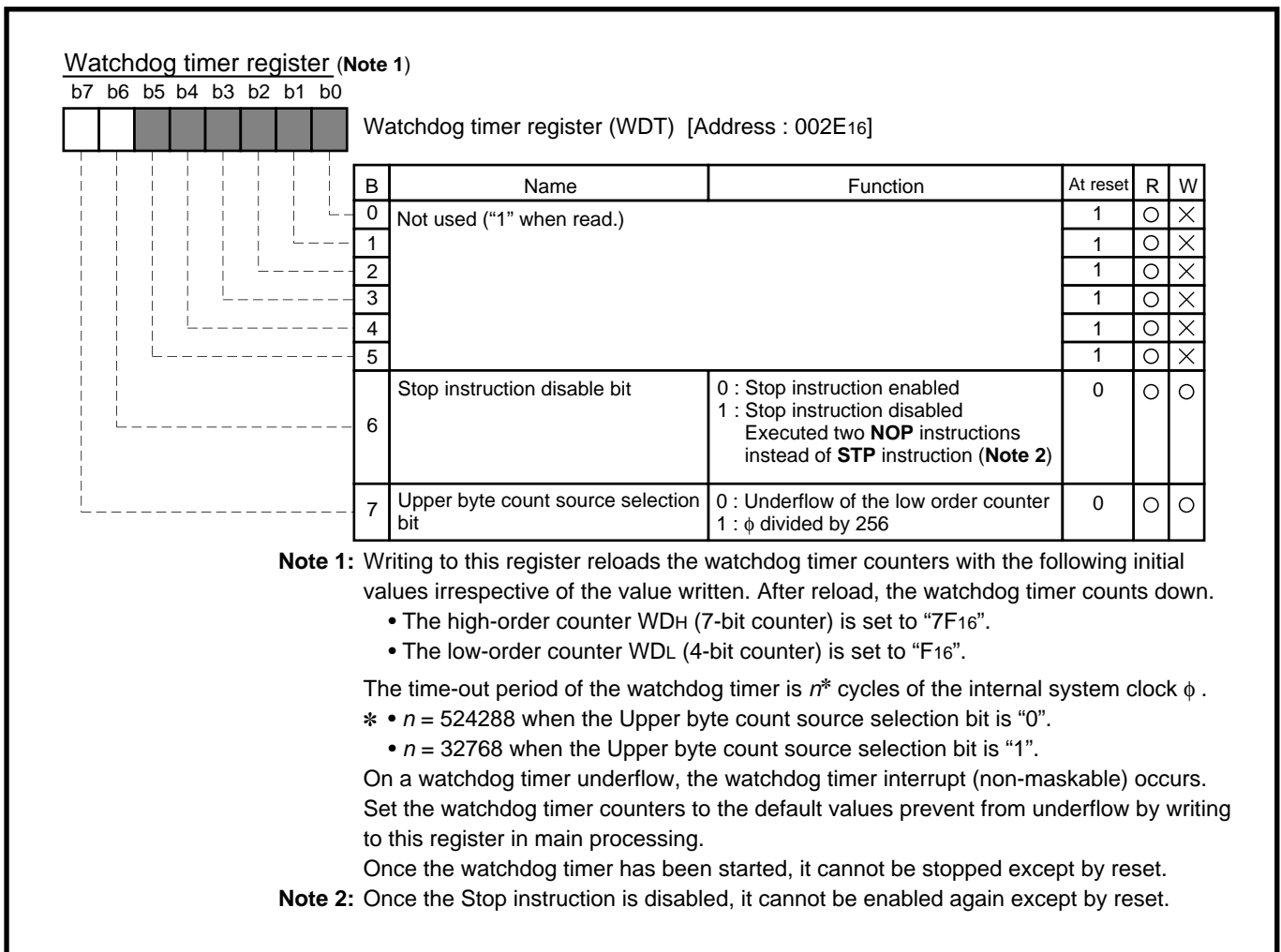


Fig. 2.9.3 Structure of Watchdog timer register

2.9.3 Application examples

As examples of application, switching procedures to Stop and Wait modes are shown below.

- (1) Ordinary mode → Stop mode
- (2) Ordinary mode → Wait mode

(1) Switching procedure from Ordinary mode to Stop mode

Figure 2.9.4 shows the switching procedure to Stop mode.

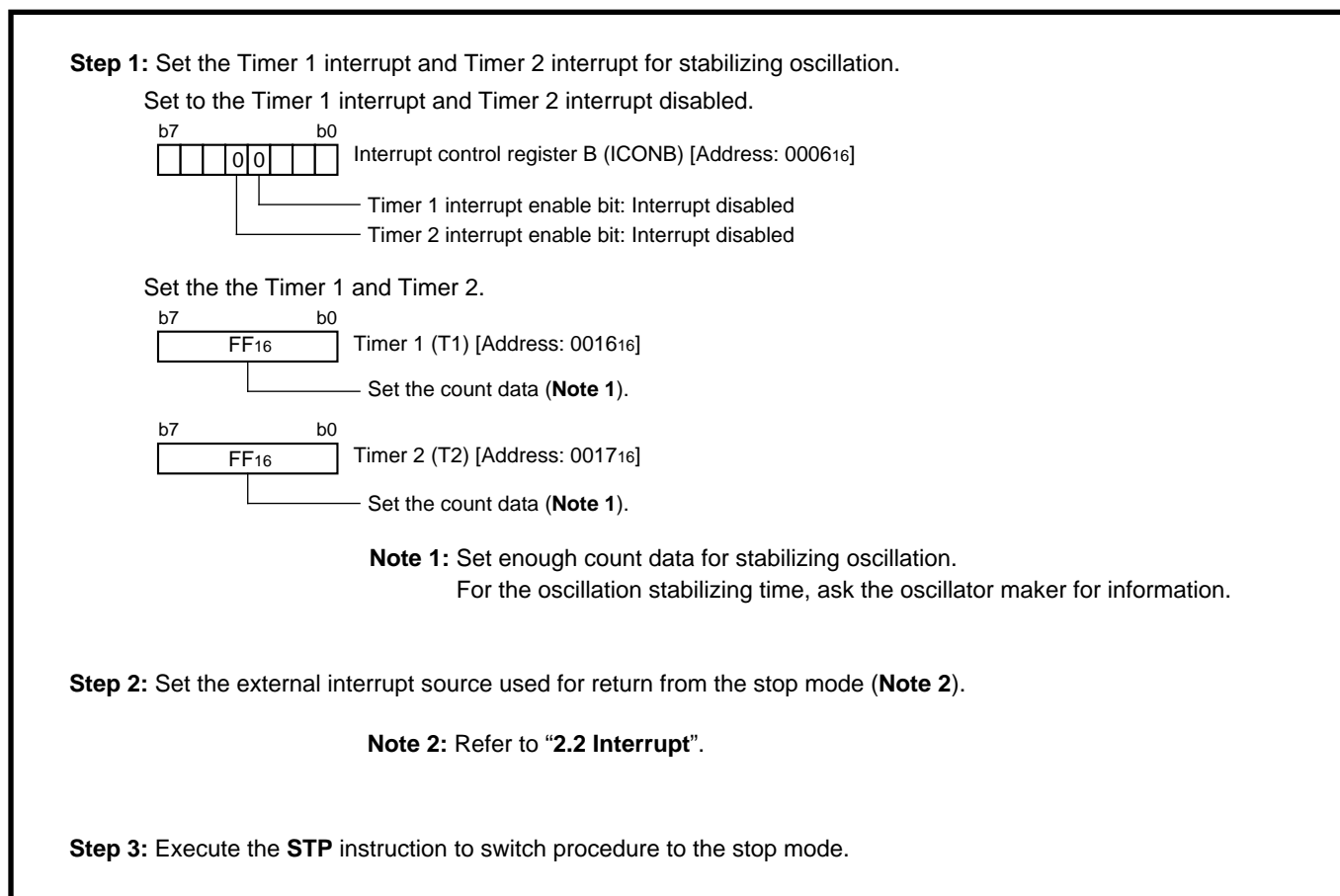


Fig. 2.9.4 Switching procedure to Stop mode

(2) Switching procedure from Ordinary mode to Wait mode

Figure 2.9.5 shows the switching procedure to Wait mode.

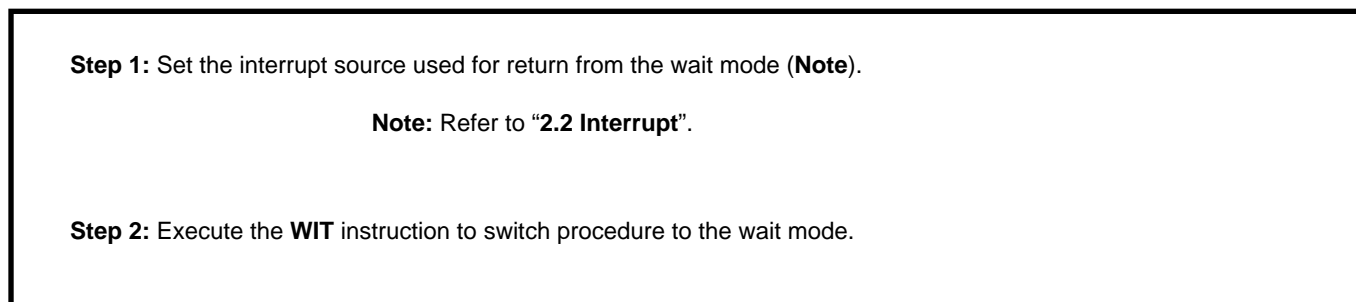


Fig. 2.9.5 Switching procedure to Wait mode

APPLICATION

2.10 Development support tools

2.10 Development support tools (M37630T-RFS)

The M37630T-RFS is a conversion board to use M37630E4FS as emulator MCU.

When an emulator is connected to the socket on the top surface, user program debugging can be performed efficiently by using a real-time trace function, etc.

Since address bus signals, data bus signals, SYNC, RD, WR and f signals are output from the socket, emulator can monitor all bus information in the microcomputer.

For details of development support systems for the M37630T-RFS, refer to the “**DEVELOPMENT SUPPORT TOOLS FOR MICROCOMPUTER**” data book.

Figure 2.10.1 shows an example of configuration example of using M37630T-RFS.

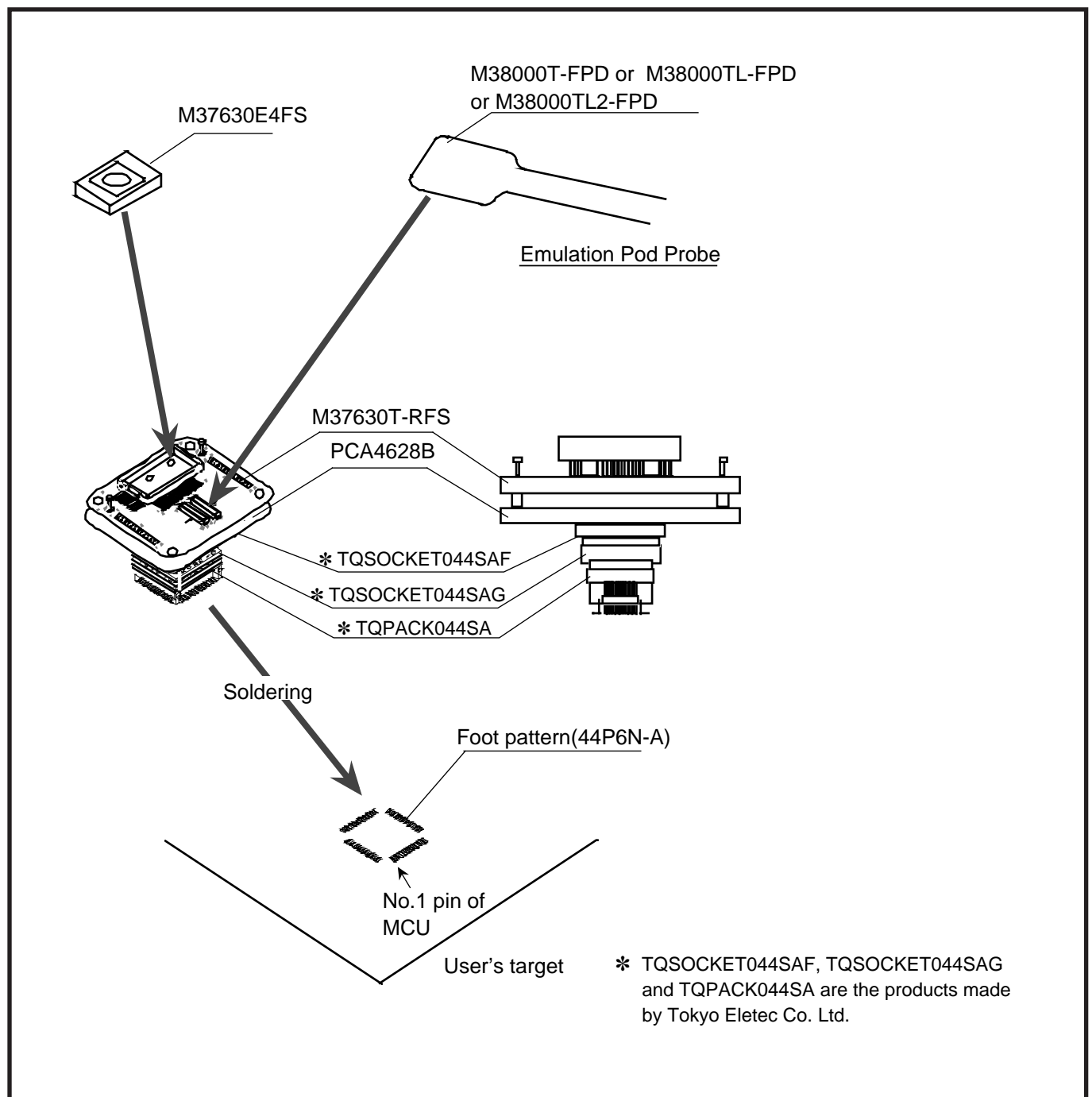


Fig. 2.10.1 Configuration example of using M37630T-RFS

2.11 Built-in PROM version

In contrast with the mask ROM version, a microcomputer incorporating a programmable ROM is called built-in PROM version.

There are two types of built-in PROM version as shown below.

- One Time PROM version

Writing to the built-in PROM can be performed only once. Neither erase nor rewrite operation is enabled.

- Built-in EPROM version

The built-in EPROM version is a programmable microcomputer with a window and can perform write, erase, and rewrite operations.

The built-in PROM version has the EPROM mode for writing to the built-in PROM in addition to the same functions as those of the mask ROM version.

For an outline of performance and a functional block diagram of the built-in PROM version, refer to “1. Hardware”.

2.11.1 Product expansion

The 7630 group supports the built-in PROM versions shown in Table 2.11.1.

Table 2.11.1 7630 group's built-in PROM version supporting products

Product name	(P)ROM size (bytes)	RAM size (bytes)	I/O Ports	Package	Remarks
M37630E4T-XXXFP	16252	512	I/O ports: 35 Input ports: 1	44P6N-A	One Time PROM version
M37630E4FP					One Time PROM version (blank)
M37630E4FS				80D0	EPROM version

APPLICATION

2.11 Built-in PROM version

2.11.2 Pin configuration

The pin configurations of the built-in PROM versions are shown in Figure 2.11.1.

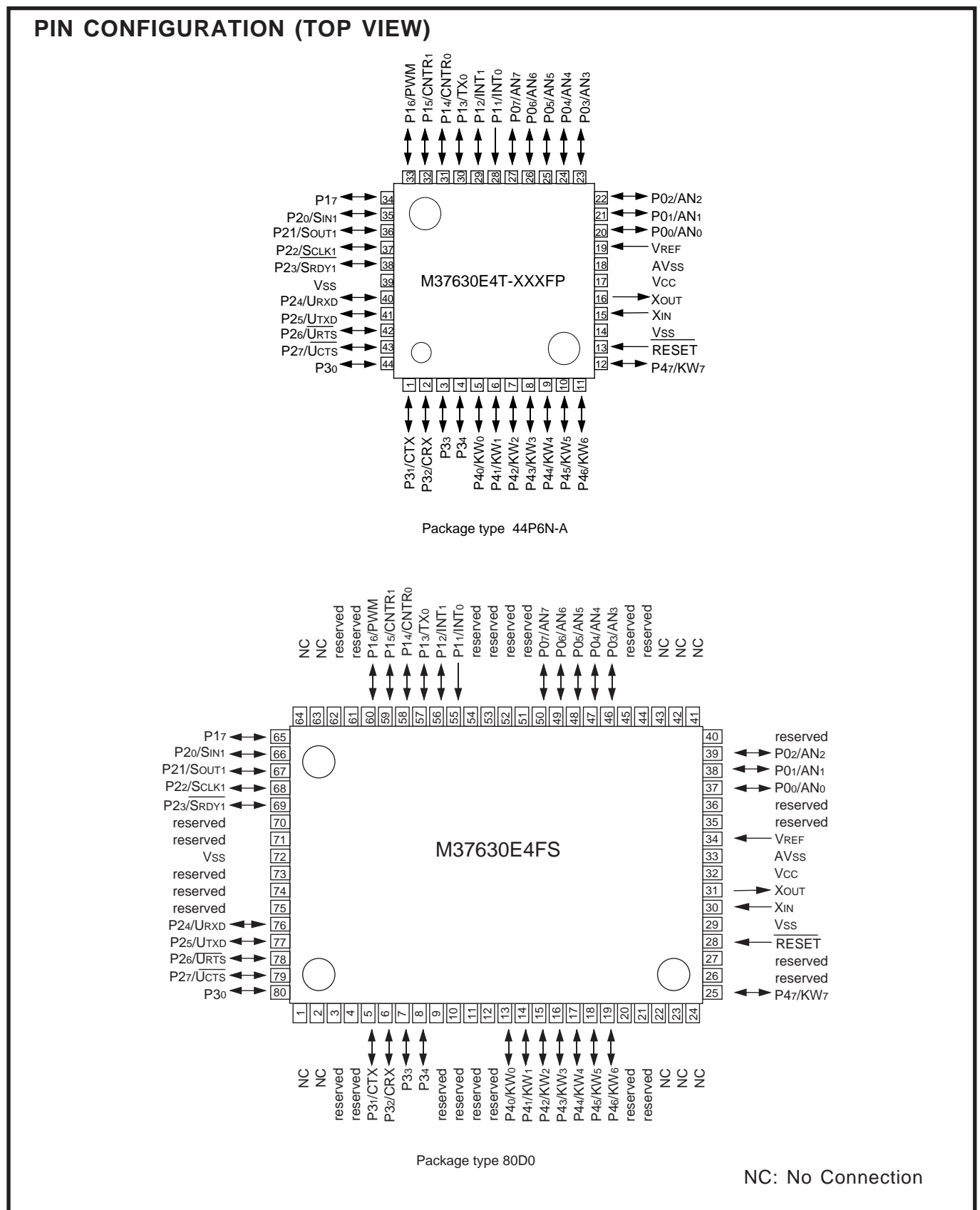


Fig. 2.11.1 Pin configuration of 7630 group's built-in PROM versions

2.11.3 Programming adapter

To write or read data into/from the internal PROM, use the dedicated programming adapter and general-purpose PROM programmer as shown in Table 2.11.2.

Table 2.11.2 Programming adapter

Microcomputer	Programming adapter
M37630E4FS	PCA7431
M37630E4FP (one-time blank)	PCA7430

(1) Write and read

In PROM mode, operation is the same as that of the M5M27C101, but programming conditions of PROM programmer are not set automatically because there are no internal device ID codes.

Accurately set the following conditions for data write/read. Take care not to apply 21 V to VPP pin, or the product may be permanently damaged.

- Programming voltage : 12.5 V
- Setting of programming adapter switch : Refer to Table 2.11.3.
- Setting of PROM programmer address : Refer to Table 2.11.4.

Table 2.11.3 Setting of programming adapter switch

Programming adapter	SW 1	SW 2
PCA7431, PCA7430	CMOS	CMOS

Table 2.11.4 Setting of PROM programmer address

Microcomputer	PROM programmer start address	PROM programmer completion address
M37630E4FS	Address : C080 ₁₆	Address : FFFB ₁₆
M37630E4FP		

(2) Erasing

Contents of the windowed EPROM are erased through an ultraviolet light source of the wavelength 2537 Angstrom. At least 15 W-sec/cm are required to erase EPROM contents.

APPLICATION

2.11 Built-in PROM version

2.11.4 Notes on use

The notes on using the built-in PROM version are shown below.

(1) All built-in PROM version products

■ Precautions at write operation

- Be careful not to apply an overvoltage to pins because a high voltage is used for a write operation. Exercise special care when turning on the power supply.
- For writing the contents of the PROM, use a dedicated programming adapter. This permits using a general-purpose PROM programmer for writing data. For details of dedicated programming adapters, refer to “2.11.3 Programming adapter”.

■ Precautions at read operation

- When reading the contents of the PROM, use a dedicated programming adapter, so that reading can be performed by a general-purpose PROM programmer. For details of dedicated programming adapters, refer to “2.11.3 Programming adapter”.

(2) One Time PROM version

■ Precautions before use

- The PROM of the One Time PROM version is not tested or screened in the assembly process and the following processes. To ensure proper operation after programming, the procedure shown in Figure 2.11.2 is recommended to verify programming.

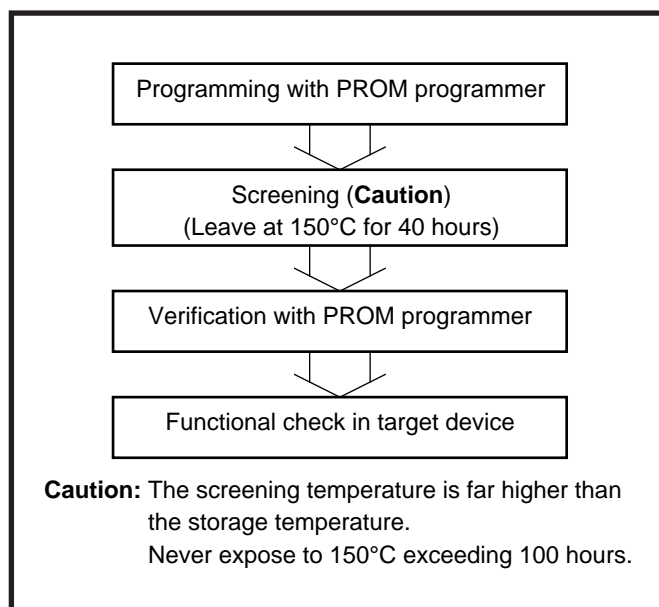


Fig. 2.11.2 Programming and testing of One Time PROM version

(3) Built-in EPROM version

■ Precautions on erasing

- Sunlight and fluorescent light include light that may erase the information written in the built-in PROM. When using the built-in EPROM version in the read mode, be sure to cover the transparent glass portion with a seal.
- This seal to cover the transparent glass portion is prepared on our side. Be careful that the seal does not touch the microcomputer lead wires when covering the glass portion with the seal because this seal is made of metal (aluminum).
- Before erasing data, clean the transparent glass. If any finger stain or seal adhesive is stuck to the transparent glass, this prevents ultraviolet rays' passing, thereby affecting the erase characteristic adversely.

■ Precautions on mounting

- M37630E4FS(Package type 80D0) has the "reserved" pins. It uses those pins at emulator MCU mode. Please open these pins described in "**2.11.2 Pin configuration**" as "reserved".



CHAPTER 3

APPENDIX

- 3.1 Electrical characteristics
- 3.2 Standard characteristics
- 3.3 Notes on use
- 3.4 Countermeasures against noise
- 3.5 List of registers
- 3.6 Mask ROM ordering method
- 3.7 Mark specification form
- 3.8 Package outline
- 3.9 List of instruction codes
- 3.10 Machine instructions
- 3.11 SFR memory map
- 3.12 Pin configuration

APPENDIX

3.1 Electrical characteristics

3.1 Electrical characteristics

3.1.1 Absolute maximum ratings

Table 3.1.1 Absolute maximum ratings

Symbol	Parameter	Conditions	Ratings	Unit
VCC	Power source voltage	All voltages with respect to VSS and output transistors are "off".	-0.3 to 7.0	V
VI	Input voltag P00–P07, P11–P17, P20–P27, P30–P34, P40–P47, RESET, XIN		-0.3 to VCC+0.3	V
VO	Output voltage P00–P07, P12–P17, P20–P27, P30–P34, P40–P47, XOUT		-0.3 to VCC+0.3	V
Pd	Power dissipation	Ta = 25°C	500	mW
Topr	Operating temperature		-40 to 85	°C
Tstg	Storage temperature		-60 to 150	°C

3.1.2 Recommended operating conditions

Table 3.1.2 Recommended operating conditions

(V_{CC} = 4.0V to 5.5V, V_{SS} = AV_{SS} = 0V, T_a = -40°C to 85°C, unless otherwise noted)

Symbol	Parameter		Limits			Unit
			Min.	Typ.	Max.	
V _{CC}	Power source voltage		4	5	5.5	V
V _{SS}				0		V
V _{IH}	"H" input voltage	P00-P07, P11-P17, P20-P27, P30-P34, P40-P47, RESET, X _{IN}	0.8V _{CC}		V _{CC}	V
V _{IL}	"L" input voltage	P00-P07, P11-P17, P20-P27, P30-P34, P40-P47, RESET, X _{IN}	0		0.2V _{CC}	V
∑I _{OH(peak)}	"H" sum peak output current	P00-P07, P12-P17, P20-P27, P30-P34, P40-P47			-80	mA
∑I _{OH(avg)}	"H" sum average output current				-40	mA
∑I _{OL(peak)}	"L" sum peak output current				80	mA
∑I _{OL(avg)}	"L" sum average output current				40	mA
I _{OH(peak)}	"H" peak output current				-10	mA
I _{OH(avg)}	"H" average output current				-5	mA
I _{OL(peak)}	"L" peak output current				10	mA
I _{OL(avg)}	"L" average output current				5	mA
I _{IO}	Input current at overvoltage condition (V _I > V _{CC})	P11-P17, P20-P27, P30-P34, P40-P47			1	mA
∑I _{IO}	Total input current at overvoltage condition (V _I > V _{CC})	P11-P17, P20-P27, P30-P34, P40-P47			16	mA
f(CNTR)	Timer input frequency (based on 50% duty)	P14/CNTR ₀ , P15/CNTR ₁ (except bi-phase counter mode)			f(X _{IN})/16	MHz
		P13/TX ₀ , P14/CNTR ₀ (bi-phase counter mode)			f(X _{IN})/32	MHz
f(X _{IN})	Clock input oscillation frequency				10	MHz

APPENDIX

3.1 Electrical characteristics

3.1.3 Electrical characteristics

Table 3.1.3 Electrical characteristics

(V_{CC} = 4.0V to 5.5V, V_{SS} = AV_{SS} = 0V, T_a = -40°C to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
V _{OH}	"H" output voltage P00–P07, P12–P17, P20–P27, P30–P34, P40–P47	I _{OH} = -5mA	0.8 • V _{CC}			V
V _{OL}	"L" output voltage P00–P07, P12–P17, P20–P27, P30–P34, P40–P47	I _{OL} = 5mA			2	V
V _{T+} –V _{T-}	Hysteresis P11/INT0, P12/INT1, P13/TX0, P14/CNTR0, P15/CNTR1, P20/SIN, P22/SCLK, P26/URTS, P27/UCTS, P32/CRX, RESET			0.5		V
I _{IH}	"H" input current P00–P07, P11–P17, P20–P27, P30–P34, P40–P47, RESET	V _I = V _{CC}			5	μA
I _{IH}	"H" input current XIN	V _I = V _{CC}		4		μA
I _{IH}	"H" input current P32, P40–P47,	V _I = V _{SS} Pull-Down="On"	20		200	μA
I _{IL}	"L" input current P00–P07, P11–P17, P20–P27, P30–P34, P40–P47, RESET	V _I = V _{SS}			-5	μA
I _{IL}	"L" input current XIN	V _I = V _{SS}		-4		μA
I _{IL}	"L" input current P00–P07,P11–P17, P20–P27,P30–P34, P40–P47,RESET	V _I = V _{SS} Pull-Up="On"	-200		-20	μA
V _{RAM}	RAM hold voltage	When clock stopped	2			V
I _{CC}	Power source current	High-speed mode, f(XIN) = 8MHz, V _{CC} = 5V, Output transistors "off", CAN module running, ADC running		11	18	mA
		High-speed mode, f(XIN) = 8MHz, V _{CC} = 5V, Output transistors "off", CAN module stopped, ADC running		9	16	mA
		Middle-speed mode, f(XIN) = 8MHz, V _{CC} = 5V, Output transistors "off", CAN module running, ADC running,		6	11	mA
		Middle-speed mode,wait mode, f(XIN) = 8MHz, V _{CC} = 5V, Output transistors "off", CAN module stopped, ADC running		2		mA
		Stop mode,f(XIN)=0MHz, T _a = 25°C, V _{CC} = 5V		0.1	1	μA
		Stop mode,f(XIN)=0MHz, T _a = 85°C, V _{CC} = 5V			10	μA

3.1.4 A-D converter characteristics

Table 3.1.4 A-D converter characteristics

(V_{CC} = 4.0V to 5.5V, V_{SS} = AV_{SS} = 0V, T_a = -40°C to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
–	Resolution				8	Bits
–	Absolute accuracy			±1	±2.5	LSB
t _{CONV}	Conversion time	High- speed mode	106		108	t _c (X _{IN})
		Middle-speed mode	424		432	t _c (X _{IN})
V _{REF}	Reference input voltage		2.0		V _{CC}	V
I _{REF}	Reference input current	V _{CC} = V _{REF} = 5.12V		150	200	μA
RLADDER	Ladder resistor value			35		kΩ
I _{IAN}	Analog input current	V _I = V _{SS} to V _{CC}		0.5	5	μA

3.1.5 Timing requirements

Table 3.1.5 Timing requirements

(V_{CC} = 4.0V to 5.5V, V_{SS} = AV_{SS} = 0V, T_a = -40°C to 85°C, unless otherwise noted)

Symbol	Parameter	Limits			Unit
		Min.	Typ.	Max.	
t _w (RESET)	Reset input "L" pulse width	2			μs
t _c (X _{IN})	External clock input cycle time	100			ns
t _{WH} (X _{IN})	External clock input "H" pulse width	37			ns
t _{WL} (X _{IN})	External clock input "L" pulse width	37			ns
t _c (CNTR)	CNTR ₀ , CNTR ₁ input cycle time (except bi-phase counter mode)	1600			ns
	CNTR ₀ , input cycle time (bi-phase counter mode)	2000			ns
t _{WH} (CNTR)	CNTR ₀ , CNTR ₁ input "H" pulse width (except bi-phase counter mode)	800			ns
	CNTR ₀ , input "H" pulse width (bi-phase counter mode)	1000			ns
t _{WL} (CNTR)	CNTR ₀ , CNTR ₁ input "L" pulse width (except bi-phase counter mode)	800			ns
	CNTR ₀ , input "L" pulse width (bi-phase counter mode)	1000			ns
t _L (CNTR ₀ -TX ₀)	Lag of CNTR ₀ and TX ₀ input edges (bi-phase counter mode)	500			ns
t _c (TX ₀)	TX ₀ input cycle time (bi-phase counter mode)	3200			ns
t _{WH} (TX ₀)	TX ₀ input "H" pulse width (bi-phase counter mode)	1600			ns
t _{WL} (TX ₀)	TX ₀ input "L" pulse width (bi-phase counter mode)	1600			ns
t _{WH} (INT)	INT ₀ , INT ₁ input "H" pulse width	460			ns
t _{WL} (INT)	INT ₀ , INT ₁ input "L" pulse width	460			ns
t _c (SCLK)	Serial I/O clock input cycle time	8t _c (X _{IN})			ns
t _{WH} (SCLK)	Serial I/O clock input "H" pulse width	4t _c (X _{IN})			ns
t _{WL} (SCLK)	Serial I/O clock input "L" pulse width	4t _c (X _{IN})			ns
t _{su} (SIN-SCLK)	Serial I/O clock input set up time	200			ns
t _H (SCLK-SIN)	Serial I/O clock input hold time	150			ns

APPENDIX

3.1 Electrical characteristics

3.1.6 Switching characteristics

Table 3.1.6 Switching characteristics

(Vcc = 4.0V to 5.5V, Vss = AVss = 0V, Ta = -40°C to 85°C, unless otherwise noted)

Symbol	Parameter	Test conditions	Limits			Unit
			Min.	Typ.	Max.	
t _{WH} (SCLK)	Serial I/O clock output "H" pulse width	Fig. 3.1.1	0.5 • tc(SCLK) – 50			ns
t _{WL} (SCLK)	Serial I/O clock output "L" pulse width		0.5 • tc(SCLK) – 50			ns
t _D (SCLK–SOUT)	Serial I/O output delay time				50	ns
t _V (SCLK–SOUT)	Serial I/O output valid time		0		50	ns
t _R (SCLK)	Serial I/O clock output rise time				50	ns
t _R (CMOS)	CMOS output rise time			10	50	ns
t _F (CMOS)	CMOS output fall time			10	50	ns

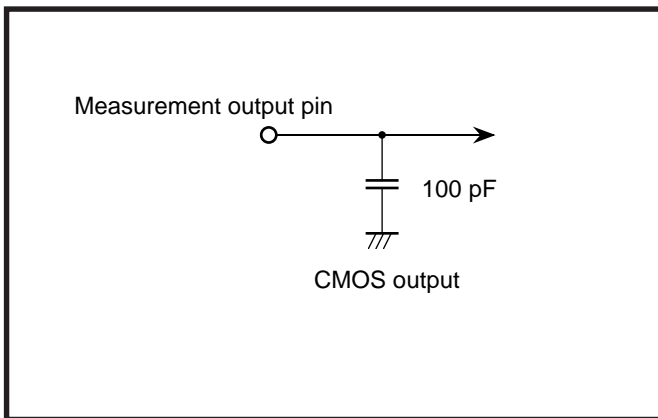


Fig. 3.1.1 Circuit for measuring output switching characteristics

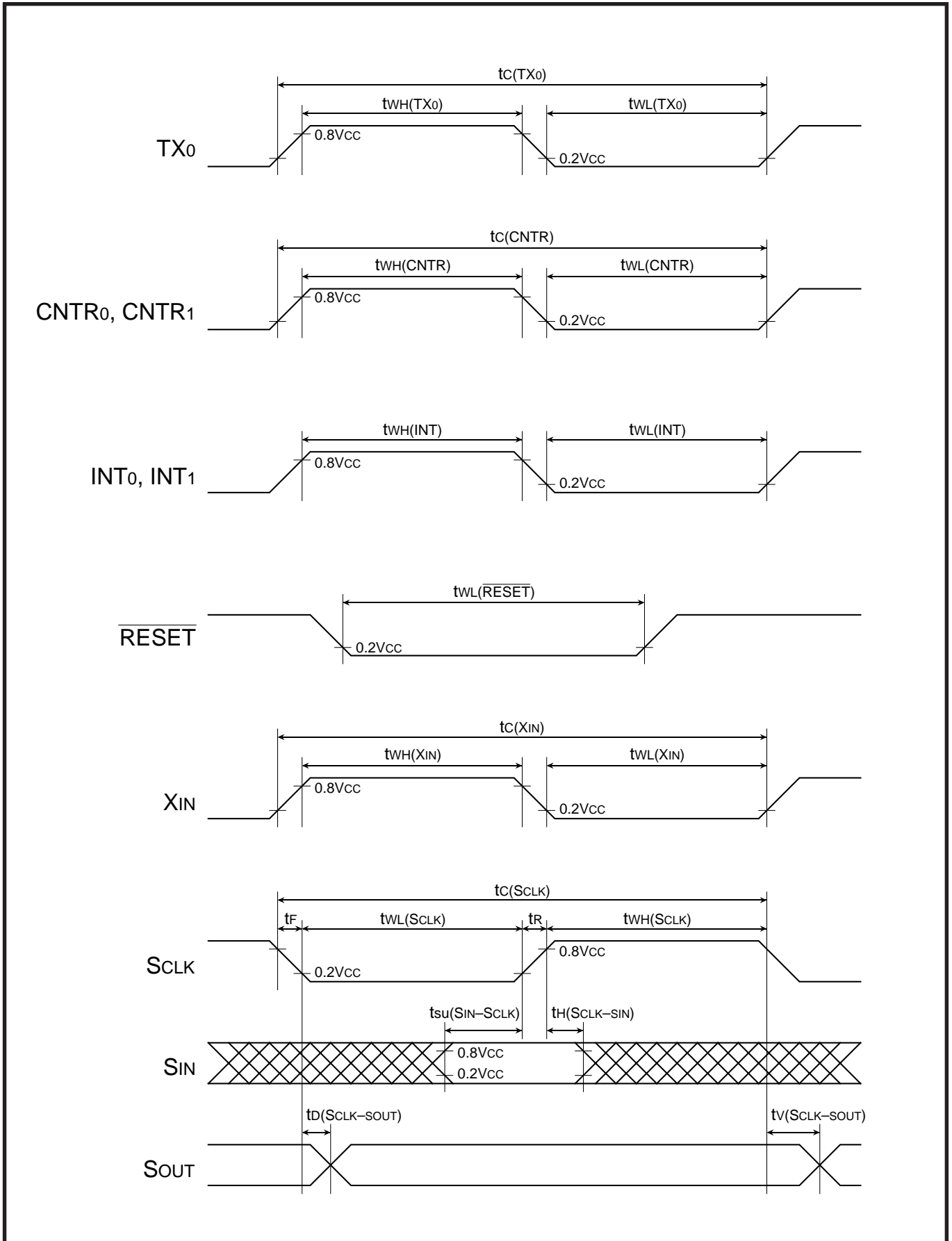


Fig. 3.1.2 Timing diagram

APPENDIX

3.2 Standard characteristics

3.2 Standard characteristics

3.2.1 Power source current standard characteristics

Figure 3.2.1 to Figure 3.2.4 show the power source current standard characteristics vs. V_{CC} and $f(X_{IN})$ is both high and middle speed mode.

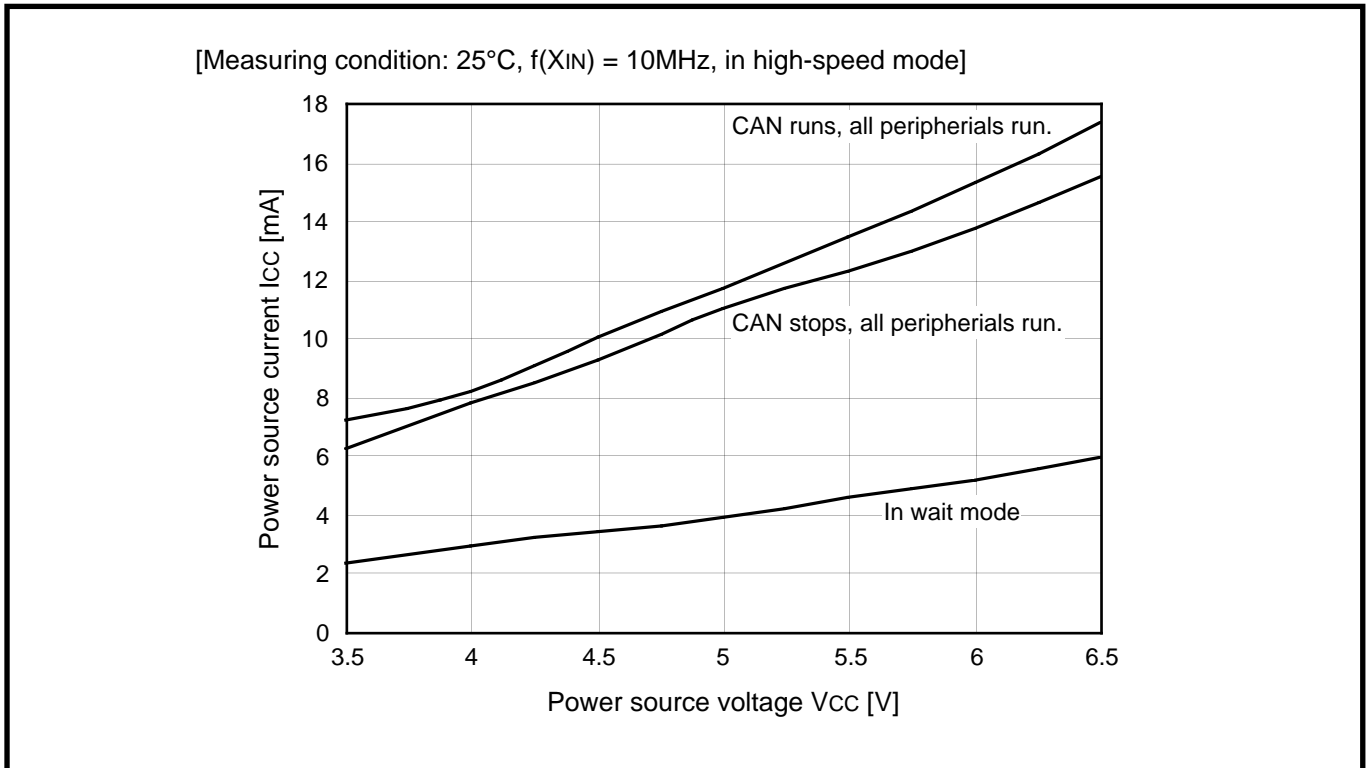


Fig. 3.2.1 I_{CC} - V_{CC} standard characteristics (in high-speed mode)

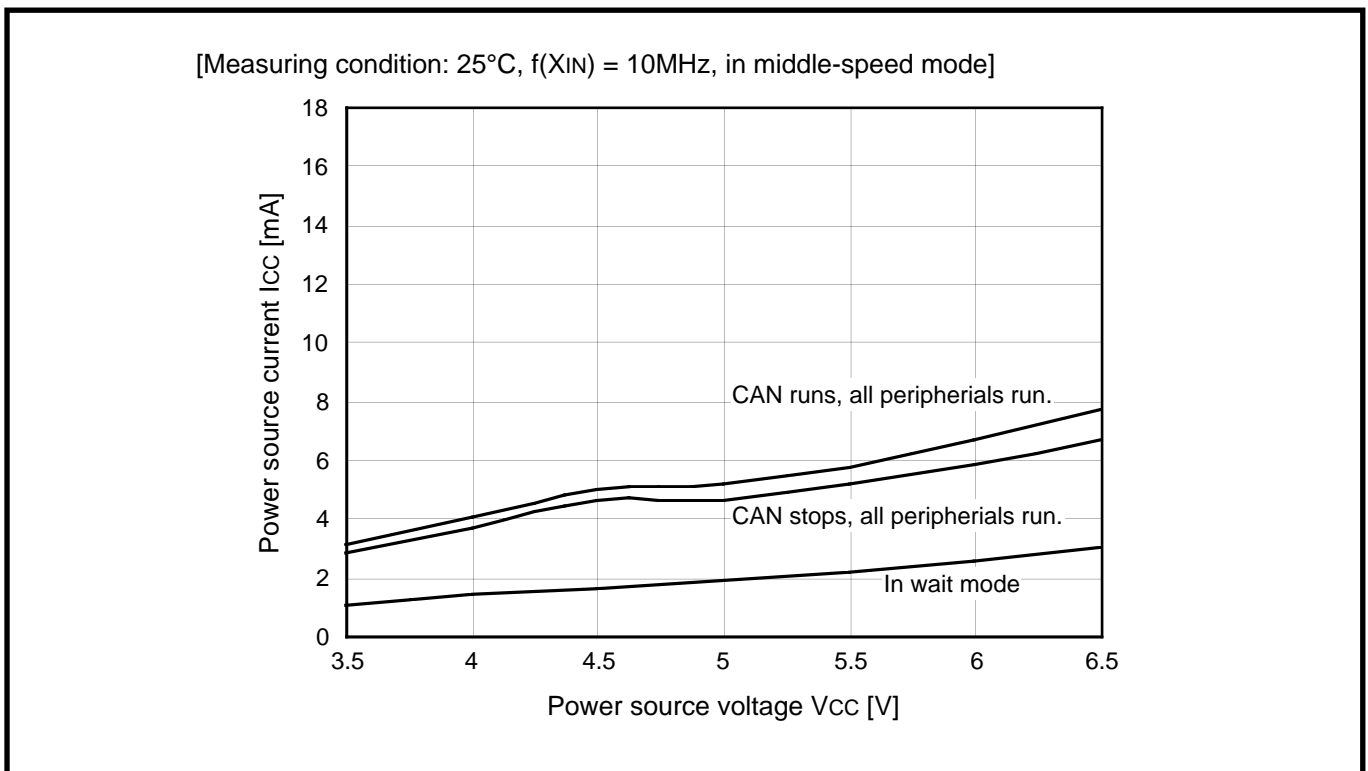


Fig. 3.2.2 I_{CC} - V_{CC} standard characteristics (in middle-speed mode)

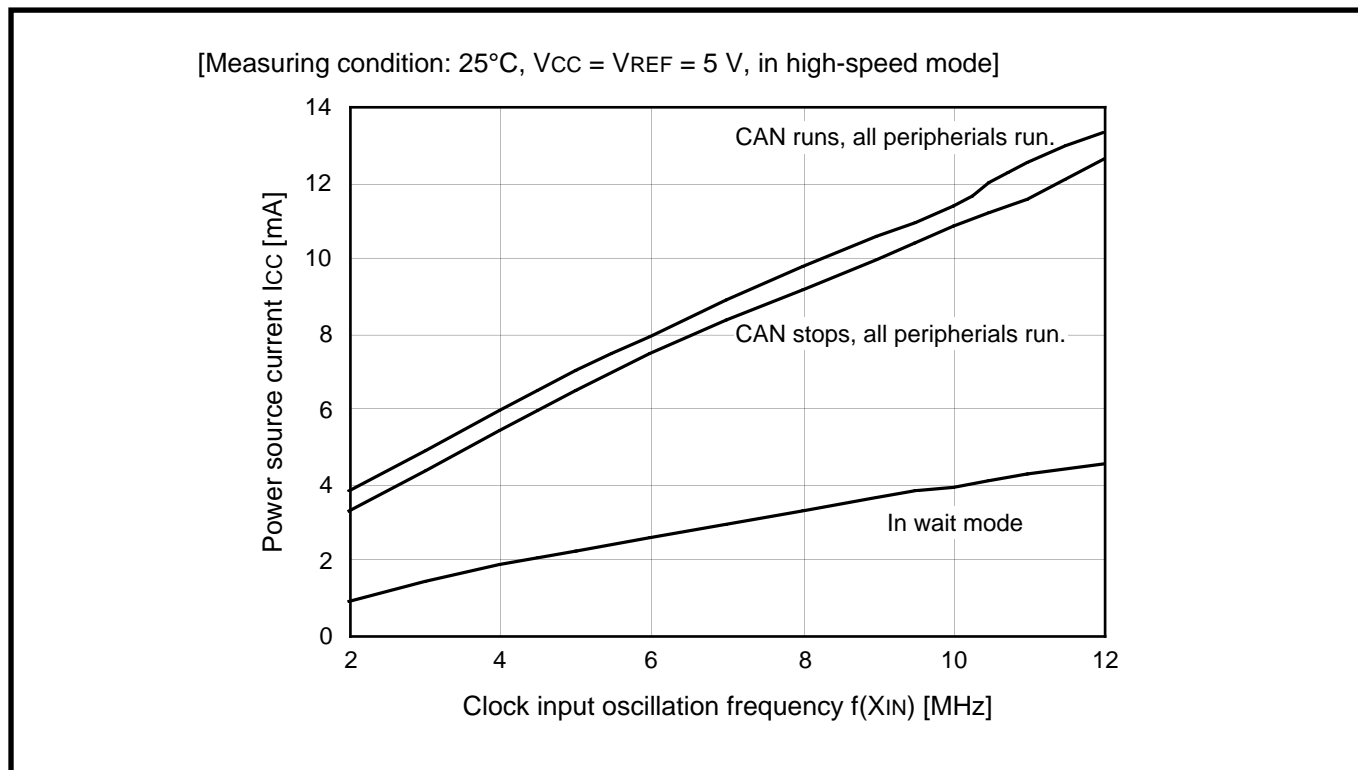


Fig. 3.2.3 Power source current standard characteristics (in high-speed mode)

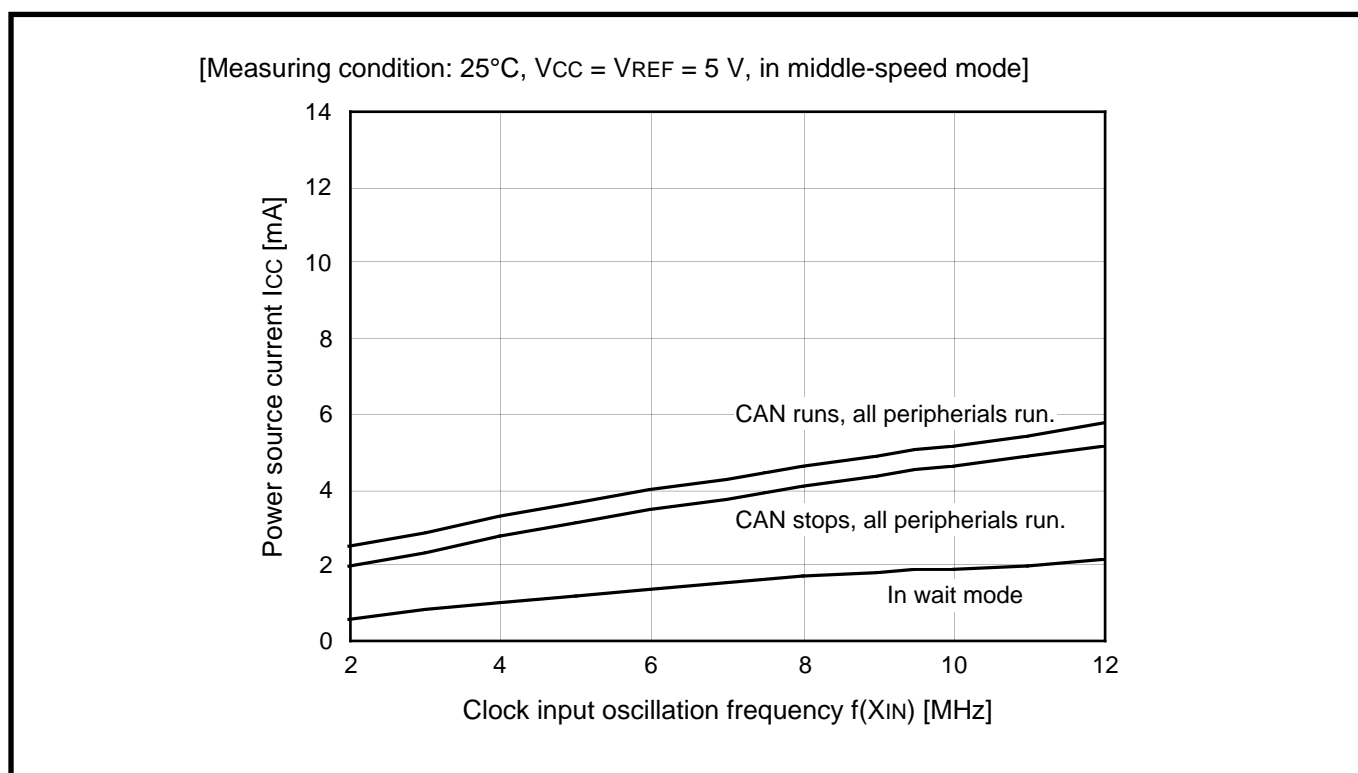


Fig. 3.2.4 Power source current standard characteristics (in middle-speed mode)

APPENDIX

3.2 Standard characteristics

3.2.2 Output current standard characteristics

Figures 3.2.5 and Figure 3.2.6 show the output current standard characteristics.

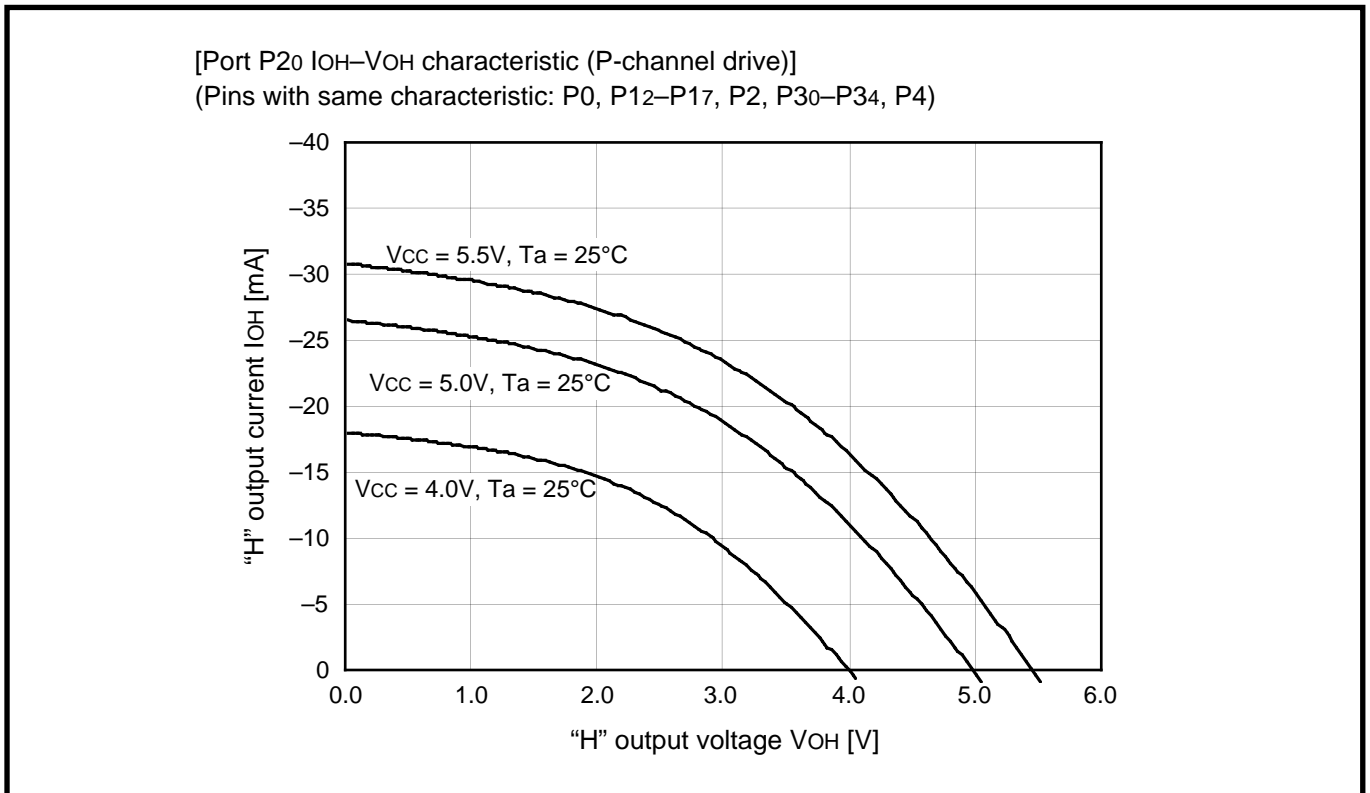


Fig. 3.2.5 Output current standard characteristics (P-channel)

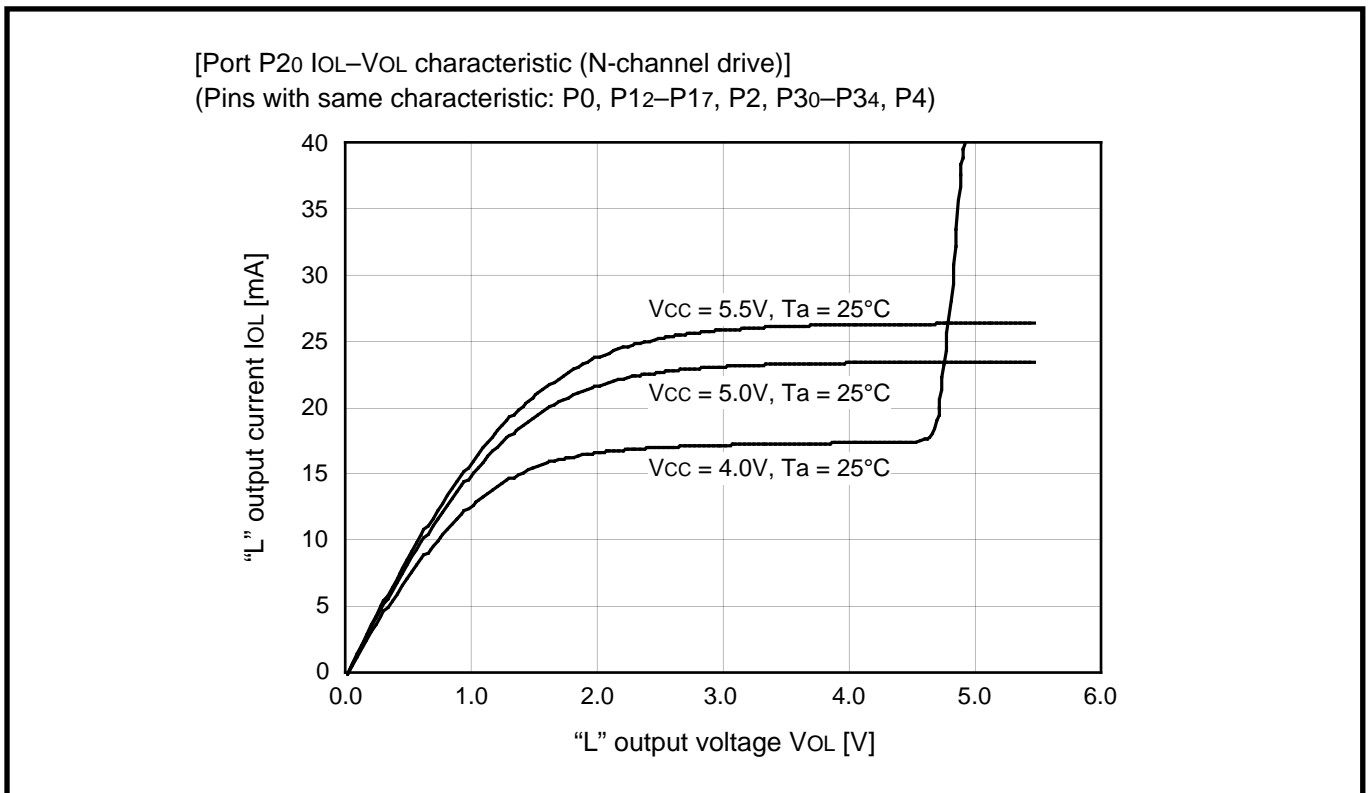


Fig. 3.2.6 Output current standard characteristics (N-channel)

3.2.3 Input current standard characteristics

Figure 3.2.7 and Figure 3.2.8 show the input current standard characteristics.

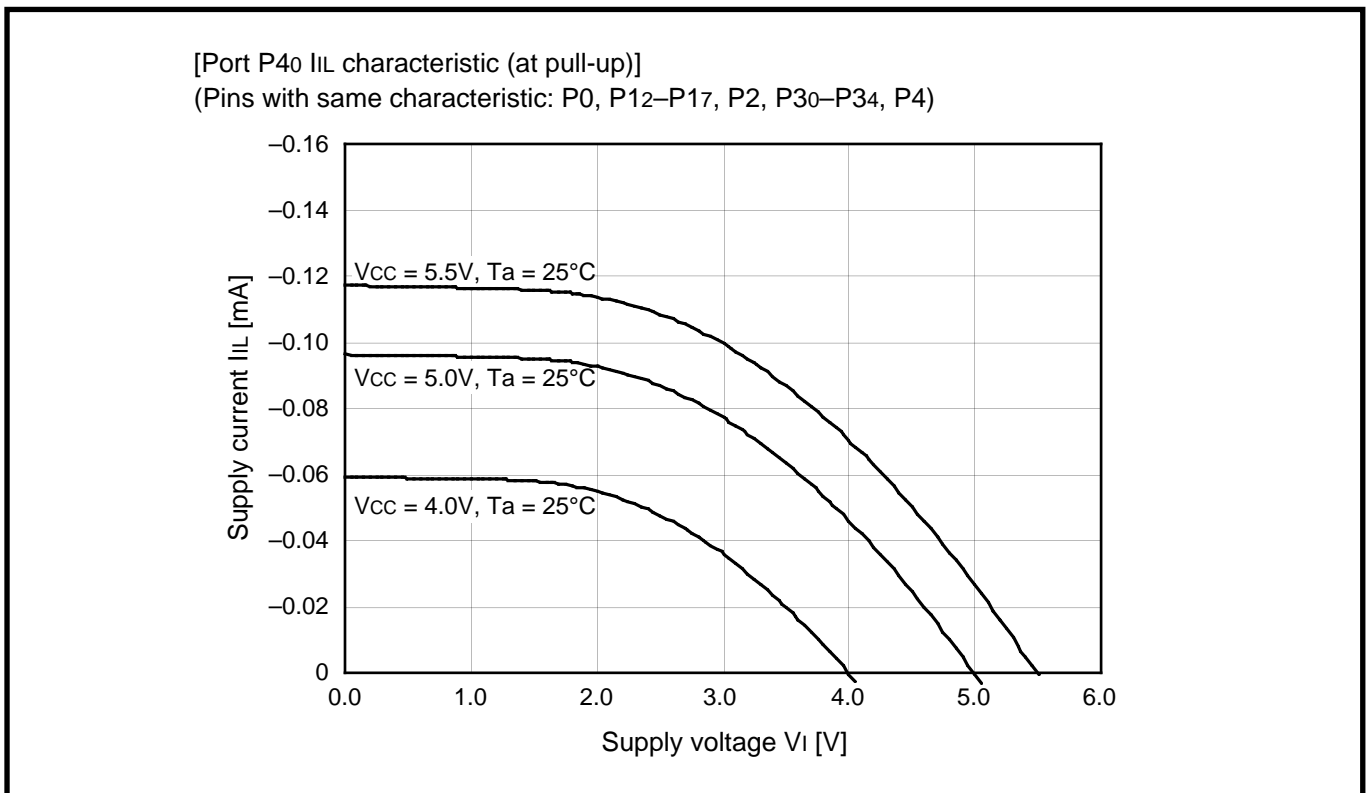


Fig. 3.2.7 Pull-up transistor standard characteristics I_{IL} - V_I

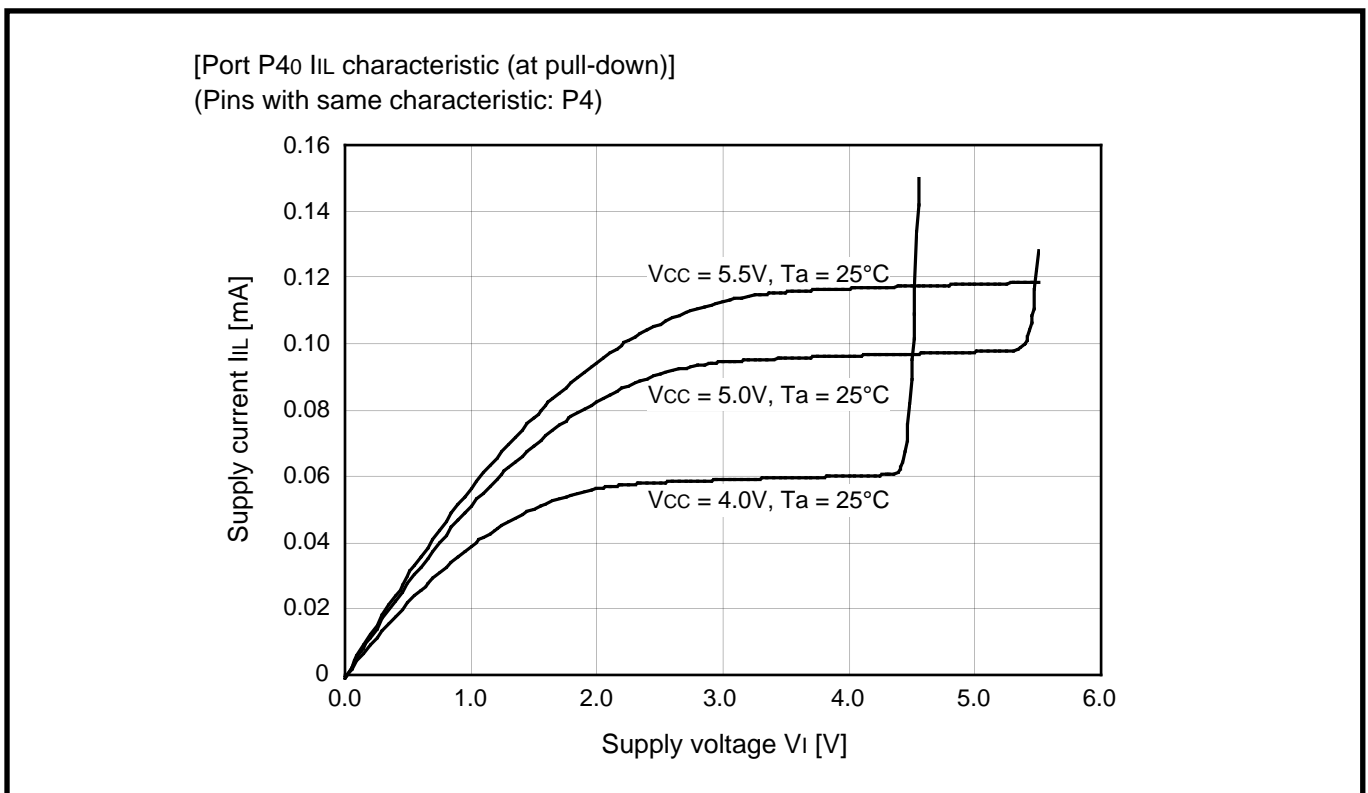


Fig. 3.2.8 Pull-down transistor standard characteristics I_{IL} - V_I

APPENDIX

3.2 Standard characteristics

3.2.4 A-D conversion standard characteristics

Figure 3.2.9 shows the A-D conversion standard characteristics.

The lower-side line on the graph indicates the absolute precision error (ERROR). It represents the deviation from the ideal value. For example, the conversion of output code from 0 to 1 occurs ideally at the point of $AN_0 = 10 \text{ mV}$, but the measured value is -2 mV . Accordingly, the measured point of conversion is represented as $"10 - (-2) = 12 \text{ [mV]}"$.

The upper-side line on the graph indicates the width of input voltages (1 LSB WIDTH) equivalent to output codes. For example, the measured width of the input voltage for output code 7 is 19 mV , so the differential nonlinear error is represented as $"19 - 20 = -1 \text{ [mV]} (-0.05 \text{ LSB})"$.

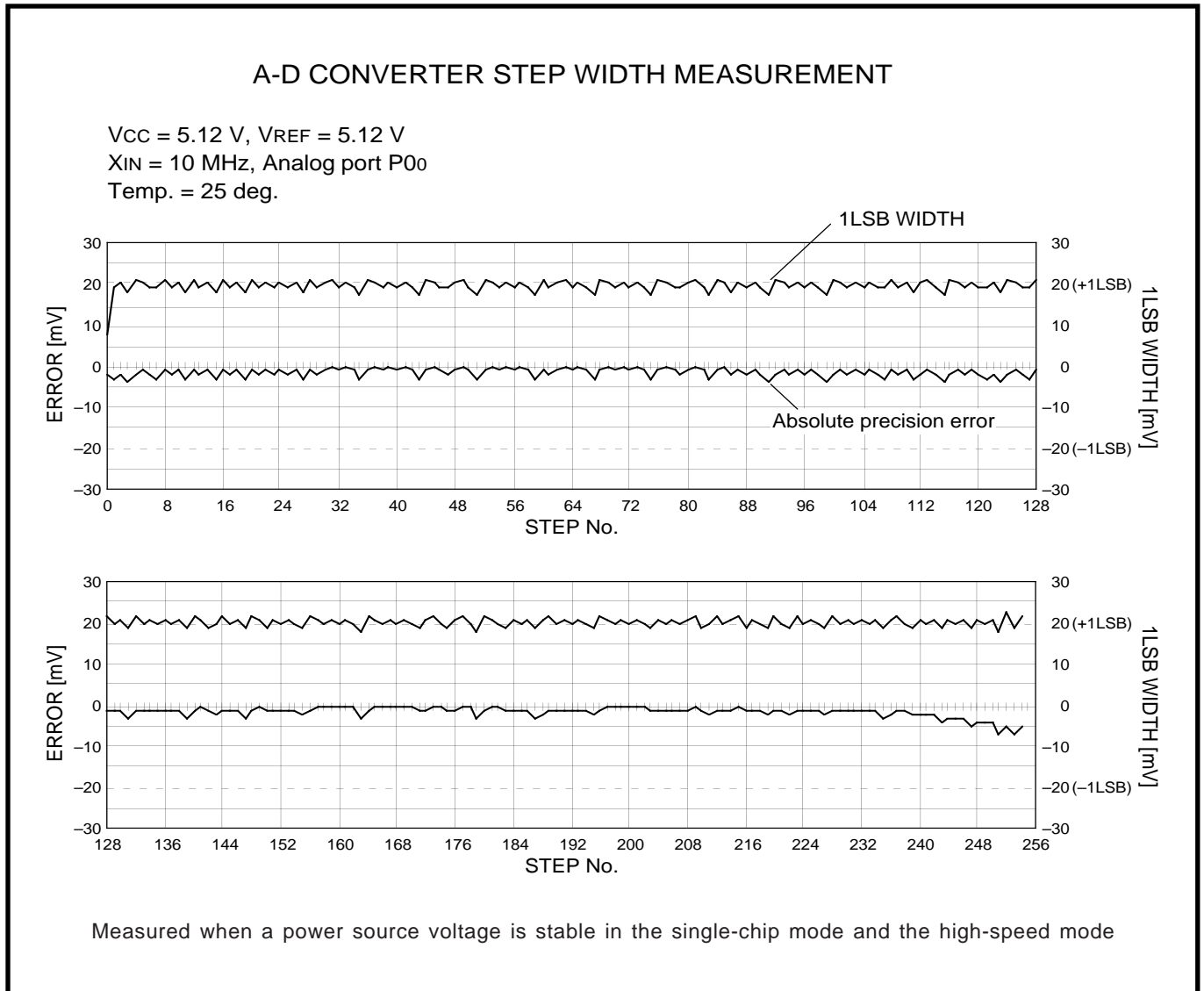


Fig. 3.2.9 A-D conversion standard characteristics

3.3 Notes on use

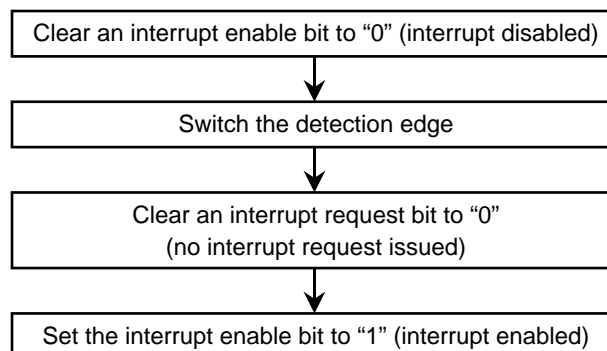
3.3.1 Notes on interrupts

(1) Switching an external interrupt detection edge

When the external interrupt detection edge must be switched, make sure the following sequence.

Reason

The interrupt circuit recognizes the switching of the detection edge as the change of external input signals. This may cause an unnecessary interrupt.



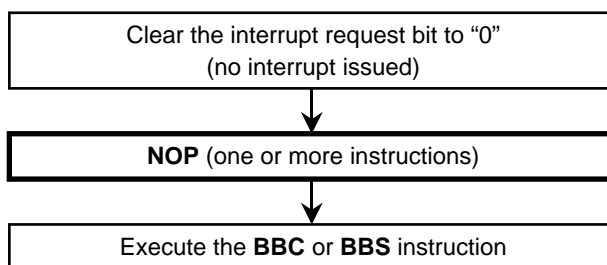
(2) Check of interrupt request bit

When executing the **BBC** or **BBS** instruction to an interrupt request bit of an interrupt request register immediately after this bit is set to "0" by using a data transfer instruction*, execute one or more instructions before executing the **BBC** or **BBS** instruction.

* data transfer instructions: **LDM**, **LDA**, **STA**, **STX**, and **STY** instruction

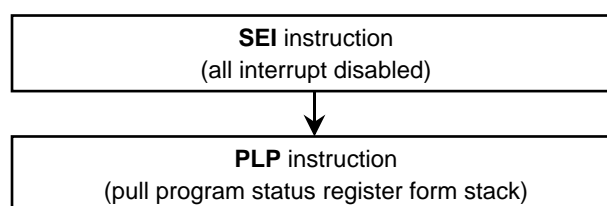
Reason

If the **BBC** or **BBS** instruction is executed immediately after an interrupt request bit of an interrupt request register is cleared to "0", the value of the interrupt request bit before being cleared to "0" is read.



(3) Stack operation of the program status register

If the current program status register is stored on the stack after executing the **PHP** instruction, the **PLP** instruction has to be entered to get the old program status register back. Before the **PLP** instruction can be executed the **SEI** instruction has to be executed.



3.3.2 Notes on A-D converter

(1) Analog input pin

Make the signal source impedance for analog input low, or equip an analog input pin with an external capacitor of 0.01 μF to 1 μF . Further, be sure to verify the operation of application products on the user side.

Reason

An analog input pin includes the capacitor for analog voltage comparison. Accordingly, when signals from signal source with high impedance are input to an analog input pin, charge and discharge noise generates. This may cause the A-D conversion precision to be worse.

APPENDIX

3.3 Notes on use

(2) AVss pin

AVss pin is the A-D converter power source pin. Regardless of using the A-D conversion function or not, connect AVss to the Vss line.

Reason

If the AVss pin is opened, the microcomputer may have a failure because of noise or others.

(3) Clock frequency during an A-D conversion

The comparator consists of a capacity coupling, and a charge of the capacity will be lost if the clock frequency is too low. Thus, make sure the following during an A-D conversion.

- $f(XIN)$ is 500 kHz or more.
- Do not execute the **STP** instruction and **WIT** instruction.

3.3.3 Notes on \overline{RESET} pin

In case where the \overline{RESET} signal rise time is long, connect a ceramic capacitor or others across the \overline{RESET} pin and the Vss pin. And use a 1000 pF or more capacitor for high frequency use. When connecting the capacitor, note the following :

- Make the length of the wiring which is connected to a capacitor as short as possible.
- Be sure to check the operation of application products on the user side.

Reason

If the several nanosecond or several ten nanosecond impulse noise enters the \overline{RESET} pin, it may cause a microcomputer failure.

3.3.4 Notes on input and output pins

(1) Notes in stand-by state

In stand-by state^{*1} for low-power dissipation, do not make input levels of an input port and an I/O port "undefined," especially for I/O ports of the P-channel and the N-channel open-drain.

Pull-up (connect the port to VCC) or pull-down (connect the port to Vss) these ports through a resistor.

When determining a resistance value, note the following:

- External circuit
- Variation of output levels during the ordinary operation

When using built-in pull-up or pull-down resistor, note on varied current values.

- When setting as an input port : Fix its input level.
- When setting as an output port : Prevent current from flow to the external.

^{*1} stand-by state : The stop mode by executing the **STP** instruction or the wait mode by executing the **WIT** instruction

Reason

Even when setting as an output port with its direction register, in the following state :

- P-channel.....when the content of the data register (port latch) is "0"
- N-channel.....when the content of the data register (port latch) is "1"

the transistor becomes the OFF state, which causes the ports to be the high-impedance state. Note that the level becomes "undefined" depending on external circuits.

Accordingly, the potential which is input to the input buffer in a microcomputer is unstable in the state that input levels of an input port and an I/O port are "undefined." This may cause power source current.

(2) Modifying output data with bit managing

When the data register (port latch) of an I/O port is modified with the bit managing instruction*², the value of the unspecified bit may be changed.

Reason

The bit managing instructions are read-modify-write form instructions for reading and writing data by a byte unit. Accordingly, when these instructions are executed on a bit of the port latch of an I/O port, the following is executed to all bits of the data register.

- As for a bit which is set for an input port:
The pin state is read in the CPU, and is written to this bit after bit managing.
- As for a bit which is set for an output port:
The bit value is read in the CPU, and is written to this bit after bit managing.

Note the following :

- Even when a port which is set as an output port is changed for an input port, its data register holds the output data.
- As for a bit of which is set for an input port, its value may be changed even when not specified with a bit managing instruction in case where the pin state differs from its data register contents.

*² bit managing instructions : **SEB**, and **CLB** instructions

3.3.5 Notes on programming

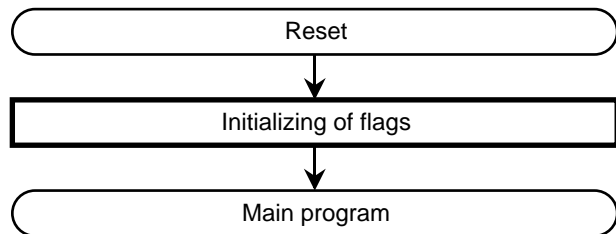
(1) Initialization of processor status register

Flags which affect program execution must be initialized after a reset.

In particular, it is essential to initialize the T and D flags because they have an important effect on calculations.

Reason

After a reset, the contents of the processor status register (PS) are undefined except for the I flag which is "1".



(2) How to reference the processor status register

To reference the contents of the processor status register (PS), execute the **PHP** instruction once then read the contents of (S)+1. If necessary, execute the **PLP** instruction to return the PS to its original status. A **NOP** instruction must be executed after every **PLP** instruction.

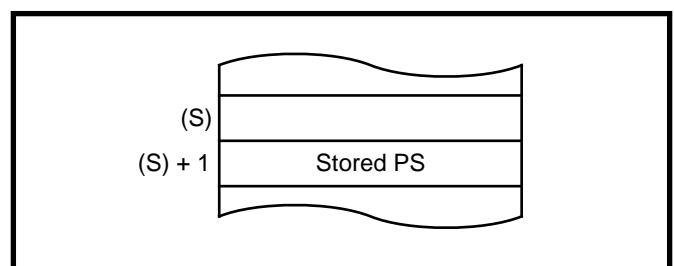
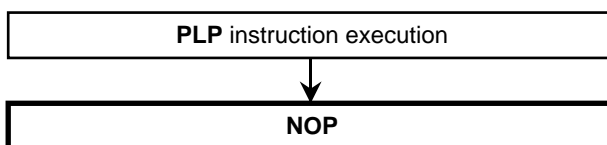


Fig. 3.3.1 Stack memory contents after instruction execution

APPENDIX

3.3 Notes on use

(3) Detection of BRK instruction interrupt source

It can be detected that the **BRK** instruction interrupt event or the least priority interrupt event by referring the stored B flag state. Refer to the stored B flag state in the interrupt routine.

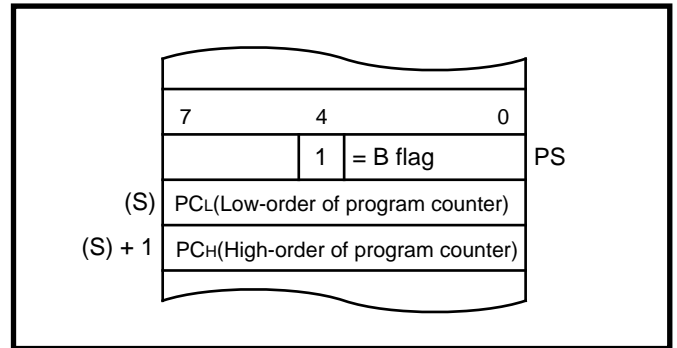


Fig. 3.3.2 Interrupt routine

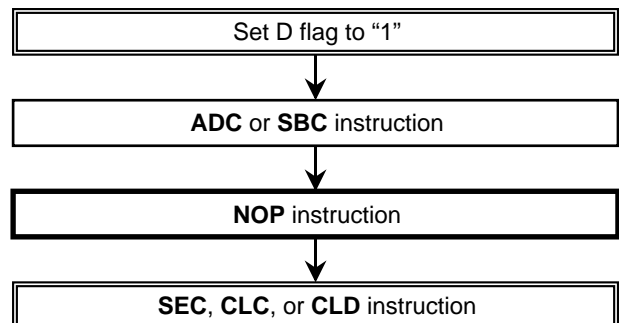
(4) Execution of decimal calculations

The **ADC** and **SBC** are the only instructions which will yield proper decimal notation, set the decimal mode flag (D) to "1" with the **SED** instruction. After executing the **ADC** or **SBC** instruction, execute another instruction before executing the **SEC**, **CLC**, or **CLD** instruction.

(5) Notes on status flags in decimal mode

When decimal mode is selected, the values of three of the flags in the status register (the N, V, and Z flags) are invalid after a **ADC** or **SBC** instruction is executed.

The carry flag (C) is set to "1" if a carry is generated as a result of the calculation, or is cleared to "0" if a borrow is generated. To determine whether a calculation has generated a carry, the C flag must be initialized to "0" before each calculation. To check for a borrow, the C flag must be initialized to "1" before each calculation.



(6) JPM instruction

When using the **JMP** instruction in indirect addressing mode, do not specify the last address on a page as an indirect address.

3.4 Countermeasures against noise

Countermeasures against noise are described below. The following countermeasures are effective against noise in theory, however, it is necessary not only to take measures as follows but to evaluate before actual use.

3.4.1 Shortest wiring length

The wiring on a printed circuit board can function as an antenna which feeds noise into the microcomputer. The shorter the total wiring length (by mm unit), the less the possibility of noise insertion into a microcomputer.

(1) Wiring for the $\overline{\text{RESET}}$ input pin

Make the length of wiring which is connected to the $\overline{\text{RESET}}$ input pin as short as possible. Especially, connect a capacitor across the $\overline{\text{RESET}}$ input pin and the Vss pin with the shortest possible wiring (within 20mm).

Reason

The width of a pulse input into the $\overline{\text{RESET}}$ pin is determined by the timing necessary conditions. If noise having a shorter pulse width than the standard is input to the $\overline{\text{RESET}}$ input pin, the reset is released before the internal state of the microcomputer is completely initialized. This may cause a program runaway.

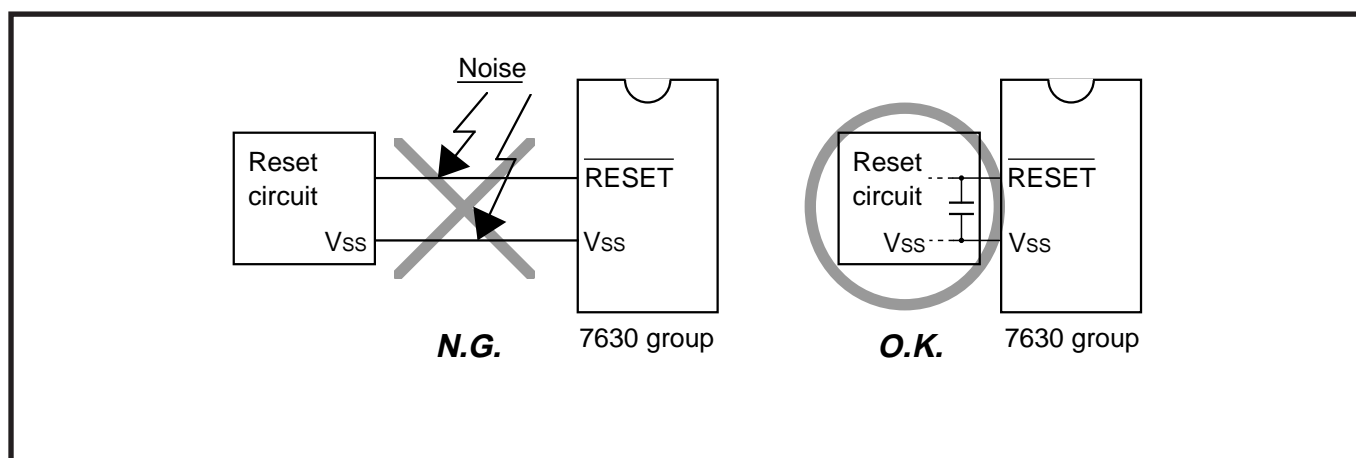


Fig. 3.4.1 Wiring for the $\overline{\text{RESET}}$ input pin

(2) Wiring for clock input/output pins

- Make the length of wiring which is connected to clock I/O pins as short as possible.
- Make the length of wiring (within 20mm) across the grounding lead of a capacitor which is connected to an oscillator and the Vss pin of a microcomputer as short as possible.
- Separate the Vss pattern only for oscillation from other Vss patterns.

Reason

If noise enters clock I/O pins, clock waveforms may be deformed. This may cause a program failure or program runaway. Also, if a potential difference is caused by the noise between the Vss level of a microcomputer and the Vss level of an oscillator, the correct clock will not be input in the microcomputer.

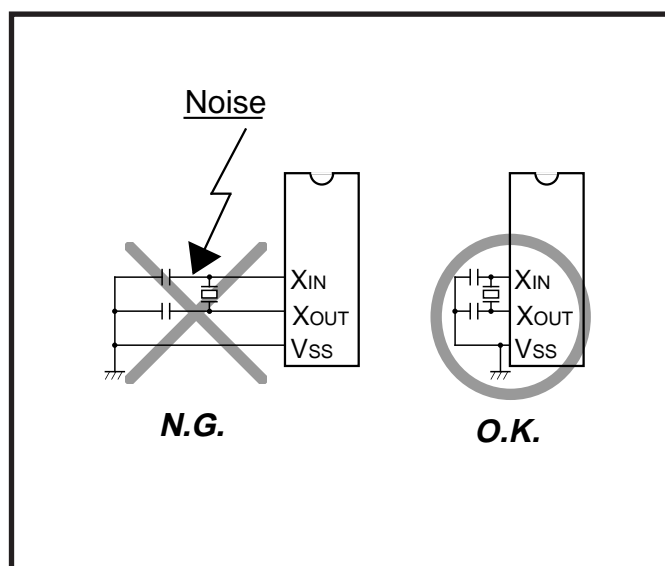


Fig. 3.4.2 Wiring for clock I/O pins

APPENDIX

3.4 Countermeasures against noise

(3) Wiring for the VPP pin of the One Time PROM version and the EPROM version

Connect an approximately 5 kΩ resistor to the VPP pin the shortest possible in series. When not connecting the resistor, make the length of wiring for the VPP pin the shortest possible.

Note: Even when a circuit which included an approximately 5 kΩ resistor is used in the Mask ROM version, the microcomputer operates correctly.

Reason

The VPP pin of the One Time PROM and the EPROM version is the power source input pin for the built-in PROM. When programming in the built-in PROM, the impedance of the VPP pin is low to allow the electric current for writing flow into the PROM. Because of this, noise can enter easily. If noise enters the VPP pin, abnormal instruction codes or data are read from the built-in PROM, which may cause a program runaway.

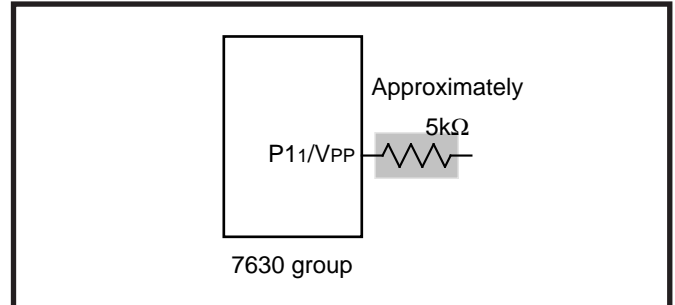


Fig. 3.4.3 Wiring for the VPP pin of the One Time PROM and the EPROM version

3.4.2 Connection of a bypass capacitor across the Vss line and the Vcc line

Connect an approximately 0.1 μF bypass capacitor across the Vss line and the Vcc line as follows:

- Connect a bypass capacitor across the Vss pin and the Vcc pin at equal length.
- Connect a bypass capacitor across the Vss pin and the Vcc pin with the shortest possible wiring.
- Use lines with a larger diameter than other signal lines for Vss line and Vcc line.
- Connect the power source wiring via a bypass capacitor to the Vss pin and the Vcc pin.

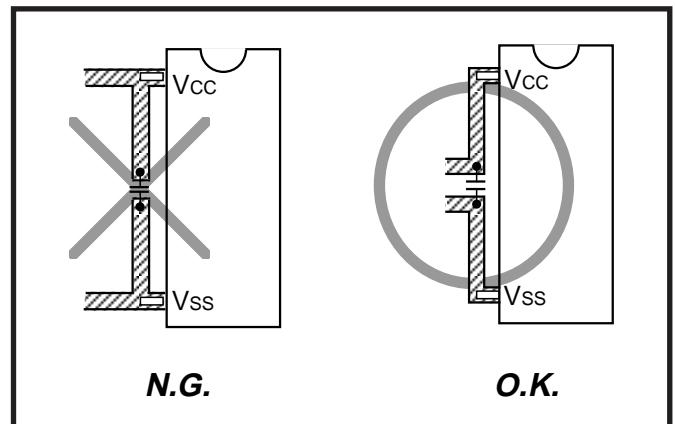


Fig. 3.4.4 Bypass capacitor across the Vss line and the Vcc line

3.4.3 Wiring to analog input pins

- Connect an approximately 100 Ω to 1 kΩ resistor to an analog signal line which is connected to an analog input pin in series. Besides, connect the resistor to the microcomputer as close as possible.
- Connect an approximately 1000 pF capacitor across the Vss pin and the analog input pin. Besides, connect the capacitor to the Vss pin as close as possible. Also, connect the capacitor across the analog input pin and the Vss pin at equal length.

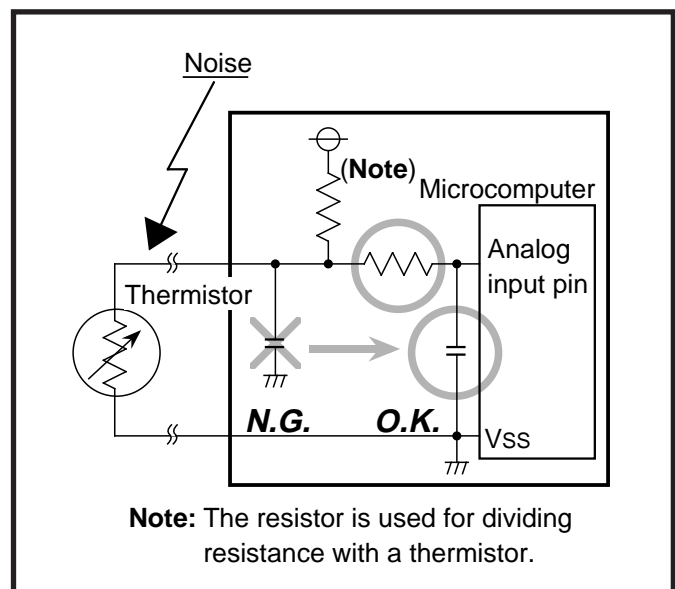


Fig. 3.4.5 Analog signal line and a resistor and a capacitor

3.4 Countermeasures against noise

Reason

Signals which is input in an analog input pin (such as an A-D converter/comparator input pin) are usually output signals from sensor. The sensor which detects a change of event is installed far from the printed circuit board with a microcomputer, the wiring to an analog input pin is longer necessarily. This long wiring functions as an antenna which feeds noise into the microcomputer, which causes noise to an analog input pin.

If a capacitor between an analog input pin and the Vss pin is grounded at a position far away from the Vss pin, noise on the GND line may enter a microcomputer through the capacitor.

3.4.4 Consideration for oscillator

Take care to prevent an oscillator that generates clocks for a microcomputer operation from being affected by other signals.

(1) Keeping oscillator away from large current signal lines

Install a microcomputer (and especially an oscillator) as far as possible from signal lines where a current larger than the tolerance of current value flows.

Reason

In the system using a microcomputer, there are signal lines for controlling motors, LEDs, and thermal heads or others. When a large current flows through those signal lines, strong noise occurs because of mutual inductance.

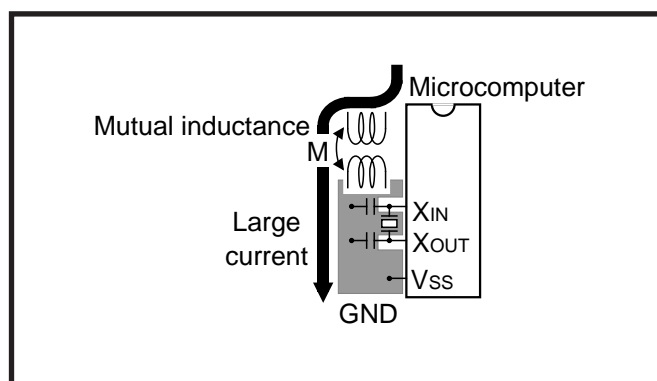


Fig. 3.4.6 Wiring for a large current signal line

(2) Installing oscillator away from signal lines where potential levels change frequently

Install an oscillator and a connecting pattern of an oscillator away from signal lines where potential levels change frequently. Also, do not cross such signal lines over the clock lines or the signal lines which are sensitive to noise.

Reason

Signal lines where potential levels change frequently (such as the CNTR pin signal line) may affect other lines at signal rising edge or falling edge. If such lines cross over a clock line, clock waveforms may be deformed, which causes a microcomputer failure or a program runaway.

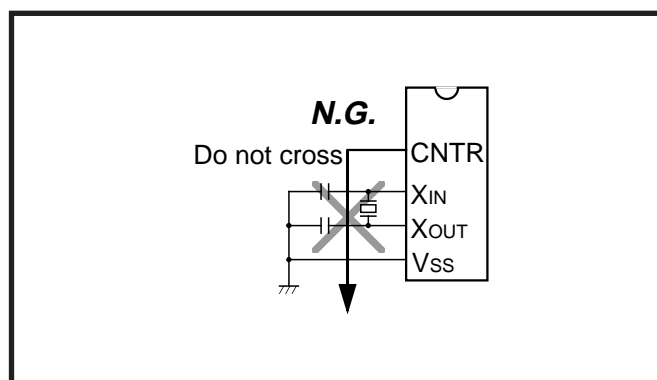


Fig. 3.4.7 Wiring to a signal line where potential levels change frequently

APPENDIX

3.4 Countermeasures against noise

(3) Oscillator protection using Vss pattern

As for a two-sided printed circuit board, print a Vss pattern on the underside (soldering side) of the position (on the component side) where an oscillator is mounted.

Connect the Vss pattern to the microcomputer Vss pin with the shortest possible wiring. Besides, separate this Vss pattern from other Vss patterns.

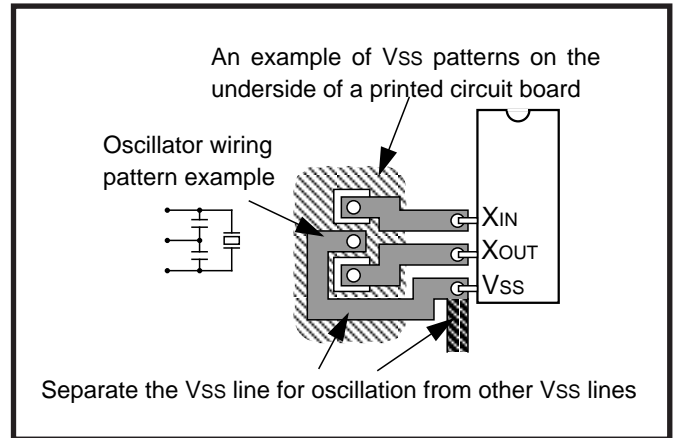


Fig. 3.4.8 Vss pattern on the underside of an oscillator

3.4.5 Setup for I/O ports

Setup I/O ports using hardware and software as follows:

<Hardware>

- Connect a resistor of 100 Ω or more to an I/O port in series.

<Software>

- As for an input port, read data several times by a program for checking whether input levels are equal or not.
- As for an output port, since the output data may reverse because of noise, rewrite data to its data register at fixed periods.
- Rewrite data to direction registers and pull-up control registers (only the product having it) at fixed periods.

When a direction register is set for input port again at fixed periods, a several-nanosecond short pulse may be output from this port. If this is undesirable, connect a capacitor to this port to remove the noise pulse.

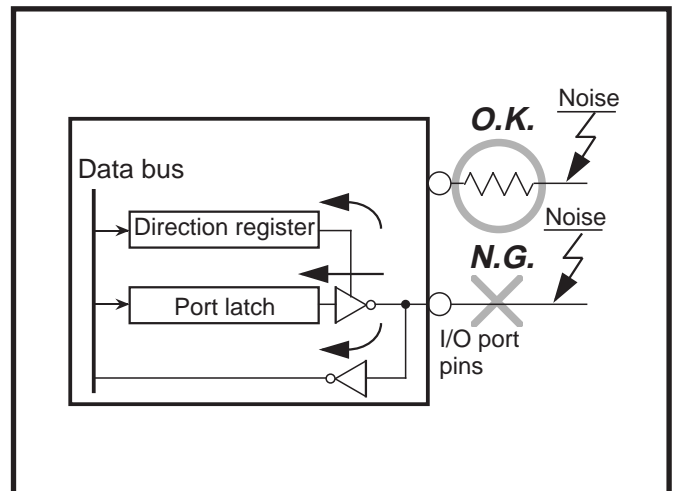


Fig. 3.4.9 Setup for I/O ports

3.4 Countermeasures against noise

3.4.6 Providing of watchdog timer function by software

If a microcomputer runs away because of noise or others, it can be detected by a software watchdog timer and the microcomputer can be reset to normal operation. This is equal to or more effective than program runaway detection by a hardware watchdog timer. The following shows an example of a watchdog timer provided by software.

In the following example, to reset a microcomputer to normal operation, the main routine detects errors of the interrupt processing routine and the interrupt processing routine detects errors of the main routine. This example assumes that interrupt processing is repeated multiple times in a single main routine processing.

<The main routine>

- Assigns a single byte of RAM to a software watchdog timer (SWDT) and writes the initial value N in the SWDT once at each execution of the main routine. The initial value N should satisfy the following condition:

$$N+1 \geq (\text{Counts of interrupt processing executed in each main routine})$$

As the main routine execution cycle may change because of an interrupt processing or others, the initial value N should have a margin.

- Watches the operation of the interrupt processing routine by comparing the SWDT contents with counts of interrupt processing after the initial value N has been set.
- Detects that the interrupt processing routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
If the SWDT contents do not change after interrupt processing.

<The interrupt processing routine>

- Decrements the SWDT contents by 1 at each interrupt processing.
- Determines that the main routine operates normally when the SWDT contents are reset to the initial value N at almost fixed cycles (at the fixed interrupt processing count).
- Detects that the main routine has failed and determines to branch to the program initialization routine for recovery processing in the following case:
If the SWDT contents are not initialized to the initial value N but continued to decrement and if they reach 0 or less.

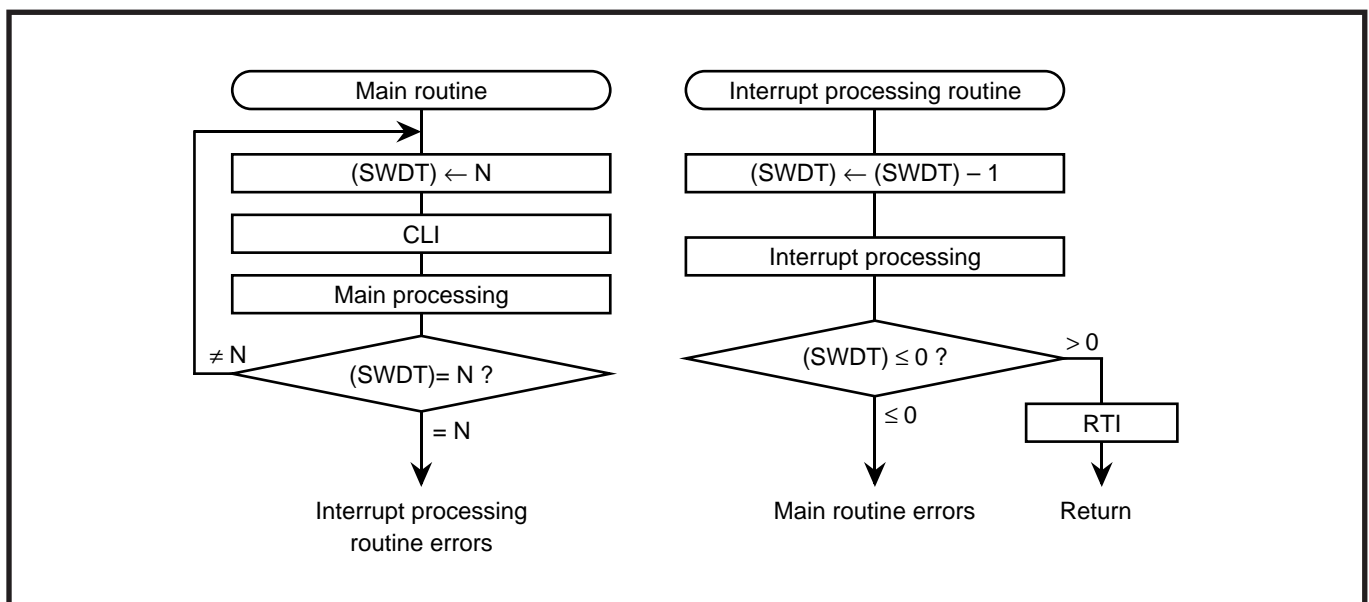


Fig. 3.4.10 Watchdog timer by software

APPENDIX

3.5 List of registers

3.5 List of registers

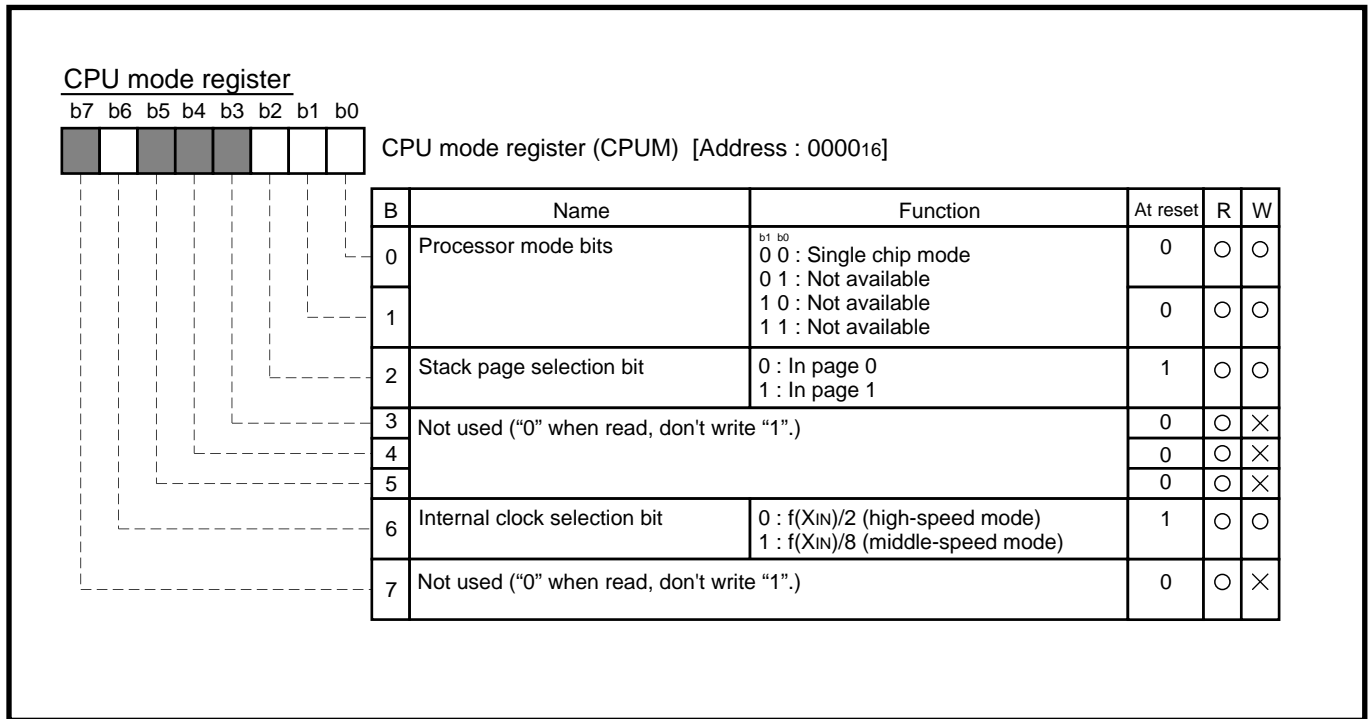


Fig. 3.5.1 Structure of CPU mode register

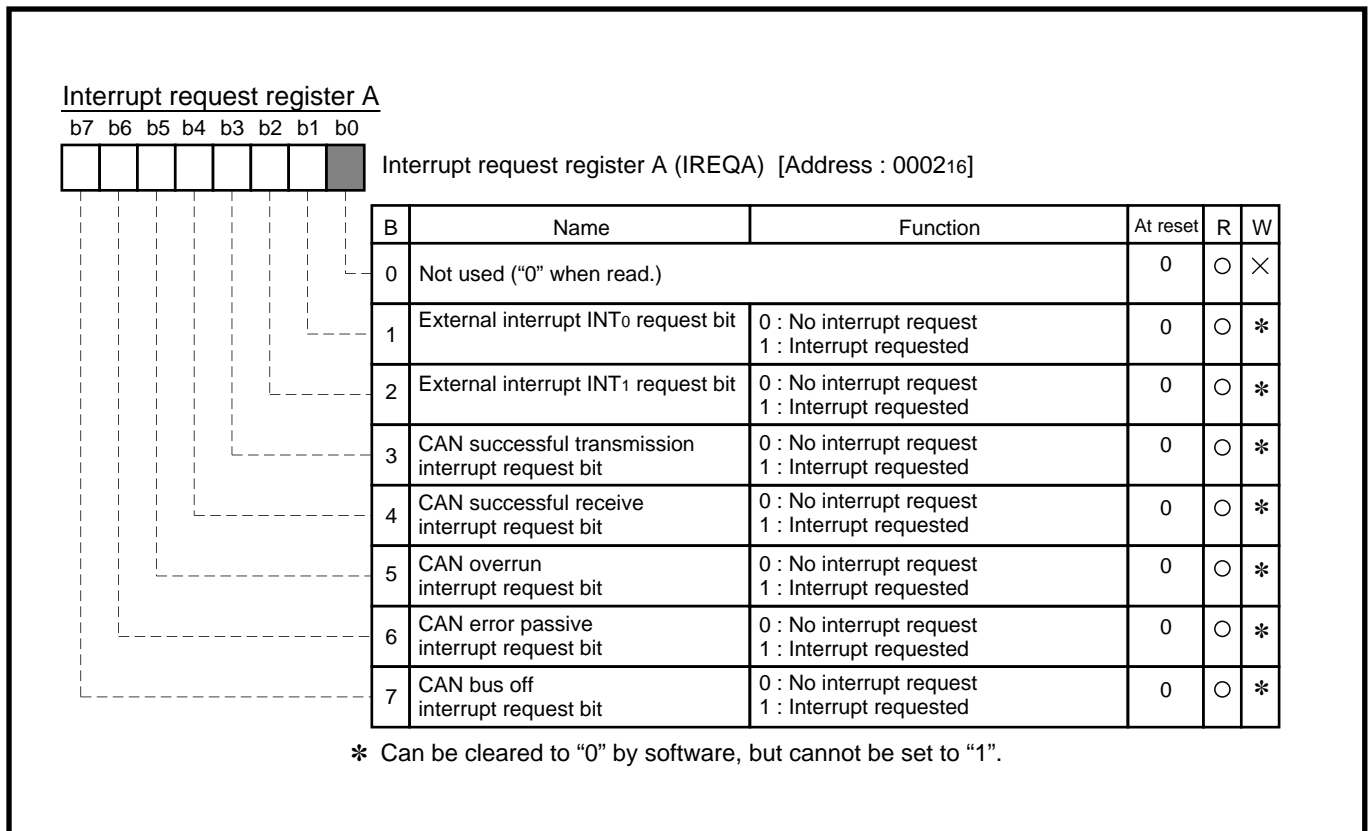


Fig. 3.5.2 Structure of Interrupt request register A

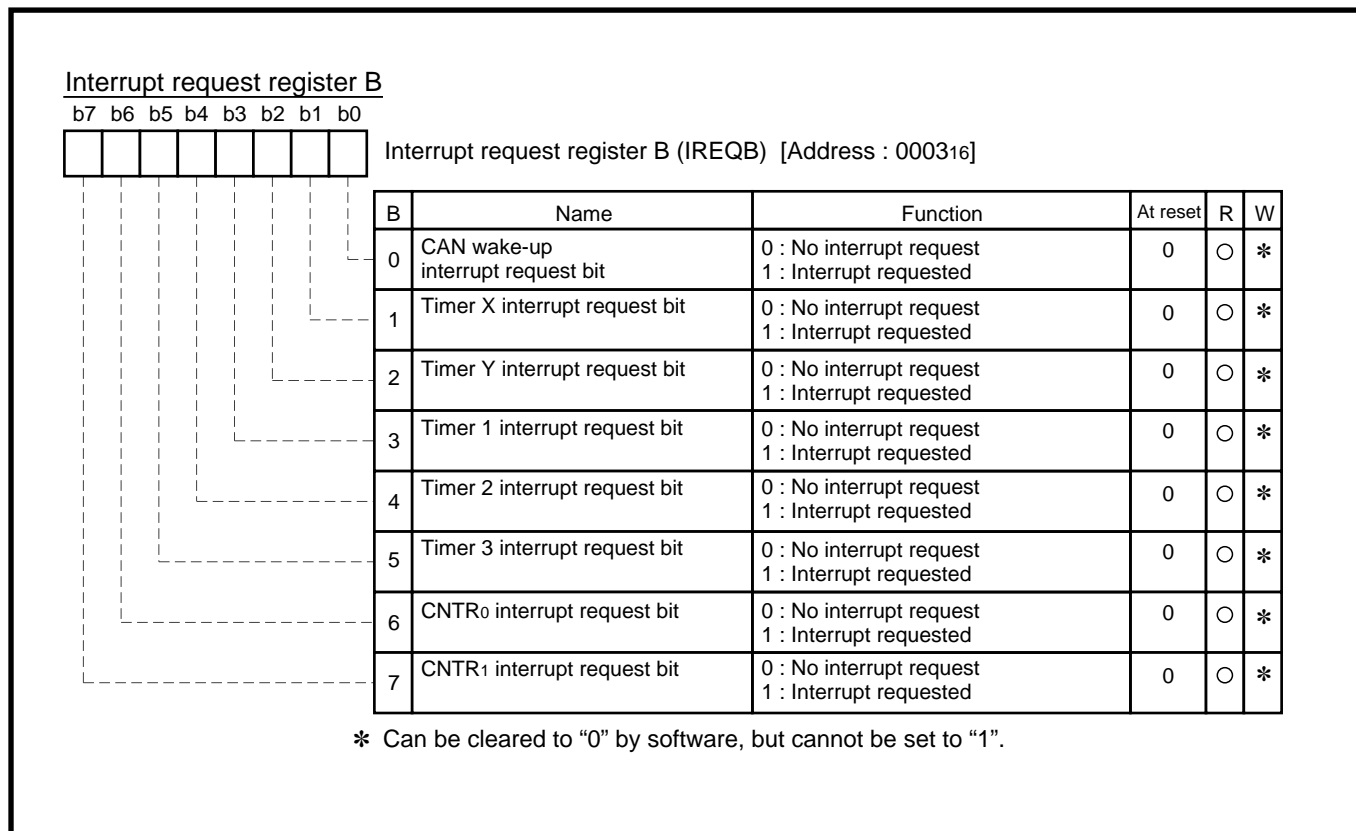


Fig. 3.5.3 Structure of Interrupt request register B

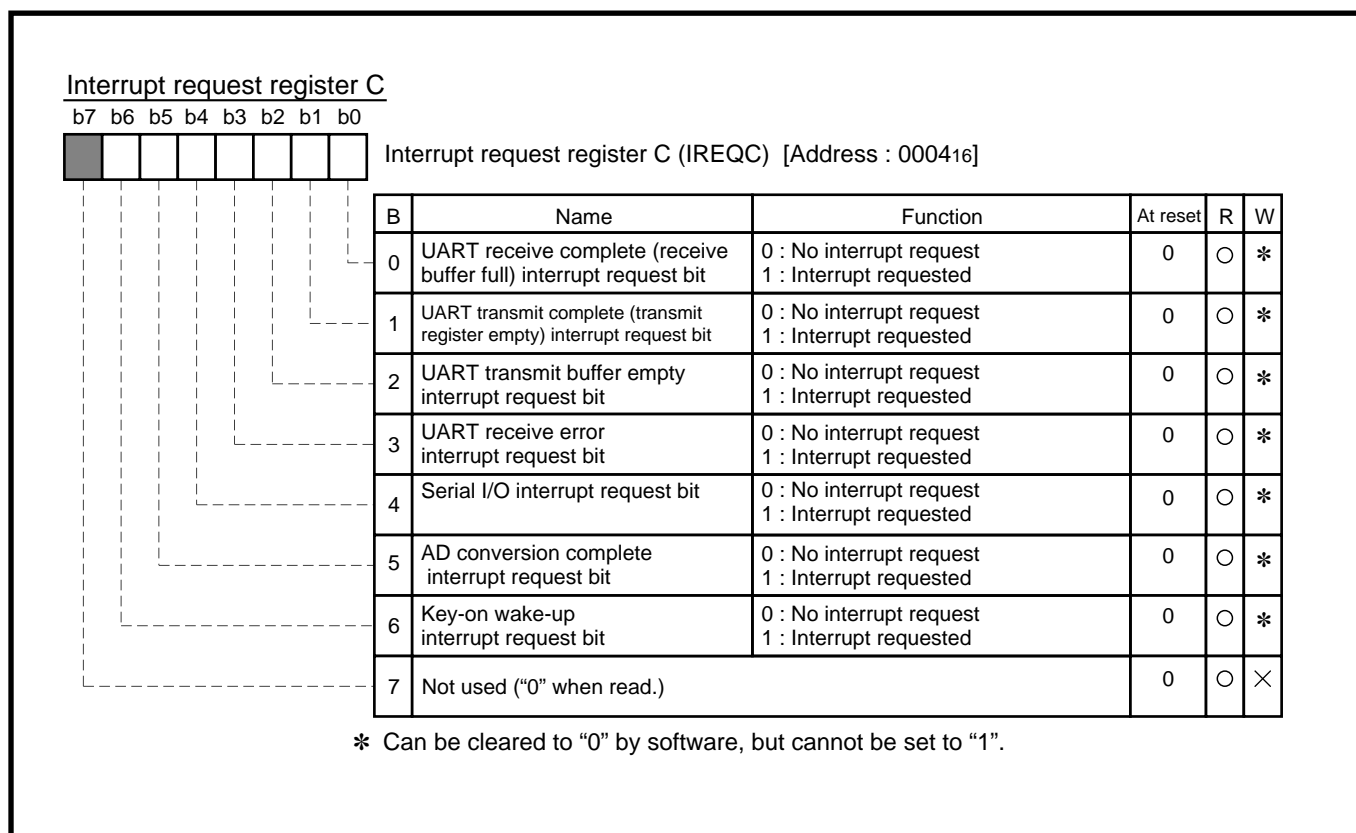


Fig. 3.5.4 Structure of Interrupt request register C

APPENDIX

3.5 List of registers

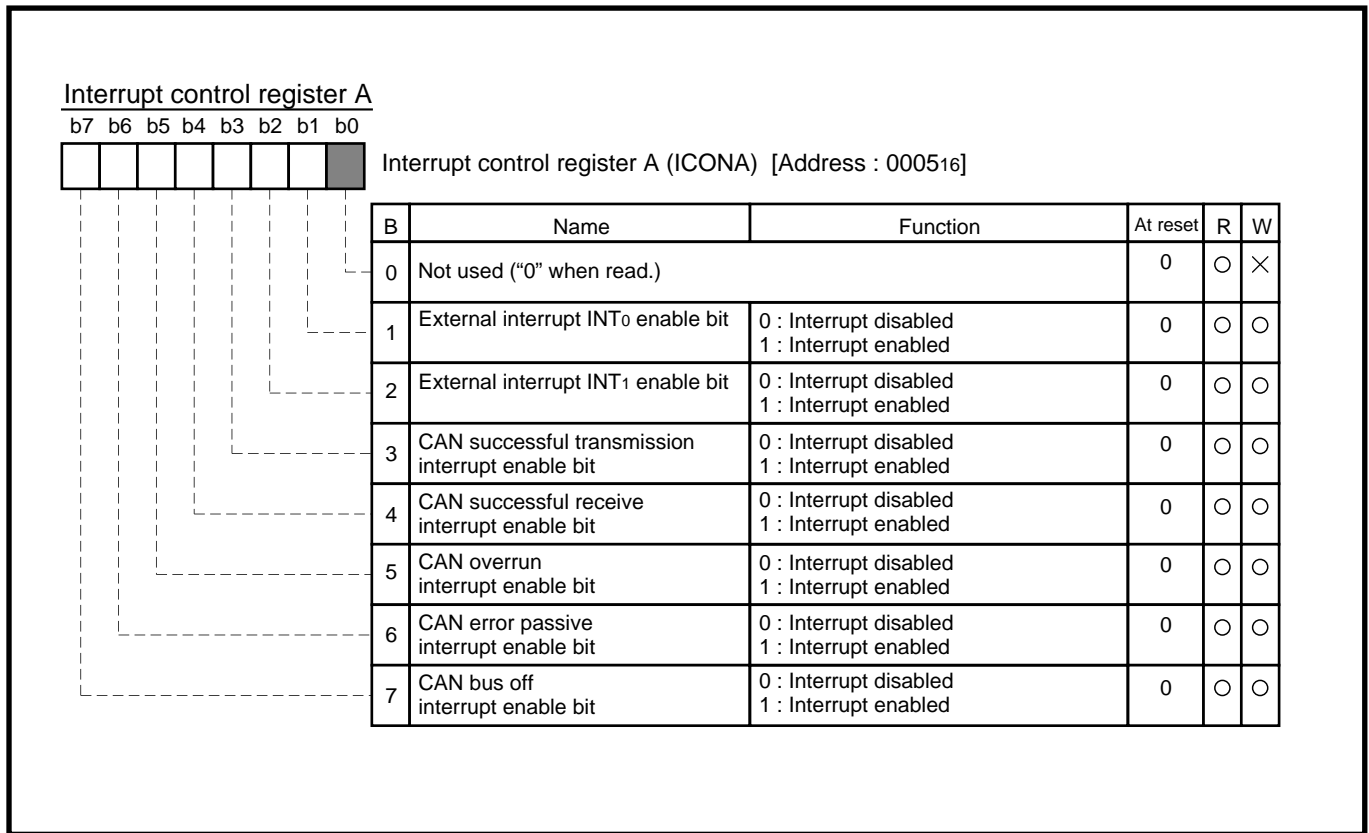


Fig. 3.5.5 Structure of Interrupt control register A

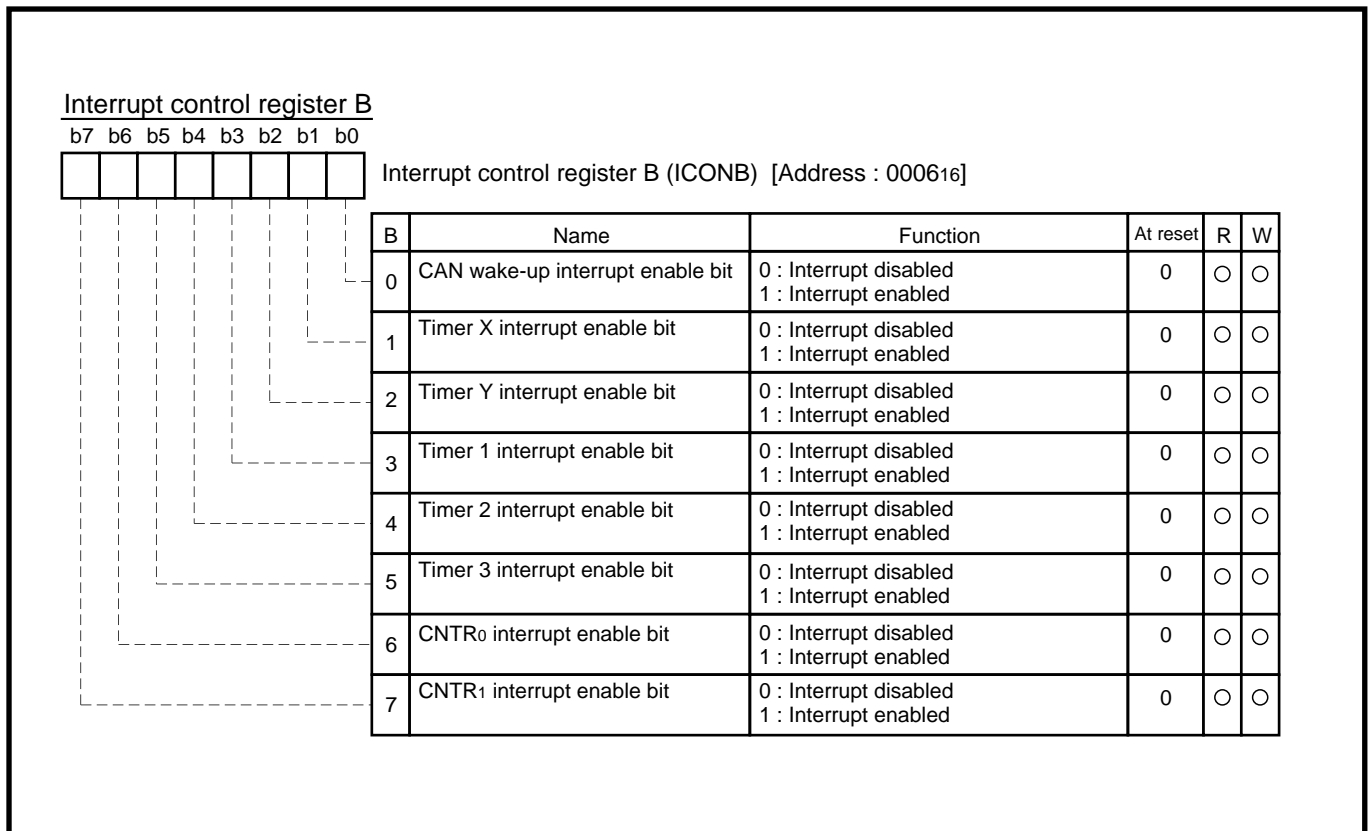


Fig. 3.5.6 Structure of Interrupt control register B

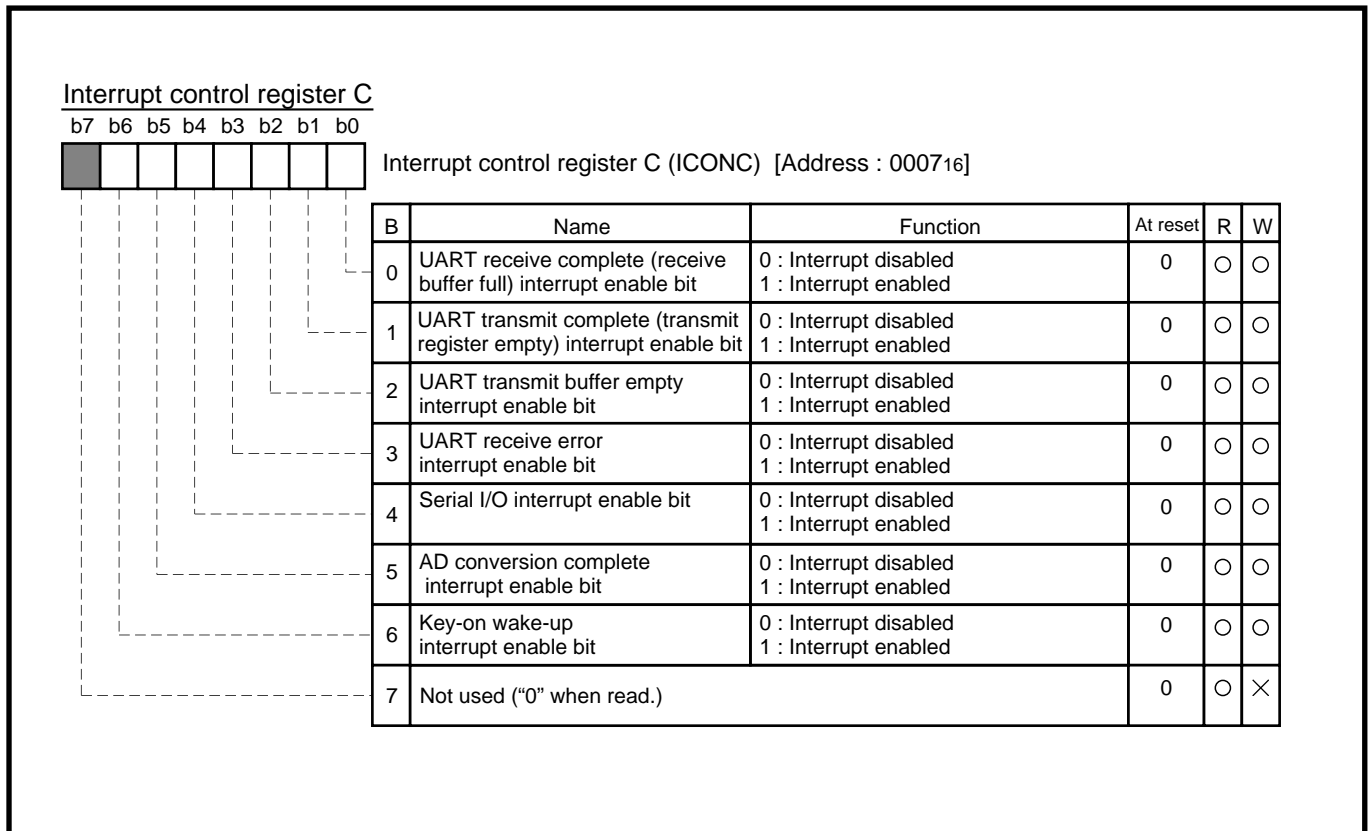


Fig. 3.5.7 Structure of Interrupt control register C

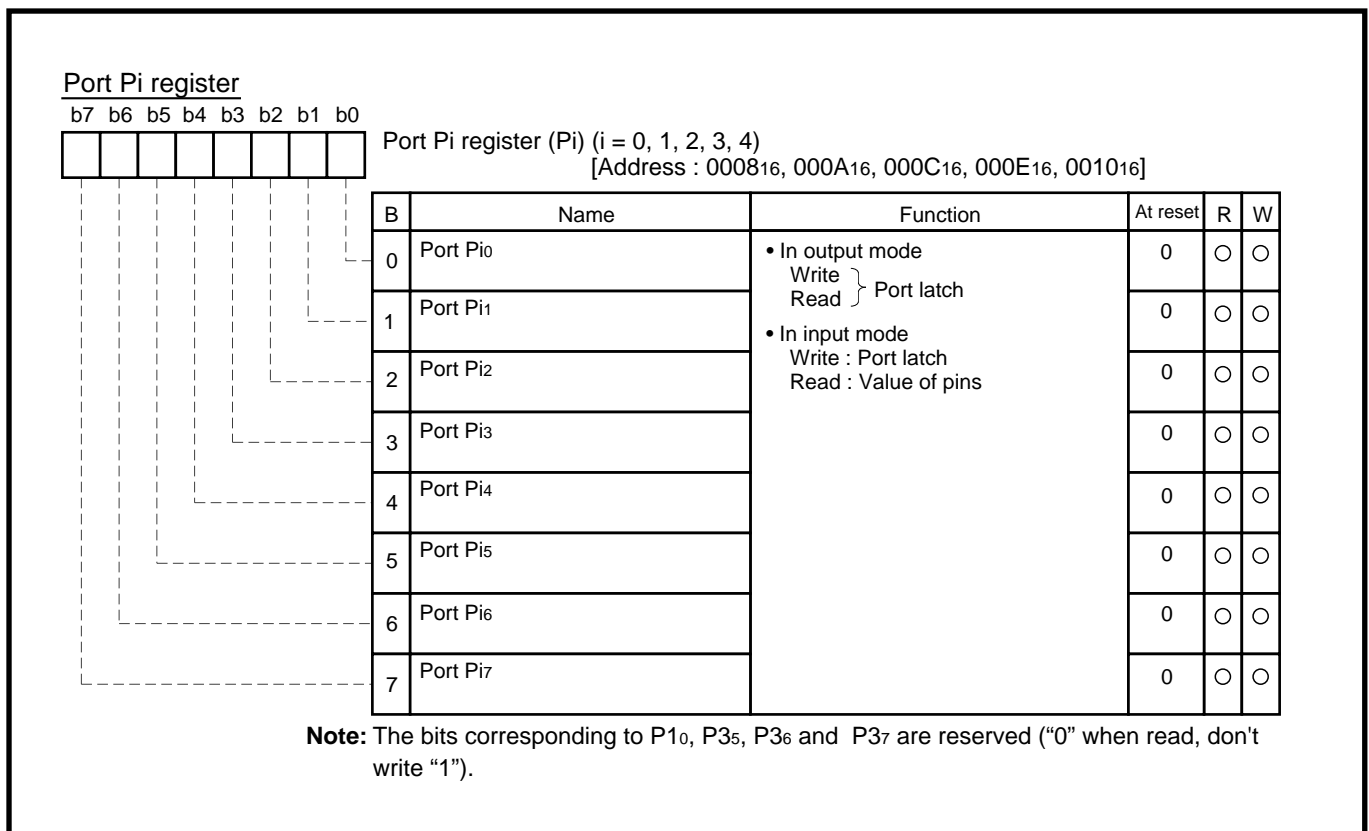


Fig. 3.5.8 Structure of Port Pi register (i = 0, 1, 2, 3, 4)

APPENDIX

3.5 List of registers

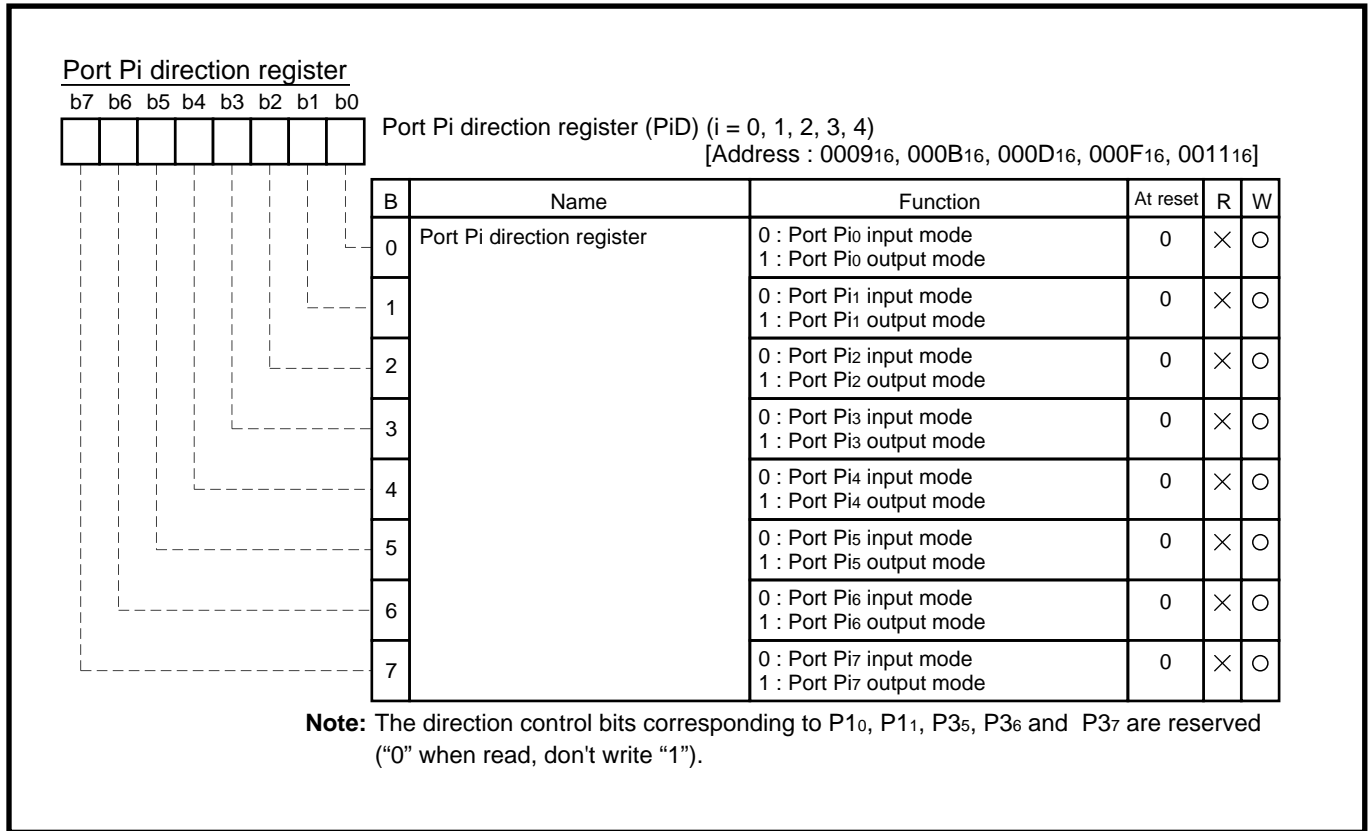


Fig. 3.5.9 Structure of Port Pi direction register (i = 0, 1, 2, 3, 4)

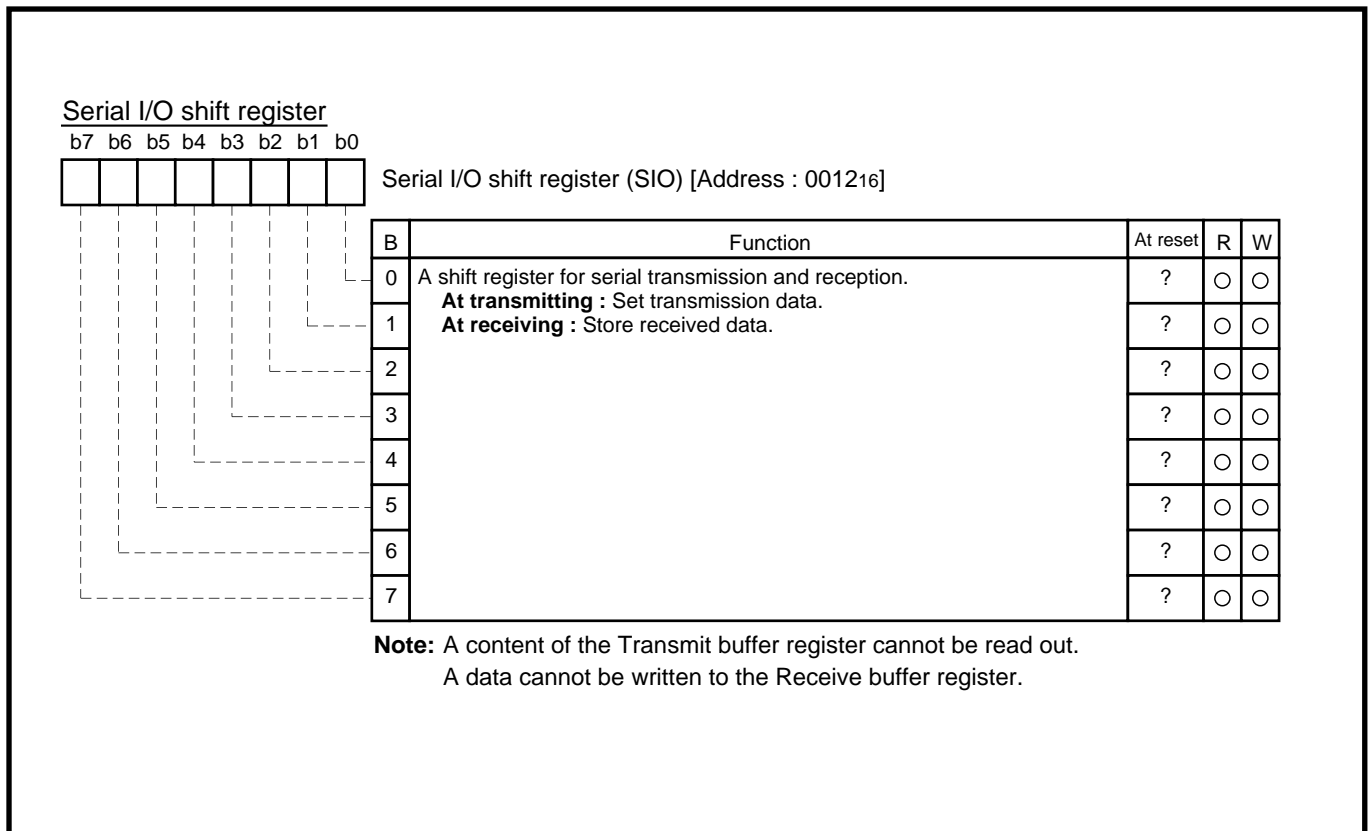


Fig. 3.5.10 Structure of Serial I/O shift register

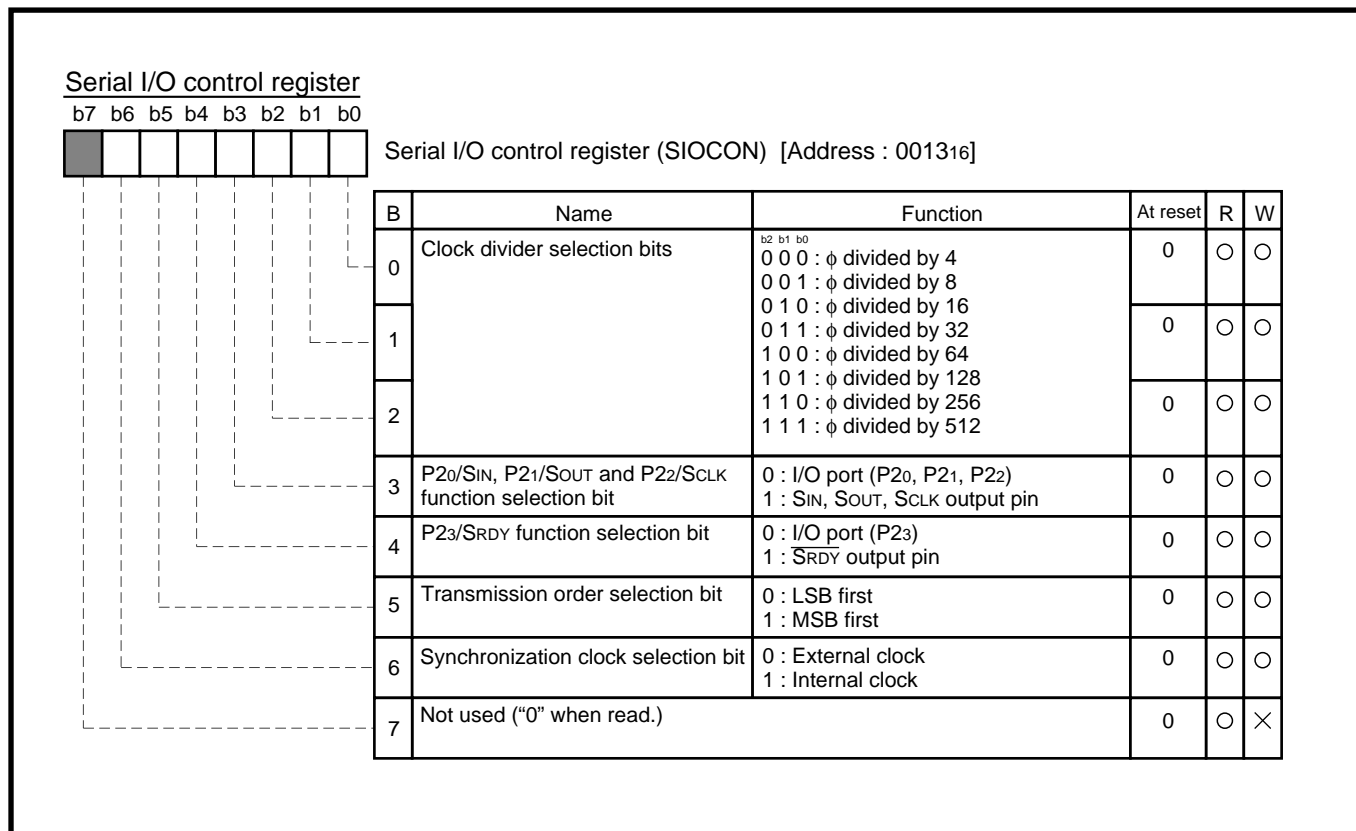


Fig. 3.5.11 Structure of Serial I/O control register

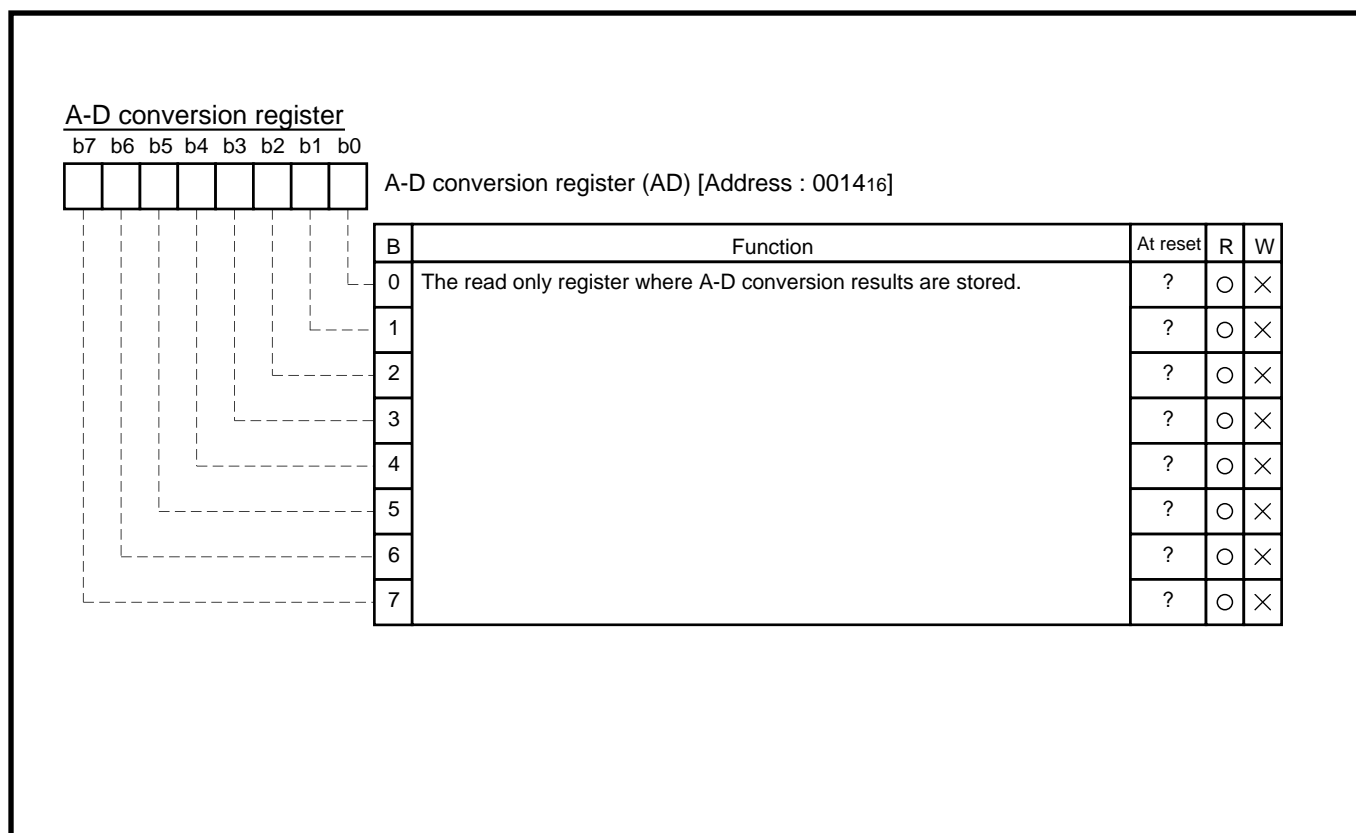


Fig. 3.5.12 Structure of A-D conversion register

APPENDIX

3.5 List of registers

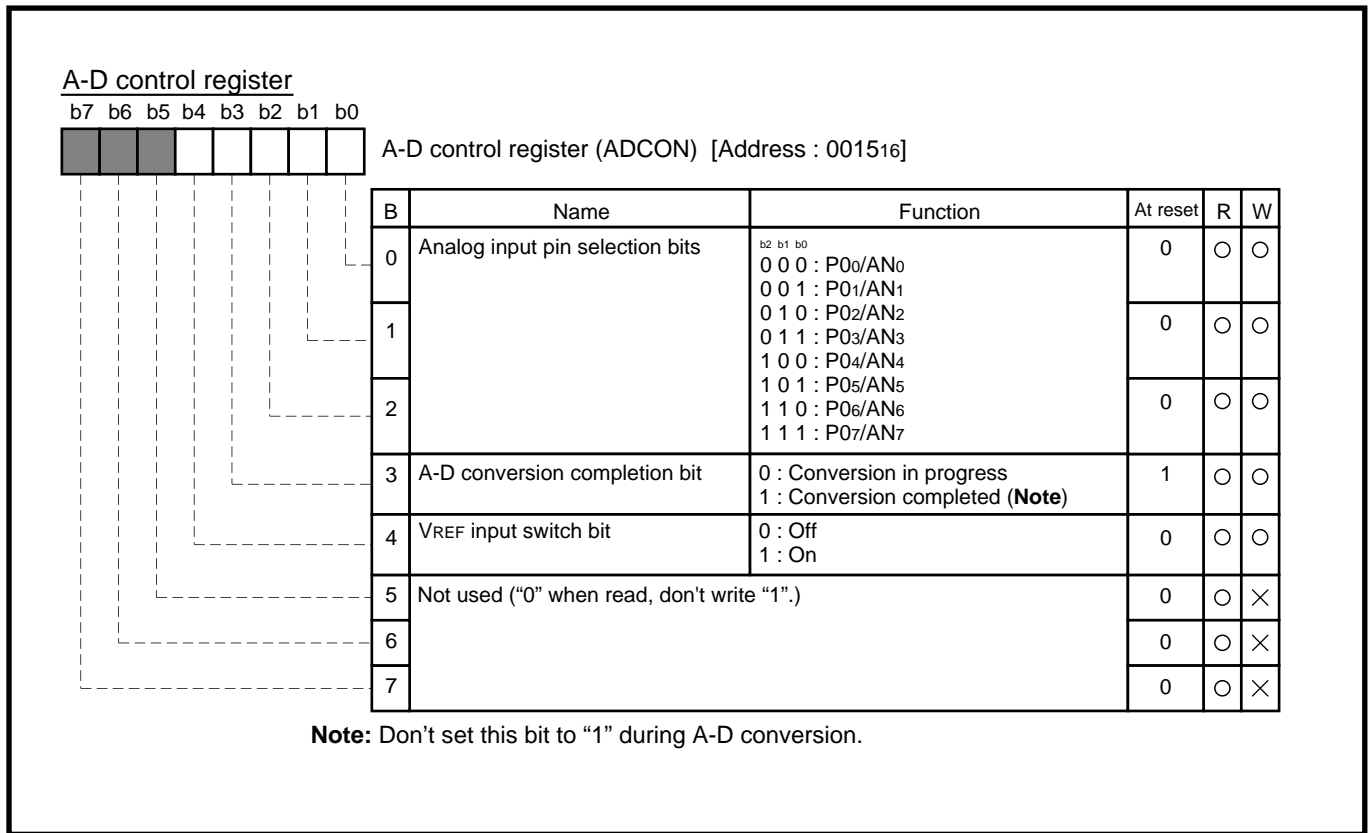


Fig. 3.5.13 Structure of A-D control register

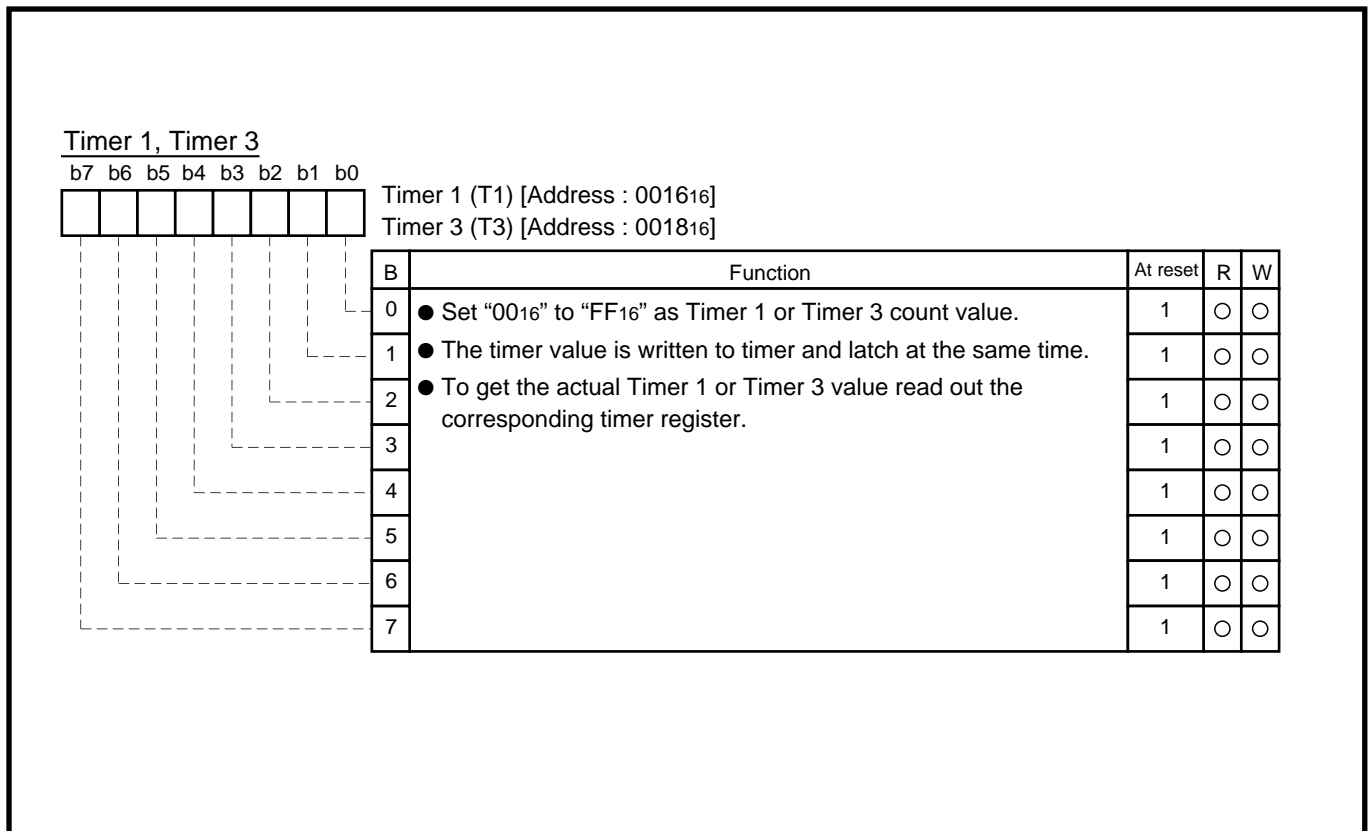


Fig. 3.5.14 Structure of Timer 1, Timer 3

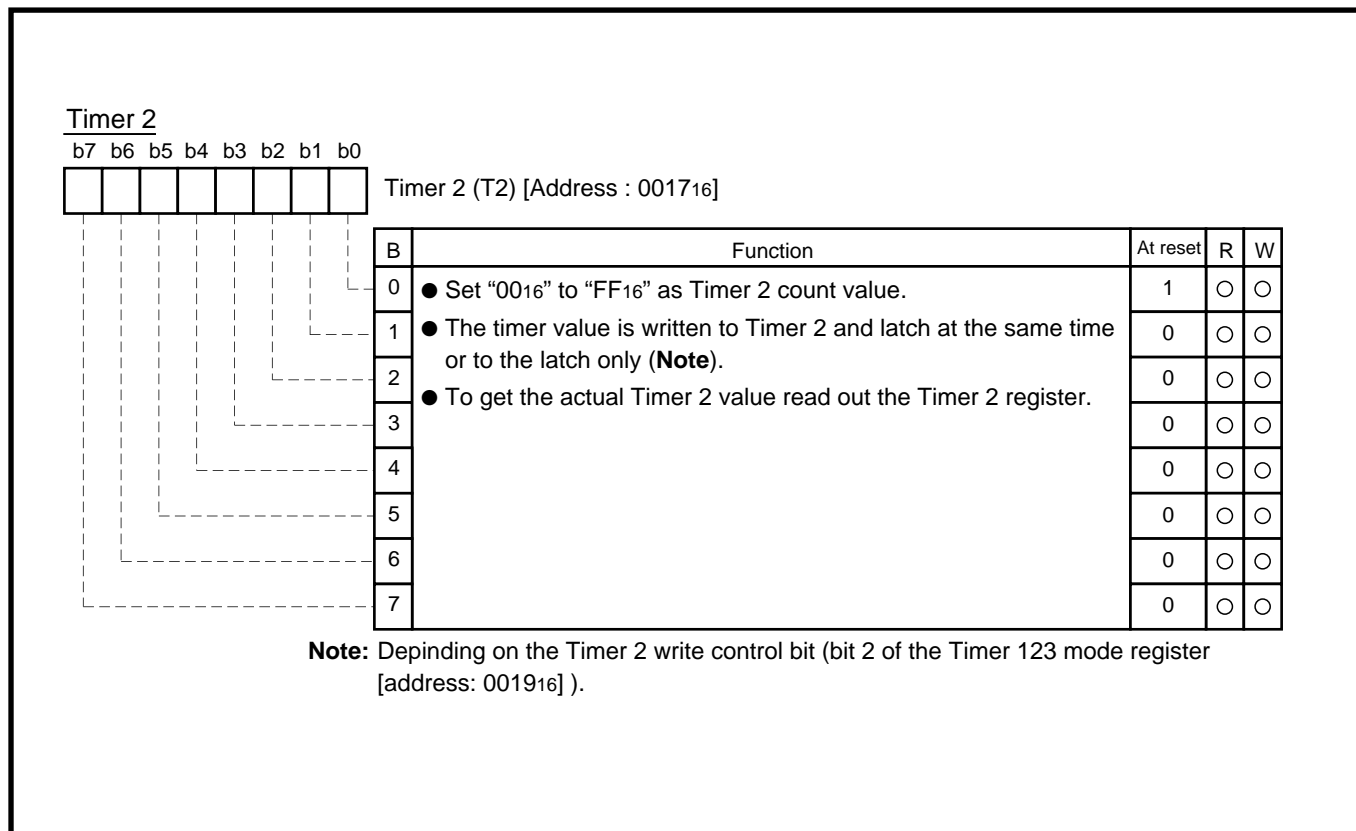


Fig. 3.5.15 Structure of Timer 2

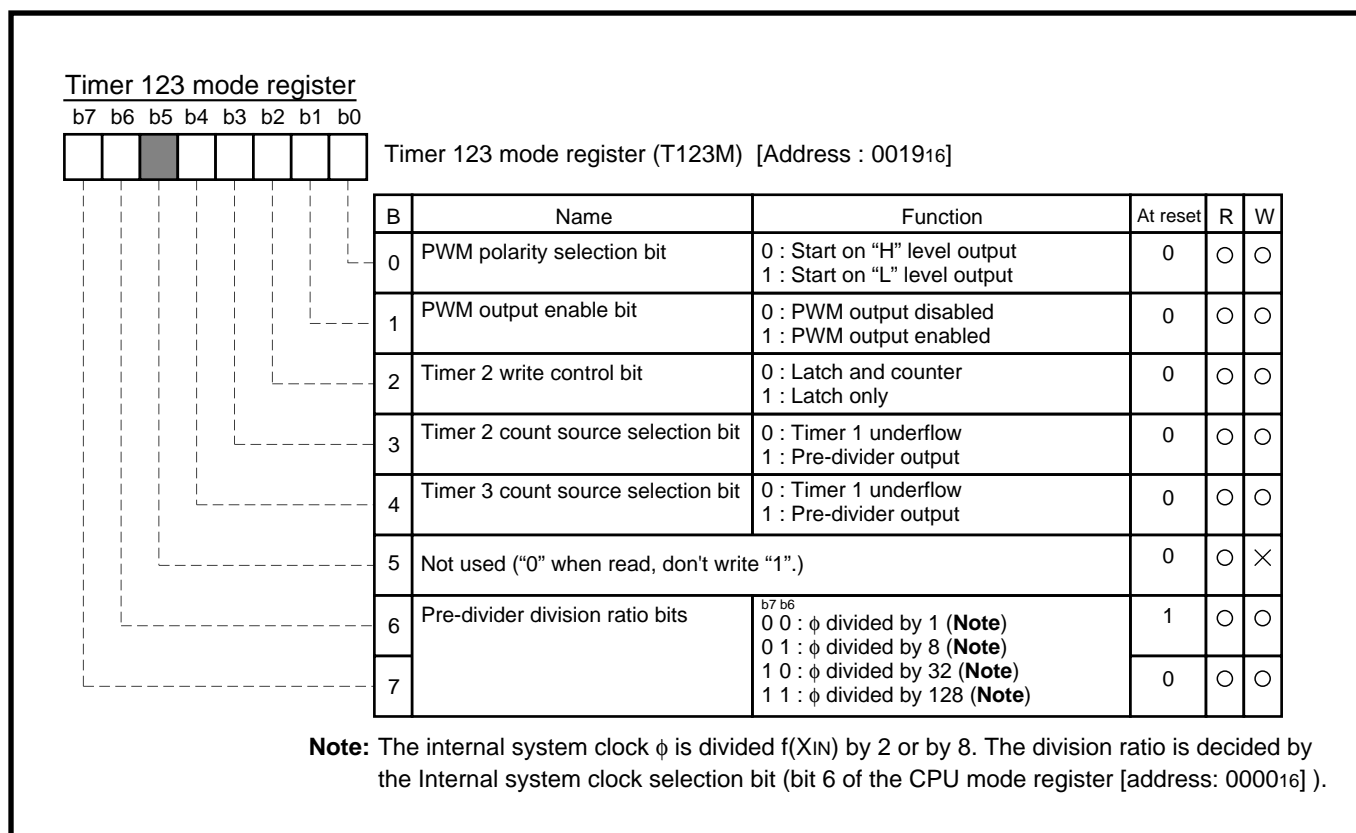


Fig. 3.5.16 Structure of Timer 123 mode register

APPENDIX

3.5 List of registers

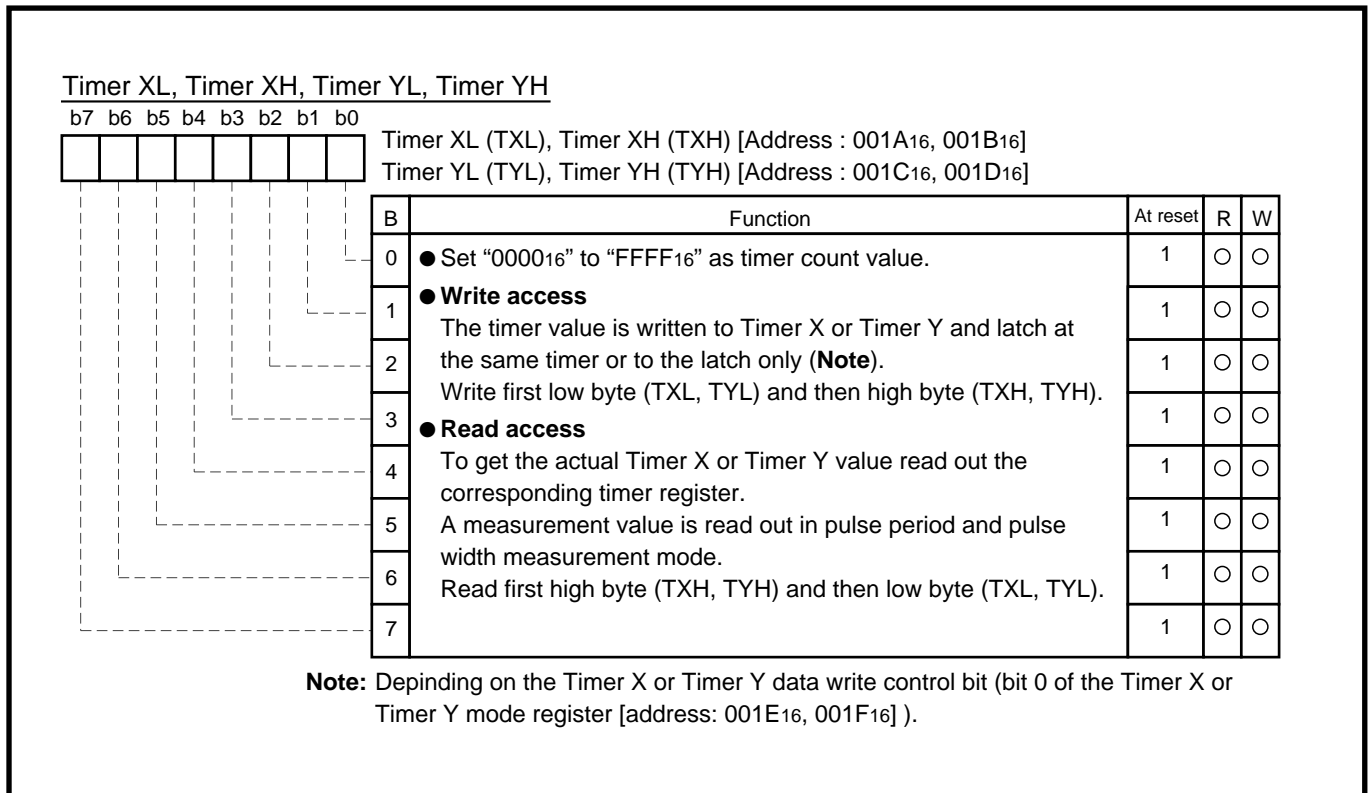


Fig. 3.5.17 Structure of Timer XL, Timer XH, Timer YL, Timer YH

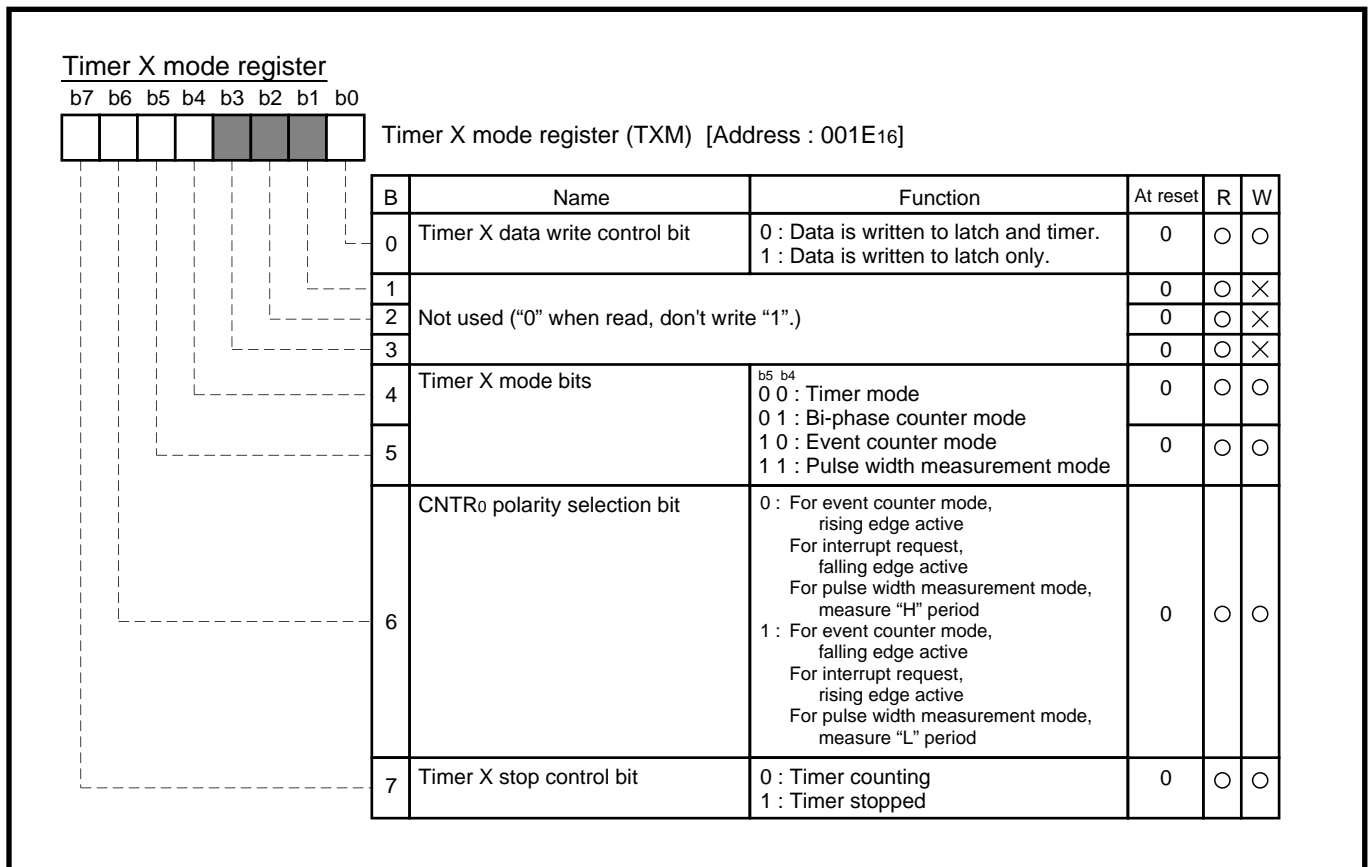


Fig. 3.5.18 Structure of Timer X mode register

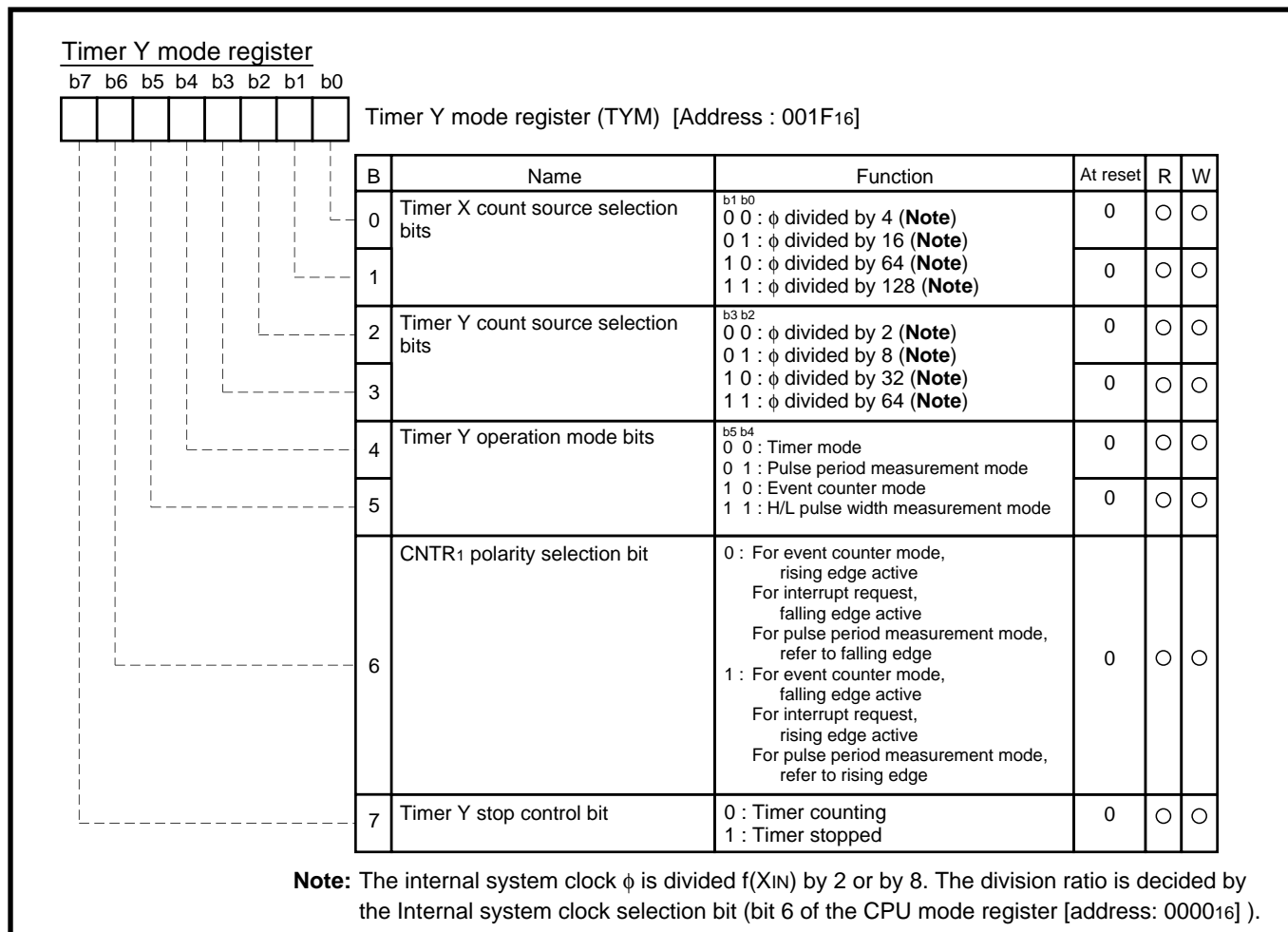


Fig. 3.5.19 Structure of Timer Y mode register

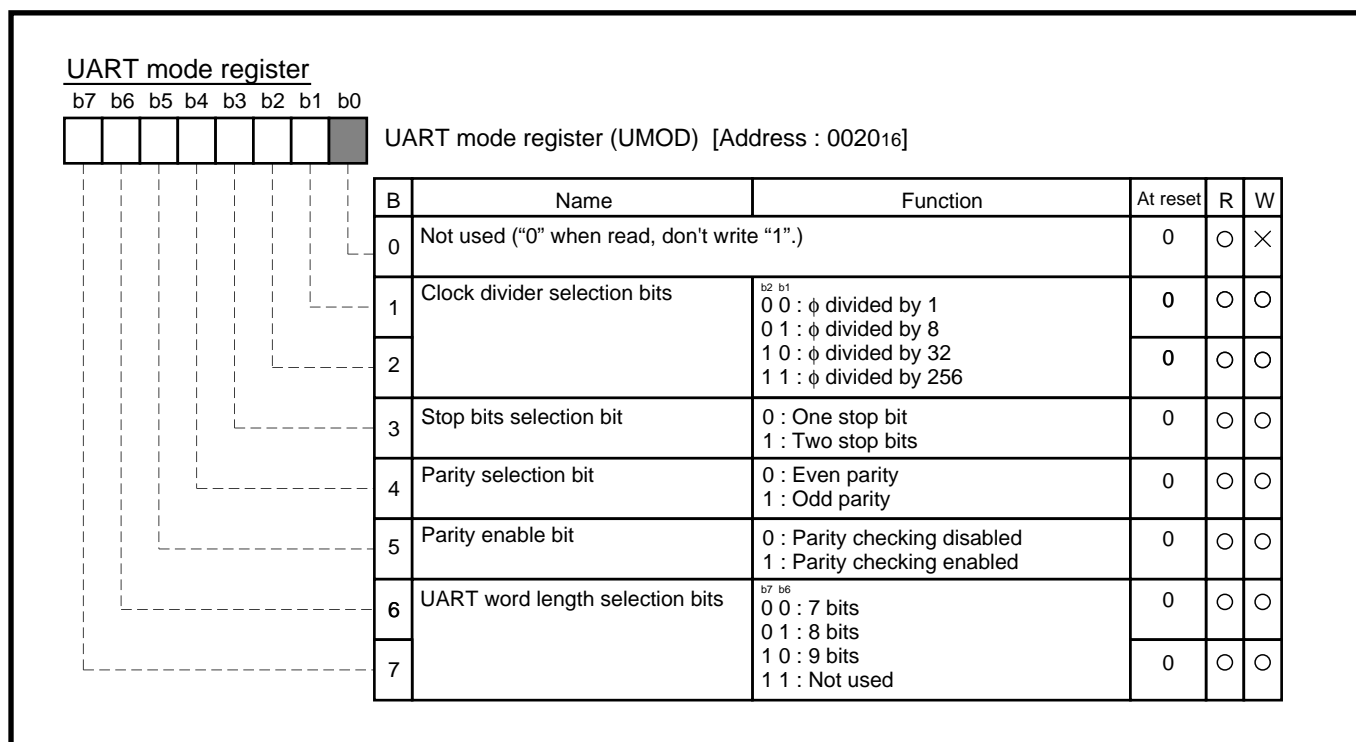


Fig. 3.5.20 Structure of UART mode register

APPENDIX

3.5 List of registers

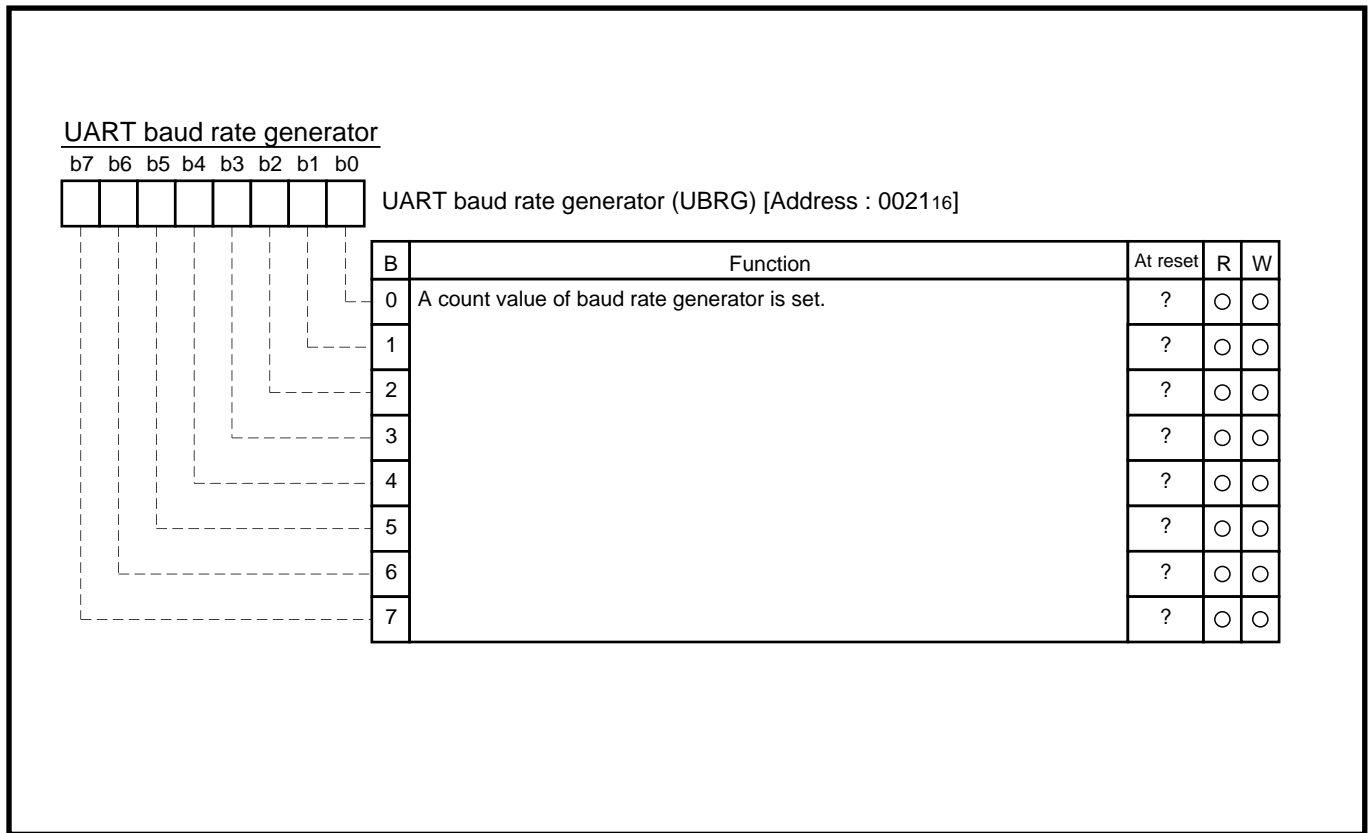


Fig. 3.5.21 Structure of UART baud rate generator

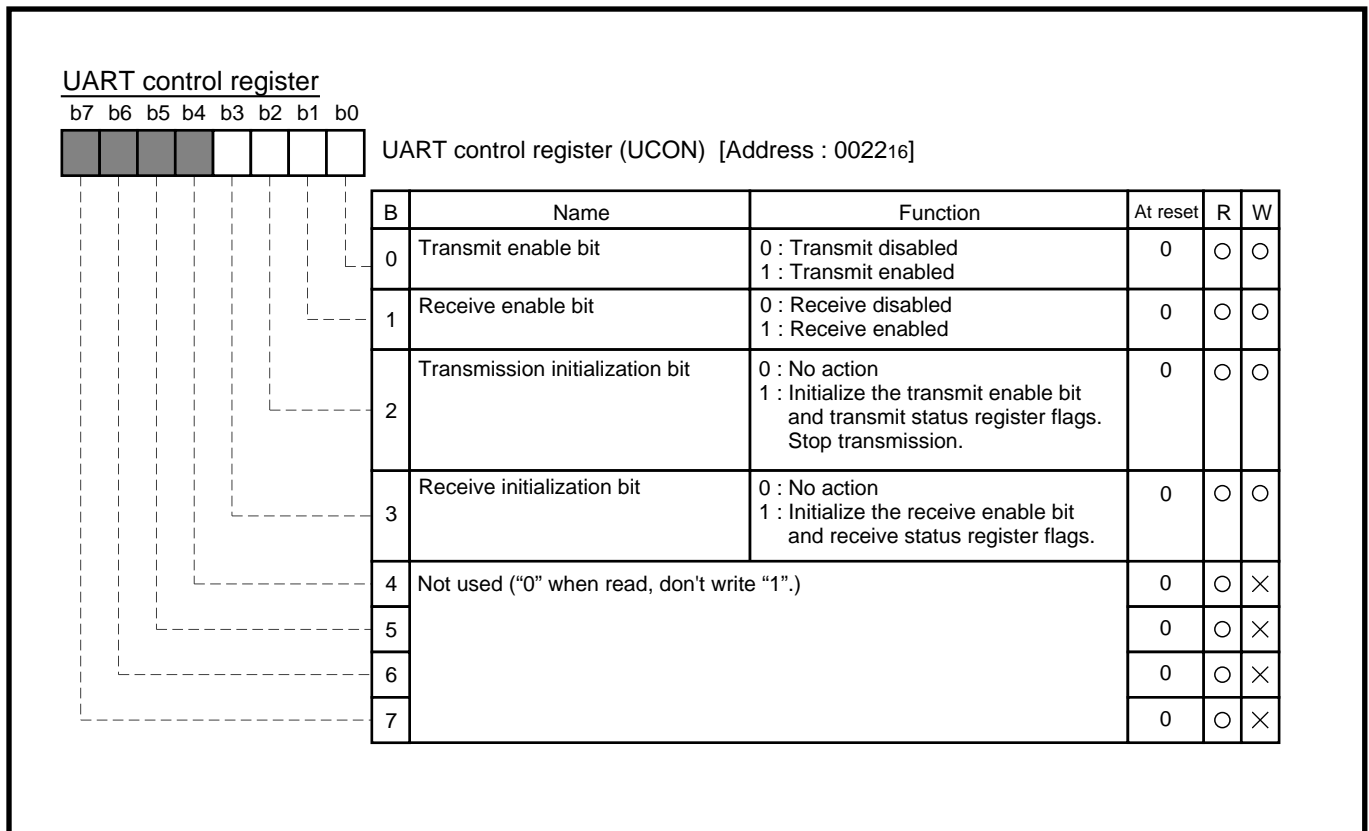


Fig. 3.5.22 Structure of UART control register

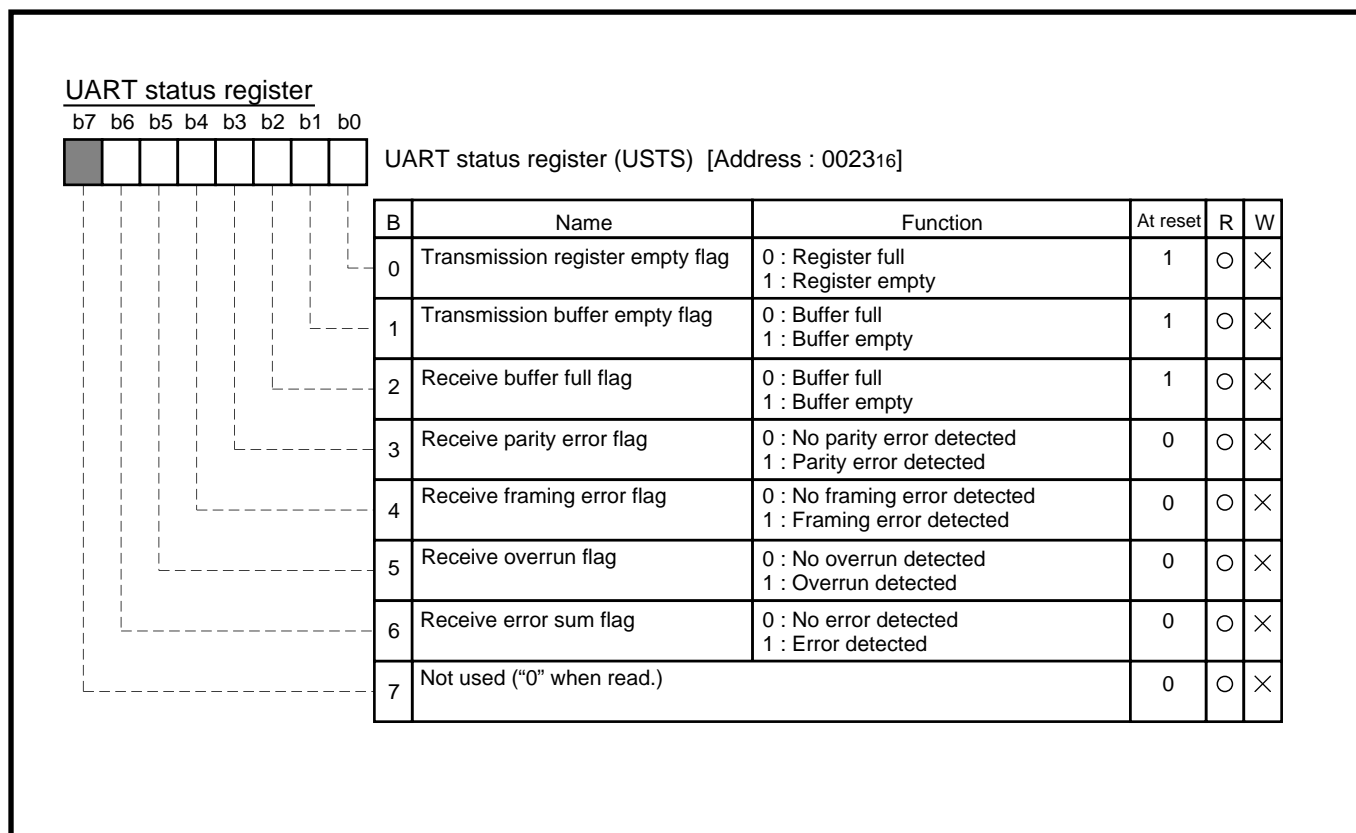


Fig. 3.5.23 Structure of UART status register

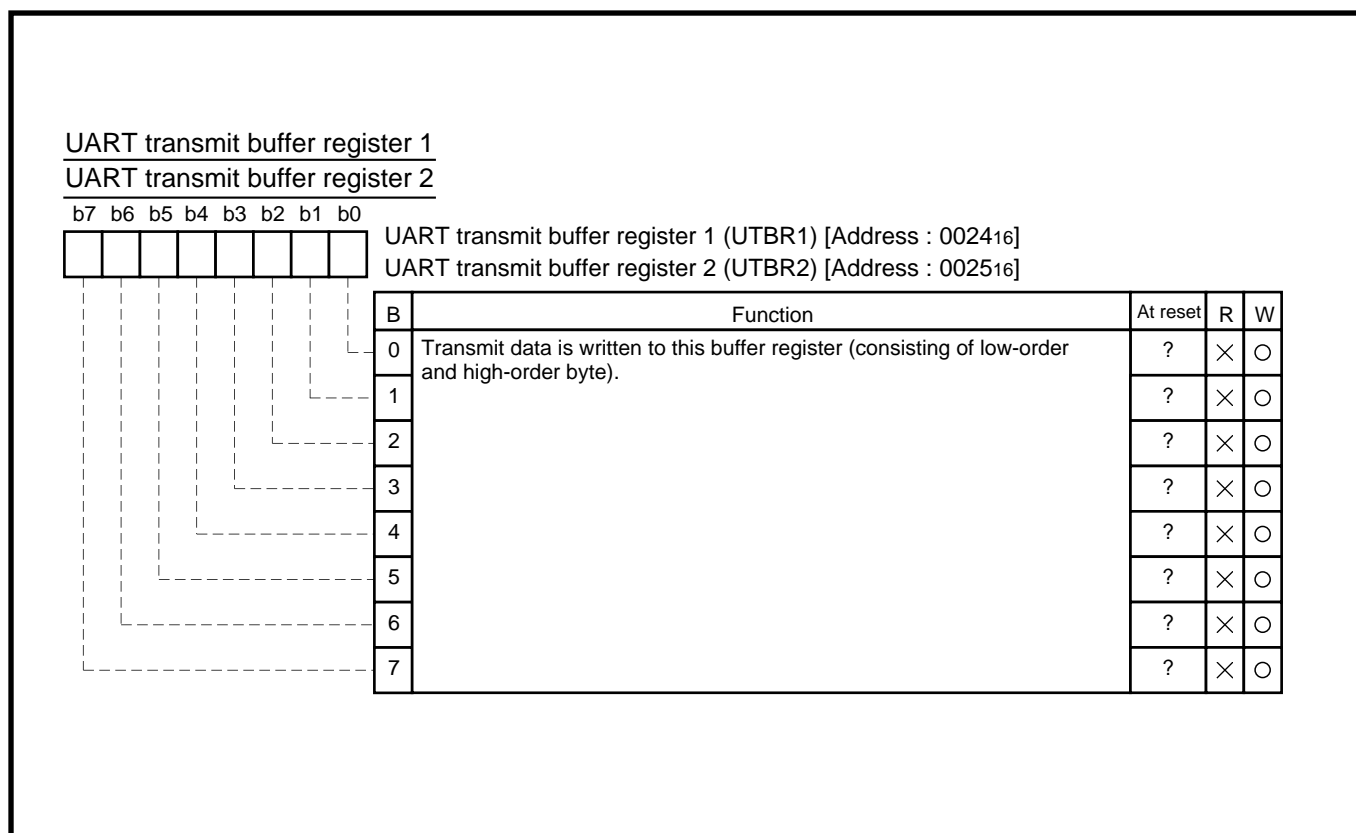


Fig. 3.5.24 Structure of UART transmit buffer register 1, 2

APPENDIX

3.5 List of registers

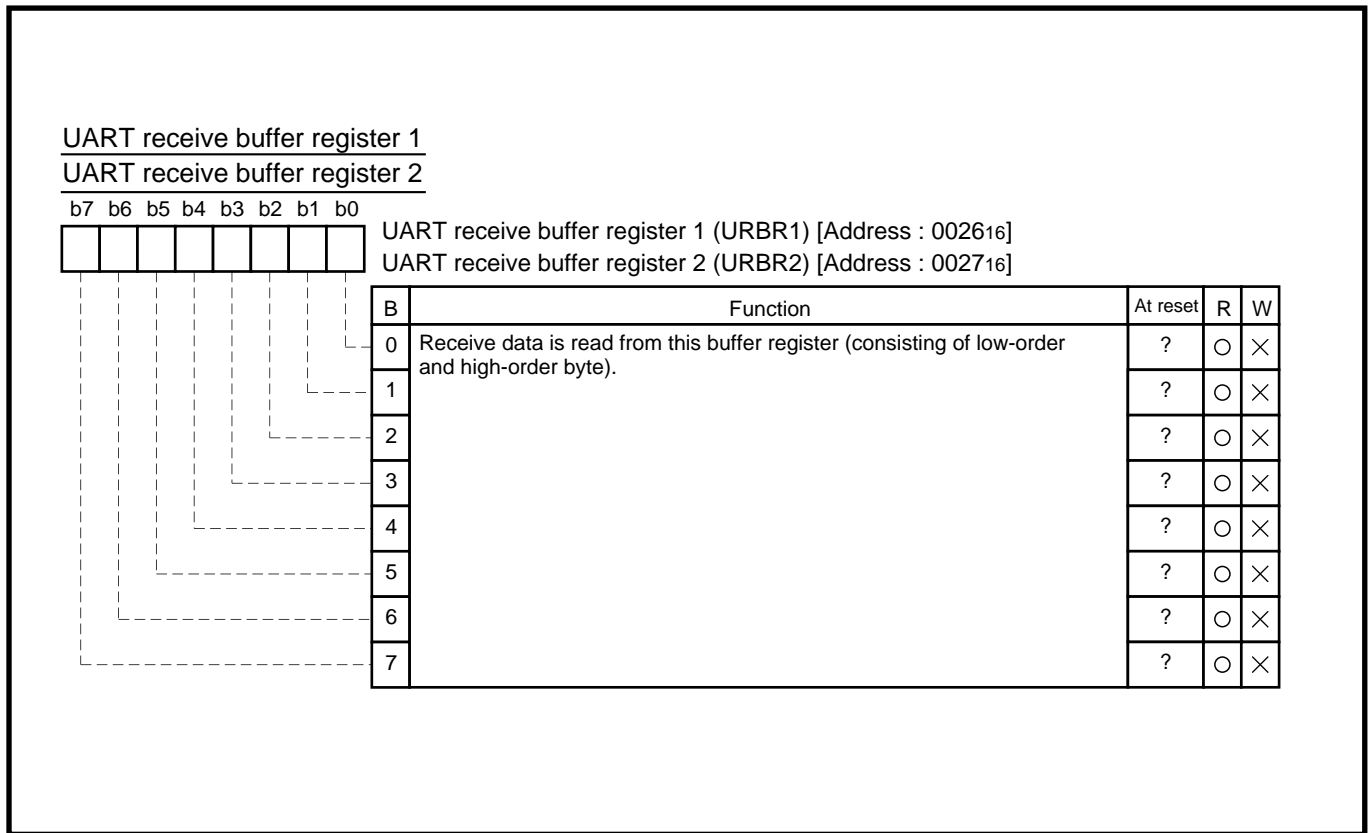


Fig. 3.5.25 Structure of UART receive buffer register 1, 2

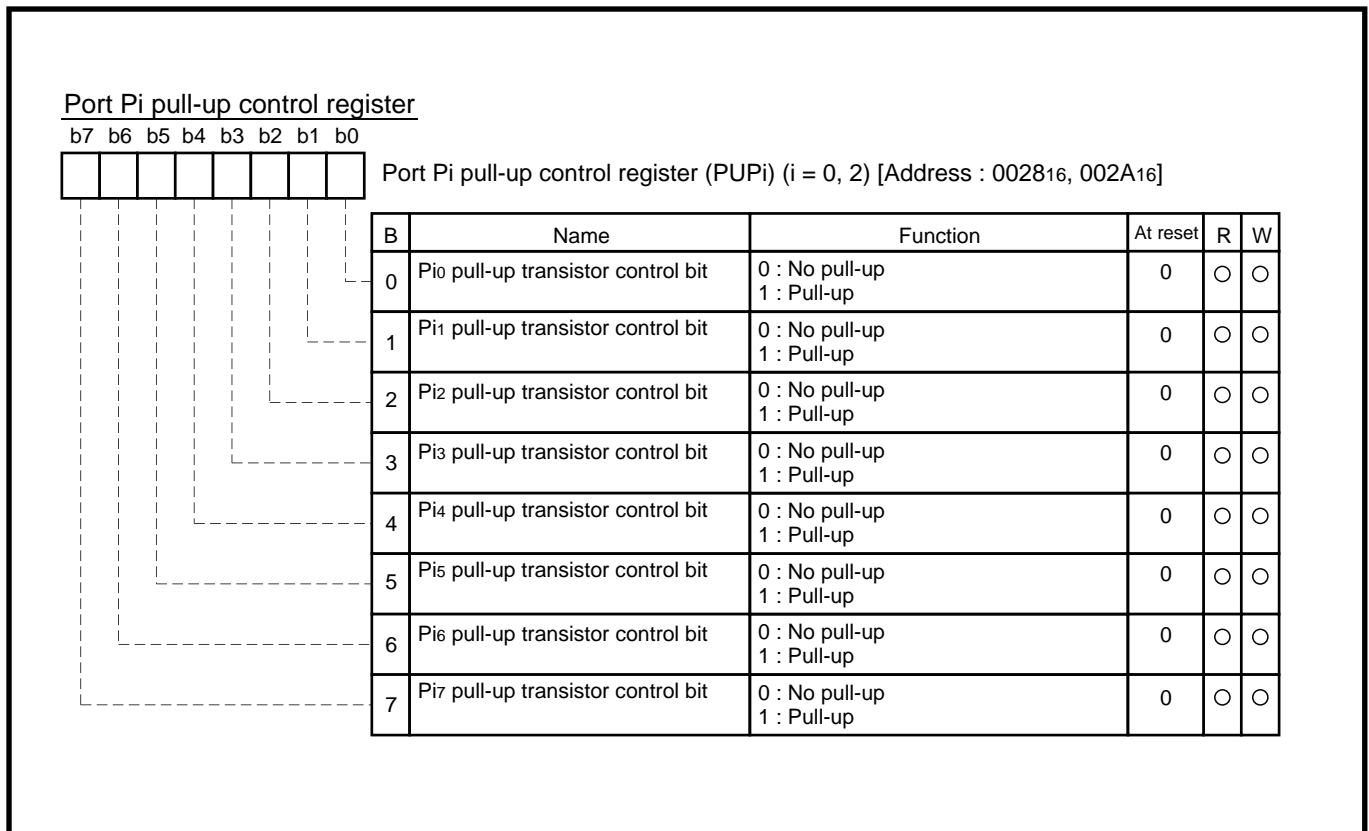


Fig. 3.5.26 Structure of Port Pi pull-up control register (i = 0, 2)

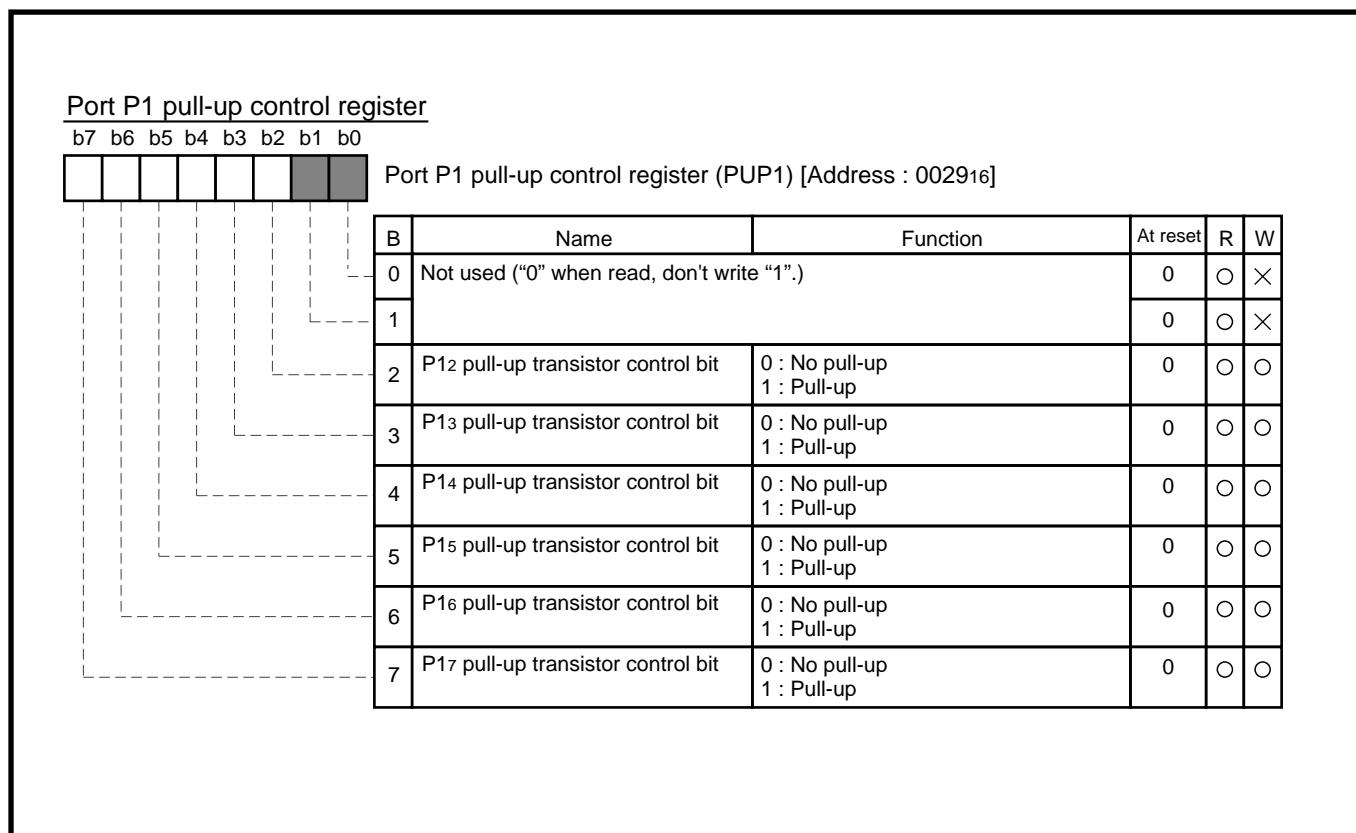


Fig. 3.5.27 Structure of Port P1 pull-up control register

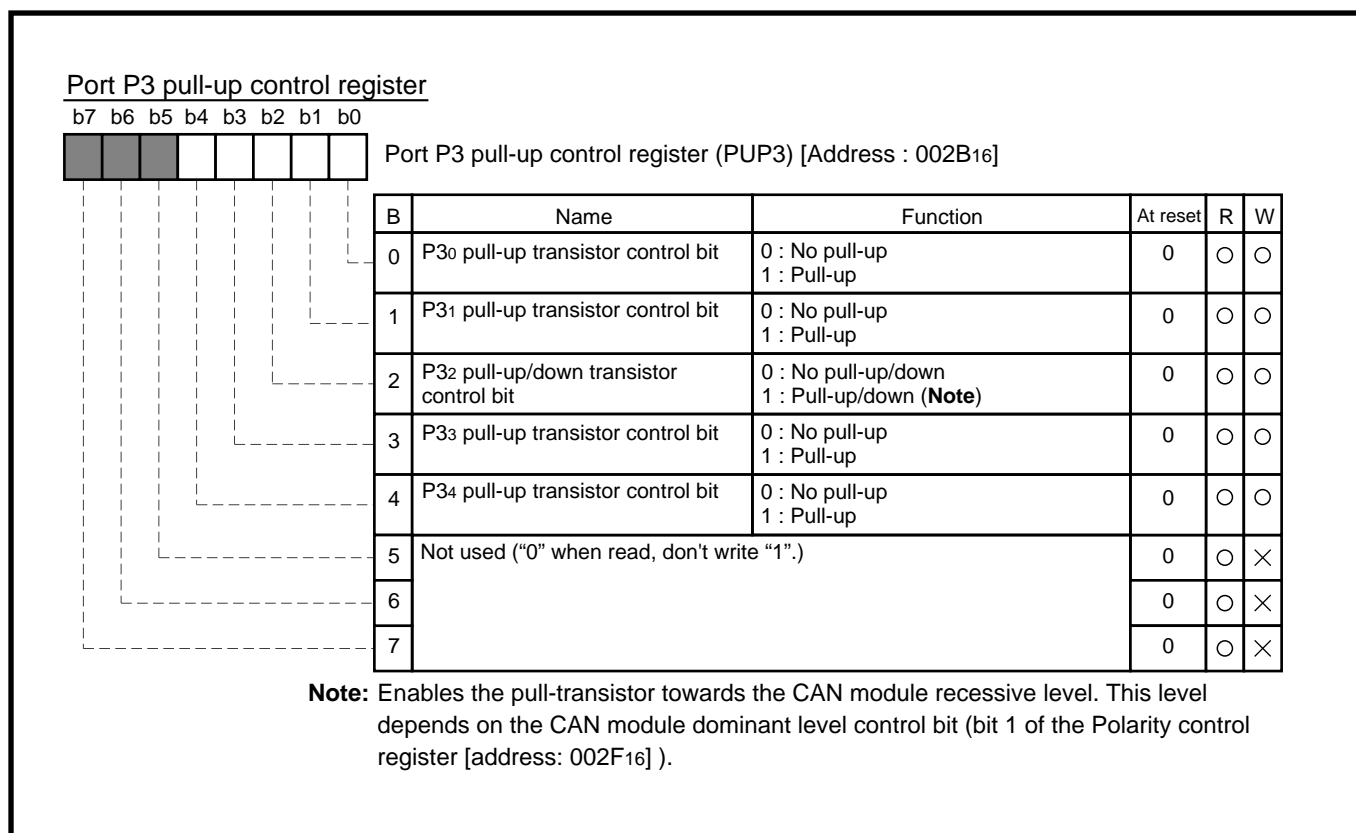


Fig. 3.5.28 Structure of Port P3 pull-up control register

APPENDIX

3.5 List of registers

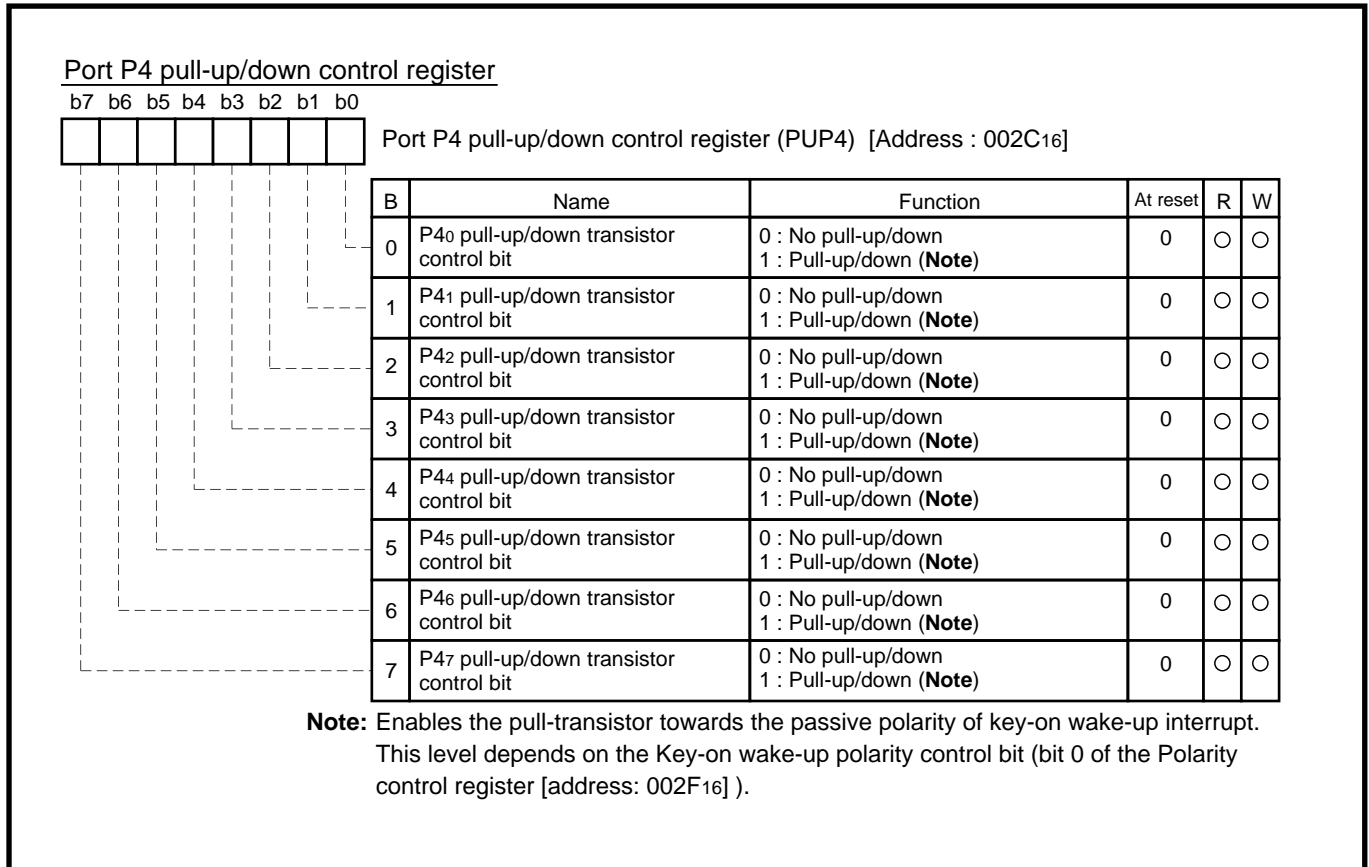


Fig. 3.5.29 Structure of Port P4 pull-up/down control register

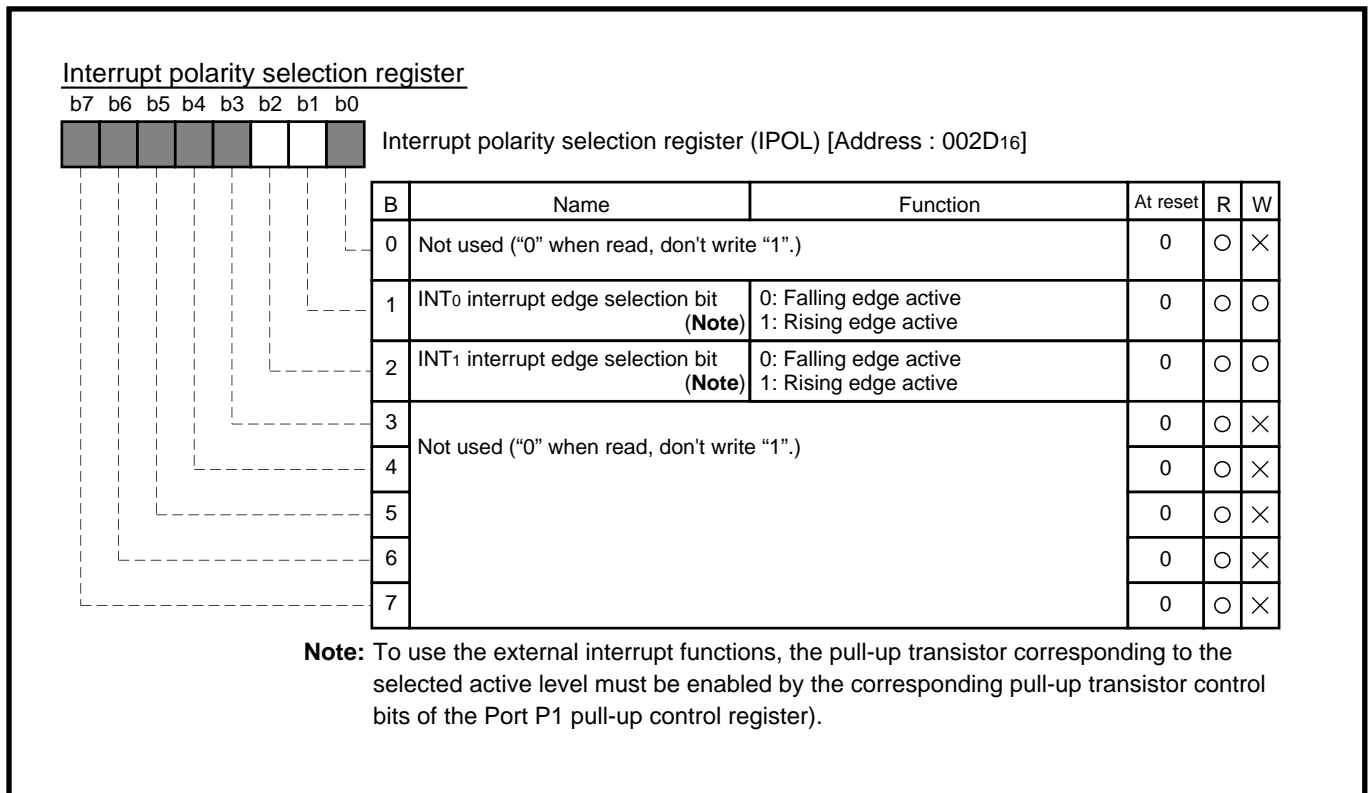


Fig. 3.5.30 Structure of Interrupt polarity selection register

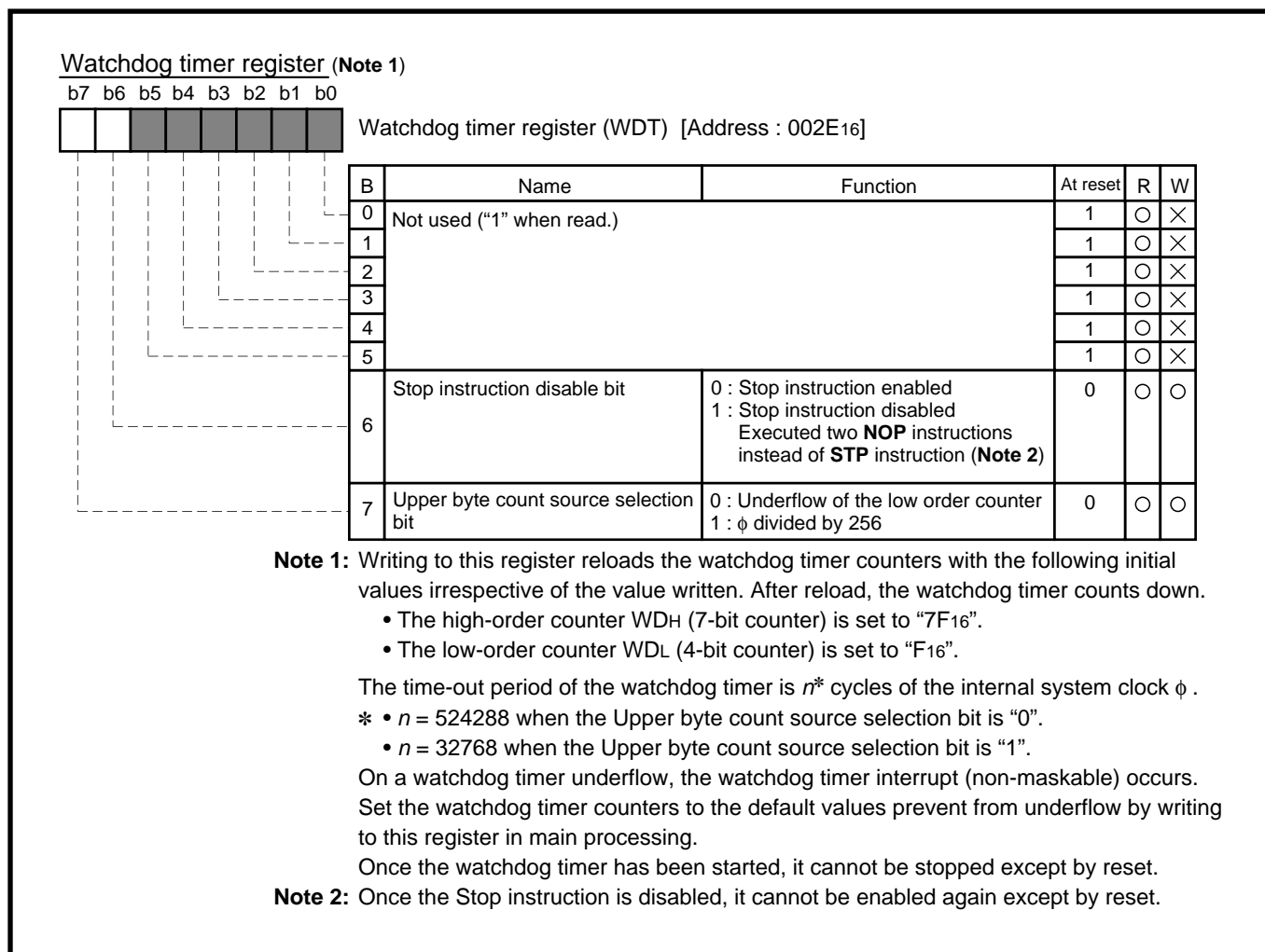


Fig. 3.5.31 Structure of Watchdog timer register

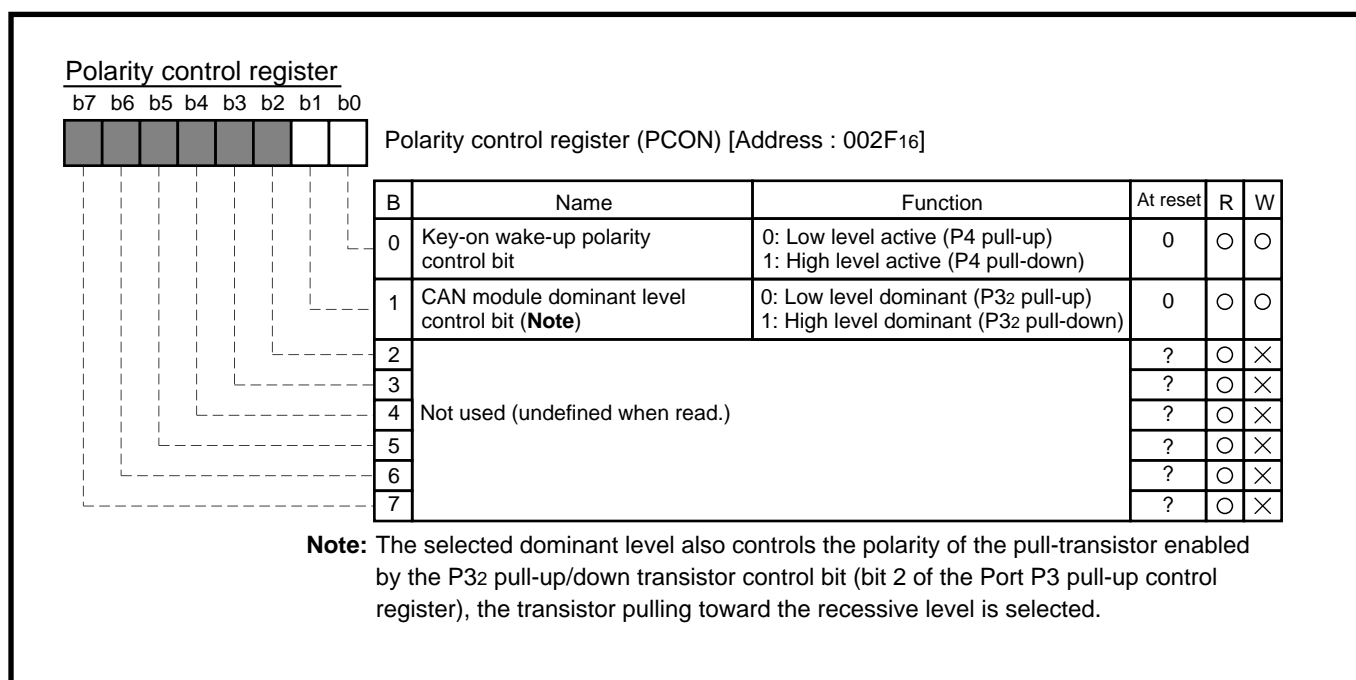


Fig. 3.5.32 Structure of Polarity control register

APPENDIX

3.5 List of registers

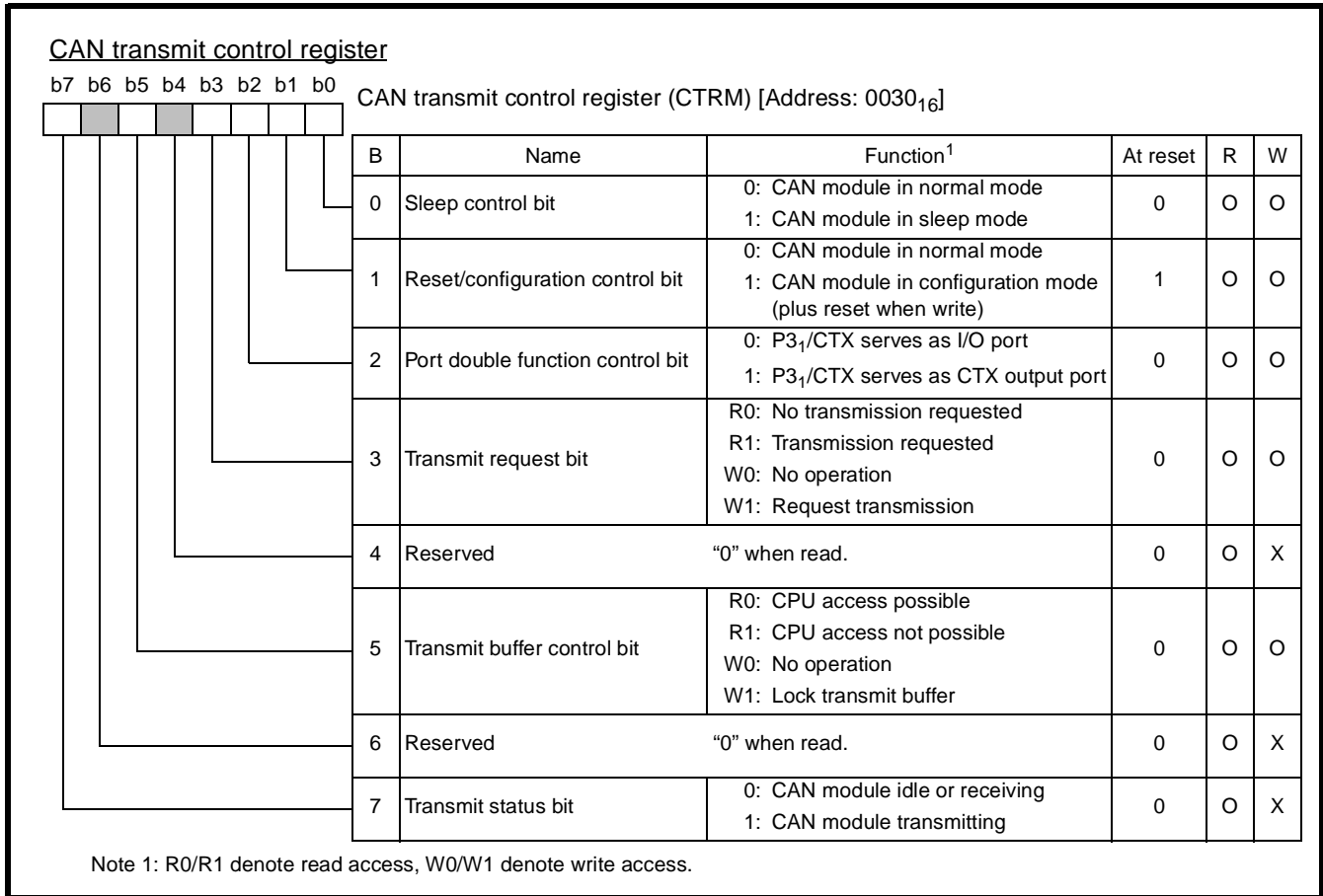


Fig. 3.5.33 Structure of CAN transmit control register (CTRM)

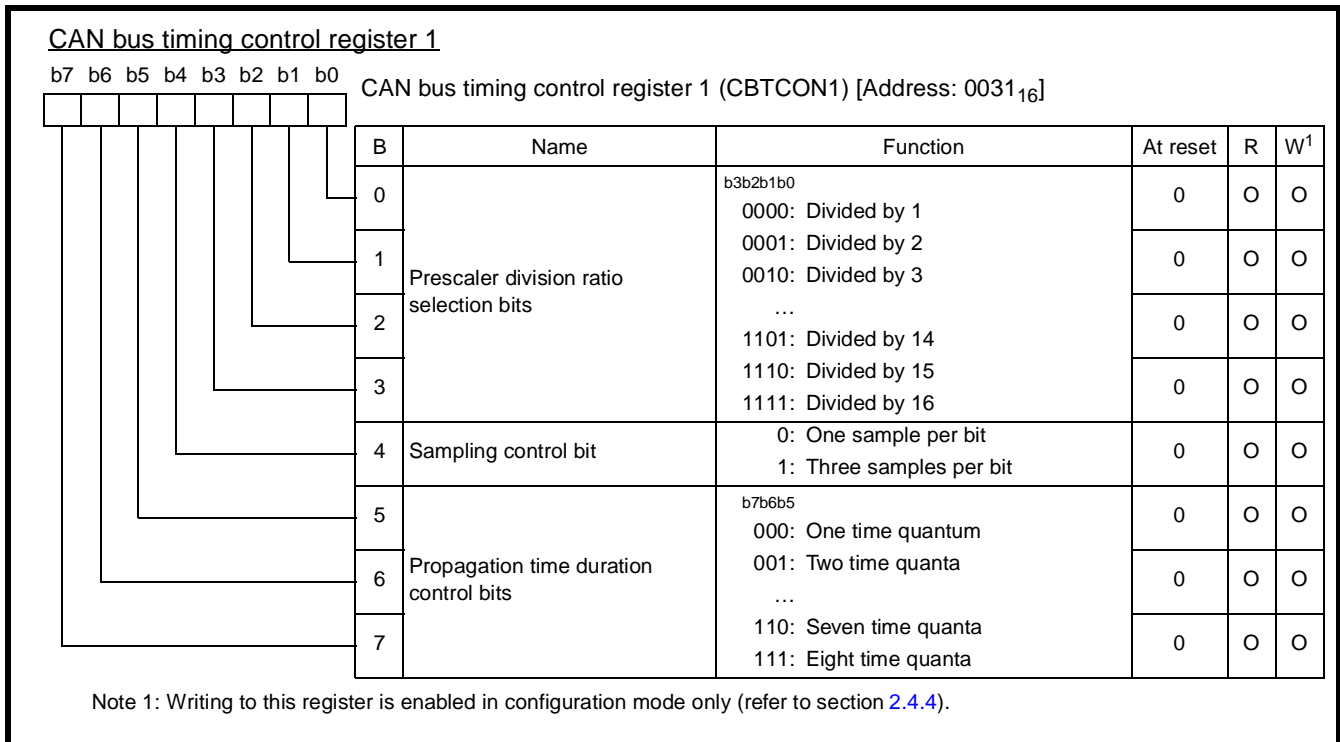


Fig. 3.5.34 Structure of CAN bus timing control register 1 (CBTCON1)

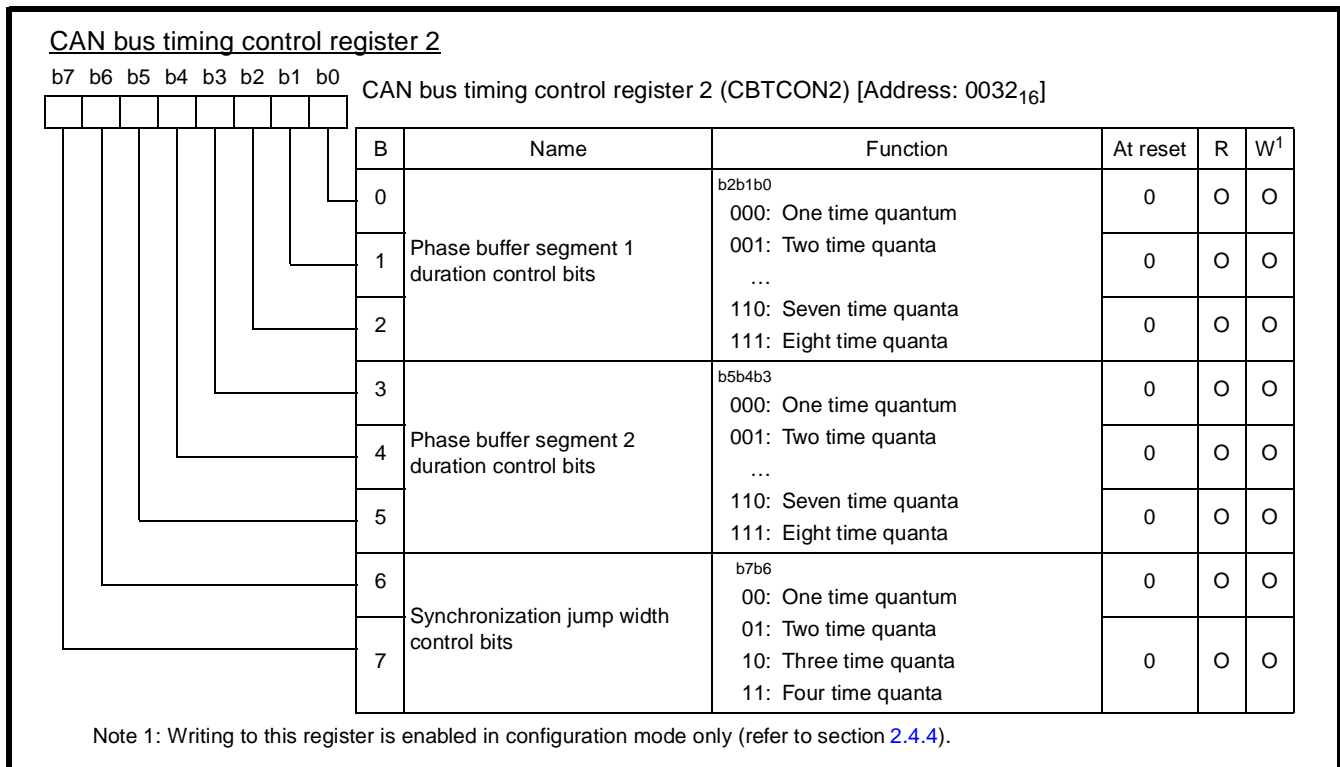


Fig. 3.5.35 Structure of CAN bus timing control register 2 (CBTCON2)

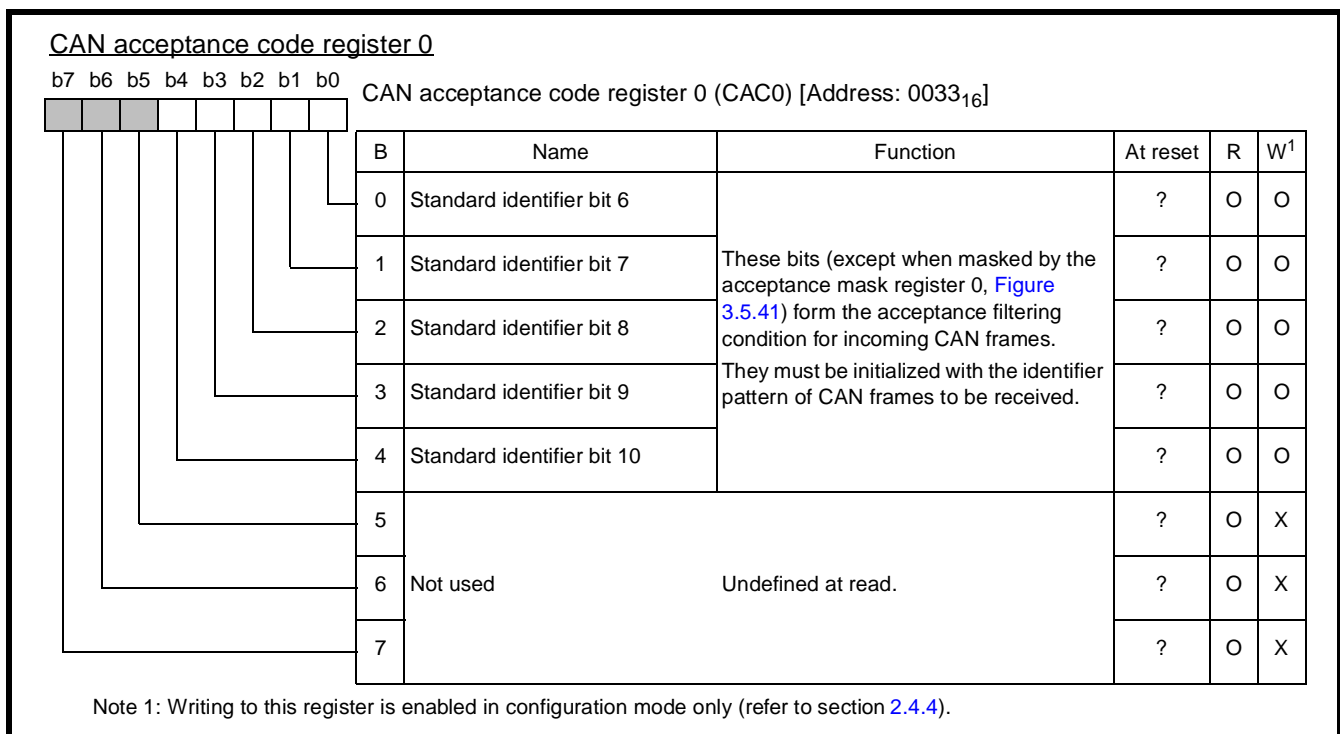


Fig. 3.5.36 Structure of CAN acceptance code register 0 (CAC0)

APPENDIX

3.5 List of registers

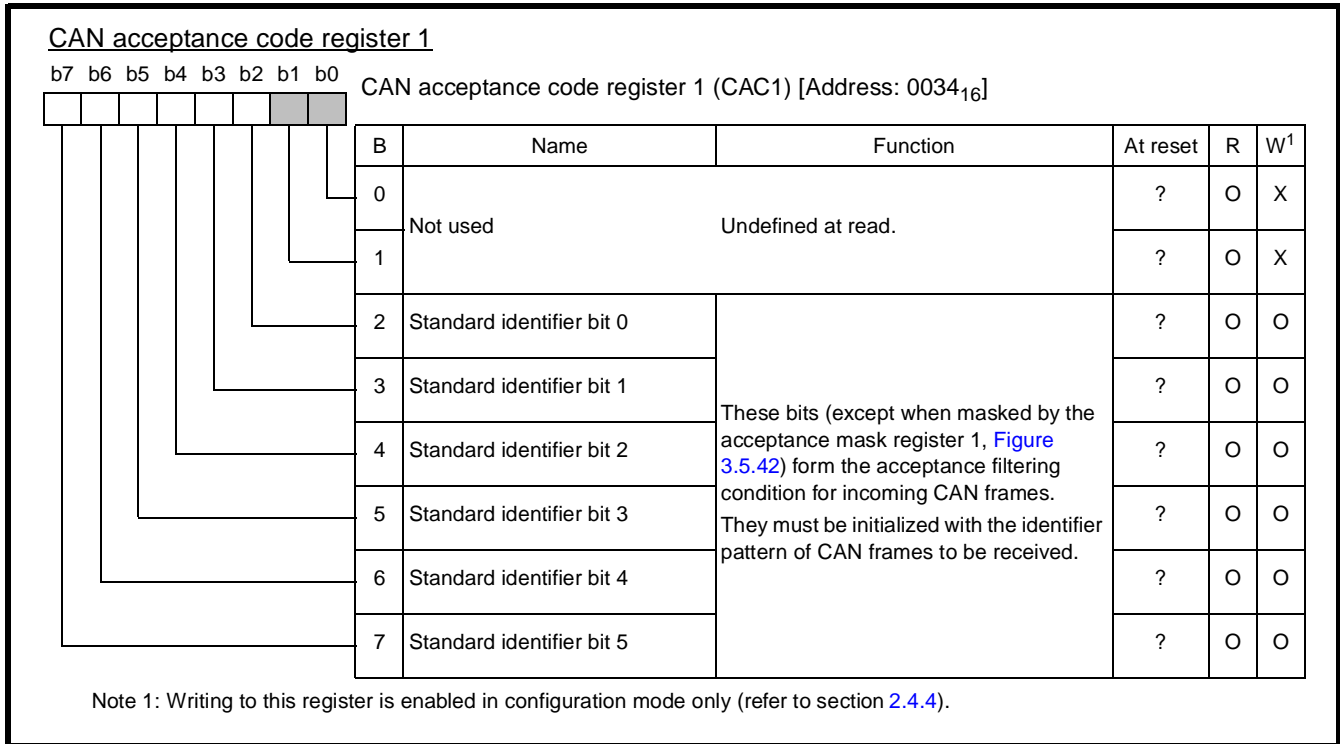


Fig. 3.5.37 Structure of CAN acceptance code register 1 (CAC1)

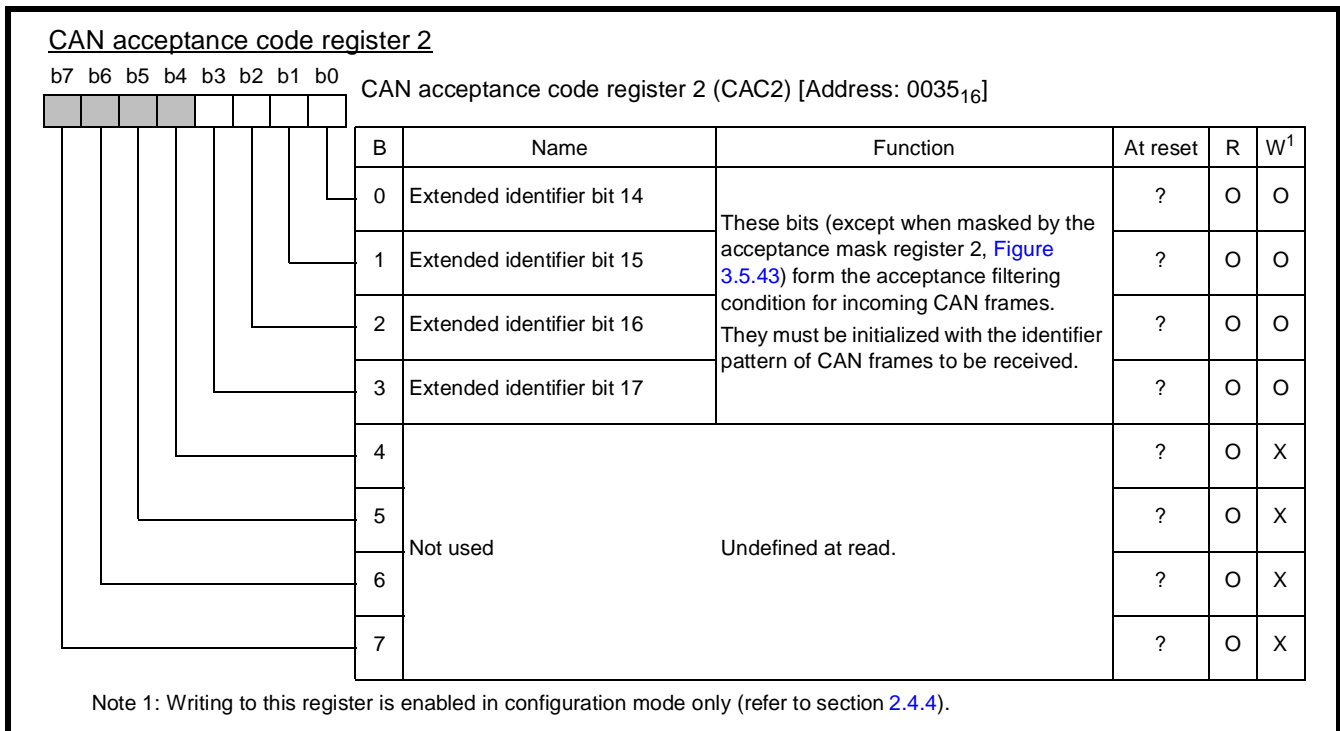


Fig. 3.5.38 Structure of CAN acceptance code register 2 (CAC2)

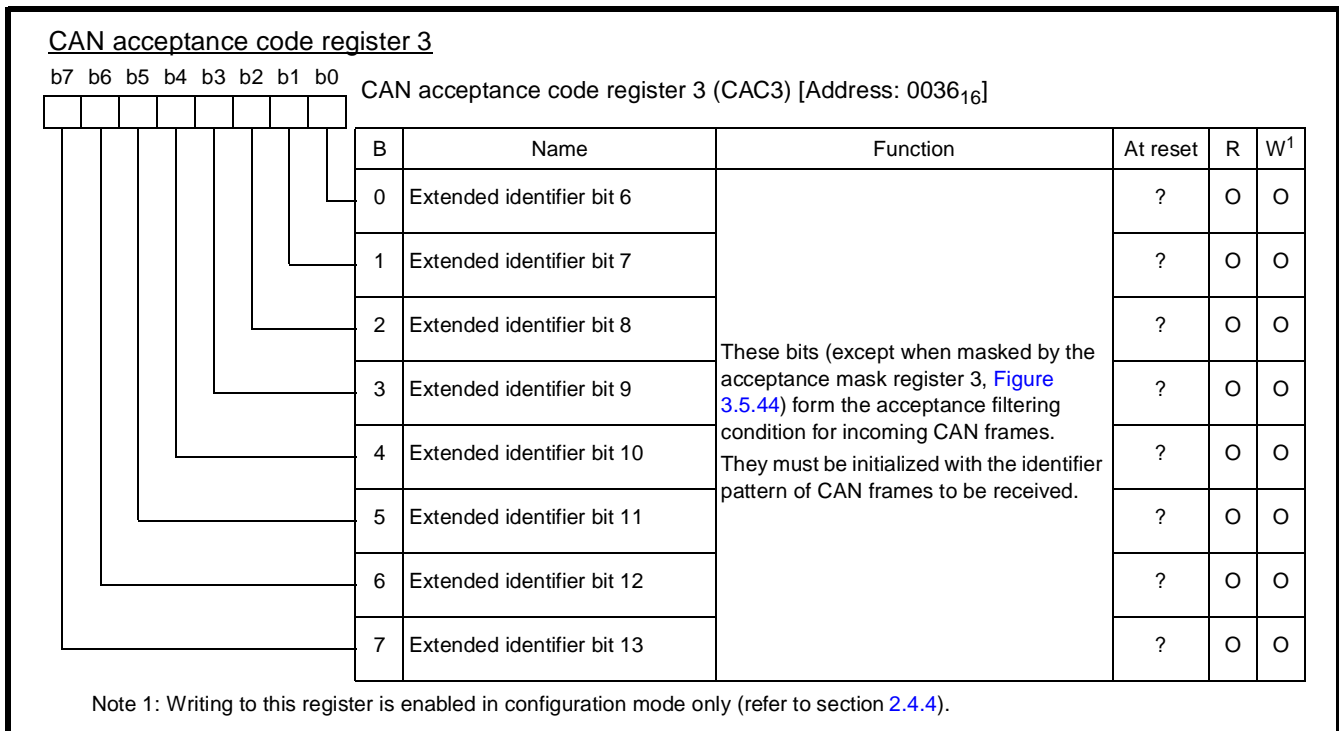


Fig. 3.5.39 Structure of CAN acceptance code register 3 (CAC3)

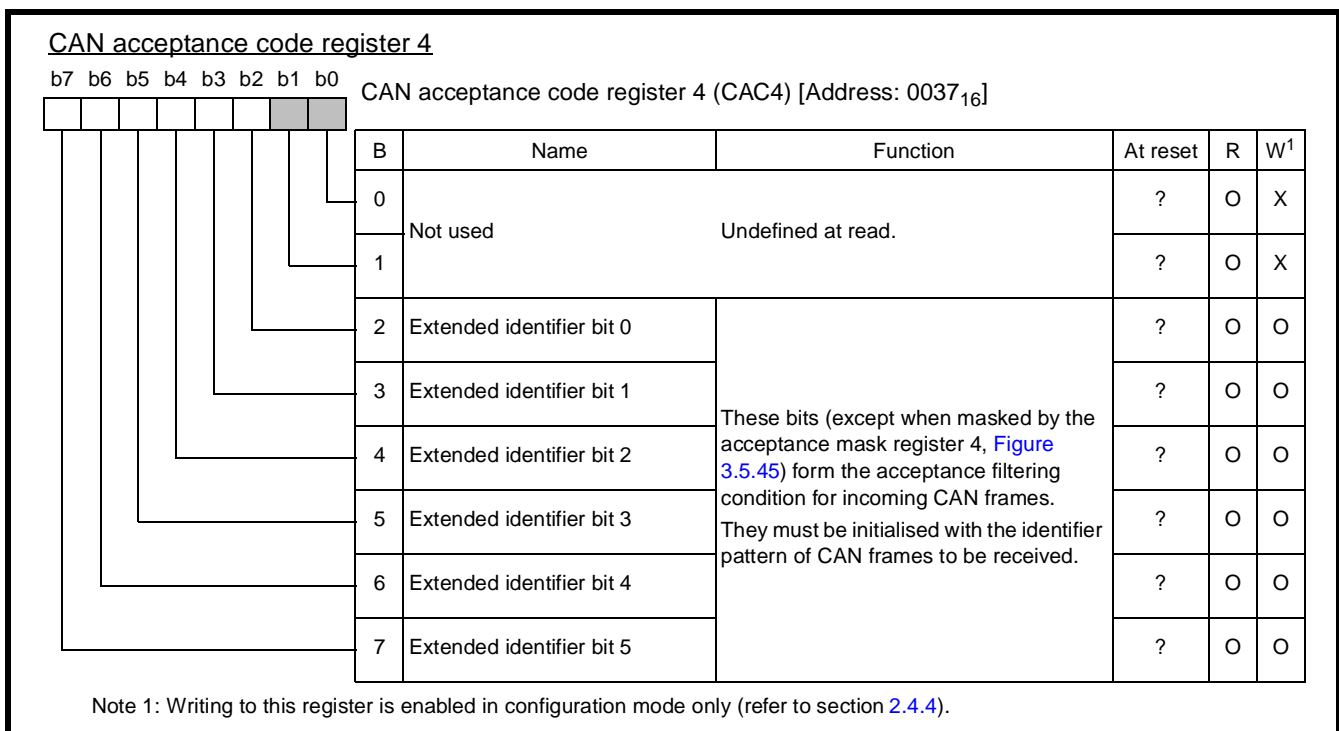


Fig. 3.5.40 Structure of CAN acceptance code register 4 (CAC4)

APPENDIX

3.5 List of registers

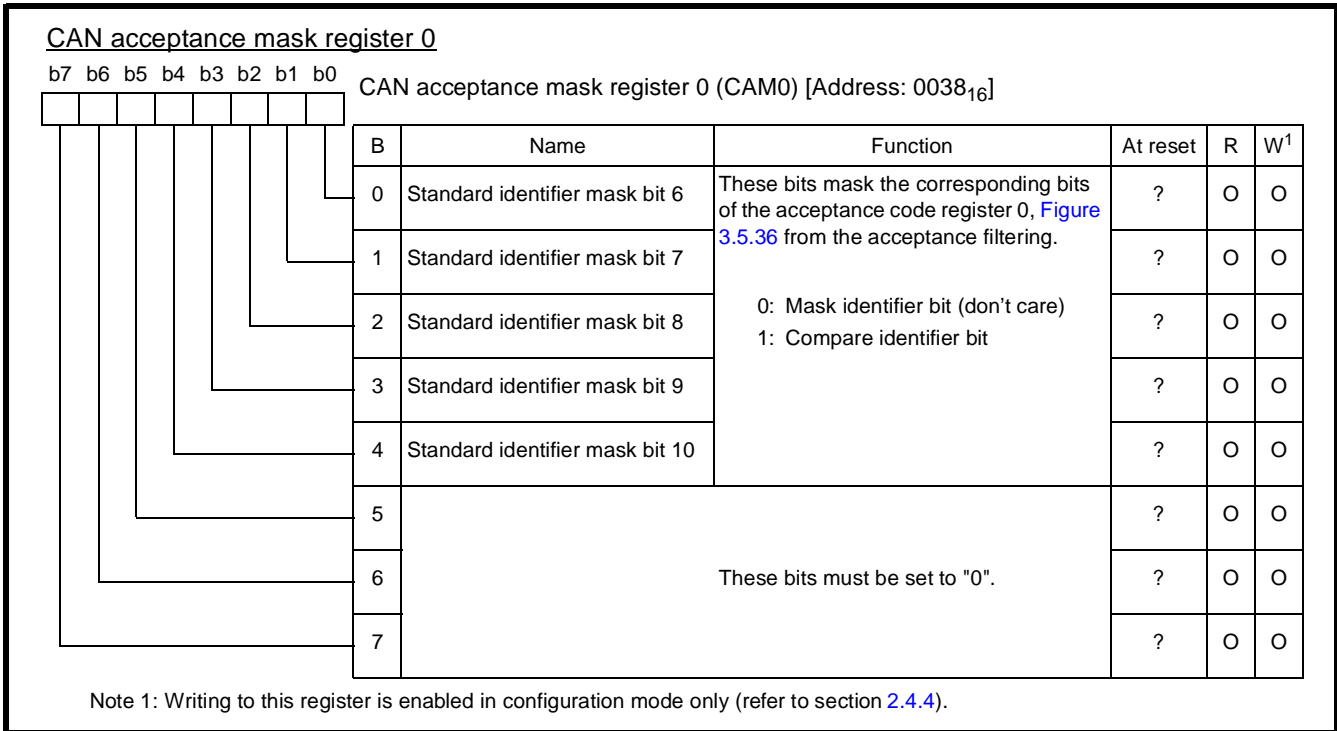


Fig. 3.5.41 Structure of CAN acceptance mask register 0 (CAM0)

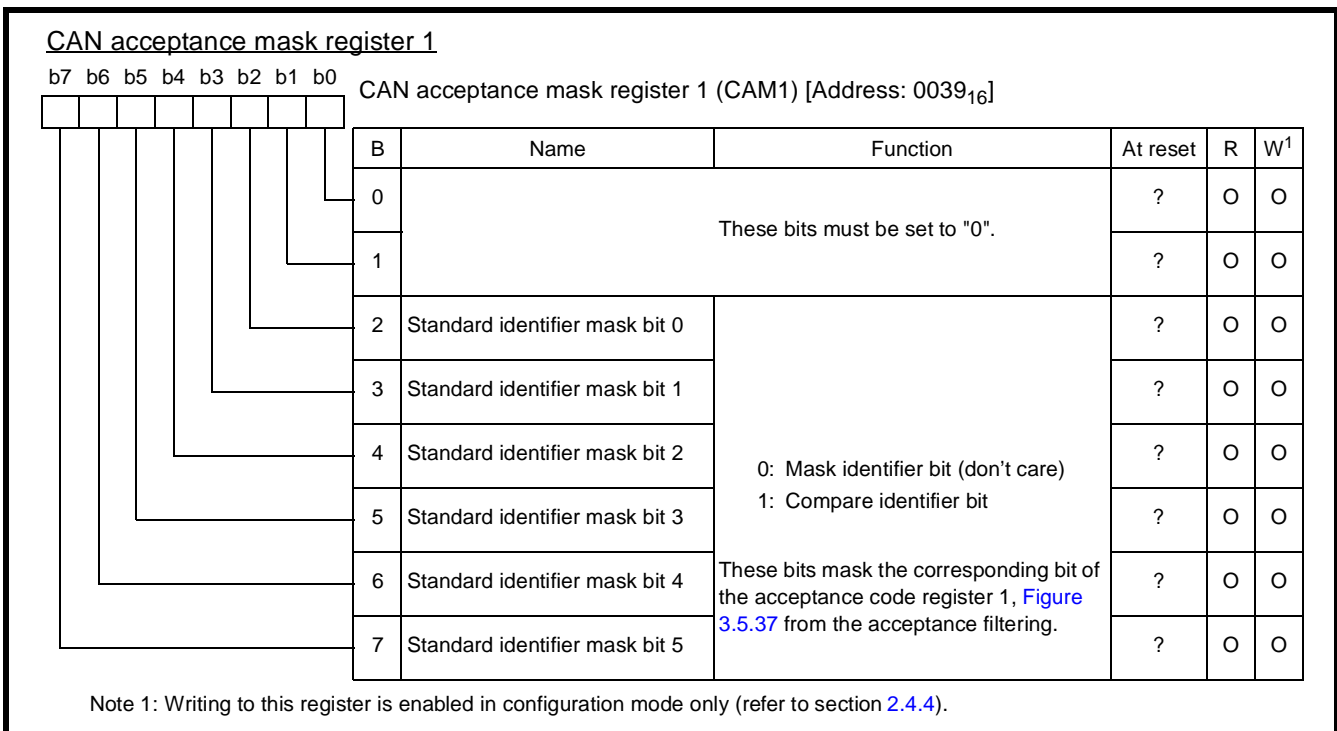


Fig. 3.5.42 Structure of CAN acceptance mask register 1 (CAM1)

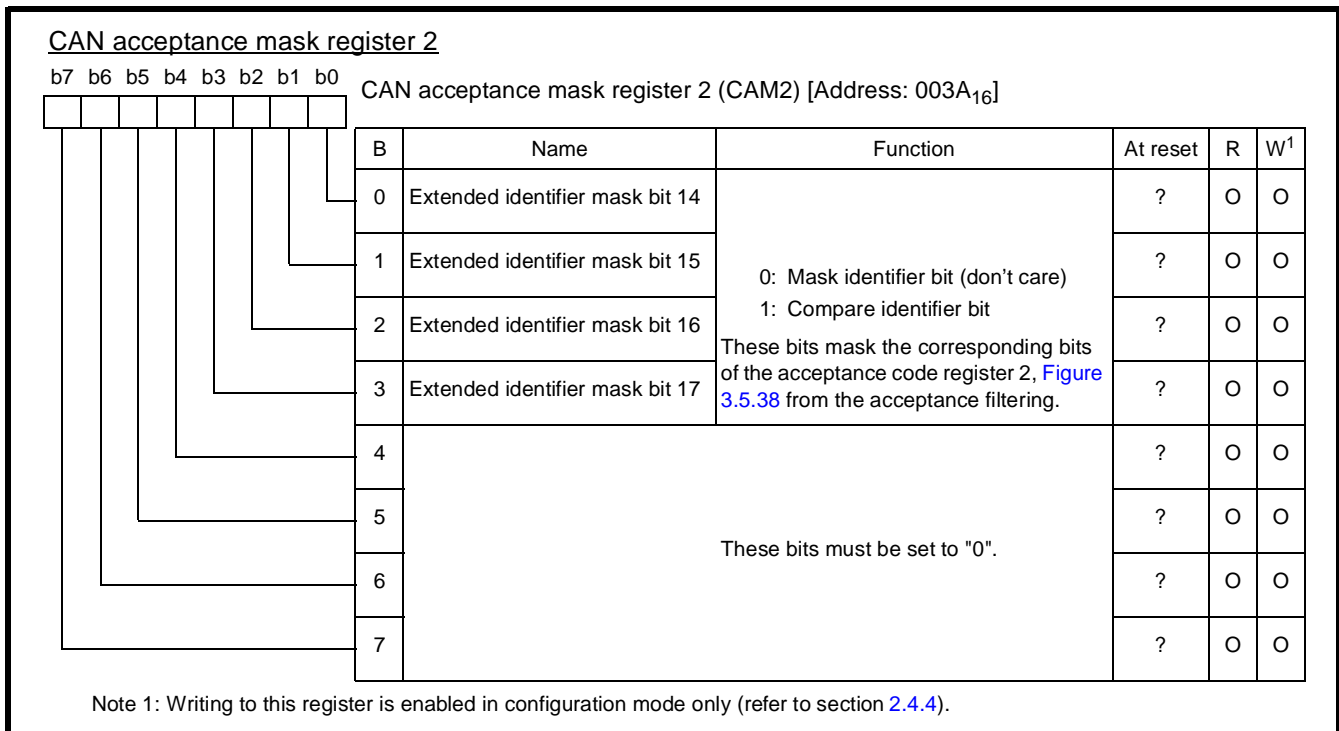


Fig. 3.5.43 Structure of CAN acceptance mask register 2 (CAM2)

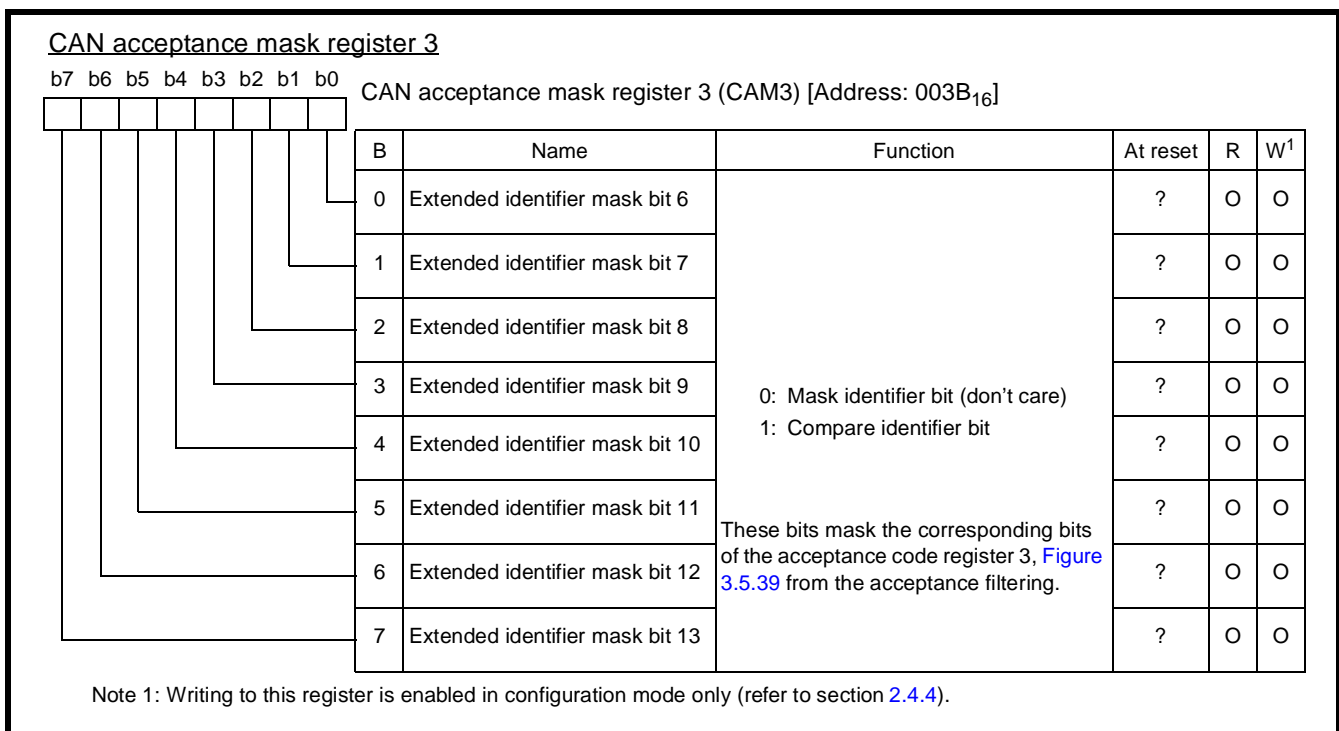


Fig. 3.5.44 Structure of CAN acceptance mask register 3 (CAM3)

APPENDIX

3.5 List of registers

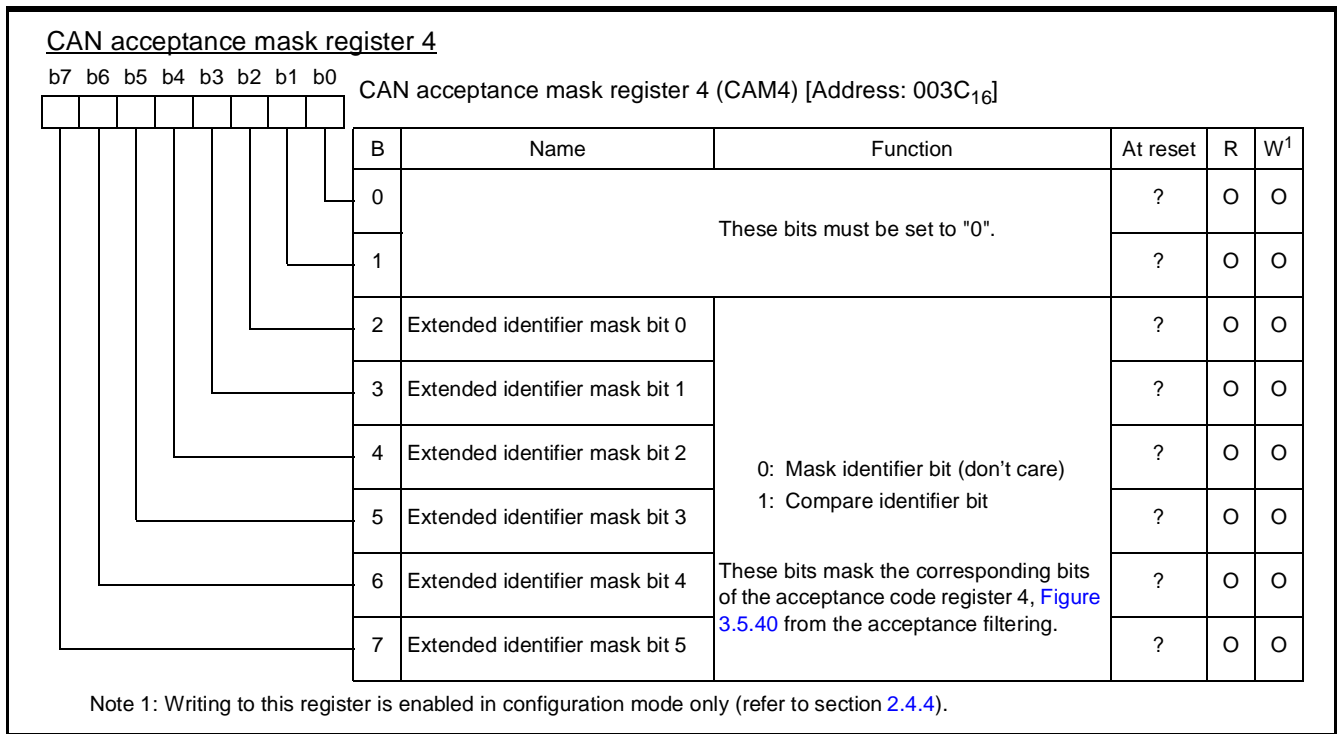


Fig. 3.5.45 Structure of CAN acceptance mask register 4 (CAM4)

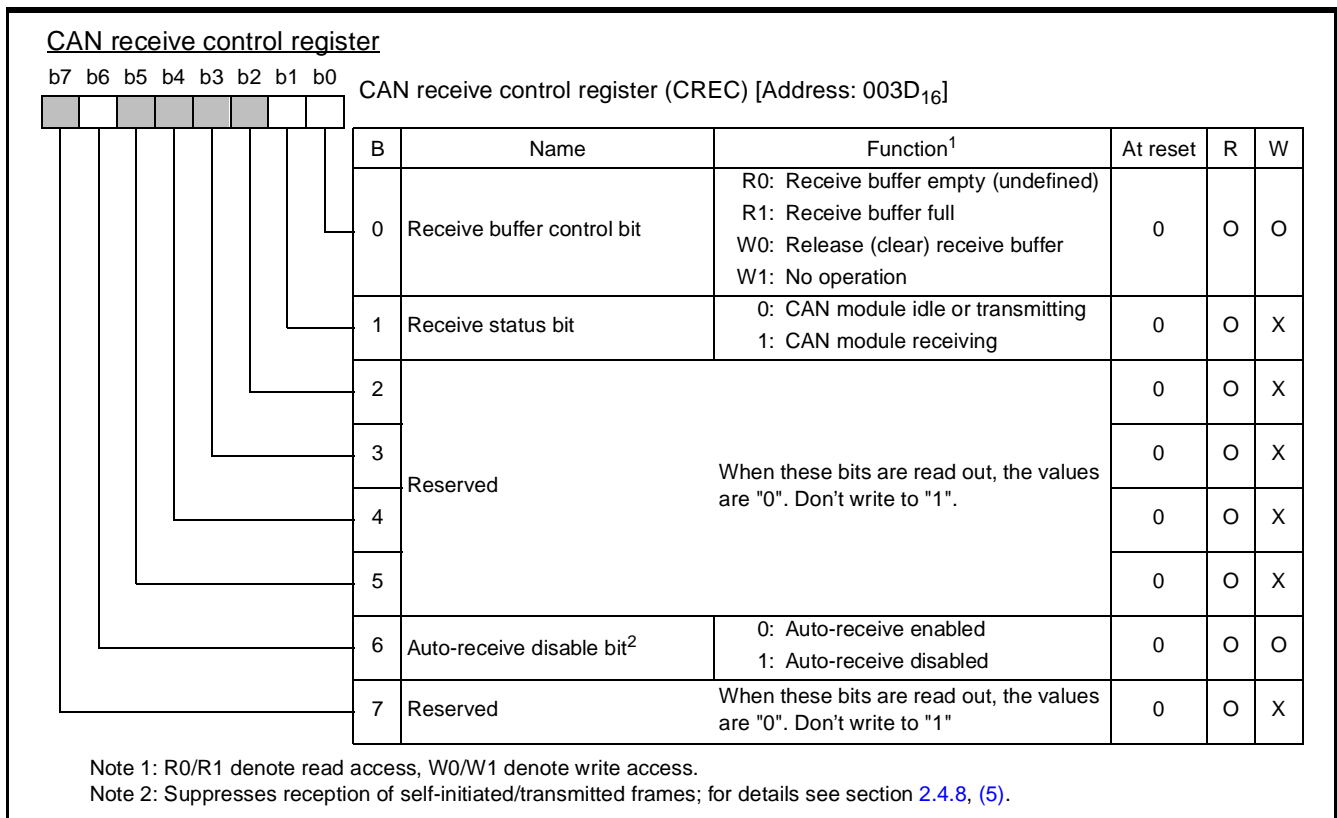


Fig. 3.5.46 Structure of CAN receive control register (CREC)

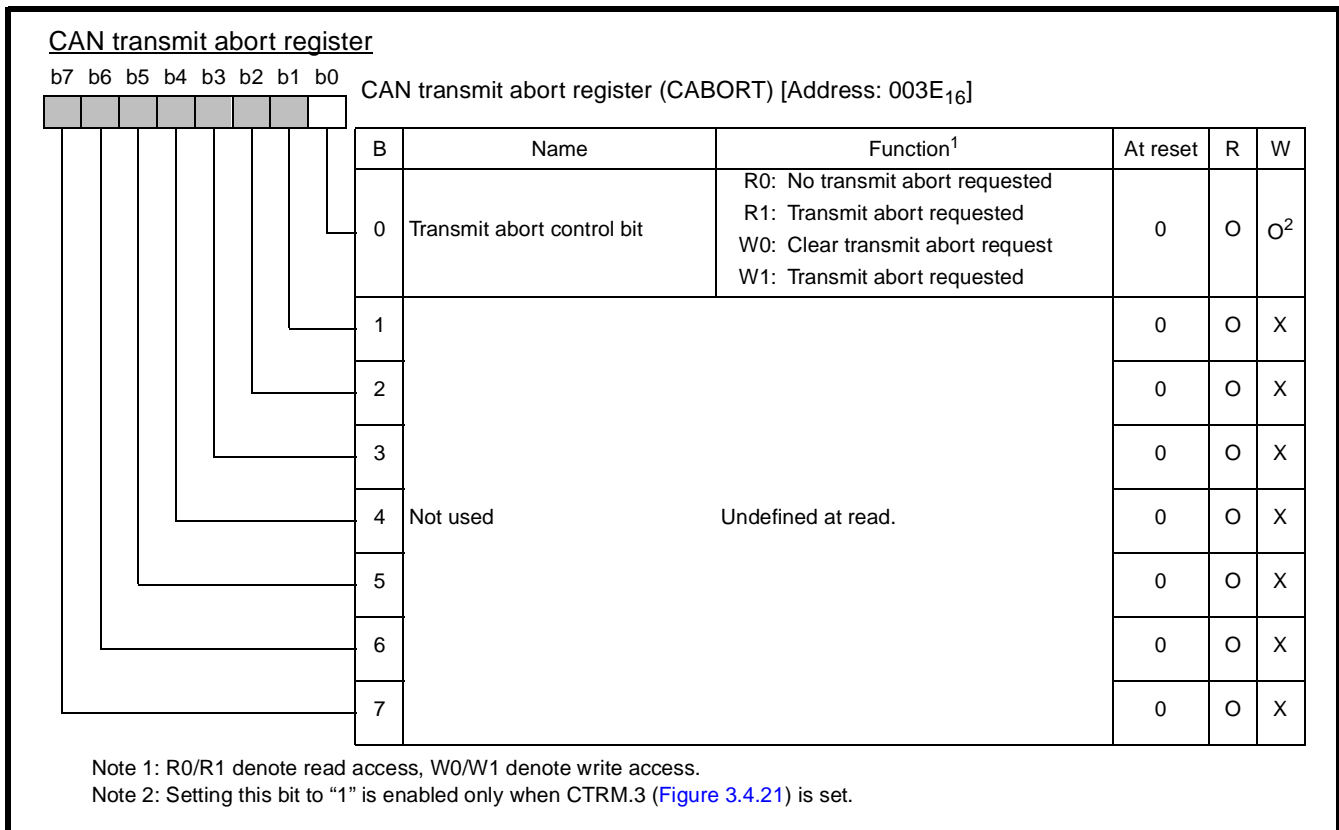


Fig. 3.5.47 Structure of CAN transmit abort register (CABORT)

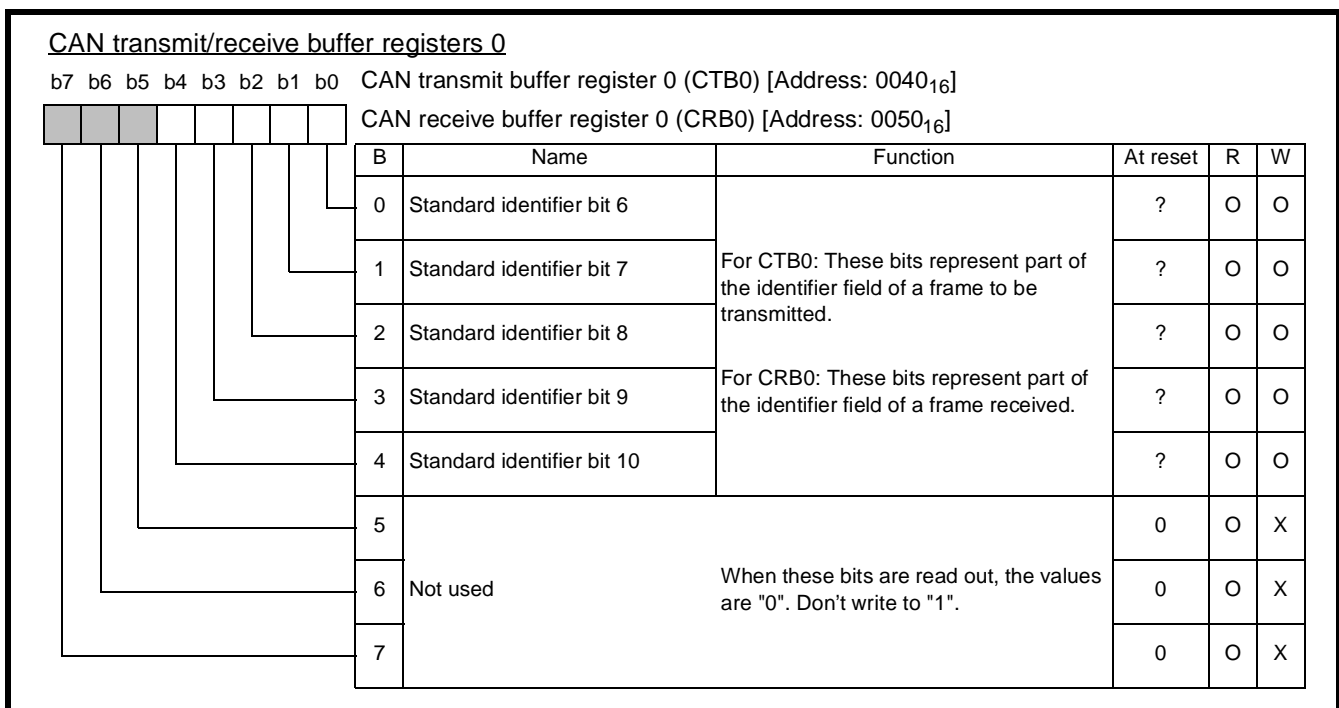


Fig. 3.5.48 Structure of CAN transmit/receive buffer registers 0 (CTB0/CRB0)

APPENDIX

3.5 List of registers

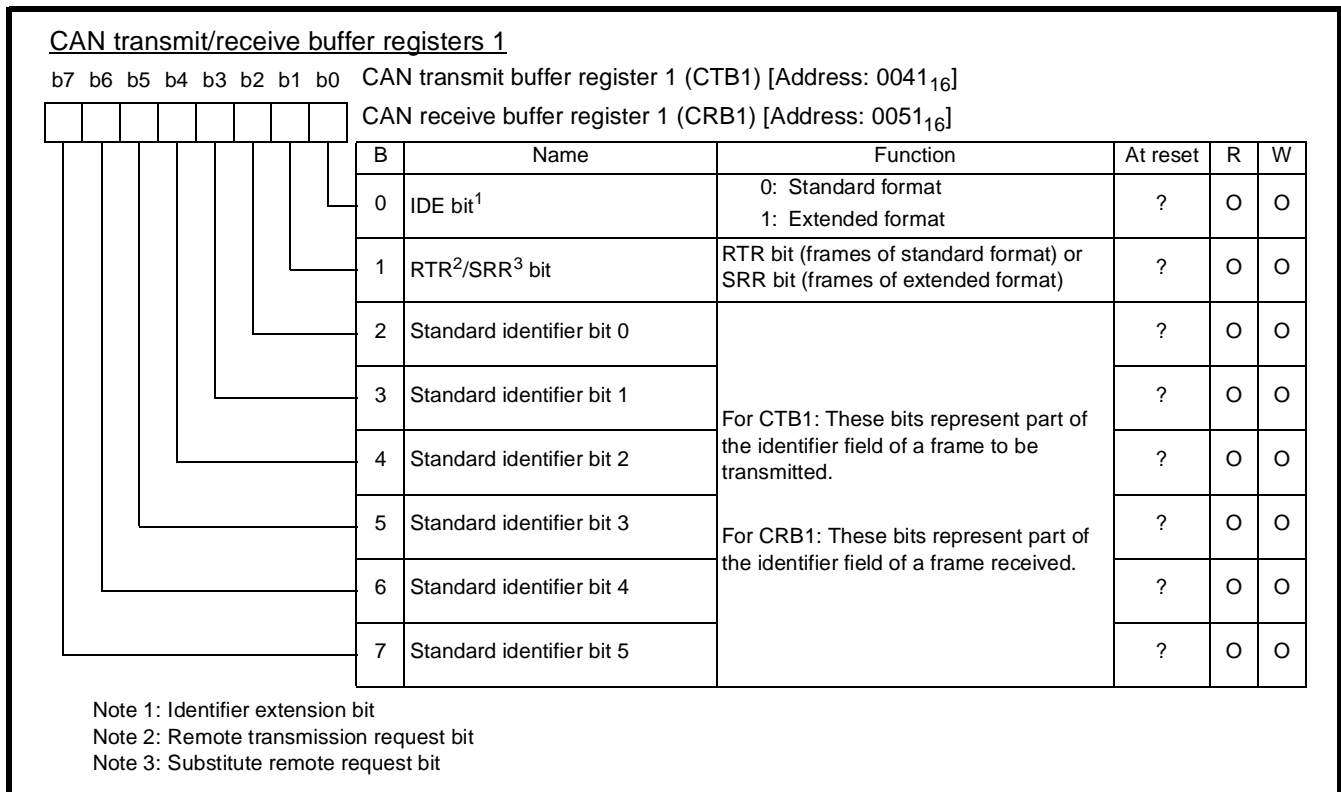


Fig. 3.5.49 Structure of CAN transmit/receive buffer registers 1 (CTB1/CRB1)

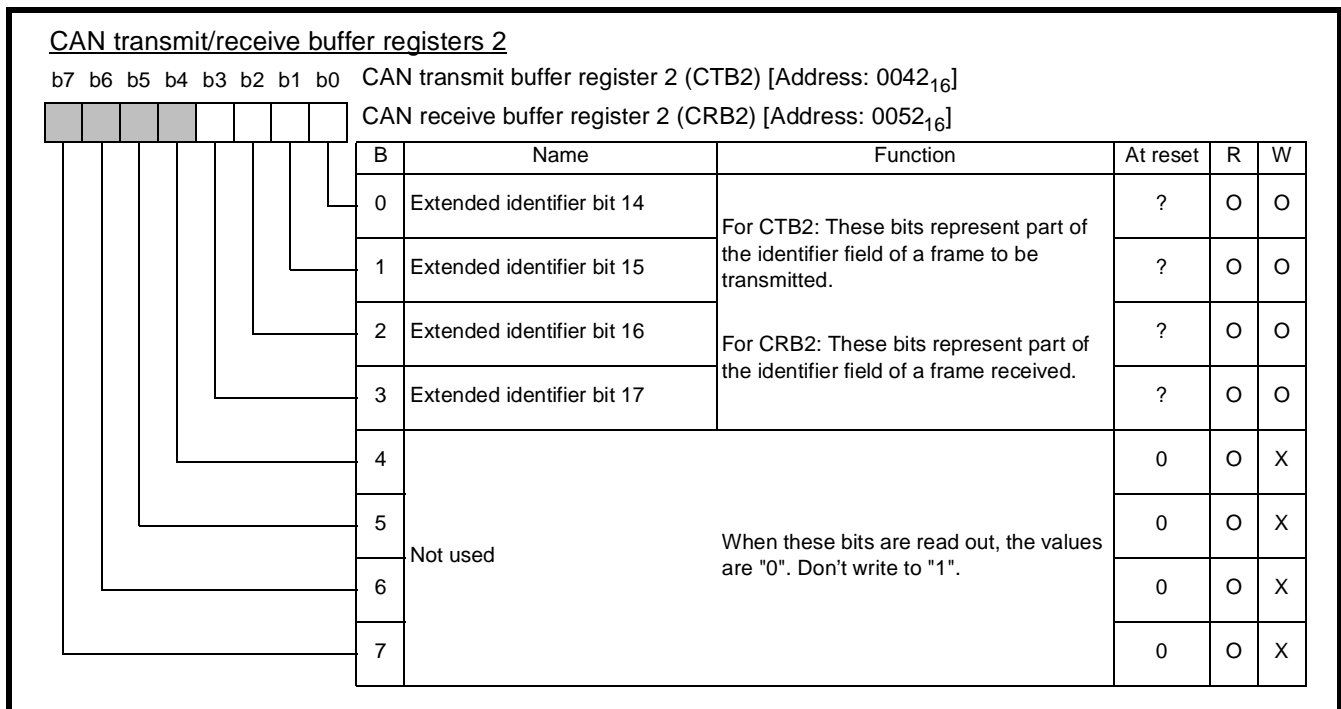


Fig. 3.5.50 Structure of CAN transmit/receive buffer registers 2 (CTB2/CRB2)

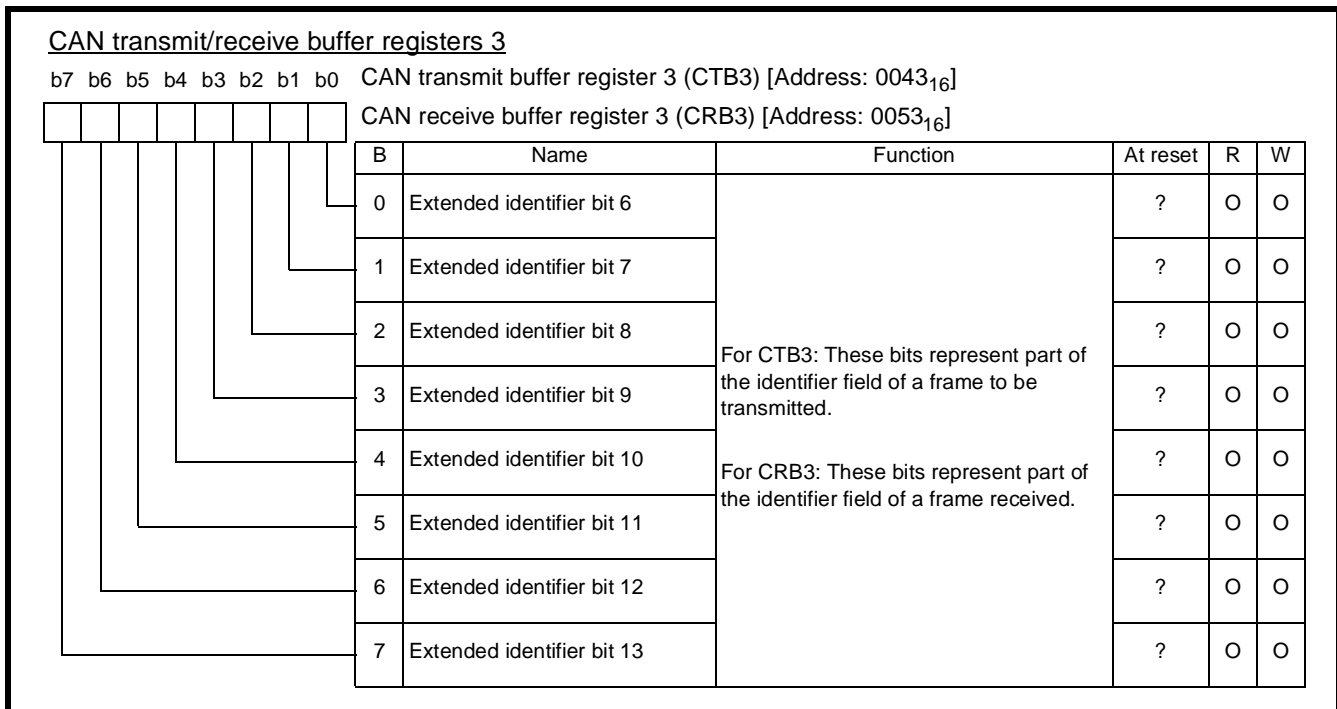


Fig. 3.5.51 Structure of CAN transmit/receive buffer registers 3 (CTB3/CRB3)

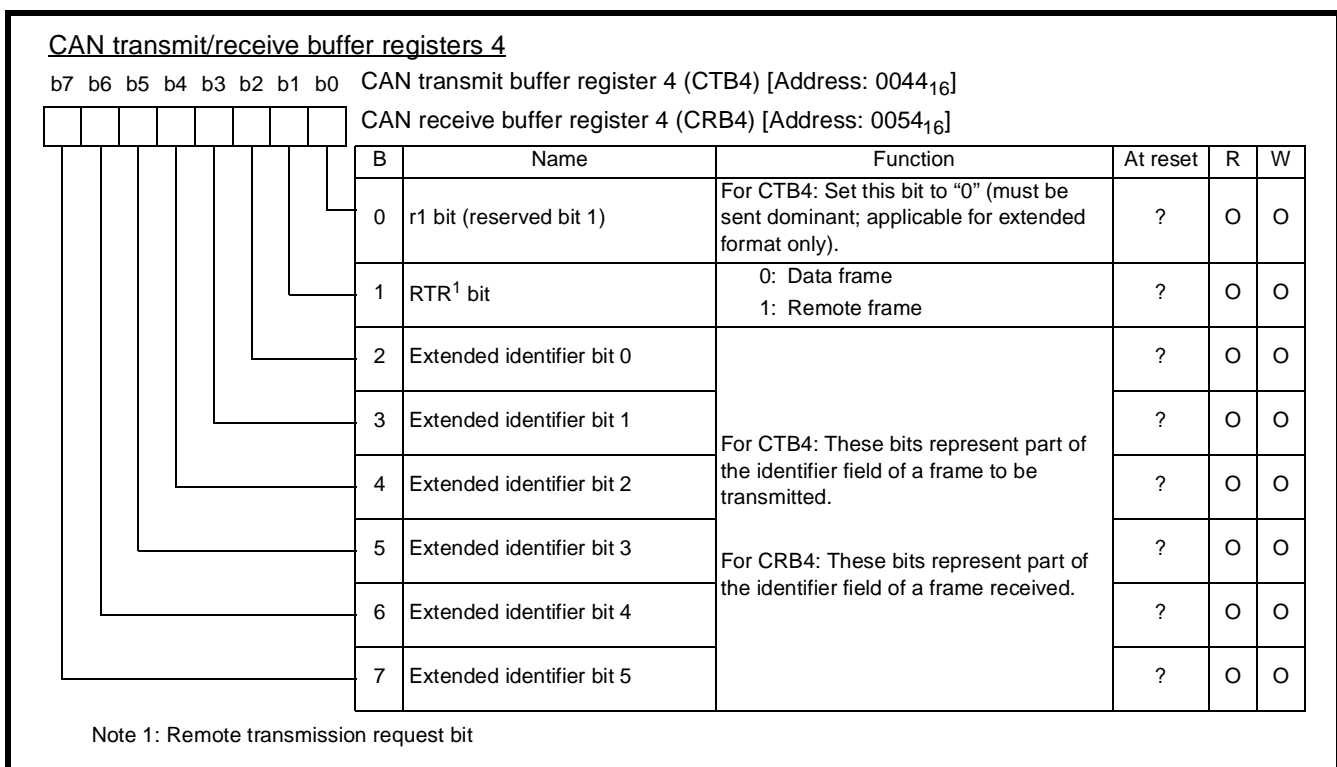


Fig. 3.5.52 Structure of CAN transmit/receive buffer registers 4 (CTB4/CRB4)

APPENDIX

3.5 List of registers

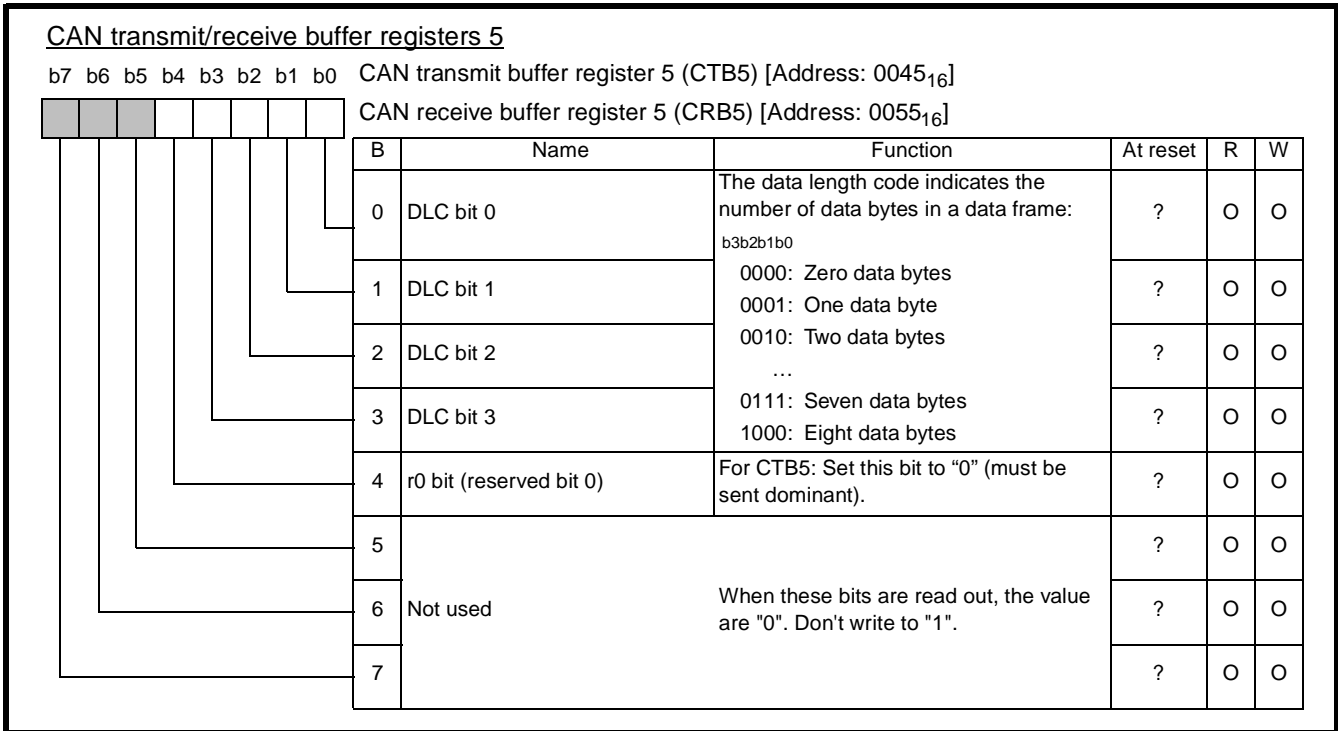


Fig. 3.5.53 Structure of CAN transmit/receive buffer registers 5 (CTB5/CRB5)

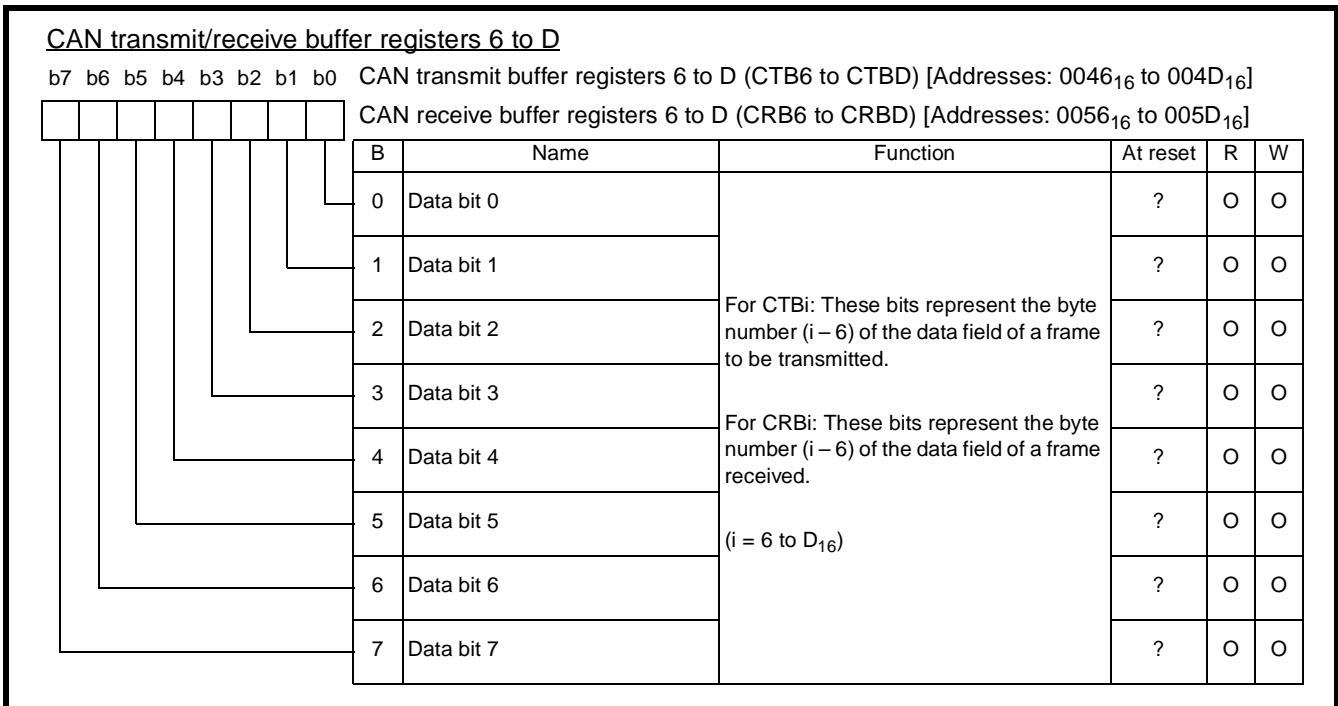


Fig. 3.5.54 Structure of CAN transmit/receive buffer registers 6 to D (CTB6–D/CRB6–D)

3.6 Mask ROM ordering method

GZZ-SH52-70B<84A0>

740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M37630M4T-XXXFP MITSUBISHI ELECTRIC

Mask ROM number	
-----------------	--

Receipt	Date:	
	Section head signature	Supervisor signature

Note : Please fill in all items marked *.

* Customer	Company name	TEL ()	Issuance signature	Submitted by	Supervisor
	Date issued	Date:			

* 1. Confirmation

Specify the type of EPROMs submitted.

Three EPROMs are required for each pattern.

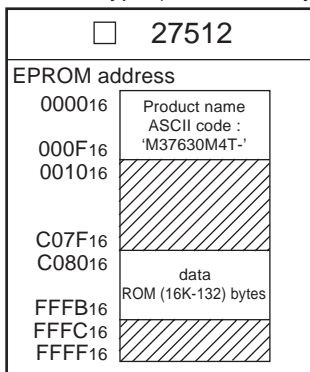
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Checksum code for entire EPROM

--	--	--	--

 (hexadecimal notation)

EPROM type (indicate the type used)



In the address space of the microcomputer, the internal ROM area is from address C080₁₆ to FFFB₁₆. The reset vector is stored in addresses FFFA₁₆ and FFFB₁₆.

- (1) Set the data in the unused area (the shaded area of the diagram) to "FF₁₆".
- (2) The ASCII codes of the product name "M37630M4T-" must be entered in addresses 0000₁₆ to 000F₁₆. And set the data "FF₁₆" in addresses 000A₁₆ to 000F₁₆. The ASCII codes and addresses are listed to the right in hexadecimal notation.

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">Address</td><td></td></tr> <tr><td>0000₁₆</td><td>'M' = 4D₁₆</td></tr> <tr><td>0001₁₆</td><td>'3' = 33₁₆</td></tr> <tr><td>0002₁₆</td><td>'7' = 37₁₆</td></tr> <tr><td>0003₁₆</td><td>'6' = 36₁₆</td></tr> <tr><td>0004₁₆</td><td>'3' = 33₁₆</td></tr> <tr><td>0005₁₆</td><td>'0' = 30₁₆</td></tr> <tr><td>0006₁₆</td><td>'M' = 4D₁₆</td></tr> <tr><td>0007₁₆</td><td>'4' = 34₁₆</td></tr> </table>	Address		0000 ₁₆	'M' = 4D ₁₆	0001 ₁₆	'3' = 33 ₁₆	0002 ₁₆	'7' = 37 ₁₆	0003 ₁₆	'6' = 36 ₁₆	0004 ₁₆	'3' = 33 ₁₆	0005 ₁₆	'0' = 30 ₁₆	0006 ₁₆	'M' = 4D ₁₆	0007 ₁₆	'4' = 34 ₁₆	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%;">Address</td><td></td></tr> <tr><td>0008₁₆</td><td>'T' = 54₁₆</td></tr> <tr><td>0009₁₆</td><td>'_' = 2D₁₆</td></tr> <tr><td>000A₁₆</td><td>FF₁₆</td></tr> <tr><td>000B₁₆</td><td>FF₁₆</td></tr> <tr><td>000C₁₆</td><td>FF₁₆</td></tr> <tr><td>000D₁₆</td><td>FF₁₆</td></tr> <tr><td>000E₁₆</td><td>FF₁₆</td></tr> <tr><td>000F₁₆</td><td>FF₁₆</td></tr> </table>	Address		0008 ₁₆	'T' = 54 ₁₆	0009 ₁₆	'_' = 2D ₁₆	000A ₁₆	FF ₁₆	000B ₁₆	FF ₁₆	000C ₁₆	FF ₁₆	000D ₁₆	FF ₁₆	000E ₁₆	FF ₁₆	000F ₁₆	FF ₁₆
Address																																					
0000 ₁₆	'M' = 4D ₁₆																																				
0001 ₁₆	'3' = 33 ₁₆																																				
0002 ₁₆	'7' = 37 ₁₆																																				
0003 ₁₆	'6' = 36 ₁₆																																				
0004 ₁₆	'3' = 33 ₁₆																																				
0005 ₁₆	'0' = 30 ₁₆																																				
0006 ₁₆	'M' = 4D ₁₆																																				
0007 ₁₆	'4' = 34 ₁₆																																				
Address																																					
0008 ₁₆	'T' = 54 ₁₆																																				
0009 ₁₆	'_' = 2D ₁₆																																				
000A ₁₆	FF ₁₆																																				
000B ₁₆	FF ₁₆																																				
000C ₁₆	FF ₁₆																																				
000D ₁₆	FF ₁₆																																				
000E ₁₆	FF ₁₆																																				
000F ₁₆	FF ₁₆																																				

APPENDIX

3.6 Mask ROM ordering method

GZZ-SH52-70B<84A0>

Mask ROM number	
-----------------	--

740 FAMILY MASK ROM CONFIRMATION FORM SINGLE-CHIP MICROCOMPUTER M37630M4T-XXXFP MITSUBISHI ELECTRIC

We recommend the use of the following pseudo-command to set the start address of the assembler source program.

EPROM type	27512
The pseudo-command	*= Δ \$0000 .BYTE Δ 'M37630M4T-'

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

※ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (44P6N) and attach it to the mask ROM confirmation form.

※ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the XIN-XOUT oscillator?

- | | |
|---|---|
| <input type="checkbox"/> Ceramic resonator | <input type="checkbox"/> Quartz crystal |
| <input type="checkbox"/> External clock input | <input type="checkbox"/> Other () |

At what frequency?

f(XIN) = MHz

※ 4. Comments

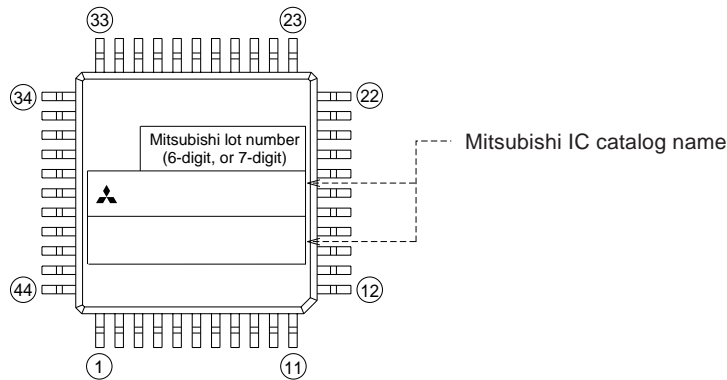
3.7 Mark specification form

44P6N (44-PIN QFP) MARK SPECIFICATION FORM

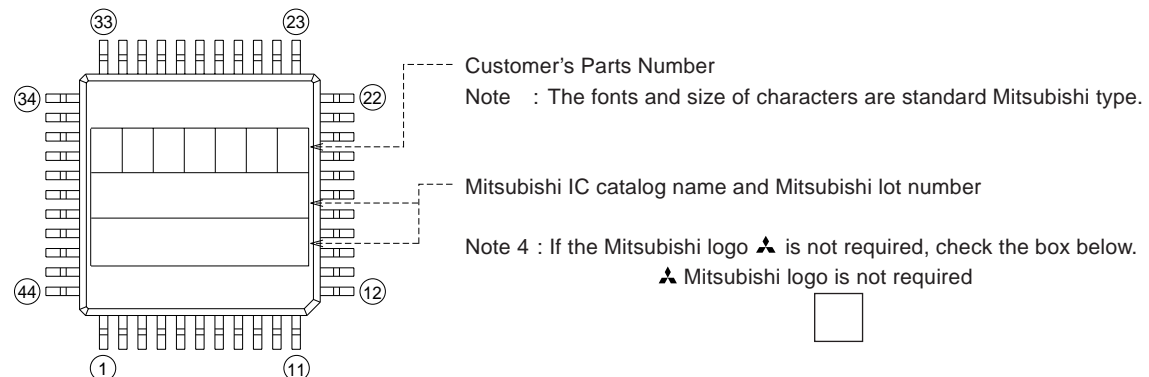
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark

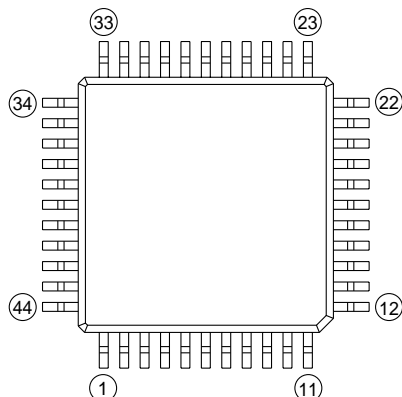


B. Customer's Parts Number + Mitsubishi IC Catalog Name



- Notes 1 : The mark field should be written right aligned.
 2 : The fonts and size of characters are standard Mitsubishi type.
 3 : Customer's parts number can be up to 7 characters:
 Only 0 to 9, A to Z, +, -, /, (,), &, ©, . (period), and , (comma) are usable.

C. Special Mark Required



- Notes1 : If the special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated as close as possible.
 Mitsubishi lot number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked.
 2 : If the customer's trade mark logo must be used in the special mark, check the box below.
 Please submit a clean original of the logo.
 For the new special character fonts, a clean font original (ideally logo drawing) must be submitted.

Special logo required

The standard Mitsubishi font is used for all characters except for a logo.

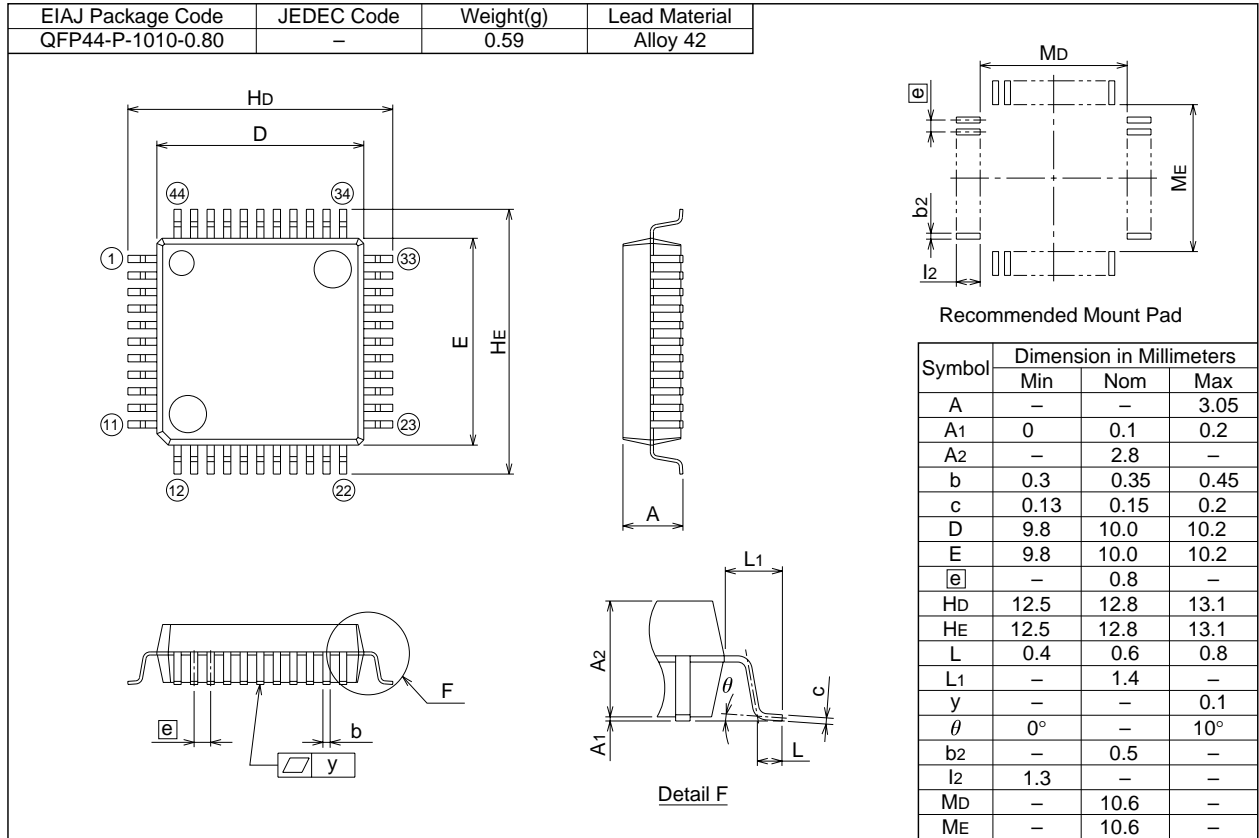
APPENDIX

3.8 Package outline

3.8 Package outline

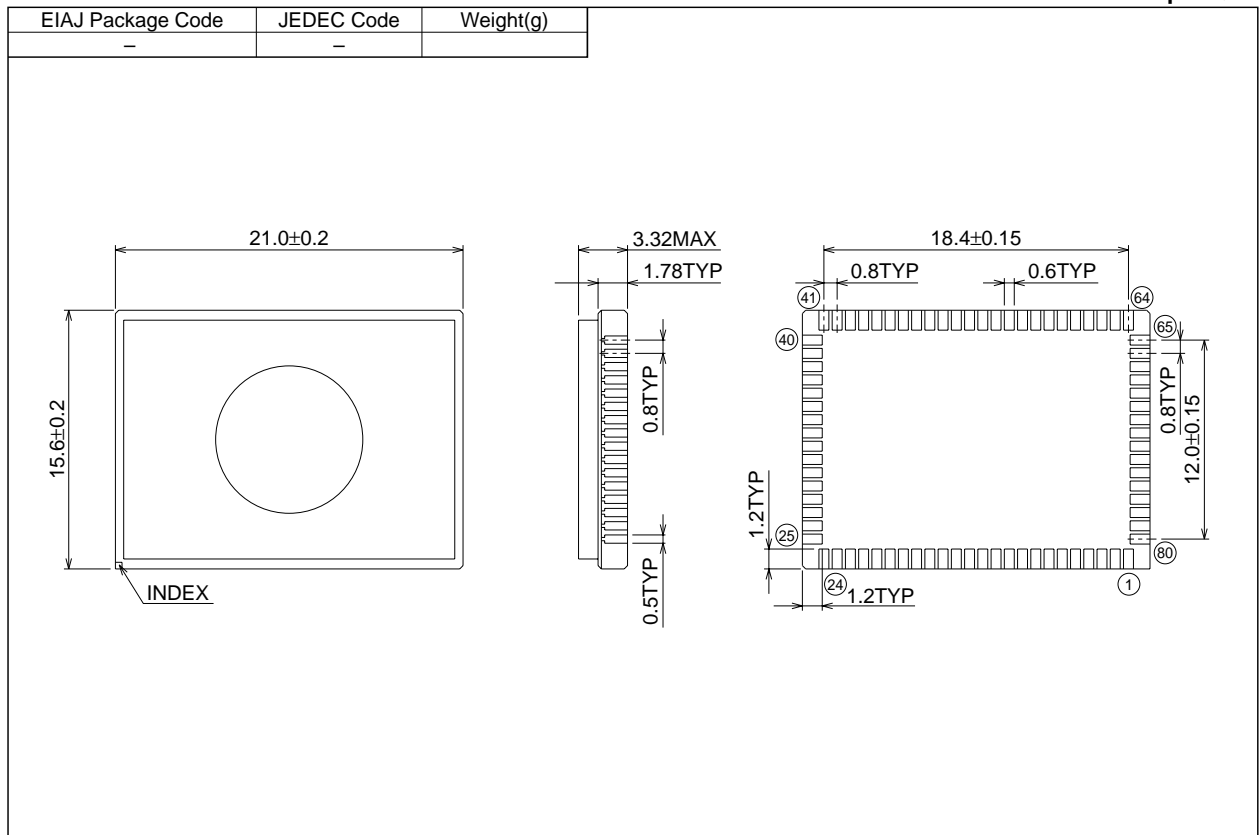
44P6N-A

Plastic 44pin 10X10mm body QFP



80D0

Glass seal 80pin QFN



3.9 List of instruction codes

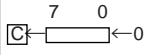
D7 – D4	D3 – D0	Hexadecimal notation															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	BRK	ORA IND, X	JSR ZP, IND	BBS 0, A	—	ORA ZP	ASL ZP	BBS 0, ZP	PHP	ORA IMM	ASL A	SEB 0, A	—	ORA ABS	ASL ABS	SEB 0, ZP
0001	1	BPL	ORA IND, Y	CLT	BBC 0, A	—	ORA ZP, X	ASL ZP, X	BBC 0, ZP	CLC	ORA ABS, Y	DEC A	CLB 0, A	—	ORA ABS, X	ASL ABS, X	CLB 0, ZP
0010	2	JSR ABS	AND IND, X	JSR SP	BBS 1, A	BIT ZP	AND ZP	ROL ZP	BBS 1, ZP	PLP	AND IMM	ROL A	SEB 1, A	BIT ABS	AND ABS	ROL ABS	SEB 1, ZP
0011	3	BMI	AND IND, Y	SET	BBC 1, A	—	AND ZP, X	ROL ZP, X	BBC 1, ZP	SEC	AND ABS, Y	INC A	CLB 1, A	LDM ZP	AND ABS, X	ROL ABS, X	CLB 1, ZP
0100	4	RTI	EOR IND, X	STP	BBS 2, A	COM ZP	EOR ZP	LSR ZP	BBS 2, ZP	PHA	EOR IMM	LSR A	SEB 2, A	JMP ABS	EOR ABS	LSR ABS	SEB 2, ZP
0101	5	BVC	EOR IND, Y	—	BBC 2, A	—	EOR ZP, X	LSR ZP, X	BBC 2, ZP	CLI	EOR ABS, Y	—	CLB 2, A	—	EOR ABS, X	LSR ABS, X	CLB 2, ZP
0110	6	RTS	ADC IND, X	MUL ZP, X	BBS 3, A	TST ZP	ADC ZP	ROR ZP	BBS 3, ZP	PLA	ADC IMM	ROR A	SEB 3, A	JMP IND	ADC ABS	ROR ABS	SEB 3, ZP
0111	7	BVS	ADC IND, Y	—	BBC 3, A	—	ADC ZP, X	ROR ZP, X	BBC 3, ZP	SEI	ADC ABS, Y	—	CLB 3, A	—	ADC ABS, X	ROR ABS, X	CLB 3, ZP
1000	8	BRA	STA IND, X	RRF ZP	BBS 4, A	STY ZP	STA ZP	STX ZP	BBS 4, ZP	DEY	—	TXA	SEB 4, A	STY ABS	STA ABS	STX ABS	SEB 4, ZP
1001	9	BCC	STA IND, Y	—	BBC 4, A	STY ZP, X	STA ZP, X	STX ZP, Y	BBC 4, ZP	TYA	STA ABS, Y	TXS	CLB 4, A	—	STA ABS, X	—	CLB 4, ZP
1010	A	LDY IMM	LDA IND, X	LDX IMM	BBS 5, A	LDY ZP	LDA ZP	LDX ZP	BBS 5, ZP	TAY	LDA IMM	TAX	SEB 5, A	LDY ABS	LDA ABS	LDX ABS	SEB 5, ZP
1011	B	BCS	LDA IND, Y	JMP ZP, IND	BBC 5, A	LDY ZP, X	LDA ZP, X	LDX ZP, Y	BBC 5, ZP	CLV	LDA ABS, Y	TSX	CLB 5, A	LDY ABS, X	LDA ABS, X	LDX ABS, Y	CLB 5, ZP
1100	C	CPY IMM	CMP IND, X	WIT	BBS 6, A	CPY ZP	CMP ZP	DEC ZP	BBS 6, ZP	INY	CMP IMM	DEX	SEB 6, A	CPY ABS	CMP ABS	DEC ABS	SEB 6, ZP
1101	D	BNE	CMP IND, Y	—	BBC 6, A	—	CMP ZP, X	DEC ZP, X	BBC 6, ZP	CLD	CMP ABS, Y	—	CLB 6, A	—	CMP ABS, X	DEC ABS, X	CLB 6, ZP
1110	E	CPX IMM	SBC IND, X	DIV ZP, X	BBS 7, A	CPX ZP	SBC ZP	INC ZP	BBS 7, ZP	INX	SBC IMM	NOP	SEB 7, A	CPX ABS	SBC ABS	INC ABS	SEB 7, ZP
1111	F	BEQ	SBC IND, Y	—	BBC 7, A	—	SBC ZP, X	INC ZP, X	BBC 7, ZP	SED	SBC ABS, Y	—	CLB 7, A	—	SBC ABS, X	INC ABS, X	CLB 7, ZP

- 3-byte instruction
- 2-byte instruction
- 1-byte instruction

APPENDIX

3.10 Machine instructions

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																				
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R					
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#			
ADC (Note 1) (Note 8)	When T = 0 $A \leftarrow A + M + C$ When T = 1 $M(X) \leftarrow M(X) + M + C$	Adds the carry, accumulator and memory contents. The results are entered into the accumulator. Adds the contents of the memory in the address indicated by index register X, the contents of the memory specified by the addressing mode and the carry. The results are entered into the memory at the address indicated by index register X.				69	2	2										65	3	2			
AND (Note 1)	When T = 0 $A \leftarrow A \wedge M$ When T = 1 $M(X) \leftarrow M(X) \wedge M$	"AND's" the accumulator and memory contents. The results are entered into the accumulator. "AND's" the contents of the memory of the address indicated by index register X and the contents of the memory specified by the addressing mode. The results are entered into the memory at the address indicated by index register X.				29	2	2										25	3	2			
ASL	 $\leftarrow 0$	Shifts the contents of accumulator or contents of memory one bit to the left. The low order bit of the accumulator or memory is cleared and the high order bit is shifted into the carry flag.							0A	1	1							06	5	2			
BBC	A_i or $M_i = 0?$	Branches when the contents of the bit specified in the accumulator or memory is "0".										$\overline{13}$ $\overline{20}_i$ (Note 4)	4	2				$\overline{17}$ $\overline{20}_i$ (Note 6)	5	3			
BBS	A_i or $M_i = 1?$	Branches when the contents of the bit specified in the accumulator or memory is "1".										$\overline{03}$ $\overline{20}_i$ (Note 4)	4	2				$\overline{07}$ $\overline{20}_i$ (Note 6)	5	3			
BCC (Note 5)	$C = 0?$	Branches when the contents of carry flag is "0".																					
BCS (Note 5)	$C = 1?$	Branches when the contents of carry flag is "1".																					
BEQ (Note 5)	$Z = 1?$	Branches when the contents of zero flag is "1".																					
BIT	$A \wedge M$	"AND's" the contents of accumulator and memory. The results are not entered anywhere.																24	3	2			
BMI (Note 5)	$N = 1?$	Branches when the contents of negative flag is "1".																					
BNE (Note 5)	$Z = 0?$	Branches when the contents of zero flag is "0".																					
BPL (Note 5)	$N = 0?$	Branches when the contents of negative flag is "0".																					
BRA (Note 6)	$PC \leftarrow PC \pm \text{offset}$	Jumps to address specified by adding offset to the program counter.																					
BRK	$B \leftarrow 1$ $PC \leftarrow PC + 2$ $M(S) \leftarrow PCH$ $S \leftarrow S - 1$ $M(S) \leftarrow PCL$ $S \leftarrow S - 1$ $M(S) \leftarrow PS$ $S \leftarrow S - 1$ $I \leftarrow 1$ $PCL \leftarrow ADL$ $PCH \leftarrow ADH$	Executes a software interrupt.	00	7	1																		

APPENDIX

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
BVC (Note 5)	$V = 0?$	Branches when the contents of overflow flag is "0".																		
BVS (Note 5)	$V = 1?$	Branches when the contents of overflow flag is "1".																		
CLB	$A_i \text{ or } M_i \leftarrow 0$	Clears the contents of the bit specified in the accumulator or memory to "0".										$1\overline{B}$ 20_i	1	1				$1\overline{F}$ 20_i	5	2
CLC	$C \leftarrow 0$	Clears the contents of the carry flag to "0".	18	1	1															
CLD	$D \leftarrow 0$	Clears the contents of decimal mode flag to "0".	D8	1	1															
CLI	$I \leftarrow 0$	Clears the contents of interrupt disable flag to "0".	58	2	1															
CLT	$T \leftarrow 0$	Clears the contents of index X mode flag to "0".	12	1	1															
CLV	$V \leftarrow 0$	Clears the contents of overflow flag to "0".	B8	1	1															
CMP (Note 3)	When $T = 0$ $A - M$ When $T = 1$ $M(X) - M$	Compares the contents of accumulator and memory. Compares the contents of the memory specified by the addressing mode with the contents of the address indicated by index register X.				C9	2	2									C5	3	2	
COM	$M \leftarrow \overline{M}$	Forms a one's complement of the contents of memory, and stores it into memory.															44	5	2	
CPX	$X - M$	Compares the contents of index register X and memory.				E0	2	2									E4	3	2	
CPY	$Y - M$	Compares the contents of index register Y and memory.				C0	2	2									C4	3	2	
DEC	$A \leftarrow A - 1$ or $M \leftarrow M - 1$	Decrements the contents of the accumulator or memory by 1.							1A	1	1						C6	5	2	
DEX	$X \leftarrow X - 1$	Decrements the contents of index register X by 1.	CA	1	1															
DEY	$Y \leftarrow Y - 1$	Decrements the contents of index register Y by 1.	88	1	1															
DIV	$A \leftarrow (M(zz + X + 1),$ $M(zz + X)) / A$ $M(S) \leftarrow 1$'s complement of Remainder $S \leftarrow S - 1$	Divides the 16-bit data that is the contents of M (zz + x + 1) for high byte and the contents of M (zz + x) for low byte by the accumulator. Stores the quotient in the accumulator and the 1's complement of the remainder on the stack.																		
EOR (Note 1)	When $T = 0$ $A \leftarrow A \vee M$ When $T = 1$ $M(X) \leftarrow M(X) \vee M$	"Exclusive-ORs" the contents of accumulator and memory. The results are stored in the accumulator. "Exclusive-ORs" the contents of the memory specified by the addressing mode and the contents of the memory at the address indicated by index register X. The results are stored into the memory at the address indicated by index register X.				49	2	2									45	3	2	
INC	$A \leftarrow A + 1$ or $M \leftarrow M + 1$	Increments the contents of accumulator or memory by 1.							3A	1	1						E6	5	2	
INX	$X \leftarrow X + 1$	Increments the contents of index register X by 1.	E8	1	1															
INY	$Y \leftarrow Y + 1$	Increments the contents of index register Y by 1.	C8	1	1															

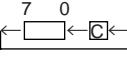
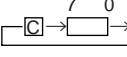

APPENDIX

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																	
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R		
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#
JMP	If addressing mode is ABS PCL ← ADL PCH ← ADH If addressing mode is IND PCL ← M (ADH, ADL) PCH ← M (ADH, ADL + 1) If addressing mode is ZP, IND PCL ← M(00, ADL) PCH ← M(00, ADL + 1)	Jumps to the specified address.																		
JSR	M(S) ← PCH S ← S - 1 M(S) ← PCL S ← S - 1 After executing the above, if addressing mode is ABS, PCL ← ADL PCH ← ADH if addressing mode is SP, PCL ← ADL PCH ← FF If addressing mode is ZP, IND, PCL ← M(00, ADL) PCH ← M(00, ADL + 1)	After storing contents of program counter in stack, and jumps to the specified address.																		
LDA (Note 2)	When T = 0 A ← M When T = 1 M(X) ← M	Load accumulator with contents of memory. Load memory indicated by index register X with contents of memory specified by the addressing mode.				A9	2	2							A5	3	2			
LDM	M ← nn	Load memory with immediate value.													3C	4	3			
LDX	X ← M	Load index register X with contents of memory.				A2	2	2							A6	3	2			
LDY	Y ← M	Load index register Y with contents of memory.				A0	2	2							A4	3	2			
LSR	$\begin{array}{c} 7 \quad 0 \\ 0 \rightarrow \boxed{} \rightarrow \boxed{C} \end{array}$	Shift the contents of accumulator or memory to the right by one bit. The low order bit of accumulator or memory is stored in carry, 7th bit is cleared.							4A	1	1				46	5	2			
MUL	M(S) · A ← A X M(zz + X) S ← S - 1	Multiplies the accumulator with the contents of memory specified by the zero page X addressing mode and stores the high byte of the result on the stack and the low byte in the accumulator.																		
NOP	PC ← PC + 1	No operation.	EA	1	1															
ORA (Note 1)	When T = 0 A ← A V M When T = 1 M(X) ← M(X) V M	“Logical OR’s” the contents of memory and accumulator. The result is stored in the accumulator. “Logical OR’s” the contents of memory indicated by index register X and contents of memory specified by the addressing mode. The result is stored in the memory specified by index register X.				09	2	2							05	3	2			

APPENDIX

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																				
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R					
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#			
PHA	$M(S) \leftarrow A$ $S \leftarrow S - 1$	Saves the contents of the accumulator in memory at the address indicated by the stack pointer and decrements the contents of stack pointer by 1.	48	3	1																		
PHP	$M(S) \leftarrow PS$ $S \leftarrow S - 1$	Saves the contents of the processor status register in memory at the address indicated by the stack pointer and decrements the contents of the stack pointer by 1.	08	3	1																		
PLA	$S \leftarrow S + 1$ $A \leftarrow M(S)$	Increments the contents of the stack pointer by 1 and restores the accumulator from the memory at the address indicated by the stack pointer.	68	4	1																		
PLP	$S \leftarrow S + 1$ $PS \leftarrow M(S)$	Increments the contents of stack pointer by 1 and restores the processor status register from the memory at the address indicated by the stack pointer.	28	4	1																		
ROL		Shifts the contents of the memory or accumulator to the left by one bit. The high order bit is shifted into the carry flag and the carry flag is shifted into the low order bit.							2A	1	1					26	5	2					
ROR		Shifts the contents of the memory or accumulator to the right by one bit. The low order bit is shifted into the carry flag and the carry flag is shifted into the high order bit.							6A	1	1					66	5	2					
RRF		Rotates the contents of memory to the right by 4 bits.														82	8	2					
RTI	$S \leftarrow S + 1$ $PS \leftarrow M(S)$ $S \leftarrow S + 1$ $PCL \leftarrow M(S)$ $S \leftarrow S + 1$ $PCH \leftarrow M(S)$	Returns from an interrupt routine to the main routine.	40	6	1																		
RTS	$S \leftarrow S + 1$ $PCL \leftarrow M(S)$ $S \leftarrow S + 1$ $PCH \leftarrow M(S)$ $PC \leftarrow PC + 1$	Returns from a subroutine to the main routine.	60	6	1																		
SBC (Note 1) (Note 8)	When $T = 0$ $A \leftarrow A - M - \bar{C}$ When $T = 1$ $M(X) \leftarrow M(X) - M - \bar{C}$	Subtracts the contents of memory and complement of carry flag from the contents of accumulator. The results are stored into the accumulator. Subtracts contents of complement of carry flag and contents of the memory indicated by the addressing mode from the memory at the address indicated by index register X. The results are stored into the memory of the address indicated by index register X.							E9	2	2					E5	3	2					
SEB	A_i or $M_i \leftarrow 1$	Sets the specified bit in the accumulator or memory to "1".														0_{P20i}	1	1			0_{F20i}	5	2
SEC	$C \leftarrow 1$	Sets the contents of the carry flag to "1".	38	1	1																		
SED	$D \leftarrow 1$	Sets the contents of the decimal mode flag to "1".	F8	1	1																		
SEI	$I \leftarrow 1$	Sets the contents of the interrupt disable flag to "1".	78	2	1																		
SET	$T \leftarrow 1$	Sets the contents of the index X mode flag to "1".	32	1	1																		

APPENDIX

3.10 Machine instructions

Symbol	Function	Details	Addressing mode																					
			IMP			IMM			A			BIT, A, R			ZP			BIT, ZP, R						
			OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#				
STA	$M \leftarrow A$	Stores the contents of accumulator in memory.																85	3	2				
STP (Note 7)		Stops the oscillator.	42	2	1																			
STX	$M \leftarrow X$	Stores the contents of index register X in memory.																	86	3	2			
STY	$M \leftarrow Y$	Stores the contents of index register Y in memory.																	84	3	2			
TAX	$X \leftarrow A$	Transfers the contents of the accumulator to index register X.	AA	1	1																			
TAY	$Y \leftarrow A$	Transfers the contents of the accumulator to index register Y.	A8	1	1																			
TST	$M = 0?$	Tests whether the contents of memory are "0" or not.																	64	3	2			
TSX	$X \leftarrow S$	Transfers the contents of the stack pointer to index register X.	BA	1	1																			
TXA	$A \leftarrow X$	Transfers the contents of index register X to the accumulator.	8A	1	1																			
TXS	$S \leftarrow X$	Transfers the contents of index register X to the stack pointer.	9A	1	1																			
TYA	$A \leftarrow Y$	Transfers the contents of index register Y to the accumulator.	98	1	1																			
WIT		Stops the internal clock.	C2	2	1																			

- Notes**
- 1 : The number of cycles "n" is increased by 3 when T is 1.
 - 2 : The number of cycles "n" is increased by 2 when T is 1.
 - 3 : The number of cycles "n" is increased by 1 when T is 1.
 - 4 : The number of cycles "n" is increased by 1 when branching has occurred.
 - 5 : The number of cycles "n" is increased by 1 when branching to the same page area has occurred.
The number of cycles "n" is increased by 2 when branching to the different page area has occurred.
 - 6 : The number of cycles "n" is increased by 1 when branching to the different page area has occurred.
 - 7 : The number of cycles "n" is 2 when the **STP** instruction is disabled.
 - 8 : V flag is invalid in decimal operation mode.

APPENDIX

3.11 SFR memory map

3.11 SFR memory map

0000 ¹⁶	CPU mode register	CPUM	0030 ¹⁶	CAN transmit control register	CTRM
0001 ¹⁶	Not used		0031 ¹⁶	CAN bus timing control register 1	CBTCON1
0002 ¹⁶	Interrupt request register A	IREQA	0032 ¹⁶	CAN bus timing control register 2	CBTCON2
0003 ¹⁶	Interrupt request register B	IREQB	0033 ¹⁶	CAN acceptance code register 0	CAC0
0004 ¹⁶	Interrupt request register C	IREQC	0034 ¹⁶	CAN acceptance code register 1	CAC1
0005 ¹⁶	Interrupt control register A	ICONA	0035 ¹⁶	CAN acceptance code register 2	CAC2
0006 ¹⁶	Interrupt control register B	ICONB	0036 ¹⁶	CAN acceptance code register 3	CAC3
0007 ¹⁶	Interrupt control register C	ICONC	0037 ¹⁶	CAN acceptance code register 4	CAC4
0008 ¹⁶	Port P0 register	P0	0038 ¹⁶	CAN acceptance mask register 0	CAM0
0009 ¹⁶	Port P0 direction register	P0D	0039 ¹⁶	CAN acceptance mask register 1	CAM1
000A ¹⁶	Port P1 register	P1	003A ¹⁶	CAN acceptance mask register 2	CAM2
000B ¹⁶	Port P1 direction register	P1D	003B ¹⁶	CAN acceptance mask register 3	CAM3
000C ¹⁶	Port P2 register	P2	003C ¹⁶	CAN acceptance mask register 4	CAM4
000D ¹⁶	Port P2 direction register	P2D	003D ¹⁶	CAN receive control register	CREC
000E ¹⁶	Port P3 register	P3	003E ¹⁶	CAN transmit abort register	CABORT
000F ¹⁶	Port P3 direction register	P3D	003F ¹⁶	Reserved	
0010 ¹⁶	Port P4 register	P4	0040 ¹⁶	CAN transmit buffer register 0	CTB0
0011 ¹⁶	Port P4 direction register	P4D	0041 ¹⁶	CAN transmit buffer register 1	CTB1
0012 ¹⁶	Serial I/O shift register	SIO	0042 ¹⁶	CAN transmit buffer register 2	CTB2
0013 ¹⁶	Serial I/O control register	SIOCON	0043 ¹⁶	CAN transmit buffer register 3	CTB3
0014 ¹⁶	A-D conversion register	AD	0044 ¹⁶	CAN transmit buffer register 4	CTB4
0015 ¹⁶	A-D control register	ADCON	0045 ¹⁶	CAN transmit buffer register 5	CTB5
0016 ¹⁶	Timer 1	T1	0046 ¹⁶	CAN transmit buffer register 6	CTB6
0017 ¹⁶	Timer 2	T2	0047 ¹⁶	CAN transmit buffer register 7	CTB7
0018 ¹⁶	Timer 3	T3	0048 ¹⁶	CAN transmit buffer register 8	CTB8
0019 ¹⁶	Timer 123 mode register	T123M	0049 ¹⁶	CAN transmit buffer register 9	CTB9
001A ¹⁶	Timer XL	TXL	004A ¹⁶	CAN transmit buffer register A	CTBA
001B ¹⁶	Timer XH	TXH	004B ¹⁶	CAN transmit buffer register B	CTBB
001C ¹⁶	Timer YL	TYL	004C ¹⁶	CAN transmit buffer register C	CTBC
001D ¹⁶	Timer YH	TYH	004D ¹⁶	CAN transmit buffer register D	CTBD
001E ¹⁶	Timer X mode register	TXM	004E ¹⁶	Reserved	
001F ¹⁶	Timer Y mode register	TYM	004F ¹⁶	Reserved	
0020 ¹⁶	UART mode register	UMOD	0050 ¹⁶	CAN receive buffer register 0	CRB0
0021 ¹⁶	UART baud rate generator	UBRG	0051 ¹⁶	CAN receive buffer register 1	CRB1
0022 ¹⁶	UART control register	UCON	0052 ¹⁶	CAN receive buffer register 2	CRB2
0023 ¹⁶	UART status register	USTS	0053 ¹⁶	CAN receive buffer register 3	CRB3
0024 ¹⁶	UART transmit buffer register 1	UTBR1	0054 ¹⁶	CAN receive buffer register 4	CRB4
0025 ¹⁶	UART transmit buffer register 2	UTBR2	0055 ¹⁶	CAN receive buffer register 5	CRB5
0026 ¹⁶	UART receive buffer register 1	URBR1	0056 ¹⁶	CAN receive buffer register 6	CRB6
0027 ¹⁶	UART receive buffer register 2	URBR2	0057 ¹⁶	CAN receive buffer register 7	CRB7
0028 ¹⁶	Port P0 pull-up control register	PUP0	0058 ¹⁶	CAN receive buffer register 8	CRB8
0029 ¹⁶	Port P1 pull-up control register	PUP1	0059 ¹⁶	CAN receive buffer register 9	CRB9
002A ¹⁶	Port P2 pull-up control register	PUP2	005A ¹⁶	CAN receive buffer register A	CRBA
002B ¹⁶	Port P3 pull-up control register	PUP3	005B ¹⁶	CAN receive buffer register B	CRBB
002C ¹⁶	Port P4 pull-up/down control register	PUP4	005C ¹⁶	CAN receive buffer register C	CRBC
002D ¹⁶	Interrupt polarity selection register	IPOL	005D ¹⁶	CAN receive buffer register D	CRBD
002E ¹⁶	Watchdog timer register	WDT	005E ¹⁶	Reserved	
002F ¹⁶	Polarity control register	PCON	005F ¹⁶	Reserved	

**MITSUBISHI SEMICONDUCTORS
USER'S MANUAL
7630 Group**

JAN. Second Edition 1999

Edited by
Committee of editing of Mitsubishi Semiconductor USER'S MANUAL

Published by
Mitsubishi Electric Corp., Semiconductor Marketing Division

This book, or parts thereof, may not be reproduced in any form without permission of Mitsubishi Electric Corporation.

©1999 MITSUBISHI ELECTRIC CORPORATION

User's Manual
7630 Group

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