Your First and Best MCU

MQ6935 Datasheet

V1.3

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Content

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

1. Change History

Version	Approved Date	Description
V0.9	2018/4/16	1st issued.
V1.0	2018/7/30	1. Add “chapter 13 Serial PROM Mode”, “chapter 14 Serial Bus Interface(SBI)/ I2C”, and “chapter 15 Synchronous Serial Interface (SIO)” 2. Supplement. Chapter 10 Timer / Counter “
V1.1	2018/9/10	1. Add “7.5 Maskable Interrupt Priority Change Function.”
V1.2	2018/11/13	1. Add “3.7. Power-on reset characteristics” and “3.8 LVD characteristics” 2. Modify “4.2.6. Special Function Register” 3. Modify “3.6. MTP Characteristics” 4. Add configuration for 16-bit timer/counter, 10-bit timer/counter,8-bit timer/counter. 5. Modify “8. I/O Port” add I/O figures. 6. Add “13.8 Security” chapter. 7. Add “Appendix B Product Number Information”
V1.3	2019/2/13	1. Update “2.4 pin description “, the description of P00,P01,P02, and P03. 2. Add “5.1.4.9 Trimming Data Reset” 3. Update “Appendix C Package Dimensions.”

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

2. Product Overview

2.1 Features

◆ Basic Information

- ■ Operating voltage: 2.0 ~ 5.5V
- ■ Operating temperature: -40°C ~ 85°C
- ■ Powerful i87 8-bit MCU core (1T instruction cycle)
  - Instruction execution time
    62.5 ns (@ 16MHz)
    122us (@ 32.768kHz)
  - 133 types and 732 basic instructions.

◆ Memory Configuration

- ■ 16384 bytes program memory ROM
- ■ 16368 bytes MTP
- ■ 2K bytes RAM

◆ I/O Pin Configuration

- ■ 40 bi-directional I/Os
- ■ Large current output :14 pins(typ. 20mA)
- ■ 4-CH 8-bit PWM output (or counter input)
- ■ 2-CH 16-bit PWM output (or counter input)
- ■ 8 external key-on wakeup pins

◆ 2 Sets (Total 8 Levels) Low Voltage Detection (LVD)

◆ Power-on reset circuit

◆ Divider detection circuit

◆ 30 Interrupt Sources

- ■ 24 internal interrupts
- ■ 6 external interrupt input pins

◆ 10-bit AD Converter (ADC)

- ■ 8 external ADC input
- ■ External AD reference voltage

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

◆ Flexible Operation Modes

- Low power consumption:
  - STOP mode: Oscillation stops. (Battery/Capacitor back-up.)
  - SLOW1 mode: Low power consumption operation using low-frequency clock.(High-frequency clock stop.)
  - SLOW2 mode: Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)
  - IDLE0 mode: CPU stops, and only the Time-Based-Timer (TBT) on peripherals operate using high frequency clock. Released when the reference time set to TBT has elapsed.
  - IDLE1 mode: The CPU stops, and peripherals operate using high frequency clock. Release by interrupts (CPU restarts).
  - IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interrupts. (CPU restarts).
  - SLEEP0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock. Released when the reference time set to TBT has elapsed.
  - SLEEP1 mode: CPU stops, and peripherals operate using low frequency clock. Release by interrupts.(CPU restarts).

◆ Timer / Counter Information

- Four 8-bit timers (Configurable to Two 16-bit timers)
- Timer, Event Counter, PWM, PPG OUTPUT modes
  - - Usable as a 16-bit timer, 12-bit PWM output and 16-bit PPG output by the cascade connection of two channels.
- One 10-bit Timer
  - 2ports output PPG (Programmed Pulse Generator)
  - Variable Duty output mode
  - 50%duty output mode
  - External-triggered start and stop
  - Emergency stop pin
- Two 16-bit Timer
  - Timer. External trigger, Event counter, Window , Pulse width measurement, PPG output modes.
- Time base timing generator (TBT)
- Watch-dog Timer (WDT)
  - Interrupt or reset can be selected by the program.
- Warm-up Counter (WUC)
- Real Time Clock (RTC)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

(with 8-bit frequency divider)

- ◆ **Clock Sources**
  - ■ External crystal or internal oscillator
  - ■ Internal oscillator frequency 8MHz
  - ■ Support 1MHz ~ 16MHz or 32768Hz external crystal
- ◆ **ISP (In-System Programming) Function**
- ◆ **2CH-UART**
  - ■ 3 set of UART pins (TX/RX)
- ◆ **UART/SIO : 1ch**
  - ■ one SIO channel can be used at the same time
- ◆ **I2C/SIO : 1ch**
- ◆ **I2C**
- ◆ **On-Chip Debugging**
  - ■ Break/Event/Trace
  - ■ RAM monitor & Flash/MTP memory writing
- ◆ **Clock operation mode control circuit: 2 circuit.**
  - ■ Single clock mode /Dual clock mode
- ◆ **Package Type**
  - ■ LQFP44 (10x10)

2.2 Preface

MQ6935 has a powerful i87 8-bit MCU core embedded with real 10-bit ADC (Analog to Digital Converter) function. For general functions of the MCU, such as registers and flags of the CPU, timers / counter information and reset / detection circuit, please refer to "iMQ i87 User Manual" for details. Specific functions of MQ6935 such as program / data memory, special function register, operation modes of the CPU, interrupts, system clocking and I/O port information are listed in the following sections.

Please note that in this document, 64K Bytes or smaller memory style is used. Therefore, the address format will be 0x0000 to 0xFFFF.

Note that pin names with low-active values, such as $\overline{\text{RESET}}$, $\overline{\text{PWM00}}$, $\overline{\text{PWM01}}$, $\overline{\text{PWM02}}$, $\overline{\text{PWM03}}$, $\overline{\text{PPGA0}}$, $\overline{\text{DVO}}$ and so on, are presented by ending with "B" in the content, meaning "bar" for inversion. Therefore, they are written as RESETB, PWM00B, PWM01B, PWM02B, PWM03B, PPGA0B, DVOB and so on.

Besides, to indicate certain bit name in a register, the representation REGISTER_NAME <BIT_NAME> is used in this document. For example, ILL <IL5> indicates the bit IL5 of the ILL register.

2.3 Block Diagram

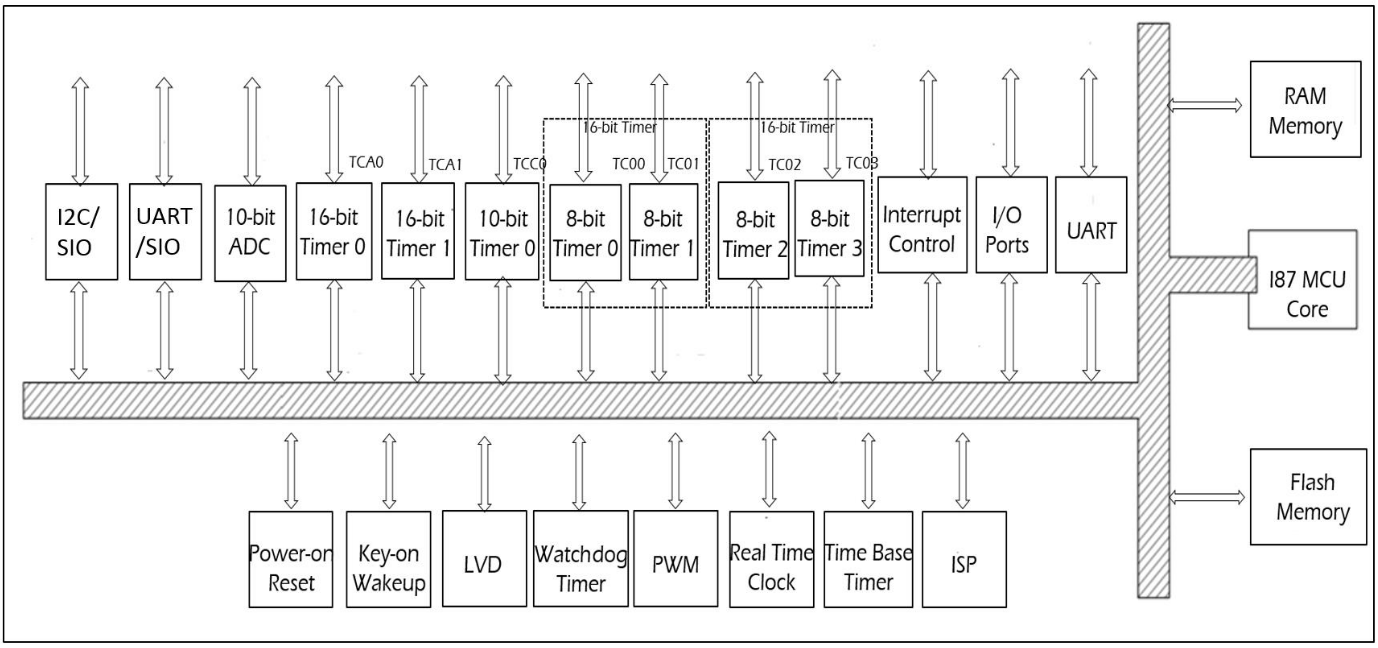
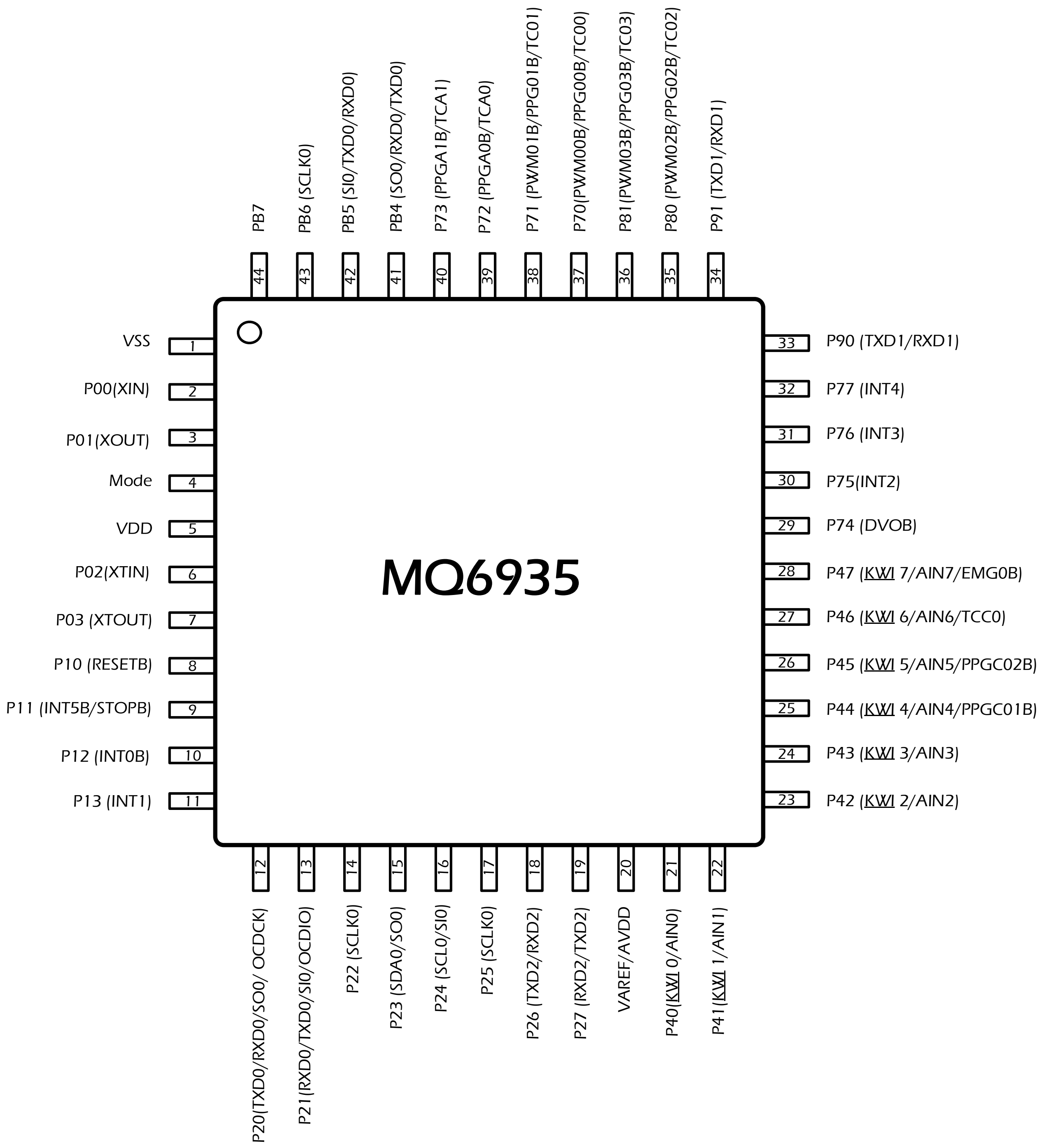


Figure 2.1 Block diagram of MQ6935

2.4 Pin Assignment / Description

LQFP44 (10x10)



iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

44-Pin No.	Pin Name/ Pin Option 1 (Note3)	I/O Type		Function Description
1	VSS	Power	–	Negative power supply / ground

2
3
	P00(XIN)
	P01(XOUT)
		I/O	Pull-up(Note 1)
			Ext. crystal (high)
				P00 and P01 are bi-directional I/O pins, which are software configurable to be with pull-up resistors. XIN and XOUT are pin-shared with P00 and P01 respectively, and are connected to high frequency external crystal for system clock. The XOUT oscillation amplitude is about 1.8V (this voltage value may have some variation due to resistance of measurement circuit or prober). This is because the working voltage of internal circuit is 1.8V to reduce the power consumption.

4	Mode	I	Test mode	This pin is connected to VSS with 10Kohm resistor during user mode and connected to VDD during programming data into flash memory.
5	VDD	Power	–	Positive power supply
6 7	P02(XTIN) P03 (XTOUT)	I/O	Pull-up(Note 1) Ext. crystal (low)	P02 and P03 are bi-directional I/O pins, which are software configurable to be with pull-up resistors. XTIN and XTOUT are pin-shared with P02 and P03 respectively, and are connected to a low frequency external crystal for system clock. The XTOUT oscillation amplitude is about 1.8V (this voltage value may have some variation due to resistance of measurement circuit or prober). This is because the working voltage of internal circuit is 1.8V to reduce the power consumption.
8	P10 (RESETB)(Note 7)	I/O	Pull-up(Note 1)	P10 is a bi-directional I/O pin, which is software configurable to be with pull-up resistor. RESETB is pin-shared with P10, which is low-active.
9 10 11	P11 (INT5B/STOPB) (Note5) P12 (INT0B) P13 (INT1)	I/O	Pull-up(Note 1) Ext. interrupt	P11, P12 and P13 are bi-directional I/O pins, which are software configurable to be with pull-up resistors. INT5B/STOPB, INT0B and INT1 are pin-shared with P11, P12 and P13 respectively. INT0B, INT5B and STOPB are low-active.
12 13	P20 (TXD0/RXD0/SO0/ OCDCK) P21 (RXD0/TXD0/SI0/OCDIO)	I/O	Pull-up(Note 2) OCDE Serial data input Serial data output Open-drain	P20 and P21 are bi-directional I/O pins, which are software configurable to be with pull-up resistors. TXD0/RXD0/OCDCK(OCD clock input) and RXD0/TXD0/OCDIO(OCD data input/output) of UART I/Os are pin-shared with P20 and P21 respectively. Serial data output 0 (SO0) and serial data input 0 (SI0) are pin shared with P20 and P21 respectively.
14	P22 (SCLK0)	I/O	Serial clock input/output 0 Pull-up(Note 2) Open-drain	P22 is a bi-directional I/O pin, which is software configurable to be with pull-up resistors as input mode or open-drain output mode. Serial clock input/output 0 is pin-shared with P22.

44-Pin No.	Pin Name	Pin Option 1 (Note3)		I/O Type
15 16	P23 (SDA0/SO0)(Note 6) P24 (SCL0/SI0) (Note 6)	I/O	I2C Serial data input Serial data output Open-drain	P23 and P24 are bi-directional I/O pins. SDA0 (I2C bus data input/output 0) and SCL0(I2C bus clock input/output 0) are pin-shared with P23 and P24 respectively. SO0 (serial data output) and SI0(serial data input) are pin-shared with P23 and P24.
17	P25 (SCLK0)	I/O	Serial clock input/output 0 Pull-up(Note 2) Open-drain	P25 is a bi-directional I/O pin, which is software configurable to be with pull-up resistors as input mode or open-drain output mode. Serial clock input/output 0 is pin-shared with P25.
18 19	P26 (TXD2/RXD2) P27 (RXD2/TXD2)	I/O	UART	P26 and P27 are bi-directional I/O pins. UART TXD2/RXD2 are pin shared with P26 and P27.
20	VAREF/AVDD	I	–	This pin is ADC external reference voltage (VAREF) pin. This pin needs to be connected to VDD whether ADC is operating or not.

21
22
23
24
	P40(KWI0/AIN0)
	P41(KWI1/AIN1)
	P42 (KWI2/AIN2)
	P43 (KWI3/AIN3)
		I/O	Pull-up(Note3)
			Wakeup
			ADC input
				P40, P41, P42 and P43are bi-directional I/O pins, which are software configurable to be with pull-up resistors. ADC input pins AIN0, AIN1, AIN2, AIN3 and wakeup pins KWI0, KWI1, KWI2, KWI3 are pin-shared with P40, P41, P42, P43 respectively.

25 26	P44 (KWI4/AIN4/PPGC01B) P45 (KWI5/AIN5/PPGC02B)	I/O	Pull-up(Note3) Wakeup ADC input PPG output	P44 and P45 are bi-directional I/O pins, which are software configurable to be with pull-up resistors. ADC input pins AIN4, AIN5, wakeup pins KWI4, KWI5 and 10-bit timer/counter pin PPGC01B, PPGC02B are pin-shared with P44, P45 respectively.
27	P46 (KWI6/AIN6/TCC0)	I/O	Pull-up(Note3) Wakeup ADC input Timer/Counter	P46 is bi-directional I/O pins, which is software configurable to be with pull-up resistors. ADC input pins AIN6 and wakeup pins KWI6, and 10-bit timer/counter pin TCC0 is pin-shared with P46.
28	P47 (KWI7/AIN7/EMG0B)	I/O	Pull-up(Note3) Wakeup ADC input Timer/Counter	P47 is bi-directional I/O pins, which is software configurable to be with pull-up resistors. ADC input pins AIN7, wakeup pins KWI7 and 10-bit timer/counter pin EMG0B is pin-shared with P47 respectively.
29	P74 (DVOB)	I/O	Divider Output	P74 is a bi-directional I/O pin. DVOB is shared with P75.
30 31 32	P75 (INT2) P76 (INT3) P77 (INT4)	I/O	Ext. interrupt	P75, P76 and P77 are bi-directional I/O pins. 3 interrupts INT2, INT3 and INT4 are shared with P75, P76 and P77.
33 34	P90 (TXD1/RXD1) P91 (TXD1/RXD1)	I/O	UART	P90 and P91 are bi-directional I/O pins. UART TXD1/RXD1 are pin shared with P90 and P91.

44-Pin No.	Pin Name	Pin Option 1 (Note3)		I/O Type
35 36	P80 (PWM02B/PPG02B/TC02) P81(PWM03B/PPG03B/TC03)	I/O	Timer/Counter PWM PPG	P80 and P81 are bi-directional I/O pins. 8-bit timer/counter pins TC02/PWM02B/PPG02B and TC03/PWM03B/PPG03B are pin-shared with P80 and P81 respectively. PWM02B/PPG02B and PWM03B/PPG03B are low-active.
37 38	P70(PWM00B/PPG00B/TC00) P71 (PWM01B/PPG01B/TC01)	I/O	Timer/Counter PWM PPG	P70 and P71 are bi-directional I/O pins. 8-bit timer/counter pins TC00/PWM00B/PPG00B and TC01/PWM01B/PPG01B are pin-shared with P70 and P71 respectively. PWM00B/PPG00B and PWM01B/PPG01B are low-active.
39 40	P72 (PPGA0B/TCA0) P73 (PPGA1B/TCA1)	I/O	Timer/Counter PWM PPG	P72 and P73 are bi-directional I/O pins. 16-bit timer/counter pin PPGA0B/TCA0 and PPGA1B/TCA1 are pin-shared with P72 and P73.
41 42	PB4 (SO0/RXD0/TXD0) PB5 (SI0/TXD0/RXD0)	I/O	Serial data output Serial data input UART	PB4 and PB5 are bi-directional I/O pins. Serial data output 0 (SO0) and UART (RXD0/TXD0) is pin-shared with PB4. Serial data input 0 (SI0) and UART (TXD0/RXD0) is pin-shared with PB5.
43	PB6 (SCLK0)	I/O	Serial clock input/output	PB6 is a bi-directional I/O pin. SCLK0 (Serial clock input/output) is pin-shared with PB6.
44	PB7	I/O	-	PB7 is a bi-directional I/O pin.

Note 1: Pull-up resistor is connected when the pin is set as input mode.
Note 2: Pull-up resistor is connected when the pin is set as input mode or open-drain output mode.
Note 3: Pull-up resistor is connected when the pin is set as input mode or wakeup.
Note 4: There are 6 pins needed to programming the data into flash memory: VDD, VSS, Mode, P10(RESETB), P20 and P21.
Note5: Please connect P11 (STOPB) to ground with a resistor (e.g. 10Kohm) otherwise MCU would be wake-up and couldn't be kept in STOP mode. This is because P11 will be set as input mode automatically in STOP mode operation. If P11 is set with pull-up resistor to VDD, STOP mode will be released.
Note 6: Please prevent P23 and P24 from being floated otherwise STOP mode current will be affected by additional leakage current in IO pads. When P23 and P24 need to pull-high, P23 and P24 should connect to a resistor 1Kohm.
Note 7: After power-on, P10 is set to be RESETB function as default, and is software configurable to be I/O pin so please don't connect P10 to ground.

There are 2 recommended external application circuits and some suggestions for MQ6935:

1. VDD pin and AVDD pin need to be equal potential.

2. Mode pin need to be connected 10K ohm pull-down resistor.

3. When ADC function is used, the voltage signal connected to AIN should be through a resistor (100 ohm). Besides, 1nF capacitor should be added in AIN pins. The purpose is to filter the noise.

4. When ADC function is used, (10uF + 0.1uF) capacitors should be added in VAREF pin. The purpose is to

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

filter noise and keep ADC external reference voltage (VAREF) stable.

5. Recommended copper pouring as grounding.

6. VDD pin and AVDD pin need to be equal potential. If the power sources of VDD and AVDD are different, 10uF and 0.1uF capacitor need to be connected respectively, 0.1uF capacitor should be close to IC as much as possible. (Figure 2.2)

7. VDD pin and AVDD pin need to be equal potential. If power source and the routing are the same, connecting 0.1uF capacitor where is near IC is recommended. (Figure 2.3)

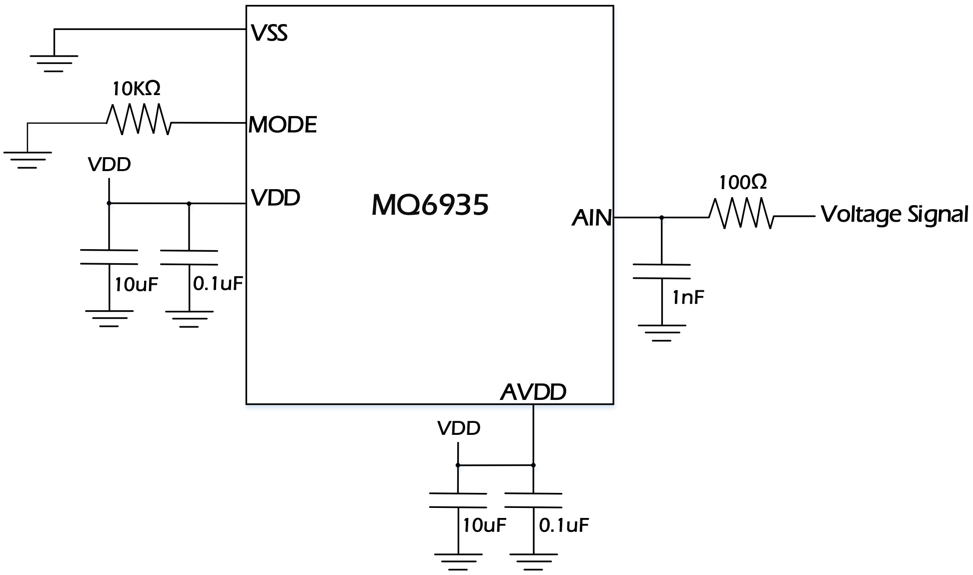


Figure 2.2 Recommended external circuit when using MQ6935 (different power sources for VDD & AVDD)

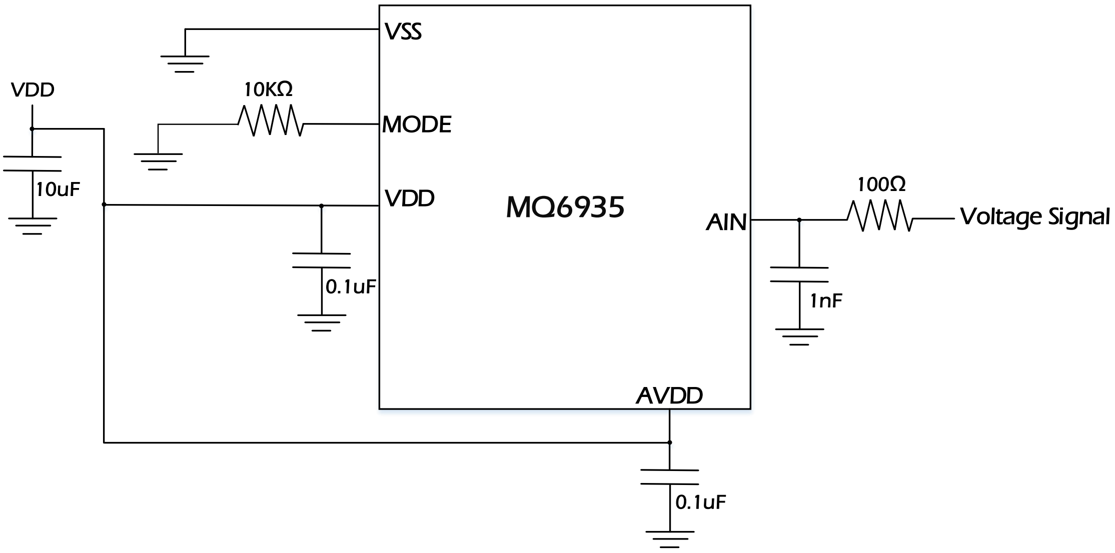
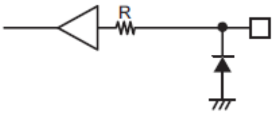


Figure 2.3 Recommended external circuit when using MQ6935 (same power source for VDD & AVDD)

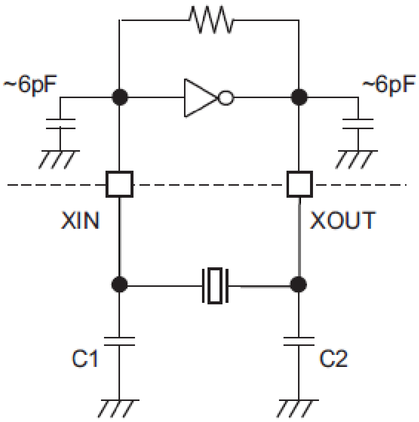
iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

2.5 Control Pins

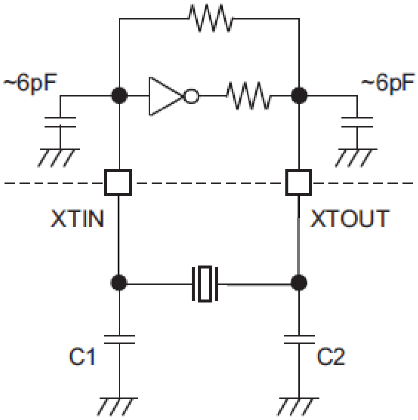
The input/output circuitries of the control pins are shown below.

Control pin	I/O	Circuitry	Remarks
XIN XOUT	Input Output	Refer to the P0 ports in the chapter of Input/Output Ports.	
XTIN XTOUT	Input Output	Refer to the P0 ports in the chapter of Input/Output Ports.	
$\overline{\text{RESET}}$	Input	Refer to the P1 ports in the chapter of Input/Output Ports.	
MODE	Input		R = 100 Ω (typ.)

2.6 Oscillating Condition



High-frequency oscillation



Low-frequency oscillation

Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: There are on-chip capacitors on XIN/XOUT and XTIN/XTOUT pins (approximatery 6pF each). So the load capacitance C1 and C2 can be reduced, or even omitted, depending on models of crystal / resonator used, the oscillation frequency accuracy required and system board design.

3. Electronic Characteristics

3.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(V_{SS} = 0V)

Parameter	Symbol	Pins	Ratings	Unit
Supply Voltage	V_{DD}		-0.3 to 6.0	V
Input Voltage	V_{IN1}	All I/O pins	-0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT}	All I/O pins	-0.3 to V_{DD} + 0.3	V
Output Current (per-pin)	I_{OUT1}	P0, P1, P2 (excluding P23 and P24), P4, P7,P8, P9, PB (tri-state port)	-1.8	mA
	I_{OUT2}	P0, P1, P2, P4, P9 (pull-up resistor)	-0.4	
	I_{OUT3}	P0, P1, P2, P4, P9 (tri-state port)	3.2	
	I_{OUT4}	P7, P8, PB (large current port)	30	
Output Current (total)	ΣI_{OUT1}	P0, P1, P2 (excluding P23 and P24), P4, P7,P8, P9, PB (tri-state port)	-30	mA
	ΣI_{OUT2}	P0, P1, P2, P4, P9 (pull-up resistor)	-4	
	ΣI_{OUT3}	P0, P1, P2, P4, P9 (tri-state port)	60	
	ΣI_{OUT4}	P7, P8, PB (large current port)	120	
Power dissipation (Topr = 85°C)	P_D		250	mW
Soldering temperature (time)	T_{sld}		260 (10 s)	°C
Storage Temperature	T_{STG}		-40 to 125	
Operating Temperature	T_{OPR}		-40 to 85	

3.2 Operation Conditions

(V_{SS} = 0V, T_{OPR} = -40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply Voltage	V_{DD}		All operation modes	2.0	5.5	V
Input High Level	V_{IH}	All I/O pins		V_{DD} x 0.7	V_{DD}	V
Input Low Level	V_{IL}	All I/O pins		0	V_{DD} x 0.3	V
Clock Frequency	f_C	XIN, XOUT	V_{DD} = 2.0 to 5.5V	1.0	16	MHz
	f_S	XTIN, XTOUT		30.0	34.0	KHz
	f_{OSC} (Note)	Internal Oscillator	V_{DD} = 2.0 to 5.5V, 0°C ~85°C	7.76	8.24	MHz
		Internal Oscillator	V_{DD} = 2.0 to 5.5V, -40°C ~85°C	7.68	8.32	MHz
	f_{cgck}		V_{DD} = 2.0 to 5.5V	0.25	16	MHz

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

3.3 D.C. Characteristics

(V_{SS} = 0V, T_{OPR} = -40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ	Max	Unit
Hysteresis Voltage	V_{HS}	All IO pins	V_{DD} = 5.5V V_{IN} = 5.5V / 0V	–	0.9	–	V
Input Current	I_{IN1}	MODE		–	–	±2	μA
	I_{IN2}	P0, P1, P2, P4, P7, P8 P9, PB					
	I_{IN3}	RESETB, STOPB					
Pull-up Resistance	R_{UP1}	P10 with RESETB enable	V_{DD} = 5.5V V_{IN} = 0V	100	220	500	KΩ
	R_{UP2}	P0, P1, P2 (excluding P23 and P24), P4		30	50	100	KΩ
Output leakage current	I_{LO1}	P23, P24 (skin open drain port)	VDD = 5.5 V VOUT = 5.5 V	-	-	2	μA
	I_{LO2}	P0, P1, P2 (excluding P23 and P24), P4, P5, P7, P8, P9, PB (tristate port)	VDD = 5.5 V VOUT = 5.5 V/0 V	-	-	±2	
Output high voltage	V_{OH}	Except P23, P24, XOUT, XTOUT	VDD = 4.5 V IOH = − 0.7 mA	4.1	-	-	V
Output low voltage	V_{OL}	Except XOUT, XTOUT	VDD = 4.5 V IOL = 1.6 mA	-	-	0.4	
Output Current	I_{OL1}	P7, P8, PB (large current port)	VDD = 4.5 V VOL = 1.0 V	-	20	-	mA

Note1: Typical values shows those at T_{OPR} = 25°C and V_{DD} = 5.0V.

Note 2: Input current I_{IN3} : The current through pull-up resistor is not included.

Note 3: VIN : The input voltage on the pin except MODE pin, VMODE : The input voltage on the MODE pin

(V_{SS} = 0V, T_{OPR} = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Supply Current in NORMAL 1, 2 modes	I_{DD}	V_{DD} = 5.5V f_{cgck} = 16.0 MHz f_S = 32.768 KHz	–	5.5	8.5	mA
Supply Current in IDLE0, 1, 2 modes			–	3.0	4.5	
Supply Current in NORMAL 1, 2 modes		V_{DD} = 5.5V f_{cgck} = 8.0 MHz f_S = 32.768 KHz	–	3.5	5.5	
Supply Current in IDLE0, 1, 2 modes			–	2.0	3.5	
Supply Current in SLOW1 modes (Note 3)		V_{DD} = 3.0V f_S = 32.768 KHz	–	40	150	µA
Supply Current in SLEEP1 modes			–	25	145	
Supply Current in SLEEP0 modes			–	20	145	
Supply Current in STOP modes		V_{DD} = 5.5V (-40 to 85°C)	–	10	120	
		V_{DD} = 5.5V (-40 to 40°C)	–	10	35	

Note 1): Typical values shown are T_{OPR} = 25°C and V_{DD} = 5.0V, unless otherwise specified.

Note 2): V_{IN}: The input voltage on I/O pins.

Note 3): Each supply current in SLOW2 mode is equivalent to that in IDLE0, IDLE1 and IDLE2 modes.

3.4 AD Conversion Characteristics

(V_{SS} = 0V, 2.7V ≦ V_{DD}≦ 5.5V, T_{OPR} = 25°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Analog Reference Voltage	V_{AREF}/ A_{VDD}	–		V_{DD}		V
Analog input voltage range	V_{AIN}	–	V_{SS}	–	V_{AREF}	V
Conversion Time		–	–	16.0	–	µs
Differential Nonlinearity Error		–	–	–	±2.0	LSB
Integral Nonlinearity Error		–	–	–	±2.0	LSB
Zero Point Error		–	–	–	±2.0	LSB
Full Scale Error		–	–	–	±2.0	LSB
Total Error		–	–	–	±2.0	LSB

(V_{SS} = 0V, 1.8V ≦ V_{DD} < 2.7V, T_{OPR} =25°C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Analog Reference Voltage	V_{AREF}/ AV_{DD}	–	V_{DD}			V
Analog input voltage range	V_{AIN}	–	V_{SS}	–	V_{AREF}	V
Conversion Time		–	–	32.0	–	µs
Differential Nonlinearity Error		–	–	–	±4.0	LSB
Integral Nonlinearity Error		–	–	–	±4.0	LSB
Zero Point Error		–	–	–	±4.0	LSB
Full Scale Error		–	–	–	±4.0	LSB
Total Error		–	–	–	±4.0	LSB

Note 1): The total error includes all errors except a quantization error, and is defined as the maximum deviation from the ideal conversion line.

Note 2): The voltage to be input to the AIN input pin must be within the range V_{AREF} to V_{SS}. If a voltage outside this range is input, converted values will become indeterminate, and converted values of other channels will be affected.

Note 3): If the AD converter is not used, fix the pin to the V_{DD}.level.

Note 4) When using ADC, please refer to「Figure 2.2 Recommended external circuit when using MQ6935」.

3.5 Flash Characteristics

(V_{SS} = 0V, 2.0V ≦ V_{DD}≦ 5.5V, T_{OPR} = -40 to 85°C)

Parameter	Condition	Min	Typ	Max	Unit
Number of guaranteed writes to flash memory		–	–	100,000	Times
Flash memory write time		–	–	40	µs
Flash memory erase time	Chip erase	–	–	40	ms
	Block erase (1KB)			40	
	Sector erase (128 Bytes)	–	–	5	

3.6 MTP Characteristics

(V_{SS} = 0V, 2.0V ≦ V_{DD}≦ 5.5V, T_{OPR} = -40 to 85°C)

Parameter	Condition	Min	Typ	Max	Unit
Number of guaranteed writes to MTP		–	–	100,000	Times
MTP write time		–	–	40	µs
MTP memory erase time	Chip erase	–	–	40	ms
	Block erase (1KB)			40	
	Sector erase (128 Bytes)	–	–	5	

3.7 Power-on Reset Characteristics

Ta=-40~85℃

Symbol	Condition	Min	Typ.	Max	Unit
VPROFF	Power-on reset releasing voltage	1.45	1.6	1.75	V
VPRON	Power-on reset detecting voltage	1.35	1.5	1.65	V
tPROFF	Power-on reset releasing response time	-	0.01	0.1	ms
tPRON	Power-on reset detecting response time	-	0.01	0.1	ms
tPRW	Power-on reset minimum pulse width	1.0	-	-	ms
tPWUP	Warming-up time after a reset is clear	-	102 x 2^{10}/fc	-	S
tVDD	Power supply rise time	-	-	5	ms

Note 1: Because the power-on reset releasing voltage and the power-on reset detecting voltage change relative to one another, the detected voltage will never become inverted.

Note 2: A clock output by an oscillating circuit is used as the input clock for a warming-up counter. Because the oscillation frequency does not stabilize until an oscillating circuit stabilizes, some errors may be included in the warming-up time.

Note 3: Boost the power supply voltage such that tVDD becomes smaller that tPWUP.

Note 4: When turning on the power, it's fc=fosc.

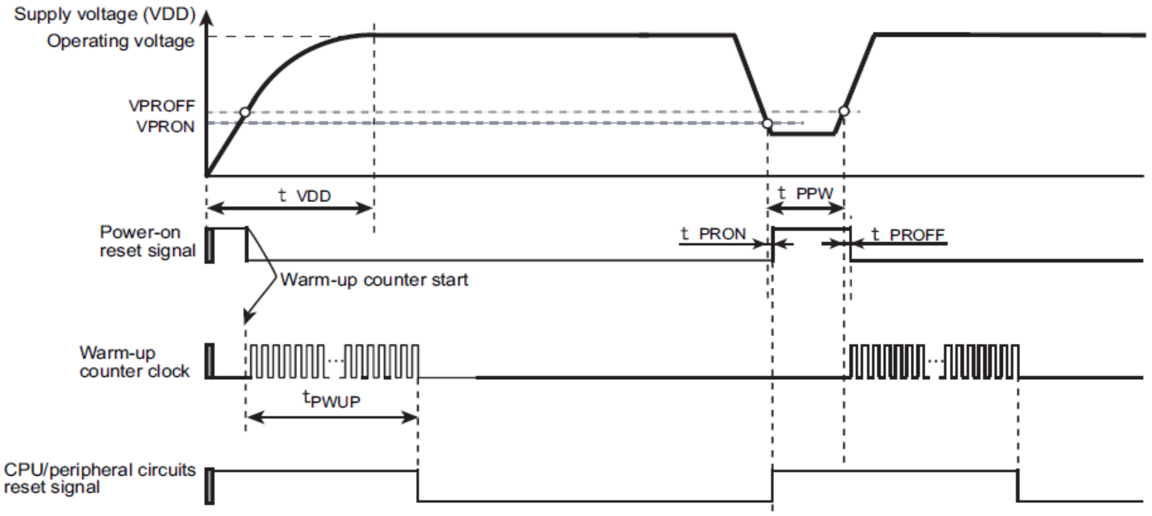


Figure 3.1 Operation Timing of Power-on Reset

Note 1): The power-on reset circuit may operate improperly, depending on fluctuations in the supply voltage (VDD). Refer to the electrical characteristics and take them into consideration when designing equipment.

3.8 LVD Characteristics

Ta=-40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
LVD	VLVD1	VD2LVL=00	1.90	2.00	2.10	V
	VLVD2	VD2LVL=01	2.25	2.35	2.45	V
	VLVD3	VD2LVL=10	2.55	2.65	2.75	V
	VLVD4	VD2LVL=11	2.75	2.85	2.95	V
	VLVD5	VD1LVL=00	3.00	3.15	3.30	V
	VLVD6	VD1LVL=01	3.55	3.70	3.85	V
	VLVD7	VD1LVL=10	4.05	4.20	4.35	V
	VLVD8	VD1LVL=11	4.35	4.50	4.65	V

Ta=-40~85℃

Symbol	Parameter	Min	Typ.	Max	Unit
tVLTOFF	Voltage detection releasing response time	-	0.01	0.1	‘ms
tVLTON	Voltage detecting detection response time	-	0.01	0.1	‘ms
tVLTPW	Voltage detecting minimum pulse width	1.0	-	-	‘ms

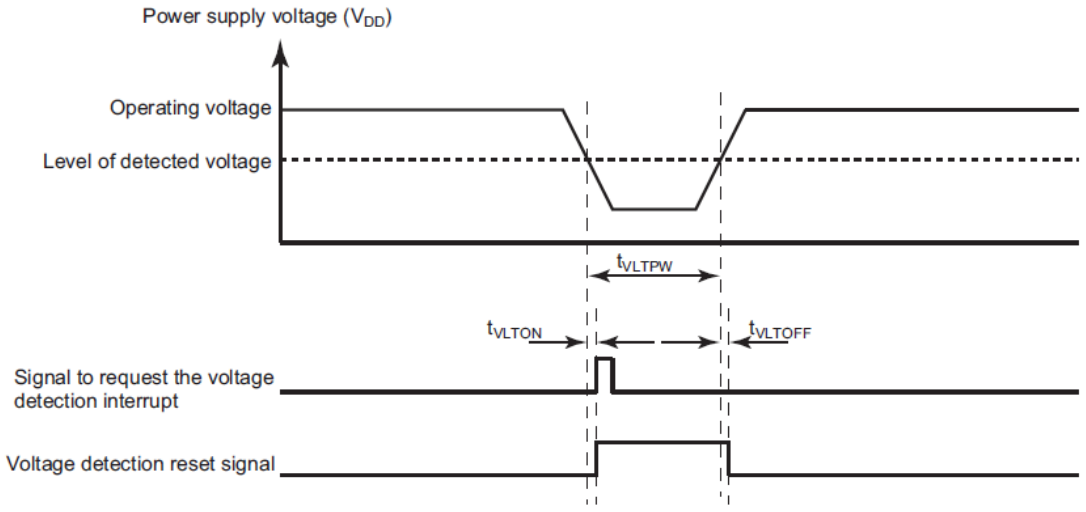

Power supply voltage (V_{DD})
Operating voltage
Level of detected voltage
t_{VLTPW}
t_{VLTON}
t_{VLTOFF}
Signal to request the voltage detection interrupt
Voltage detection reset signal

Figure 3.2 Operation Timing of Voltage Detecting Circuit

Note: Care must be taken in system designing since the power-on reset circuit may not fulfill its functions due to the fluctuations in the power supply voltage (VDD).

3.9 A.C. Characteristics

(V_{SS} = 0V, VDD=2.0~5.5V, T_{OPR} = -40 to 85˚C)

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.0625	-	4	us
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	-	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input). fc = 16.0 MHz	-	31.25	-	‘ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	-	15.26	-	‘us
Low-level clock pulse width	t_{WSL}					

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

4. Central Processing Unit (CPU)

4.1 General Concept

MQ6935 adopts i87 8-bit MCU core with embedded Flash memory (16KB, i.e. 16384 bytes). Besides, MQ6935 also has 16368 bytes of MTP (multi-times programmable memory). The introduction of the powerful central processing unit (CPU) can be divided into eight major parts: (1) Addressing Space of Program / Data Memory and Special Function Registers (SFR), (2) Operation Modes, (3) Stack Area / Pointer, (4) Program Counter (PC), (5) General Purpose Registers, (6) Program Status Word (PSW), (7) Low Power Consumption Function and (8) Key-on Wakeup.

The 870/C1 CPU memory space consists of a code area to be accessed as instruction operation codes and operands and a data area to be accessed as sources and destinations of transfer and calculation instructions. Both the code and data areas have independent 64-Kbyte address spaces.

4.2 Addressing Space

4.2.1 Data Area

Figure 4.1 shows the data area of MQ6935, including SRF1, SRF2, SFR3, RAM and program memory (Flash) memory. If there is rolling code request, suggest to reserve 0x000~0x800F to save rolling code.

The data area stores the data to be accessed as sources and destinations of transfer and calculation instructions. The SFR, the RAM, the BOOTROM and the FLASH are mapped in the data area.

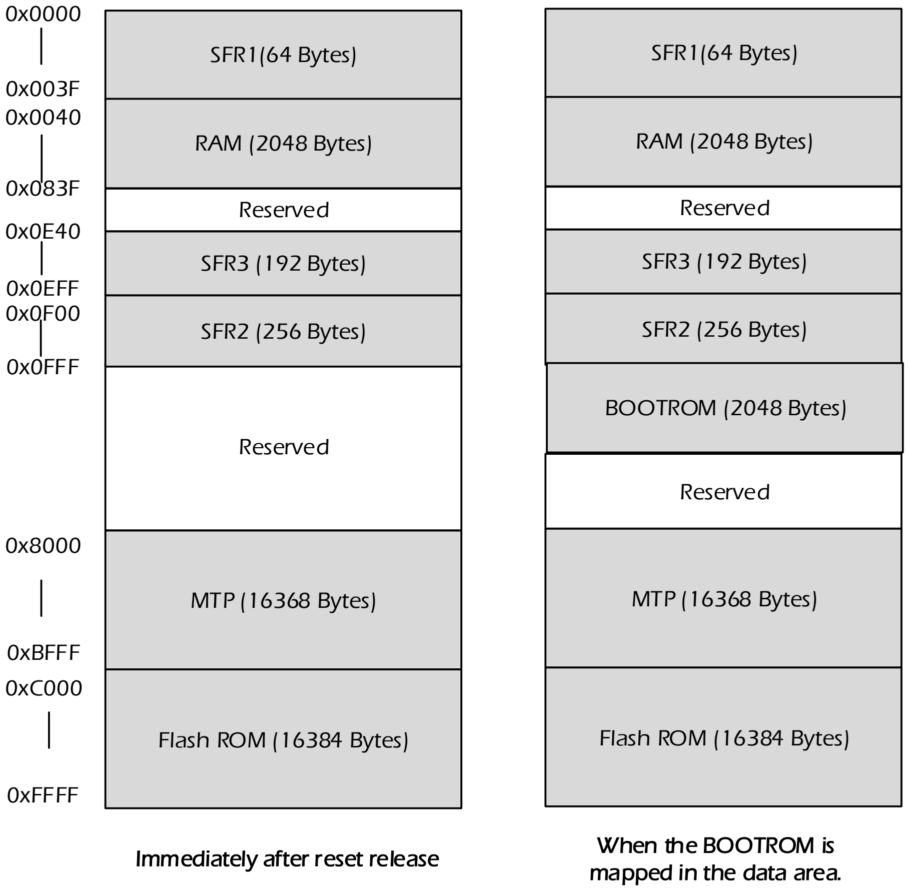


Figure 4.1Addressing Map in the Data Area

Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

4.2.2.1 RAM

The RAM is mapped to 0x0040 to 0x083F in the data area after reset release.

Note:The contents of the RAM become unstable when the power is turned on and immediately after a reset is released.

To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

Example: RAM initialization program

```
LD       HL,RAM_TOP_ADDRESS          ;Head of address of the RAM to be initialized
LD       A,0x00                      ;Initialization data
LD       BC,BYTE_OF_CLEAR_BYTES      ;Number of bytes of RAM to be initialized-1
LD       (HL),A                      ;Initialization of the RAM
INC      HL                          ;Initialization address increment
DEC      BC                          ; Have all the RAMs been transferred?
J        F,TRANS_RAM
```

4.2.2.2 BOOTROM

The BOOTROM is not mapped in the code area or the data area after reset release.
Setting FLSCR1<BAREA> to "1" and writing 0xD5 to FLSCR2 maps the BOOTROM to 0x1000 to 0x17FF in the code area and to 0x1000 to 0x17FF in the data area. Flash memory can be easily programmed by using the API (Application Programming Interface) contained in the BOOTROM.

Note 1: When the BOOTROM is not mapped in the data area, 0xFF is read from 0x1000 to 0x17FF.

Note2: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

Flash Memory Control Register 1

FLSCR1 (0x0FD0)	7	6	5	4	3	2	1	0
Bit Symbol	(FLSMD)			BAREA	(FAREA)		(ROMSEL)	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	1	0	0	0	0	0	0

BAREA	Specifies mapping of the BOOTROM in the code and data areas	0: The BOOTROM is not mapped to 0x1000 to 0x17FF in the code area and to 0x1000 to 0x17FF in the data area. 1: The BOOTROM is mapped to 0x1000 to 0x17FF in the code area and to 0x1000 to 0x17FF in the data area.

Note: The flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Flash Memory Control Register 2

FLSCR2 (0x0FD1)	7	6	5	4	3	2	1	0
Bit Symbol	CR1EN							
Read/Write	W							
After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable/disable control	0xD5: Enable a change in the FLSCR1 setting. Others: Reserved.

4.2.2.3 Flash

The Flash is mapped to 0x8000 to 0xFFFF in the data area after reset release.

4.2.2 Code Area

The code area stores operation codes, operands, vector tables for vector call instructions and interrupt vector tables. The RAM, the BOOTROM and the Flash are mapped in the code area.

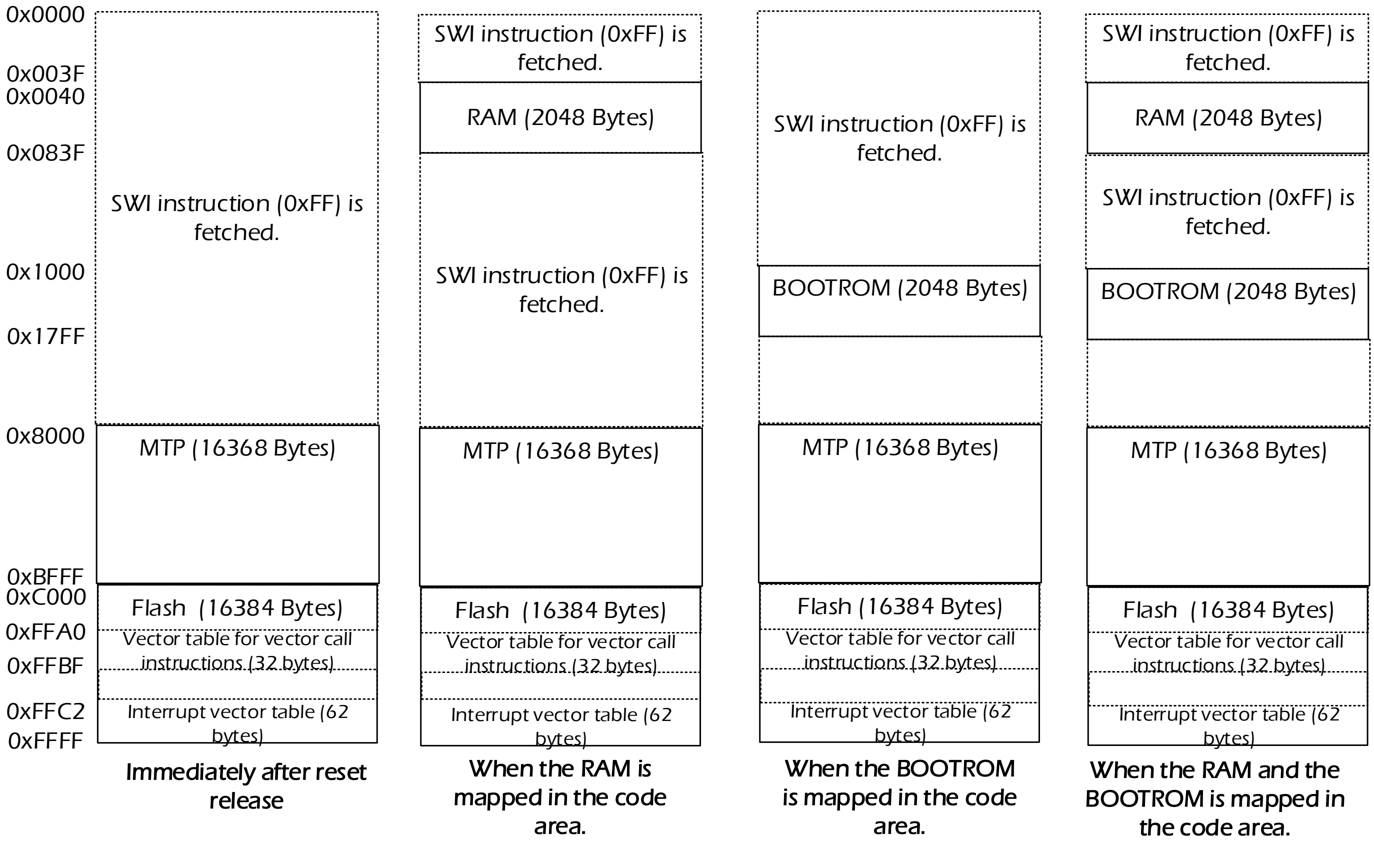

0x0000
0x003F
0x0040
0x083F
0x1000
0x17FF
0x8000
0xBFFF
0xC000
0xFFA0
0xFFBF
0xFFC2
0xFFFF
SWI instruction (0xFF) is fetched.
RAM (2048 Bytes)
BOOTROM (2048 Bytes)
MTP (16368 Bytes)
Flash (16384 Bytes)
Vector table for vector call instructions (32 bytes)
Interrupt vector table (62 bytes)
Immediately after reset release
When the RAM is mapped in the code area.
When the BOOTROM is mapped in the code area.
When the RAM and the BOOTROM is mapped in the code area.

Figure 4.2Addressing Map in the Code Area

Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

4.2.2.1 RAM

The RAM is mapped in data area after reset release.

By setting SYSCR3<RAREA> to "1" and writing 0xD4 to SYSCR4, RAM can be mapped to 0x0040to 0x083F in the code area to execute the program. At this time, by setting SYSCR<RVCTR> to "1" and writing 0xD4 to SYSCR4, vector table for vector call instructions and interrupt except reset can be mapped to RAM.

In the serial PROM mode, the RAM is mapped to 0x0040 to 0x083F in the code area, regardless of the value of SYSCR3<RAREA>. The program can be executed on the RAM using the RAM loader function.

Note 1: When the RAM is not mapped in the code area, the SWI instruction is fetched from 0x0040 to 0x083F.

Note2: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

System Control Register 3

SYSCR3 (0x0FDE)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	RVCTR	RAREA	(RSTDIS)
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

RAREA	Specifies mapping of the RAM in the code area	0: The RAM is not mapped from 0x0040 to 0x083F in the code area. 1: The RAM is mapped from 0x0040 to 0x083F in the code area.
RVCTR	Specifies mapping of the vector table for vector call instructions and interrupts	Vector table for vector call instructions 0:0xFFA0 to 0xFFBF in the code area 1: 0x01A0 to 0x01BF in the code area Vector table for interrupt 0: 0xFFC2 to 0xFFFF in the code area 1: 0x01C2 to 0x01FD in the code area
RSTDIS	External reset input enable register	0: Enable the external reset input 1: Disable the external reset input

Note 1): The enabled SYSCR3 <RSTDIS> is initialized by a power-on reset only, and cannot be initialized by an external reset input or internal factor reset. The value written in SYSCR3 is reset by a power-on reset, external reset input or internal factor reset.

Note 2): The value of SYSCR3 <RSTDIS> is invalid until 0xB2 is written into SYSCR4.

Note 3): After SYSCR3 <RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3 <RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR <FCGCKSEL> = 00). Otherwise, SYSCR 3<RSTDIS> may be enabled at unexpected timing.

Note 4): Bits 7 to 3 of SYSCR3 are read as "0".

System Control Register 4

SYSCR4 (0x0FDF)	7	6	5	4	3	2	1	0
Bit Symbol	SYSCR4							
Read/Write	Write only							
After reset	0	0	0	0	0	0	0	0

SYSCR4	Write the SYSCR3 data control code	0xB2:	Enable the contents of SYSCR3 <RSTDIS>
		0xD4	Enable the contents of SYSCR3 <RAREA> and SYSCR3 <RVCTR>
		0x71:	Enable the contents of IRSTSR <FCLR>
		Others:	Invalid

Note 1): SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2): After SYSCR3 <RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3 <RSTDIS>) in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3 <RSTDIS> may be enabled at unexpected timing.

Note 3): After IRSTSR <FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR <FCLR> in NORMAL mode when fcgck is fc/4 (CGCR <FCGCKSEL>=00). Otherwise, IRSTSR <FCLR> may be enabled at unexpected timing.

System Control Status Register 4

SYSSR4 (0x0FDF)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	RVCTRS	RAREAS	(RSTDISS)
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

RAREAS	Status of mapping of the RAM in the code area	0: The enabled SYSCR3<RAREA> data is "0". 1: The enabled SYSCR3<RAREA> data is "1".
RVCTRS	Status of mapping of the vector address in the area	0: The enabled SYSCR3<RVCTR> data is "0". 1: The enabled SYSCR3<RVCTR> data is "1".

Note 1): The enabled SYSCR3 <RSTDIS> is initialized by a power-on reset only, and cannot be initialized by any other reset signals.

The value written in SYSCR3 is reset by a power-on reset and other reset signals.

Note 2): Bits 7 to 3 of SYSCR4 are read as "0".

Note 3): Example- program transfer(Transfer the program saved in the data area to the RAM)

```
LD      HL,TRANSFER_STAR_ADDRESS     ;Destination RAM address
LD      DE,PROGRAM_START_ADDRESS     ;Source ROM address
LD      BC,BYTE_OF_PROGRAM           ;Number of bytes of the program to be executed-1
LD      A,(DE)                       ;Reading the program to be transferred
LD      (HL),A                       ;Writing the program to be transferred
INC     HL                           ;Destination address increment
INC     DE                           ;Source address increment
DEC     BC                           ;Source address increment
J       F,TRANS_RAM                  ;Have all the programs been transferred?
```

4.2.2.2 BOOTROM

The BOOTROM is not mapped in the code area or the data area after reset release.

Setting FLSCR1<BAREA> to "1" and writing 0xD5 to FLSCR2 maps the BOOTROM to 0x1000 to 0x17FF in the code area and to 0x1000 to 0x17FF in the data area.

Note 1: When the BOOTROM is not mapped in the code area, an instruction is fetched from the Flash or an SWI instruction is fetched, depending on the capacity of the internal Flash.

Note 2: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Flash Memory Control Register 1

FLSCR1 (0x0FD0)	7	6	5	4	3	2	1	0
Bit Symbol	(FLSMD)			BAREA	(FAREA)		(ROMSEL)	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	1	0	0	0	0	0	0

BAREA	Specifies mapping of the BOOTROM in the code and data areas	0: The BOOTROM is not mapped to 0x1000 to 0x17FF in the code area and to 0x1000 to 0x17FF in the data area. 1: The BOOTROM is mapped to 0x1000 to 0x17FF in the code area and to 0x1000 to 0x17FF in the data area.

Note: The flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

Flash Memory Control Register 2

FLSCR2 (0x0FD1)	7	6	5	4	3	2	1	0
Bit Symbol	CR1EN							
Read/Write	W							
After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable/disable control	0xD5: Enable a change in the FLSCR1 setting. Others: Reserved.

4.2.2.3 Flash

The Flash is mapped to 0x8000 to 0xFFFF in the code area after reset release.

4.2.3 Program Memory - Flash

The program memory (Flash) is used to store the program instructions which are to be executed. It also contains data, table, and interrupt entries, and is organized into 8K + 128 bytes format which is addressed by the PC and table pointer. The Flash ranges from 0xC000 to 0xFFFF (16K bytes).

Certain locations in the Flash are reserved for special usage:

Location 0xFFFE is reserved for program initialization. After chip reset, the program always begins execution at this location.

Location 0xFFFC is reserved for the software / undefined instruction interrupt service program. If the software / undefined instruction output pin is activated and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFFC.

Location 0xFFF8 is reserved for the WDT interrupt service program. If the WDT output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFF8.

Location 0xFFF6 is reserved for the wakeup interrupt service program. If the wakeup output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFF6.

Location 0xFFF4 is reserved for the time-base timer (TBT) interrupt service program. If the TBT output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFF4.

Location 0xFFEC is reserved for the voltage detection interrupt service program. If the voltage detection output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFEC.

Location 0xFFEA is reserved for the analog to digital converter (ADC) interrupt service program. If the ADC output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFEA.

Location 0xFFE8 is reserved for the RTC service program. If the RTC output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFE8.

Location 0xFFE6 is reserved for the TC00interrupt service program. If the TC00 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFE6.

Location 0xFFE4 is reserved for theTC01interrupt service program. If the TC01 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFE4.

Location 0xFFE2 is reserved for the TCA0 interrupt service program. If the TCA0 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFE2.

Location 0xFFDE is reserved for the external interrupt 0 service program. If the INT0 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFDE.

Location 0xFFDC is reserved for the external interrupt 1 service program. If the INT1 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFDC..

Location 0xFFD2 is reserved for the UART Receiver 1 (INTRXD1)interrupt service program. If the INTRXD1 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFD2.

Location 0xFFD0 is reserved for the UART Transmitter 1 (INTTXD1) interrupt service program. If the INTTXD1 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFD0.

Location 0xFFCE is reserved for the TC02 interrupt service program. If the TC02 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFCE.
Location 0xFFCC is reserved for the TC03 interrupt service program. If the TC03 output pin is activated, and if the interrupt is enabled and the stack is not full, the program begins execution at location 0xFFCC.

4.2.4 Special Function Register - SFR

The SFR is mapped to 0x0000 to 0x003F (SFR1), 0x0F00 to 0x0FFF (SFR2) and 0x0E40 to 0x0EFF (SFR3) in the data area after reset release.

	SFR1
0x0000	P0DR
0x0001	P1DR
0x0002	P2DR
0x0003	
0x0004	P4DR
0x0005	
0x0006	
0x0007	P7DR
0x0008	P8DR
0x0009	P9DR
0x000A	
0x000B	PBDR
0x000C	
0x000D	P0PRD
0x000E	P1PRD
0x000F	P2PRD
0x0010	
0x0011	P4PRD
0x0012	
0x0013	
0x0014	P7PRD
0x0015	P8PRD
0x0016	P9PRD
0x0017	
0X0018	PBPRD
0x0019	
0x001A	UART0CR1
0x001B	UART0CR2
0x001C	UART0DR
0x001D	UART0SR
0x001E	TD0BUF/RD0BUF
0x001F	SIO0CR
0x0020	SIO0SR
0x0021	SIO0BUF
0x0022	SBI0CR1
0x0023	SBI0CR2/SBI0SR2
0x0024	I2C0AR
0x0025	SBI0DBR
0x0026	T00REG
0x0027	T01REG
0x0028	T00PWM
0x0029	T01PWM
0x002A	T00MOD
0x002B	T01MOD
0x002C	T001CR
0x002D	TA0DRAL
0x002E	TA0DRAH
0x002F	TA0DRBL
0x0030	TA0DRBH
0x0031	TA0MOD
0x0032	TA0CR
0x0033	TA0SR
0x0034	ADCCR1
0x0035	ADCCR2
0x0036	ADCDRL
0x0037	ADCDRH
0x0038	DVOCR
0x0039	TBTCR
0x003A	EIRL
0x003B	EIRH
0x003C	EIRE
0x003D	EIRD
0x003E	
0x003F	PSW

	SFR2
0x0F00	
......	......
0x0F19	
0x0F1A	P0CR
0x0F1B	P1CR
0x0F1C	P2CR
0x0F1D	
0x0F1E	P4CR
0x0F1F	
0x0F20	
0x0F21	P7CR
0x0F22	P8CR
0x0F23	P9CR
0x0F24	
0x0F25	PBCR
0x0F26	
0x0F27	P0PU
0x0F28	P1PU
0x0F29	P2PU
0x0F2A	
0x0F2B	P4PU
0x0F2C	
0x0F2D	
0x0F2E	
0x0F2F	
0x0F30	P9PU
0x0F31	
0x0F32	
0x0F33	
0x0F34	P0FC
0x0F35	
0x0F36	P2FC
0x0F37	
0x0F38	P4FC
0x0F39	
0x0F3A	
0x0F3B	P7FC
0x0F3C	P8FC
0x0F3D	P9FC
0x0F3E	
0x0F3F	PBFC
......	......
0x0F42	
0x0F43	P2OUTCR
0x0F44	
......	......
0x0F49	
0x0F4A	P9OUTCR
0x0F4B	
0x0F4C	PBOUTCR
0x0F4D	
......	......
0x0F53	
0x0F54	UART1CR1
0x0F55	UART1CR2
0x0F56	UART1DR
0x0F57	UART1SR
0x0F58	TD1BUF/RD1BUF
0x0F59	
0x0F5A	UART2CR1
0x0F5B	UART2CR2
0x0F5C	UART2DR
0x0F5D	UART2SR
0x0F5E	TD2BUF/RD2BUF

	SFR2
0x0F5F	
......	......
0x0F73	
0x0F74	POFFCR0
0x0F75	POFFCR1
0x0F76	POFFCR2
0x0F77	POFFCR3
0x0F78	
......	......
0x0F87	
0x0F88	T02REG
0x0F89	T03REG
0x0F8A	T02PWM
0x0F8B	T03PWM
0x0F8C	T02MOD
0x0F8D	T03MOD
0x0F8E	T023CR
0x0F8F	
......	......
0x0FA7	
0x0FA8	TA1DRAL
0x0FA9	TA1DRAH
0x0FAA	TA1DRBL
0x0FAB	TA1DRBH
0x0FAC	TA1MOD
0x0FAD	TA1CR
0x0FAE	TA1SR
0x0FAF	
......	......
0x0FC3	
0x0FC4	KWUCR0
0x0FC5	KWUCR1
0x0FC6	VDCR1
0x0FC7	VDCR2
0x0FC8	RTCCR
0x0FC9	
0x0FCA	ITSEL
0x0FCB	SERSEL
0x0FCC	IRSTSR
0x0FCD	WUCCR
0x0FCE	WUCDR
0x0FCF	CGCR
0x0FD0	FLSCR1
0x0FD1	FLSCR2/FLSCRM
0x0FD2	FLSSTB
0x0FD3	SPCR
0x0FD4	WDCTR
0x0FD5	WDCDR
0x0FD6	WDCNT
0x0FD7	WDST
0x0FD8	EINTCR1
0x0FD9	EINTCR2
0x0FDA	EINTCR3
0x0FDB	EINTCR4
0x0FDC	SYSCR1
0x0FDD	SYSCR2
0x0FDE	SYSCR3
0x0FDF	SYSCR4/SYSSR4
0x0FE0	ILL
0x0FE1	ILH
0x0FE2	ILE
0x0FE3	ILD
0x0FE4	
......	......

	SFR2
0x0FEF	
0x0FF0	ILPRS1
0x0FF1	ILPRS2
0x0FF2	ILPRS3
0x0FF3	ILPRS4
0x0FF4	ILPRS5
0x0FF5	ILPRS6
0x0FF6	ILPRS7
0x0FF7	
......	......
0x0FFF	

	SFR3
0x0E40	
......	......
0x0E56	
0x0E57	UATCNG
0x0E58	
......	......
0x0E97	
0x0E98	TC0CR1
0x0E99	TC0CR2
0x0E9A	TC0CR3
0x0E9B	TC0DRAL
0x0E9C	TC0DRAH
0x0E9D	TC0DRBL
0x0E9E	TC0DRBH
0x0E9F	TC0DRCL
0x0EA0	TC0DRCH
0x0EA1	TC0DRDL
0x0EA2	TC0DRDH
0x0EA3	TC0DREL
0x0EA4	TC0DREH
0x0EA5	TC0CAPAL
0x0EA6	TC0CAPAH
0x0EA7	TC0CAPBL
0x0EA8	TC0CAPBH
0x0EA9	
......	......
0x0EFF	

Note : Don't access the reserved SFR.

Figure 4.3 SFR1, SFR2, and SFR3

4.3 Operation Modes

4.3.1 Operation Mode Control Circuit

The operation mode control circuit starts and stops the oscillation circuits for the high-frequency and low-frequency clocks, and switches the main system clock (fm).There are three operating modes: the single-clock mode, the dual-clock mode and the STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2).

4.3.1.1 Single-clock Mode

Only the gear clock (fcgck) is used for the operation in the single-clock mode. The main system clock (fm)is generated from the gear clock (fcgck). Therefore, the machine cycle is 1/fcgck [s]. The gear clock (fcgck) is generated from the high-frequency clock (fh).

The high-frequency reference clock (fh) can be selected from the external high-frequency clock (fc) and the internal high-frequency clock (fosc). When the internal high-frequency clock (fosc) is used as the high-frequency reference clock (fh), pins P00 (XIN) and P01 (XOUT) of the external high-frequency clock oscillation circuit can be used as general purpose I/O ports.

Before switching the operating mode, be sure to select either the external high-frequency clock (fc) or the internal high-frequency clock (fosc) and then stop either of the high-frequency clocks not to be used. If a mode transition is made with both the external and internal high-frequency clocks enabled, the transition may not be performed properly. In the single-clock mode, pins P02 (XTIN) and P03 (XTOUT) of the external low-frequency clock ocillation circuit can be used as general-purpose I/O ports.

(a) NORMAL1 Mode

In this mode, the CPU core and the peripheral circuits operate using the gear clock (fcgck). The NORMAL1 mode becomes active after reset release.

(b) IDLE1 Mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck).

The IDLE1 mode is activated by setting SYSCR2 <IDLE> to "1" in the NORMAL1 mode. When the IDLE1 mode is activated, the CPU and the watchdog timer stop. When the interrupt latch enabled by the interrupt enable register EIR becomes "1", the IDLE1 mode is released to the NORMAL1 mode.

When the IMF (interrupt master enable flag) is "1" (interrupts enabled), the operation returns normal after the interrupt processing is completed. When the IMF is "0" (interrupts disabled),

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

the operation is restarted by the instruction that follows the IDLE1 mode activation instruction.

(c) IDLE0 Mode

In this mode, the CPU and the peripheral circuits stop, except the oscillation circuits and the time base timer.

In the IDLE0 mode, the peripheral circuits stop in the states when the IDLE0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the IDLE0 mode, refer to the section of each peripheral circuit.

The IDLE0 mode is activated by setting SYSCR2 <TGHALT> to "1" in the NORMAL1 mode. When the IDLE0 mode is activated, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer. When the falling edge of the source clock selected at TBTCR <TBTCK> is detected, the IDLE0 mode is released, the timing generator starts the clock supply to all the peripheral circuits and the NORMAL1 mode is restored. Note that the IDLE0 mode is activated and restarted, regardless of the setting of TBTCR <TBTEN>.

When the IDLE0 mode is activated with TBTCR <TBTEN> set at "1", the INTTBT interrupt latch is set after the NORMAL mode is restored. When the IMF is "1" and the EF5 (the individual interrupt enable flag for the time base timer)is "1", the operation returns normal after the interrupt processing is completed. When the IMF is "0" or when the IMF is "1" and the EF5 (the individual interrupt enable flag for the time base timer) is "0", the operation is restarted by the instruction that follows the IDLE0 mode activation instruction.

4.3.1.2 Dual-clock Mode

The gear clock (fcgck) and the low-frequency clock (fs) are used for the operation in the dual-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck) in the NORMAL2 or IDLE2 mode, and generated from the clock that is a quarter of the low-frequency clock (fs) in the SLOW1/2 or SLEEP0/1 mode. Therefore, the machine cycle time is 1/fcgck [s] in the NORMAL2 or IDLE2 mode and is 4/fs [s] in the SLOW1/2 or SLEEP0/1 mode.

Pins P02 (XTIN) and P03 (XTOUT) are used for the low-frequency clock oscillation circuit. (These pins cannot be used as I/O ports in the dual-clock mode.) The gear clock (fcgck) is generated from the high-frequency reference clock (fh). The high-frequency reference clock (fh) can be selected from the external high-frequency clock (fc) and the internal high-frequency clock (fosc). When the internal high-frequency clock (fosc) is used as the high-frequency reference clock (fh), pins P00 (XIN) and P01(XOUT) of the external high-frequency clock oscillation circuit can be used as general purpose I/O ports.

Before switching the operating mode, be sure to select either the external high-frequency cock (fc) or the internal high-frequency clock (fosc) and then stop either of the high-frequency clocks not to

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

be used. If a mode transition is made with both the external and internal high-frequency clocks enabled, the transition may not be performed properly. For how to switch the high-frequency reference clock (fh), refer to "(1) High-frequency reference clock (fh)".

SYSCR1<OSCSEL> cannot be changed when SYSCR1<SYSCK> is "1". Therefore, when switching between the SLOW1 and SLOW2 modes, the high-frequency reference clock (fh) must be set in advance in the NORMAL1 or NORMAL2 mode.

The operation of the MCU core becomes the single-clock mode after reset release. To operate it in the dual-clock mode, allow the low-frequency clock to oscillate at the beginning of the program.

(a) NORMAL2 Mode

In NORMAL2 mode, the CPU core operates using the gear clock (fcgck), and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

(b) SLOW2 Mode

In SLOW2 mode, the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2 <SYSCK> to switch the operation mode from NORMAL2 to SLOW2 or from SLOW2 to NORMAL2.In the SLOW2 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(c) SLOW1 Mode

In SLOW1 mode, the high-frequency clock oscillation circuit stops operation and the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

This mode requires less power to operate the high-frequency clock oscillation circuit than in the SLOW2 mode.

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2 <XEN> to switch the operation between the SLOW1 and SLOW2 modes. In the SLOW1 or SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

(d) IDLE2 Mode

In IDLE2 mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

The IDLE2 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the NORMAL2 mode after this mode is released.

(e) SLEEP1 Mode

In SLEEP1 mode, the high-frequency clock oscillation circuit stops operation, the CPU and the watchdog timer stop, and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLEEP1 mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP1 mode, refer to the section of each peripheral circuit. The SLEEP1 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLOW1 or SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(f) SLEEP0 Mode

In SLEEP0 mode, the high-frequency clock oscillation circuit stops operation, the time base timer operates using the clock that is a quarter of the low-frequency clock (fs), and the core and the peripheral circuits stop.

In the SLEEP0 mode, the peripheral circuits stop in the states when the SLEEP0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP0 mode, refer to the section of each peripheral circuit.
The SLEEP0 mode can be activated and released in the same way as for the IDLE0 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLEEP0 mode, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

4.3.1.3 STOP Mode

In STOP mode, all the operations in the system including the oscillation circuits are stopped and the internal states in effect before the system was stopped are held with low power consumption.

In the STOP mode, the peripheral circuits stop in the states when the STOP mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the STOP mode, refer to the section of each peripheral circuit. The STOP mode is activated by setting SYSCR1 <STOP> to "1".

The STOP mode is released by the STOP mode release signals. After the warm-up time has elapsed, the operation returns to the mode that was active before the STOP mode, and the operation is restarted by the instruction that follows the STOP mode activation instruction.

4.3.1.4 Transition of Operation Modes

Operation mode		Oscillation circuit		CPU core	Watchdog timer	Time base timer	AD converter	Other peripheral circuits	Machine cycle time
		High frequency reference clock (fh)	Low-frequency clock (fs)						
Single clock	RESET	Oscillation	Stop	Reset	Reset	Reset	Reset	Reset	1 / fcgck [s]
	NORMAL1			Operate	Operate	Operate	Operate	Operate	
	IDLE1			Stop	Stop				
	IDLE0						Stop	Stop	
	STOP	Stop				Stop			-
Dual clock	NORMAL2	Oscillation	Oscillation	Operate with the high frequency	Operate with the high frequency	Operate	Operate	Operate	1 / fcgck [s]
	IDLE2			Stop	Stop				
	SLOW2			Operate with the low frequency	Operate with the low frequency		Stop		4/ fs [s]
	SLOW1	Stop		Operate with the low frequency	Operate with the low frequency				
	SLEEP1			Stop	Stop				
	SLEEP0			Stop	Stop			Stop	
	STOP		Stop	Stop	Stop	Stop			-

Table 4.1 Operation Modes and Conditions

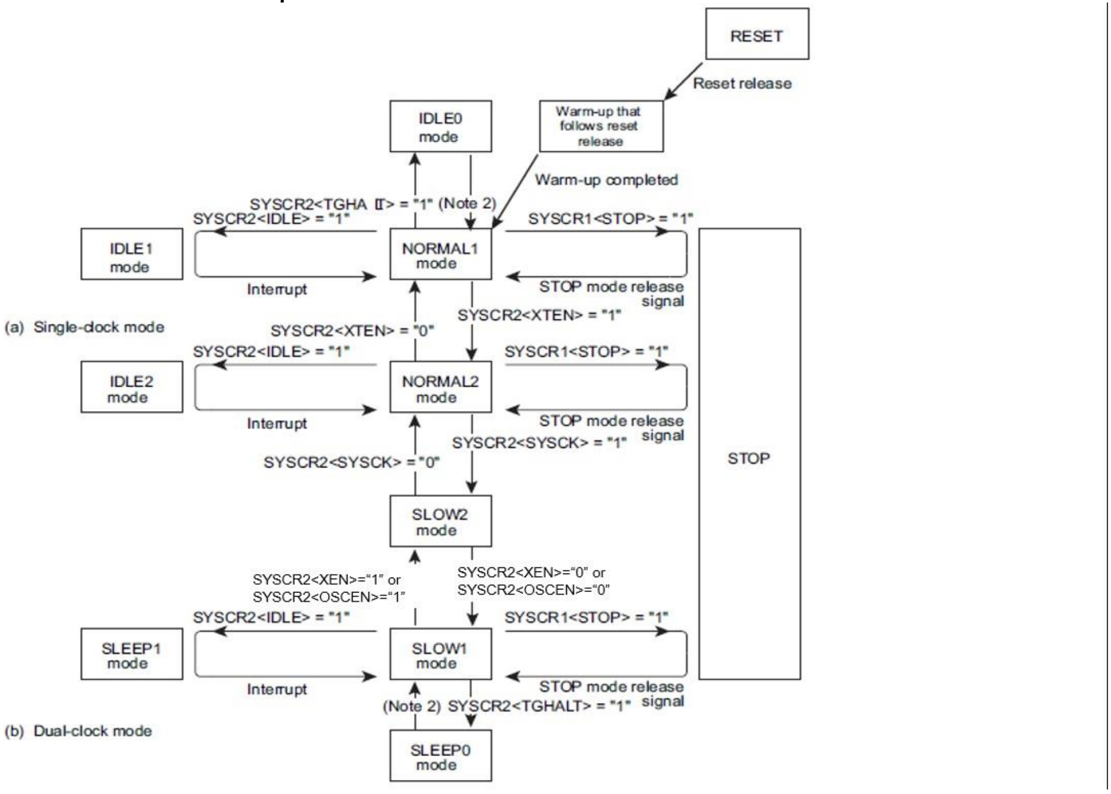


Figure 4.4 Operation Mode Transition Diagram

Note 1): The NORMAL1 and NORMAL2 modes are generically called the NORMAL mode; the SLOW1 and SLOW2 modes

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

are called the SLOW mode; the IDLE0, IDLE1 and IDLE2 modes are called the IDLE mode; and the SLEEP0 and SLEEP1 are called the SLEEP mode.

Note 2): The mode is released by the falling edge of the source clock selected at TBTCR <TBTCK>.

4.3.2 Operation Mode Control

4.3.2.1 STOP Mode

The STOP mode is controlled by system control register 1 (SYSCR1) and the STOP mode release signals.

(a) Start the STOP Mode

The STOP mode is started by setting SYSCR1<STOP> to "1". In the STOP mode, the following states are maintained:

1. Both the high-frequency and low-frequency clock oscillation circuits stop oscillation and all internal operations are stopped.
2. The data memory, the registers and the program status word are all held in the states in effect before STOP mode was started. The port output latch is determined by the value of SYSCR1 <OUTEN>.
3. The prescaler and the divider of the timing generator are cleared to "0".
4. The program counter holds the address of the instruction 2 ahead of the instruction (e.g., [SET (SYSCR1).7]) which started the STOP mode.

(b) Release the STOP Mode

The STOP mode is released by the following STOP mode release signals. It is also released by a reset by the RESETB pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

1. Release by the STOPB pin
2. Release by key-on wakeup
3. Release by the voltage detection circuits

Note): During the STOP period (from the start of the STOP mode to the end of the warm-up), due to changes in the external interrupt pin signal, interrupt latches may be set to "1" and interrupts may be accepted immediately after the STOP mode is released. Before starting the STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

1. Release by the STOPB pin

Release the STOP mode by using the STOPB pin.

The STOP mode release by the STOP pin includes the level-sensitive release mode and the edge-sensitive release mode, either of which can be selected at SYSCR1<RELM>.

The STOPB pin is also used as the P11 and the INT5B pin.

- **Level-sensitive release mode**

The STOPB mode is released by setting the STOPB pin high.

Setting SYSCR1 <RELM> to "1" selects the level-sensitive release mode.

This mode is used for the capacitor backup when the main power supply is cut off and the long term battery backup.

Even if an instruction for starting the STOP mode is executed while the STOPB pin input is high, the STOP mode does not start. Thus, to start the STOP mode in the level- release mode, it is necessary for the program to first confirm that the STOPB pin input is low. This can be confirmed by testing the port by the software or using interrupt.

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Example:
Starting the STOP mode from NORMAL mode after testing P00 port.
(Warm-up time at release of the STOP mode is about 300μs at fc= 10MHz.)

```
LD       (SYSCR1), 0x40           ;Sets up the level-sensitive release mode
TEST     (P0PRD).5                ;Wait until STOP pin becomes L level.
J        F,SSTOPH
LD       (WUCCR),0x01             ;WUCCR<WUCDIV> = 00 (No division) (Note)
         (WUCDR),0x2F             ;Sets the warm-up time
                                  ;300μs / 6.4μs = 46.9 → round up to 0x2F
DI                                ;IMF = 0
SET      (SYSCR1).7               ;Starts the STOP mode
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Example:
Starting the STOP mode from the SLOW mode with an INT5 interrupt
(Warm-up time at release of the STOP mode is about 450ms at fs=32.768 kHz.)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

```
                                           ;To reject noise, the STOP mode does not start
TEST      (P0PRD).5                        ;if the STOP pin input is high.
J         F,SINT5                          ;Sets up the level-sensitive release mode
LD        (SYSCR1),0x40                    ;WUCCR<WUCDIV> = 00 (No division)
LD        (WUCCR),0x03                     ;Sets the warm-up time
LD        (WUCDR),0xE8                     450 ms/1.953 ms = 230.4 → round up to 0xE8
DI                                         ;IMF = 0
SET       (SYSCR1).7                       ;Starts the STOP mode
RETI
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

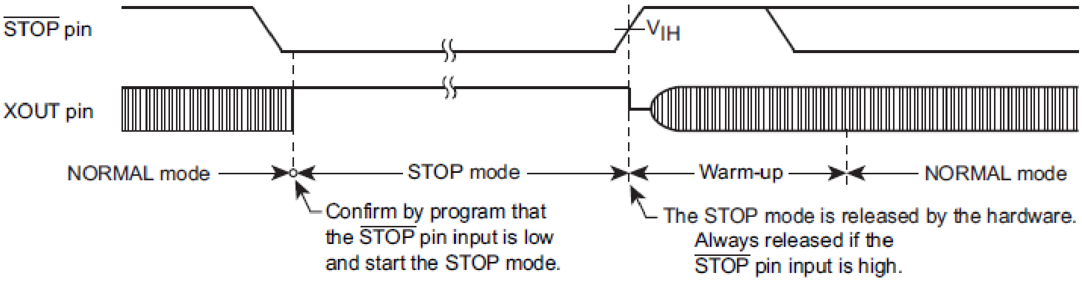

STOP pin
XOUT pin
V_{IH}
NORMAL mode
STOP mode
Warm-up
NORMAL mode
Confirm by program that the STOP pin input is low and start the STOP mode.
The STOP mode is released by the hardware. Always released if the STOP pin input is high.

Figure 4.5 Level-sensitive Release Mode (Example when the high-frequency clock oscillation circuit is selected)

- **Edge-sensitive release mode**

In this mode, the STOP mode is released at the rising edge of the STOP pin input. Setting SYSCR1 <RELM> to "0" selects the edge-sensitive release mode.

This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (such as a clock from a low-power consumption oscillator) is input to the STOPB pin. In the edge-sensitive release mode, the STOP mode is started even when the STOPB pin input is high.

Example:
Starting the STOP mode from the NORMAL mode
(Warm-up time at release of the STOP mode is about 200μs at fc=10 MHz.)

```
LD        (WUCCR),0x01       ;WUCCR<WUCDIV> = 00 (No division)
LD        (WUCDR),0x20       ;Sets the warm-up time
                             ;200μs / 6.4μs = 31.25 → round up to 0x20
DI                           ;IMF = 0
```

```
        LD      (SYSCR1),0x80     ;Starts the STOP mode with the edge-sensitive release ;
                                  mode selected
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>

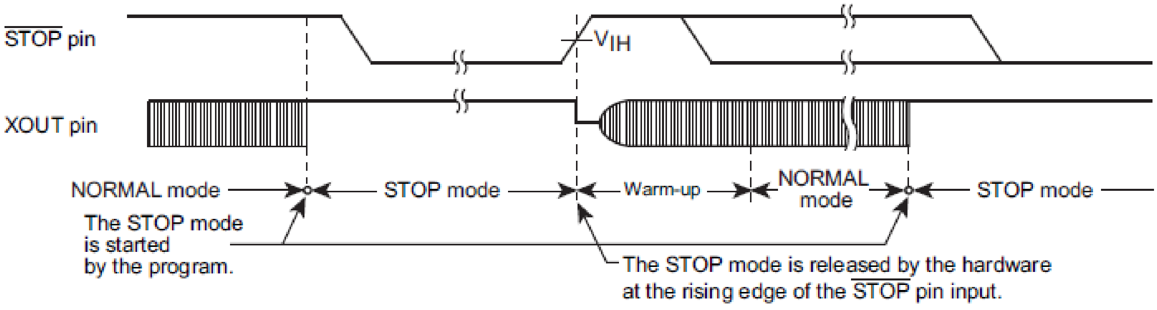

STOP pin
XOUT pin
VIH
NORMAL mode
STOP mode
Warm-up
NORMAL mode
STOP mode
The STOP mode is started by the program.
The STOP mode is released by the hardware at the rising edge of the STOP pin input.

Figure 4.6 Edge-sensitive Release Mode (Example when the high-frequency clock oscillation circuit is selected)

Note: If the rising edge is input to the STOP pin within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

2. Release by the Key-on Wakeup

The STOP mode is released by inputting the prescribed level to the key-on wakeup pin. The level to release the STOP mode can be selected from "H" and "L"

Note): If the key-on wakeup pin input becomes the opposite level to the release level after the warm-up starts, the STOP mode is not restarted.

3. Release by the Voltage Detection Circuits

The STOP mode is released by the supply voltage detection by the voltage detection circuits. To release the STOP mode by using the voltage detection circuits, set VDCR2 <VDSS> to "01" or "10". If the voltage detection operation mode of the voltage detection circuits is set to generate reset signals (when VDCR2 <VDxMOD> is 1 (x=1 to 2)), the STOP mode is re- leased and a reset is applied as soon as the supply voltage becomes lower than the detection voltage.

When the supply voltage becomes equal to or higher than the detection voltage of the voltage detection circuits, the reset is released and the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

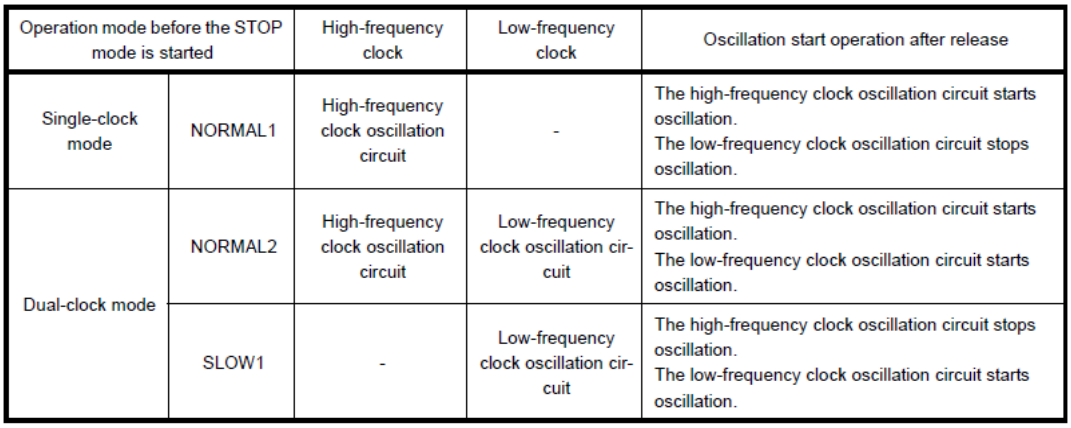
If the voltage detection operation mode of the voltage detection circuits is set to

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

generate interrupt request signals (when VDCR2 <VDxMOD> is 0 (x=1 to 2)), the STOP mode is released when the supply voltage becomes equal to or higher than the detection voltage. For details, refer to the section of the voltage detection circuits.

Note): If the supply voltage becomes equal to or higher than the detection voltage within 1 machine cycle after SYSCR1 <STOP> is set to "1", the STOP mode will not be released.

(c) STOP Mode Release Operation



Operation mode before the STOP mode is started		High-frequency clock	Low-frequency clock	Oscillation start operation after release
Single-clock mode	NORMAL1	High-frequency clock oscillation circuit	-	The high-frequency clock oscillation circuit starts oscillation. The low-frequency clock oscillation circuit stops oscillation.
Dual-clock mode	NORMAL2	High-frequency clock oscillation circuit	Low-frequency clock oscillation circuit	The high-frequency clock oscillation circuit starts oscillation. The low-frequency clock oscillation circuit starts oscillation.
	SLOW1	-	Low-frequency clock oscillation circuit	The high-frequency clock oscillation circuit stops oscillation. The low-frequency clock oscillation circuit starts oscillation.

Table 4.2 Oscillation Start Operation at Release of the STOP Mode

Note): When the operation returns to the NORMAL2 mode, fc is input to the frequency division circuit of the warm-up counter.

The STOP mode is released in the following sequence:

1. Oscillation starts. For the oscillation start operation in each mode, refer to "Table 4.2 Oscillation Start Operation at Release of the STOP Mode".
2. Warm-up is executed to secure the time required to stabilize oscillation. The internal operations remain stopped during warm-up. The warm-up time is set by the warm-up counter, depending on the oscillator characteristics.
3. After the warm-up time has elapsed, the normal operation is restarted by the instruction that follows the STOP mode start instruction. At this time, the prescaler and the divider of the timing generator are cleared to "0".

Note): When the STOP mode is released with a low hold voltage, the following cautions must be observed. The supply voltage must be at the operating voltage level before releasing the STOP mode. The RESETB pin input must also be "H" level, rising together with the supply voltage. In this case, if an external time constant circuit has been connected, the RESETB pin input voltage will increase at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur if the input voltage level of the RESETB pin drops below the non-inverting high-level input voltage (Hysteresis input).

4.3.2.2 IDLE1/2 and SLEEP1 Modes

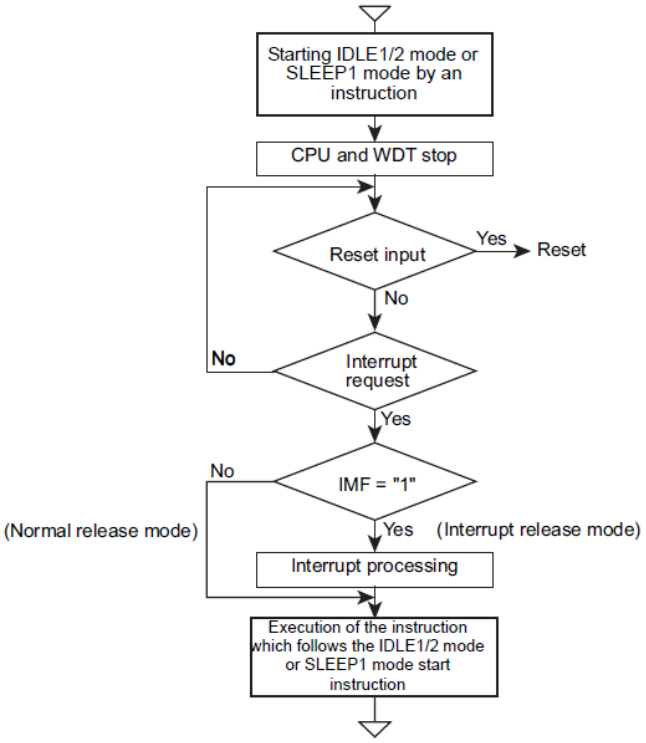

Starting IDLE1/2 mode or SLEEP1 mode by an instruction
CPU and WDT stop
Reset input
Yes
Reset
No
No
Interrupt request
Yes
No
IMF = "1"
(Normal release mode)
Yes
(Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE1/2 mode or SLEEP1 mode start instruction

Figure 4.7 IDLE1/2 and SLEEP1 Modes

The IDLE1/2 and SLEEP1 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following states are maintained during these modes.

1. The CPU and the watchdog timer stop their operations. The peripheral circuits continue to operate.
2. The data memory, the registers, the program status word and the port output latches are all held in the status in effect before IDLE1/2 or SLEEP1 mode was started.
3. The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE1/2 or SLEEP1 mode.

(a) Start the IDLE1/2 and SLEEP1 Modes

After the interrupt master enable flag (IMF) is set to "0", set the individual interrupt enable flag (EF) to "1", which releases IDLE1/2 and SLEEP1 modes. To start the IDLE1/2 or SLEEP1 mode, set SYSCR2 <IDLE> to "1".If the release condition is satisfied when it is attempted to start the IDLE1/2 or SLEEP1 mode, SYSCR2 <IDLE> remains cleared and the IDLE1/2 or SLEEP1 mode will not be started.

Note 1): When a watchdog timer interrupt is generated immediately before the IDLE1/2 or SLEEP1 mode is started, the watchdog timer interrupt will be processed but the IDLE1/2 or SLEEP1 mode will not be started.

Note 2): Before starting the IDLE1/2 or SLEEP1 mode, enable the interrupt request signals to be generated to release

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

the IDLE1/2 or SLEEP1 mode and set the individual interrupt enable flag.

(b) Release the IDLE1/2 and SLEEP1 Modes

The IDLE1/2 and SLEEP1 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (IMF). After releasing IDLE1/2 or SLEEP1 mode, SYSCR2 <IDLE> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE1/2 or SLEEP1 mode.

The IDLE1/2 and SLEEP1 modes are also released by a reset by the RESETB pin, a power-on reset and a reset by the voltage detection circuits. After releasing the reset, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

1. **Normal release mode (IMF = "0")**

   The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (EF) is "1". The operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction. Normally, the interrupt latch (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

2. **Interrupt release mode (IMF = "1")**

   The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (EF) is "1". After the interrupt is processed, the operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction.

4.3.2.3 IDLE0 and SLEEP0 Modes

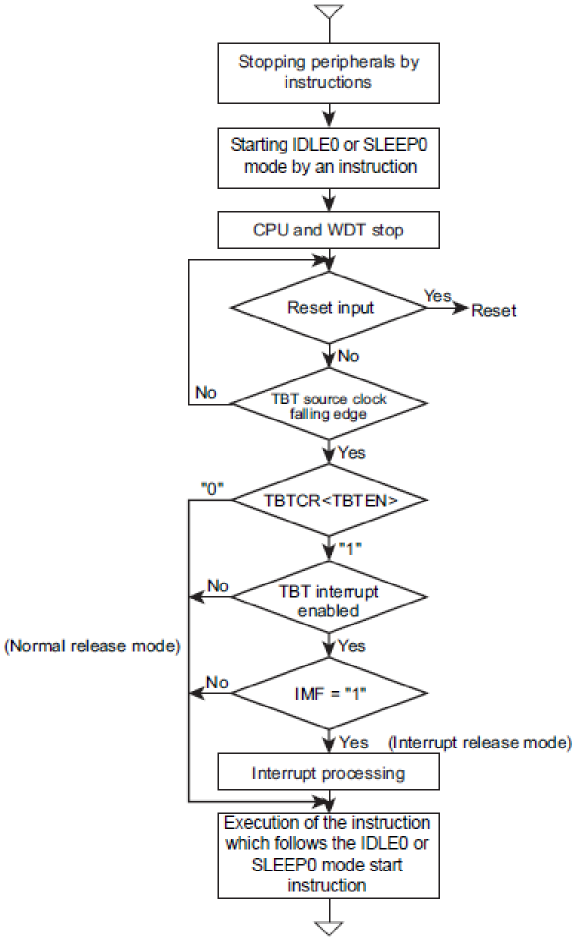

Stopping peripherals by instructions
Starting IDLE0 or SLEEP0 mode by an instruction
CPU and WDT stop
Reset input
Yes
Reset
No
No
TBT source clock falling edge
Yes
"0"
TBTCR<TBTEN>
"1"
No
TBT interrupt enabled
(Normal release mode)
Yes
No
IMF = "1"
Yes
(Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE0 or SLEEP0 mode start instruction

Figure 4.8 IDLE0 and SLEEP0 Modes

The IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCR). The following states are maintained during the IDLE0 and SLEEP0 modes:

1. The timing generator stops the clock supply to the peripheral circuits except the time base timer.
2. The data memory, the registers, the program status word and the port output latches are all held in the states in effect before the IDLE0 or SLEEP0 mode was started.
3. The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE0 or SLEEP0 mode.

(a) Start the IDLE0 and SLEEP0 Modes

Stop (disable) the peripherals such as a timer counter. To start the IDLE0 or SLEEP0 mode, set SYSCR2 <TGHALT> to "1".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

(b) Release the IDLE0 and SLEEP0 Modes

The IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (IMF), the individual interrupt enable flag (EF5) for the time base timer and TBTCR <TBTEN>. After releasing the IDLE0or SLEEP0 mode, SYSCR2 <TGHALT> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE0 or SLEEP0 mode. If TBTCR <TBTEN>has been set at "1", the INTTBT interrupt latch is set.

The IDLE0 and SLEEP0 modes are also released by a reset by the RESETB pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

1. **Normal Release Mode (IMF, EF5, TBTCR<TBTEN> = "0")**

   The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR <TBTCK> is detected. After the IDLE0 or SLEEP0 mode is released, the operation is restarted by the instruction that follows the IDLE0 or SLEEP0 mode start instruction.

   When TBTCR <TBTEN> is "1", the time base timer interrupt latch is set.

2. **Interrupt Release Mode (IMF, EF5, TBTCR<TBTEN> = "1")**

   The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR <TBTCK> is detected. After the release, the INTTBT interrupt processing is started.

Note 1): The IDLE0 or SLEEP0 mode is released to the NORMAL1 or SLOW1 mode by the asynchronous internal clock selected at TBTCR <TBTCK>. Therefore, the period from the start to the release of the mode may be shorter than the time specified at TBTCR <TBTCK>.
Note 2): When a watchdog timer interrupt is generated immediately before the IDLE0 or SLEEP0 mode is started, the watchdog timer interrupt will be processed but the IDLE0 or SLEEP0 mode will not be started.

4.3.2.4 SLOW Mode

The SLOW mode is controlled by system control register 2 (SYSCR2).

(a) Switching from the NORMAL2 Mode to the SLOW1 Mode

Set SYSCR2 <SYSCK> to "1".

When a maximum of 2/fcgck + 10/fs [s] has elapsed since SYSCR2 <SYSCK> is set to "1", the main system clock (fm) is switched to fs/4. After switching, wait for 2 machine cycles or longer, and then clear SYSCR2 <XEN> to "0" to turn off the high-frequency clock oscillator. If the oscillation of the low-frequency clock (fs) is unstable, confirm the stable oscillation at the warm-up counter before implementing the procedure described above.

Note 1: Be sure to follow this procedure to switch the operation from the NORMAL2 mode to the SLOW1 mode.

Note 2: It is also possible to allow the basic clock for the high-frequency clock to oscillate continuously to return to NORMAL2 mode. However, be sure to turn off the oscillation of the basic clock for the high-frequency clock when the STOP mode is started from the SLOW mode.

Note 3: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 4: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the basic clock (fs) for the low-frequency clock. For the synchronization, fm is stopped for a period of 10/fs or shorter.

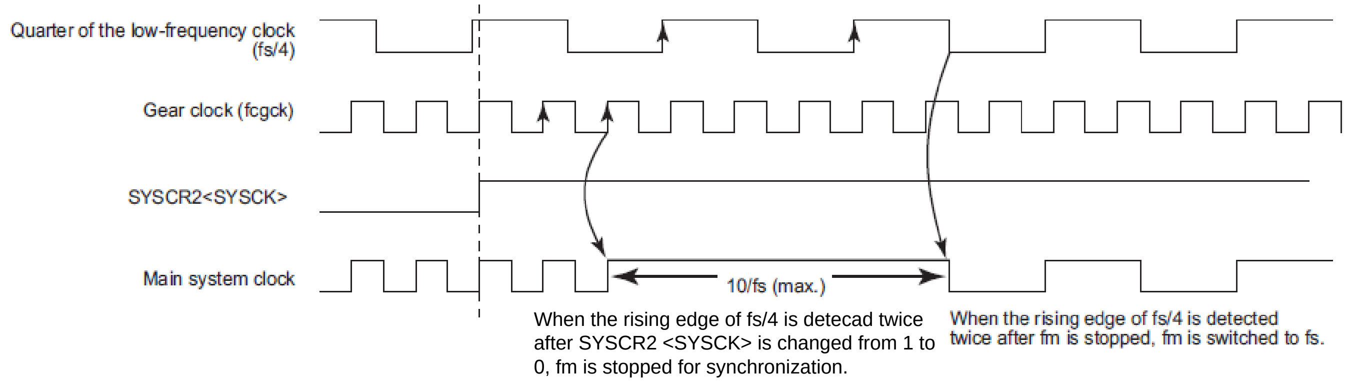

Quarter of the low-frequency clock
(fs/4)
Gear clock (fcgck)
SYSCR2<SYSCK>
Main system clock
10/fs (max.)
When the rising edge of fs/4 is detecad twice after SYSCR2 <SYSCK> is changed from 1 to 0, fm is stopped for synchronization.
When the rising edge of fs/4 is detected twice after fm is stopped, fm is switched to fs.

Figure 4.9 Switching of the Main system clock (fm)
(Switching from fcgck to fs/4)

Example: Switching from the NORMAL2 mode to the SLOW1 mode (when fc is used as the basic clock for the high-frequency clock)

```
SET     (SYSCR2).4      ;SYSCR2<SYSCK> = 1
                        ;(Switches the main system clock to the basic clock for
                        the
                        ;low-frequency clock for the SLOW2 mode)
NOP                     ;Waits for 2 machine cycles
NOP
CLR     (SYSCR2).6      ;SYSCR2<XEN> = 0
                        ;(Turns  off  the  high-frequency  clock  oscillation
                         circuit)
```

Example: While operating with the external high-frequency clock, switching to the SLOW1 mode after the stable oscillation of the external low-frequency clock oscillation circuit is confirmed at the warm-up counter (fs=32.768kHz, warm-up time = about 100 ms)

```
; #### Initialize routine ####
```

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

```
SET       (P0FC).2          ;P0FC2 = 1 (Uses P02/03 as oscillators)
¦
¦
LD        (WUCCR),0x02      ;WUCCR<WUCDIV> = 00 (No division)
                            ;WUCCR<WUCSEL> = 1 (Selects fs as the source clock)
LD        (WUCDR),0x33      ;Sets the warm-up time
                            ;(Determines the time depending on the oscillator
                            ; characteristics) 100 ms/1.95 ms = 51.2 → round up to
                            ;0x33
SET          (EIRL).4       ;Enables INTWUC interrupts
SET         (SYSCR2).5      ;SYSCR2<XTEN> = 1
                            ;(Starts the low-frequency clock oscillation and starts
                            the
                            ;warm-up counter)

¦
; #### Interrupt service routine of warm-up counter interrupts ####
SET       (SYSCR2).4        ;SYSCR2<SYSCK> = 1
                            ; (Switches the main system clock to the low-frequency
                            ; clock)
NOP                         ;Waits for 2 machine cycles
NOP
CLR       (SYSCR2).6        ;SYSCR2<XEN> = 0
                            ;(Turns  off  the  high-frequency  clock  oscillation
                             circuit)
RETI
¦
DW        PINTWUC           ;INTWUC vector table
```

(b) Switching from the SLOW1 Mode to the NORMAL1 Mode

Set SYSCR2 <XEN> to "1" to enable the high-frequency clock (fc) to oscillate. Confirm at the warm-up counter that the oscillation of the basic clock for the high-frequency clock has stabilized, and then clear SYSCR2 <SYSCK> to "0".

When a maximum of 8/fs + 2.5/fcgck [s] has elapsed since SYSCR2 <SYSCK> is cleared to"0", the main system clock (fm) is switched to fcgck. After switching, wait for 2 machine cycles or longer, and then clear SYSCR2 <XTEN> to "0" to turn off the low-frequency clock oscillator.

The SLOW mode is also released by a reset by the RESETB pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

Note 1): Be sure to follow this procedure to switch the operation from the SLOW1 mode to the NORMAL1 mode.

Note 2): After switching SYSCR2 <SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2

<XTEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 3):When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the basic clock (fs) for the low-frequency clock. For the synchronization, fm is stopped for a period of 2.5/fcgck [s] or shorter.

Note 4): When P0FC0 is "0", setting SYSCR2 <XEN> to "1" causes a system clock reset.

Note 5): When SYSCR2 <XEN> is set at "1", writing "1" to SYSCR2 <XEN> does not cause the warm-up counter to start counting the source clock.

Note 6):SYSCR1<OSCSEL> should be set while SYSCR2<SYSCK> is "0" (during the NORMAL1 or NORMAL 2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> is "1" (during the SLOW1 or SLOW2 mode) has no effect.

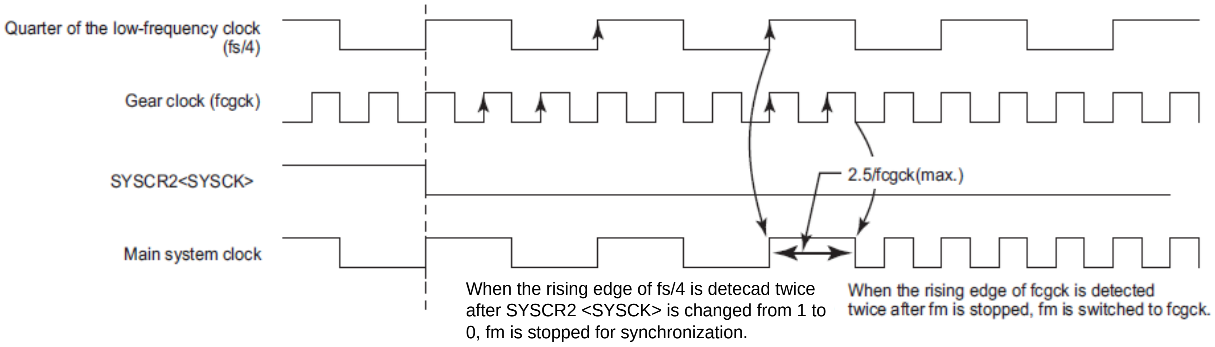

Quarter of the low-frequency clock (fs/4)
Gear clock (fcgck)
SYSCR2<SYSCK>
Main system clock
2.5/fcgck(max.)
When the rising edge of fs/4 is detecad twice after SYSCR2 <SYSCK> is changed from 1 to 0, fm is stopped for synchronization.
When the rising edge of fcgck is detected twice after fm is stopped, fm is switched to fcgck.

**Figure 4.9 Switching of the Main system clock (fm)
(Switching from fs/4 to fcgck)**

Example : Switching from the SLOW1 mode to the NORMAL1 mode after the stability of the external high-frequency clock oscillation circuit is confirmed at the warm-up counter (fc = 10 MHz, warm-up time = 4.0 ms)

```
; #### Initialize routine ####
SET       (P0FC).2           ;P0FC2 = 1 (Uses P02/03 as oscillators)
¦
¦
LD        (WUCCR),0x09       ;WUCCR<WUCDIV> = 10 (Divided by 2)
                             ;WUCCR<WUCSEL> = 0 (Selects fc as the source clock)
LD        (WUCDR),0x9D       ;Sets the warm-up time
                             ;(Determine the time depending on the frequency
                             ; and the oscillator characteristics)
                             ; 4ms / 25.6us = 156.25 → round up to 0x9D
SET       (EIRL).4           ;Enables INTWUC interrupts
SET       (SYSCR2).5         ;SYSCR2<XEN> = 1
                             ;(Starts the oscillation of the high-frequency clock
                             oscillation
                             ; circuit)

¦
; #### Interrupt service routine of warm-up counter interrupts ####
CLR       (SYSCR2).4         ;SYSCR2<SYSCK> = 0
```

```
                                    ;(Switches the main system clock to the gear clock)
          NOP                       ;Waits for 2 machine cycles
          NOP
          NOP
          CLR       (SYSCR2).5      ;SYSCR2<XEN> = 0
                                    ;(Turns  off  the  external  low-frequency  clock
                                     oscillation
                                    ; circuit)
          RETI
          ¦
          DW        PINTWUC         ;INTWUC vector table
```

4.4 Stack Area and Stack Pointer

4.4.1 Stack Area

A stack is an area in memory for temporarily saving the PC, PSW and other values during subroutines and interrupts.

When a subroutine is called by the [CALL mn] or [CALLV n] instruction, the CPU pushes (saves) the high-order and low-order bytes of the return address on the stack before jumping to the subroutine entry address. When the software interrupt instruction, SWI, is executed and when a hardware interrupt is accepted, the CPU saves the PSW and then return address on the stack.

When the return-from-subroutine instruction, RET, is executed, the CPU pops (restores) the return address into the PC. When the return-from-interrupt instruction, RETI or RETN, is executed, the CPU restores the PC and PSW from the stack.

A stack can be allocated anywhere in the data area.

4.4.2 Stack Pointer

The Stack Pointer (SP) is a 16-bit register that holds the address of the next available location on the stack. The SP is post-decremented on subroutine calls, PUSH operations and interrupts, and pre-incremented on returns from subroutines and interrupts and POP operations. The stack grows downwards from high addresses to low addresses as it is filled.

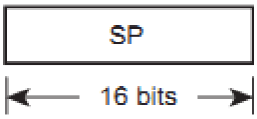


Figure 4.10 Stack Pointer

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Figure 4.10 shows the contents of the stack and the SP register as each of the following instructions is executed.

The SP register defaults to 0x00FF upon hardware reset.

Like an index register, the SP register can be modified by using load / store and ALU instructions. The SP register can also be used as an index register in Indexed Addressing.

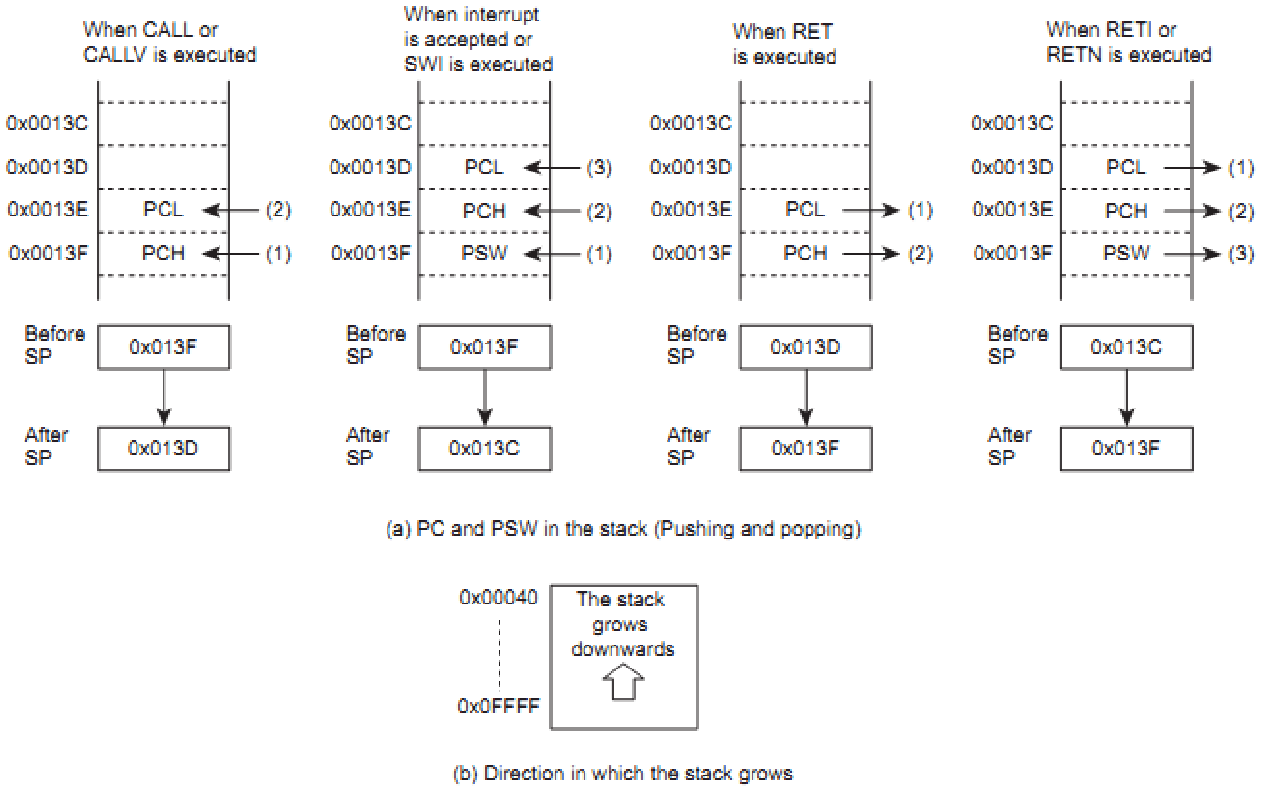

When CALL or CALLV is executed
When interrupt is accepted or SWI is executed
When RET is executed
When RETI or RETN is executed
0x0013C
0x0013D
0x0013E
0x0013F
PCL
PCH
PSW
(1)
(2)
(3)
Before SP
After SP
0x013F
0x013D
0x013C
0x013F
0x013D
0x013F
0x013C
0x013F
(a) PC and PSW in the stack (Pushing and popping)
0x00040
0x0FFFF
The stack grows downwards
(b) Direction in which the stack grows

Figure 4.11 Stack

4.5 Program Counter (PC)

4.5.1 Program Counter - PC

The Program Counter (PC) is an 8-bit register that holds the address of next instruction to be executed in the code area. When the reset signal is released, the CPU loads the reset vector stored in the vector table (at 0xFFFF and 0xFFFE in MCU mode) into the PC; thus the program can start at an arbitrary address. The iMQ i87 Series is pipelined; that is, CPU instructions are pre-fetched. Therefore, the PC points to an address two bytes after the address of the instruction being executed. For example, the PC contains 0xC125 while the single-byte instruction stored at 0xC123 is being executed.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

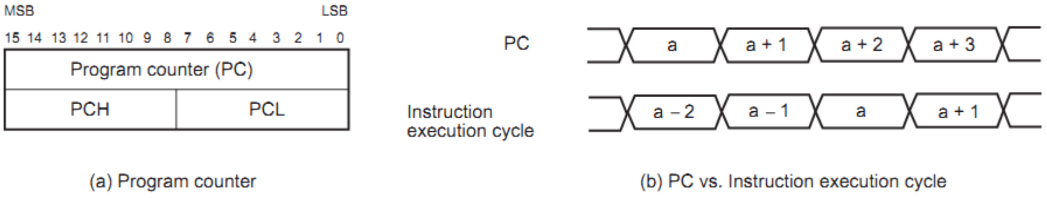


Figure 4.12 Program Counter

4.5.2 Effects of Jump Instructions on the PC Value

There are relative and absolute jump instructions. The jump destination is limited within the code area; a jump cannot occur to the data area. The following describes the effects of jump instructions on the PC value.

(1) Relative Jump Instruction with a 5-bit Displacement (JRS cc, $ + 2 + d)

When the memory location at 0xE8C4 contains the instruction "JRS T, $ + 2 + 0x08", if JF = 1, the PC is incremented by 0x08; i.e., a jump occurs to the address 0xE8CE. (The PC points to an address two bytes after the address of the instruction being executed. In this example, the PC contains 0xE8C4 + 2 = 0xE8C6 before the jump.)

(2) Relative Jump Instructions with an 8-bit Displacement (JR cc, $ + 2 + d / JR cc, $ + 3 + d)

When the memory location at 0xE8C4 contains the instruction "JR Z, $ + 2 + 0x80", if ZF = 1, a jump occurs to an address that is calculated by PC + 0xFF80 (-128). Thus the jump destination is 0xE846.

(3) 16-bit Absolute Jump Instruction (JP a)

When the memory location at 0xE8C4 contains the instruction "JP 0xC235", a jump occurs unconditionally to the address 0xC235. The absolute jump instruction can jump to a location within the full range of the code area (therefore, 8K Bytes for MQ6935).

4.6 General-Purpose Register

MQ6935 has eight 8-bit general-purpose registers called W, A, B, C, D, E, H and L. These registers can be used as 16-bit register pairs called WA, BC, DE and HL.

The general-purpose registers are not mapped to the address space. The contents of the general-purpose registers are undefined after power-up and reset.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

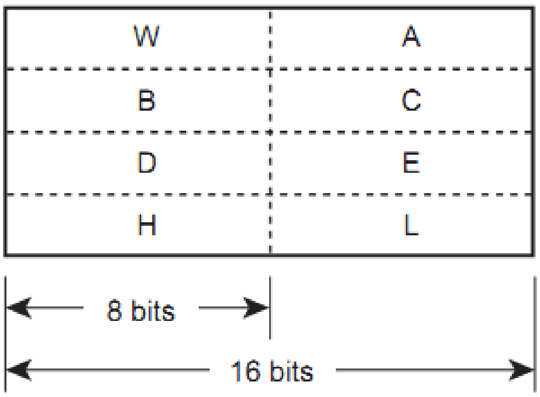


Figure 4.13 General-Purpose Registers

The W, A, B, C, D, E, H and L registers are individually used by the 8-bit load/store and ALU instructions.

The WA, BC, DE and HL register pairs are used by the 16-bit load/store and ALU instructions. These registers also provide the functionalities discussed in the following subsections in addition to the common characteristics as general-purpose registers.

4.6.1 A Registers

Bit manipulation instructions can use the A register to specify a bit position in a register whose value should be tested or changed.

The A register is also used as an offset register in PC-Relative Register Indirect Addressing (PC + A).

4.6.2 C Registers

For divide instructions, the C register holds the divisor. The remainder is written back into the upper byte of the register pair specified as the dividend; the quotient is written back into the lower byte.

The C register is also used as an offset register in Register Indexed Addressing (HL + C).

4.6.3 DE Registers

In Register Indirect Addressing, the DE register holds the address of the memory location where the operand resides.

4.6.4 HL Registers

In Register Indirect Addressing, the HL register holds the address of the memory location where the operand resides. In Indexed Addressing, the HL register is used as an index register.

4.6.5 16-Bit General-Purpose Registers (IX, IY)

MQ6935 has two 16-bit general-purpose registers called IX and IY. In Register Indirect Addressing, these registers hold the address of the memory location where the operand resides. In Indexed Addressing, they are used as index registers.

The contents of the IX and IY registers are undefined after power-up and reset.

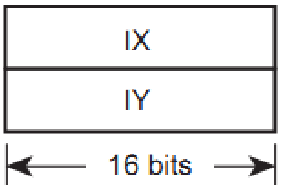


Figure 4.14 16-Bit General-Purpose Registers

The load/store and ALU instructions can also use the IX and IY registers as 16-bit general-purpose registers.

Note:Two register banks (BANK0 and BANK1) are available. Each bank consists of 8-bit general-purpose registers (W, A, B, C, D, E, H, and L) and 16-bit general-purpose registers (IX and IY).

Example : Saving/restoring registers, using an instruction for transfer with data memory (with the main task using the register bank BANK0)

```
PINTxx：         LD                    RBS,1               ;Switches to the register bank
                                                           BANK1
                 Interrupt
                 processing
                 RETI                                      ;RETURN
                                                           (Makes a return automatically to
                                                           BANK0 that was being used by the
                                                           main task when the PSW is
                                                           restored)
```

4.7 Program Status Word (PSW)

The Program Status Word, which resides at address 0x003F in the SFR, consists of the following seven flags:

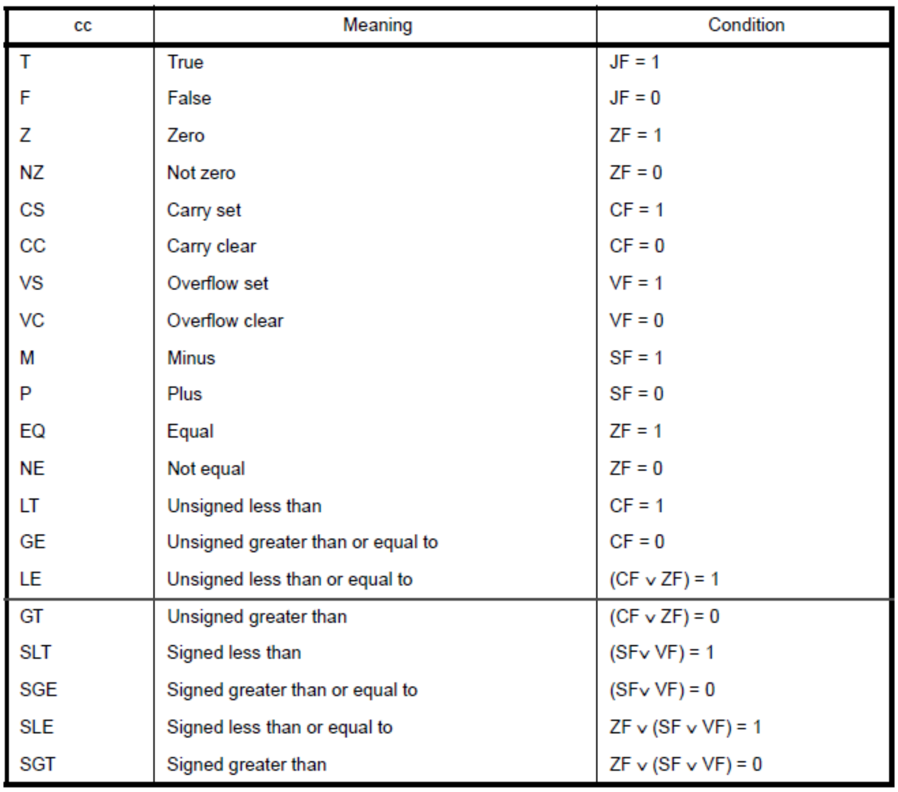
- Jump Status Flag (JF)
- Zero Flag (ZF)
- Carry Flag (CF)
- Half Carry Flag (HF)
- Sign Flag (SF)
- Overflow Flag (VF)

Dedicated instructions are available to access the PSW. General load instructions can also be used to read the PSW.

Organization of the PSW

PSW (0x003F)	7	6	5	4	3	2	1	0
	JF	ZF	CF	HF	SF	VF	-	-

The PSW consists of seven bits of status information that are set or cleared by CPU operations. The flags can be specified as a condition code (cc) in conditional jump instructions, "JR cc, a" and "JRS cc, a", exceptHF.



cc	Meaning	Condition
T	True	JF = 1
F	False	JF = 0
Z	Zero	ZF = 1
NZ	Not zero	ZF = 0
CS	Carry set	CF = 1
CC	Carry clear	CF = 0
VS	Overflow set	VF = 1
VC	Overflow clear	VF = 0
M	Minus	SF = 1
P	Plus	SF = 0
EQ	Equal	ZF = 1
NE	Not equal	ZF = 0
LT	Unsigned less than	CF = 1
GE	Unsigned greater than or equal to	CF = 0
LE	Unsigned less than or equal to	(CF v ZF) = 1
GT	Unsigned greater than	(CF v ZF) = 0
SLT	Signed less than	(SFv VF) = 1
SGE	Signed greater than or equal to	(SFv VF) = 0
SLE	Signed less than or equal to	ZF v (SF v VF) = 1
SGT	Signed greater than	ZF v (SF v VF) = 0

Table 4.3 Condition Code (cc) Table

The instruction "LD PSW clears all the other bits in the PSW.

An attempt to write to the address 0x3F using a load instruction is ignored. Instead, the PSW bits are set or cleared, as predefined for a given instruction.

Upon an interrupt, the PSW is pushed (saved) onto the stack, together with the Program Counter. The content of the stack is popped (restored) to the PSW by the return-from-interrupt instructions, RETI and RETN.

The values of the PSW bits become undefined upon power-up and reset.

4.7.1 Zero Flag (ZF)

The ZF bit is set to 1 when the result of the last ALU instruction or the operand of the last load/store instruction is 0x00 (for 8-bit ALU or load/store operations) or 0x0000 (for 16-bit ALU operations). The ZF bit is also set to 1 when the value of the bit specified by the last bit manipulation instruction is zero; otherwise, the ZF bit is cleared to 0. Also, the ZF bit is set when the high-order eight bits of the product of the last multiply instruction or the remainder of the last divide instruction is 0x00; otherwise, the ZF bit is cleared to 0.

4.7.2 Carry Flag (CF)

The CF bit contains a carry from an addition or a borrow as a result of subtraction. The CF bit is also set to 1 when the divisor of the last divide instruction is 0x00 (divided-by-zero error) or the quotient is equal to or greater than 0x100 (quotient overflow error). Shift and rotate instructions operate with and through the CF bit. For bit manipulation instructions, the CF bit serves as a single-bit Boolean accumulator. The CF bit can be set, cleared and complemented via instructions.

4.7.3 Half Carry Flag (HF)

The HF bit contains a carry to bit 4 or a borrow from bit 4 as a result of an 8-bit addition or subtraction. The HF bit is used for binary-coded decimal (BCD) addition / subtraction and correction, DAA r and DASr.

4.7.4 Sign Flag (SF)

The SF bit is set to 1 when the most significant bit (MSB) of the result of the last arithmetic operation is one. Otherwise, the SF bit is cleared to 0.

4.7.5 Overflow Flag (VF)

The VF bit is set to 1 when there is an overflow as a result of an arithmetic operation. Otherwise, the VF bit is cleared to 0. For example, the VF bit is set when adding two positive numbers gives a negative result or when adding two negative numbers gives a positive result.

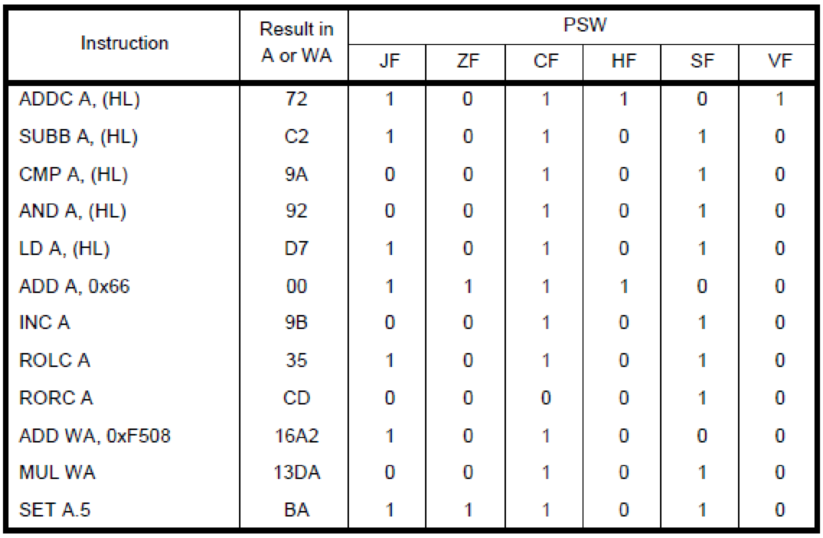
4.7.6 Jump Status Flag (JF)

The JF bit is usually set to 1, and is cleared to 0 or hold a carry according to a specific instruction. The JF bit is used as a condition for conditional jump instructions, "JR T/F, a" and "JRS T/F, a" (where T and F represent true and false condition codes).

Example: The assumptions are:

WA register = 0x219A
HL register = 0x00C5
Data Memory location at 0x000C5 = 0xD7
CF = 1, HF = 0, SF = 1, VF = 0

The following table shows how the A and WA registers and the PSW bits are affected by various instructions.



Instruction	Result in A or WA	PSW					
		JF	ZF	CF	HF	SF	VF
ADDC A, (HL)	72	1	0	1	1	0	1
SUBB A, (HL)	C2	1	0	1	0	1	0
CMP A, (HL)	9A	0	0	1	0	1	0
AND A, (HL)	92	0	0	1	0	1	0
LD A, (HL)	D7	1	0	1	0	1	0
ADD A, 0x66	00	1	1	1	1	0	0
INC A	9B	0	0	1	0	1	0
ROLC A	35	1	0	1	0	1	0
RORC A	CD	0	0	0	0	1	0
ADD WA, 0xF508	16A2	1	0	1	0	0	0
MUL WA	13DA	0	0	1	0	1	0
SET A.5	BA	1	1	1	0	1	0

Table 4.4 Examples of How A and WA Registers, and the PSW Bits Affected by Various Instructions

4.8 Low Power Consumption Function for Peripherals

MQ6935 has low power consumption registers (POFFCRn) that save power when specific peripheral functions are unused. Each bit of the low power consumption registers can be set to enable or disable each peripheral function. (n = 0, 1, 2, 3)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

The basic clock supply to each peripheral function is disabled for power saving, by setting the corresponding bit of the low power consumption registers (POFFCRn) to "0". (The disabled peripheral functions become unavailable.) The basic clock supply to each peripheral function is enabled and the function becomes available by setting the corresponding bit of the low power consumption registers (POFFCRn) to "1".

After reset, the low power consumption registers (POFFCRn) are initialized to "0", and thus the peripheral functions are unavailable. When each peripheral function is used for the first time, be sure to set the corresponding bit of the low power consumption registers (POFFCRn) to "1" in the initial settings of the program (before operating the control register for the peripheral function).

When a peripheral function is operating, the corresponding bit of the low power consumption registers (POFFCRn) must not be changed to "0". If it is changed, the peripheral function may operate unexpectedly.

Low Power Consumption Register 0

POFFCR0 (0x0F74)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCA1EN	TCA0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TC023EN	TC02, 03 enable control	0: Disable 1: Enable
TC001EN	TC00, 01 enable control	0: Disable 1: Enable
TCC0EN	TCC0 enable control	0: Disable 1: Enable
TCA1EN	TCA1 enable control	0: Disable 1: Enable
TCA0EN	TCA0 enable control	0: Disable 1: Enable

Low Power Consumption Register 1

POFFCR1 (0x0F75)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 control	0: Disable 1: Enable
UART2EN	UART2 control	0: Disable 1: Enable
UART1EN	UART1 control	0: Disable 1: Enable
UART0EN	UART0 control	0: Disable 1: Enable

Low Power Consumption Register 2

POFFCR2 (0x0F76)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	RTCEN	-	-	-	-	SIO0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

RTCEN	RTC enable control	0: Disable 1: Enable
SIO0EN	SIO0 control	0: Disable 1: Enable

Low Power Consumption Register 3

POFFCR3 (0x0F77)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	INT5EN	INT4EN	INT3EN	INT2EN	INT1EN	INT0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 Control	0: Disable 1: Enable
INT4EN	INT4 Control	0: Disable 1: Enable
INT3EN	INT3 Control	0: Disable 1: Enable
INT2EN	INT2 Control	0: Disable 1: Enable
INT1EN	INT1 Control	0: Disable 1: Enable
INT0EN	INT0 Control	0: Disable 1: Enable

4.9 Key-on Wakeup (KWU)

The key-on wakeup is a function for releasing the STOP mode at pins KWI7 through KWI2.

4.9.1 Configuration

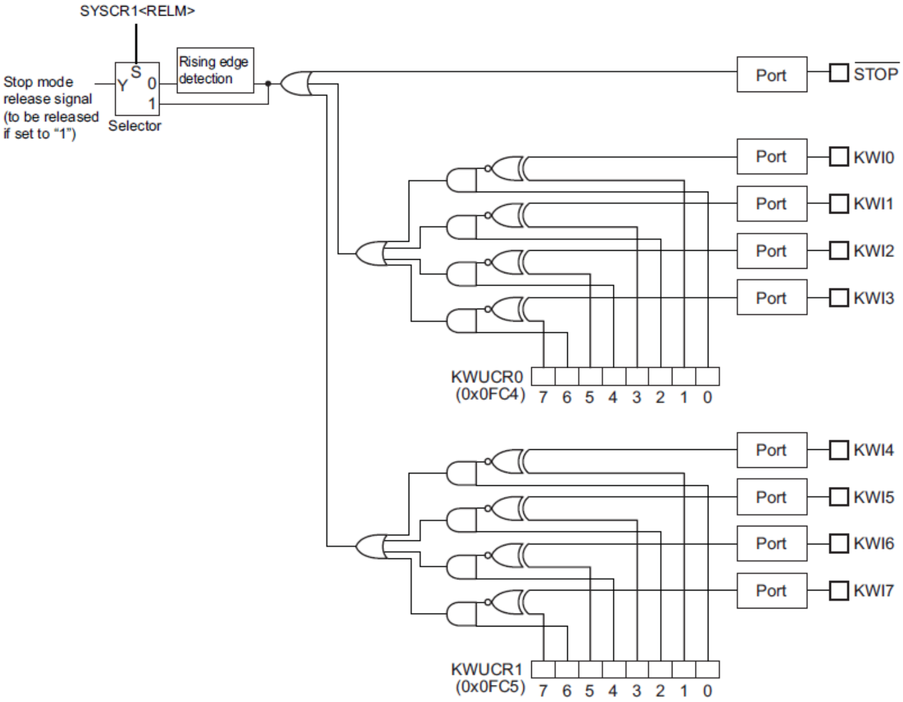


Figure 4.15 Key-on Wakeup Circuit

4.9.2 Control

Key-on wakeup control registers (KWUCR0 and KWUCR1) can be configured to designate the key-on wakeup pins (KWI7 through KWI2) as STOP mode release pins and to specify the STOP mode release levels of each of these designated pins.

Key-on Wakeup Control Register 0

KWUCR0 (0x0FC4)	7	6	5	4	3	2	1	0
Bit Symbol	KW3LE	KW3EN	KW2LE	KW2EN	KW1LE	KW1EN	KW0LE	KW0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

KW3LE	STOP mode release level of KWI3 pin	0: Low level 1: High level
KW3EN	Input enable / disable control of KWI3 pin	0: Disable 1: Enable
KW2LE	STOP mode release level of KWI2 pin	0: Low level 1: High level

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

KW2EN	Input enable / disable control of KWI2 pin	0: Disable 1: Enable
KW1LE	STOP mode release level of KWI1 pin	0: Low level 1: High level
KW1EN	Input enable / disable control of KWI1 pin	0: Disable 1: Enable
KW0LE	STOP mode release level of KWI0 pin	0: Low level 1: High level
KW0EN	Input enable / disable control of KWI0 pin	0: Disable 1: Enable

Key-on Wakeup Control Register 1

KWUCR1 (0x0FC5)	7	6	5	4	3	2	1	0
Bit Symbol	KW7LE	KW7EN	KW6LE	KW6EN	KW5LE	KW5EN	KW4LE	KW4EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

KW7LE	STOP mode release level of KWI7 pin	0: Low level 1: High level
KW7EN	Input enable / disable control of KWI7 pin	0: Disable 1: Enable
KW6LE	STOP mode release level of KWI6 pin	0: Low level 1: High level
KW6EN	Input enable / disable control of KWI6 pin	0: Disable 1: Enable
KW5LE	STOP mode release level of KWI5 pin	0: Low level 1: High level
KW5EN	Input enable / disable control of KWI5 pin	0: Disable 1: Enable
KW4LE	STOP mode release level of KWI4 pin	0: Low level 1: High level
KW4EN	Input enable / disable control of KWI4 pin	0: Disable 1: Enable

4.9.3 Function

By using the key-on wakeup function, the STOP mode can be released at KWIm pin (m: 0 through 7). To designate the KWIm pin as a STOP mode release pin, it is necessary to configure the key-on wakeup control register (KWUCRn) (n: 0 or 1).

4.9.3.1 Setting KWUCRn and P4PU Registers

To designate a key-on wakeup pin (KWIm) as a STOP mode release pin, set KWUCRn <KWmEN> to "1". After KWIm pin is set to "1" at KWUCRn <KWmEN>, a specific STOP mode release level can be specified for this pin at KWUCRn <KWmLE>. If KWUCRn <KWmLE> is set to "0", STOP mode is released when an input is at a low level. If it is set to "1", STOP mode is released when an input is at a high level. For example, if you want to release STOP mode by inputting a high-level signal into a

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

KWI0 pin, set KWUCR0 <KW0EN> to "1", and KWUCR0 <KW0LE> to "1".

4.9.3.2 Starting STOP Mode

To start the STOP mode, set SYSCR1 <RELM> to "1" (level release mode), and SYSCR1 <STOP> to "1".

To use the key-on wakeup function, do not set SYSCR1 <RELM> to "0" (edge release mode). If the key-on wakeup function is used in edge release mode, STOP mode cannot be released. This is because the KWIm pin enabling inputs to be received is at a release level after the STOP mode starts.

4.9.3.3 Releasing STOP Mode

To release STOP mode, input a specific release level into the KWIm pin for which receipt of inputs is enabled.

If the KWIm pin is already at a release level when the STOP mode starts, the following instruction will be executed without starting the STOP mode (with no warm-up performed).

Note): Do not applied an analog voltage to KWIm pin for which receipt of inputs is enabled by the key-on wakeup control register (KWUCRn) setting, or a penetration current will flow.

Pin name	Release level (edge)		
	SYSCR1<RELM>="1" (level release mode)		SYSCR1<RELM>="0" (edge release mode)
	KWUCRn<KWmLE>="0"	KWUCRn<KWmLE>="1"	
STOPB	"H" level		Rising edge
KWIm	"L" level	"H" level	Don't use

Table 4.5 STOP Mode Release Level (edge)

Example : A case in which STOP mode is started with the release level of the STOPB pin set to a high level and the release level of KWI0 set to a low level (connected to an internal pull-up resistor of the KWI0 pin)

```
DI                              ; IMF←0
SET     (P4PU).0                ; KWI0 (P40) connected to a pull-up resistor
LD      (KWUCR0), 0y00000001    ; the KWI0 pin is set to enable inputs,
                                ; and its release level is set to a low level.
LD      (SYSCR1), 0y10100000    ; Starting in level release mode
```

5. Reset Function

5.1 Reset Control Circuit

The reset circuit controls the external and internal factor resets and initializes the system.

5.1.1 Configuration

The reset circuit controls the external and internal factor resets and initializes the system.

1. External reset input (external factor)
2. Power-on reset (internal factor)
3. Voltage detection reset (internal factor)
4. Watchdog timer reset (internal factor)
6. System clock reset (internal factor)
7. Trimming data reset (internal factor)
8. Flash standby reset (internal factor)

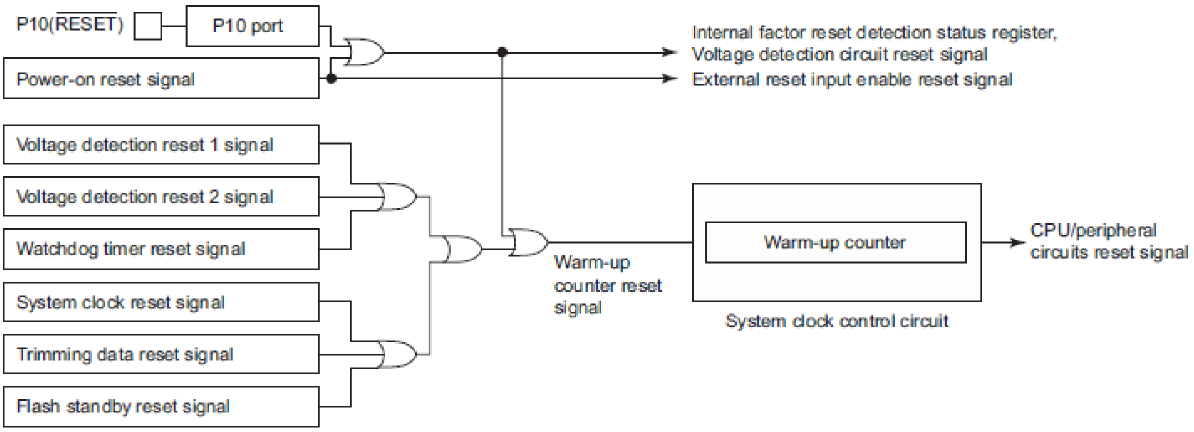

P10(RESET)
P10 port
Power-on reset signal
Voltage detection reset 1 signal
Voltage detection reset 2 signal
Watchdog timer reset signal
System clock reset signal
Trimming data reset signal
Flash standby reset signal
Internal factor reset detection status register,
Voltage detection circuit reset signal
External reset input enable reset signal
Warm-up counter reset signal
Warm-up counter
System clock control circuit
CPU/peripheral circuits reset signal

Figure 5.1 Reset Control Circuit

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

5.1.2 Control

The reset control circuit is controlled by system control register 3 (SYSCR3), system control register 4 (SYSCR4), system control status register (SYSSR4) and the internal factor reset detection status register (IRSTSR).

System Control Register 3

SYSCR3 (0x0FDE)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	RVCTR	RAREA	(RSTDIS)
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

RAREA	Specifies mapping of the RAM in the code area	0: The RAM is not mapped from 0x0040 to 0x083F in the code area. 1: The RAM is mapped from 0x0040 to 0x083F in the code area.
RVCTR	Specifies mapping of the vector table for vector call instructions and interrupts	Vector table for vector call instructions 0:0xFFA0 to 0xFFBF in the code area 1: 0x01A0 to 0x01BF in the code area Vector table for interrupt 0: 0xFFC2 to 0xFFFF in the code area 1: 0x01C2 to 0x01FD in the code area
RSTDIS	External reset input enable register	0: Enable the external reset input 1: Disable the external reset input

Note 1): The enabled SYSCR3 <RSTDIS> is initialized by a power-on reset only, and cannot be initialized by an external reset input or internal factor reset. The value written in SYSCR3 is reset by a power-on reset, external reset input or internal factor reset.

Note 2): The value of SYSCR3 <RSTDIS> is invalid until 0xB2 is written into SYSCR4.

Note 3): After SYSCR3 <RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3 <RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR <FCGCKSEL> = 00). Otherwise, SYSCR 3<RSTDIS> may be enabled at unexpected timing.

Note 4): Bits 7 to 3 of SYSCR3 are read as "0".

System Control Register 4

SYSCR4 (0x0FDF)	7	6	5	4	3	2	1	0
Bit Symbol	SYSCR4							
Read/Write	Write only							
After reset	0	0	0	0	0	0	0	0

SYSCR4	Write the SYSCR3 data control code	0xB2:	Enable the contents of SYSCR3 <RSTDIS>
		0xD4	Enable the contents of SYSCR3 <RAREA> and SYSCR3 <RVCTR>
		0x71:	Enable the contents of IRSTSR <FCLR>
		Others:	Invalid

Note 1): SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit

operation.

Note 2): After SYSCR3 <RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3 <RSTDIS>) in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3 <RSTDIS> may be enabled at unexpected timing.

Note 3): After IRSTSR <FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR <FCLR> in NORMAL mode when fcgck is fc/4 (CGCR <FCGCKSEL>=00). Otherwise, IRSTSR <FCLR> may be enabled at unexpected timing.

System Control Status Register 4

SYSSR4 (0x0FDF)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	RVCTRS	RAREAS	(RSTDISS)
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

RAREAS	Status of mapping of the RAM in the code area	0: The enabled SYSCR3<RAREA> data is "0". 1: The enabled SYSCR3<RAREA> data is "1".
RVCTRS	Status of mapping of the vector address in the area	0: The enabled SYSCR3<RVCTR> data is "0". 1: The enabled SYSCR3<RVCTR> data is "1".

Note 1): The enabled SYSCR3 <RSTDIS> is initialized by a power-on reset only, and cannot be initialized by any other reset signals. The value written in SYSCR3 is reset by a power-on reset and other reset signals.

Note 2): Bits 7 to 3 of SYSCR4 are read as "0".

Note 3): Example- program transfer(Transfer the program saved in the data area to the RAM)

```
LD      HL,TRANSFER_STAR_ADDRESS      ;Destination RAM address
LD      DE,PROGRAM_START_ADDRESS      ;Source ROM address
LD      BC,BYTE_OF_PROGRAM            ;Number of bytes of the program to be executed-1
LD      A,(DE)                        ;Reading the program to be transferred
LD      (HL),A                        ;Writing the program to be transferred
INC     HL                            ;Destination address increment
INC     DE                            ;Source address increment
DEC     BC                            ;Source address increment
J       F,TRANS_RAM                   ;Have all the programs been transferred?
```

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Internal Factor Reset Detection Status Register

IRSTSR (0x0FCC)	7	6	5	4	3	2	1	0
Bit Symbol	FCLR	FLSRF	TRMDS	TRMRF	LVD2RF	LVD1RF	SYSRF	WDTRF
Read/Write	W	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

FCLR	Flag initialization control	0: - 1: Clear the internal factor reset flag to "0".
FLSRF	Flash standby reset detection flag	0:- 1: Detect the flash standby reset.
TRMDS	Trimming data status	0: - 1: Detect state of abnormal trimming data
TRMRF	Trimming data reset detection flag	0: - 1: Detects the trimming data reset.
LVD2RF	Voltage detection reset 2 detection flag	0: - 1: Detect the voltage detection 2 reset.
LVD1RF	Voltage detection reset 1 detection flag	0: - 1: Detect the voltage detection 1 reset.
SYSRF	System clock reset detection flag	0: - 1: Detect the system clock reset.
WDTRF	Watchdog timer reset detection flag	0:- 1: Detect the watchdog timer reset.

Note 1: Internal reset factor flag (IRSTSR<FLSRF, TRMDS, TRMRF, LVD2RF, LVD1RF, SYSRF, WDTRF>) is initialized only by a power-on reset, an external reset input or IRSTSR <FCLR>. It is not initialized by an internal factor reset.

Note 2: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 3: If SYSCR4 is set to 0x71 after IRSTSR<FCLR> is set to "1", internal factor reset flag is cleared to "0" and IRSTSR<FCLR> is automatically cleared to "0".

Note 4: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

Note 5: Bit 7 of IRSTSR is read as "0"..

5.1.3 Function

The power-on reset, external reset input and internal factor reset signals are input to the warm-up circuit of the clock generator.

During reset, the warm-up counter circuit is reset, and the CPU and the peripheral circuits are reset.

After reset is released, the warm-up counter starts counting the high frequency clock (fc), and executes the warm-up operation that follows reset release.

During the warm-up operation that follows reset release, the trimming data is loaded from the embedded flash memory for adjustment of the ladder resistor that generates the comparison voltage

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

for the power-on reset and the voltage detection circuits.

When the warm-up operation that follows reset release is finished, the CPU starts execution of the program from the reset vector address stored in addresses 0xFFFE to 0xFFFF.

When a reset signal is input during the warm-up operation that follows reset release, the warm-up counter circuit is reset.

The reset operation is common to the power-on reset, external reset input and internal factor resets, except for the initialization of some special function registers and the initialization of the voltage detection circuits.

When a reset is applied, the peripheral circuits become the states as shown in Table 5.1.

Built-in Hardware	During Reset	During the warm-up operation that follows reset release	Immediately after the warm-up operation that follows reset release
Program counter (PC)	0xFFFE	0xFFFE	0xFFFE
Stack pointer (SP)	0x00FF	0x00FF	0x00FF
RAM	Indeterminate	Indeterminate	Indeterminate
General-purpose registers (W, A, B, C, D, E, H, L, IX and IY)	Indeterminate	Indeterminate	Indeterminate
Jump status flag (JF)	Indeterminate	Indeterminate	Indeterminate
Zero flag (ZF)	Indeterminate	Indeterminate	Indeterminate
Carry flag (CF)	Indeterminate	Indeterminate	Indeterminate
Half carry flag (HF)	Indeterminate	Indeterminate	Indeterminate
Sign flag (SF)	Indeterminate	Indeterminate	Indeterminate
Overflow flag (VF)	Indeterminate	Indeterminate	Indeterminate
Interrupt master enable flag (IMF)	0	0	0
Individual interrupt enable flag (EF)	0	0	0
Interrupt latch (IL)	0	0	0
Hi-freq. clock oscillation circuit	Oscillation enabled	Oscillation enabled	Oscillation enabled
Low-freq. clock oscillation circuit	Oscillation disabled	Oscillation disabled	Oscillation disabled
Warm-up counter	Reset	Start	Stop
Timing generator prescaler and divider	0	0	0
Watchdog timer	Disabled	Disabled	Enabled
Voltage detection circuit	Disabled or enabled	Disabled or enabled	Disabled or enabled
I/O port pin status	HiZ	HiZ	HiZ
Special function register	Refer to the SFR map.	Refer to the SFR map.	Refer to the SFR map.

Table 5.1 Initialization of Built-in Hardware by Reset Operation and Its Status after Release

Note 1): The voltage detection circuits are disabled by an external reset input or power-on reset only.

Note 2): "HiZ" indicates high-impedance.

5.1.4 Reset Signal Generating Factors

Reset signals are generated by each factor as follows:

5.1.4.1 External Reset Input (RESETB Pin Input)

Port P10 is also used as the RESETB pin, and it serves as the RESETB pin after the power is turned on.

- When the supply voltage rises rapidly：

When the power supply rise time (tVDD) is shorter than 5 ms with enough margin, the reset can be released by a power-on reset or an external reset (RESETB pin input).

The power-on reset logic and external reset (RESETB pin input) logic are ORed. This means that the MCU is reset when either or both of these reset sources are asserted. Therefore, the reset time is determined by the reset source with a longer reset period.

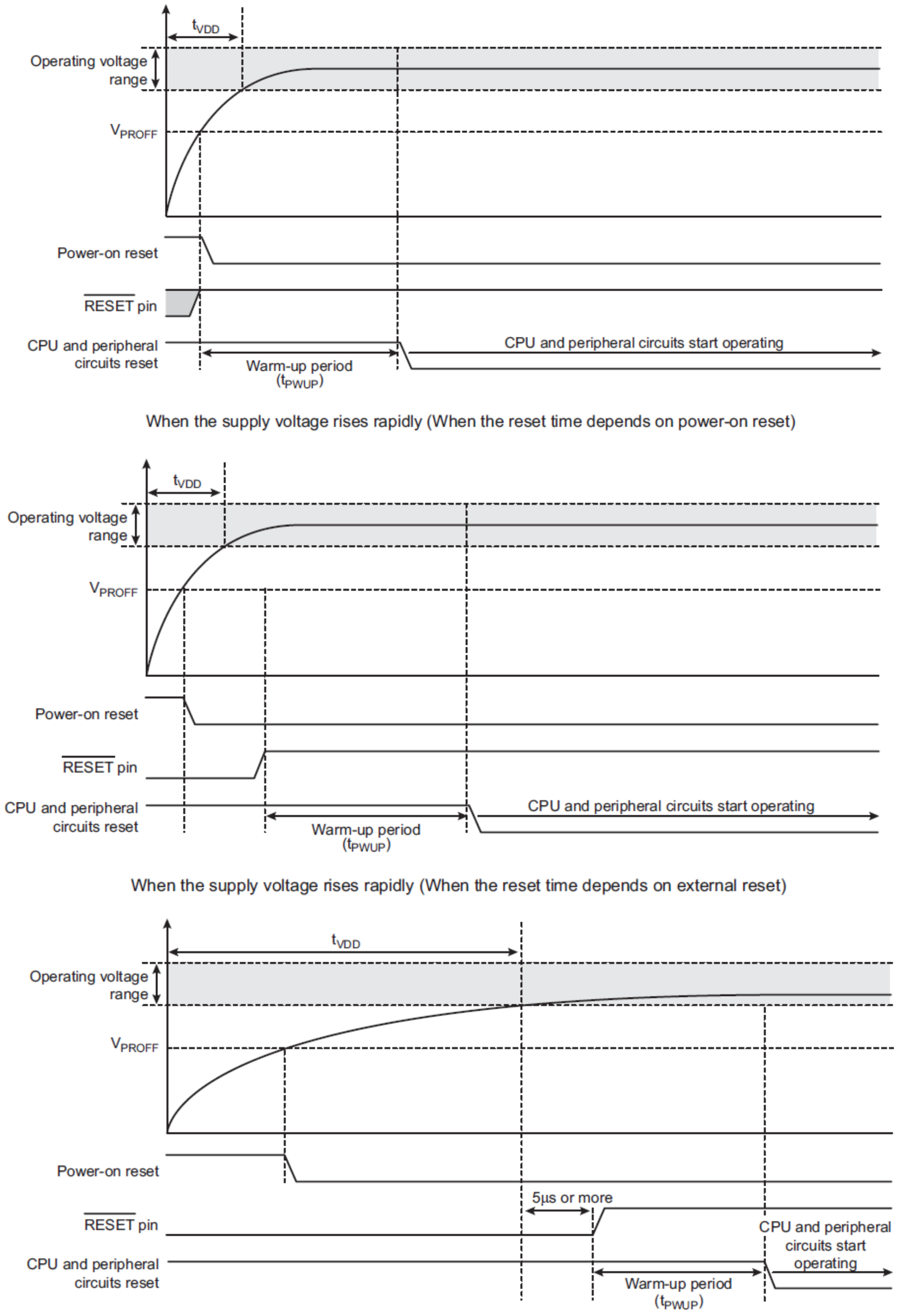
If the RESETB pin level changes from Low to High before the supply voltage rises above the power-on-reset release voltage (VPROFF) (or if the RESETB pin level is "H" from the beginning), the reset time depends on the power-on reset. If the RESETB pin level changes from Low to High after the supply voltage rises above VPROFF, the reset time depends on the external reset.

In the former case, a warm-up period begins when the power-on reset signal is released. In the latter case, a warm-up period begins when the RESETB pin level becomes "H". Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 5.2).

- When the supply voltage rises slowly：

When the power supply rise time (tVDD) is longer than 5 ms, the reset must be released by using the RESETB pin.

In this case, hold the RESETB pin "L" until the supply voltage rises to the operating voltage range and oscillation is stabilized. When this state is achieved, wait at least 5 µs and then pull the RESETB pin "H". Changing the RESETB pin level to "H" starts a warm-up period. Upon completion of the warm-up period, the CPU and peripheral circuits start operating (Figure 5.2).


t_{VDD}
Operating voltage range
V_{PROFF}
Power-on reset
RESET pin
CPU and peripheral circuits reset
Warm-up period (t_{PWUP})
CPU and peripheral circuits start operating
5µs or more
CPU and peripheral circuits start operating

When the supply voltage rises rapidly (When the reset time depends on power-on reset)

When the supply voltage rises rapidly (When the reset time depends on external reset)

When the supply voltage rises slowly

Figure 5.2 External Reset Input (During Power-Up)

If the supply voltage is within the recommended operating voltage range, the RESETB pin is kept at the "L" level for 5 µs with the stabilized oscillation, and then a reset is applied.

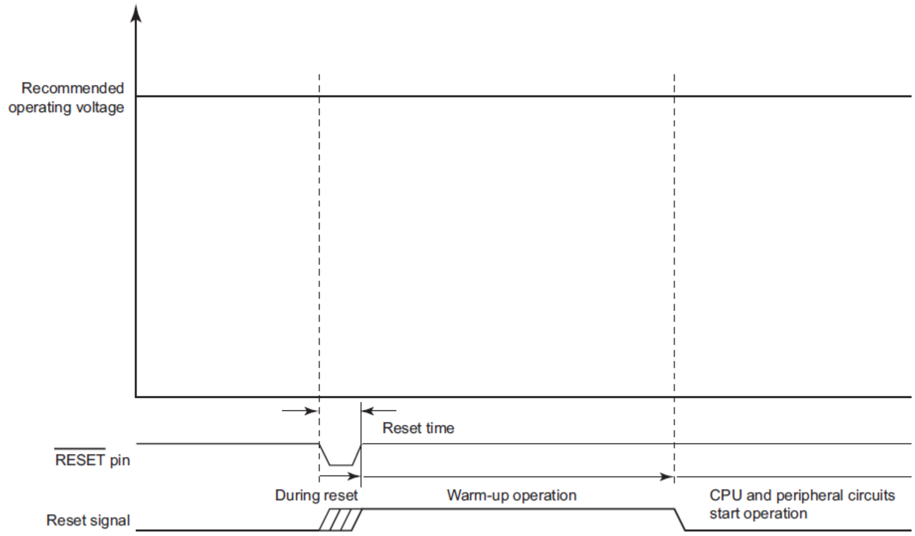

Recommended operating voltage
Reset time
RESET pin
During reset
Warm-up operation
CPU and peripheral circuits start operation
Reset signal

Figure 5.3 External Reset Input (when the power is stabilized)

In each case, after a reset is applied, it is released by turning the RESETB pin to "H" and the warm-up operation that follows reset release gets started.

Note): When the supply voltage is equal to or lower than the detection voltage of the power-on reset circuit, the power-on reset remains active, even if the RESETB pin is turned to "H".

5.1.4.2 Power-on Reset

The power-on reset is an internal factor reset that occurs when the power is turned on.

When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a reset signal is generated, and if it is higher than the releasing voltage of the power-on reset circuit, a reset signal is released.

When power supply voltage goes down, if the supply voltage is equal to or lower than the detecting voltage of the power-on reset circuit, a reset signal is generated.

5.1.4.3 Voltage Detection Reset

The voltage detection reset is an internal factor reset that occurs when it is detected that the supply voltage has reached a predetermined detection voltage. Refer to "5.3 Voltage Detection Circuit".

5.1.4.4 Watchdog Timer Reset

The watchdog timer reset is an internal factor reset that occurs when an overflow of the watchdog timer is detected. Refer to "10.1 Watchdog Timer".

5.1.4.5 System Clock Reset

The system clock reset is an internal factor reset that occurs when it is detected that the oscillation enable register is set to a combination that puts the CPU into deadlock. Refer to "6 System Clock Control".

5.1.4.6 Flash Standby Reset

The flash standby reset is an internal factor reset generated by the reading or writing of data of the flash memory while it is on standby. Refer to "14 Flash Memory".

5.1.4.7 Internal Factor Reset Detection Status Register

By reading the internal factor reset detection status register IRSTSR after the release of an internal factor reset, except the power-on reset, the factor which causes a reset can be detected.

The internal factor reset detection status register is initialized by an external reset input or power-on reset.

Set IRSTSR <FCLR> to "1" and write 0x71 to SYSCR4. This enables IRSTSR <FCLR> and the internal factor reset detection status register is clear to "0". IRSTSR <FCLR> is cleared to "0" automatically after initializing the internal factor reset detection status register.

Note 1): Care must be taken in system designing since the IRSTSR may not fulfill its functions due to noises and other disturbances.
Note 2): After IRSTSR <FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR <FCLR> in NORMAL mode when fcgck is fc/4 (CGCR <FCGCKSEL> = 00). Otherwise, IRSTSR <FCLR> may be enabled at unexpected timing.

5.1.4.8 Trimming Data Reset

The trimming data reset is an internal factor reset that occurs when the trimming data latched in the internal circuit is broken down during operation due to noise or other factors.

The trimming data is a data bit provided for adjustment of the ladder resistor that generates the comparison voltage for the power-on reset and the voltage detection circuits. This bit is loaded from the non-volatile exclusive use memory during the warm-up time that follows reset release (tPWUP) and latched into the internal circuit. If the trimming data loaded from the non-volatile exclusive use memory during the warm-up operation that follows reset release is abnormal, IRSTSR<TRMDS> is set to "1".

When IRSTSR<TRMDS> is read as "1" in the initialize routine immediately after reset release, the trimming data need to be reloaded by generating an internal factor reset, such as a system clock reset, and activating the warm-up operation again. If IRSTSR<TRMDS> is still set to "1" after repeated reading, the detection voltage of the voltage detection circuit and power-on reset circuit does not satisfy the characteristic specified in the electric characteristics. Design the system so that the system will not be damaged in such a case.

5.1.4.9 How to Use P10 as an External Reset

To use P10 as an external reset, keep P10 at the "H" level until the power is turned on and the warm-up operation that follows reset release is finished.

After the warm-up operation that follows power-on reset is finished, set P1CR0 to "0", and connect a pull-up resistor to P10. Then clear SYSCR3 <RSTDIS> to "0" and write 0xB2 to SYSCR4. This enables the external reset function and makes P10 as a reset input pin.

To use the pin as an IO pin when it is used as a reset, set SYSCR3 <RSTDIS> to "1" and write 0xB2 to SYSCR4. This enables the IO function and makes the pin usable as an open-drain IO pin.

Note 1): If you switch the external reset input pin to a port or switch the pin used as a port to the external reset input pin, do it when the pin is stabilized at the "H" level. Switching the pin function when the "L" level is input may cause a reset.
Note 2): If the external reset input is used as a port, the statement which clears SYSCR3 <RSTDIS> to "0" is not written in a program. By this abnormal execution of program, the external reset input set as a port may be changed as the external reset input at unexpected timing.
Note 3): After SYSCR3 <RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3 <RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR <FCGCKSEL> = 00). Otherwise, SYSCR3 <RSTDIS> may be enabled at unexpected timing.

5.2 Power-on Reset Circuit

The power-on reset circuit generates a reset when the power is turned on. When the supply voltage is lower than the detection voltage of the power-on reset circuit, a power-on reset signal is generated.

5.2.1 Configuration

The power-on reset circuit consists of a reference voltage generation circuit and a comparator. The supply voltage divided by ladder resistor is compared with the voltage generated by the reference voltage generation circuit by the comparator.

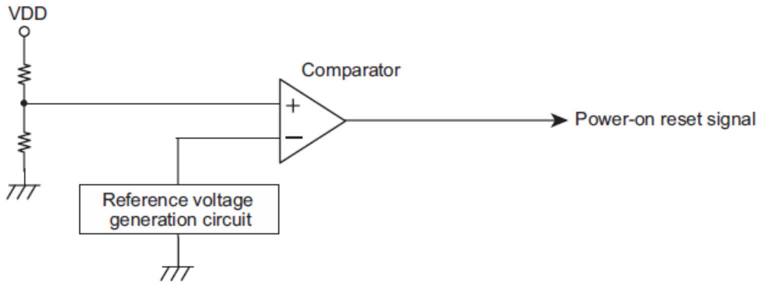


Figure 5.4 Power-on Reset Circuit

5.2.2 Function

When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated and if it is higher than the releasing voltage of the power-on reset circuit, a power-on reset signal is released.

When power supply voltage goes down, if the supply voltage is equal to or lower than the detecting voltage of the power-on reset circuit, a power-on reset signal is generated.

Until the power-on reset signal is generated, a warm-up circuit and a CPU is reset.

When the power-on reset signal is released, the warm-up circuit is activated. The reset of the CPU and peripheral circuits is released after the warm-up time that follows reset release has elapsed.

Increase the supply voltage into the operating range during the period from detection of the power-on reset release voltage until the end of the warm-up time that follows reset release. If the supply voltage has not reached the operating range by the end of the warm-up time that follows reset release, the MCU cannot operate properly.

The detail of power-on characteristics please refer to "3.7 Powe-on Reset Characteristics"

5.3 Voltage Detection Circuit

The voltage detection circuit detects any decrease in the supply voltage and generates INTLVDinterrupt request signals and voltage detection reset signals.

Note): The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (VDD). Refer to the electrical characteristics and take them into consideration when designing equipment.

5.3.1 Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (VDD) is divided by the ladder resistor and input to the detection voltage selection circuit. A voltage is selected in the detection voltage selection circuit, depending on the detection voltage (VDxLVL), and compared to the reference voltage in the comparator. When the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL), a voltage detection interrupt request signal or a voltage detection reset signal is generated. (x = 1to 2)

Whether to generate a voltage detection reset signal or an INTLVD interrupt request signal can be programmed by software. In the former case, a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL). In the latter case, an INTLVD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage level.

Note): Since the comparators used for voltage detection do not have a hysteresis structure, INTLVD interrupt request signals may be generated frequently if the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTLVD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

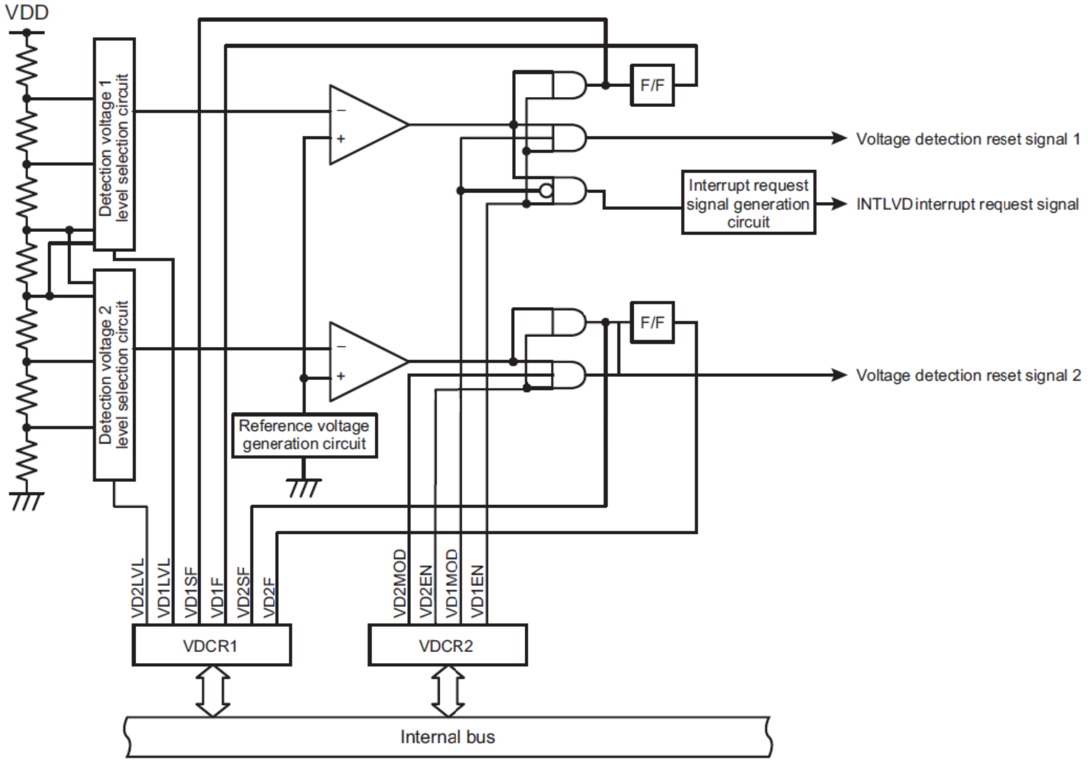

VDD
Detection voltage 1 level selection circuit
Detection voltage 2 level selection circuit
F/F
F/F
Voltage detection reset signal 1
Interrupt request signal generation circuit
INTLVD interrupt request signal
Voltage detection reset signal 2
Reference voltage generation circuit
VD2LVL
VD1LVL
VD1SF
VD1F
VD2SF
VD2F
VD2MOD
VD2EN
VD1MOD
VD1EN
VDCR1
VDCR2
Internal bus

Figure 5.5 Voltage Detection Circuit

5.3.2 Control

The voltage detection circuit is controlled by voltage detection control registers 1 and 2.

Voltage Detection Control Register 1

VDCR1 (0x0FC6)	7	6	5	4	3	2	1	0
Bit Symbol	VD2F	VD2SF	VD2LVL		VD1F	VD1SF	VD1LVL	
Read/Write	R/W	R	R/W		R/W	R	R/W	
After reset	0	0	0	0	0	0	0	0

VD2F	Voltage detection 2 flag (Retain the state when VDD <VD2LVL is detected)		Read	Writer
		0 1	0: VDD ≥ VD2LVL 1: VDD < VD2LVL	Clears VD2F to "0" -
VD2SF	Voltage detection 2 status flag (Magnitude relation of VDD and VD2LVL when they are read)	0 1	0: VDD ≥ VD2LVL 1: VDD < VD2LVL	
VD2LVL	Selection for detection voltage 2	00 01 10 11	2.85V +/- 0.1 V 2.65V +/- 0.1 V 2.35V +/- 0.1 V 2.00V +/- 0.1 V	
VD1F	Voltage detection 1 flag (Retain the state when VDD <VD1LVL is detected)		Read	Writer
		0 1	0: VDD ≥ VD1LVL 1: VDD < VD1LVL	Clears VD1F to "0" -
VD1SF	Voltage detection 1 status flag (Magnitude relation of VDD and VD1LVL when they are read)	0 1	0: VDD ≥ VD1LVL 1: VDD < VD1LVL	
VD1LVL	Selection for detection voltage 1	00 01 10 11	4.50V +/- 0.15 V 4.20V +/- 0.15 V 3.70V +/- 0.15 V 3.15V +/- 0.15 V	

Note 1): VDCR1 is initialized by a power-on reset or an external reset input.

Note 2): When VD2F or VD1F is cleared by the software and is set due to voltage detection at the same time, the setting due to voltage detection is given priority.

Note 3): VD2F and VD1F cannot be programmed to "1" by the software.

Note 4): After VD1EN and VD2EN enabled (and confirmed VD1SF and VD2SF are "0"), need to clear VD1F or VD2F to "0"

Voltage Detection Control Register 2

VDCR2 (0x0FC7)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	"0"	"0"	VD2MOD	VD2EN	VD1MOD	VD1EN
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

VD2MOD	Select the operation mode of voltage detection 2	0: Generate a INTLVDinterrupt request signal 1: Generate a voltage detection reset 2 signal
VD2EN	Enable / disable the operation of voltage detection 2	0: Disable the operation of voltage detection 2 1: Enable the operation of voltage detection 2
VD1MOD	Select the operation mode of voltage detection 1	0: Generate a INTLVD interrupt request signal 1: Generate a voltage detection reset 1 signal
VD1EN	Enable / disable the operation of voltage detection 1	0: Disable the operation of voltage detection 1 1: Enable the operation of voltage detection 1

Note 1): VDCR2 is initialized by a power-on reset or an external reset input.

Note 2): Bits 7 and 6 of VDCR2 are read as "0".

Note 3): Bits 5 and 4 of VDCR2 should be cleared to "0".

Note 4): To use VD2LVL, besides bit 3 and bit 2, bit 0 of VDCR2 should also be set as "1".

5.3.3 Function

Two detection voltages (VDxLVL, x = 1 to 2) can be set in the voltage detection circuit. For each voltage, enabling/disabling the voltage detection and the operation to be executed when the supply voltage (VDD) falls to or below the detection voltage (VDxLVL) can be programmed.

5.3.3.1 Enabling / Disabling the Voltage Detection Operation

Setting VDCR2 <VDxEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation. VDCR2 <VDxEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note): When the supply voltage (VDD) is lower than the detection voltage (VDxLVL), setting VDCR2 <VDxEN> to "1" generates an INTLVD interrupt request signal or a voltage detection reset signal at the time.

5.3.3.2 Selecting the Voltage Detection Operation Mode

When VDCR2 <VDxMOD> is set to "0", the voltage detection operation mode is set to generate INTLVD interrupt request signals. When VDCR2 <VDxMOD> is set to "1", the operation mode is set to generate voltage detection reset signals.

(a) When the operation mode is set to generate INTLVD interrupt signals (VDCR2 <VDxMOD>="0")

When VDCR2<VDxEN>="1", an INTLVD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage (VDxLVL).

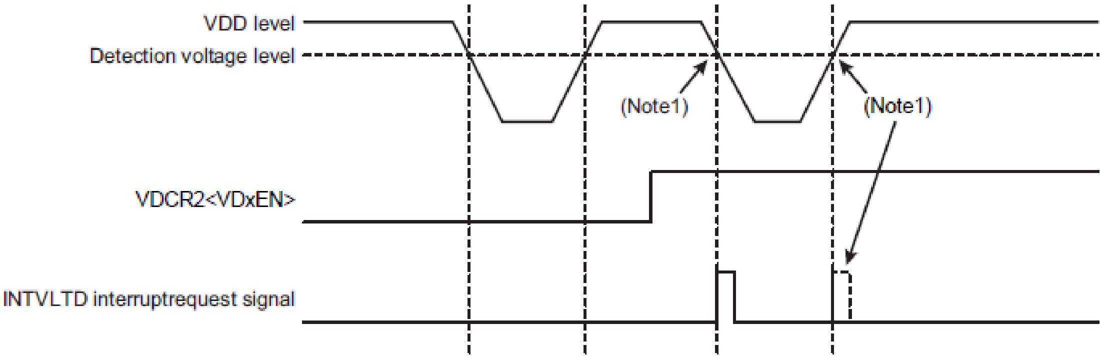


Figure 5.6 Voltage Detection Interrupt Request

Note 1): Since the comparators used for voltage detection do not have a hysteresis structure, INTLVD interrupt request signals may be generated frequently when the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTLVD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

Note 2): If the supply voltage (VDD) falls to the detection voltage (VDxLVL) during IDLE0 or SLEEP0 mode, an INTLVD interrupt request signal is generated after the TBT counts the specified period and IDLE0 or SLEEP mode is released.

(b) When the operation mode is set to generate voltage detection reset signals (VDCR2 <VDxMOD>="1")

When VDCR2 <VDxEN> = "1", a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL).

VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. A voltage detection reset signal is generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDxLVL).

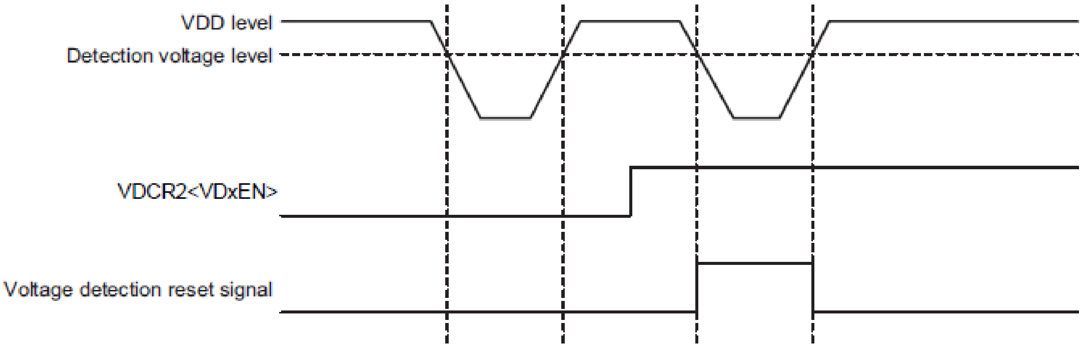


Figure 5.7 Voltage Detection Reset Signal

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

5.3.3.3 Selecting the Detection Voltage Level

Select a detection voltage at VDCR1<VDxLVL>.

5.3.3.4 Voltage Detection Flag and Voltage Detection Status Flag

The magnitude relation between the supply voltage (VDD) and the detection voltage (VDxLVL) can be checked by reading VDCR1 <VDxF> and VDCR1 <VDxSF>.

If VDCR2 <VDxEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL), VDCR1 <VDxF> is set to "1" and is held in this state. VDCR1<VDxF> is not cleared to "0" when the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDxLVL).

When VDCR2 <VDxEN> is cleared to "0" after VDCR1 <VDxF> is set to "1", the previous state is still held. To clear VDCR1 <VDxF>, "0" must be written to it.

If VDCR2 <VDxEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL), VDCR1 <VDxSF> is set to "1". When the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDxLVL), VDCR1 <VDxSF> is cleared to "0".

Unlike VDCR1 <VDxF>, VDCR1 <VDxSF> does not hold the set state.

Note 1): When the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL) in the STOP, IDLE0 or SLEEP0 mode, the voltage detection flag and the voltage detection status flag are changed after the operation mode is returned to NORMAL or SLOW mode.

Note 2): Depending on the voltage detection timing, the voltage detection status flag (VDxSF) may be changed earlier than the voltage detection flag (VD2F) by a maximum of 2/fcgck[s].

Note 3): After VD1EN and VD2EN set at "1"(and confirmed VD1SF and VD2SF are "0"), need to clear VD1F or VD2F to "0" by software.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

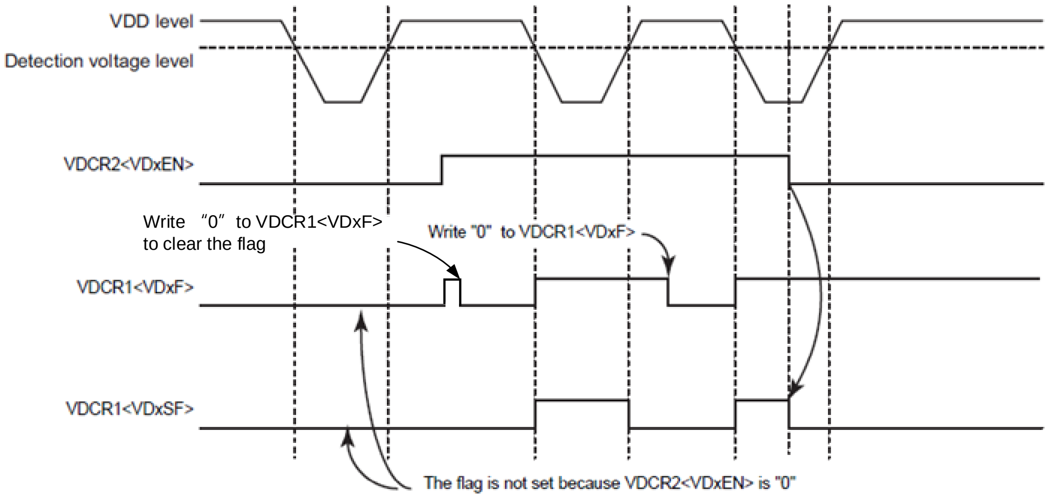


Figure 5.8 Changes in the Voltage Detection Flag and the Voltage Detection Status Flag

5.3.4 Register Setting

5.3.4.1 When the Operation Mode is Set to Generate INTLVD Interrupt Request Signals

When the operation mode is set to generate INTLVD interrupt request signal, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Set the detection voltage at VDCR1<VDxLVL>, x = 1 to 2.
3. Clear VDCR2 <VDxMOD> to "0" to set the operation mode to generate INTLVD interrupt request signals.
4. Set VDCR2<VDxEN> to "1" to enable the voltage detection operation.
5. Wait for 5 µs or more until the voltage detection circuit becomes stable.
6. Make sure that VDCR1 <VDxSF> is "0".
7. Clear the voltage detection circuit interrupt latch to "0" and set the interrupt enable flag to "1" to enable interrupts.

Note): When the supply voltage (VDD) is close to the detection voltage (VDxLVL), voltage detection request signals may be generated frequently. If this may pose any problem, execute appropriate wait processing depending on fluctuations in the system power supply and clear the interrupt latch before returning from the INTLVD interrupt service routine.

To disable the voltage detection circuit while it is enabled with the INTLVD interrupt request, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Clear VDCR2 <VDxEN> to "0" to disable the voltage detection operation.

Note): If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

5.3.4.2 When the Operation Mode is Set to Generate Voltage Detection Reset Signals

When the operation mode is set to generate voltage detection reset signals, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Set the detection voltage at VDCR1 <VDxLVL>, x = 1 to 2.
3. Clear VDCR2 <VDxMOD> to "0" to set the operation mode to generate INTLVD interrupt request signals.
4. Set VDCR2 <VDxEN> to "1" to enable the voltage detection operation.
5. Wait for 5 μs or more until the voltage detection circuit becomes stable.
6. Make sure that VDCR1 <VDxSF> is "0".
7. Clear VDCR1 <VDxF> to “0”.
8. Set VDCR2 <VDxMOD> to "1" to set the operation mode to generate voltage detection reset signals.

Note 1): VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. If the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL) in the period from release of the voltage detection reset until clearing of VDCR2 <VDxEN> to "0", a voltage detection reset signal is generated immediately.
Note 2): The voltage detection reset signals are generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDxLVL).

To disable the voltage detection circuit while it is enabled with the voltage detection reset, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Clear VDCR2 <VDxMOD> to "0" to set the operation mode to generate INTLVD interrupt request signals.
3. Clear VDCR2 <VDxEN> to "0" to disable the voltage detection operation.

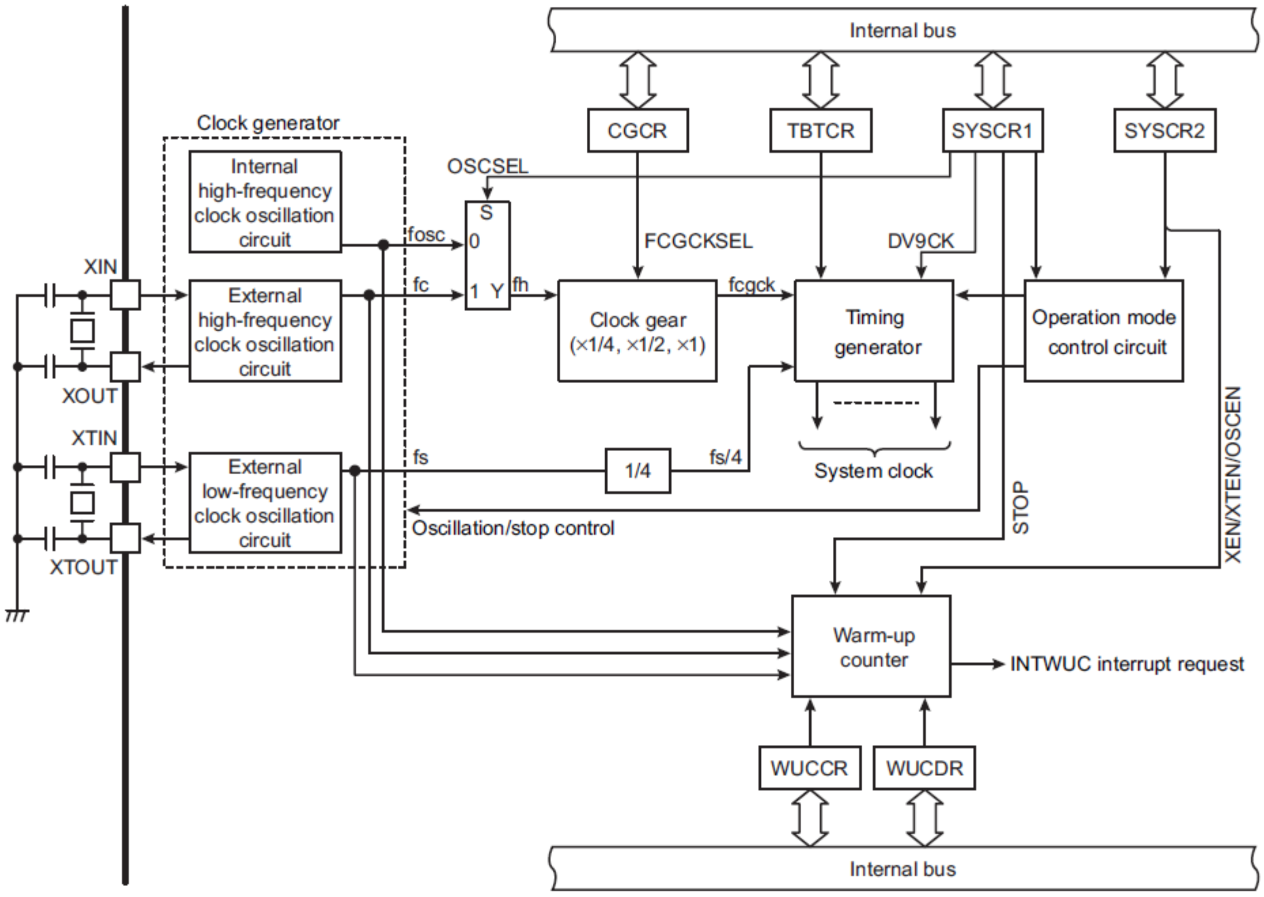
Note): If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

6. System Clock Controller

6.1 Configuration

The system clock controller consists of a clock generator, a clock gear, a timing generator, a warm-up counter and an operation mode control circuit.



Note 1): It's unnecessary to add extra capacitor in circuit when using external crystal because there is an internal RC in MCU.

Note 2): The location of external crystal is recommended to be as close to MCU as possible.

Figure 6.1 System Clock Controller

6.2 Control

The system clock controller is controlled by system control register 1 (SYSCR1), system control register 2 (SYSCR2), the warm-up counter control register (WUCCR), the warm-up counter data register (WUCDR) and the clock gear control register (CGCR).

System Control Register 1

SYSCR1 (0x0FDC)	7	6	5	4	3	2	1	0
Bit Symbol	STOP	RELM	OUTEN	DV9CK	OSCSEL	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0-

STOP	Activate the STOP mode	0 1	Operate the CPU and the peripheral circuit Stop the CPU and the peripheral circuit (activate the STOP mode)
RELM	Select the STOP mode release method	0 1	Edge-sensitive release mode (Release the STOP mode at the rising edge of the STOP mode release signal) Level-sensitive release mode (Release the STOP mode at the "H"level of the STOP mode release signal)
OUTEN	Select the port output state in the STOP mode	0 1	High impedance Output hold
DV9CK	Select the input clock to stage 9 of the divider	0 1	fcgck/2^9 fs/4
OSCSEL	Select the high-frequency reference clock (fh)	0 1	Internal high-frequency clock (fosc) External high-frequency clock (fc)

Note 1):fosc: Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], fcgck: Gear clock [Hz], fs: External low-frequency clock [Hz]

Note 2): Bits 2, 1 and 0 of SYSCR1 are read as "0".

Note 3): If the STOP mode is activated with SYSCR1 <OUTEN> set at "0", the port internal input is fixed to "0". Therefore, an external interrupt may be set at the falling edge, depending on the pin state when the STOP mode is activated.

Note 4): The P11 pin is also used as the STOP pin. When the STOP mode is activated, the pin reverts to high impedance state and is put in input mode, regardless of the state of SYSCR1<OUTEN>.

Note 5): Writing of the second byte data will be executed improperly if the operation is switched to the STOP state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 6): Don't set SYSCK1 <DV9CK> to "1" before the oscillation of the external low-frequency clock oscillation circuit becomes stable.

Note 7): In the SLOW1/2 or SLEEP1 mode, fs/4 is input to stage 9 of the divider, regardless of the state of SYSCR1 < DV9CK >.

Note 8): SYSCR1 <OSCSEL> should be set while SYSCR2 <SYSCK> is "0" (during the NORMAL1 or NORMAL2 mode). Writing to SYSCR1 <OSCSEL> while SYSCR2 <SYSCK> = "1" (during the SLOW1 or SLOW2 mode) has no effect.

System Control Register 2

SYSCR2 (0x0FDD)	7	6	5	4	3	2	1	0
Bit Symbol	OSCEN	XEN	XTEN	SYSCK	IDLE	TGHALT	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R
After reset	1	0	0	0	0	0	0	0

OSCEN	Control internal high-freq. clock (fosc)	0 1	Disable internal high-frequency clock oscillation circuit Enable internal high-frequency clock oscillation circuit
XEN	Control the external high-freq. clock (fc)	0 1	Stop oscillation Continue or start oscillation
XTEN	Control the external low-freq. clock (fs)	0 1	Stop oscillation Continue or start oscillation
SYSCK	Select a system clock	0 1	Gear clock (fcgck) (NORMAL1/2 or IDLE1/2 mode) Low-frequency clock (fs/4) (SLOW1/2 or SLEEP1 mode)
IDLE	CPU and WDT control (IDLE1/2 or SLEEP1 mode)	0 1	Operate the CPU and the WDT Stop the CPU and the WDT (Activate IDLE1/2 or SLEEP1 mode)
TGHALT	TG control (IDLE0 or SLEEP0 mode)	0 1	Enable the clock supply from the TG to all the peripheral circuits Disable the clock supply from the TG to the peripheral circuits except the TBT(Activate IDLE0 or SLEEP0 mode)

Note 1): fosc: Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], fcgck: Gear clock [Hz], fs: External low-frequency clock [Hz]

Note 2): WDT: Watchdog timer, TG: Timing generator

Note 3): Don't set both SYSCR2 <IDLE> and SYSCR2 <TGHALT> to "1" simultaneously.

Note 4): Writing of the second byte data will be executed improperly if the operation is switched to the IDLE state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 5): When the IDLE1/2 or SLEEP1 mode is released, SYSCR2 <IDLE> is cleared to "0" automatically.

Note 6): When the IDLE0 or SLEEP0 mode is released, SYSCR2 <TGHALT> is cleared to "0" automatically.

Note 7): Bits 1 and 0 of SYSCR2 are read as "0".

Note 8): Do not set both SYSCR2<OSCEN> and SYSCR2<XEN> to "1" simultaneously except when switching the high-frequency reference clock (fh). (When the switching of the reference clock (fh) is complete, one of the two high-frequency clocks not to be used should be stopped.)

Warm-up Counter Control Register

WUCCR (0x0FCD)	7	6	5	4	3	2	1	0
Bit Symbol	WUCRST	-	-	-	WUCDIV		WUCSEL	
Read/Write	W	R	R	R	R/W		R/W	
After reset	0	0	0	0	1	1	0	0

WUCRST	Reset and stop the warm-up counter	0 1	- Clear and stop the counter
WUCDIV	Select the frequency division of the warm-up counter source clock	00 01 10 11	Source clock Source clock/2 Source clock/2^2 Source clock/2^3
WUCSEL	Select the warm-up counter source clock	00 01 10 11	Select the internal high-frequency clock (fosc) Select the external high-frequency clock (fc) Select the external low-frequency clock (fs) Reserved

Note 1): fosc: Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], fcgck: Gear clock [Hz], fs: External low-frequency clock [Hz]

Note 2): WUCCR <WUCRST> is cleared to "0" automatically, and need not be cleared to "0" after bring set to "1".

Note 3): Bits 7 to 4 of WUCCR are read as "0".

Note 4): Before starting the warm-up counter operation, set the source clock and the frequency division rate at WUCCR and set the warm-up time at WUCDR.

Warm-up Counter Data Register

WUCDR (0x0FCE)	7	6	5	4	3	2	1	0
Bit Symbol	WUCDR							
Read/Write	R/W							
After reset	0	1	1	0	0	1	1	0

WUCDR	Warm-up time setting

Note): Don't start the warm-up counter operation with WUCDR set at "0x00".

Clock Gear Control Register

CGCR (0x0FCF)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	FCGCKSEL	
Read/Write	R	R	R	R	R	R	R/W	
After reset	0	0	0	0	0	0	0	0

FCGCKSEL	Clock gear setting	00
		01
		10
		11
			fcgck = fh / 4
			fcgck = fh / 2
			fcgck = fh
			Reserved

Note 1):fh: High-frequency reference clock [Hz], fcgck: Gear clock [Hz]

Note 2): Don't change CGCR <FCGCKSEL> in the SLOW mode.

Note 3): Bits 7 to 2 of CGCR are read as "0".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

6.3 Function

6.3.1 Clock Generator

The clock generator generates the basic clock for the system clocks to be supplied to the CPU core and peripheral circuits. It contains two oscillation circuits: one for the high-frequency clock and the other for the low-frequency clock.

The oscillation circuit pins are also used as ports P0. For the setting to use them as ports, refer to the chapter of I/O Ports. To use ports P00 and P01 as the high-frequency clock oscillation circuits (the XIN and XOUT pins), set P0FC0 to "1" and then set SYSCR2 <XEN> to "1".To use ports P02 and P03 for the external low-frequency clock oscillation circuit (as the XTIN and XTOUT pins), set P0FC2 to "1" and then set SYSCR2<XTEN> to "1".

The external high-frequency (fc) clock and the external low-frequency (fs) clock can easily be obtained by connecting an oscillator between the XIN and XOUT pins. Enabling/disabling the oscillation of the external high-frequency clock oscillation circuit and the external low-frequency clock oscillation circuit and switching the pin function to ports are controlled by the software and hardware.

The software control is executed by SYSCR2 <XEN>, SYSCR2 <XTEN> and the P0 port function control register P0FC. The hardware control is executed by reset release and the operation mode control circuit when the operation is switched to the STOP mode as described in "4.3.1 Operation Mode Control Circuit".

Note): No hardware function is available for external direct monitoring of the basic clock. The oscillation frequency can be adjusted by programming the system to output pulses at a certain frequency to a port (for example, a clock output) with interrupts disabled and the watchdog timer disabled and monitoring the output. An adjustment program must be created in advance for a system that requires adjustment of the oscillation frequency.

To prevent the dead lock of the CPU core due to the software-controlled enabling/disabling of the oscillation, an internal factor reset is generated depending on the combination of values of the clock selected as the main system clock, SYSCR2 <XEN>, SYSCR2 <XTEN> and the P0 port function control register P0FC0.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

P0FC0	SYSCR2 <XEN>	SYSCR2 <XTEN>	SYSCR2 <SYSCK>	State
Don't Care	0	0	Don't Care	All the oscillation circuits are stopped.
Don't Care	Don't Care	0	1	The low-frequency clock (fs) is selected as the main system clock, but the low-frequency clock oscillation circuit is stopped.
Don't Care	0	Don't Care	0	The high-frequency clock (fc) is selected as the main system clock, but the high-frequency clock oscillation circuit is stopped.
0	1	Don't Care	Don't Care	The high-frequency clock oscillation circuit is allowed to oscillate, but the port is set as a general-purpose port.

Table 6.1 Prohibited Combinations of Oscillation Enable Register Conditions

Note): It takes a certain period of time after SYSCR2 <SYSCK> is changed before the main system clock is switched. If the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 6.1 and a system clock reset occurs. For details of clock switching, refer to "4.3.2 Operation Mode Control".

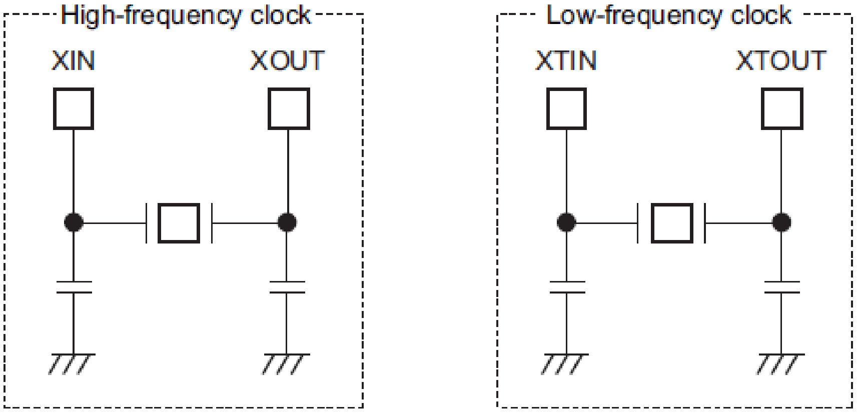


Figure 6.2Examples of Oscillator Connection

Note 1)：The appropriate oscillation circuit and providing proper capacitance are necessary for stable clock, these are highly correlated with the circuit board. System must be confirmed stable after all the components on the board is mounted.

Note 2)：XIN/XOUT pin has build-in capacitance (6pF each). Load capacitance can be designed according to character of oscillator, accuracy of clock and design of circuit board

(1) High-frequency reference clock (fh)

The high-frequency reference clock (fh) is used to operate the MCU at high speed. When SYSCR1<OSCSEL> = "1", the external high-frequency clock (fc) is used as the reference clock (fh). When SYSCR1<OSCSEL> = "0", the internal high-frequency clock (fosc) is used as the reference clock (fh). Upon reset release, SYSCR1<OSCSEL> is cleared to "0" and the internal high frequency clock (fosc) is used as the reference clock (fh).

When the high-frequency reference clock (fh) is switched, both the external high-frequency clock (fc) and the internal high-frequency clock (fosc) need to be oscillating. To switch the high-frequency reference clock (fh), be sure to follow the steps explained below.

In the process of switching the high-frequency reference clock (fh), there is a time when both the external high-frequency clock (fc) and the internal high-frequency clock (fosc) are enabled simultaneously. Mode transitions, as explained in "4.3.1 Operation mode control circuit", must not be made in this state. Once the reference clock has been switched, be sure to stop either of the high-frequency clocks not to be used.

・Switching from fosc to fc

With the relevant bits in the P0FC0 register set to "1", set SYSCR2<XEN> to "1" to enable the external high-frequency clock (fc).　After making sure that the external high-frequency clock (fc) has achieved stable oscillation by using the warm-up counter, set SYSCR1<OSCSEL> to "1".

A maximum of 2/fosc + 2.5/fc [s] after SYSCR1<OSCSEL> is set to "1", the high-frequency reference clock (fh) changes to the external high-frequency clock (fc). After the reference clock (fh) has been switched, wait for at least 2 machine cycles, and then clear SYSCR2<OSCEN> to "0" to stop the internal high-frequency clock (fosc). If SYSCR2<OSCEN> is cleared to "0" while the reference clock (fh) is being switched, a system clock reset is generated.

Note 1: When the high-frequency reference clock (fh) is switched, the hardware synchronizes the external high-frequency clock (fc) and the internal high-frequency clock (fosc). While this is done, fh stops for a maximum of 2.5/fc [s].

Note 2: After changing SYSCR1<OSCSEL>, be sure to wait for at least 2 machine cycles before clearing SYSCR2<OSCEN> to "0". If SYSCR2<OSCEN> is cleared to "0" without waiting for at least 2 machine cycles, a system clock reset is generated.

Note 3: SYSCR1<OSCSEL> must be set while SYSCR2<SYSCK> = "0" (during the NORMAL1 or NORMAL2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> = "1" (during the SLOW1 or SLOW2 mode) has no effect.

Note 4: Setting SYSCR2<XEN> to "1" while P0FC0 = "0" generates a system clock reset.

Note 5: If SYSCR2<XEN> is set to "1" while SYSCR2<XEN> = "1", the warm-up counter does not start counting the source clock.

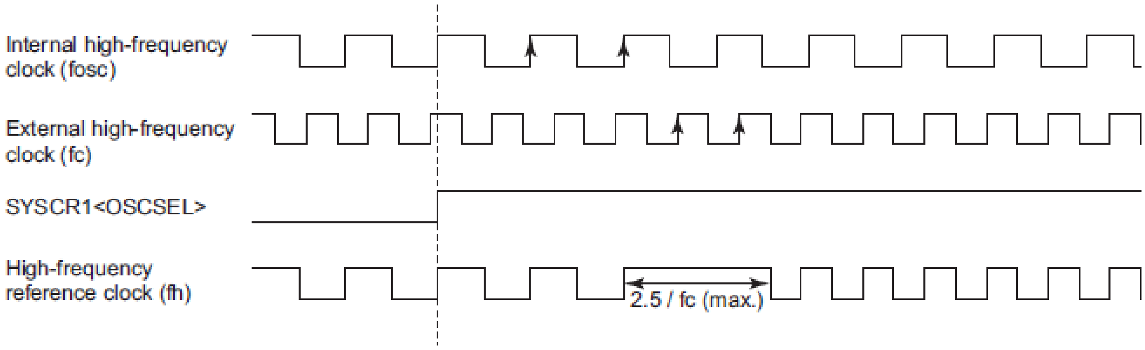

Internal high-frequency clock (fosc)
External high-frequency clock (fc)
SYSCR1<OSCSEL>
High-frequency reference clock (fh)
2.5 / fc (max.)

Figure 6.3 Switching the High-Frequency Reference Clock (fh) (fosc->fc)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Step	P0FC0	SYSCR2 <OSCEN>	SYSCR2 <XEN>	SYSCR1 <OSCSEL>	Main system clock	State
1	0	1	0	0	fosc	The high-frequency reference clock is fosc, and ports P00 and P01 are used as I/O ports.
2	1	1	0	0	fosc	Ports P00 and P01 are set as oscillation pins.
3	1	1	1	0	fosc	The high-frequency clock oscillation circuit is warming up.
4	1	1	1	1	fosc →fc	The high-frequency reference clock is being switched to fc.
5	1	0	1	1	fc	The high-frequency reference clock has been switched to fc.

Table 6.2 Steps for Switching the High-Frequency Reference Clock (fh) from fosc to fc

Note:Be sure to follow the above steps when switching the high-frequency reference clock.

Example: Setting ports P00 and P01 as oscillation pins and switching the high-frequency reference clock from fosc to fc (warm-up time: approx. 300 µs at fc = 8 MHz)

```
LD      (WUCCR), 0y00000001        ;WUCCR<WUCDIV>←"00" (No division)
                                   ;WUCCR<WUCSEL>←"01" (Selects fc as the source clock)
LD      (WUCDR), 0x26              ;Sets the warm-up time
                                   ;(Determine the time depending on the oscillator
                                   characteristics)
                                   ;300µs / 8 µs =37.5 round up to 0x26
SET     (EIRL).4                   ;Enables INTWUC interrupts
SET     (P0FC).0                   ;P0FC0←"1" (Set P00 and P01 as oscillation pins)
SET     (SYSCR2). 6                ;SYSCR2<XTEN>←"1"
                                   ;(Starts the external high-frequency clock
                                   oscillation and starts the ;warm-up counter)
SET     (SYSCR1). 3                ;SYSCR1<OSCSEL>←"1"
                                   ;(Switches the high-frequency reference clock from
                                   fosc to fc)
NOP                                ;Waits for 2 machine cycles
NOP                                ;Waits for 2 machine cycles
CLR     (SYSCR2).7                 ;SYSCR2<OSCEN>←"0" (Stops fosc)
RETI
```

・Switching from fc to fosc

Set SYSCR1<OSCEN> to "1" to enable the internal high-frequency clock (fosc). After making sure that the internal high-frequency clock (fosc) has achieved stable oscillation by using the warm-up counter, clear SYSCR1<OSCSEL> to "0".

A maximum of 2/fc + 2.5/fosc [s] after SYSCR1<OSCSEL> is cleared to "0", the high-frequency reference clock (fh) changes to the internal high-frequency clock (fosc). After the reference clock (fh) has been switched, wait for at least 2 machine cycles, and then clear SYSCR2<XEN> to "0" to stop the external high-frequency clock (fc). If SYSCR2<XEN> is cleared to "0" while the reference clock (fh) is being switched, a system clock reset is generated.

Note 1: When the high-frequency reference clock (fh) is switched, the hardware synchronizes the external high-frequency clock (fc) and the internal high-frequency clock (fosc). While this is done, fh stops for a maximum of 2.5/fosc [s].

Note 2: After changing SYSCR1<OSCSEL>, be sure to wait for at least 2 machine cycles before clearing SYSCR2<XEN> to "0". If SYSCR2<XEN> is cleared to "0" without waiting for at least 2 machine cycles, a system clock reset is generated.

Note 3: SYSCR1<OSCSEL> must be set while SYSCR2<SYSCK> = "0" (during the NORMAL1 or NORMAL2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> = "1" (during the SLOW1 or SLOW2 mode) has no effect.

Note 4: Setting SYSCR2<XEN> to "1" while P0FC0 = "0" generates a system clock reset.

Note 5: If SYSCR2<XEN> is set to "1" while SYSCR2<XEN> = "1", the warm-up counter does not start counting the source clock.

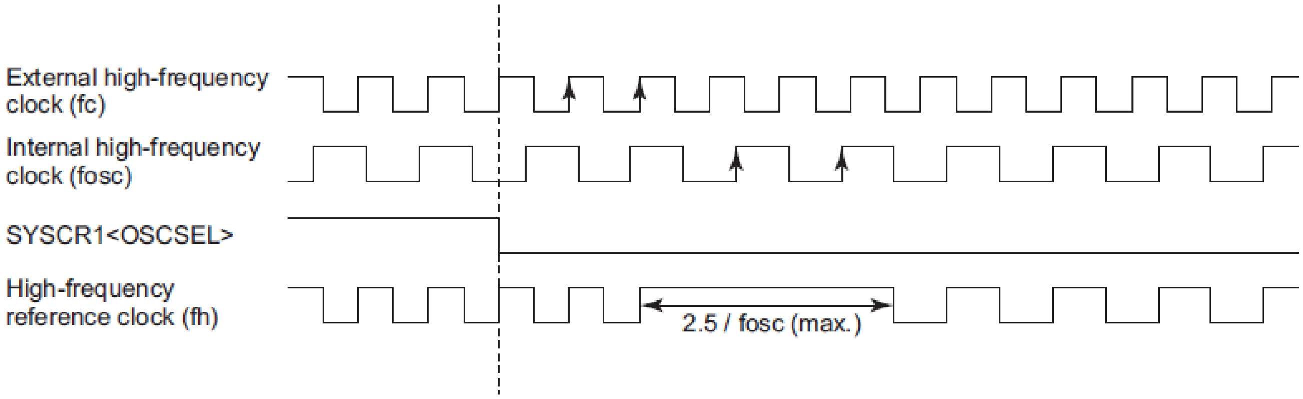

External high-frequency clock (fc)
Internal high-frequency clock (fosc)
SYSCR1<OSCSEL>
High-frequency reference clock (fh)
2.5 / fosc (max.)

Figure 6.4 Switching the High-Frequency Reference Clock (fh) (fc->fosc)

Step	P0FC0	SYSCR2 <OSCEN>	SYSCR2 <XEN>	SYSCR1 <OSCSEL>	Main system clock	State
1	0	1	0	0	fosc	The high-frequency reference clock is fosc, and ports P00 and P01 are used as I/O ports.
2	1	1	0	0	fosc	Ports P00 and P01 are set as oscillation pins.
3	1	1	1	0	fosc	The high-frequency clock oscillation circuit is warming up.
4	1	1	1	1	fosc →fc	The high-frequency reference clock is being switched to fc.

Table 6.3 Steps for Switching the High-Frequency Reference Clock (fh) from fc to fosc

Note:Be sure to follow the above steps when switching the high-frequency reference clock.

Example: Switching the high-frequency reference clock from fc to fosc (warm-up time: approx. 100 μs = at fosc = 5 MHz)

```
LD      (WUCCR), 0y00000000       ;WUCCR<WUCDIV>←"00" (No division)
                                  ;WUCCR<WUCSEL>←"01" (Selects fc as the source clock)
LD      (WUCDR), 0x08             ;Sets the warm-up time
                                  ;(Determine the time depending on the oscillator
                                  ; characteristics)
                                  ;100µs / 12.8 µs = 7.8 round up to 0x08
SET     (EIRL).4                  ;Enables INTWUC interrupts
SET     (SYSCR2). 7
SET     (SYSCR2). 6               ;SYSCR2<OSCEN>←"1"
                                  ;(Starts the internal high-frequency clock oscillation
                                  ;and ;starts the warm-up counter)
SET     (SYSCR1). 3               SYSCR1<OSCSEL>←"0"
                                  (Switches the high-frequency reference clock from fc to
                                  fosc)
NOP                               ;Waits for 2 machine cycles
NOP                               ;Waits for 2 machine cycles
CLR     (SYSCR2).6                ;SYSCR2<OSCEN>←"0" (Stops fosc)
RETI
```

(2) Low-frequency reference clock (fs)
The low-frequency reference clock (fs) is used to operate the MCU at low speed. Power consumption can be reduced.

6.3.2 Clock Gear

The clock gear is a circuit that selects a gear clock (fcgck) obtained by dividing the high-frequency reference clock (fh) and inputs it to the timing generator. Select a divided clock at CGCR <FCGCKSEL>. Two machine cycles are needed after CGCR <FCGCKSEL> is changed before the gear clock (fcgck) is changed.

CGCR <FCGCKSEL>	fcgck
00	fh / 4
01	fh / 2
10	fh
11	Reserved

Table 6.4 Gear Clock (fcgck)

Note): Don't change CGCR <FCGCKSEL> in the SLOW mode. This may stop the gear clock (fcgck) from being changed.

The gear clock (fcgck) may be longer than the set clock width, immediately after CGCR <FCGCKSEL>

is changed. Immediately after reset release, the gear clock (fcgck) becomes the clock that is a quarter of the high-frequency reference clock (fh).

6.3.3 Timing Generator

The timing generator is a circuit that generates system clocks to be supplied to the CPU core and the peripheral circuits, from the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs). The timing generator has the following functions:

1. Generation of the main system clock (fm)
2. Generation of clocks for the timer counter, the time base timer and other peripheral circuits

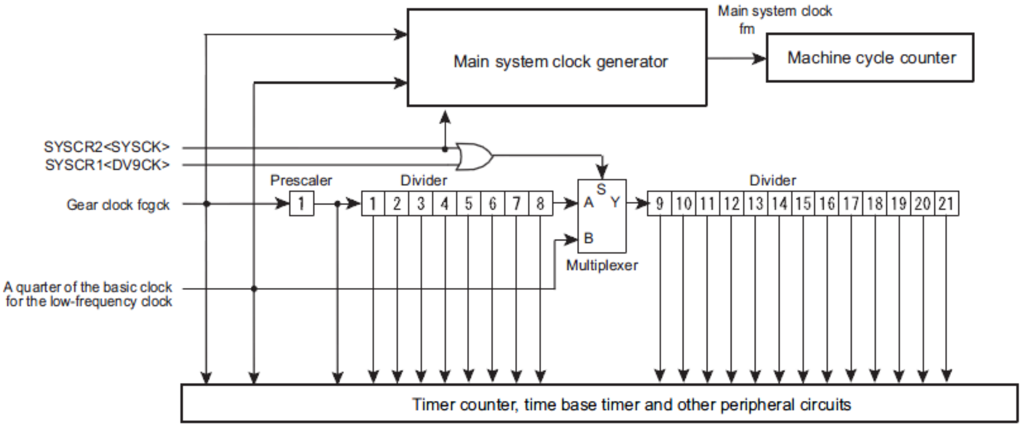


Figure 6.5 Configuration of Timing Generator

The timing generator consists of a main system clock generator, a prescaler, a 21-stage divider and a machine cycle counter.

6.3.3.1 Main System Clock Generator

This circuit selects the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs) for the main system clock (fm) to operate the CPU core.

Clearing SYSCR2 <SYSCK> to "0" selects the gear clock (fcgck). Setting it to "1" selects the clock that is a quarter of the low-frequency clock (fs). It takes a certain period of time after SYSCR2<SYSCK> is changed before the main system clock is switched. If the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 6.1 and a system clock reset occurs.

6.3.3.2 Prescaler and Divider

These circuits divide fcgck. The divided clocks are supplied to the timer counter, the time base timer and other peripheral circuits.

When both SYSCR1 <DV9CK> and SYSCR2 <SYSCK> are "0", the input clock to stage 9 of the divider becomes the output of stage 8 of the divider.

When SYSCR1 <DV9CK> or SYSCR2 <SYSCK> is "1", the input clock to stage 9 of the divider becomes fs/4. When SYSCR2 <SYSCK> is "1", the outputs of stages 1 to 8 of the divider and prescaler are stopped.

The prescaler and divider are cleared to "0" at a reset and at the end of the warm-up operation that follows the release of STOP mode.

6.3.3.3 Machine Cycle

Instruction execution is synchronized with the main system clock (fm).

The minimum instruction execution unit is called a "machine cycle". One machine cycle corresponds to one main system clock.

There are a total of 11 different types of instructions for the i87 Series: 10 types ranging from 1-cycle instructions, which require one machine cycle for execution, to 10-cycle instructions, which require 10 machine cycles for execution, and 13-cycle instructions, which require 13 machine cycles for execution.

6.4 Warm-up Counter

The warm-up counter is a circuit that counts the internal high-frequency clock (fosc), the external high-frequency clock (fc) and the external low-frequency clock (fs), and it consists of a source clock selection circuit, a 3-stage frequency division circuit and a 14-stage counter.

The warm-up counter is used to secure the time after a power-on reset is released before the supply voltage becomes stable and secure the time after the STOP mode is released or the operation mode is changed before the oscillation by the oscillation circuit becomes stable.

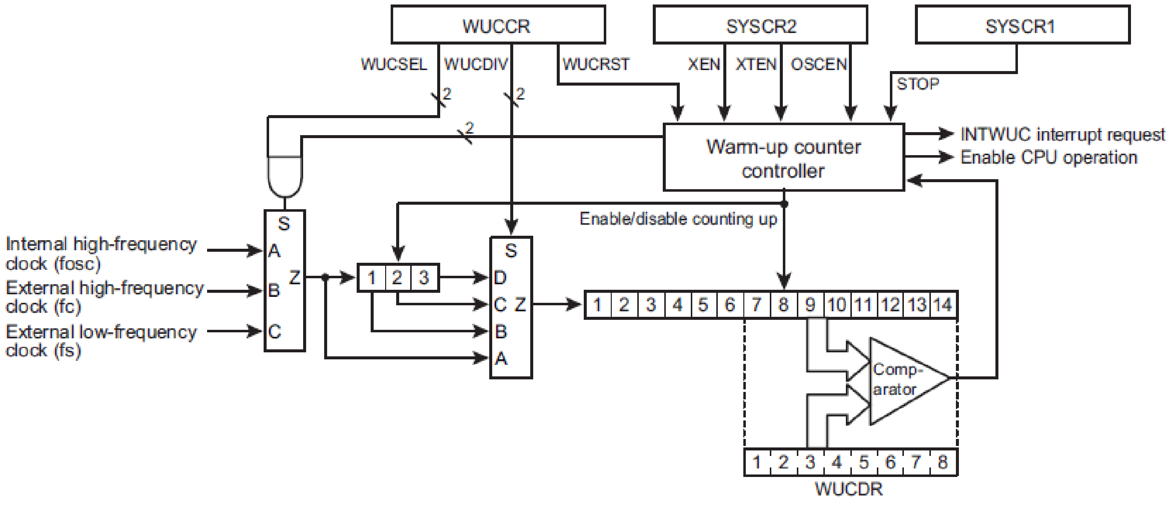


Figure 6.6 Warm-up Counter Circuit

6.4.1 Warm-up Counter Operation When the Oscillation is Enabled by Hardware

6.4.1.1When a Power-on Reset is Released or a Reset is Released

The warm-up counter serves to secure the time after a power-on reset is released before the supply voltage becomes stable and the time after a reset is released before the oscillation by the high-frequency clock oscillation circuit becomes stable.

When the power is turned on and the supply voltage exceeds the power-on reset release voltage, the warm-up counter reset signal is released. At this time, the CPU and the peripheral circuits are held in the reset state.

A reset signal initializes WUCCR <WUCSEL> to "0" and WUCCR <WUCDIV> to "11", which selects the high-frequency clock (fc) as the input clock to the warm-up counter.

When a reset is released for the warm-up counter, the internal high-frequency clock (fosc) is input to the warm-up counter, and the 14-stage counter starts counting the internal high-frequency clock (fosc).

When the upper 8 bits of the warm-up counter become equal to WUCDR, counting is stopped and a reset is released for the CPU and the peripheral circuits.

WUCDR is initialized to 128 (decimal value) after reset release, which makes the warm-up time $102 \times 2^{9}/fc$ [s].

Note): The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable.

6.4.1.2 When the STOP Mode is Released

The warm-up counter serves to secure the time after the oscillation is enabled by the hardware before the oscillation becomes stable at the release of the STOP mode.

The high-frequency clock (fc) or the low-frequency clock (fs), which generates the main system clock when the STOP mode is activated, is selected as the input clock for frequency division circuit, regardless of WUCCR <WUCSEL>.Before the STOP mode is activated, select the division rate of the input clock to the warm-up counter at WUCCR <WUCDIV> and set the warm-up time at WUCDR.

When the STOP mode is released, the 14-stage counter starts counting the input clock selected in the frequency division circuit.

When the upper 8 bits of the warm-up counter become equal to WUCDR, counting is stopped and the operation is restarted by an instruction that follows the STOP mode activation instruction.

Clock that generated the main system clock when the STOP mode was activated	WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
fosc	Don't Care	00	fosc	2^6 / fosc to 255 × 2^6 / fosc
		01	fosc / 2	2^7 / fosc to 255 × 2^7 / fosc
		10	fosc / 2^2	2^8 / fosc to 255 × 2^8 / fosc
		11	fosc / 2^3	2^9 / fosc to 255 × 2^9 / fosc
fc	Don't Care	00	fc	2^6 / fc to 255 × 2^6 / fc
		01	fc / 2	2^7 / fc to 255 × 2^6 / fc
		10	fc / 2^2	2^8 / fc to 255 × 2^8 / fc
		11	fc / 2^3	2^9 / fc to 255 × 2^9 / fc
fs	Don't Care	00	fs	2^6 / fs to 255 × 2^6 / fs
		01	fs / 2	2^7 / fs to 255 × 2^7 / fs
		10	fs / 2^2	2^8 / fs to 255 × 2^8 / fs
		11	fs / 2^3	2^9 / fs to 255 × 2^9 / fs

Note 1): When the operation is switched to the STOP mode during the warm-up for the oscillation enabled by the software, the warm-up counter holds the value at the time and restarts counting after the STOP mode is released. In this case, the warm-up time at the release of the STOP mode becomes insufficient. Don't switch the operation to the STOP mode during the warm-up for the oscillation enabled by the software.

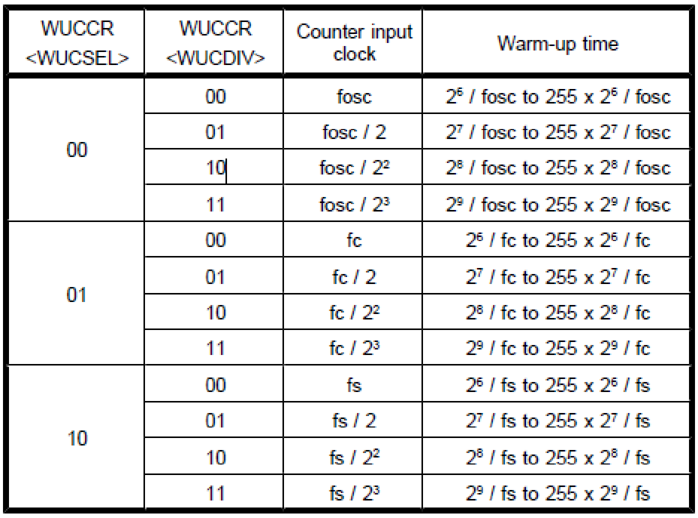
Note 2): The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

6.4.2 Warm-up Counter Operation When the Oscillation is Enabled by Software

The warm-up counter serves to secure the time after the oscillation is enabled by the software before the oscillation becomes stable, at a mode change from NORMAL1 to NORMAL2 or from SLOW1 to SLOW2.Select the input clock to the frequency division circuit at WUCCR <WUCSEL>. Select the input clock to the 14-stage counter at WUCCR <WUCDIV>.

After the warm-up time is set at WUCDR, setting SYSCR2 <OSCEN>, SYSCR2 <XEN> or SYSCR2 <XTEN> to "1" allows the stopped oscillation circuit to start oscillation and the 14-stage counter to start counting the selected input clock. When the upper 8 bits of the counter become equal to WUCDR, an INTWUC interrupt occurs, counting is stopped and the counter is cleared.

Set WUCCR <WUCRST> to "1" to discontinue the warm-up operation. By setting it to "1", the count-up operation is stopped, the warm-up counter is cleared, and WUCCR <WUCRST> is cleared to "0". SYSCR2 <OSCEN>, SYSCR2<XEN> and SYSCR2 <XTEN> hold the values when WUCCR <WUCRST> is set to "1". To restart the warm-up operation, SYSCR2 <OSCEN>, SYSCR2 <XEN> or SYSCR2<XTEN> must be cleared to "0".



WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
00	00	fosc	2^6 / fosc to 255 x 2^6 / fosc
	01	fosc / 2	2^7 / fosc to 255 x 2^7 / fosc
	10	fosc / 2^2	2^8 / fosc to 255 x 2^8 / fosc
	11	fosc / 2^3	2^9 / fosc to 255 x 2^9 / fosc
01	00	fc	2^6 / fc to 255 x 2^6 / fc
	01	fc / 2	2^7 / fc to 255 x 2^7 / fc
	10	fc / 2^2	2^8 / fc to 255 x 2^8 / fc
	11	fc / 2^3	2^9 / fc to 255 x 2^9 / fc
10	00	fs	2^6 / fs to 255 x 2^6 / fs
	01	fs / 2	2^7 / fs to 255 x 2^7 / fs
	10	fs / 2^2	2^8 / fs to 255 x 2^8 / fs
	11	fs / 2^3	2^9 / fs to 255 x 2^9 / fs

Note 1): The warm-up counter starts counting when SYSCR2 <OSCEN>, SYSCR2 <XEN> or SYSCR2 <XTEN>is changed from "0" to "1". The counter will not start counting by writing "1" to SYSCR2 <OSCEN>, SYSCR2 <XEN> or SYSCR2 <XTEN> when it is in the state of "1".

Note 2): The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

7 Interrupts

MQ6935 has a total of 30 interrupt sources excluding reset. Interrupts can be nested with priorities. Three of the internal interrupt sources are non-maskable while the rest are maskable. Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and have independent vector addresses. When a request for an interrupt is generated, its interrupt latch is set to "1", which requests the CPU to accept the interrupt. Acceptance of interrupts is enabled or disabled by software using the interrupt master enable flag (IMF) and individual enable flag (EF) for each interrupt source. If multiple maskable interrupts are generated simultaneously, the interrupts are accepted in order of descending priority, as Table 7.1. However, there are no prioritized interrupt sources among non-maskable interrupts.

Interrupt sources		Enable condition	Interrupt latch	Vector Address (MCU mode)		Basic priority
				RVCTR=0 enabled	RVCTR=1 enabled	
Internal/ External	(Reset)	Non-maskable	-	0xFFFE	-	1
Internal	INTSWI	Non-maskable	-	0xFFFC	0x01FC	2
Internal	INTUNDEF	Non-maskable	-	0xFFFC	0x01FC	2
Internal	INTWDT	Non-maskable	ILL<IL3>	0xFFF8	0x01F8	2
Internal	INTWUC	IMF AND EIRL<EF4> = 1	ILL<IL4>	0xFFF6	0x01F6	5
Internal	INTTBT	IMF AND EIRL<EF5> = 1	ILL<IL5>	0xFFF4	0x01F4	6
Internal	INTRXD0 / INTSIO0	IMF AND EIRL<EF6> = 1	ILL<IL6>	0xFFF2	0x01F2	7
Internal	INTTXD0	IMF AND EIRL<EF7> = 1	ILL<IL7>	0xFFF0	0x01F0	8
External	INT5B	IMF AND EIRH<EF8> = 1	ILH<IL8>	0xFFEE	0x01EE	9
Internal	INTVLTD	IMF AND EIRH<EF9> = 1	ILH<IL9>	0xFFEC	0x01EC	10
Internal	INTADC	IMF AND EIRH<EF10> = 1	ILH<IL10>	0xFFEA	0x01EA	11
Internal	INTRTC	IMF AND EIRH<EF11> = 1	ILH<IL11>	0xFFE8	0x01E8	12
Internal	INTTC00	IMF AND EIRH<EF12> = 1	ILH<IL12>	0xFFE6	0x01E6	13
Internal	INTTC01	IMF AND EIRH<EF13> = 1	ILH<IL13>	0xFFE4	0x01E4	14
Internal	INTTCA0	IMF AND EIRH<EF14> = 1	ILH<IL14>	0xFFE2	0x01E2	15
Internal	INTSBI0 / INTSIO0	IMF AND EIRH<EF15> = 1	ILH<IL15>	0xFFE0	0x01E0	16
External	INT0	IMF AND EIRE<EF16> = 1	ILE<IL16>	0xFFDE	0x01DE	17
External	INT1	IMF AND EIRE<EF17> = 1	ILE<IL17>	0xFFDC	0x01DC	18
External	INT2	IMF AND EIRE<EF18> = 1	ILE<IL18>	0xFFDA	0x01DA	19
External	INT3	IMF AND EIRE<EF19> = 1	ILE<IL19>	0xFFD8	0x01D8	20
External	INT4	IMF AND EIRE<EF20> = 1	ILE<IL20>	0xFFD6	0x01D6	21
Internal	INTTCA1	IMF AND EIRE<EF21> = 1	ILE<IL21>	0xFFD4	0x01D4	22
Internal	INTRXD1	IMF AND EIRE<EF22> = 1	ILE<IL22>	0xFFD2	0x01D2	23
Internal	INTTXD1	IMF AND EIRE<EF23> = 1	ILE<IL23>	0xFFD0	0x01D0	24
Internal	INTTC02	IMF AND EIRD<EF24> = 1	ILD<IL24>	0xFFCE	0x01CE	25
Internal	INTTC03	IMF AND EIRD<EF25> = 1	ILD<IL25>	0xFFCC	0x01CC	26
Internal	INTRXD2	IMF AND EIRD<EF26> = 1	ILD<IL26>	0xFFCA	0x01CA	27
Internal	INTTXD2	IMF AND EIRD<EF27> = 1	ILD<IL27>	0xFFC8	0x01C8	28
Internal	INTEMG0	IMF AND EIRD<EF28> = 1	ILD<IL28>	0xFFC6	0x01C6	29
Internal	INTTCC0P	IMF AND EIRD<EF29> = 1	ILD<IL29>	0xFFC4	0x01C4	20
Internal	INTTCC0T	IMF AND EIRD<EF30> = 1	ILD<IL30>	0xFFC2	0x01C2	31

Table 7.1 Interrupt Information Table

Note 1): To use the watchdog timer interrupt (INTWDT), clear WDTCR1<WDTOUT> to "0" (It is set for the "Reset request" after reset is released). For details, see "10.1 Watchdog Timer".

Note 2): Vector address areas can be changed by the SYSCR3<RVCTR> setting. To assign vector address areas to RAM, set SYSCR3<RVCTR> to "1", and also set SYSCR3<RAREA> to "1".

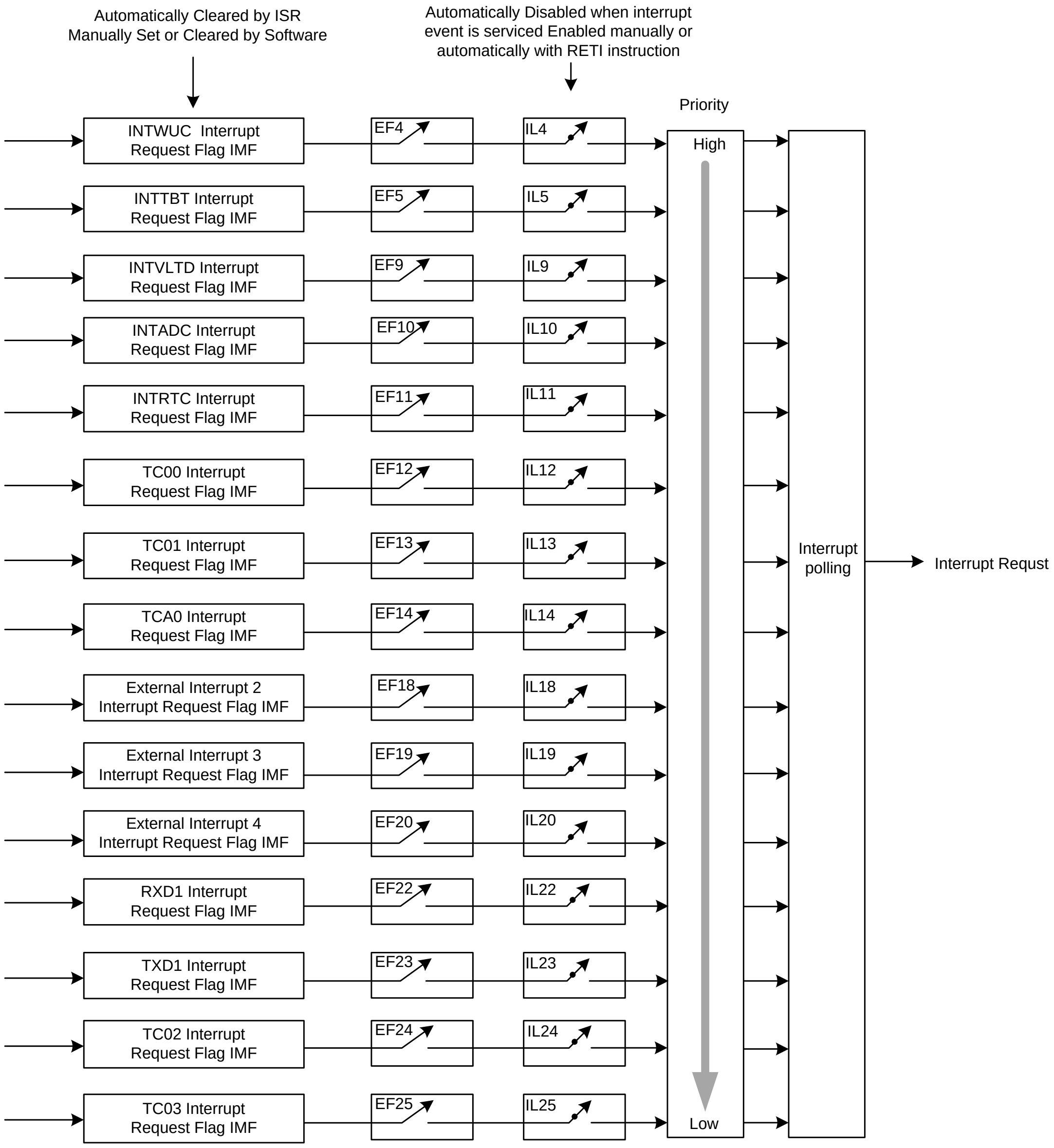

Automatically Cleared by ISR
Manually Set or Cleared by Software
Automatically Disabled when interrupt event is serviced Enabled manually or automatically with RETI instruction
Priority
INTWUC Interrupt Request Flag IMF
INTTBT Interrupt Request Flag IMF
INTVLTD Interrupt Request Flag IMF
INTADC Interrupt Request Flag IMF
INTRTC Interrupt Request Flag IMF
TC00 Interrupt Request Flag IMF
TC01 Interrupt Request Flag IMF
TCA0 Interrupt Request Flag IMF
External Interrupt 2 Interrupt Request Flag IMF
External Interrupt 3 Interrupt Request Flag IMF
External Interrupt 4 Interrupt Request Flag IMF
RXD1 Interrupt Request Flag IMF
TXD1 Interrupt Request Flag IMF
TC02 Interrupt Request Flag IMF
TC03 Interrupt Request Flag IMF
EF4
EF5
EF9
EF10
EF11
EF12
EF13
EF14
EF18
EF19
EF20
EF22
EF23
EF24
EF25
IL4
IL5
IL9
IL10
IL11
IL12
IL13
IL14
IL18
IL19
IL20
IL22
IL23
IL24
IL25
High
Low
Interrupt polling
Interrupt Requst

Figure7.1 Interrupts Control Scheme

7.1 Interrupts Latches (IL25 to IL3)

An interrupt latch is provided for each interrupt source, except for a software interrupt and an undefined instruction execution interrupt. When an interrupt request is generated, the latch is set to "1", and then the CPU is requested to accept the interrupt if its acceptance is enabled. The interrupt latch is cleared to "0" immediately after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

The interrupt latches are located at addresses 0x0FE0, 0x0FE1, 0x0FE2and 0xFE3 in SFR area. Each latch can be cleared to "0" individually by an instruction. However, IL3 interrupt latches cannot be cleared by instructions.

Do not use any read-modify-write instruction, such as a bit manipulation or operation instruction, because it may clear interrupt requests generated while the instruction is executed.

Interrupt latches cannot be set to "1" by using an instruction. Writing "1" to an interrupt latch is equivalent to deny clearing of the interrupt latch, and not setting the interrupt latch.

Since interrupt latches can be read by instructions, the status of interrupt requests can be monitored by software.

Note): In the main program, before manipulating an interrupt latch (IL), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the IL (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the IL before setting the IMF to "1".

7.2 Interrupt Enable Register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (software interrupt, undefined instruction interrupt and watchdog interrupt). Non-maskable interrupts are accepted regardless of the contents of the EIR.

The EIR consists of the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located at addresses 0x003A, 0x003B, 0x003C and 0x003D in the SFR area, and they can be read and written by instructions (including read-modify-write instructions such as bit manipulation or operation instructions).

7.3 Interrupt Master Enable Flag (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all maskable interrupts. Clearing the IMF to "0" disables the acceptance of all maskable interrupts. Setting the IMF to "1" enables the acceptance of the interrupts that are specified by the individual interrupt enable flags.

When an interrupt is accepted, the IMF is stacked and then cleared to "0", which temporarily disables the subsequent maskable interrupts. After the interrupt service routine is executed, the stacked data, which was the status before interrupt acceptance, reloads on the IMF by return interrupt instruction [RETI] / [RETN].

The IMF is located on bit 0 in EIRL (Address: 0x003A in SFR), and can be read and written by instructions. The IMF is normally set and cleared by [EI] and [DI] instructions respectively. During reset, the IMF is initialized to "0".

7.4 Individual Interrupt Enable Flag (EF30 to EF4)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of its interrupt, and setting the bit to "0" disables acceptance.

During reset, all the individual interrupt enable flags are initialized to "0" and no maskable interrupts are accepted until the flags are set to "1".

Note): In the main program, before manipulating the interrupt enable flag (EF), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the EF (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the EF before setting the IMF to "1".

Interrupt Latch (ILL)

ILL (0x0FE0)	7	6	5	4	3	2	1	0
Bit Symbol	IL7	IL6	IL5	IL4	IL3	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
After reset	0	0	0	0	0	0	0	0
Function	INTTXD0	INTRXD0 /INTSIO0	INTTBT	INTWUC	INTWDT	-	-	-

Interrupt Latch (ILH)

ILH (0x0FE1)	7	6	5	4	3	2	1	0
Bit Symbol	IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	INTSBI0/ INTSIO0	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTLVD	INT5

Interrupt Latch (ILE)

ILE (0x0FE2)	7	6	5	4	3	2	1	0
Bit Symbol	IL23	IL22	IL21	IL20	IL19	IL18	IL17	IL16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	INTTXD1	INTRXD1	INTTCA1	INT4	INT3	INT2	INT1	INT0

Interrupt Latch (ILD)

ILD (0x0FE3)	7	6	5	4	3	2	1	0
Bit Symbol	-	IL30	IL29	IL28	IL27	IL26	IL25	IL24
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	-	INTTCC0T	INTTCC0P	INTEMG0	INTTXD2	INTRXD2	INTTC03	INTTC02

			Read	Write
IL30 to IL4	Interrupt latch	0	No interrupt request	Clear the interrupt request (Notes 2 and 3)
		1	Interrupt request	Does not clear the interrupt request (Interrupt is not set by writing "1")
IL3		0 1	No interrupt request Interrupt request	-

Note 1):IL3 is a read-only register. Writing the register does not affect interrupt latch.

Note 2): In the main program, before manipulating an interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to"0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the IL (Enable interrupt by EI instruction).

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, ifusing multiple interrupt in the interrupt service routine, manipulate the IL before setting the IMF to "1".

Note 3): Do not clear IL with read-modify-write instructions such as bit operations.

Note 4): When a read instruction is executed on ILL, bits 0 to 2 are read as "0". Other unused bits are read as "0".

Interrupt Enable Register (EIRL)

EIRL (0x003A)	7	6	5	4	3	2	1	0
Bit Symbol	EF7	EF6	EF5	EF4	-	-	-	IMF
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R/W
After reset	0	0	0	0	0	0	0	0
Function	INTTXD0	INTRXD0 /INTSIO0	INTTBT	INTWUC	-	-	-	IMF

Interrupt Enable Register (EIRH)

EIRH (0x003B)	7	6	5	4	3	2	1	0
Bit Symbol	EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	INTSBI0 / INTSIO0	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTLVD	INT5

Interrupt Enable Register (EIRE)

EIRE (0x003C)	7	6	5	4	3	2	1	0
Bit Symbol	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	INTTXD1	INTRXD1	INTTCA1	INT4	INT3	INT2	INT1	INT0

Interrupt Enable Register (EIRD)

EIRD (0x003D)	7	6	5	4	3	2	1	0
Bit Symbol	-	EF30	EF29	EF28	EF27	EF26	EF25	EF24
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	-	INTTCC0T	INTTCC0P	INTEMG0	INTTXD2	INTRXD2	INTTC03	INTTC02

EF30 to EF4	Individual Interrupt Enable Flag (specified for each bit)	0 1	Disable the acceptance of each maskable interrupt Enable the acceptance of each maskable interrupt
IMF	Interrupt Master Enable Flag	0 1	Disable the acceptance of all maskable interrupts Enable the acceptance of all maskable interrupts

Note 1): Do not set the IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.

Note 2): In the main program, before manipulating the interrupt enable flag (EF), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the EF (Enable interrupt by EI instruction).

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the EF before setting the IMF to "1".

Note 3):When a read instruction is executed on EIRL, bits 3 to 1 are read as "0". Other unused bits are read as "0".

7.5 Maskable Interrupt Priority Change Function

The priority of maskable interrupts (IL4 to IL30) can be changed to four levels, Levels 0 to 3, regardless of the basic priorities 5 to 31. Interrupt priorities can be changed by the interrupt priority change control register (ILPRS1 to ILPRS7). To raise the interrupt priority, set the Level to a larger number. To lower the interrupt priority, set the Level to a smaller number. When different maskable interrupts are generated simultaneously at the same level, the interrupt with higher basic priority is processed preferentially. For example, when the ILPRS1 register is set to 0xC0 and interrupts IL4 and IL7 are generated at the same time, IL7 is preferentially processed (provided that EF4 and EF7 have been enabled).

After reset is released, all maskable interrupts are set to priority level 0 (the lowest priority).

Note: In the main program, before manipulating the interrupt priority change control register (ILPRS1 to 7), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Set the IMF to "1" as required after operating ILPRS1 to 7 (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate ILPRS1 to 7 before setting the IMF to "1".

Interrupt priority change control register 1

ILPRS1 (0x0FF0)	7	6	5	4	3	2	1	0
Bit Symbol	IL07P		IL06P		IL05P		IL04P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL07P	Sets the interrupt priority of IL7.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL06P	Sets the interrupt priority of IL6	
IL05P	Sets the interrupt priority of IL5	
IL04P	Sets the interrupt priority of IL4	

Interrupt priority change control register 2

ILPRS2 (0x0FF1)	7	6	5	4	3	2	1	0
Bit Symbol	IL11P		IL10P		IL09P		IL08P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL11P	Sets the interrupt priority of IL11.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL10P	Sets the interrupt priority of IL10	
IL09P	Sets the interrupt priority of IL9	
IL08P	Sets the interrupt priority of IL8	

Interrupt priority change control register 3

ILPRS3 (0x0FF2)	7	6	5	4	3	2	1	0
Bit Symbol	IL15P		IL14P		IL13P		IL12P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL15P	Sets the interrupt priority of IL15.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL14P	Sets the interrupt priority of IL14	
IL13P	Sets the interrupt priority of IL13	
IL12P	Sets the interrupt priority of IL12	

Interrupt priority change control register 4

ILPRS4 (0x0FF3)	7	6	5	4	3	2	1	0
Bit Symbol	IL19P		IL18P		IL17P		IL16P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL19P	Sets the interrupt priority of IL19.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL18P	Sets the interrupt priority of IL18	
IL17P	Sets the interrupt priority of IL17	
IL16P	Sets the interrupt priority of IL16	

Interrupt priority change control register 5

ILPRS5 (0x0FF4)	7	6	5	4	3	2	1	0
Bit Symbol	IL23P		IL22P		IL21P		IL20P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL23P	Sets the interrupt priority of IL23.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL22P	Sets the interrupt priority of IL22	
IL21P	Sets the interrupt priority of IL21	
IL20P	Sets the interrupt priority of IL20	

Interrupt priority change control register 6

ILPRS6 (0x0FF5)	7	6	5	4	3	2	1	0
Bit Symbol	IL27P		IL26P		IL25P		IL24P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL27P	Sets the interrupt priority of IL27.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL26P	Sets the interrupt priority of IL26	
IL25P	Sets the interrupt priority of IL25	
IL24P	Sets the interrupt priority of IL24	

Interrupt priority change control register 7

ILPRS7 (0x0FF6)	7	6	5	4	3	2	1	0
Bit Symbol	-		IL30P		IL29P		IL28P	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

IL30P	Sets the interrupt priority of IL30.	00: Level 0 (lower priority) 01: Level 1 10: Level 2 11:Level3 (higher priority)
IL29P	Sets the interrupt priority of IL29	
IL28P	Sets the interrupt priority of IL28	

7.6 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to "0" by resetting or an instruction. Interrupt acceptance sequence requires 8-machine cycles after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts).

7.6.1 Initial Setting

Using an interrupt requires specifying an SP (stack pointer) for it in advance. The SP is a 16-bit register pointing at the start address of a stack. The SP is post-decremented when a subroutine call or a push instruction is executed or when an interrupt request is accepted. It is pre-incremented when a return or pop instruction is executed. Therefore, the stack becomes deeper toward lower stack location addresses. Be sure to reserve a stack area having an appropriate size based on the SP setting.

The SP is initialized to 00FFH after a reset. If you need to change the SP, do so right after a reset or when the interrupt master enable flag IMF) is "0".

Example :SP setting

```
LD      SP, 023FH      ; SP = 023FH
LD      SP, SP+04H     ; SP = SP + 04H
ADD     SP, 0010H      ; SP = SP + 0010H
```

7.6.2 Interrupt acceptance processing

Interrupt acceptance processing as following:

1. The interrupt master enable flag (IMF) is cleared to "0" in order to disable the acceptance of any following interrupt.

2. The interrupt latch (IL) for the interrupt source accepted is cleared to "0".

3. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.

4. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.

5. The instruction stored at the entry address of the interrupt service program is executed.

Note : When the contents of PSW are saved on the stack, the contents of register bank and IMF are also saved.

A maskable interrupt is not accepted until the IMF is set to "1" even if the maskable interrupt is requested in the interrupt service routine. In order to utilize nested interrupt service, the IMF must be set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to "1". As for non-maskable interrupt, keep interrupt service shorter compared with length between interrupt requests.

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program.

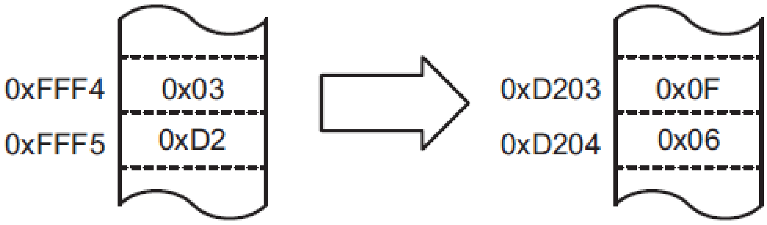


Figure 7.2 Vector table address and Entry address

7.6.3 Saving/restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the general purpose registers are not. These registers must be saved by software if necessary. When multiple interrupt services are nested, it is also necessary to avoid using the same data memory area for saving registers. The following methods are used to save/restore the general-purpose registers.

<u>PUSH and POP instructions</u>

To save only a specific register, PUSH and POP instructions are available.

Example :Using PUSH and POP instructions

```
PUSH                    WA          ; Save WA register
Interrupt processing
POP                     WA          ; Restore WA register
```

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

```
RETI                                  ; RETURN
```

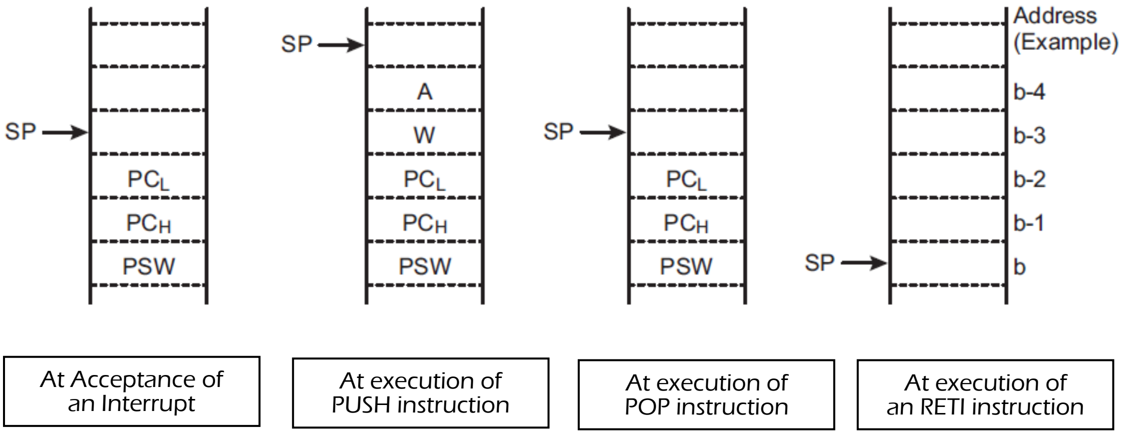

SP
PCL
PCH
PSW
SP
A
W
PCL
PCH
PSW
SP
PCL
PCH
PSW
SP
Address
(Example)
b-4
b-3
b-2
b-1
b
At Acceptance of an Interrupt
At execution of PUSH instruction
At execution of POP instruction
At execution of an RETI instruction

Figure 7.3 Saving/restoring general-purpose registers

Data transfer instructions

To save only a specific register without nested interrupts, data transfer instructions are available.

Example :Save/store register using data transfer instructions

```
LD                        (GSAVA),    ; Save A register
                          A
Interrupt processing
LD                        A,          ; Restore A register
                          (GSAVA)
RETI                                  ; RETURN
```

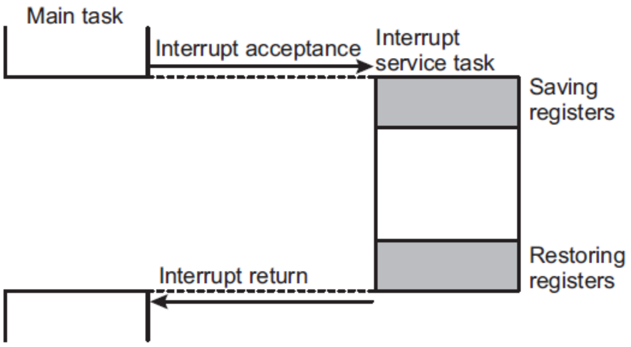


Figure 7.4 Saving/Restoring General-purpose Registers under Interrupt Processing

Using a register bank to save/restore general-purpose registers

In non-multiple interrupt handling, the register bank function can be used to save/restore the general-purpose registers at a time. The register bank function saves (switches) the general-purpose registers by executing a register bank manipulation instruction (such as LD RBS,1) at the beginning of an interrupt service task. It is unnecessary to re-execute the register bank manipulation instruction at the end of the interrupt service task because executing the RETI instruction makes a return automatically to the register bank that was being used by the main task according to the content of the PSW.

Note: Two register banks (BANK0 and BANK1) are available. Each bank consists of 8-bit general-purpose registers (W, A, B, C, D, E, H, and L) and 16-bit general-purpose registers (IX and IY).

Example : Saving/restoring registers, using an instruction for transfer with data memory (with the main task using the register bank BANK0)

```
LD                    RBS,1              ;Switches to the register bank BANK1
Interrupt processing
RETI                                     ;RETURN
                                         (Makes a return automatically to
                                         BANK0 that was being used by the
                                         main task when the PSW is restored)
```

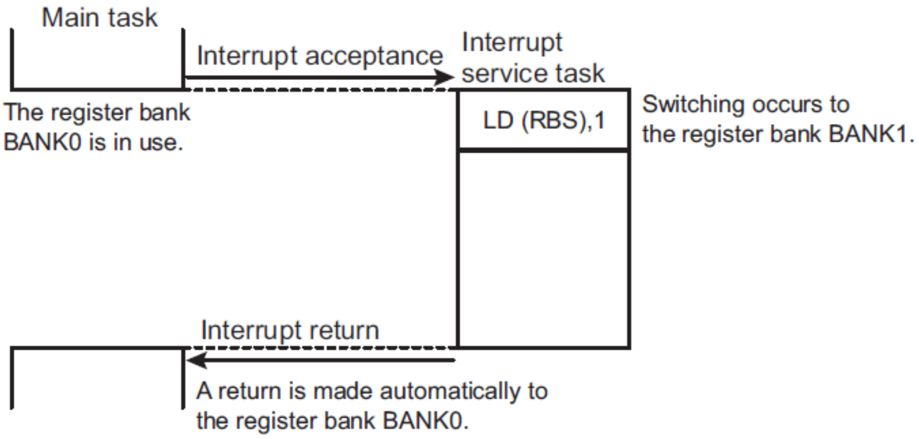


Figure 7.5 Saving/Restoring General-purpose Registers under Interrupt Processing

Interrupt return

Interrupt return instructions [RETI]/[RETN] as following.

[RETI]/[RETN] Interrupt Return	1. Program counter (PC) and program status word (register bank) are restored from the stack. 2. Stack pointer (SP) is incremented by 3.

7.7 External Interrupt Control Circuit

External interrupts detects the change of the input signal and generates an interrupt request. Noise can be removed by the built-in digital noise canceller.

7.7.1 Configuration

The external interrupt control circuit consists of a noise canceller, an edge detection circuit, a level detection circuit and an interrupt signal generation circuit.

Externally input signals are input to the rising edge or falling edge or level detection circuit for each external interrupt, after noise is removed by the noise canceller.

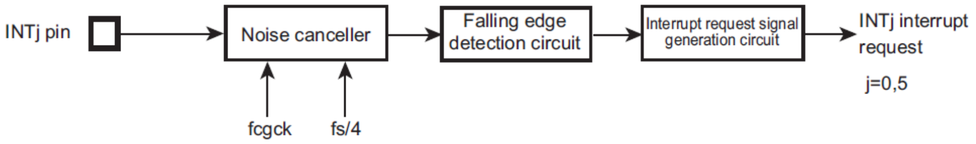

INTj pin
Noise canceller
Falling edge detection circuit
Interrupt request signal generation circuit
INTj interrupt request
j=0,5
fcgck
fs/4

Figure7.6 External Interrupts 0/5

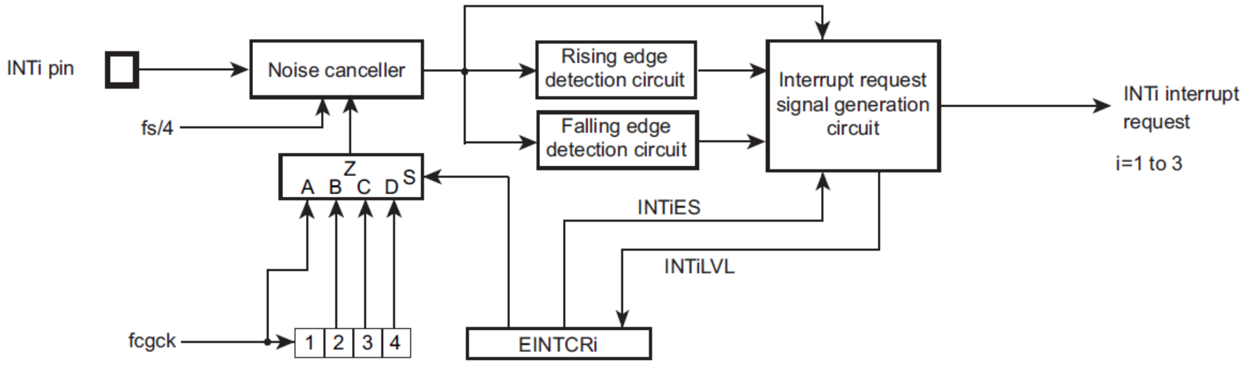

INTi pin
Noise canceller
fs/4
Rising edge detection circuit
Falling edge detection circuit
Interrupt request signal generation circuit
INTi interrupt request
i=1 to 3
Z
A B C D S
INTiES
INTiLVL
fcgck
1 2 3 4
EINTCRi

Figure7.7 External Interrupts 1/2/3

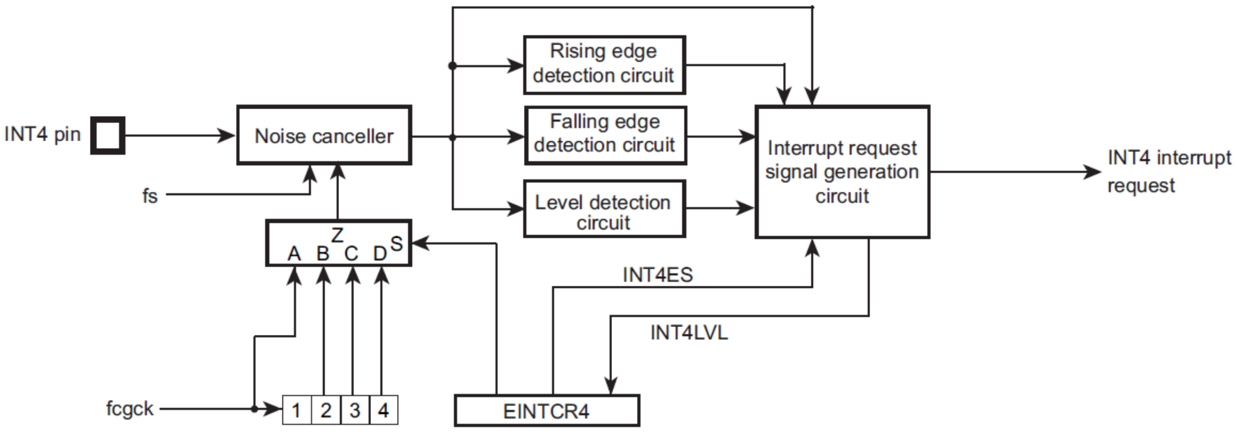


Figure7.8 External Interrupts 4

7.7.2 Control

External interrupts are controlled by the following registers:

Low Power Consumption Register 3

POFFCR3 (0x0F77)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	INT5EN	INT4EN	INT3EN	INT2EN	INT1EN	INT0EN
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 Control	0 1	Disable Enable
INT4EN	INT4 Control	0 1	Disable Enable
INT3EN	INT3 Control	0 1	Disable Enable
INT2EN	INT2 Control	0 1	Disable Enable
INT1EN	INT1 Control	0 1	Disable Enable
INT0EN	INT0 Control	0 1	Disable Enable

Note 1): Clearing INTxEN (x=0 to 5) to "0" stops the clock supply to the external interrupts. This invalidates the data written in the control register for each external interrupt. When using the external interrupts, set INTxEN to "1" and then write data into the control register for each external interrupt.

Note 2): Interrupt request signals may be generated when INTxEN is changed. Before changing INTxEN, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3):Bits 7 and 6 of POFFSET3 are read as "0".

External Interrupt Control Register 1

EINTCR1 (0x0FD8)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT1LVL	INT1ES		INT1NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INT1LVL	Signal level that passes noise canceller when the interrupt request signal is generated for external interrupt 1	0: Initial state or signal level "L" 1: Signal level "H"	
INT1ES	Select the interrupt request generating condition for external interrupt 1	00: An interrupt request is generated at the rising edge of the noise canceller pass signal 01: An interrupt request is generated at the falling edge of the noise canceller pass signal 10: An interrupt request is generated at both edges of the noise canceller pass signal 11: Reserved	
INT1NC	Set the noise canceller sampling interval for external interrupt 1	NORMAL 1/2, IDLE 1/2	SLOW 1/2, SLEEP1
		00: fcgck [Hz] 01: fcgck / 2^2 [Hz] 10: fcgck / 2^3 [Hz] 11: fcgck / 2^4 [Hz]	00: fs/4 [Hz] 01: fs/4 [Hz] 10: fs/4 [Hz] 11: fs/4 [Hz]

Note 1): fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2): Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3): Interrupt requests may be generated when EINTCRx(x = 1~4) is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note4: The contents of EINTCRx <INTxLVL> are updated each time an interrupt request signal is generated.

Note 5): Bits 7 to 5 of EINTCR1 are read as "0".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

External Interrupt Control Register 2

EINTCR2 (0x0FD9)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT2LVL	INT2ES		INT2NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INT2LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 2	0: Initial state or signal level "L" 1: Signal level "H"	
INT2ES	Select the interrupt request generating condition for external interrupt 2	00: An interrupt request is generated at the rising edge of the noise canceller pass signal 01: An interrupt request is generated at the falling edge of the noise canceller pass signal 10: An interrupt request is generated at both edges of the noise canceller pass signal 11: Reserved	
INT2NC	Set the noise canceller sampling interval for external interrupt 1	NORMAL 1/2, IDLE 1/2	SLOW 1/2, SLEEP1
		00: fcgck [Hz] 01: fcgck / 2^2 [Hz] 10: fcgck / 2^3 [Hz] 11: fcgck / 2^4 [Hz]	00: fs/4 [Hz] 01: fs/4 [Hz] 10: fs/4 [Hz] 11: fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR2 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/ fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR2 are read as "0".

External Interrupt Control Register 3

EINTCR3 (0x0FDA)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT3LVL	INT3ES		INT3NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INT3LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 3	0: Initial state or signal level "L" 1: Signal level "H"	
INT3ES	Select the interrupt request generating condition for external interrupt 3	00: An interrupt request is generated at the rising edge of the noise canceller pass signal 01: An interrupt request is generated at the falling edge of the noise canceller pass signal 10: An interrupt request is generated at both edges of the noise canceller pass signal 11: Reserved	
INT3NC	Set the noise canceller sampling interval for external interrupt 3	NORMAL 1/2, IDLE 1/2	SLOW 1/2, SLEEP1
		00: fcgck [Hz] 01: fcgck / 2^2 [Hz] 10: fcgck / 2^3 [Hz] 11: fcgck / 2^4 [Hz]	00: fs/4 [Hz] 01: fs/4 [Hz] 10: fs/4 [Hz] 11: fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR3 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR3 are read as "0".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

External Interrupt Control Register 4

EINTCR4 (0x0FDB)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT4LVL	INT4ES		INT4NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INT4LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 4	0: Initial state or signal level "L" 1: Signal level "H"	
INT4ES	Select the interrupt request generating condition for external interrupt 4	00: An interrupt request is generated at the rising edge of the noise canceller pass signal 01: An interrupt request is generated at the falling edge of the noise canceller pass signal 10: An interrupt request is generated at both edges of the noise canceller pass signal 11: Reserved	
INT4NC	Set the noise canceller sampling interval for external interrupt 4	NORMAL 1/2, IDLE 1/2	SLOW 1/2, SLEEP1
		00: fcgck [Hz] 01: fcgck / 2^2 [Hz] 10: fcgck / 2^3 [Hz] 11: fcgck / 2^4 [Hz]	00: fs/4 [Hz] 01: fs/4 [Hz] 10: fs/4 [Hz] 11: fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/ 2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR4 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/ 2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: The contents of EINTCRx<INTxLVL> are updated each time an interrupt request signal is generated.

Note 5: Bits 7 to 5 of EINTCR4 are read as "0".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

7.7.3 Function

The condition for generating interrupt request signals and the noise cancel time can be set for external interrupts 2 to 4.

Source	Pin	Enable Conditions	Interrupt request signal generated	External interrupt pin input signal width and noise removal	
				NORMAL 1/2,IDLE 1/2	SLOW 1/2, SLEEP 1
INT0	INT0B	IMF=1 EF16 =1	Falling edge	Less than 1/fcgck: Noise More than 1/fcgck and less than 2/fcgck: Indeterminate More than 2/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT1	INT1	IMF=1 EF17=1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT2	INT2	IMF=1 EF18=1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT3	INT3	IMF=1 EF19=1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT4	INT4	IMF=1 EF20=1	Falling edge Rising edge Both edges "H" level	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT5	INT5B	IMF=1 EF8=1	Falling edge	Less than 1/fcgck: Noise More than 1/fcgck and less than 2/fcgck: Indeterminate More than 2/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal

Table 7.2 External Interrupts

Note): fcgck: Gear clock [Hz]; fs: Low frequency clock [Hz]; fspl: Sampling interval [Hz]

7.7.3.1 Low Power Consumption Function

External interrupts have a function that saves power by using the low power consumption register (POFFCR3) when they are not used. Setting POFFCR3<INTxEN> to "0" stops (disables) the basic clock for external interrupts and helps save power. Note that this makes external interrupts unavailable. Setting POFFCR3<INTxEN> to "1" supplies (enables) the basic clock for external interrupts and makes external interrupts available.

After reset, POFFCR3<INTxEN> is initialized to "0" and external interrupts become unavailable. When using the external interrupt function for the first time, be sure to set POFFCR3<INTxEN> to "1" in the initial setting of software (before operating the external interrupt control registers).

Note): Interrupt request signals may be generated when INTxEN is changed. Before changing INTxEN, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

7.7.3.2 External Interrupt 0 to 5

External interrupt 0

External interrupt 0 detects the falling edge of the INT0 pin and generates interrupt request signals.
In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1/fcgck are removed as noise and pulses of 2/fcgck or more are recognized as signals.
In SLOW/SLEEP mode, pulses of less than 4/fs are removed as noise and pulses of 8/fs or more are recognized as signals.

External interrupt 1/2/3

External interrupt 1/2/3 detects the falling edge, the rising edge, both edges or "H" level of the INTx pin and generates interrupt request signals.

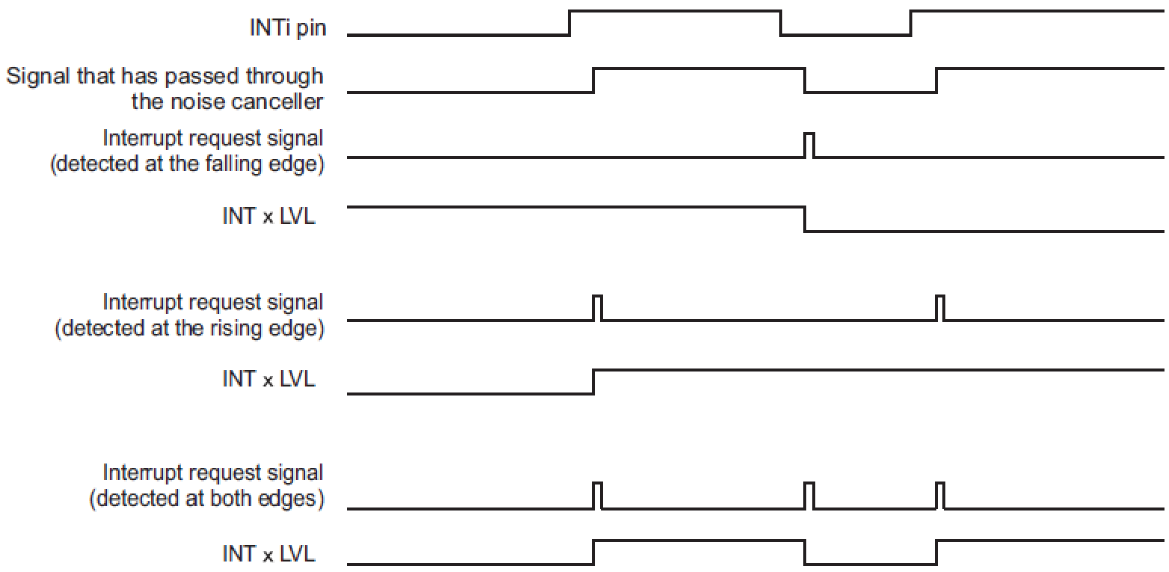
(a) Interrupt Request Signal Generating Condition Detection Function

Select an interrupt request signal generating condition at EINTCRx<INTxES> for external interrupt 1/2/3.

EINTCRx<INTxES>	Detectedat
00	Rising edge
01	Falling edge
10	Both edges
11	"H" level interrupt

Table 7.3 Selection of Interrupt Request Generation Edge

(b) A Noise Canceller Pass Signal Monitoring Function when Interrupt Request Signals Generated


INTi pin
Signal that has passed through the noise canceller
Interrupt request signal (detected at the falling edge)
INT x LVL
Interrupt request signal (detected at the rising edge)
INT x LVL
Interrupt request signal (detected at both edges)
INT x LVL

Note: The contents of EINTCRx<INTxLVL> are updated each time an interrupt request signal is generated.

Figure7.9 Interrupt Request Generation and EINTCRx<INTxLVL> (x = 1 to 3)

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCRx <INTxLVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCRx <INTxLVL>.

(c) Noise Cancel Time Selection Function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCRx<INTxNC>. If the same level is detected three consecutive times, the signal is recognized as a signal. If not, the signal is removed as noise.

EINTCRx <INTxES>	Sampling Interval
00	fcgck
01	fcgck/2^2
10	fcgck/2^3
11	fcgck/2^4

Table 7.4Noise Canceller Sampling Clock

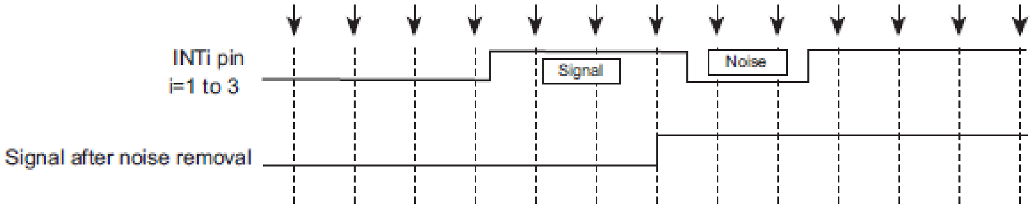


Figure7.10 Noise Cancel Operation

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

In SLOW1/2 or SLEEP1 mode, a signal is sampled by the low frequency clock divided by 4. If the same level is detected twice consecutively, the signal is recognized as a signal.

In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

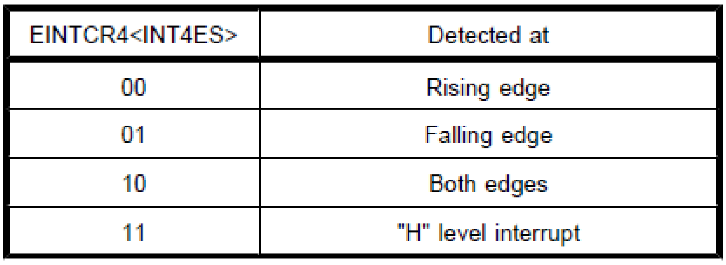
Note 1): When noise is input consecutively during sampling external interrupt pins, the noise cancel function does not work properly. Set EINTCRx <INTxNC> according to the cycle of externally input noise.
Note 2): When an external interrupt pin is used as an output port, the input signal to the port is fixed to "L" when the mode is switched to the output mode, and thus an interrupt request occurs. To use the pin as an output port, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt.
Note 3): Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/ fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

External interrupt 4

External interrupt 4 detects the falling edge, the rising edge, both edges or "H" level of the INT4 pin and generates interrupt request signals.

(a) Interrupt request signal generating condition detection function

Select an interrupt request signal generating condition at EINTCR4<INT4ES> for external interrupt 4.



EINTCR4<INT4ES>	Detected at
00	Rising edge
01	Falling edge
10	Both edges
11	"H" level interrupt

Table 7.5 Selection of Interrupt Request Generation Edge

(b) Noise canceller pass signal monitoring function when interrupt request signals are generated

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCR4<INT4LVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCR4<INT4LVL>.

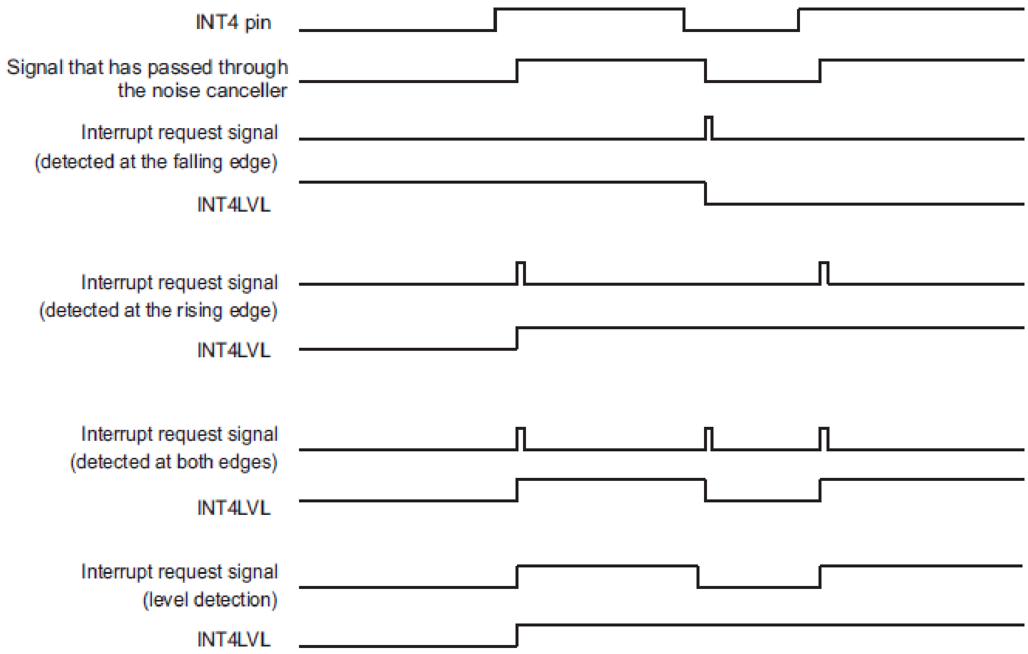

INT4 pin
Signal that has passed through the noise canceller
Interrupt request signal (detected at the falling edge)
INT4LVL
Interrupt request signal (detected at the rising edge)
INT4LVL
Interrupt request signal (detected at both edges)
INT4LVL
Interrupt request signal (level detection)
INT4LVL

Figure 7.11 Interrupt Request Generation and EINTCR4<INT4LVL>

(c) Noise cancel time selection function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCRx<INT4NC>. If the same level is detected three consecutive times, the signal is recognized as a signal. If not, the signal is removed as noise.

EINTCR4<INT4NC>	Sampling interval
00	fcgck
01	fcgck/2^2
10	fcgck/2^3
11	fcgck/2^4

Table 7.6 Noise Canceller Sampling Lock

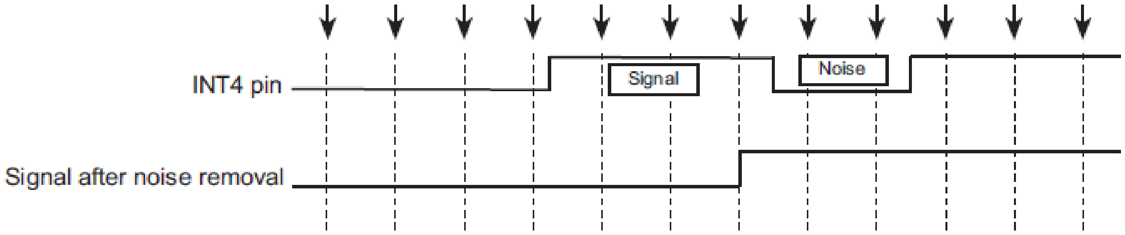


Figure 7.12 Noise Cancel Operation

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

In SLOW1/2 or SLEEP1 mode, a signal is sampled by the low frequency clock divided by 4. If the same level is detected twice consecutively, the signal is recognized as a signal. In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

Note 1: When noise is input consecutively during sampling external interrupt pins, the noise cancel function does not work properly. Set EINTCRx<INTxNC> according to the cycle of externally input noise.
Note 2: When an external interrupt pin is used as an output port, the input signal to the port is fixed to "L" when the mode is switched to the output mode, and thus an interrupt request occurs. To use the pin as an output port, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt.
Note 3: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1,wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

External interrupt 5

External interrupt 5 detects the falling edge of the INT5 pin and generates interrupt request signals. In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1/fcgck are removed as noise and pulses of 2/fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4/fs are removed as noise and pulses of 8/fs or more are recognized as signals.

7.8 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSW is the top-priority interrupt). Use the SWI instruction only for address error detection or for debugging described below.

Address error detection:
0xFF is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code 0xFF is an SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing 0xFF to unused areas in the program memory.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Debugging:

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

7.9 Undefined Instruction Interrupt (INTUNDEF)

When the CPU tries to fetch and execute an instruction that is not defined, INTUNDEF is generated and starts the interrupt processing. INTUNDEF is accepted even if another non-maskable interrupt is in process. The current process is discontinued and the INTUNDEF interrupt process starts soon after it is requested.

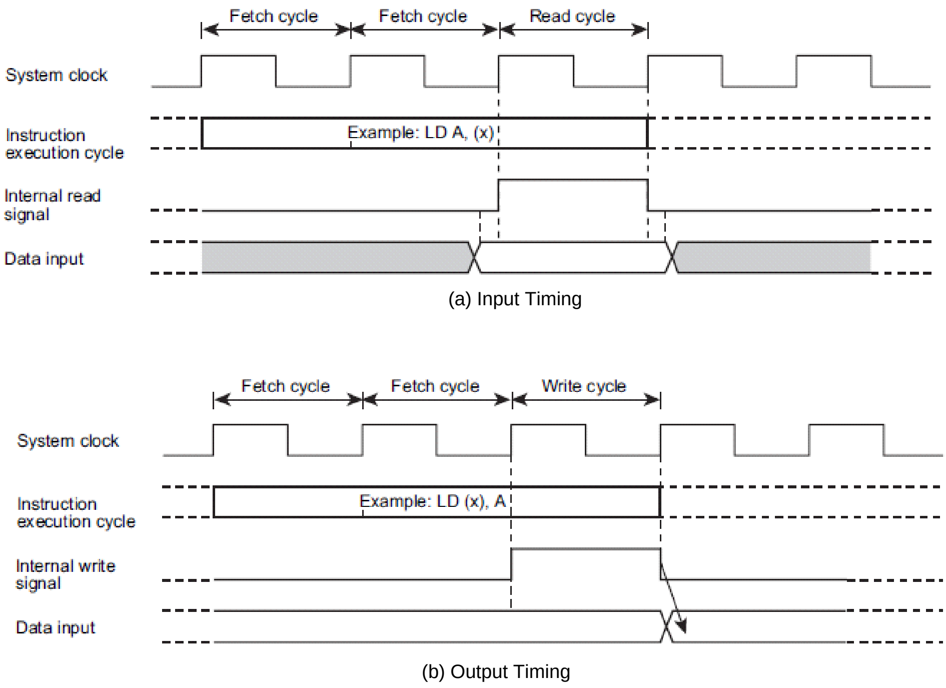
Note: The undefined instruction interrupt (INTUNDEF) forces the CPU to jump into the interrupt vector address, as software interrupt (SWI) does.

8. I/O Ports

MQ6935 has 8 parallel input / output ports (40 I/O pins) as follows:

Port Name	Pin Name	No. of Pins	Input/output	Secondary Functions
Port P0	P03 to P00	4	Input / Output	Also used as the high-frequency oscillator connection pin and the low-frequency oscillator connection pin
Port P1	P13 to P10	4	Input / Output	Also used as the external reset input, the external interrupt input and the STOP mode release signal input
Port P2	P27to P20	8	Input / Output	Also used as the UART input/output, the serial interface input/output and the serial bus interface input/output
Port P4	P47 to P40	8	Input / Output	Also used as the timer counter input/output, the analog input and the key-on wakeup input
Port P7	P77 to P70	8	Input / Output	Also used as the timer counter input/output and the external interrupt input
Port P8	P81 to P80	2	Input / Output	Also used as the timer counter input/output
Port P9	P91 to P90	2	Input / Output	Also used as the UART input/output
Port PB	PB7 to PB4	4	Input / Output	Also used as the UART input/output and the serial interface input/output

Table 8.1 List of I/O Ports



(a) Input Timing

(b) Output Timing

Figure 8.1 Input / Output Timing (Example)

Note: The positions of the read and write cycles may vary, depending on the instruction.

Each output port contains a latch, which holds the output data. No input port has a latch, so the external input data should be externally held until the input data is read from outside or reading should be performed several times before processing. Figure 8.1 shows input / output timing examples.

External data is read from an I/O port in the read cycle during execution of the read instruction. This timing cannot be recognized from outside, so that transient input such as chattering must be processed by the program. Data is output to an I/O port in the next cycle of the write cycle during execution of the write instruction.

8.1 I/O Port Control Registers

The following control registers are used for I/O ports. (The port number is indicated in place of x.) Registers that can be set vary depending on the port. For details, refer to the description of each port.

PxDR Register

This is the register for setting output data. When a port is set to the "output mode", the value specified at PxDR is output from the port.

PxPRD Register

This is the register for reading input data. When a port is set to the "input mode", the current port input status can be read by reading PxPRD.

PxCR Register

This register switches a port between input and output. A port can be switched between the "input mode" and the "output mode".

PxFC Register

This register enables the secondary function output of each port. The secondary function output of each port can be enabled or disabled.

PxPU Register

This register determines whether or not the built-in pull-up resistor is connected when a port is used in the input mode.

PxOUTCR Register

This register switches the port output between the C-MOS output and the open drain output.

8.2 List of I/O Ports Settings

For the setting methods for individual I/O ports, refer to the following table.

Port Name	Pin Name	Function	Register Set Value			
			PxCR	PxOUTCR	PxFC	Other Required Settings
Port P0	P03 to P00	Port input	0	Without register	0	
		Port output	1		0	
	P03	XTOUT	*		Without register	
	P02	XTIN	*		1	
	P01	XOUT	*		Without register	
	P00	XIN	*		1	
Port P1	P13 to P11	Port input	0	Without register	Without register	
		Port output	1			
	P10	Port input	0			Note 1
	P10	Port output	1			Note 1
	P13	INT1 input	0			
	P12	INT0B input	0			
	P11	INT5B input	0			
	P11	STOPB input	0			
	P10	RESETB input	*			Note 1
Port P2	P27 to P20	Port input	0	*	*	
		Port output	1	**	0	
	P27	RXD2 input	0	*	0	UATCNG<UAT2IO>="0"
		TXD2 output	1	**	1	UATCNG<UAT2IO>="1"
	P26	TXD2 output	1	**	1	UATCNG<UAT2IO>="0"
		RXD2 input	0	*	0	UATCNG<UAT2IO>="1"
	P25	SCLK0 input	0	*	*	SERSEL<SRSEL0>="01"
		SLK0 output	1	**	1	SERSEL<SRSEL0>="01"
	P24	SCL0 input/output	1	Without register	1	SERSEL<SRSEL0>="*0"
		SI input	0		*	SERSEL<SRSEL0>="01"
	P23	SDA0 input/output	1	Without register	1	SERSEL<SRSEL0>="*0"
		SO output	1		1	SERSEL<SRSEL0>="01"
	P22	SCLK0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"
		SCLK0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"
	P21	RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="0"
		TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0"

						UATCNG<UAT0IO>="1"
		SI0 input	0	*	*	SERSEL<SRSEL0>="10*" SERSEL<SRSEL2>="0"
	P20	TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="0"
		RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="0" UATCNG<UAT0IO>="1"
		SO0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="0"
Port 4	P47 to P40	Port input	0	Without register	0	
		Port output	1		0	
		AIN7 to AIN0	0		1	
		KWI7 to KWI4	*		*	KWUCR1
		KWI3 to KWI0	*		*	KWUCR0
	P47	EMG0B input	0		0	ITSEL<ITSEL1>="1"
	P46	TTC0 input	0		0	ITSEL<ITSEL0>="1"
	P45	PPGC02B output	1		1	
	P44	PPGC01B output	1		1	
Port P7	P77 to P70	Port input	0	Without register	*	
		Port output	1		0	
	P77	INT4 input	0		Without register	
	P76	INT3 input	0		Without register	
	P75	INT2 input	0		Without register	
	P74	DOVB output	1		1	
	P73	TCA1 input	0		*	
		PPGA1B output	1		1	
	P72	TCA0 input	0		*	SERSEL<TCA0SEL>="00"
		PPGA0B output	1		1	
	P71	TC01 input	0		*	
		PPG01 / PWM01 output	1		1	
	P70	TC00 input	0		*	
		PPG00 / PWM00 output	1		1	
Port P8	P81 to P80	Port input	0	Without register	*	
		Port output	1		0	
	P81	TC03 input	0		*	
		PPG03 / PWM03 output	1		1	
	P80	TC02 input	0		*	
		PPG02 / PWM02 output	1		1	

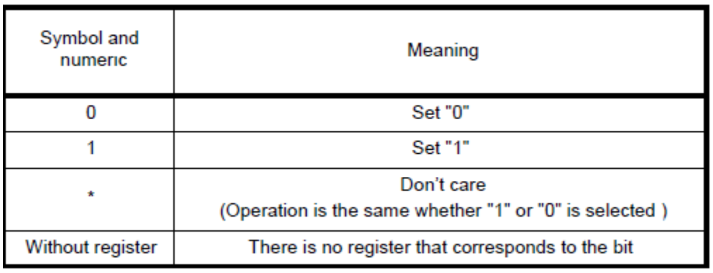
iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Port Name	Pin Name	Function	Register Set Value			
			PxCR	PxOUTCR	PxFC	Other Required Settings
Port P9	P92 to P90	Port input	0	*	*	
		Port output	1	**	0	
	P91	RXD1 input	0	*	0	UATCNG<UAT1IO>="0"
		TXD1 output	1	**	1	UATCNG<UAT1IO>="1"
	P90	TXD1 output	1	**	1	UATCNG<UAT1IO>="0"
		RXD1 input	0	*	0	UATCNG<UAT1IO>="1"
Port PB	PB7 to PB4	Port input	0	*	*	
		Port output	1	**	0	
	PB6	SCLK0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"
		SCLK0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"
	PB5	RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="0"
		TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="1"
		SI0 input	0	*	*	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"
	PB4	TXD0 output	1	**	1	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="0"
		RXD0 input	0	*	*	SERSEL<SRSEL0>="0*" SERSEL<SRSEL2>="1" UATCNG<UAT0IO>="1"
		SO0 output	1	**	1	SERSEL<SRSEL0>="10" SERSEL<SRSEL2>="1"

Table 8.2 List of I/O Port Settings

Note 1): After the power is turned on, pin P10 serves as an external reset input. To use pin P10 as a port, refer to "How to use the external reset input pin as a port" of "5.1 Reset Control Circuit".

Note 2): The symbol and numeric characters in the table have the following meanings:



Symbol and numeric	Meaning
0	Set "0"
1	Set "1"
*	Don't care (Operation is the same whether "1" or "0" is selected)
Without register	There is no register that corresponds to the bit

8.3 I/O Port Control Register

8.3.1 Port P0 (P03 to P00) Register

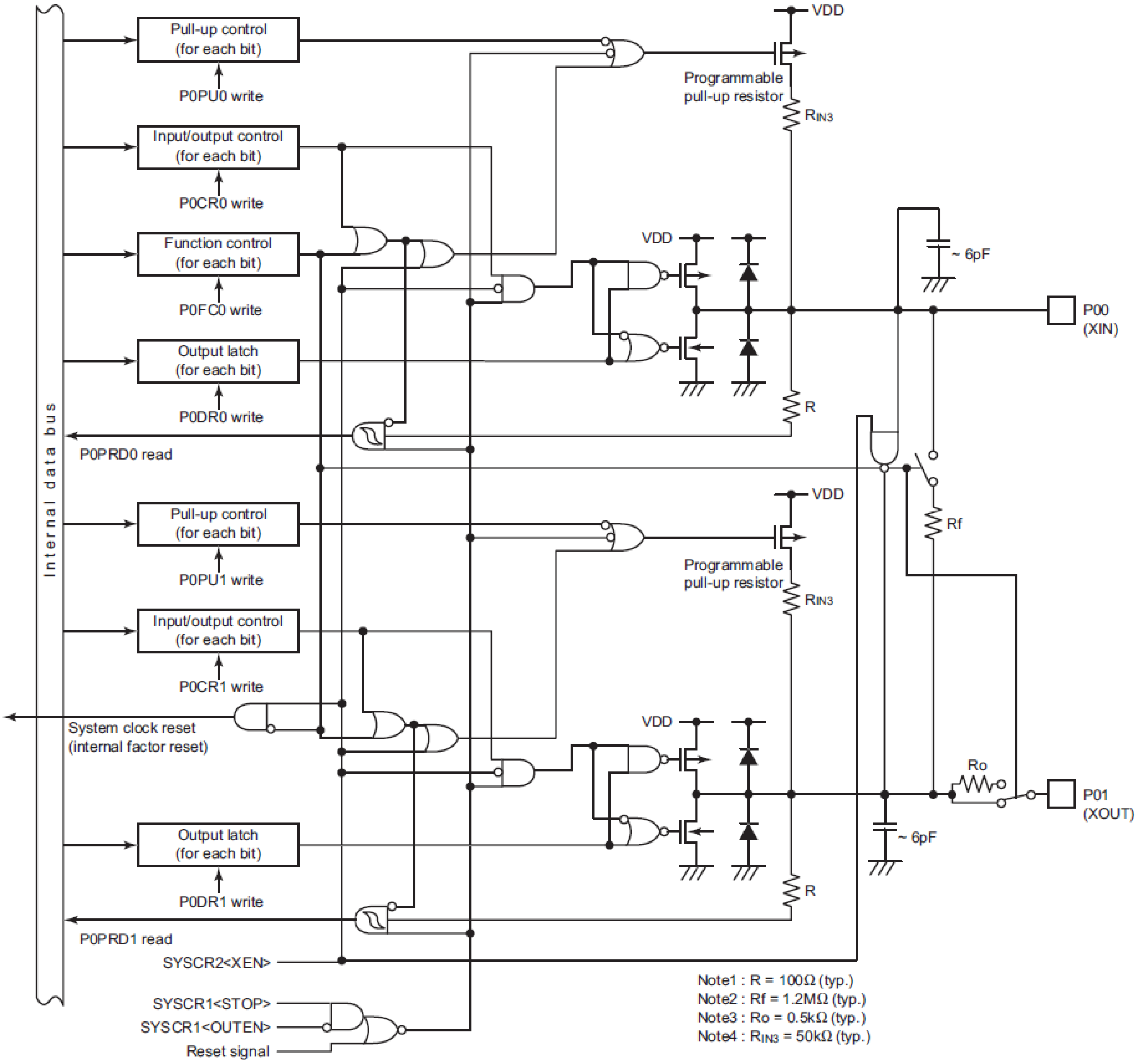
Port P0 is a 4-bit input / output port that can be set to input or output for each bit individually, and it is also used as the high-frequency external crystal connection pin and the low-frequency external crystal connection pin.

Port P0 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode.

Port Name	P03	P02	P01	P00
Secondary function	XTOUT	XTIN	XOUT	XIN

Table 8.3 Port P0

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.


Internal data bus
Pull-up control (for each bit)
P0PU0 write
Input/output control (for each bit)
P0CR0 write
Function control (for each bit)
P0FC0 write
Output latch (for each bit)
P0DR0 write
P0PRD0 read
Pull-up control (for each bit)
P0PU1 write
Input/output control (for each bit)
P0CR1 write
System clock reset (internal factor reset)
Output latch (for each bit)
P0DR1 write
P0PRD1 read
SYSCR2<XEN>
SYSCR1<STOP>
SYSCR1<OUTEN>
Reset signal
VDD
Programmable pull-up resistor
RIN3
VDD
~ 6pF
P00 (XIN)
R
Rf
VDD
Programmable pull-up resistor
RIN3
VDD
Ro
P01 (XOUT)
~ 6pF
R

Note1 : R = 100Ω (typ.)
Note2 : Rf = 1.2MΩ (typ.)
Note3 : Ro = 0.5kΩ (typ.)
Note4 : R_{IN3} = 50kΩ (typ.)

Figure 8.2 Port P0 (P00,P01)

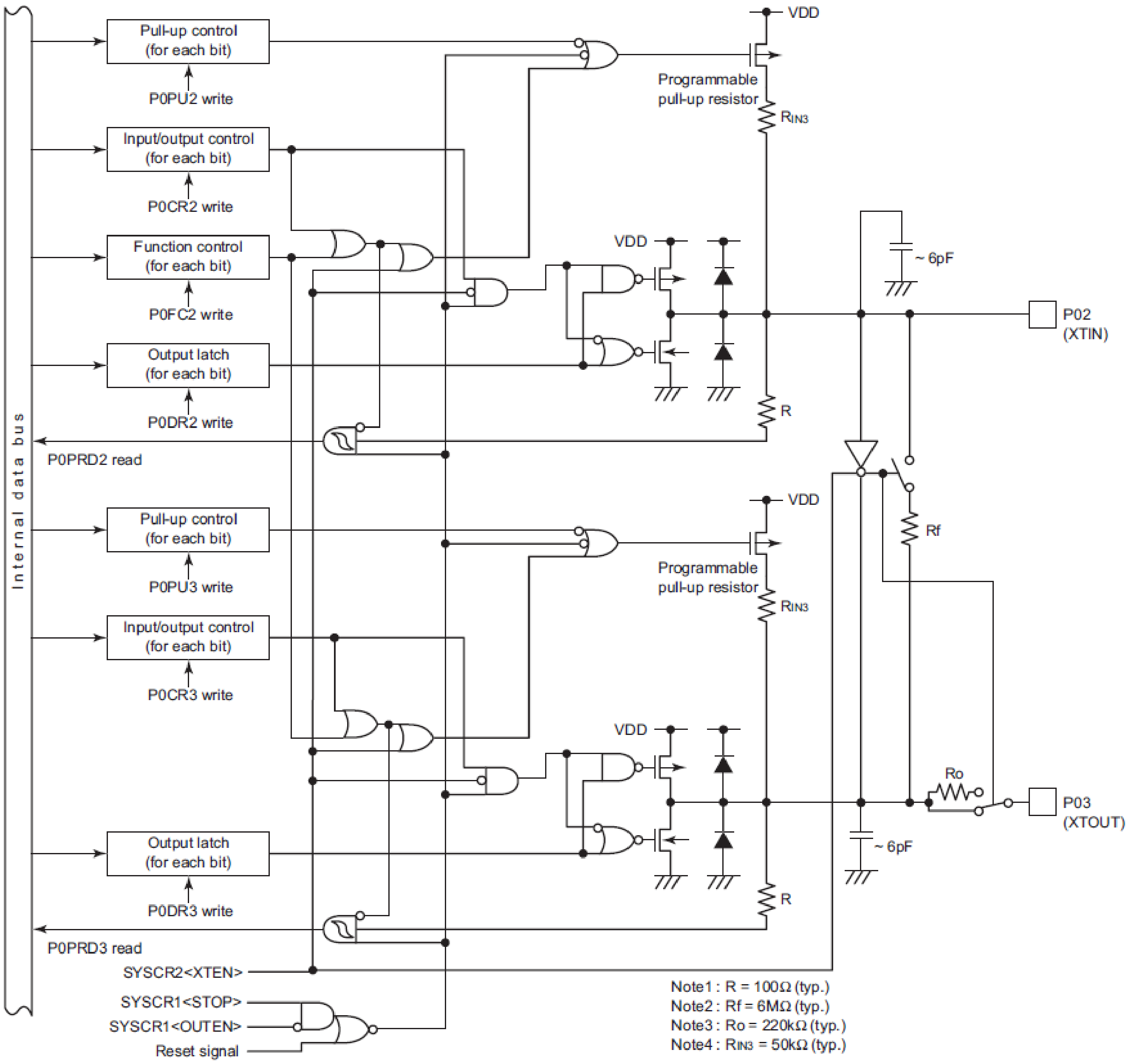

Internal data bus
Pull-up control (for each bit)
P0PU2 write
Input/output control (for each bit)
P0CR2 write
Function control (for each bit)
P0FC2 write
Output latch (for each bit)
P0DR2 write
P0PRD2 read
Pull-up control (for each bit)
P0PU3 write
Input/output control (for each bit)
P0CR3 write
Output latch (for each bit)
P0DR3 write
P0PRD3 read
SYSCR2<XTEN>
SYSCR1<STOP>
SYSCR1<OUTEN>
Reset signal
VDD
Programmable pull-up resistor
RIN3
VDD
R
~ 6pF
P02 (XTIN)
Rf
VDD
Programmable pull-up resistor
RIN3
VDD
R
Ro
~ 6pF
P03 (XTOUT)
Note1 : R = 100Ω (typ.)
Note2 : Rf = 6MΩ (typ.)
Note3 : Ro = 220kΩ (typ.)
Note4 : RIN3 = 50kΩ (typ.)

Figure 8.3 Port P0 (P02,P03)

Port P0 Output Latch Register

P0DR (0x0000)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P03	P02	P01	P00
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Outputs L level when the output mode is selected			
	1:					Outputs H level when the output mode is selected			

Port P0 Input / Output Control Register

P0CR (0x0F1A)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P0CR3	P0CR2	P0CR1	P0CR0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Input mode (port input)			
	1:					Output mode (port output)			

Port P0 Function Control Register

P0FC (0x0F34)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	P0FC2	-	P0FC0
Read/Write		R	R	R	R	R	R/W	R	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:						Port function		Port function
	1:						XTIN (1)		XIN (1)

Note 1): When SYSCR2 <XEN> is "1", setting P0FC0 to "0" generates a system clock (internal factor) reset. Normally, ports P00 or P01 are not used as ports, so P0FC0 must be set to "1".

Note 2): Symbol "I" means secondary function input.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

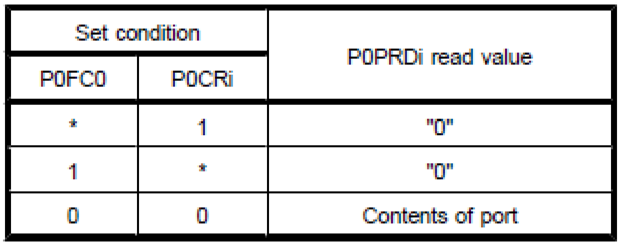
Port P0 Built-in Pull-up Resistor Control Resistor

P0PU (0x0F27)		7	6	5	4	3	2	1	0
BitSymbol		-	-	-	-	P0PU3	P0PU2	P0PU1	P0PU0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:					The built-in pull-up resistor is not connected.			
	1:					The built-in pull-up resistor is connected. (Note)			

Note): The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected.

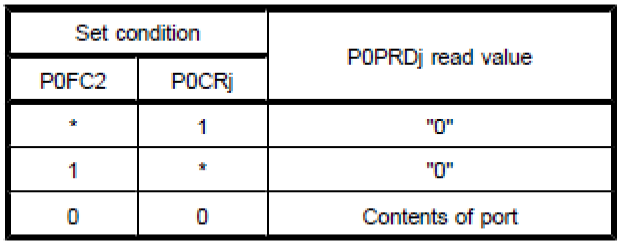
Port P0 Input Data Register

P0PRD (0x000D)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	P0PRD3	P0PRD2	P0PRD1	P0PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	*	*	*	*
Function					If the port is in the input mode, the contents of the port are read. If not, "0" is read.			



Set condition		P0PRDi read value
P0FC0	P0CRi	
*	1	"0"
1	*	"0"
0	0	Contents of port

P0PRD Read Value (P00 to P01)



Set condition		P0PRDj read value
P0FC2	P0CRj	
*	1	"0"
1	*	"0"
0	0	Contents of port

P0PRD Read Value (P02 to P03)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

8.3.2 Port P1(P13 to P10) Register

Port P1 is a 4-bit input / output port that can be set to input or output for each bit individually.

Port P1 contains a programmable pull-up resistor on the VDD side. This pull-up can be used when the port is used in the input mode.

After reset, pin P10 serves as an I/O port. To use pin P10 as an external reset input, refer to "How to use the external reset input pin as a port" of "5.1 Reset Control Circuit".

Port Name	P13	P12	P11	P10
Secondary function	INT1	INT0B	INT5B STOPB	RESETB

Table 8.4 Port P1

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

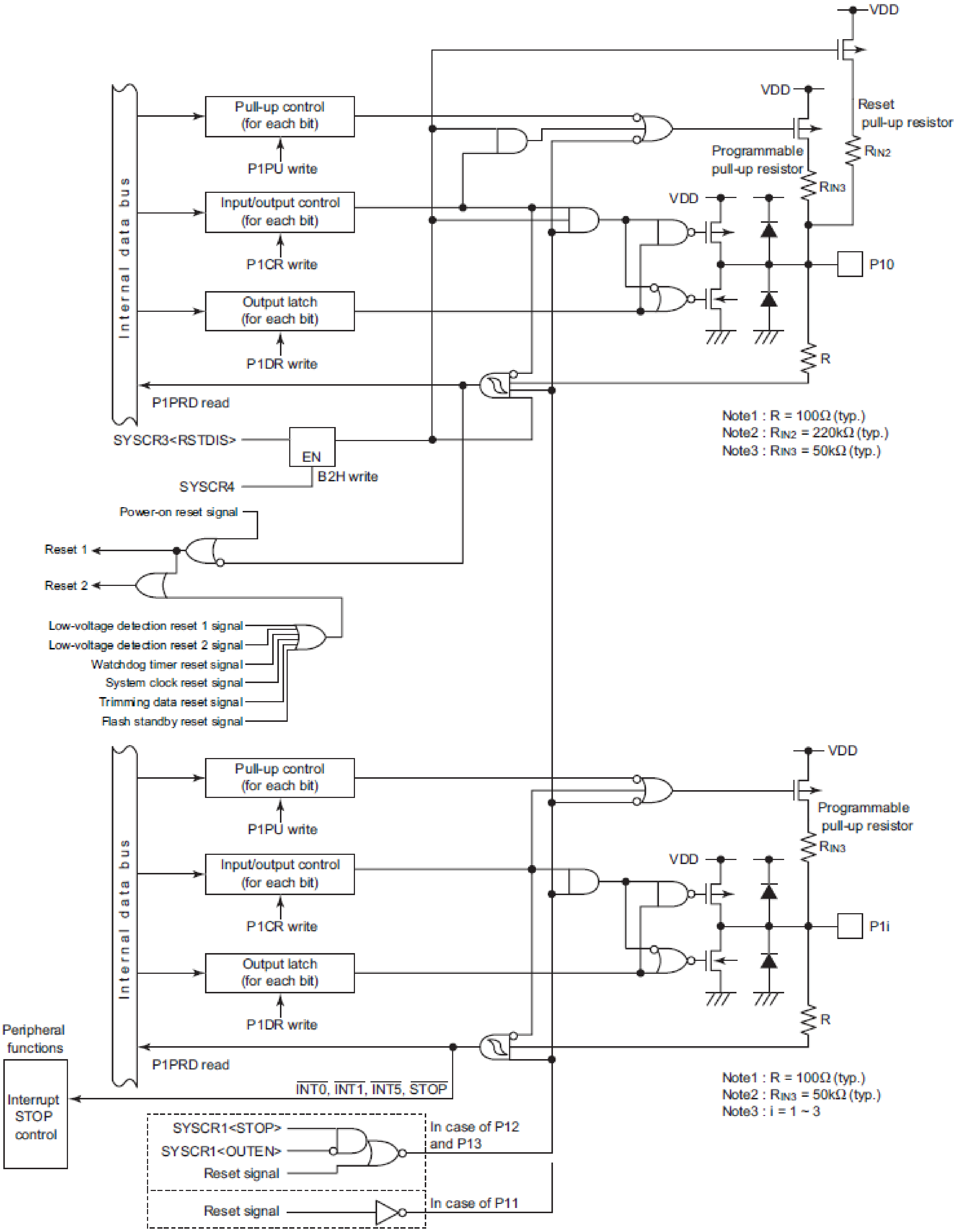


Figure 8.4 Port P1

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Port P1 Output Latch Register

P1DR (0x0001)		7	6	5	4	3	2	1	0
BitSymbol		-	-	-	-	P13	P12	P11	P10
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:					Outputs L level when the output mode is selected.			
	1:					Outputs H level when the output mode is selected.			

Port P1 Input / Output Control Register

P1CR (0x0F1B)		7	6	5	4	3	2	1	0
BitSymbol		-	-	-	-	P1CR3	P1CR2	P1CR1	P1CR0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:					Input mode (port input)			
						INT1 (I)	INT0B (I)	INT5B (I) STOPB (I)	-
	1:					Output mode (port output)			

Port P1 Built-in Pull-up Resistor Control Resistor

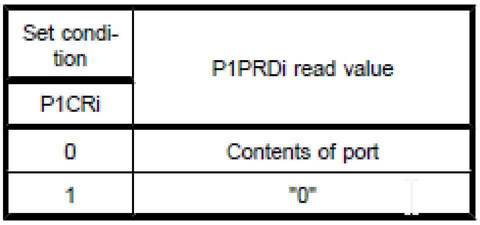
P1PU (0x0F28)		7	6	5	4	3	2	1	0
BitSymbol		-	-	-	-	P1PU3	P1PU2	P1PU1	P1PU0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:					The built-in pull-up resistor is not connected.			
	1:					The built-in pull-up resistor is connected.			

Note): The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected.

Port P1 Input Data Register

P1PRD (0x000E)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	P1PRD3	P1PRD2	P1PRD1	P1PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	*	*	*	*
Function					If the port is in the input mode, the contents of the port are read. If not, "0" is read.			

Note: "" means "don't care".*



Set condition	P1PRDi read value
P1CRi	
0	Contents of port
1	"0"

Note: i=0 to 3

8.3.3 Port P2 (P27 to P20) Register

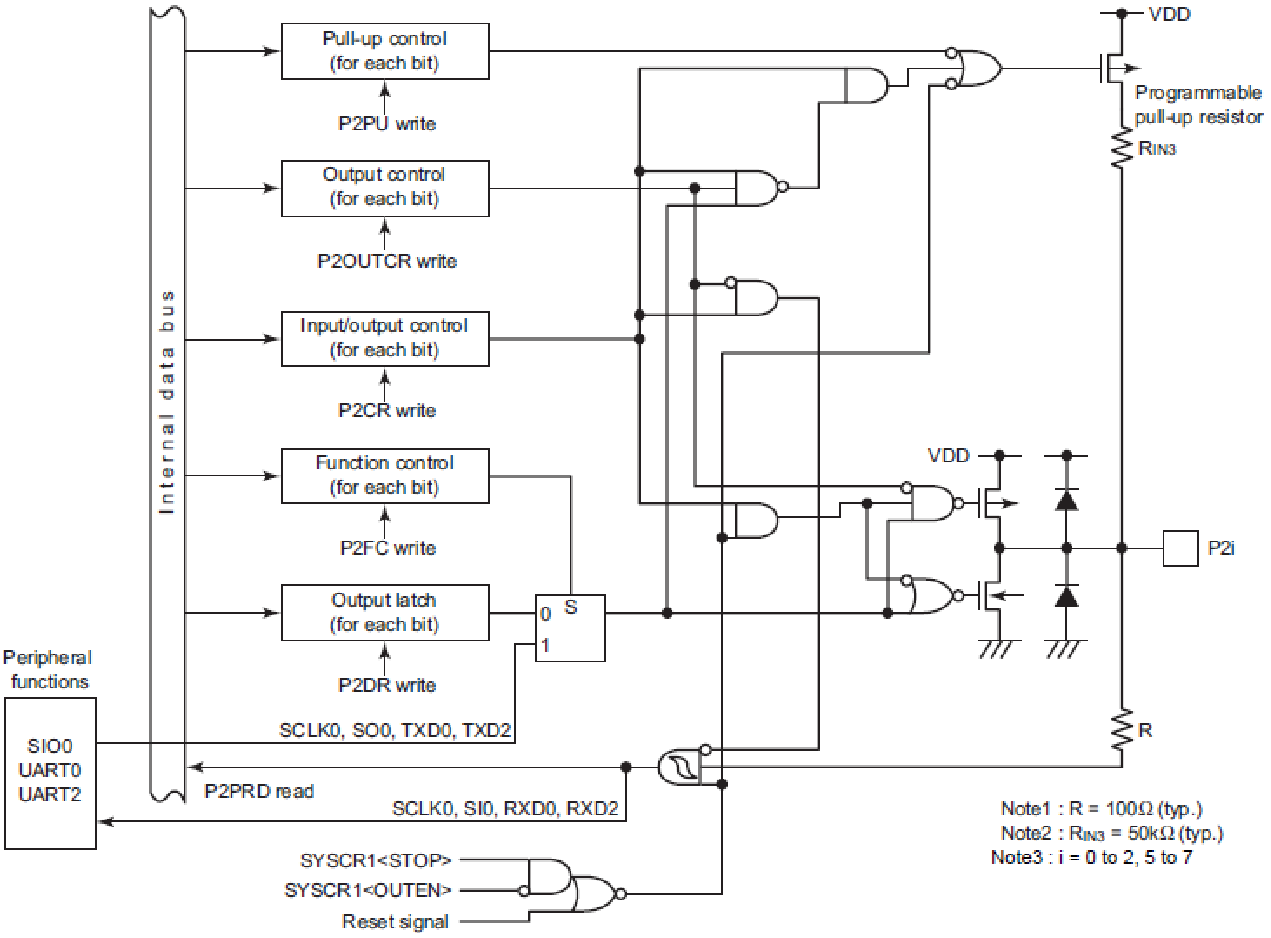
Port P2 is a 8-bit input / output port that can be set to input or output for each bit individually.

The output circuit has the P-channel output control function and either the sink open drain output or the CMOS output can be selected. Port P2 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode or as a sink open drain output.

Note: Please prevent P23 and P24 from being floated otherwise STOP mode current will be affected by additional leakage current in IO pads. When P23 and P24 need to pull-high, P23 and P24 should connect to a resistor 1Kohm.

Port Name	P27	P26	P25	P24	P23	P22	P21	P20
Secondary function	RXD2 TXD2	TXD2 RXD2	SCLK0	SI0 SCL0	SO0 SDA0	SCLK0	SI0 RXD0 TXD0 OCDIO	SO0 TXD0 RXD0 OCDCK

Table 8.5 Port P2



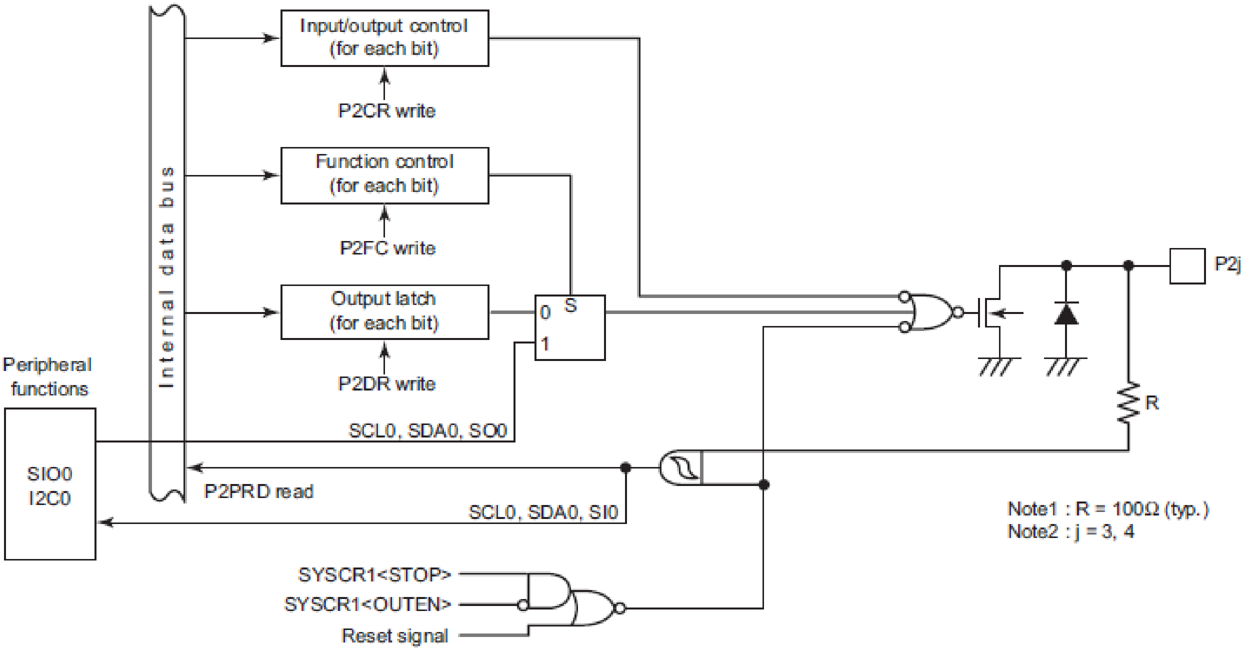


Figure 8.5 Port P2

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Port P2 Output Latch Register

P2DR (0x0002)		7	6	5	4	3	2	1	0
Bit Symbol		P27	P26	P25	P24	P23	P22	P21	P20
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected, which serves as Hi-Z or pull-up depending on settings of P2OUTCR and P2PU.							

Port P2 Input / Output Control Register

P2CR (0x0F1C)		7	6	5	4	3	2	1	0
Bit Symbol		P2CR7	P2CR6	P2CR5	P2CR4	P2CR3	P2CR2	P2CR1	P2CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
		RXD2 (I)	RXD2 (I)	SCLK0(I)	SI0 (I)	-	SCLK0 (I)	RXD0(I) SI0 (I)	RXD0 (I)
	1:	Output mode (port output)							
		TXD2 (O)	TXD2 (O)	SCLK0 (O)	SCL0 (I/O)	SDA0 (I/O) SO(O)	SCLK0 (O)	TXD0 (O)	TXD0 (O) SO0 (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output. Symbol "I/O" means secondary function input/output

Port P2 Function Control Register

P2FC (0x0F36)		7	6	5	4	3	2	1	0
Bit Symbol		P2FC7	P2FC6	P2FC5	P2FC4	P2FC3	P2FC2	P2FC1	P2FC0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Port function							
	1:	TXD2 (O)	TXD2 (O)	SCLK0 (O)	SCL0 (I/O)	SDA0 (I/O) SO0 (O)	SCLK0 (O)	TXD0(O)	TXD0(O) SO0 (O)

Port P2 Output Control Register

P2OUTCR (0x0F43)		7	6	5	4	3	2	1	0
Bit Symbol		P2OUT7	P2OUT6	P2OUT5	-	-	P2OUT2	P2OUT1	P2OUT0
Read/Write		R/W	R/W	R/W	R	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	CMOS output					CMOS output		
	1:	Open-drain output					Open-drain output		

Port P2 Built-in Pull-up Resistor Control Resistor

P2PU (0x0F29H)		7	6	5	4	3	2	1	0
Bit Symbol		P2PU7	P2PU6	P2PU5	-	-	P2PU2	P2PU1	P2PU0
Read/Write		R/W	R/W	R/W	R	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up resistor is not connected.					The built-in pull-up resistor is not connected.		
	1:	The built-in pull-up resistor is connected. (Note.)					The built-in pull-up resistor is connected. (Note.)		

Note. : The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions, setting to "1" does not make the resistor connected.

Port P2 Input Data Register

P2PRD (0x000F)	7	6	5	4	3	2	1	0
Bit Symbol	P2PRD7	P2PRD6	P2PRD5	P2PRD4	P2PRD3	P2PRD2	P2PRD1	P2PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.			The contents of the port are read without condition.		If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.		

Set Condition		P2PRDi read value
P2CRi	P2OUTCRi	
0	*	Contents of port
1	0	“0”
1	1	Contents of port

Table 8.6 P2PRD Read Value (P20 to P22, P25 to P27)

*Note 1): * : Don’t care*

Note 2): i= 0 to 2, 5 to 7.

8.3.4 Port P4 (P47 to P40) Register

Port P4 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the key-on wakeup input.

Except P46, Port P4 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode.

Port Name	P47	P46	P45	P44	P43	P42	P41	P40
Secondary function	AIN7 KWI7 EMG0B	AIN6 KWI6 TCC	AIN5 KWI5 PPGC02B	AIN4 KWI4 PPGC01B	AIN3 KWI3	AIN2 KWI2	AIN1 KWI1	AIN0 KWI0

Table 8.7 Port P4

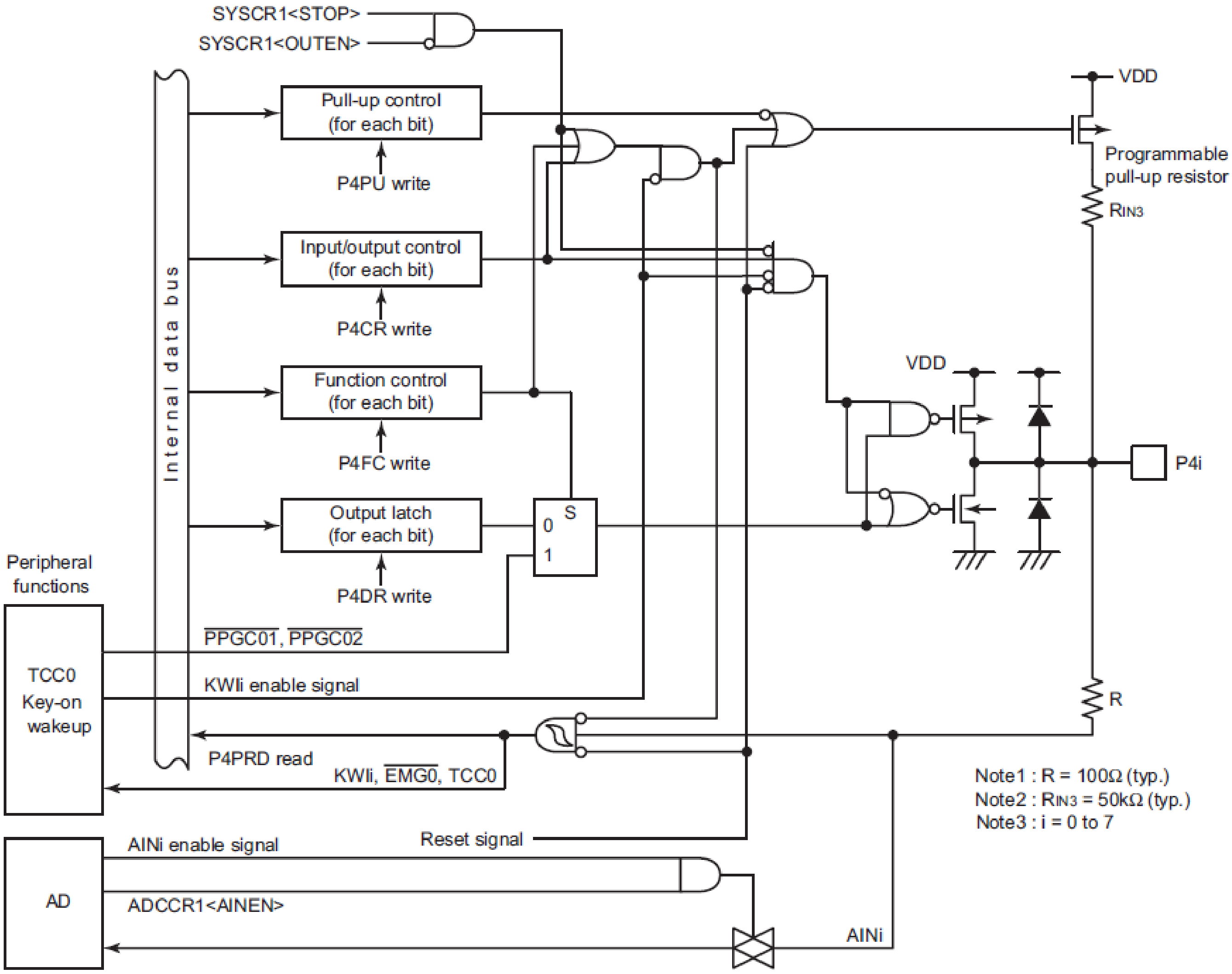


Figure 8.6 Port P4

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Port P4 Output Latch Register

P4DR (0x0004)		7	6	5	4	3	2	1	0
BitSymbol		P47	P46	P45	P44	P43	P42	P41	P40
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected.							

Port P4 Input / Output Control Register

P4CR (0x0F1E)		7	6	5	4	3	2	1	0
BitSymbol		P4CR7	P4CR6	P4CR5	P4CR4	P4CR3	P4CR2	P4CR1	P4CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
		AIN7 (I) EMG0B(I)	AIN6 (I) TCC0(I)	AIN5 (I)	AIN4 (I)	AIN3 (I)	AIN2 (I)	AIN1 (I)	AIN0 (I)
	1:	Output mode (port output)							
		-	-	PPGC02B(O)	PPGC01B(O)				

Note): Symbol "I" means secondary function input.

Port P4 Function Control Register

P4FC (0x0F38)		7	6	5	4	3	2	1	0
BitSymbol		P4FC7	P4FC6	P4FC5	P4FC4	P4FC3	P4FC2	P4FC1	P4FC0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:	Port function							
	1:	AIN7 (I)	AIN6 (I)	AIN5 (I) PPGC02B(O)	AIN4 (I) PPGC01B(O)	AIN3 (I)	AIN2 (I)	AIN1 (I)	AIN0 (I)

Note 1: When the key-on wakeup input (KWIi) is enabled, there is no need to set P4FCi.

Port P4 Built-in Pull-up Resistor Control Register

P4PU (0x0F2B)		7	6	5	4	3	2	1	0
BitSymbol		P4PU7	P4PU6	P4PU5	P4PU4	P4PU3	P4PU2	P4PU1	P4PU0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Afterreset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up resistor is not connected.							
	1:	The built-in pull-up resistor is connected. (Note)							

Note): The resistor is connected only when the key-on wakeup input (KWIi) is enabled or the port is used in the input mode (P4FCi="0" and P4CRi="0"). Under any other conditions, setting to "1" does not make the resistor connected.

Port P4 Input Data

P4PRD (0x0011)	7	6	5	4	3	2	1	0
Bit Symbol	P4PRD7	P4PRD6	P4PRD5	P4PRD4	P4PRD3	P4PRD2	P4PRD1	P4PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	0
Function	If the port is in the input mode, the contents of the port are read. If not, "0" is read.							

Set Condition		P4PRDi read value
P4CRi	P4FCi	
0	0	Contents of port
*	1	“0”
1	*	“0”

Table 8.8 P4PRD Read Value

*Note 1): * : Don't care*

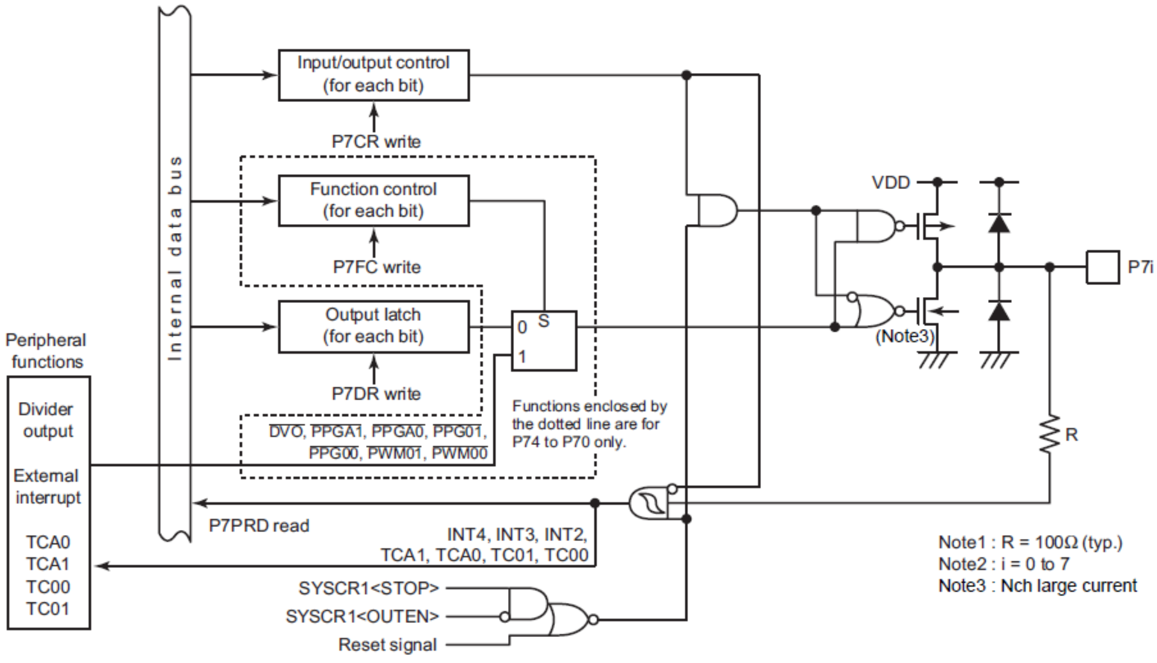
Note 2): i= 0 to 7

8.3.5 Port P7 (P77 to P70) Register

Port P7 is a 8-bit input / output port that can be set to input or output for each bit individually, and it is also used as the external interrupt input, the divider output and the timer counter input/output.

	P77	P76	P75	P74	P73	P72	P71	P70
Secondary function	INT4	INT3	INT2	DVOB	PPGA1B TCA1	PPGA0B TCA0	PPG01B PWM01B TC01	PPG00B PWM00B TC00

Table 8.9 Port P7



Note1 : R = 100Ω (typ.)
Note2 : i = 0 to 7
Note3 : Nch large current

Figure 8.7 Port P7

Port P7 Output Latch Register

P7DR (0x0007)	7	6	5	4	3	2	1	0
Bit Symbol	P77	P76	P75	P74	P73	P72	P71	P70
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function 0:	Outputs L level when the output mode is selected.							
1:	Outputs H level when the output mode is selected.							

Port P7 Input / Output Control Register

P7CR (0x0F21)	7	6	5	4	3	2	1	0
BitSymbol	P7CR7	P7CR6	P7CR5	P7CR4	P7CR2	P7CR2	P7CR1	P7CR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Afterreset	0	0	0	0	0	0	0	0
0:	Input mode (port input)							
	INT4 (I)	INT3 (I)	INT2 (I)	–	TCA1 (I)	TCA0 (I)	TC01 (I)	TC00 (I)
1:	Output mode (port output)							
	–	–	–	DVOB (O)	PPGA1B (O)	PPGA0B (O)	PPG01B (O) PMW01B (O)	PPG00B (O) PMW00B (O)

Note): Symbol "I" means secondary function input. Symbol "O" means secondary function output.

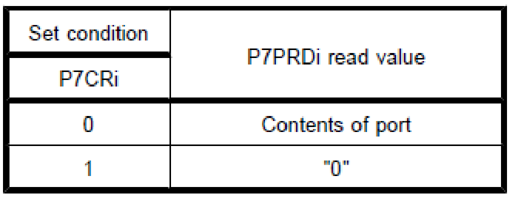
Port P7 Function Control Register

P7FC (0x0F3B)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	P7FC4	P7FC3	P7FC2	P7FC1	P7FC0
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Afterres et	0	0	0	0	0	0	0	0
Function 0:				Port function				
1:				DVOB (O)	PPGA1B (O)	PPGA0B (O)	PPG01B (O) PWM01B (O)	PPG00B (O) PWM00B (O)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Port P7 Input Data

P7PRD (0x0014)	7	6	5	4	3	2	1	0
Bit Symbol	P7PRD7	P7PRD6	P7PRD5	P7PRD4	P7PRD3	P7PRD2	P7PRD1	P7PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is in the input mode, the contents of the port are read. If not, "0" is read.							



Set condition	P7PRDi read value
P7CRi	
0	Contents of port
1	"0"

Table 8.10 P7RRD Read Value

Note: i=0 to 7

8.3.6 Port P8 (P81 to P80) Register

Port P8 is a 2-bit input/output port that can be set to input or output for each bit individually, and it is also used as the timer counter input/output.

Port Name	P81	P80
Secondary function	PPG03B PWM03B TC03	PPG02B PWM02B TC02

Table 8.11 Port P8

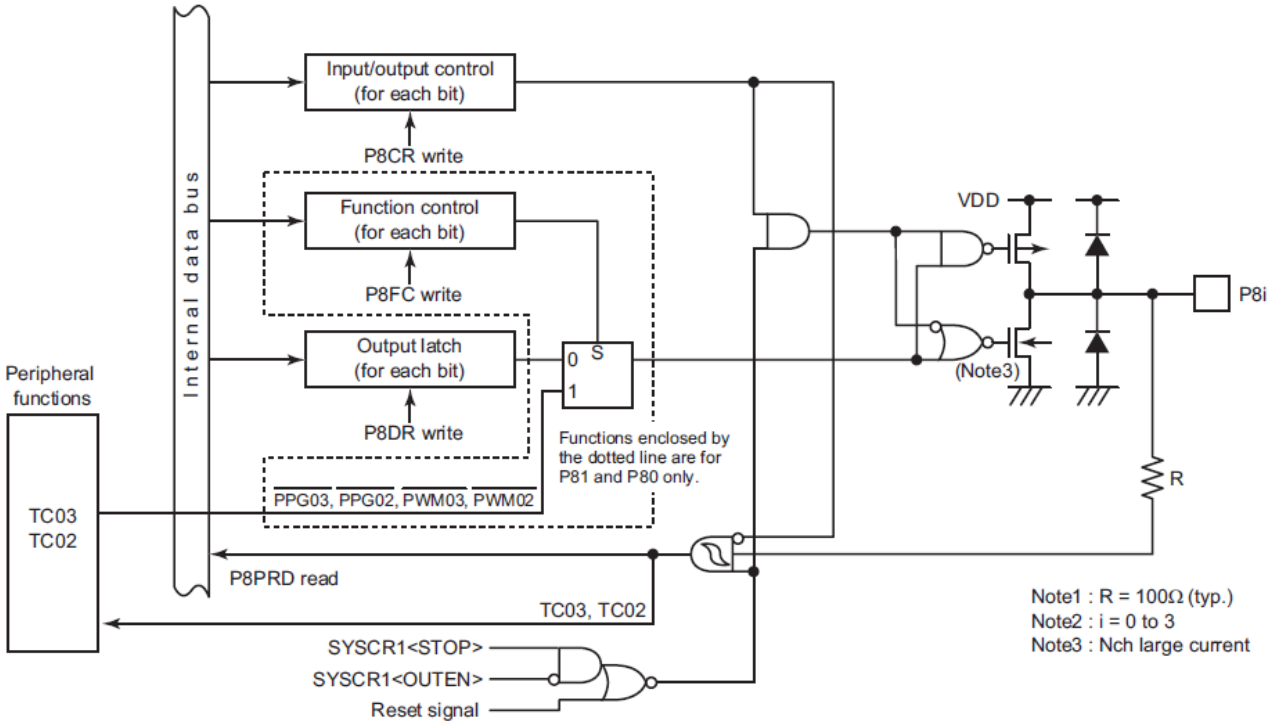


Figure 8.8 Port P8

Port P8 Output Latch Register

P8DR (0x0008)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P81	P80
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Outputs L level when the output mode is selected	
	1:							Outputs H level when the output mode is selected	

Port P8 Input / Output Control Register

P8CR (0x0F22)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P8CR1	P8CR0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Input mode (port input)	
								TC03 (I)	TC02 (I)
	1:							Output mode (port output)	
								PPG03B (O) PWM03 (O)	PPG02B (O) PWM02 (O)

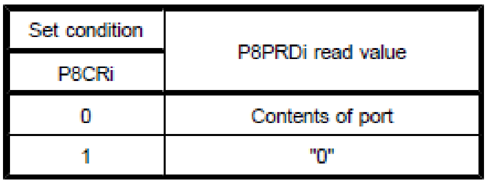
Port P8 Function Control Register

P8FC (0x0F3C)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P8FC1	P8FC0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Port function	
	1:							PPG03B (O) PWM03 (O)	PPG02B (O) PWM02 (O)

Port P8 Input Data Register

P8PRD (0x0015)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	P8PRD1	P8PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*
Function							If the port is in the input mode, the contents of the port are read. If not, "0" is read.	

Note : "" means "don't care".*



Set condition	P8PRDi read value
P8CRi	
0	Contents of port
1	"0"

Table 8.11 P8PRD Read Value

8.3.7 Port P9 (P91 to P90)

Port P9 is a 2-bit input/output port that can be set to input or output for each bit individually, and it is also used as the UART.

The output circuit has the P-channel output control function and either the sink open drain output or the CMOS output can be selected. Port P9 contains a programmable pull-up resistor on the VDD side. This pullup resistor can be used when the port is used in the input mode or as a sink open drain output.

Port Name	P91	P90
Secondary function	RXD1 TXD1	TXD1 RXD1

Table 8.12 Port P9

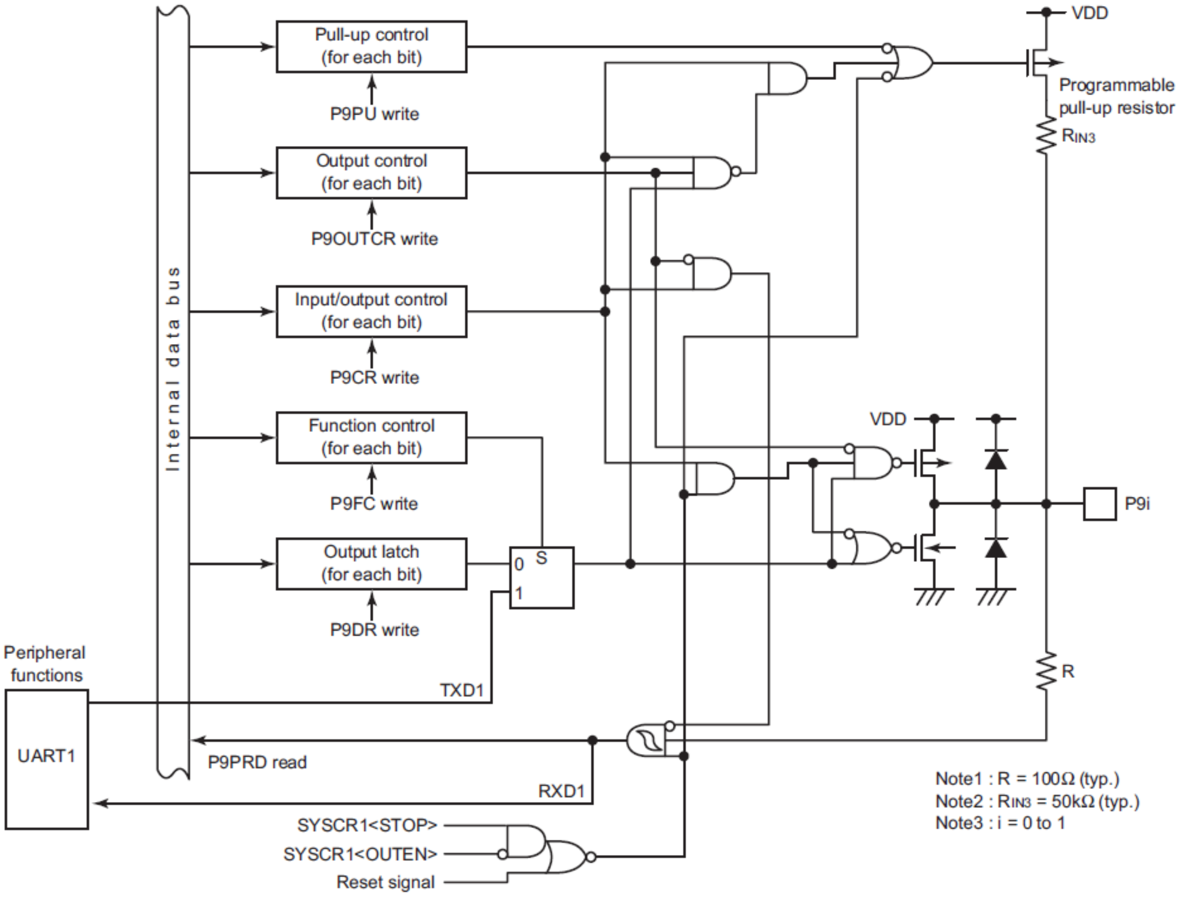


Figure 8.9 Port P9

Port P9 Output Latch Register

P9DR (0x0009)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P91	P90
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Outputs L level when the output mode is selected	
	1:							Outputs H level when the output mode is selected. (Note.)	

Note.: Serves as Hi-Z or pull-up depending on settings of P9OUTCR and P9PU.

Port P9 Input / Output Control Register

P9CR (0x0F23)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9CR1	P9CR0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Input mode (port input)	
								RXD1 (I)	RXD1 (I)
	1:							Output mode (port output)	
								TXD1(O)	TXD1(O)

Port P9 Function Control Register

P9FC (0x0F3D)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9FC1	P9FC0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							Port function	
	1:							TXD1(O)	TXD1(O)

Port P9 Output Control

P9OUTCR (0x0F4A)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9OUT1	P9OUT0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							C-MOS output	
	1:							Open drain output	

Port P9 built-in pull-up resistor control

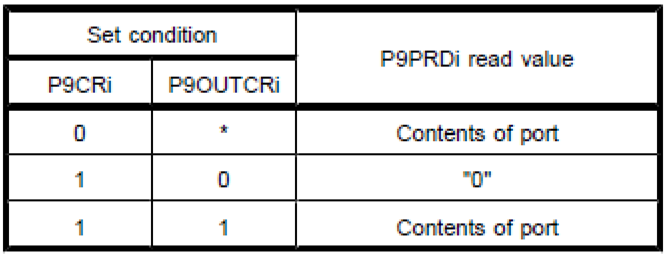
P9PU (0x0F30)		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	P9PU1	P9PU0
Read/Write		R	R	R	R	R	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:							The built-in pull-up resistor is not connected.	
	1:							(Note)	

Note: The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions, setting to "1" does not make the resistor connected.)

Port P9 Input Data Register

P9PRD (0x0016)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	P9PRD1	P9PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*
Function							If the port is used in the input mode or as the sink open drain output, the contents of the port are read. If not, "0" is read.	

Note : "" means "don't care".*



Set condition		P9PRDi read value
P9CRi	P9OUTCRi	
0	*	Contents of port
1	0	"0"
1	1	Contents of port

Note: i=0 to 1

Table 8.13 P9PRD Read Value

8.3.8 Port PB (PB7 to PB4)

Port PB is a 4-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial interface input/output and the UART input/output.

The output circuit has the P-channel output control function and either the sink open drain output or the CMOS output can be selected.

Port Name	PB7	PB6	PB5	PB4
Secondary function	-	SCLK0	SI0 RXD0 TXD0	SO0 TXD0 RXD0

Table 8.14 Port PB

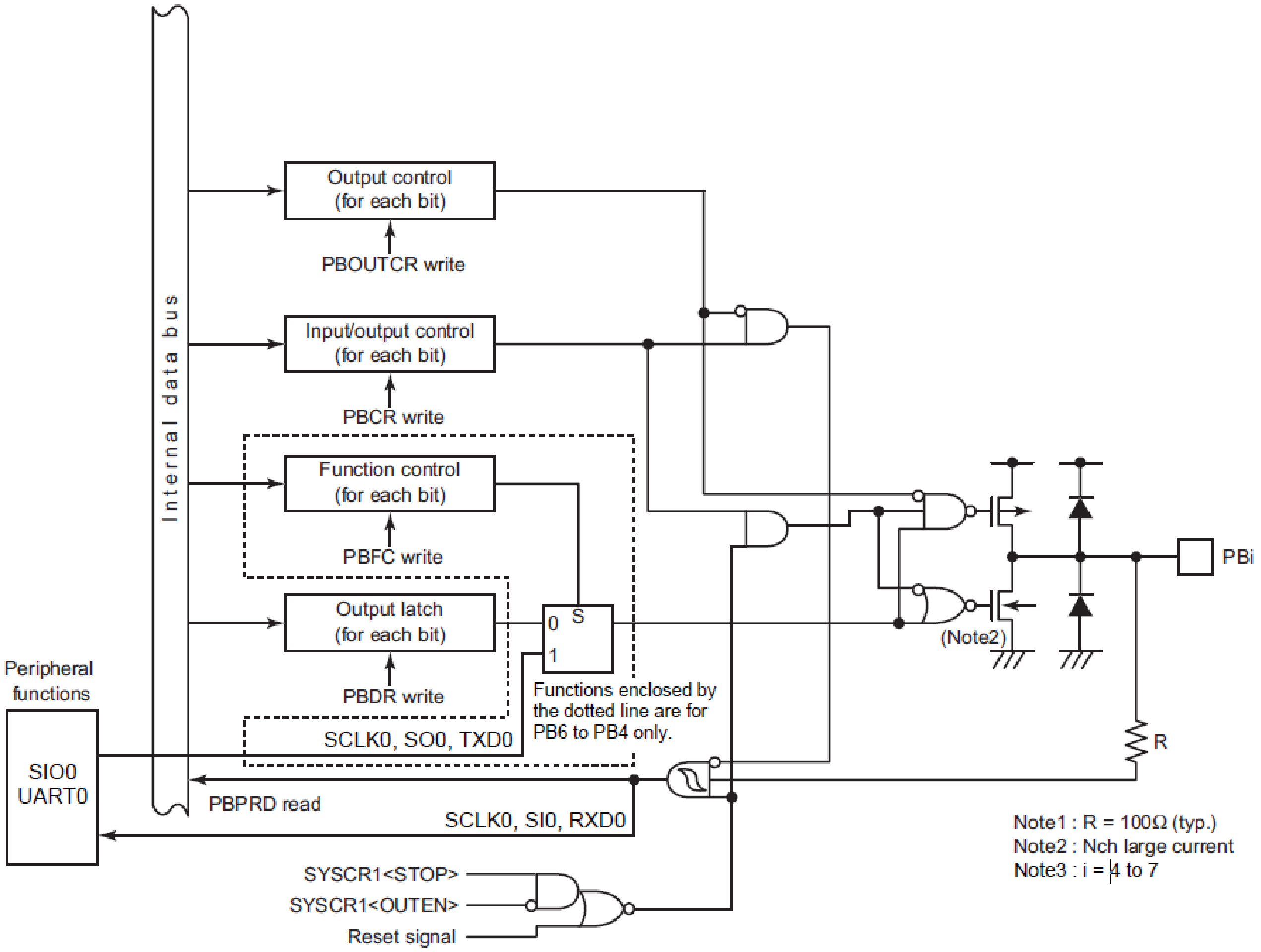


Figure 8.10 Port PB

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Port PB Output Latch Register

PBDR (0x000B)		7	6	5	4	3	2	1	0
Bit Symbol		PB7	PB6	PB5	PB4	-	-	-	-
Read/Write		R/W	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected.							

Port PB Input / Output Control Register

PBCR (0x0F25)		7	6	5	4	3	2	1	0
Bit Symbol		PBCR7	PBCR6	PBCR5	PBCR4	-	-	-	-
Read/Write		R/W	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
	1:	Output mode (port output)							

Port PB Function Control Register

PBFC (0x0F3F)		7	6	5	4	3	2	1	0
Bit Symbol		-	PBFC6	PBFC5	PBFC4	-	-	-	-
Read/Write		R	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:		Port function						
	1:		SCLK0 (O)	TXD0(O)	TXD0(O) SO0 (O)				

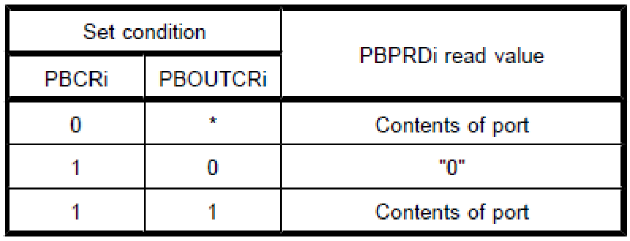
Port PB Output Control

PBOUTCR (0x0F4C)		7	6	5	4	3	2	1	0
Bit Symbol		PBOUT7	PBOUT6	PBOUT5	PBOUT4	-	-	-	-
Read/Write		R/W	R/W	R/W	R/W	R	R	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:	C-MOS output							
	1:	Open drain output							

Port PB Input Data Register

PBPRD (0x0018)	7	6	5	4	3	2	1	0
Bit Symbol	PBPRD7	PBPRD6	PBPRD5	PBPRD4	-	-	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.							

Note : "" means "don't care".*



Set condition		PBPRDi read value
PBCRi	PBOUTCRi	
0	*	Contents of port
1	0	"0"
1	1	Contents of port

*Note 1: * : Don’t care*

Note 2: i = 4 to 7

Table 8.15 PBPRD Read Value

8.4 Serial Interface Selecting Function

In MQ6935, the built-in serial interface (SIO, UART and I2C) communication pins and interrupt source assignment can be changed. Two out of three functions, SIO0, UART0 and I2C0, can be used at the same time by using this selecting function.

The input pins of the 16-bit timer counter A0 input (TCA0 input) can be changed by using this selecting function.

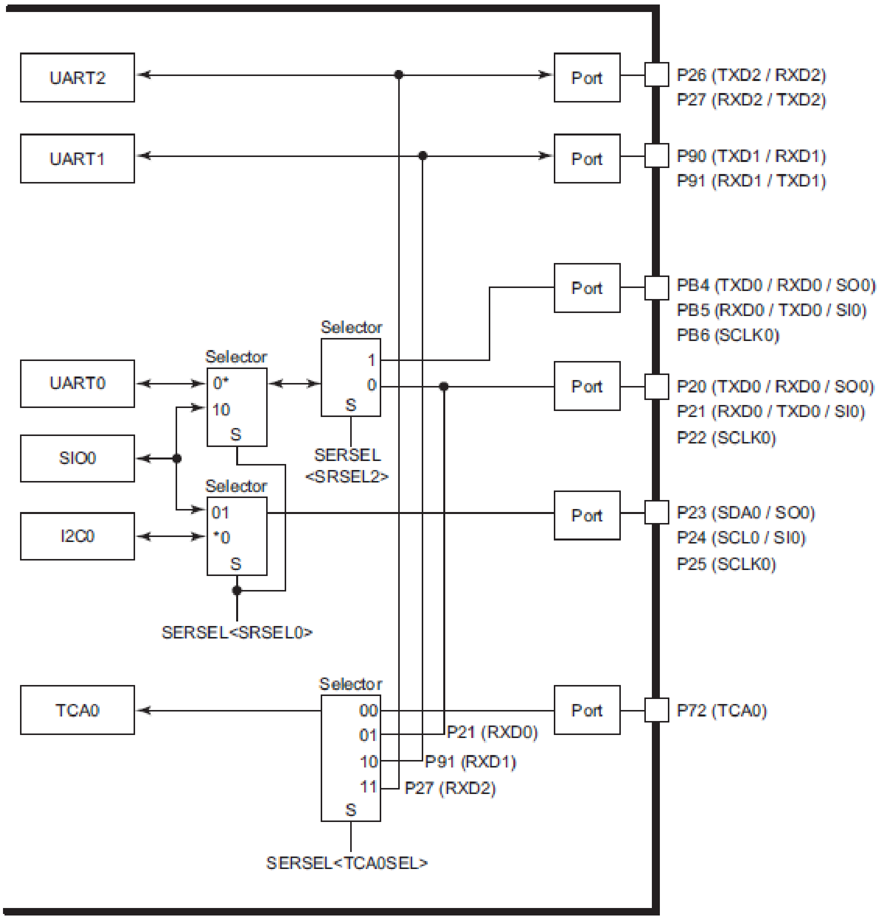


Figure 8.11 Serial Interface Selecting Function

Peripheral Function Input Selection Control Register

ITSEL (0x0FCA)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	ITSEL1	ITSEL0
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

ITSEL1	Select KWI7	0: 1:	KWI7 EMG0B
ITSEL0	Select KWI6	0: 1:	KWI6 TCC0

Note 1: The operation for changing ITSEL must be executed while the applicable serial interface and timer counter operations are stopped. If ITSEL is switched during operation of these peripheral functions, each peripheral function may receive (transmit) unexpected data and operate improperly.

Note 2: It is recommended to clear the interrupt latch for the applicable peripheral function immediately after changing ITSEL.

Serial Interface Selection Control Register

SERSEL (0x0FCB)	7	6	5	4	3	2	1	0
Bit Symbol	TCA0SEL		-	SRSEL2	-	-	SRSEL0	
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TCA0SEL	16-bit timer counter A0 input switching	00: 01: 10: 11:	P72 input (TCA0) P21 input (also used as RXD0) P91 input (also used as RXD1) P27 input (also used as RXD2)
SRSEL2	Select UART0/SIO0 input/output port	0: 1:	Select P20, P21, P22 Select PB4, PB5, PB6
SRSEL0	Serial interface selection 0	00: 01: 10: 11:	Select UART0, I2C0 Select UART0, SIO0 Select SIO0, I2C0 Reserved

Note 1: The operation for changing SERSEL must be executed while the applicable serial interface and timer counter operations are stopped. If SERSEL is switched during operation of these peripheral functions, each peripheral function may receive (transmit) unexpected data and operate improperly.

Note 2: It is recommended to clear the interrupt latch for the applicable serial interface immediately after changing SERSEL. Interrupt latches are common to INTRXD and INTSIO and to INTSBI and INTSIO. Therefore, if an interrupt occurs before or after SERSEL is switched, it is difficult to tell which function has caused the interrupt.

UART Input/Output Change Control Register

UATCNG (0x0E57)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	UAT2IO	UAT1IO	UAT0IO
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

			RXD pin		TXD pin	
UAT2IO	Select UART2 input/ output port	0: 1:	P27 P26		P26 P27	
UAT1IO	Select UART1 input/ output port	0: 1:	P91 P90		P90 P91	
UAT0IO	Select UART0 input/ output port		SERSEL <SERSEL2>="0"	SERSEL <SERSEL2>="1"	SERSEL <SERSEL2>="0"	SERSEL <SERSEL2>="1"
		0: 1:	P21 P20	PB5 PB4	P20 P21	PB4 PB5

Note 1: The operation for changing UATCNG must be executed while the applicable serial interface operations are stopped.

SERSEL <SRSEL0>	SERSEL <SRSEL2>	UATCNG <UAT0IO>	Port									Interrupt		
			UART0/SIO0						I2C0/SIO0					
			PB4	PB5	PB6	P20	P21	P22	P23	P24	P25	IL7	IL6	IL15
00:	0:	0:	Note 1	Note 1	Note 1	TXD0	RXD0	Note 1	SDA0	SCL0	Note 1	INTTXD0	INTRXD0	INTSBI0
		1:				RXD0	TXD0							
	1:	0:	TXD0	RXD0	Note 1	Note 1	Note 1	Note 1						
		1:	RXD0	TXD0										
01:	0:	0:	Note 1	Note 1	Note 1	TXD0	RXD0	Note 1	SO0	SI0	SCLK 0	INTTXD0	INTRXD0	INTSIO0
		1:				RXD0	TXD0							
	1:	0:	TXD0	RXD0	Note 1	Note 1	Note 1	Note 1						
		1:	RXD0	TXD0										
10:	0:	0 or 1:	Note 1	Note 1	Note 1	SO0	SI0	SCLK 0	SDA0	SCL0	Note 1	-	INTSIO0	INTSBI0
	1:	0 or 1:	SO0	SI0	SCLK 0	Note 1	Note 1	Note 1						
11:	0 or 1:	0 or 1:	Reserved											

Note 1: Can be used as a port. (Set the function register (PxFC) to "0".)

Table 8.16: Select input/output port and interrupt

9. 10-bit AD Converter (ADC)

MQ6935 has a real 10-bit AD converter (ADC), which is a successive approximation type ADC.

9.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 9.1.

It consists of control registers ADCCR1 and ADCCR2, converted value registers ADCDRL and ADCDRH, a DA converter, a sample-hold circuit, a comparator, a successive comparison circuit, etc.

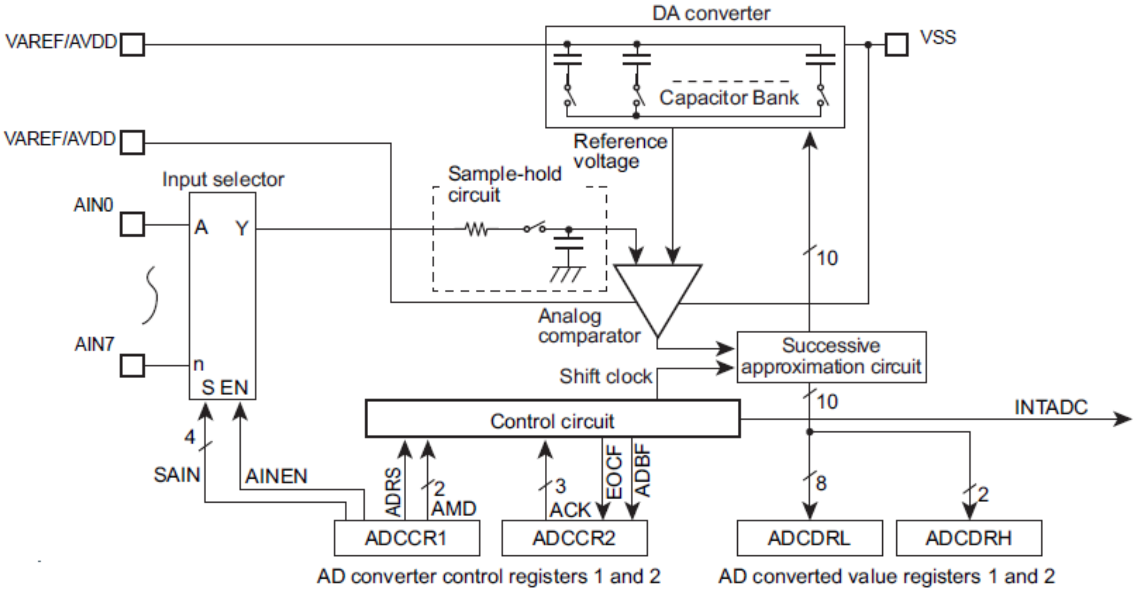


Figure 9.1 10-bit AD Converter

Note 1): Before using the AD converter, set an appropriate value to the I/O port register which is also used as an analog input port. For details, see the section on "8 I/O ports".

9.2 Control

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)
   This register selects an analog channel in which to perform AD conversion, selects an AD conversion operation mode, and controls the start of the AD converter.
2. AD converter control register 2 (ADCCR2)
   This register selects the AD conversion time, and monitors the operating status of the AD converter.
3. AD converter reference voltage register (ADCVRF)
   The register selects the reference voltage source of the AD converter.
4. AD converted value registers (ADCDRH and ADCDRL)
   These registers store the digital values generated by the AD converter.

AD Converter Control Register 1

ADCCR1 (0x0034H)	7	6	5	4	3	2	1	0
Bit Symbol	ADRS	AMD		AINEN	SAIN			
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

ADRS	AD conversion start	0: 1:	- AD conversion start
AMD	AD operating mode	00: 01: 10: 11:	AD operation disable, forcibly stop AD operation Single mode Reserved Repeat mode
AINEN	Analog input control	0: 1:	Analog input disable Analog input enable
SAIN	Analog input channel select	0000 0001: 0010: 0011: 0100: 0101: 0110: 0111: 1000 1001 1010 1011 1100 Others	AIN0 AIN1 AIN2 AIN3 AIN4 AIN5 AIN6 AIN7 Reserved Reserved Reserved Reserved Reserved -

Note 1): Do not perform the following operations on the ADCCR1 register while AD conversion is being executed (ADCCR2 <ADBF>="1").

- Changing SAIN

- Setting AINEN to "0"

- Changing AMD (except a forced stop by setting AMD to "00")

- Setting ADRS to "1"

Note 2): If you want to disable all analog input channels, set AINEN to "0".

Note 3): Although analog input pins are also used as input/output ports, it is recommended for the purpose of maintaining the accuracy of AD conversion that you do not execute input/output instructions during AD conversion. Additionally, do not input widely varying signals into the ports adjacent to analog input pins.

Note 4): When STOP, IDLE0 or SLOW mode is started, ADRS, AMD and AINEN are initialized to "0". If you use the AD converter after returning to NORMAL mode, you must reconfigure ADRS, AMD and AINEN.

Note 5): After the start of AD conversion, ADRS is automatically cleared to "0" ("0" is read).

AD Converter Control Register 2

ADCCR2 (0x0035)	7	6	5	4	3	2	1	0
Bit Symbol	EOCF	ADBF	-	-	"0"	ACK		
Read/Write	R	R	R	R	W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

EOCF	AD conversion end flag	0: 1:	Before conversion or during conversion Conversion end
ADBF	AD conversion BUSY flag	0: 1:	AD conversion being halted AD conversion being executed
ACK	AD conversion time select		Refer to Table 9.1 for AD conversion time

Note 1): Make sure that you make the ACK setting when AD conversion is in a halt condition (ADCCR2 <ADBF>="0").

Note 2): Make sure that you write "0" to bit 3 of ADCCR2.

Note 3): If STOP, IDLE0 or SLOW mode is started, EOCF and ADBF are initialized to "0".

Note 4): If the AD converted value register (ADCDRH) is read, EOCF is cleared to "0". It is also cleared to "0" if AD conversion is started (ADCCR1 <ADRS>="1") without reading ADCDRH after completing AD conversion in single mode.

Note 5): If an instruction to read ADCCR2 is executed, 0 is read from bits 3 through 5.

		Frequency (fcgck)									
ACK setting	Conversion time	16MHz	10MHz	8MHz	5MHz	4MHz	2.5MHz	2MHz	1MHz	0.5MHz	0.25 MHz
000	32/fcgck	-	-	-	-	-	12.8 μs	16.0 μs	32.0 μs	64.0 μs	128.0 μs
001	64/fcgck	-	-	-	12.8 μs	16.0 μs	25.6 μs	32.0 μs	64.0 μs	128.0 μs	-
010	128/fcgck	-	12.8 μs	16.0 μs	25.6 μs	32.0 μs	51.2 μs	64.0 μs	128.0 μs	-	-
011	256/fcgck	16.0 μs	25.6 μs	32.0 μs	51.2 μs	64.0 μs	102.4 μs	128.0 μs	-	-	-
100	512/fcgck	32.0 μs	51.2 μs	64.0 μs	102.4 μs	128.0 μs	-	-	-	-	-
101	1024/fcgck	64.0 μs	102.4 μs	128.0 μs	-	-	-	-	-	-	-
11*	Reserved										

Table 9.1 ACK Settings and Conversion Times Relative to Frequencies

Note 1): Spaces indicated by "-" in the above table mean that it is prohibited to establish conversion times in these spaces. fcgck: High Frequency oscillation clock [Hz]

Note 2): The conversion time must be longer than the following time by analog reference voltage (VAREF)

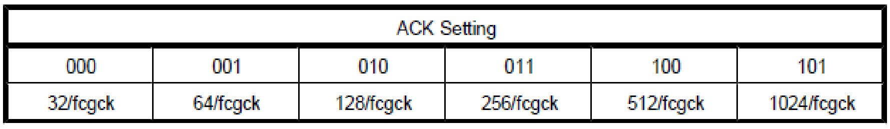
- VAREF = 2.7 to 5.5V 12.8μs or longer.

- VAREF = 2.0 to 2.7V 25.6μs or longer.

Note 3): Above conversion times do not include the time shown below.

- Time from when ADCCR1<ADRS> is set to 1 to when AD conversion is started

- Time from when AD conversion is finished to when a converted value is stored in ADCDRL and ADCDRH.
Please refer to below table for longest wakeup time vs. ACK setting



ACK Setting					
000	001	010	011	100	101
32/fcgck	64/fcgck	128/fcgck	256/fcgck	512/fcgck	1024/fcgck

Table 9.2 ADC Wakeup Time vs. ACK Setting

AD Converted Value Register (Lower Side)

ADCDRL (0x0036)	7	6	5	4	3	2	1	0
Bit Symbol	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

AD Converted Value Register (Upper Side)

ADCDRH (0x0037)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	AD09	AD08
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Note 1): A read of ADCDRL or ADCDRH must be read after the INTADC interrupt is generated or after ADCCR2 <EOCF> becomes "1".

Note 2): In single mode, do not read ADCDRL or ADCDRH during AD conversion (ADCCR2 <ADBF>="1"). (If AD conversion is finished in the interim between a read of ADCDRL and a read of ADCDRH, the INTADC interrupt request is canceled, and the conversion result is lost.)

Note 3): If STOP, IDLE0 or SLOW mode is started, ADCDRL and ADCDRH are initialized to "0".

Note 4): If ADCCR1<AMD> is set to "00", ADCDRL and ADCDRH are initialized to "0".

Note 5): If an instruction to read ADCDRH is executed, "0" is read from bits 7 through 2.

Note 6): If AD conversion is finished in repeat mode in the interim between a read of ADCDRL and a read of ADCDRH, the previous converted value is retained without overwriting the AD converted value register. In this case, the INTADC interrupt request is canceled, and the conversion result is lost.

9.3 Function

The 10-bit AD converter operates in either single mode in which AD conversion is performed only once or repeat mode in which AD conversion is performed repeatedly.

9.3.1 Single Mode

In single mode, the voltage at a designated analog input pin is AD converted only once.

Setting ADCCR1 <ADRS> to "1" after setting ADCCR1 <AMD> to "01" allows AD conversion to start. ADCCR1 <ADRS> is automatically cleared after the start of AD conversion. As AD conversion starts, ADCCR2 <ADBF> is set to "1". It is cleared to "0" if AD conversion is finished or if AD conversion is forced to stop.

After AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2 <EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. The AD converted value registers (ADCDRL and ADCDRH) should be usually read according to the INTADC interrupt processing routine. If the upper side (ADCDRH) of the AD converted value register is read, ADCCR2 <EOCF> is cleared to "0".

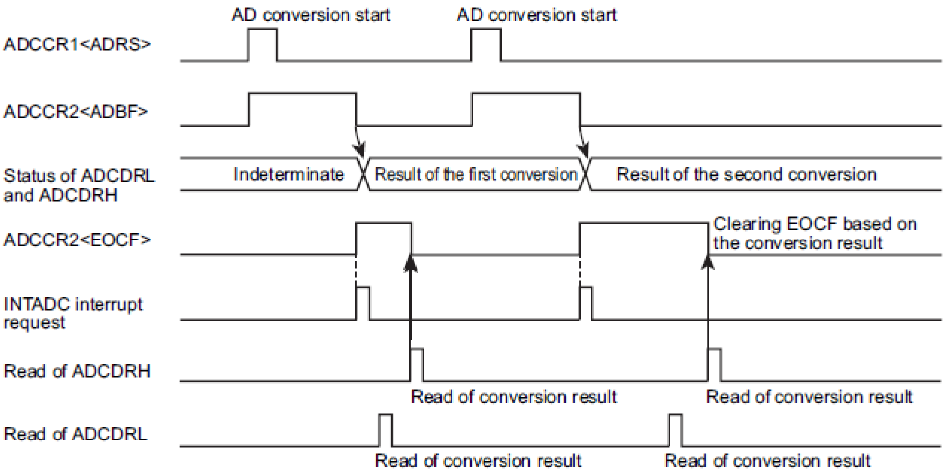

AD conversion start
AD conversion start
ADCCR1<ADRS>
ADCCR2<ADBF>
Status of ADCDRL and ADCDRH
Indeterminate
Result of the first conversion
Result of the second conversion
ADCCR2<EOCF>
Clearing EOCF based on the conversion result
INTADC interrupt request
Read of ADCDRH
Read of conversion result
Read of conversion result
Read of ADCDRL
Read of conversion result
Read of conversion result

Figure 9.2 Single Mode

Note): Do not perform the following operations on the ADCCR1 register when AD conversion is being executed (ADCCR2 <ADBF>="1"). If the following operations are performed, there is the possibility that AD conversion may not be executed properly.

- *Changing the ADCCR1<SAIN> setting*
- *Setting ADCCR1<AINEN> to "0"*
- *Changing the ADCCR1<AMD> setting (except a forced stop by setting AMD to "00")*
- *Setting ADCCR1<ADRS> to "1"*

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

9.3.2 Repeat Mode

In repeat mode, the voltage at an analog input pin designated at ADCCR1<SAIN> is AD converted repeatedly. Setting ADCCR1 <ADRS> to "1" after setting ADCCR1 <AMD> to "11" allows AD conversion to start.

After the start of AD conversion, ADCCR1 <ADRS> is automatically cleared. After the first AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2 <EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. After this interrupt is generated, the second (next) AD conversion starts immediately.

The AD converted value registers (ADCDRL and ADDRH) should be read before the next AD conversion is finished. If the next AD conversion is finished in the interim between a read of ADCDRL and a read of ADCDRH, the previous converted value is retained without overwriting the AD converted value registers (ADCDRL and ADCDRH). In this case, the INTADC interrupt request is not generated, and the conversion result is lost. (See Figure 9.3)

To stop AD conversion, write "00" (AD operation disable) to ADCCR1 <AMD>. As "00" is written to ADCCR1 <AMD>, AD conversion stops immediately. In this case, the converted value is not stored in the AD converted value register. As AD conversion starts, ADCCR2 <ADBF> is set to "1". It is cleared to "0" if "00" is written to AMD.

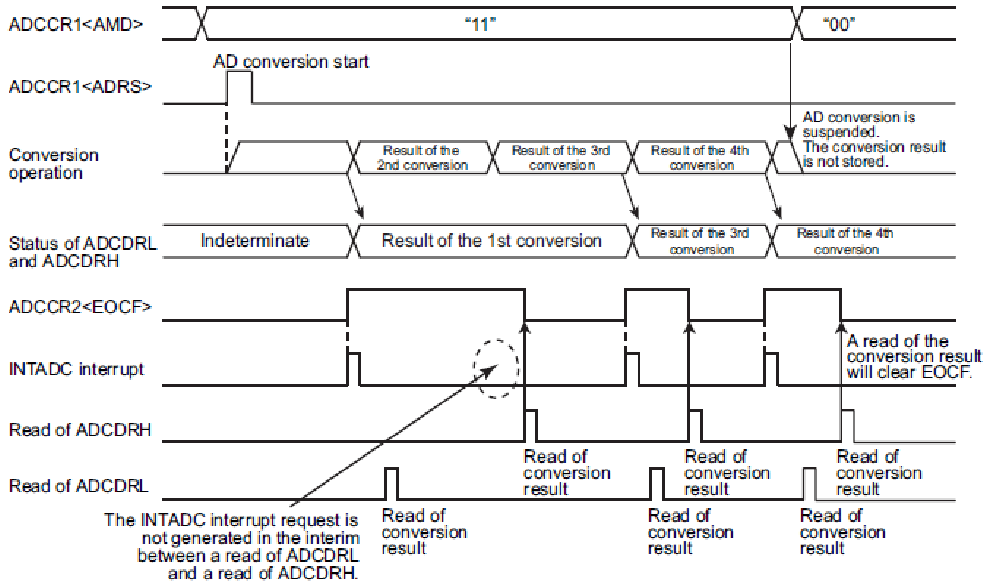

ADCCR1<AMD>
"11"
"00"
AD conversion start
ADCCR1<ADRS>
Conversion operation
Result of the 2nd conversion
Result of the 3rd conversion
Result of the 4th conversion
AD conversion is suspended. The conversion result is not stored.
Status of ADCDRL and ADCDRH
Indeterminate
Result of the 1st conversion
Result of the 3rd conversion
Result of the 4th conversion
ADCCR2<EOCF>
INTADC interrupt
A read of the conversion result will clear EOCF.
Read of ADCDRH
Read of ADCDRL
Read of conversion result
Read of conversion result
Read of conversion result
Read of conversion result
Read of conversion result
Read of conversion result
The INTADC interrupt request is not generated in the interim between a read of ADCDRL and a read of ADCDRH.

Figure 9.3 Repeat Mode

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

9.3.3 AD Operation Disable and Forced Stop of AD Operation

If you want to force the AD converter to stop when AD conversion is ongoing in single mode or if you want to stop the AD converter when AD conversion is ongoing in repeat mode, set ADCCR1 <AMD> to "00".

If ADCCR1 <AMD> is set to "00", registers ADCCR2 <EOCF>, ADCCR2 <ADBF>, ADCDRL, and ADCDRH are initialized to "0".

9.4 Register Setting

1. Set the AD converter control register 1 (ADCCR1) as described below:
   i) From the AD input channel select (SAIN), select the channel in which AD conversion is to be performed.
   ii) Set the analog input control (AINEN) to "Analog input enable".
   iii) At AMD, specify the AD operating mode (single or repeat mode).
2. Set the AD converter control register 2 (ADCCR2) as described below:
   At the AD conversion time (ACK), specify the AD conversion time. For information on how to specify the conversion time, refer to the AD converter control register 2 and Table 9.1.
3. After the above two steps are completed, set "1" on the AD conversion start (ADRS) of the AD converter control register 1 (ADCCR1), and AD conversion starts immediately if single mode is selected.
4. As AD conversion is finished, the AD conversion end flag (EOCF) of the AD converter control register 2 (ADCCR2) is set to "1", the AD conversion result is stored in the AD converted value registers (ADCDRH and ADCDRL), and the INTADC interrupt request is generated.
5. After the conversion result is read from the AD converted value register (ADCDRH), EOCF is cleared to "0". EOCF will also be cleared to "0" if AD conversion is performed once again before reading the AD converted value register (ADCDRH). In this case, the previous conversion result is retained until AD conversion is finished.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Example: After selecting the conversion time 16.0 μs at 8 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, store the conversion result in the HL register. The operation mode is single mode.

```
:(Port setting)               ;Before setting AD converter registers, make an appropriate port
                              ;register setting.(For further details, refer to the section
                              ;that describes I/O ports.)
LD    (ADCCR1), 0y00110011    ;Select AIN3 and operation mode
LD    (ADCCR2), 0y00000010    ;Select conversion time (128/fcgck)
SET   (ADCCR1). 7             ;ADRS = 1 (AD conversion start)
TEST  (ADCCR2). 7             ;EOCF = 1 ?
J     T, SLOOP
LD    HL, (ADCDRL)            ;Read result data
```

9.5 Starting STOP/IDLE0/SLOW Modes

If STOP/IDLE0/SLOW mode is started, registers ADCCR1 <ADRS, AMD, AINEN>, ADCCR2 <EOCF, ADBF>, ADCDRL and ADCDRH are initialized to "0". If any of these modes is started during AD conversion, AD conversion is suspended, and the AD converter stops (registers are likewise initialized). When restored from STOP/IDLE0/SLOW mode, AD conversion is not automatically restarted. Therefore, registers must be reconfigured as necessary.

If STOP/IDLE0/SLOW mode is started during AD conversion, analog reference voltage is automatically disconnected and, therefore, there is no possibility of current flowing into the analog reference voltage.

9.6 Analog Input Voltage and AD Conversion Result

Analog input voltages correspond to AD-converted, 10-bit digital values, as shown in Figure 9.4.

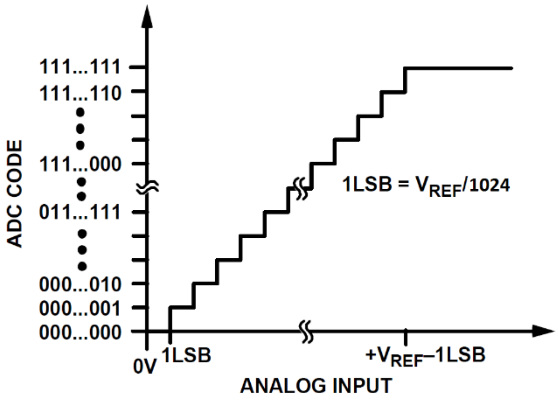


Figure 9.4 Relationships between Analog Input Voltages and AD-converted Values (Typical Values)

9.7 Precautions about the AD Converter

9.7.1 Analog Input Pin Voltage Range

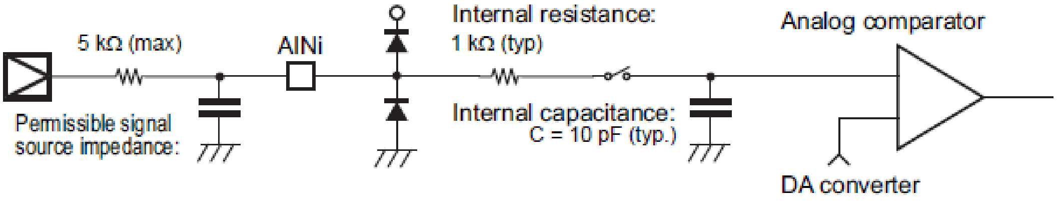
Analog input pins (AIN0 through AIN9) should be used at voltages from VAREF to VSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain, and converted values on other pins will also be affected.

9.7.2 Analog Input Pins Used as Input / Output Ports

Analog input pins (AIN0 through AIN9) are also used as input/output ports. In using one of analog input pins (ports) to execute AD conversion, input/output instructions at all other pins (ports) must not be executed. If they are executed, there is the possibility that the accuracy of AD conversion may deteriorate. This also applies to pins other than analog input pins; if one pin receives inputs or generates outputs, noise may occur and its adjacent pins may be affected by that noise.

9.7.3 Noise Countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 9.5. The higher the output impedance of the analog input source, the more susceptible it becomes to noise. Therefore, make sure the output impedance of the signal source in your design is 5 KΩ or less. It is recommended that a capacitor be attached externally.



Note): I = 8 to 0

Figure 9.5 Analog Input Equivalent Circuit and Example of Input Pin Processing

10. Timer / Counter

10.1 Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to detect rapidly the CPU malfunctions such as endless loops due to spurious noises or the deadlock conditions, and return the CPU to a system recovery routine.

The watchdog timer signals used for detecting malfunctions can be programmed as watchdog interrupt request signals or watchdog timer reset signals.

Note): Care must be taken in system designing since the watchdog timer may not fulfill its functions due to disturbing noise and other effects.

10.1.1 Configuration

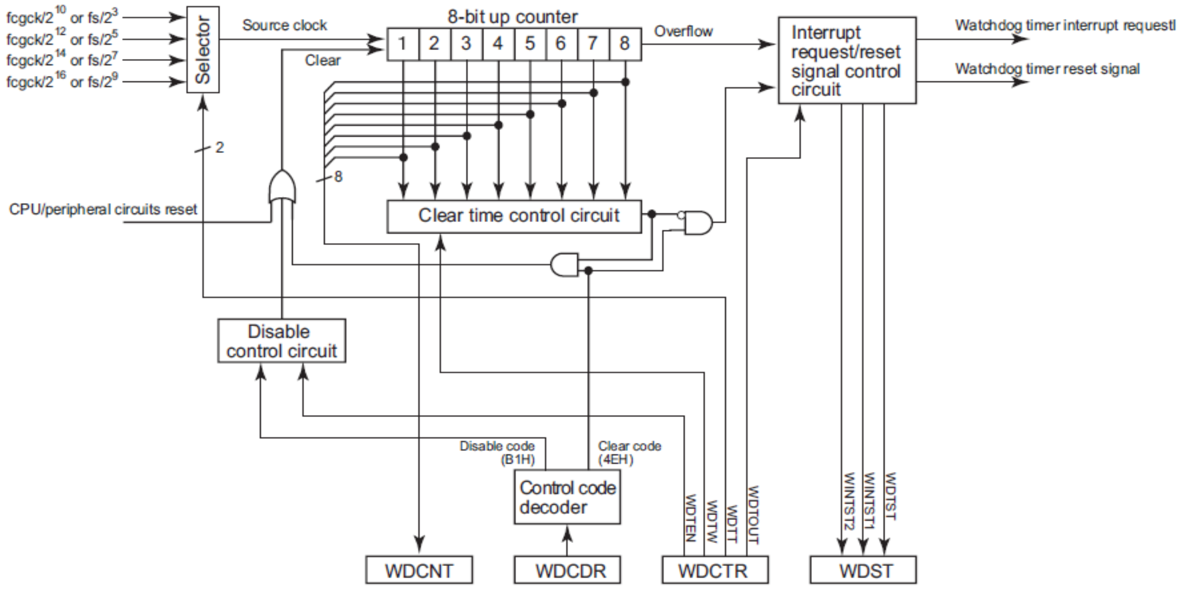


Figure 10.1 Watchdog Timer Configuration

10.1.2 Control

The watchdog timer is controlled by the watchdog timer control register (WDCTR), the watchdog timer control code register (WDCDR), the watchdog timer counter monitor (WDCNT) and the watchdog timer status (WDST).

The watchdog timer is enabled automatically just after the warm-up operation that follows reset is finished.

Watchdog Timer Control Register

WDCTR (0x0FD4)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	WDTEN	WDTW		WDTT		WDTOUT
Read/Write	R	R	R/W	R/W		R/W		R/W
After reset	1	0	1	0	0	1	1	0

<table>
<tr><td>WDTEN</td><td>Enable / disable the watch-dog timer</td><td colspan="4">0: Disable
1: Enable</td></tr>
<tr><td>WDTW</td><td>Set the clear time of the 8-bit up counter.</td><td colspan="4">00: The 8-bit up counter is cleared by writing the clear code at any point within the overflow time of the 8-bit up counter.
01: A watchdog timer interrupt request is generated by writing the clear code at a point within the first quarter of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first quarter of the overflow time has elapsed.
10: A watchdog timer interrupt request is generated by writing the clear code at a point within the first half of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first half of the overflow time has elapsed.
11: A watchdog timer interrupt request is generated by writing the clear code at a point within the first three quarters of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first three quarters of the overflow time have elapsed.</td></tr>
<tr><td rowspan="6">WDTT</td><td rowspan="6">Set the overflow time of the 8-bit up counter.</td><td rowspan="2"></td><td colspan="2">NORMAL mode</td><td rowspan="2">SLOW mode</td></tr>
<tr><td>DV9CK=0</td><td>DV9CK=1</td></tr>
<tr><td>00:</td><td>2^{18}/fcgck</td><td>2^{11}/fs</td><td>2^{11}/fs</td></tr>
<tr><td>01:</td><td>2^{20}/fcgck</td><td>2^{13}/fs</td><td>2^{13}/fs</td></tr>
<tr><td>10:</td><td>2^{22}/fcgck</td><td>2^{15}/fs</td><td>2^{15}/fs</td></tr>
<tr><td>11:</td><td>2^{24}/fcgck</td><td>2^{17}/fs</td><td>2^{17}/fs</td></tr>
<tr><td>WDTOUT</td><td>Select an overflow detection signal of the 8-bit up counter.</td><td colspan="4">0 : Watchdog timer interrupt request signal
1 : Watchdog timer reset request signal</td></tr>
</table>

Note 1): fcgck, Gear clock [Hz]; fs, Low frequency clock [Hz]

Note 2): WDCTR <WDTW>, WDCTR <WDTT> and WDCTR <WDTOUT> cannot be changed when WDCTR <WDTEN> is "1". If WDCTR <WDTEN> is "1", clear WDCTR <WDTEN> to "0" and write the disable code (0xB1) into WDCDR to disable the watchdog timer operation. Note that WDCTR <WDTW>, WDCTR <WDTT> and WDCTR <WDTOUT> can be changed at the same time as setting WDCTR <WDTEN> to "1".

Note 3): Bit 7 and bit 6 of WDCTR are read as "1" and "0" respectively.

Watchdog Timer Control Code Register

WDCDR (0x0FD5)	7	6	5	4	3	2	1	0
Bit Symbol	WDTCR2							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

WDTCR2	Write watchdog timer control codes.	0x4E: Clear the watchdog timer. (clear code) 0xB1: Disable the watchdog timer operation and clear the 8-bit up counter when WDCTR <WDTEN> is "0". (disable code) Others: Invalid

8-bit Up Counter Monitor

WDCNT (0x0FD6)	7	6	5	4	3	2	1	0
Bit Symbol	WDCNT							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0

WDCNT	Monitor the count value of the 8-bit up counter.	The count value of the 8-bit up counter is read.

Watchdog Timer Status

WDST (0x0FD7)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	WINTST2	WINTST1	WDTST
Read/Write	R	R	R	R	R	R	R	R
After reset	0	1	0	1	1	0	0	1

WINTST2	Watchdog timer interrupt request signal factor status 2	0: No watchdog timer interrupt request signal has occurred. 1: A watchdog timer interrupt request signal has occurred due to the overflow of the 8-bit up counter.
WINTST1	Watchdog timer interrupt request signal factor status 1	0: No watchdog timer interrupt request signal has occurred. 1: A watchdog timer interrupt request signal has occurred due to releasing of the 8-bit up counter outside the clear time.
WDTST	Watchdog timer operating state status	0: Operation disabled 1: Operation enabled

Note 1): WDST <WINTST2> and WDST <WINTST1> are cleared to "0" by reading WDST.

Note 2): Values after reset are read from bits 7 to 3 of WDST.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

10.1.3 Function

The watchdog timer can detect the CPU malfunctions and deadlock by detecting the overflow of the 8-bit up counter and detecting releasing of the 8-bit up counter outside the clear time.

The watchdog timer stoppage and other abnormalities can be detected by reading the count value of the 8-bit up counter at random times and comparing the value to the last read value.

10.1.3.1 Setting of Enabling / Disabling the Watchdog Timer Operation

Setting WDCTR <WDTEN> to "1" enables the watchdog timer operation, and the 8-bit up counter starts counting the source clock.

WDCTR <WDTEN> is initialized to "1" after the warm-up operation that follows reset is released. This means that the watchdog timer is enabled.

To disable the watchdog timer operation, clear WDCTR <WDTEN> to "0" and write 0xB1 into WDCDR. Disabling the watchdog timer operation clears the 8-bit up counter to "0".

Note): If the overflow of the 8-bit up counter occurs at the same time as 0xB1 (disable code) is written into WDCDR with WDCTR <WDTEN> set at "1", the watchdog timer operation is disabled preferentially and the overflow detection is not executed.

To re-enable the watchdog timer operation, set WDCTR <WDTEN> to "1". There is no need to write a control code into WDCDR.

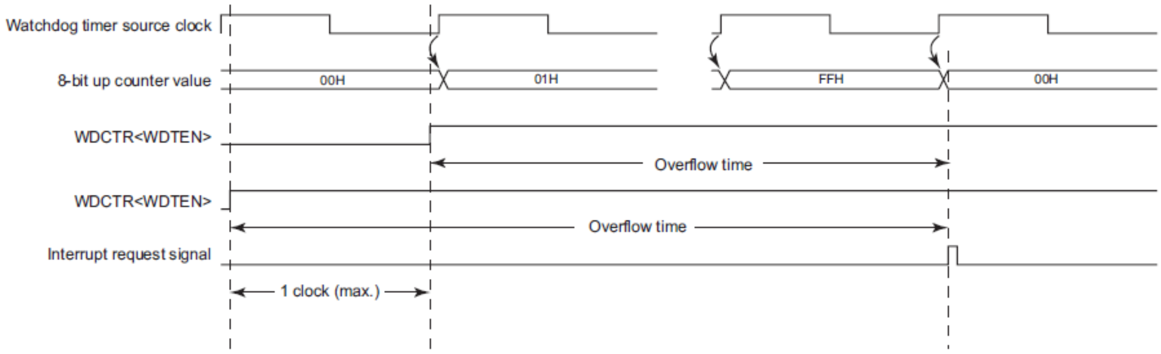


Figure 10.2 WDCTR <WDTEN> Set Timing and Overflow Time

Note): The 8-bit up counter source clock operates out of synchronization with WDCTR <WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR <WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within the period of the overflow time minus 1 source clock cycle.

10.1.3.2 Setting the Clear Time of the 8-bit Up Counter

WDCTR <WDTW> sets the clear time of the 8-bit up counter.

When WDCTR <WDTW> is "00", the clear time is equal to the overflow time of the 8-bit up counter, and the 8-bit up counter can be cleared at any time.

When WDCTR <WDTW> is not "00", the clear time is fixed to only a certain period within the overflow time of the 8-bit up counter. If the operation for releasing the 8-bit up counter is attempted outside the clear time, a watchdog timer interrupt request signal occurs.

At this time, the watchdog timer is not cleared but continues counting. If the 8-bit up counter is not cleared within the clear time, a watchdog timer reset request signal or a watchdog timer interrupt request signal occurs due to the overflow, depending on the WDCTR <WDTOUT> setting.

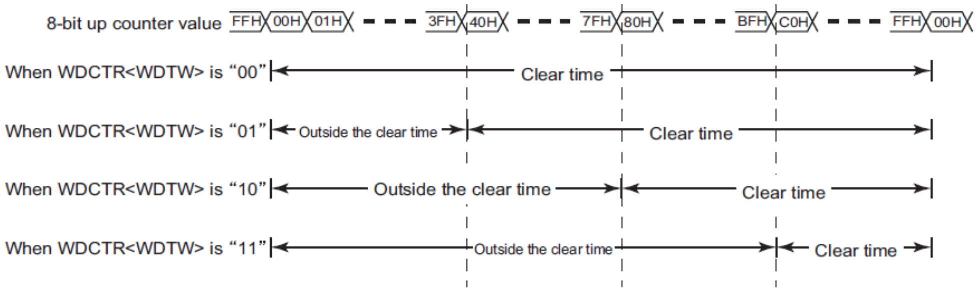


Figure 10.3 WDCTR <WDTW> and the 8-bit Up Counter Clear Time

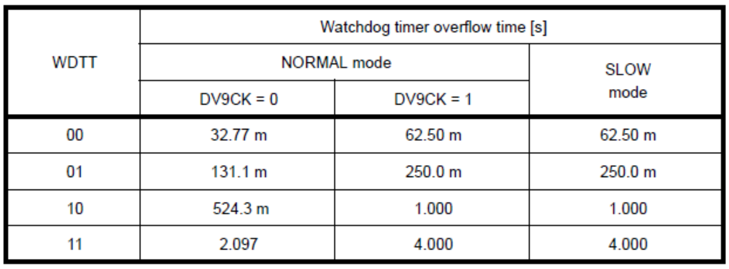
10.1.3.3 Setting the Overflow Time of the 8-bit Up Counter

WDCTR <WDTT> sets the overflow time of the 8-bit up counter.

When the 8-bit up counter overflows, a watchdog timer reset request signal or a watchdog timer interrupt request signal occurs, depending on the WDCTR <WDTOUT> setting.

If the watchdog timer interrupt request signal is selected as the malfunction detection signal, the watchdog counter continues counting, even after the overflow has occurred.

The watchdog timer temporarily stops counting up in the STOP mode (including warm-up) or in the IDLE / SLEEP mode, and restarts counting up after the STOP / IDLE / SLEEP mode is released. To prevent the 8-bit up counter from overflowing immediately after the STOP / IDLE / SLEEP mode is released, it is recommended to clear the 8-bit up counter before the operation mode is changed.



WDTT	Watchdog timer overflow time [s]		
	NORMAL mode		SLOW mode
	DV9CK = 0	DV9CK = 1	
00	32.77 m	62.50 m	62.50 m
01	131.1 m	250.0 m	250.0 m
10	524.3 m	1.000	1.000
11	2.097	4.000	4.000

Table 10.1 Watchdog Timer Overflow Time (fcgck=8.0 MHz; fs=32.768 KHz)

Note): The 8-bit up counter source clock operates out of synchronization with WDCTR <WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR <WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within a period of the overflow time minus 1 source clock cycle.

10.1.3.4 Setting an Overflow Detection Signal of the 8-bit Up Counter

WDCTR <WDTOUT> selects a signal to be generated when the overflow of the 8-bit up counter is detected.

(a) When Watchdog Timer Interrupt Request Signal is Selected (as WDCTR <WDTOUT> is "0")

Releasing WDCTR <WDTOUT> to "0" causes a watchdog timer interrupt request signal to occur when the 8-bit up counter overflows.

A watchdog timer interrupt is a non-maskable interrupt, and its request is always accepted, regardless of the interrupt master enable flag (IMF) setting.

Note): When a watchdog timer interrupt is generated while another interrupt, including a watchdog timer interrupt, is already accepted, the new watchdog timer interrupt is processed immediately and the preceding interrupt is put on hold. Therefore, if watchdog timer interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

(b) When Watchdog Timer Reset Request Signal is Selected (as WDCTR <WDTOUT> is "1")

Setting WDCTR <WDTOUT> to "1" causes a watchdog timer reset request signal to occur when the 8-bit up counter overflows.

This watchdog timer reset request signal resets the MQ8S MCU series IC, and starts the warm-up operation.

10.1.3.5 Writing the Watchdog Timer Control Codes

The watchdog timer control codes are written into WDCDR.

By writing 0x4E (clear code) into WDCDR, the 8-bit up counter is cleared to "0" and continues counting the source clock.

When WDCTR <WDTEN> is "0", writing 0xB1 (disable code) into WDCDR disables the watchdog timer operation.

To prevent the 8-bit up counter from overflowing, clear the 8-bit up counter in a period shorter than the overflow time of the 8-bit up counter and within the clear time.

By designing the program so that no overflow will occur, the program malfunctions and deadlock can be detected through interrupts generated by watchdog timer interrupt request signals.

By applying a reset to the microcomputer using watchdog timer reset request signals, the CPU can be restored from malfunctions and deadlock.

Example: When WDCTR<WDTEN> is "0", set the watchdog timer detection time to 220/fcgck [s], set the counter clear time to half of the overflow time, and allow a watchdog timer reset request signal to occur if a malfunction is detected.

```
LD      (WDCTR), 0y00110011      ;WDTW←10,    WDTT←01,
                                 WDTOUT←1
```

Clear the 8-bit up counter at a point after half of its overflow time and within a period of the overflow time minus 1 source clock cycle.

```
LD     (WDCDR), 0x4E             ;Clear  the  8-bit  up
                                 counter
```

Clear the 8-bit up counter at a point after half of its overflow time and within a period of the overflow time minus 1 source clock cycle.

```
LD     (WDCDR), 0x4E             ;Clear  the  8-bit  up
                                 counter
```

Note: If the overflow of the 8-bit up counter and writing of 0x4E (clear code) into WDCDR occur simultaneously, the 8-bit up counter is cleared preferentially and the overflow detection is not executed.

10.1.3.6 Reading the 8-bit Up Counter

The counter value of the 8-bit up counter can be read by reading WDCNT.

The stoppage of the 8-bit up counter can be detected by reading WDCNT at random times and comparing the value to the last read value.

10.1.3.7 Reading the Watchdog Timer Status

The watchdog timer status can be read at WDST.

WDST <WDTST> is set to "1" when the watchdog timer operation is enabled, and it is cleared to "0" when the watchdog timer operation is disabled.

WDST <WINTST2> is set to "1" when a watchdog timer interrupt request signal occurs due to the overflow of the 8-bit up counter.

WDST <WINTST1> is set to "1" when a watchdog timer interrupt request signal occurs due to the operation for releasing the 8-bit up counter outside the clear time.

You can know which factor has caused a watchdog timer interrupt request signal by reading WDST <WINTST2> and WDST <WINTST1> in the watchdog timer interrupt service routine.

WDST <WINTST2> and WDST <WINTST1> are cleared to "0" when WDST is read. If WDST is read at the same time as the condition for turning WDST <WINTST2> or WDST <WINTST1> to "1" is satisfied, WDST <WINTST2> or WDST <WINTST1> is set to "1", rather than being cleared.

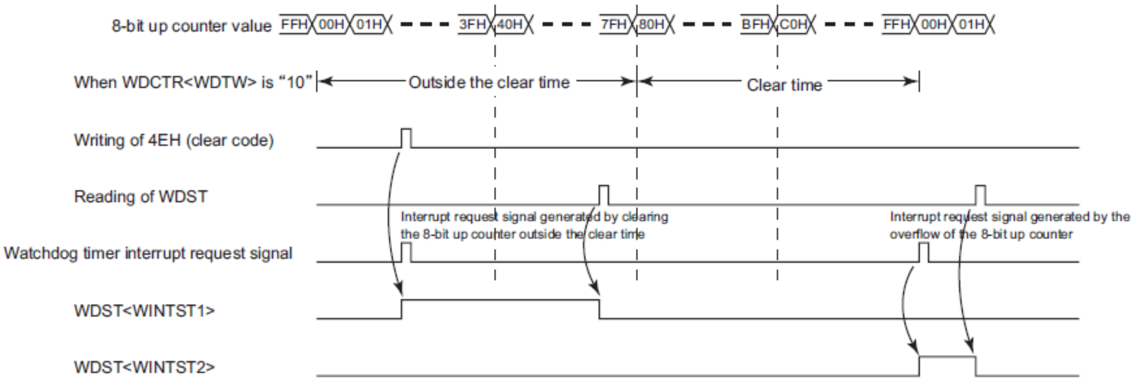

8-bit up counter value
FFH
00H
01H
3FH
40H
7FH
80H
BFH
C0H
FFH
00H
01H
When WDCTR<WDTW> is "10"
Outside the clear time
Clear time
Writing of 4EH (clear code)
Reading of WDST
Interrupt request signal generated by clearing the 8-bit up counter outside the clear time
Interrupt request signal generated by the overflow of the 8-bit up counter
Watchdog timer interrupt request signal
WDST<WINTST1>
WDST<WINTST2>

Figure 10.4 Changes in the Watchdog Timer Status

10.2 Divider Output (DVOB)

This function outputs approximately 50% duty pulses that can be used to drive the piezoelectric buzzer or other device.

10.2.1 Configuration

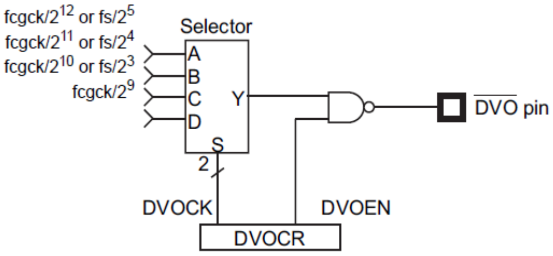

fcgck/2^{12} or fs/2^{5}
fcgck/2^{11} or fs/2^{4}
fcgck/2^{10} or fs/2^{3}
fcgck/2^{9}
Selector
A
B
C
D
Y
S
2
$\overline{\text{DVO}}$ pin
DVOCK
DVOEN
DVOCR

Figure 10.5 Divider Output

10.2.2 Control

The divider output is controlled by the divider output control register (DVOCR).

Divider Output Control Register

DVOCR (0x0038)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	DVOEN	DVOCK	
Read/Write	R	R	R	R	R	R/W	R/W	
After reset	0	0	0	0	0	0	0	0

<table>
<tr><td>DVOEN</td><td>Enable / disable the divider output</td><td colspan="4">0: Disable
1: Enable</td></tr>
<tr><td rowspan="6">DVOCK</td><td rowspan="6">Select the divider output frequency
Unit: [Hz]</td><td rowspan="2"></td><td colspan="2">Normal 1/2, IDLE 1/2 mode</td><td rowspan="2">SLOW 1/2 mode
SLEEP 1/2 mode</td></tr>
<tr><td>DV9CK=0</td><td>DV9CK=1</td></tr>
<tr><td>00:</td><td>fcgck/2^{12}</td><td>fs/2^{5}</td><td>fs/2^{5}</td></tr>
<tr><td>01:</td><td>fcgck/2^{11}</td><td>fs/2^{4}</td><td>fs/2^{4}</td></tr>
<tr><td>10:</td><td>fcgck/2^{10}</td><td>fs/2^{3}</td><td>fs/2^{3}</td></tr>
<tr><td>11:</td><td>fcgck/2^{9}</td><td>Reserved</td><td>Reserved</td></tr>
</table>

Note 1): fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2): DVOCR <DVOEN> is cleared to "0" when the operation is switched to STOP or IDLE0/SLEEP0 mode. DVOCR <DVOCK> holds the value.

Note 3): When SYSCR1 <DV9CK> is "1" in the NORMAL 1/2 or IDLE 1/2 mode, the DVO frequency is subject to some fluctuations to synchronize fs and fcgck.

Note 4): Bits 7 to 3 of DVOCR are read as "0".

10.2.3 Function

Select the divider output frequency at DVOCR <DVOCK>.

The divider output is enabled by setting DVOCR <DVOEN> to "1". Then, the rectangular waves selected by DVOCR <DVOCK> are output from DVOB pin.

It is disabled by clearing DVOCR <DVOEN> to "0". And DVOB pin keeps "H" level.

When the operation is changed to STOP or IDLE0 / SLEEP0 mode, DVOCR <DVOEN> is cleared to "0" and the DVOB pin outputs the "H" level.

The divider output source clock operates, regardless of the value of DVOCR <DVOEN>.

Therefore, the frequency of the first divider output after DVOCR <DVOEN> is set to "1" is not the frequency set at DVOCR <DVOCK>.

When the operation is changed to the software, STOP or IDLE0/SLEEP0 mode is activated and DVOCR <DVOEN> is cleared to "0", the frequency of the divider output is not the frequency set at DVOCR <DVOCK>.

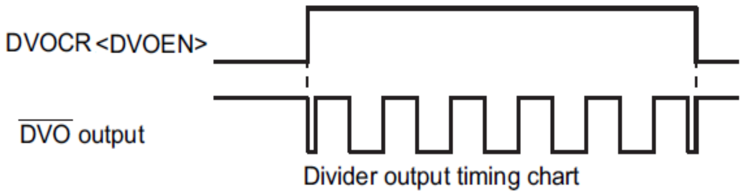


Figure 10.6 Divider Output Timing

When the operation is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, the divider output frequency does not reach the expected value due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs).

DVOCK	Divider output frequency [Hz]		
	NORMAL 1/2, IDLE 1/2 mode		SLOW1/2, SLEEP1/2 mode
	DV9CK = 0	DV9CK = 1	
00	1.953 k	1.024 k	1.024 k
01	3.906 k	2.048 k	2.048 k
10	7.813 k	4.096 k	4.096 k
11	15.625 k	Reserved	Reserved

Table 10.2 Divider Output Frequency
(Example: fcgck = 8.0 MHz, fs = 32.768 kHz)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

10.3 Time Base Timer (TBT)

The time base timer generates the time base for key scanning, dynamic display and other processes. It also provides a time base timer interrupt (INTTBT) in a certain cycle.

10.3.1 Configuration

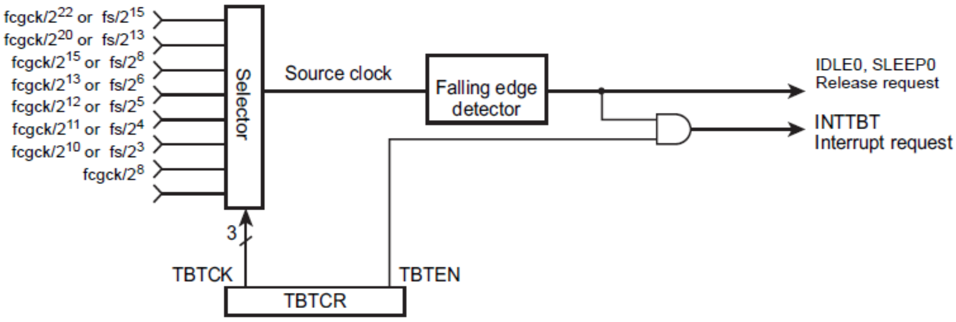

$fcgck/2^{22}$ or $fs/2^{15}$
$fcgck/2^{20}$ or $fs/2^{13}$
$fcgck/2^{15}$ or $fs/2^{8}$
$fcgck/2^{13}$ or $fs/2^{6}$
$fcgck/2^{12}$ or $fs/2^{5}$
$fcgck/2^{11}$ or $fs/2^{4}$
$fcgck/2^{10}$ or $fs/2^{3}$
$fcgck/2^{8}$
Selector
Source clock
Falling edge detector
IDLE0, SLEEP0
Release request
INTTBT
Interrupt request
3
TBTCK
TBTEN
TBTCR

Figure 10.7 Time Base Timer Configuration

10.3.2 Control

The time base timer is controlled by the time base timer control register (TBTCR).

Time Base Timer Control Register

TBTCR (0x0039)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	TBTEN	TBTCK		
Read/Write	R	R	R	R	R/W	R/W		
After reset	0	0	0	0	0	0	0	0

TBTEN	Enable / disable the time base timer interrupt requests.	0: Disable 1: Enable			
TBTCK	Select the time base timer interrupt frequency Unit: [Hz]		Normal 1/2, IDLE 1/2 mode		SLOW 1/2 mode SLEEP 1/2 mode
			DV9CK=0	DV9CK=1	
		000:	$fcgck/2^{22}$	$fs/2^{15}$	$fs/2^{15}$
		001:	$fcgck/2^{20}$	$fs/2^{13}$	$fs/2^{13}$
		010:	$fcgck/2^{15}$	$fs/2^{8}$	Reserved
		011:	$fcgck/2^{13}$	$fs/2^{6}$	Reserved
		100:	$fcgck/2^{12}$	$fs/2^{5}$	Reserved
		101:	$fcgck/2^{11}$	$fs/2^{4}$	Reserved
		110:	$fcgck/2^{10}$	$fs/2^{3}$	Reserved
		111:	$fcgck/2^{8}$	Reserved	Reserved

Note 1): fcgck : Gear clock [Hz], fs : Low-frequency clock [Hz]

Note 2): When the operation is changed to the STOP mode, TBTCR <TBTEN> is cleared to "0" and TBTCR <TBTCK> maintains the value.

Note 3): TBTCR <TBTCK> should be set when TBTCR <TBTEN> is "0".

Note 4): When SYSCR1 <DV9CK> is "1" in the NORMAL 1/2 or IDLE1/2 mode, the interrupt request is subject to some fluctuations to synchronize fs and fcgck.

Note 5): Bits 7 to 4 of TBTCR are read as "0".

10.3.3 Function

Select the source clock frequency for the time base timer by TBTCR <TBTCK>. TBTCR <TBTCK> should be changed when TBTCR <TBTEN> is "0". Otherwise, the INTTBT interrupt request is generated at unexpected timing.

Setting TBTCR <TBTEN> to "1" causes interrupt request signals to occur at the falling edge of the source clock. When TBTCR <TBTEN> is cleared to "0", no interrupt request signal will occur.

When the operation is changed to the STOP mode, TBTCR < TBTEN> is cleared to "0". The source clock of the time base timer operates regardless of the TBTCR <TBTEN> value.

A time base timer interrupt is generated at the first falling edge of the source clock after a time base timer interrupt request is enabled. Therefore, the period from the time TBTCR <TBTEN> is set to "1" to the time the first interrupt request occurs is shorter than the frequency period set at TBTCR <TBTCK>.

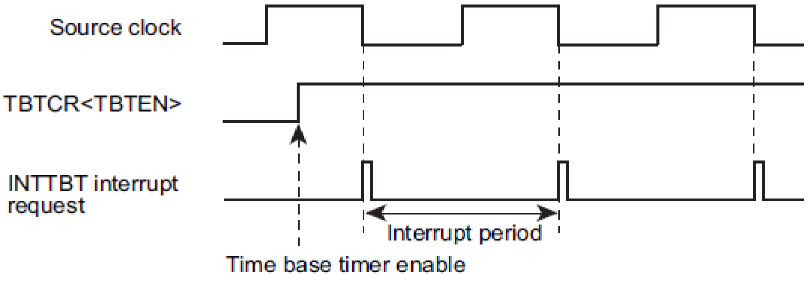
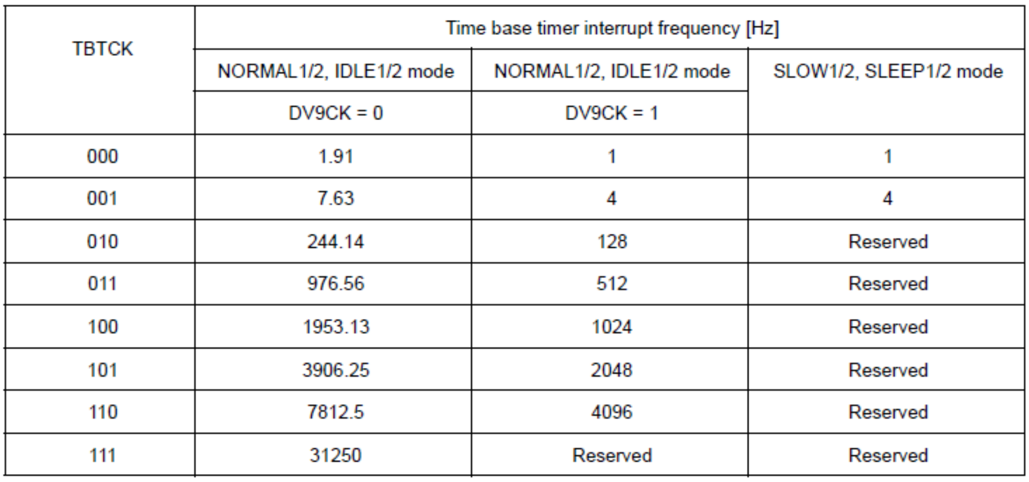


Figure 10.8 Time Base Timer Interrupt

When the operation is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, the interrupt request will not occur at the expected timing due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs). It is recommended that the operation mode is changed when TBTCR <TBTEN> is "0".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.



TBTCK	Time base timer interrupt frequency [Hz]		
	NORMAL1/2, IDLE1/2 mode	NORMAL1/2, IDLE1/2 mode	SLOW1/2, SLEEP1/2 mode
	DV9CK = 0	DV9CK = 1	
000	1.91	1	1
001	7.63	4	4
010	244.14	128	Reserved
011	976.56	512	Reserved
100	1953.13	1024	Reserved
101	3906.25	2048	Reserved
110	7812.5	4096	Reserved
111	31250	Reserved	Reserved

Table 10.3 Time Base Timer Interrupt Frequency
(Example: fcgck = 8.0 MHz, fs = 32.768 kHz)

Example: Set the time base timer interrupt frequency to fcgck/215 [Hz] and enable interrupts.

```
 DI                              ;IMF ← 0
SET   (EIRL). 5                  ;Set the interrupt enable register
 EI                              ;IMF ← 1
 LD   (TBTCR), 0y00000010        ;Set the interrupt frequency
 LD   (TBTCR), 0y00001010        ;Enable generation of interrupt request signals
```

10.4 Real Time Clock (RTC)

The real time clock is a function that generates interrupt requests at certain intervals using the low-frequency clock.

The number of interrupts is counted by the software to realize the clock function. The real time clock can be used only in the operation modes where the low-frequency clock oscillates, except for SLEEP0.

10.4.1 Configuration

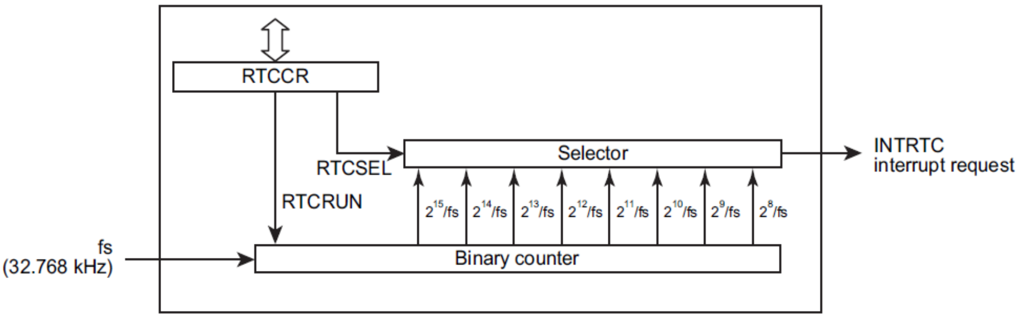


Figure 10.9 Real Time Clock

10.4.2 Control

The real time clock is controlled by following registers.

Low Power Consumption Register 2

POFFCR2 (0x0F76)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	RTCEN	-	-	-	-	SIO0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0: Disable 1: Enable
SIO0EN	SIO0 control	0: Disable 1: Enable

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Real Time Clock Control Register

RTCCR (0x0FC8)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	RTCSEL			RTCRUN
Read/Write	R	R	R	R	R/W			R/W
After reset	0	0	0	0	0	0	0	0

RTCSEL	Selects the interrupt generation interval	000:2^{15}/fs(1.000[s]@fs=32.768kHz) 001:2^{14}/fs(0.500[s]@fs=32.768kHz) 010:2^{13}/fs(0.250[s]@fs=32.768kHz) 011: 2^{12}/fs (125.0[ms]@fs=32.768kHz) 100:2^{11}/fs(62.50[ms]@fs=32.768kHz) 101:2^{10}/fs(31.25[ms]@fs=32.768kHz) 110: 2^{9}/fs(15.62[ms]@fs=32.768kHz) 111:2^{8}/fs(7.81[ms]@fs=32.768kHz)
RTCRUN	Enables/disables the realtime clock operation	0: Disable 1: Enable

Note 1): fs: Low-frequency clock [Hz]

Note 2): RTCCR <RTCSEL> can be rewritten only when RTCCR <RTCRUN> is "0". If data is written into RTCCR <RTCSEL> when RTCCR <RTCRUN> is "1", the existing data remains effective. RTCCR <RTCSEL> can be rewritten at the same time as enabling the real time clock, but it cannot be rewritten at the same time as disabling the real time clock.

Note 3): If the real time clock is enabled and when 1) SYSCR2 <XTEN> is cleared to "0" to stop the low-frequency clock oscillation circuit or 2) the operation is changed to the STOP mode or the SLEEP0 mode, the data in RTCCR <RTCSEL> is maintained and RTCCR <RTCRUN> is cleared to "0".

10.4.3 Function

10.4.3.1 Low Power Consumption Function

Real time clock has the low power consumption registers (POFFCR2) that save power when the real time clock is not being used. Setting POFFCR2 <RTCEN> to "0" disables the basic clock supply to real time clock to save power. Note that this renders the real time clock unusable. Setting POFFCR2 <RTCEN> to "1" enables the basic clock supply to real time clock and allows the real time clock to operate.

After reset, POFFCR2 <RTCEN> are initialized to "0", and this renders the real time clock unusable. When using the real time clock for the first time, be sure to set POFFCR2 <RTCEN> to "1" in the initial setting of the program (before the real time clock control registers are operated).

Do not change POFFCR2 <RTCEN> to "0" during the real time clock operation. Otherwise real time clock may operate unexpectedly.

10.4.3.2 Enabling / Disabling the Real Time Clock Operation

Setting RTCCR <RTCRUN> to "1" enables the real time clock operation. Setting RTCCR <RTCRUN> to "0" disables the real time clock operation. RTCCR <RTCRUN> is cleared to "0" just after reset release.

10.4.3.3 Selecting the Interrupt Generation Interval

The interrupt generation interval can be selected at RTCCR <RTCSEL>. RTCCR <RTCSEL> can be rewritten only when RTCCR <RTCRUN> is "0". If data is written into RTCCR <RTCSEL> when RTCCR <RTCRUN> is "1", the existing data remains effective.

RTCCR <RTCSEL> can be rewritten at the same time as enabling the real time clock operation, but it cannot be re-written at the same time as disabling the real time clock operation.

10.4.4 Real Time Clock Operation

10.4.4.1 Enabling the Real Time Clock Operation

Set the interrupt generation interval to RTCCR <RTCSEL>, and at the same time, set RTCCR <RTCRUN> to "1".When RTCCR <RTCRUN> is set to "1", the binary counter for the real time clock starts counting of the low-frequency clock. When the interrupt generation interval selected at RTCCR <RTCSEL> is reached, a real time clock interrupt request (INTRTC) is generated and the counter continues counting.

10.4.4.2. Disabling the Real Time Clock Operation

Clear RTCCR <RTCRUN> to "0".When RTCCR <RTCRUN> is cleared to "0", the binary counter for the real time clock is cleared to "0" and stops counting of the low-frequency clock.

10.5 8-bit Timer Counters (TC0)

MQ6935 contains 4 channels of high-performance 8-bit timer counters 00, 01, 02 and 03 (TC0). Each timer can be used for time measurement and pulse output with a prescribed width. Two 8-bit timer counters are cascadable to form a 16-bit timer.

This chapter describes 2 channels of 8-bit timer counters 00 and 01. For 8-bit timer counters 02 and 03, replace the SFR addresses and pin names as shown in Table 10.4 and Table 10.5.

	16-bit mode	T0xREG (Address)	T0xPWM (Address)	T0xMOD (Address)	T0xxCR (Address)	Low power consumption register
Timer counter 00	Lower	T00REG (0x0026)	T00PWM (0x0028)	T00MOD (0x002A)	T001CR (0x002C)	POFFCR0 <TC001EN>
Timer counter 01	Higher	T01REG (0x0027)	T01PWM (0x0029)	T01MOD (0x002B)		
Timer counter 02	Lower	T02REG (0x0F88)	T02PWM (0x0F8A)	T02MOD (0x0F8C)	T023CR (0x0F8E)	POFFCR0 <TC023EN>
Timer counter 03	Higher	T03REG (0x0F89)	T03PWM (0x0F8B)	T03MOD (0x0F8D)		

Table 10.4 SFR Address Assignment

	Timer Input Pin	PWM Output Pin	PPG Output Pin
Timer counter 00	TC00 pin	PWM00B pin	PPG00B pin
Timer counter 01	TC01 pin	PWM01B pin	PPG01B pin
Timer counter 02	TC02 pin	PWM02B pin	PPG02B pin
Timer counter 03	TC03 pin	PWM03B pin	PPG03B pin

Table 10.5 Pin Names

10.5.1 Configuration

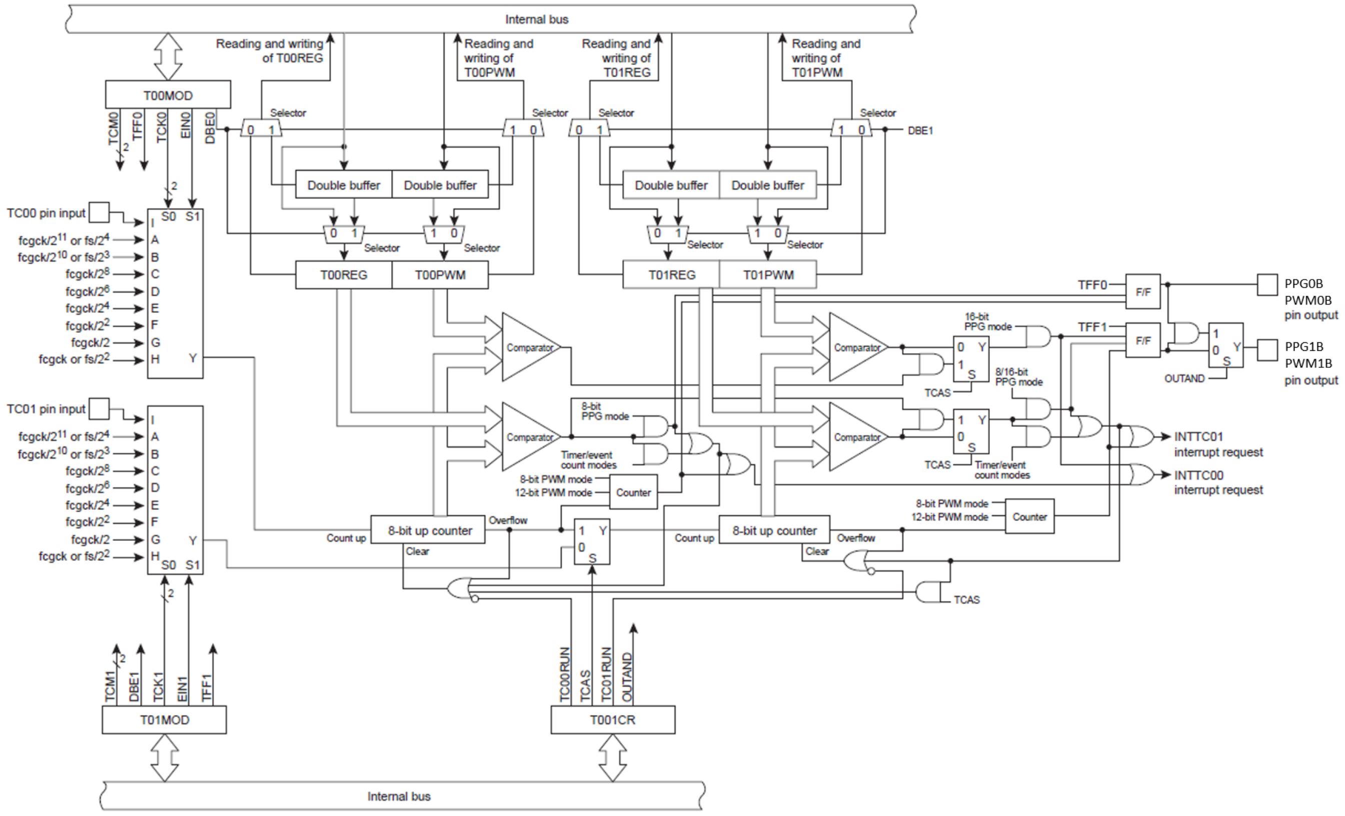


Figure 10.10 8-bit Timer Counter 00 and 01

10.5.2 Control

10.5.2.1 Timer Counter 00

The timer counter 00 is controlled by the timer counter 00 mode register (T00MOD) and two 8-bit timer registers (T00REG and T00PWM).

Timer Register 00

T00REG (0x0026)	15	14	13	12	11	10	9	8
Bit Symbol	T00REG							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Timer Register 00

T00PWM (0x0028)	7	6	5	4	3	2	1	0
Bit Symbol	T00PWM							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Note): For the configuration of T00PWM in the 8-bit and 12-bit PWM modes, refer to "10.5.3.3 8-bit pulse width modulation (PWM) output mode" and "10.5.3.7 12-bit pulse width modulation (PWM) output mode".

Timer Counter 00 Mode Register

T00MOD (0x002A)	7	6	5	4	3	2	1	0
Bit Symbol	TFF0	DBE0	TCK0			EIN0	TCM0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	0	0	0	0	0	0

<table>
<tr><td>TFF0</td><td>Timer F/F0 control</td><td colspan="4">0: Clear
1: Set</td></tr>
<tr><td>DBE0</td><td>Double buffer control</td><td colspan="4">0: Disable the double buffer
1: Enable the double buffer</td></tr>
<tr><td rowspan="10">TCK0</td><td rowspan="10">Operation clock selection</td><td rowspan="2"></td><td colspan="2">Normal 1/2, IDLE 1/2 mode</td><td rowspan="2">SLOW 1/2 mode
SLEEP 1 mode</td></tr>
<tr><td>SYSCR1 <DV9CK&gt;=0</td><td>SYSCR1 <DV9CK&gt;=1</td></tr>
<tr><td>000:</td><td>fcgck/2^{11}</td><td>fs/2^4</td><td>fs/2^4</td></tr>
<tr><td>001:</td><td>fcgck/2^{10}</td><td>fs/2^3</td><td>fs/2^3</td></tr>
<tr><td>010:</td><td>fcgck/2^8</td><td>fcgck/2^8</td><td>-</td></tr>
<tr><td>011:</td><td>fcgck/2^6</td><td>fcgck/2^6</td><td>-</td></tr>
<tr><td>100:</td><td>fcgck/2^4</td><td>fcgck/2^4</td><td>-</td></tr>
<tr><td>101:</td><td>fcgck/2^2</td><td>fcgck/2^2</td><td>-</td></tr>
<tr><td>110:</td><td>fcgck/2</td><td>fcgck/2</td><td>-</td></tr>
<tr><td>111:</td><td>fcgck</td><td>fcgck</td><td>fs/2^2</td></tr>
<tr><td>EIN0</td><td>Selection for using external source clock</td><td colspan="4">0: Select the internal clock as the source clock.
1: Select an external clock a s the source clock.
(the falling edge of the TC00 pin)</td></tr>
<tr><td rowspan="4">TCM0</td><td rowspan="4">Operation mode selection</td><td>00:</td><td colspan="3">8-bit timer / event counter modes</td></tr>
<tr><td>01:</td><td colspan="3">8-bit timer / event counter modes</td></tr>
<tr><td>10:</td><td colspan="3">8-bit pulse width modulation output (PWM) mode</td></tr>
<tr><td>11:</td><td colspan="3">8-bit programmable pulse generate (PPG) mode</td></tr>
</table>

Note 1): fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2): Set T00MOD while the timer is stopped. Writing data into T00MOD is invalid during the timer operation.

Note 3): In the 8-bit timer/event modes, the TFF0 setting is invalid. In this mode, when the PWM00B and PPG0B pins are set as the function output pins in the port setting, the pins always output the "H" level.

Note 4): When EIN0 is set to "1" and the external clock input is selected as the source clock, the TCK0 setting is ignored.

Note 5): When the T001CR <TCAS> bit is "1", timer 00 operates in the 16-bit mode. The T00MOD setting is invalid and timer 00 cannot be used independently in this mode. When the PWM00B and PPG0B pins are set to the function output pins in the port setting, the pins always output the "H" level.

Note 6): When the 16-bit mode is selected at T001CR <TCAS>, the timer start is controlled at T001CR <T01RUN>. Timer 00 is not started by writing data into T001CR <T00RUN>.

10.5.2.2 Timer Counter 01

Timer counter 01 is controlled by timer counter 01 mode register (T01MOD) and two 8-bit timer registers (T01REG and T01PWM).

Timer Register 01

T01REG (0x0027)	15	14	13	12	11	10	9	8
Bit Symbol	T01REG							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Timer Register 01

T01PWM (0x0029)	7	6	5	4	3	2	1	0
Bit Symbol	T01PWM							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Timer Counter 01 Mode Register

T01MOD (0x002B)	7	6	5	4	3	2	1	0
Bit Symbol	TFF1	DBE1	TCK1			EIN1	TCM1	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	0	0	0	0	0	0

TFF1	Timer F/F1 control	0: Clear 1: Set			
DBE1	Double buffer control	0: Disable the double buffer 1: Enable the double buffer			
TCK1	Operation clock selection		Normal 1/2, IDLE 1/2 mode		SLOW 1/2 mode SLEEP 1 mode
			SYSCR1 <DV9CK>=0	SYSCR1 <DV9CK>=1	
		000:	fcgck/2^{11}	fs/2^4	fs/2^4
		001:	fcgck/2^{10}	fs/2^3	fs/2^3
		010:	fcgck/2^8	fcgck/2^8	-
		011:	fcgck/2^6	fcgck/2^6	-
		100:	fcgck/2^4	fcgck/2^4	-
		101:	fcgck/2^2	fcgck/2^2	-
		110:	fcgck/2	fcgck/2	-
		111:	fcgck	fcgck	fs/2^2
EIN1	Selection for using external source clock	0: Select the internal clock as the source clock. 1: Select an external clock as the source clock. (the falling edge of the TC01 pin)			
TCM1	Operation mode selection		T001CR <TCAS>="0" (8-bitmode)	T001CR <TCAS>="1" (16-bitmode)	
		00:	8-bittimer/event counter modes	16-bittimer/event counter modes	
		01:	8-bittimer/event counter modes	16-bittimer/event counter modes	
		10:	8-bit pulse width modulation output (PWM) mode	12-bit pulse width modulation output (PWM) mode	
		11:	8-bit programmable pulse generate (PPG) mode	16-bitprogrammablepulsegenerate(PPG)mode	

Note 1): fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2): Set T01MOD while the timer is stopped. Writing data into T01MOD is invalid during the timer operation.

Note 3): In the 8-bit timer/event modes, the TFF1 setting is invalid. In this mode, when the PWM1B and PPG1B pins are set as the function output pins in the port setting, the pins always output the "H" level.

Note 4): When EIN1 is set to "1" and the external clock input is selected as the source clock, the TCK1 setting is ignored.

10.5.2.3 Common to Timer Counters 00 and 01

Timer counters 00 and 01 have the low power consumption register (POFFCR0) and timer 00 and 01 control registers in common.

Low Power Consumption Register 0

POFFCR0 (0x0F74)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCA1EN	TCA0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,TC 03 control	0: Disable 1: Enable
TC001EN	TC00, TC01 control	0: Disable 1: Enable
TCC0EN	TCC0 control	0: Disable 1: Enable
TCA1EN	TCA1 control	0: Disable 1: Enable
TCA0EN	TCA0 control	0: Disable 1: Enable

Timer 00 and 01 Control Register

T001CR (0x002C)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	OUTAND	TCAS	T01RUN	T00RUN
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

OUTAND	Timer 00 and 01 output control	0: Output the timer 00 output from the PWM00B and PPG0B pins and the timer 01 output from the PWM1B and PPG1B pins. 1: Output a pulse that is a logical ANDed product of the outputs of timer 00 and 01 from the PWM1B and PPG1B pins.
TCA0EN	Timer 00 and 01 cascade control	0: Use timer 00 and 01 independently (8-bit mode) 1: Cascade timer 00 and 01 (16-bit mode)
T01RUN	Timer 01 control Timer 00/01 control (16-bit mode)	0: Stop and clear the timer 1: Start
T00RUN	Timer 00 control	0: Stop and clear the timer 1: Start

Note 1): When STOP mode is started, T00RUN and T01RUN are cleared to "0" and the timers stop. Set T001CR again to use

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

timers 00 and 01 after STOP mode is released.

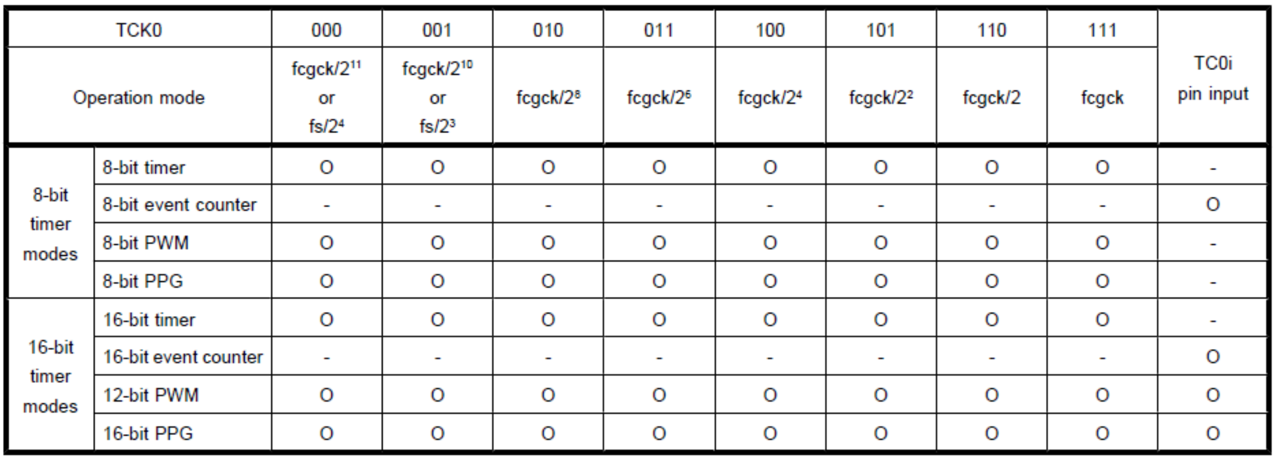
Note 2): When a read instruction is executed on T001CR, bits 7 to 4 are read as "0".

Note 3): When OUTAND is "1", output is obtained from the PWM1B and PPG1B pins only. There is no timer output to the PWM00B and PPG0B pins. If the PWM00B and PPG0B pins are set as the function output pins in the port setting, the pins always output "H".

Note 4): OUTAND and TCAS can be changed only when both TC01RUN and TC00RUN are "0". When either TC01RUN or TC00RUN is "1" or both are "1", the register values remain unchanged by executing write instructions on OUTAND and TCAS. OUTAND and TCAS can be changed at the same time as TC01RUN and TC00RUN are changed from "0" to "1".

10.5.2.4 Operation Modes and Usable Source Clocks

The operation modes of the 8-bit timers and the usable source clocks are listed below.



TCK0		000	001	010	011	100	101	110	111	TC0i pin input
Operation mode		fcgck/2^{11} or fs/2^4	fcgck/2^{10} or fs/2^3	fcgck/2^8	fcgck/2^6	fcgck/2^4	fcgck/2^2	fcgck/2	fcgck	
8-bit timer modes	8-bit timer	O	O	O	O	O	O	O	O	-
	8-bit event counter	-	-	-	-	-	-	-	-	O
	8-bit PWM	O	O	O	O	O	O	O	O	-
	8-bit PPG	O	O	O	O	O	O	O	O	-
16-bit timer modes	16-bit timer	O	O	O	O	O	O	O	O	-
	16-bit event counter	-	-	-	-	-	-	-	-	O
	12-bit PWM	O	O	O	O	O	O	O	O	O
	16-bit PPG	O	O	O	O	O	O	O	O	O

Table 10.6 Operation Modes and Usable Source Clocks (NORMAL1/2 and IDLE1/2 Modes)

Note 1): ○: Usable, -: Unusable

Note 2): Set the source clock in the 16-bit modes on the TC01 side (TCK1).

Note 3): When the low-frequency clock, fs, is not oscillating, it must not be selected as the source clock. If fs is selected when it is not oscillating, no source clock is supplied to the timer, and the timer remains stopped.

Note 4): i=0, 1 (i=0 only in the 16-bit modes)

TCK0		000	001	010	011	100	101	110	111	TC0i
Operation mode		$fs/2^4$	$fs/2^3$	-	-	-	-	-	$fs/2^2$	pin input
8-bit timer modes	8-bit timer	O	O	-	-	-	-	-	O	-
	8-bit event counter	-	-	-	-	-	-	-	-	O
	8-bit PWM	O	O	-	-	-	-	-	O	-
	8-bit PPG	O	O	-	-	-	-	-	O	-
16-bit timer modes	16-bit timer	O	O	-	-	-	-	-	O	-
	16-bit event counter	-	-	-	-	-	-	-	-	O
	12-bit PWM	O	O	-	-	-	-	-	O	O
	16-bit PPG	O	O	-	-	-	-	-	O	O

Table 10.7 Operation Modes and Usable Source Clocks (SLOW1/2 and SLEEP1 Modes)

Note 1): ○: Usable, -: Unusable

Note 2): Set the source clock in the 16-bit modes on the TC01 side (TCK1).

Note 3): i=0, 1 (i=0 only in the 16-bit modes)

10.5.3 Low Power Consumption Function

Timer counters 00 and 01 have the low power consumption registers (POFFCR0) that save power when the timers are not used. Setting POFFCR0 <TC001EN> to "0" disables the basic clock supply to timer counters 00 and 01 to save power. Note that this renders the timers unusable. Setting POFFCR0 <TC001EN> to "1" enables the basic clock supply to timer counters 00 and 01 and allows the timers to operate.

After reset, POFFCR0 <TC001EN> are initialized to "0", and this makes the timers unusable. When using the timers for the first time, be sure to set POFFCR0 <TC001EN> to "1" in the initial setting of the program (before the timer control registers are operated).

Do not change POFFCR0 <TC001EN> to "0" during the timer operation. Otherwise timer counters 00 and 01 may operate unexpectedly.

10.5.4 Function

Timer counters TC00 and TC01 have 8-bit modes in which they are used independently and 16-bit modes in which they are cascaded. (The same to TC02 and TC03)

The 8-bit modes include four operation modes: 8-bit timer mode, 8-bit event counter mode, 8-bit pulse width modulation output (PWM) mode and 8-bit programmable pulse generated output (PPG) mode.

The 16-bit modes include four operation modes: the 16-bit timer mode, the 16-bit event counter mode, the 12-bit PWM mode and the 16-bit PPG mode.

10.5.4.1 8-bit Timer Mode

In the 8-bit timer mode, the up counter counts up using the internal clock, and interrupts can be generated regularly at specified times. The operation of TC00 is described below, and the same applies to the operation of TC01, TC02 and TC03. (Replace TC00- by TC01-, TC02- or TC03-).

(a) Setting

TC00 is put into the 8-bit timer mode by setting T00MOD <TCM0> to "00" or "01", T001CR <TCAS> to "0" and T00MOD <EIN0> to "0". Select the source clock at T00MOD <TCK0>. Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD <DBE0> to "1" to use the double buffer.

Setting T001CR <T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

(b) Operation

Setting T001CR <T00RUN> to "1" allows the 8-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the T00REG set value is detected, an INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR <T00RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x00".

(c) Double Buffer

The double buffer can be used for T00REG by setting T00MOD <DBE0>. The double buffer is disabled by setting T00MOD <DBE0> to "0" or enabled by setting T00MOD <DBE0> to "1".

1. When the Double Buffer is Enabled

When a write instruction is executed on T00REG during the timer operation, the set value is initially stored in the double buffer, and T00REG is not immediately updated. T00REG compares the previous set value with the up counter value. When the values match, an INTTC00 interrupt request is generated and the double buffer set value is stored in T00REG. Subsequently, the match detection is executed using a new set value.

When a write instruction is executed on T00REG while the timer is stopped, the set value is immediately stored in both the double buffer and T00REG.

2. When the Double Buffer is Disabled

When a write instruction is executed on T00REG during the timer operation, the set value is immediately stored in T00REG. Subsequently, the match detection is executed using a new set value.

If the value set to T00REG is smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If the value set to T00REG is equal to the up counter value, the match detection is executed immediately after data is written into T00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock (Figure 10.11). If these are problems, enable the double buffer.

When a write instruction is executed on T00REG while the timer is stopped, the set value is immediately stored in T00REG.

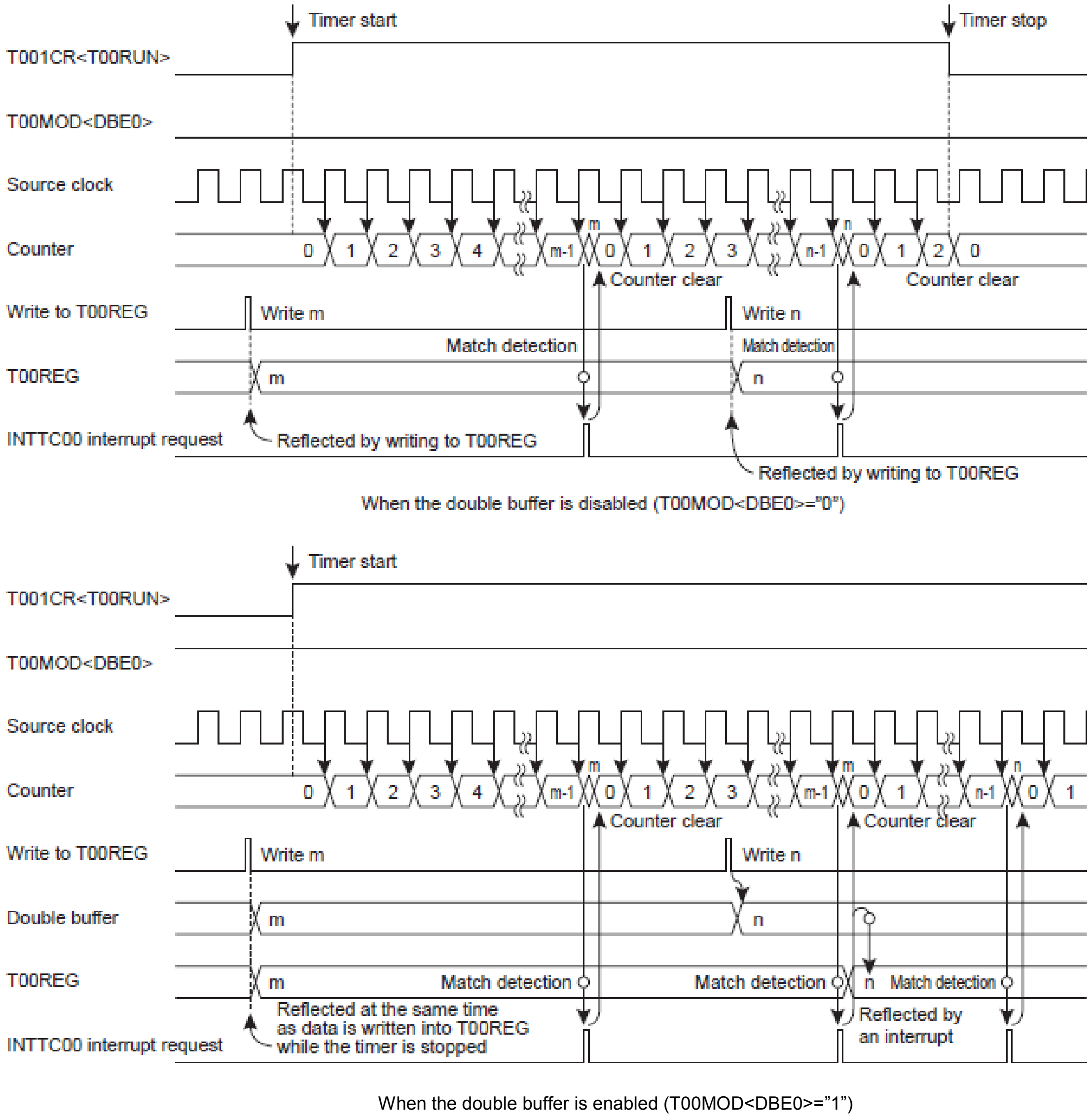
When a read instruction is executed on T00REG, the last value written into T00REG is read out, regardless of the T00MOD <DBE0> setting.

T00MOD <TCK0>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK>="0"	SYSCR1<DV9CK>="1"					
000	fcgck/2^{11}	fs/2^4	fs/2^4	204.8μs	488.2μs	52.2ms	124.5ms
001	fcgck/2^{10}	fs/2^3	fs/2^3	102.4μs	244.1μs	26.1ms	62.3ms
010	fcgck/2^8	fcgck/2^8	-	25.6μs	-	6.5ms	-
011	fcgck/2^6	fcgck/2^6	-	6.4μs	-	1.6ms	-
100	fcgck/2^4	fcgck/2^4	-	1.6μs	-	408μs	-
101	fcgck/2^2	fcgck/2^2	-	400ns	-	102μs	-
110	fcgck/2	fcgck/2	-	200ns	-	51μs	-
111	fcgck	fcgck	fs/2^2	100ns	122.1μs	25.5μs	31.1ms

Table 10.8 8-bit Timer Mode Resolution and Maximum Time Setting

Example: Operate TC00 in the 8-bit timer mode with the operation clock of fcgck/2^2 [Hz] and generate interrupts at 64 μs intervals (fcgck = 10 MHz)

```
LD      (POFFCR0),0x10       ; Sets TC001EN to "1"
DI                           ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4             ; Sets the INTTC00 interrupt enable register to "1"
EI                           ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xE8        ; Selects the 8-bit timer mode and fcgck/2^2
LD      (T00REG),0xA0        ; Sets the timer register (64μs / (2^2/fcgck) = 0xA0)
SET     (T001CR).0           ; Starts TC00
```



When the double buffer is disabled (T00MOD<DBE0>="0")

When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 10.11 Timer Mode Timing Chart

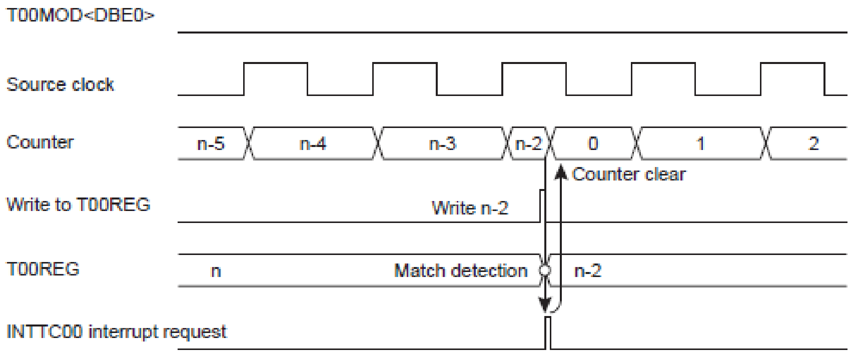


Figure 10.12 Operation When T00REG and the Up Counter Have the Same Value

10.5.4.2 8-bit Event Counter Mode

In the 8-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 or TC01 pin (and similarly, TC02 or TC03 pin). The operation of TC00 is described below, and the same applies to the operation of TC01, TC02 and TC03.

(a) Setting

Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG. TC00 is put into the 8-bit event counter mode by setting T00MOD <TCM0> to "00", T001CR <TCAS> to "0" and T00MOD <EIN0> to "1". Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

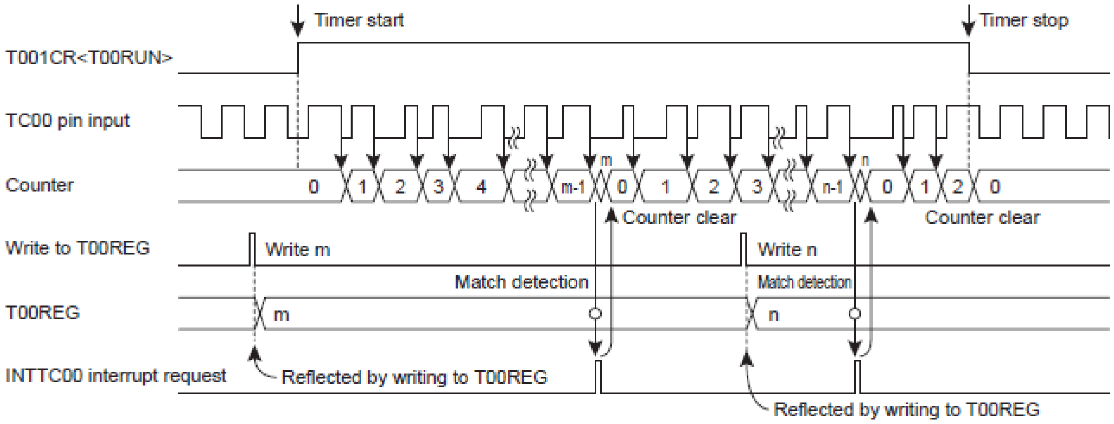
Set T00MOD <DBE0> to "1" to use the double buffer.

Setting T001CR <T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

(b) Operation

Setting T001CR <T00RUN> to "1" allows the 8-bit up counter to increment at the falling edge of the TC00 pin. When a match between the up counter value and the T00REG set value is detected, an INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR <T00RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x00".

The maximum frequency to be supplied is fcgck/2^2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.



When the double buffer is disabled (T00MOD<DBE0>="0")

Figure 10.13 Event Counter Mode Timing Chart

(c) **Double Buffer**

Refer to “10.5.3.1 - (c) Double Buffer”.

Example: Operate TC00 in the 8-bit event counter mode and generate an interrupt each time 16 falling edges are detected at the TC00 pin.

```
LD      (POFFCR0),0x10       ; Sets TC001EN to "1"
DI                           ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4             ; Sets the INTTC00 interrupt enable register to "1"
EI                           ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xC4        ; Selects to the 8-bit event counter mode
LD      (T00REG),0x10        ; Sets the timer register
SET     (T001CR).0           ; Starts TC00
```

10.5.4.3 8-bit Pulse Width Modulation (PWM) Output Mode

The pulse-width modulated pulses with a resolution of 7 bits are output in the 8-bit PWM mode. An additional pulse can be added to the 2 × n-th duty pulse. This enables PWM output with a resolution nearly equivalent to 8 bits. (n=1, 2, 3...)

The operation of TC00 is described below, and the same applies to the operation of TC01, and similarly, TC02 and TC03.

(a) **Setting**

TC00 is put into the 8-bit PWM mode by setting T00MOD <TCM0> to "10" and T001CR <TCAS> to "0". To use the internal clock as the source clock, set T00MOD <EIN0> to "0" and select the clock at T00MOD <TCK0>. To use an external clock as the source clock, set T00MOD <EIN0> to "1". Set the count value to be used for the match detection and the additional pulse value at the PWM register T00PWM.

Set T00MOD <DBE0> to "1" to use the double buffer.

Setting T001CR <T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

In the 8-bit PWM mode, the T00PWM register is configured as follows:

Timer Register 00

T00PWM (0x0028)	7	6	5	4	3	2	1	0
Bit Symbol	PWMDUTY							PWMAD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Timer Register 01

T01PWM (0x0029)	7	6	5	4	3	2	1	0
Bit Symbol	PWMDUTY							PWMAD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

PWMDUTY is a 7-bit register used to set the duty pulse width value (the time before the first output change) in a cycle (128 counts of the source clock).

PWMAD is a register used to set the additional pulse. When PWMAD is "1", an additional pulse that corresponds to 1 count of the source clock is added to the 2 × n-th duty pulse (n=1, 2, 3...). In other words, the 2 × n-th duty pulse has the output of PWMDUTY+1.

The additional pulse is not added when PWMAD is "0".

Set the initial state of the PWM00B pin at T00MOD <TFF0>. Setting T00MOD <TFF0> to "0" selects the "L" level as the initial state of the PWM00B pin. Setting T00MOD <TFF0> to "1" selects the "H" level as the initial state of the PWM00B pin. If the PWM00B pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD <TFF0> is output to the PWM00B pin. Table 10.9 shows the list of output levels of the PWM00B pin.

And by setting "1" to T001CR <OUTAND> bit, a logical product (AND) pulse of TC00 and TC01's output can be output to PWM00B pin. By using this function, the remote-control waveform can be created easily.

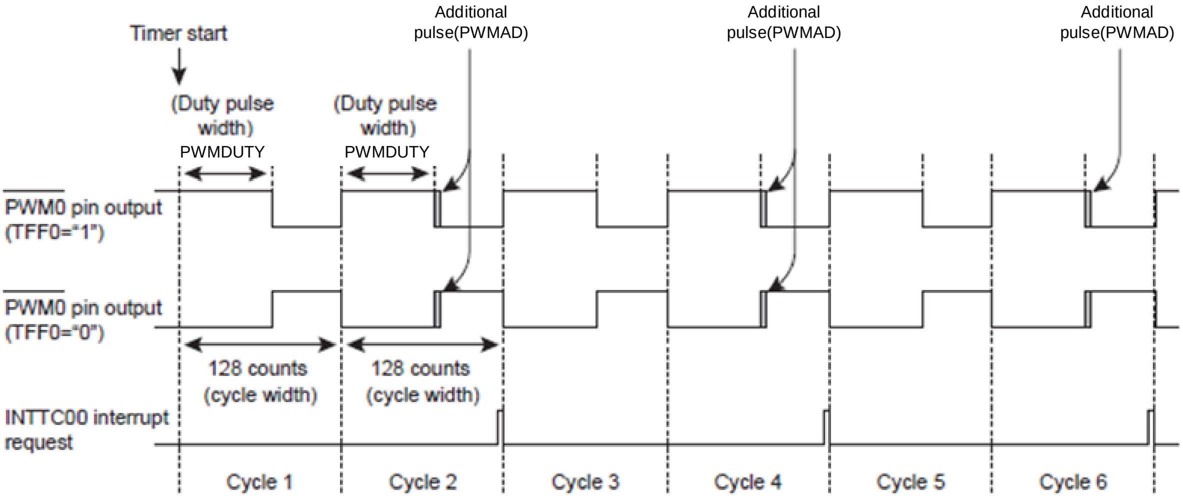
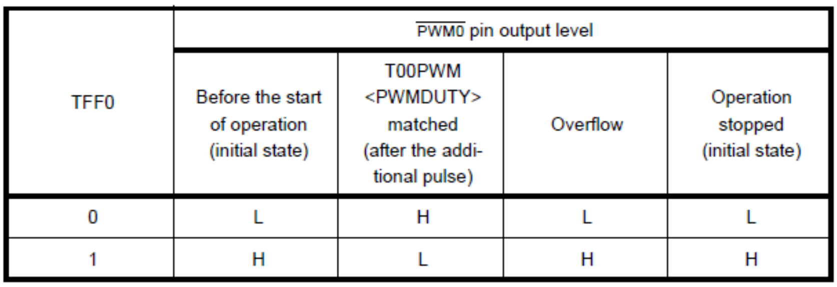


Figure 10.14 PWM00B Pulse Output



TFF0	$\overline{\text{PWM0}}$ pin output level			
	Before the start of operation (initial state)	T00PWM <PWMDUTY> matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

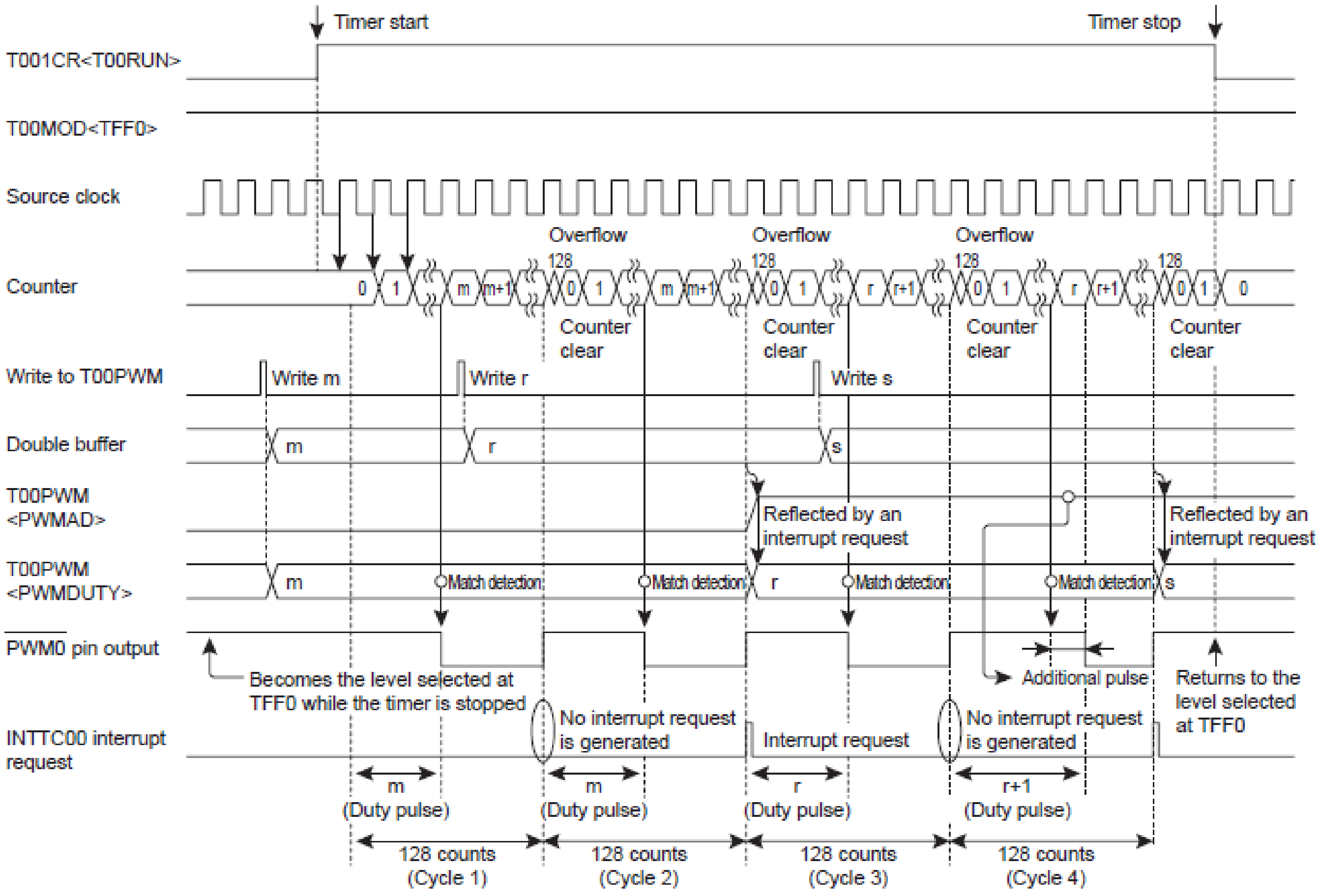
Table 10.9 List of Output Levels of PWM00B Pin

(b) Operation

Setting T001CR <T00RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the lower 7 bits of the up counter value and the value set to T00PWM <PWMDUTY> is detected, the output of the PWM00B pin is reversed. When T00MOD <TFF0> is "0", the PWM00B pin changes from the "L" to "H" level. When T00MOD <TFF0> is "1", the PWM00B pin changes from the "H" to "L" level.

If T00PWM <PWMAD> is "1", an additional pulse that corresponds to 1 count of the source clock is added at the 2 × n-th match detection (n=1, 2, 3...). In other words, the PWM00B pin output is reversed at the timing of T00PWM <PWMDUTY> +1. When T00MOD <TFF0> is "0", the period of the "L" level becomes longer than the value set to T00PWM <PWMDUTY> by 1 source clock. When T00MOD <TFF0> is "1", the period of the "H" level becomes longer than the value set to T00PWM <PWMDUTY> by 1 source clock. This function allows two cycles of output pulses to be handled with a resolution nearly equivalent to 8 bits.

No additional pulse is inserted when T00PWM <PWMAD> is "0".



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 10.15 8-bit PWM Mode Timing Chart

Subsequently, the up counter continues counting up. When the up counter value reaches 128, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of PWM00B pin is reversed. When T00MOD <TFF0> is "0", the PWM00B pin changes from the "H" to "L" level. When T00MOD <TFF0> is "1", the PWM00B pin changes from the "L" to "H" level. If the 2 × n-th overflow occurs at this time, an INTTC00 interrupt request is generated. (No interrupt request is generated at the 2 × n-th -1 overflow.) Subsequently, the up counter continues counting up.

When T001CR <T00RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The PWM00B pin returns to the level selected at T00MOD <TFF0>.

When an external source clock is selected, the maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

Example: Operate TC00 in the 8-bit PWM mode with the operation clock of fcgck/2 and output a duty pulse nearly equivalent to 11.6 μs (fcgck = 10 MHz)
(Actually, output a total duty pulse of 23.2 μs in 2 cycles (51.2 μs))

```
SET     (P7FC).0             ; Sets P7FC0 to "1"
SET     (P7CR).0             ; Sets P7CR0 to "1"
LD      (POFFCR0),0x10       ; Sets TC001EN to "1"
DI                           ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4             ; Sets the INTTC00 interrupt enable register to "1"
EI                           ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xF2        ; Selects the 8-bit PWM mode and fcgck/2
LD      (T00PWM),0x74        ; Sets the timer register (duty pulse)
                             ; (11.6μs × 2) / (2/fcgck) = 0x74
SET     (T001CR).0           ; Starts TC00
```

(c) Double Buffer

The double buffer can be used for T00PWM by setting T00MOD <DBE0>. The double buffer is disabled by setting T00MOD <DBE0> to "0" or enabled by setting T00MOD <DBE0> to "1".

1. When the Double Buffer is Enabled

When a write instruction is executed on T00PWM during the timer operation, the set value is first stored in the double buffer, and T00PWM is not updated immediately. T00PWM compares the previous set value with the up counter value. When the 2 × n-th overflow occurs, an INTTC00 interrupt request is generated and the double buffer set value is stored in T00PWM. Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on T00PWM, the value in the double buffer (the last set value) is read out, not the T00PWM value (the currently effective value). When a write instruction is executed on T00PWM while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWM.

2. When the Double Buffer is Disabled

When a write instruction is executed on T00PWM during the timer operation, the set value is immediately stored in T00PWM. Subsequently, the match detection is executed using a new set value. If the value set to T00PWM is smaller than the up counter value, the PWM00B pin is not reversed until the up counter overflows and match detection is executed using a new set value. If the value set to T00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM. Therefore, the timing of changing the PWM00B pin may not be an integral multiple of the source clock (Figure 10.15). Similarly, if T00PWM is set during the additional pulse output, the timing of changing the PWM00B pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

When a write instruction is executed on T00PWM while the timer is stopped, the set value is immediately stored in T00PWM.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

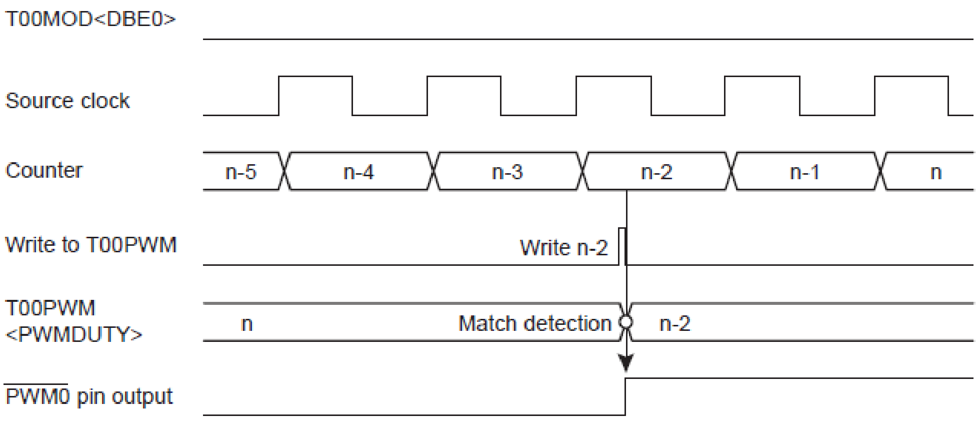


Figure 10.16 Operation When T00PWM and the Up Counter Have the Same Value

T00MOD <TCK0>	Source clock [Hz]			Resolution		7-bit cycle (period × 2)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=8MHz	fs=32.768KHz	fcgck=8MHz	fs=32.768KHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	fcgck/2^{11}	fs/2^4	fs/2^4	256us	488.2us	32.8ms (65.5ms)	62.5ms (125ms)
001	fcgck/2^{10}	fs/2^3	fs/2^3	128us	244.1us	16.4ms (32.8ms)	31.3ms (62.5ms)
010	fcgck/2^8	fcgck/2^8	-	32us	-	4.1ms (8.2ms)	-
011	fcgck/2^6	fcgck/2^6	-	8us	-	1.0ms (2.0ms)	-
100	fcgck/2^4	fcgck/2^4	-	2us	-	256us (512us)	-
101	fcgck/2^2	fcgck/2^2	-	500ns	-	64us (128us)	-
110	fcgck/2	fcgck/2	-	250ns	-	32us (64us)	-
111	fcgck	fcgck	fs/2^2	125ns	122.1us	16us (32us)	15.6ms (31.3ms)

Table 10.10 Resolutions and Cycles in the 8-bit PWM Mode

10.5.4.4 8-bit Programmable Pulse Generate (PPG) Output Mode

In the 8-bit PPG mode, the pulses with arbitrary duty and cycle are output by using the T00REG and T00PWM registers.

By setting the T001CR <OUTAND> register, a pulse that is a logical ANDed product of the TC00 and TC01 outputs can be output to the TC01 pin. This function facilitates the generation of remote-controlled waveforms, for example.

The operation of TC00 is described below, and the same applies to the operation of TC01, and similarly, TC02 and TC03.

(a) Setting

TC00 is put into the 8-bit PPG mode by setting T00MOD <TCM0> to "1" and T001CR <TCAS> to"0". To use the internal clock as the source clock: Set T00MOD <EIN0> to "0" and select the clock at T00MOD <TCK0>. To use an external clock as the source clock, set T00MOD <EIN0> to "1". Set the duty pulse width at T00PWM and the cycle width at T00REG.

Set T00MOD <DBE0> to "1" to use the double buffer.

Setting T001CR <T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

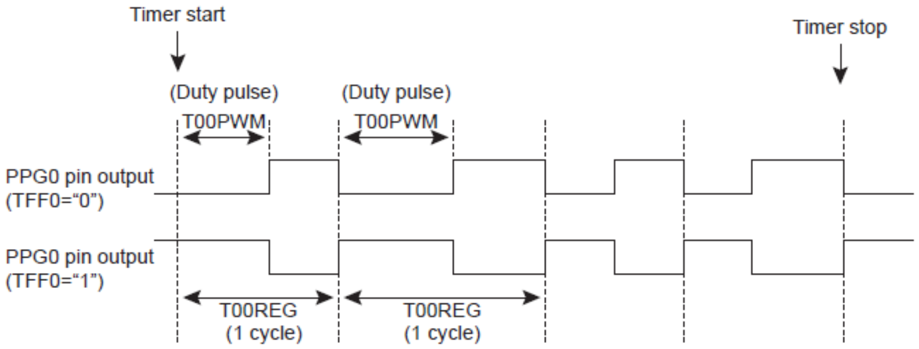


Figure 10.17 PPG00B Pulse Output

Set the initial state of the PPG0B pin at T00MOD <TFF0>. Setting T00MOD <TFF0> to "0" selects the "L" level as the initial state of the PPG0B pin. Setting T00MOD <TFF0> to "1" selects the "H" level as the initial state of the PPG0B pin. If the PPG0B pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD <TFF0> is output to the PPG0B pin. Table 10.11 shows the list of output levels of the PPG0B pin.

Setting the T001CR <OUTAND> bit to "1" allows the PPG0B pin to output a pulse that is a logical ANDed product of the TC00 and TC01 outputs.

TFF0	$\overline{\text{PPG0}}$ pin output level			
	Before the start of operation (initial state)	T00PWM matched	T00REG matched	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

Table 10.11 List of Output Levels of PPG00B Pin

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

(b) Operation

Setting T001CR <T00RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the internal up counter value and the value set to T00PWM is detected, the output of the PPG0B pin is reversed. When T00MOD <TFF0> is "0", the PPG0B pin changes from the "L" to "H" level. When T00MOD <TFF0> is "1", the PPG0B pin changes from the "H" to "L" level.

Subsequently, the up counter continues counting up. When a match between the up counter value and T00REG is detected, the output of the PPG0B pin is reversed again. When T00MOD <TFF0> is "0", the PPG0B pin changes from the "H" to "L" level. When T00MOD <TFF0> is "1", the PPG0B pin changes from the "L" to "H" level. At this time, an INTTC00 interrupt request is generated.

When T001CR <T00RUN> is set to "0" during the operation, the up counter is stopped and cleared to "0x00". The PPG0B pin returns to the level selected at T00MOD <TFF0>.

When the external source clock is selected, the maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

(c) Double Buffer

The double buffer can be used for T00PWM and T00REG by setting T00MOD <DBE0>. The double buffer is disabled by setting T00MOD <DBE0> to "0" or enabled by setting T00MOD <DBE0> to "1".

1. When the Double Buffer is Enabled

When a write instruction is executed on T00PWM (T00REG) during the timer operation, the set value is first stored in the double buffer, and T00PWM (T00REG) is not updated immediately. T00PWM (T00REG) compares the previous set value with the up counter value. When an INTTC00 interrupt request is generated, the double buffer set value is stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on T00PWM (T00REG), the value in the double buffer (the last set value) is read out, not the T00PWM (T00REG) value (the currently effective value).

When a write instruction is executed on T00PWM (T00REG) while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWM (T00REG).

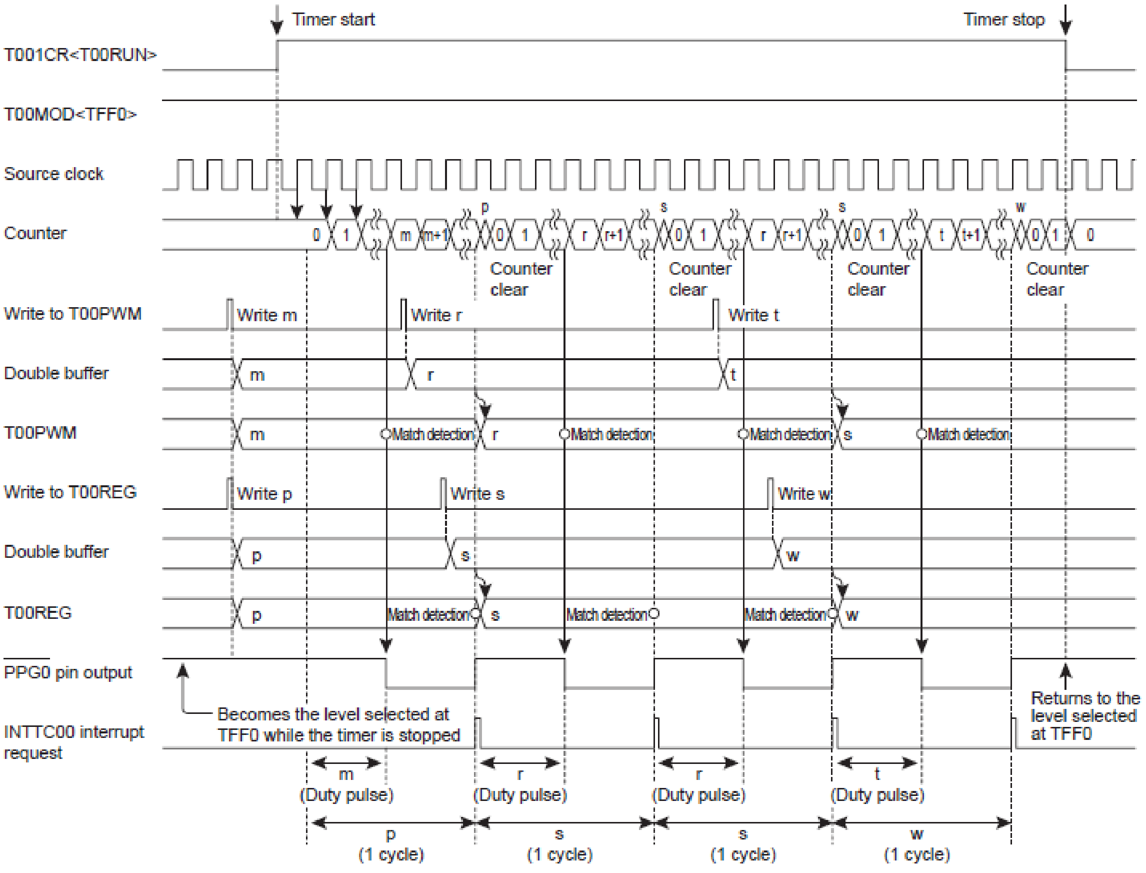
2. When the Double Buffer is Disabled

When a write instruction is executed on T00PWM (T00REG) during the timer operation,

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

the set value is immediately stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value. If the value set to T00PWM (T00REG) is smaller than the up counter value, the PPG0B pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T00PWM (T00REG) is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM (T00REG). Therefore, the timing of changing the PPG0B pin may not be an integral multiple of the source clock (Figure 10.18). If these are problems, enable the double buffer.

When a write instruction is executed on T00PWM (T00REG) while the timer is stopped, the set value is immediately stored in T00PWM (T00REG).



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 10.18 8-bit PPG Mode Timing Chart

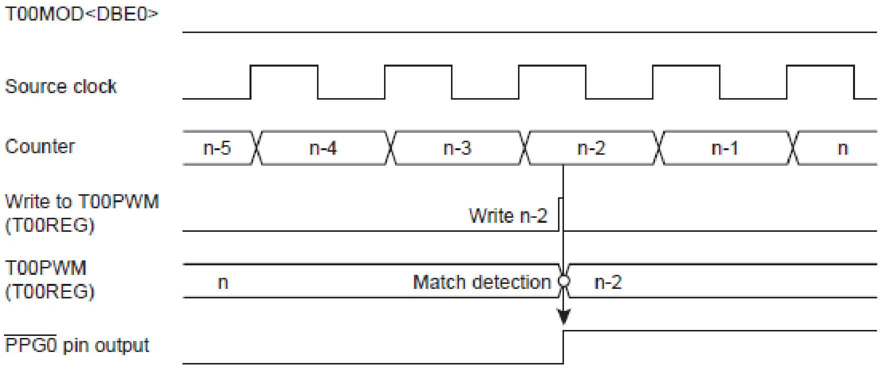


Figure 10.19 Operation When T00PWM (T00REG) and the Up Counter Have the Same Value

Example: Operate TC00 in the 8-bit PPG mode with the operation clock of fcgck/2 and output the 8μs duty pulse in 32μs cycles (fcgck = 10 MHz)

```
SET     (P7FC).0             ; Sets P7FC0 to "1"
SET     (P7CR).0             ; Sets P7CR0 to "1"
LD      (POFFCR0),0x10       ; Sets TC001EN to "1"
DI                           ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4             ; Sets the INTTC00 interrupt enable register to "1"
EI                           ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xF3        ; Selects the 8-bit PPG mode and fcgck/2
LD      (T00REG),0xA0        ; Sets the timer register (cycle)
                             ; 32μs/(2/fcgck) = 0xA0
LD      (T00PWM),0x28        ; Sets the timer register (duty pulse)
                             ; 8us/(2/fcgck) = 0x28
SET     (T001CR).0           ; Starts TC00
```

10.5.4.5 16-bit Timer Mode

In the 16-bit timer mode, TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer. (The same to TC02 and TC03)

(a) Setting

Setting T001CR <TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit timer mode is activated by setting T01MOD <TCM1> to "00" or "01" and T01MOD <EIN1> to "0". Select the source clock at T01MOD <TCK1>.

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are reflected on the double buffer or T01+00REG when a write instruction is executed on T01REG. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD <DBE1> to "1" to use the double buffer.

Setting T001CR <T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR <T00RUN> and <T01RUN> are "0".)

(b) Operation

Setting T001CR <T01RUN> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the T00+01REG set value is detected, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR <T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

(c) Double Buffer

The double buffer can be used for T01+00REG by setting T01MOD <DBE1>. The double buffer is disabled by setting T01MOD <DBE1> to "0" or enabled by setting T01MOD <DBE1> to "1".

1. When the Double Buffer is Enabled

When write instructions are executed on T00REG and T01REG in this order during the timer operation, the set value is first stored in the double buffer, and T01+00REG is not updated immediately. T01+00REG compares the previous set value with the up counter value. When the values are matched, an INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00REG. Then, the match detection is executed using a new set value.

When write instructions are executed on T00REG and T01REG in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00REG.

2. When the Double Buffer is Disabled

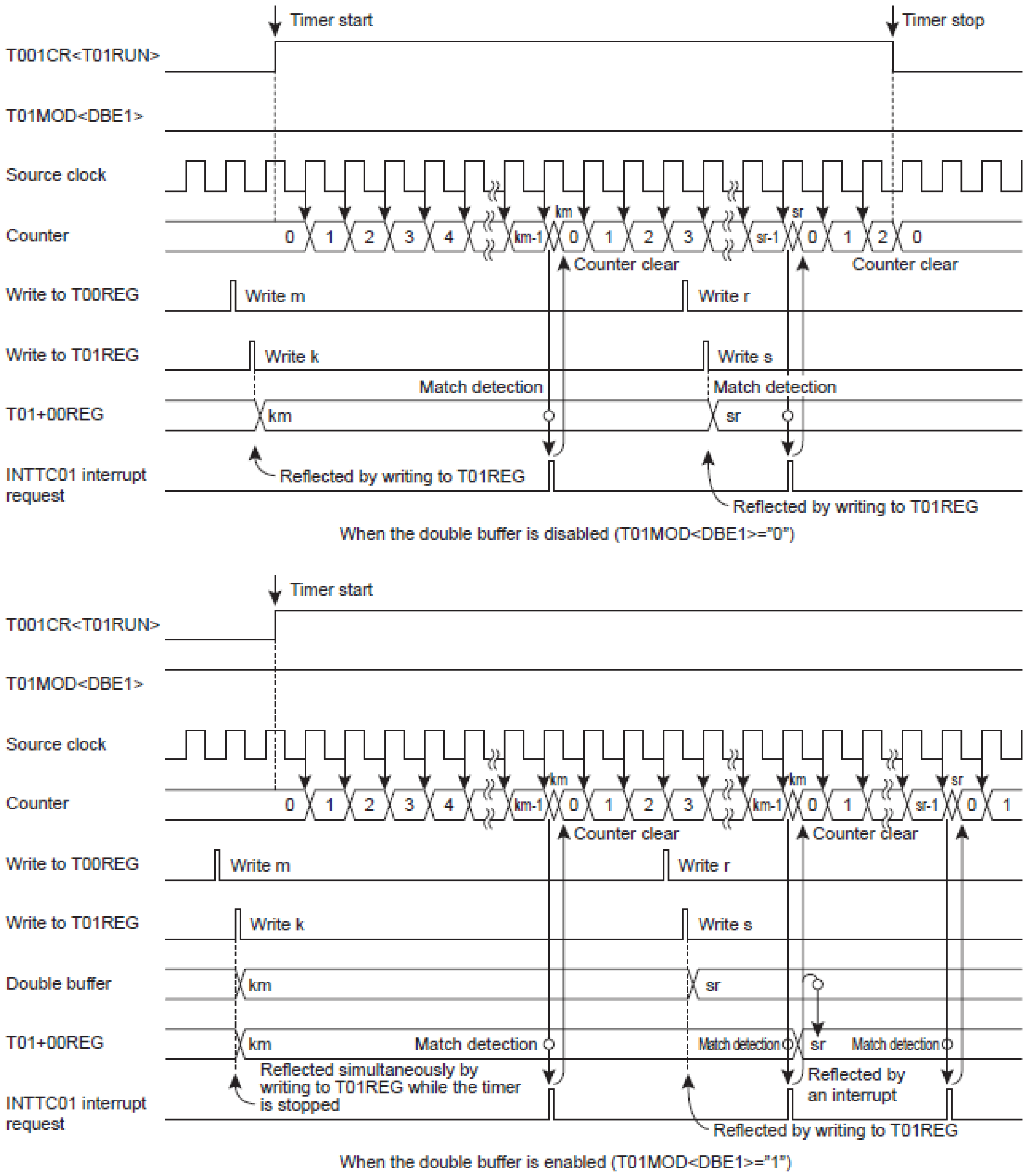
When write instructions are executed on T00REG and T01REG in this order during the timer operation, the set value is immediately stored in T01+00REG. Subsequently, the match detection is executed using a new set value.

If the value set to T01+00REG is smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If the value set to T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

When write instructions are executed on T00REG and T01REG in this order while the

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

timer is stopped, the set value is immediately stored in T01+00REG. When a read instruction is executed on T01+00REG, the last value written into T01+00REG is read out, regardless of the T00MOD <DBE1> setting.

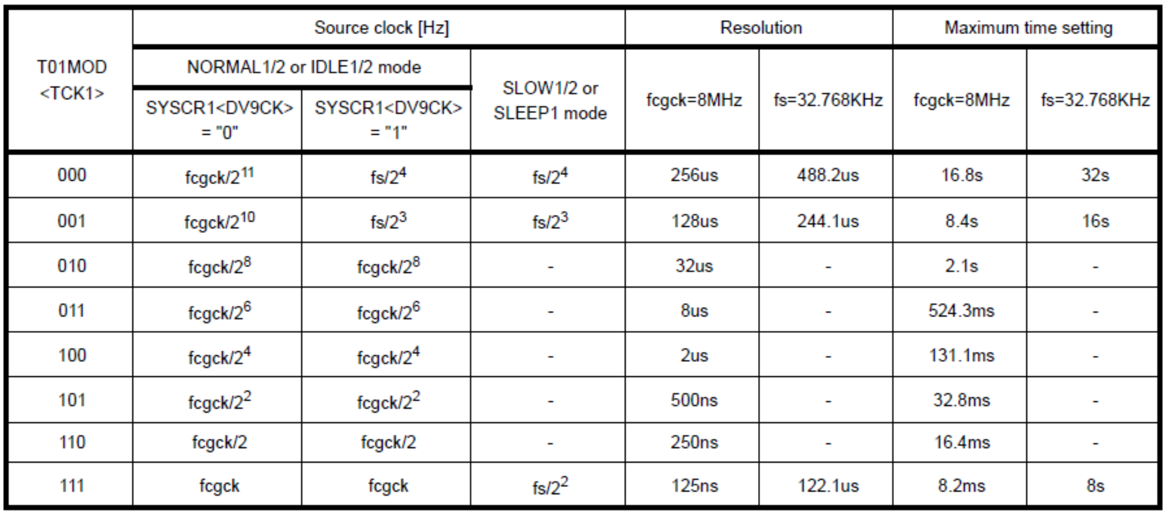


When the double buffer is disabled (T01MOD<DBE1>="0")

When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 10.20 16-bit Timer Counter Timing Chart

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.



T01MOD <TCK1>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=8MHz	fs=32.768KHz	fcgck=8MHz	fs=32.768KHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	fcgck/2^{11}	fs/2^4	fs/2^4	256us	488.2us	16.8s	32s
001	fcgck/2^{10}	fs/2^3	fs/2^3	128us	244.1us	8.4s	16s
010	fcgck/2^8	fcgck/2^8	-	32us	-	2.1s	-
011	fcgck/2^6	fcgck/2^6	-	8us	-	524.3ms	-
100	fcgck/2^4	fcgck/2^4	-	2us	-	131.1ms	-
101	fcgck/2^2	fcgck/2^2	-	500ns	-	32.8ms	-
110	fcgck/2	fcgck/2	-	250ns	-	16.4ms	-
111	fcgck	fcgck	fs/2^2	125ns	122.1us	8.2ms	8s

Table 10.12 16-bit Timer Mode Resolution and Maximum Time Setting

Example: Operate TC00 and TC01 in the 16-bit timer mode with the operation clock of fcgck/2 [Hz] and generate interrupts at 96 μs intervals (fcgck = 10 MHz)

```
LD      (POFFCR0),0x10       ; Sets TC001EN to "1"
DI                           ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4             ; Sets the INTTC00 interrupt enable register to "1"
EI                           ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF0        ; Selects the 16-bit timer mode and fcgck/2
LD      (T00REG),0xE0        ; Sets the timer register (96us/(2/fcgck)=0x1E0)
LD      (T01REG),0x01        ; Sets the timer register
LD      (T01CR) ,0x06        ; Starts TC00 and TC001(16-bit mode)
```

10.5.4.6 16-bit Event Counter Mode

In the 16-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 pin. TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer. (The same to TC02 and TC03)

(a) Setting

Setting T001CR <TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode. The 16-bit timer mode is activated by setting T01MOD <TCM1> to "00" or "01" and T01MOD <EIN0> to “1”.

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and set the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are reflected on the double buffer or T01+00REG when a write instruction is executed on T01REG. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD <DBE1> to "1" to use the double buffer.

Setting T001CR <T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR <T00RUN> and <T01RUN> are "0".)

(b) Operation

Setting T001CR <T01RUN> to "1" allows the 16-bit up counter to increment at the falling edge of the TC00 pin. When a match between the up counter value and the T00+01REG set value is detected, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR <T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

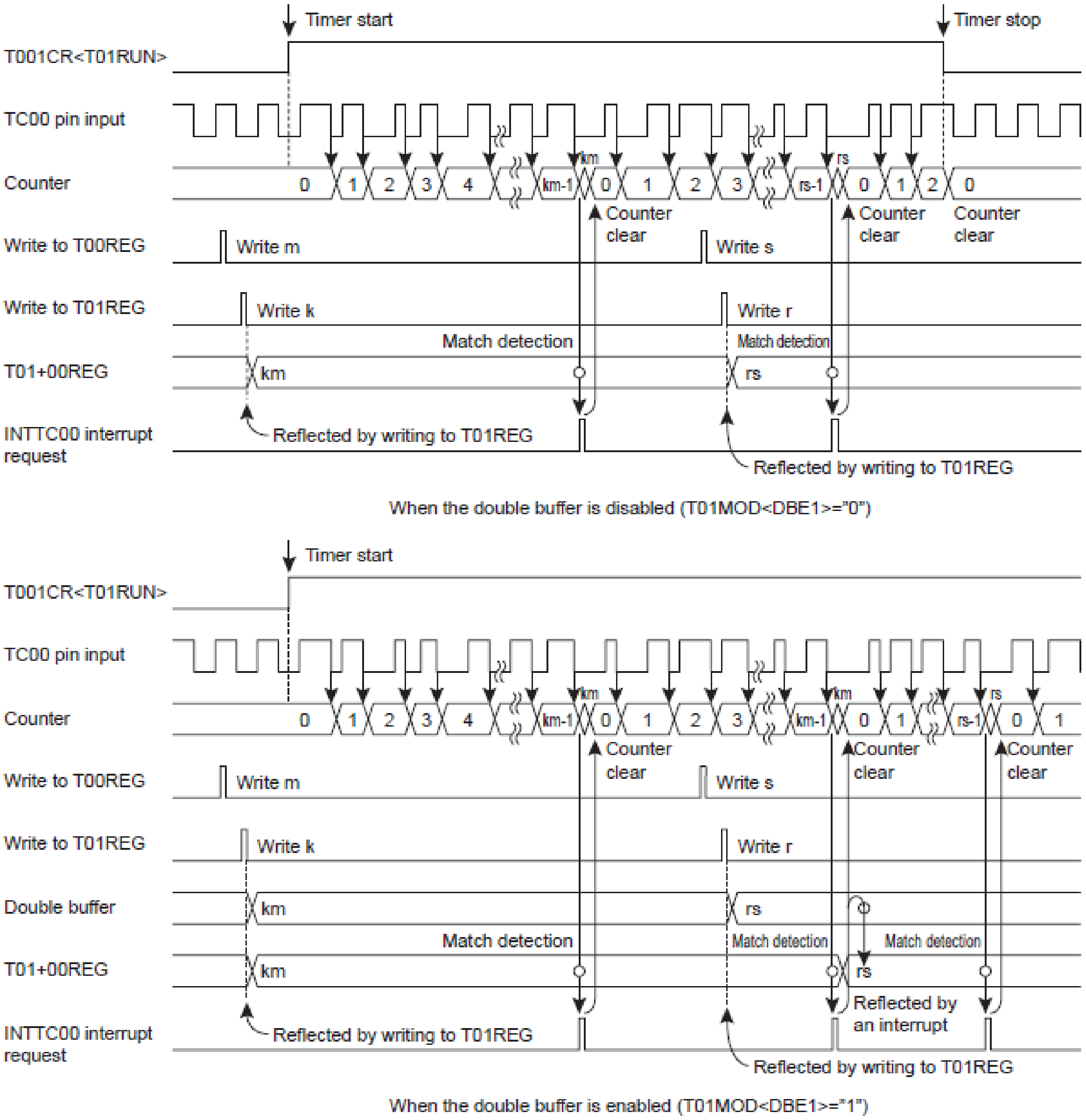

Timer start
Timer stop
T001CR<T01RUN>
TC00 pin input
Counter
0
1
2
3
4
km-1
km
rs-1
rs
Counter clear
Write to T00REG
Write m
Write s
Write to T01REG
Write k
Write r
Match detection
T01+00REG
km
rs
INTTC00 interrupt request
Reflected by writing to T01REG
When the double buffer is disabled (T01MOD<DBE1>="0")
Double buffer
Reflected by an interrupt
When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 10.21 16-bit Event Counter Mode Timing Chart

(c) Double Buffer

Refer to “10.5.3.5 - (c) Double Buffer”.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Example: Operate TC00 and TC01 in the 16-bit event counter mode and generate an interrupt each time the 384th falling edge is detected at the TC00 pin

```
LD      (POFFCR0),0x10        ; Sets TC001EN to "1"
DI                            ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4              ; Sets the INTTC00 interrupt enable register to "1"
EI                            ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF0         ; Selects the 16-bit timer mode and fcgck/2
LD      (T00REG),0xE0         ; Sets the timer register (96us/(2/fcgck)=0x1E0)
LD      (T01REG),0x01         ; Sets the timer register
LD      (T01CR) ,0x06         ; Starts TC00 and TC001(16-bit mode)
```

10.5.4.7 12-bit Pulse Width Modulation (PWM) Output Mode

In the 12-bit PWM output mode, TC00 and TC01 are cascaded to output the pulse-width modulated pulses with a resolution of 8 bits. An additional pulse of 4 bits can be inserted, which enables PWM output with a resolution nearly equivalent to 12 bits. (The same to TC02 and TC03)

(a) Setting

Setting T001CR <TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 12-bit PWM mode is selected by setting T01MOD <TCM1> to "10". To use the internal clock as the source clock, set T01MOD <EIN1> to "0" and select the clock at T01MOD <TCK1>. To use an external clock as the source clock, set T01MOD<EIN1> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR <T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR <T00RUN> and <T01RUN> are "0".)

Set the count value to be used for the match detection and the additional pulse value as a 12-bit value at the timer registers T00PWM and T01PWM. Set bits 11 to 8 of the 12-bit value at the lower 4 bits of T01PWM and set bits 7 to 0 at T00PWM. Refer to the following table for the register configuration. (Hereinafter, the 12-bit value specified by the combined setting of T00PWM and T01PWM is indicated as T01+00PWM.) The timer register settings are reflected on the double buffer or T01+00PWM when a write instruction is executed on T01PWM. Be sure to execute the write instructions on T00PWM and T01PWM in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Timer Register 00

T00PWM (0x0028)	7	6	5	4	3	2	1	0
Bit Symbol	PWMDUTYL				PWMAD3	PWMAD2	PWMAD1	PWMAD0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

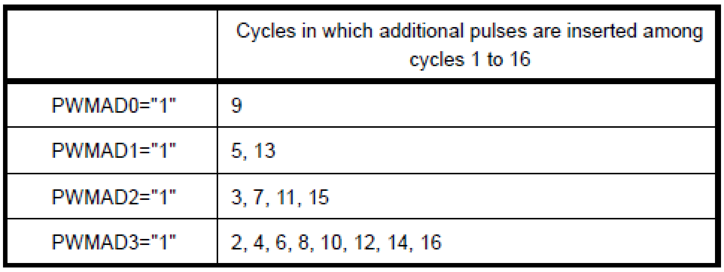
Timer Register 01

T01PWM (0x0029)	7	6	5	4	3	2	1	0
Bit Symbol	-				PWMDUTYH			
Read/Write	-				R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Bits 7 to 4 of T01PWM are not used in the 12-bit PWM mode. However, data can be written to these bits of T01PWM and the written values are read out as they are when the bits are read. Normally, set these bits to "0".

PWMDUTYH and PWMDUTYL are 4-bit registers. They are combined to set an 8-bit value of duty pulse width (time before the first change in the output) for one cycle (256 counts of the source clock). Hereinafter, an 8-bit value specified by the combined setting of PWMDUTYH and PWMDUTYL is indicated as PWMDUTY.

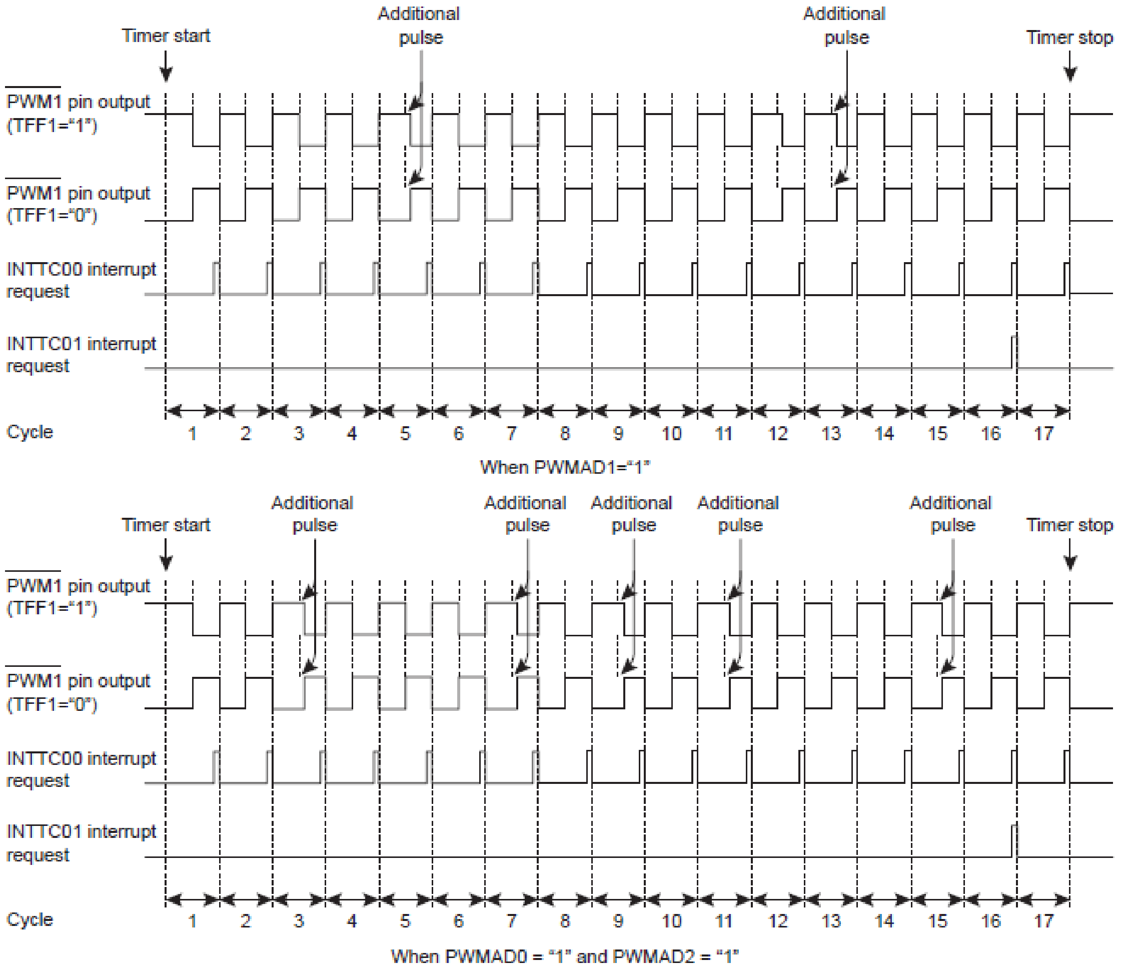
PWMAD3 to PWMAD0 are the additional pulse setting register. Additional pulses can be inserted in specific cycles of the duty pulse by setting each bit to "1". The additional pulses are inserted in the positions listed in Table 10.13. PWMAD3 to PWMAD0 can be combined to specify the number of times of inserting the additional pulses in 16 cycles to any number from 1 to 16. Examples of inserting additional pulses are shown in Figure 10.21.



	Cycles in which additional pulses are inserted among cycles 1 to 16
PWMAD0="1"	9
PWMAD1="1"	5, 13
PWMAD2="1"	3, 7, 11, 15
PWMAD3="1"	2, 4, 6, 8, 10, 12, 14, 16

Table 10.13Cycles in Which Additional Pulses Are Inserted

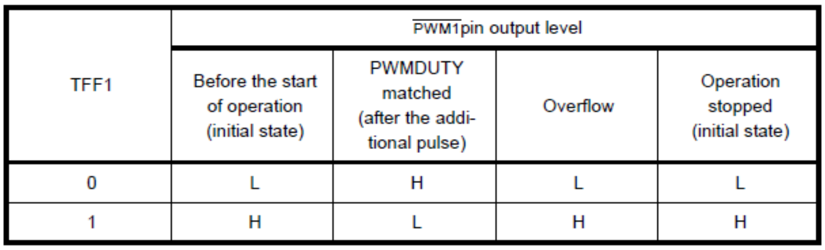
Set the initial state of the PWM1B pin at T01MOD <TFF1>. Setting T01MOD<TFF1> to "0" selects the "L" level as the initial state of the PWM1B pin. Setting T01MOD <TFF1> to "1" selects the "H" level as the initial state of the PWM1B pin. If the PWM1B pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD <TFF1> is output to the PWM1B pin. Table 10.14 shows the list of output levels of the PWM1B pin.



When PWMAD1="1"

When PWMAD0 = "1" and PWMAD2 = "1"

Figure 10.22 Examples of Inserting Additional Pulses



TFF1	$\overline{\text{PWM1}}$pin output level			
	Before the start of operation (initial state)	PWMDUTY matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

Table 10.14 List of Output Levels of PWM01B Pin

(b) Operation

Setting T001CR <T01RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the lower 8 bits of the up counter value and the value set to PWMDUTY is detected, the output of the PWM1B pin is reversed. When T01MOD <TFF1> is "0", the PWM1B pin changes from the "L" to "H" level. When T01MOD <TFF1> is "1", the PWM1B pin changes from the "H" to "L" level.

If any of PWMAD3 to PWMAD0 is "1", an additional pulse that corresponds to 1 count of the source clock is inserted in specific cycles of the duty pulse. In other words, the PWM1B pin output is reversed at the timing of PWMDUTY+1. When T00MOD <TFF0> is "0", the period of the "L" level becomes longer than the value set to PWMDUTY by 1 source clock. When T00MOD <TFF0> is "1", the period of the "H" level becomes longer than the value set to PWMDUTY by 1 source clock. This function allows 16 cycles of output pulses to be handled with a resolution nearly equivalent to 12 bits.

No additional pulse is inserted when PWMAD3 to PWMAD0 are all "0".

Subsequently, the up counter continues counting up. When the up counter value reaches 256, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the PWM1B pin is reversed. When T01MOD <TFF1> is "0", the PWM1B pin changes from the "H" to "L" level. When T01MOD <TFF1> is "1", the PWM1B pin changes from the "L" to "H" level. At this time, an INTTC00 interrupt request is generated (an INTTC00 interrupt request is generated each time an overflow occurs.) An INTTC01 interrupt request is generated at the 16 × n-th overflow (n=1, 2, 3...). Subsequently, the up counter continues counting up.

When T001CR <T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The PWM1B pin returns to the level selected at T01MOD <TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/2^4 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

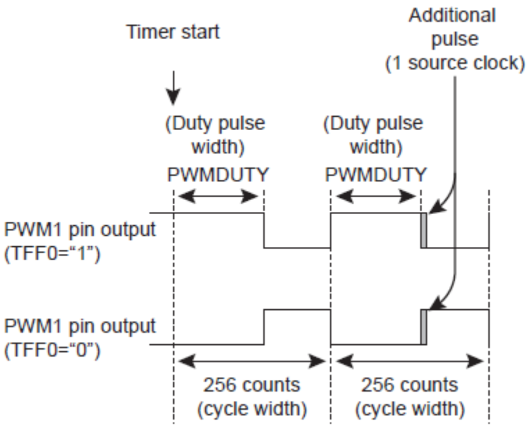
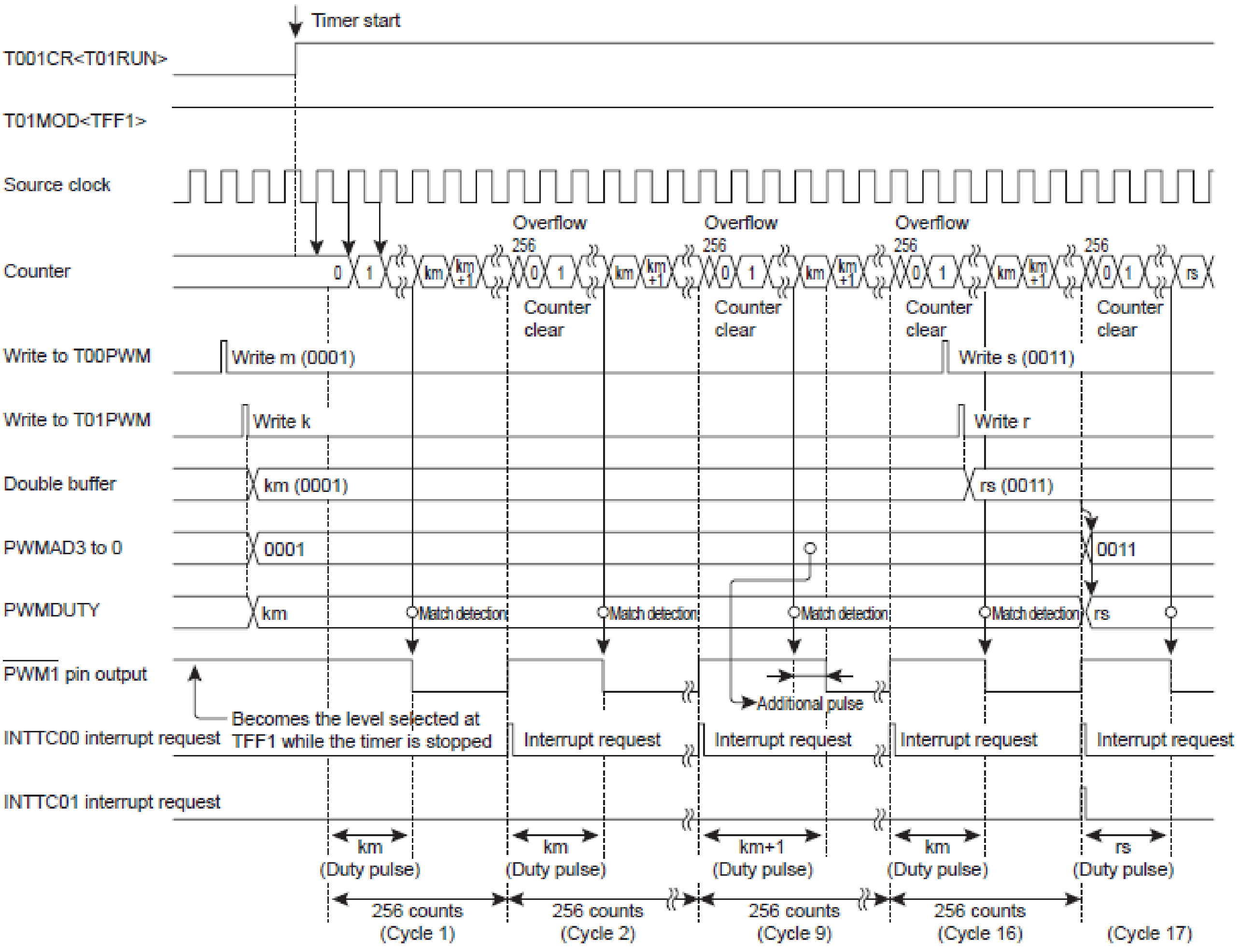


Figure 10.23 PWM1B Pin Output

(c) Double Buffer

The double buffer can be used for T01+00PWM by setting T01MOD <DBE1>. The double buffer is disabled by setting T01MOD <DBE1> to "0" or enabled by setting T01MOD <DBE1> to "1".



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 10.24 12-bit PWM Mode Timing Chart

1. **When the Double Buffer is Enabled**

When write instructions are executed on T00PWM and T01PWM in this order during the timer operation, the set value is first stored in the double buffer, and T01+00PWM is not updated immediately. T01+00PWM compares the previous set value with the up counter value. When the 16 × n-th overflow occurs, an INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00PWM. Subsequently, the match detection is executed using a new set value.

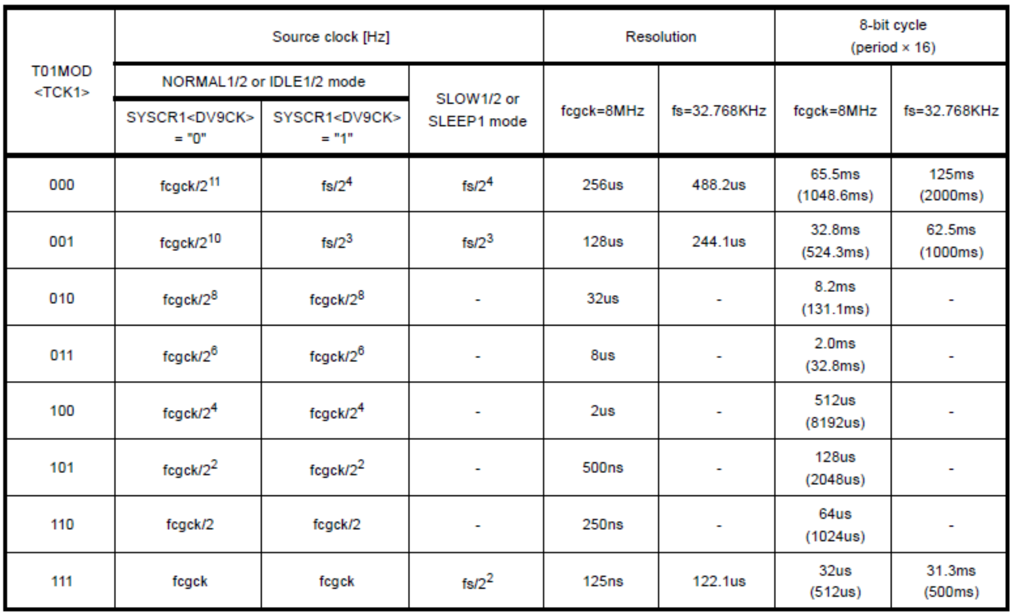
When a read instruction is executed on T01+00PWM (T00REG), the value in the double buffer (the last set value) is read out, not the T01+00PWM value (the currently effective value).

When write instructions are executed on T00PWM and T01PWM in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00PWM.

2. **When the Double Buffer is Disabled**

When write instructions are executed on T00PWM and T01PWM in this order during the timer operation, the set value is immediately stored in T01+00PWM. Subsequently, the match detection is executed using a new set value. If the value set to T01+00PWM is smaller than the up counter value, the PWM1B pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T01+00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM. Therefore, the timing of changing the PWM1B pin may not be an integral multiple of the source clock. Similarly, if T01+00PWM is set during the additional pulse output, the timing of changing the PWM1B pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

When write instructions are executed on T00PWM and T01PWM in this order while the timer is stopped, the set value is immediately stored in T01+00PWM.



T01MOD <TCK1>	Source clock [Hz]			Resolution		8-bit cycle (period × 16)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=8MHz	fs=32.768KHz	fcgck=8MHz	fs=32.768KHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	fcgck/2^{11}	fs/2^4	fs/2^4	256us	488.2us	65.5ms (1048.6ms)	125ms (2000ms)
001	fcgck/2^{10}	fs/2^3	fs/2^3	128us	244.1us	32.8ms (524.3ms)	62.5ms (1000ms)
010	fcgck/2^8	fcgck/2^8	-	32us	-	8.2ms (131.1ms)	-
011	fcgck/2^6	fcgck/2^6	-	8us	-	2.0ms (32.8ms)	-
100	fcgck/2^4	fcgck/2^4	-	2us	-	512us (8192us)	-
101	fcgck/2^2	fcgck/2^2	-	500ns	-	128us (2048us)	-
110	fcgck/2	fcgck/2	-	250ns	-	64us (1024us)	-
111	fcgck	fcgck	fs/2^2	125ns	122.1us	32us (512us)	31.3ms (500ms)

Table 10.15 Resolutions and Cycles in the 12-bit PWM Mode

Example: Operate TC00 and TC01 in the 12-bit PWM mode with the operation clock of fcgck/2 and output a duty pulse nearly equivalent to 14.0625 μs in 51.2μs cycles (fcgck = 10 MHz)

(Actually, output a duty pulse of 225 μs in total in 16 cycles (819.2 μs))

```
SET     (P7FC).1            ; Sets P7FC1 to "1"
SET     (P7CR).1            ; Sets P7CR1 to "1"
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF2       ; Selects the 12-bit PWM mode and fcgck/2
LD      (T00PWM),0x65       ; Sets the timer register (duty pulse)
                            ; (14.0625μs × 16) / (2/fcgck) = 0x465
LD      (T01PWM),0x04       ; Sets the timer register (duty pulse)
LD      (T001CR),0x06       ; Starts TC00 and TC01
```

10.5.4.8 16-bit Programmable Pulse Generate (PPG) Output Mode

In the 16-bit PPG mode, TC00 and TC01 are cascaded to output the pulses that have a resolution of 16 bits and arbitrary pulse width and duty. Two 16-bit registers, T01+00REG and T01+00PWM, are used to output the pulses. This enables output of longer pulses than an 8-bit timer.

(a) Setting

Setting T001CR <TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

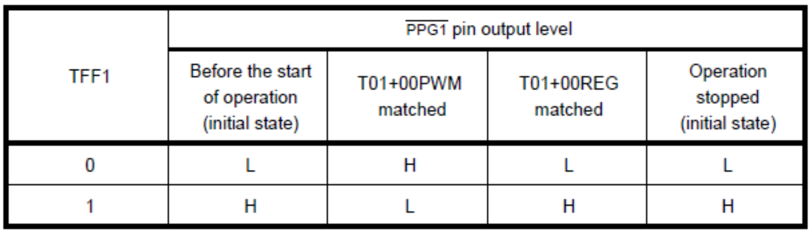
The 16-bit PPG mode is selected by setting T01MOD <TCM1> to "11". To use the internal clock as the source clock, set T01MOD <EIN1> to "0" and select the clock at T01MOD <TCK1>. To use an external clock as the source clock, set T01MOD <EIN0> to "1".

Set T01MOD <DBE1> to "1" to use the double buffer.

Set the count value that corresponds to a cycle as a 16-bit value at the timer registers T01REG and T00REG. Set the count value that corresponds to a duty pulse as a 16-bit value at T01PWM and T00PWM. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG, and the 16-bit value specified by the combined setting of T01PWM and T00PWM is indicated as T01+00PWM). The timer register settings are reflected on the double buffer or T01+00PWM and T01+00REG when a write instruction is executed on T01PWM. Be sure to execute the write instructions on T00REG, T01REG and T00PWM before executing a write instruction on T01PWM. (When data is written to T01PWM, the set values of the four timer registers become effective at the same time.)

Set the initial state of the PPG1B pin at T01MOD <TFF1>. Setting T01MOD <TFF1> to "0" selects the "L" level as the initial state of the PPG1B pin. Setting T01MOD <TFF1> to "1" selects the "H" level as the initial state of the PPG1B pin. If the PPG1B pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD <TFF1> is output to the PPG1B pin. Table 10.16 shows the list of output levels of the PPG1B pin.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.



TFF1	$\overline{\text{PPG1}}$ pin output level			
	Before the start of operation (initial state)	T01+00PWM matched	T01+00REG matched	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

Table 10.16 List of Output Levels of PPG1B Pin

(b) Operation

Setting T001CR <T01RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the up counter value and the value set to T01+00PWM is detected, the output of the PPG1B pin is reversed. When T01MOD <TFF1> is "0", the PPG1B pin changes from the "L" to "H" level. When T01MOD <TFF1> is "1", the PPG1B pin changes from the "H" to "L" level. At this time, an INTTC00 interrupt request is generated.

The up counter continues counting up. When a match between the up counter value and the value set to T01+00REG is detected, the output of the PPG1B pin is reversed again. When T01MOD <TFF1> is "0", the PPG1B pin changes from the "H" to "L" level. When T01MOD <TFF1> is "1", the PPG1B pin changes from the "L" to "H" level. At this time, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000".

When T001CR <T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x0000". The PPG1B pin returns to the level selected at T01MOD <TFF1>.
When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or $fs/2^4$ [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

(c) Double Buffer

The double buffer can be used for T01+00PWM and T01+00REG by setting T01MOD <DBE1>. The double buffer is enabled by setting T01MOD <DBE1> to "0" or disabled by setting T01MOD <DBE1> to "1".

1. When the Double Buffer is Enabled

When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM during the timer operation, the set values are first stored in the double buffer, and T01+00PWM and T01+00REG are not updated immediately. T01+00PWM and T01+00REG compare the previous set values with the up counter value. When a match between the up counter value and the T01+00REG set value is detected, an INTTC01 interrupt request is generated and the double buffer set values are stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM while the timer is stopped, the set values are immediately stored in both the double buffer and T01+00PWM and T01+00REG.

2. **When the Double Buffer is Disabled**

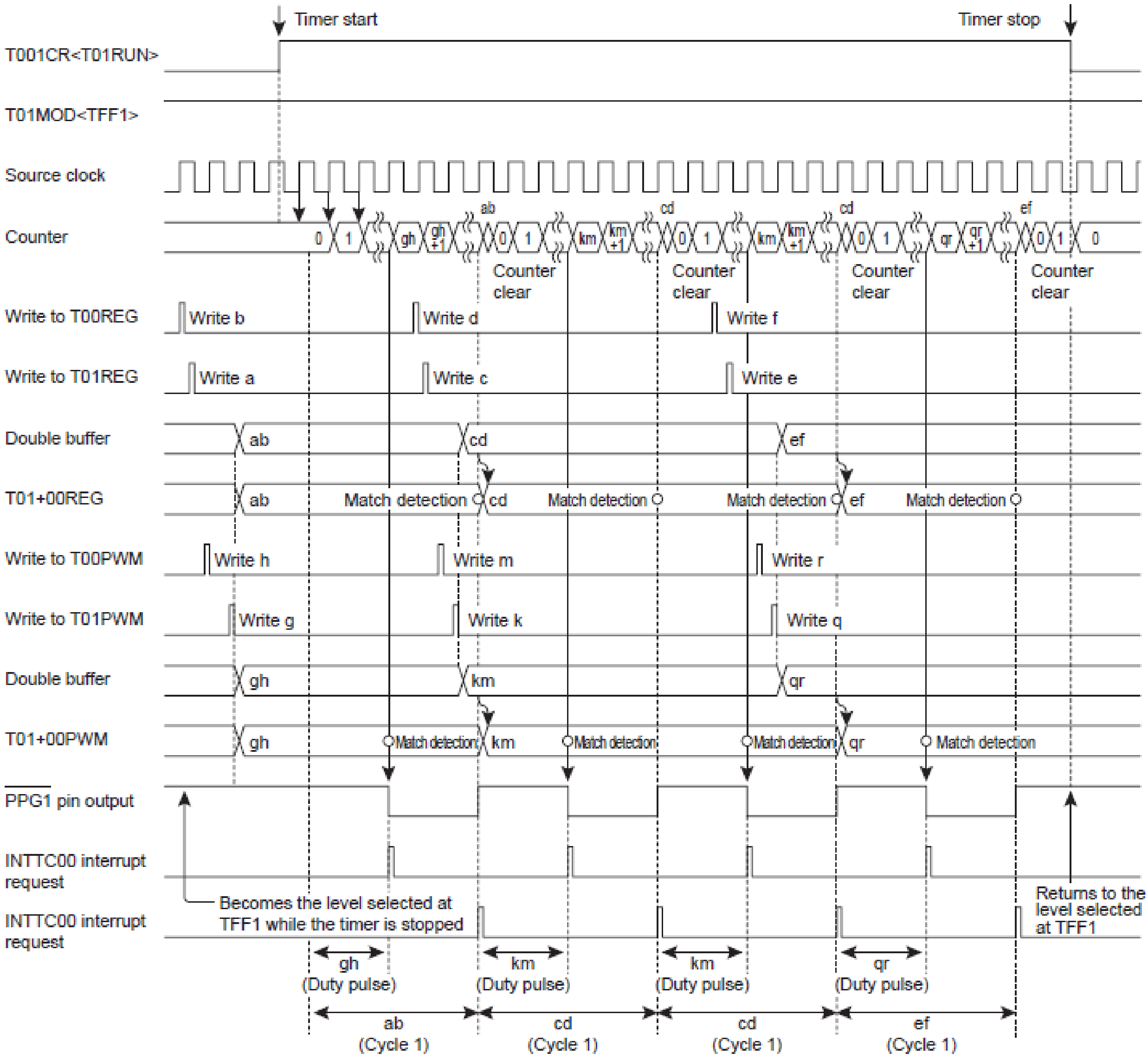
When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM during the timer operation, the set values are immediately stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

If the value set to T01+00PWM or T01+00REG is smaller than the up counter value, the PPG1B pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T01+00PWM or T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM and T01+00REG. Therefore, the timing of changing the PPG1B pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM while the timer is stopped, the set values are immediately stored in T01+00PWM and T01+00REG.

When read instructions are executed on T01+00PWM and T01+00REG, the last value written into T01+00REG is read out, regardless of the T00MOD <DBE1> setting.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 10.25 16-bit PPG Output Mode Timing Chart

Example: Operate TC00 and TC01 in the 16-bit PPG mode with the operation clock of fcgck/2 and output the 68μs duty pulse in 96μs cycles (fcgck = 10 MHz)

```
SET     (P7FC).1            ; Sets P7FC0 to "1"
SET     (P7CR).1            ; Sets P7CR0 to "1"
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF3       ; Selects the 8-bit PPG mode and fcgck/2
LD      (T00REG),0xE0       ; Sets the timer register (cycle)
LD      (T01REG),0x01       ; Sets the timer register (cycle)
                            ; 96μs /(2/fcgck)= 0x01E0
LD      (T00PWM),0x54       ; Sets the timer register (duty pulse)
LD      (T01PWM),0x01       ; Sets the timer register (duty pulse)
                            ; 68us/ (2/fcgck)=0x0154
LD      (T001CR),0x06       ; Starts TC00 and TC01
```

10.6 10-bit Timer Counter (TCC)

MQ6935 contains 1 channels of high-performance 10-bit timer counters (TCC).

The 10-BitTimer/CounterC0 has one trigger input (TCC0) for start/stop/clear/capture of the counter. Timer has two PPG outputs (PPGC01, PPGC02) that can perform synchronous operation or individual operation. Timer has one emergency input(EMG0B) for stop PPG output.

	TCxCR1 (Address)	TCxCR2 (Address)	TCxCR3 (Address)
Timer counter C0	TC0CR1 (0x0E98)	TC0CR2 (0x0E99)	TC0CR3 (0x0E9A)

Table 10.17 SFR Address Assignment

	TCxDRA (Address)	TCxDRB (Address)	TCxDRC (Address)	TCxDRD (Address)	TCxDRE (Address)	TCxCAPA (Address)	TCxCAPB (Address)	Low power consumption register
Timer counter C0	TC0DRA (0x0E9C) (0x0E9B)	TC0DRB (0x0E9E) (0x0E9D)	TC0DRC (0x0EA0) (0x0E9F)	TC0DRD (0x0EA2) (0x0EA1)	TC0DRE (0x0EA4) (0x0EA3)	TC0CAPA (0x0EA6) (0x0EA5)	TC0CAPB (0x0EA8) (0x0EA7)	POFFCR0 <TCC0EN> (0x0F74)

Table 10.18 SFR Address Assignment (cont.)

	Timer Input Pin	PPG Output Pin	EMG Input Pin
Timer counter C0	TCC0 pin	PPGC01B pin PPGC02B pin	EMG0B pin

Table 10.19 Pin Names

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

10.6.1 Configuration

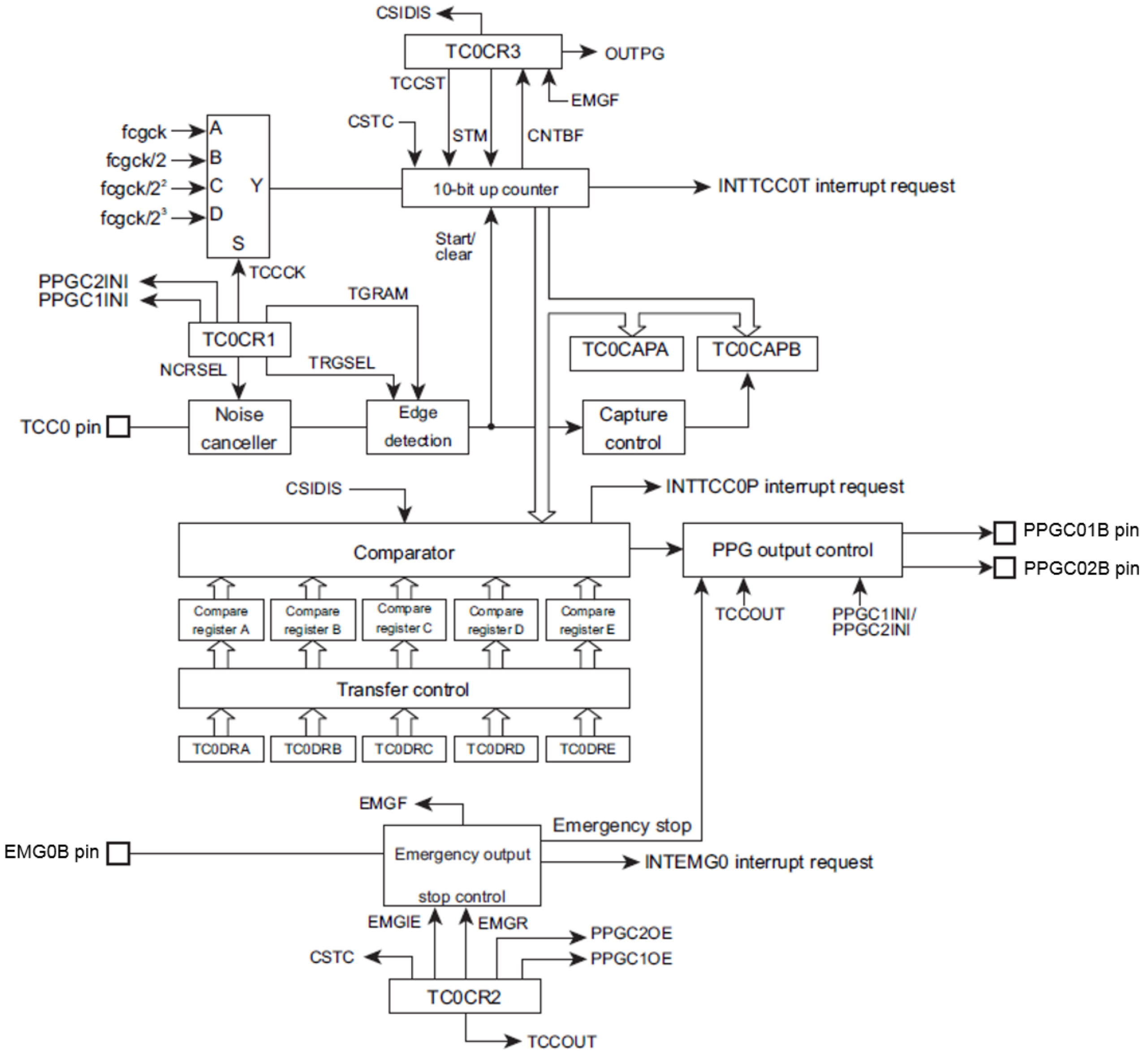


Figure 10.26 10-bit Timer Counter

10.6.2 Control

Timer/counter C0 is controlled by low power consumption registers (POFFCR0),timer/counter control register 1 (TC0CR1), timer/counter control register 2(TC0CR2), timer/counter control register 3 (TC0CR3), 10-bit dead time 1 setup register (TC0DRA), pulse width 1 setup register (TC0DRB), period setup register (TC0DRC), dead time 2 setup register (TC0DRD), pulse width 2 setup register (TC0DRE), and two capture value registers (TC0CAPA and TC0CAPB).

Low Power Consumption Register 0

POFFCR0 (0x0F74)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCA1EN	TCA0EN
Read/Write	R	R	R/W	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,TC03 enable control	0: Disable 1: Enable
TC001EN	TC00,TC01 enable control	0: Disable 1: Enable
TCC0EN	TCC0 enable control	0: Disable 1: Enable
TCA1EN	TCA1 enable control	0: Disable 1: Enable
TCA0EN	TCA0 enable control	0: Disable 1: Enable

Timer/ Counter C0 Control Register 1

TC0CR1 (0x0E98)	7	6	5	4	3	2	1	0
Bit Symbol	TRGAM	TRGSEL	PPGC2INI	PPGC1INI	NCRSEL		TCCCK	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TRGAM	Trigger edge acceptance control	0: Always accept trigger edges. 1: Refer to the setting of TC0CR2<PPGC2OE, PPGC1OE>
TRGSEL	Select a trigger start edge.	0: Start on trigger falling edge. 1: Start on trigger rising edge.
PPGC2INI	Specify the initial value of PPG2 output.	0: Low (Positive logic) 1: High (Negative logic)
PPGC1INI	Specify the initial value of PPG1 output.	0: Low (Positive logic) 1: High (Negative logic)
NCRSEL	Select the duration of noise elimination for TCC0 input (after passing through the flip-flop).	00: Eliminate pulses shorter than 16/fcgck [s] as noise. 01: Eliminate pulses shorter than 8/fcgck [s] as noise. 10: Eliminate pulses shorter than 4/fcgck [s] as noise. 11: Do not eliminate noise. (Note)
TCCCK	Select a source clock	00: fcgck [Hz] 01: fcgck/2 [Hz] 10: fcgck/4 [Hz] 11: fcgck/8[Hz]

Note: Due to the circuit configuration, a pulse shorter than 1/fcgck may be eliminated as noise or accepted as a trigger.

Timer/ Counter C0 Control Register 2

TC0CR2 (0x0E99)	7	6	5	4	3	2	1	0
Bit Symbol	EMGR	EMGIE	PPGC2OE	PPGC1OE	CSTC		TCCOUT	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

EMGR	Cancel the emergency output stop state.	0: - 1: Cancel the emergency output stop state. (Upon canceling the state,this bit is automatically cleared to 0.)
EMGIE	Enable/disable input on the EMG0B pin.	0: Disable input. 1: Enable input.
PPGC2OE	Trigger edge accept control during the PPGC02B output.	**When TC0CR1<TRGAM> is "0"** 0: Always accept trigger edges. 1: Always accept trigger edges. **When TC0CR1<TRGAM> is "1"** 0:Always accept trigger edges. 1:Do not accept trigger edges during active period.
PPGC1OE	Trigger edge accept control during the PPGC01B output.	**When TC0CR1<TRGAM> is "0"** 0: Always accept trigger edges. 1: Always accept trigger edges. **When TC0CR1<TRGAM> is "1"** 0: Always accept trigger edges. 1:Do not accept trigger edges during active period.
CSTC	Select a count start mode.	00: Command start and capture mode 01: Command start and trigger start mode. 10: Trigger start mode 11: Reserved
TCCOUT	Select an output waveform mode.	00: PPGC01/PPGC02 independent output 01: Reserved 10: Output with variable duty ratio 11: Output with 50% duty ratio

Timer/ Counter C0 Control Register 3

TC0CR3 (0x0E9A)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	EMGF	CNTBF	CSIDIS	STM		TCCST
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

EMGF	Emergency output stop flag	0: Operating normally 1: Output stopped in emergency
CNTBF	Counting status flag	0: Counting stopped 1: Counting in progress
CSIDIS	Disable the first interrupt at upon a command start.	0: Allow a periodic interrupt (INTTCC0P) to occur in the first period upon a command start. 1: Do not allow a periodic interrupt (INTTCC0P) to occur in the first period upon a command start.
STM	Select the state when stopped. Select continuous or one-time output.	**When TCCST = "0"** 00: Immediately stop and clear the counter with the output initialized 01: Immediately stop and clear the counter with the output maintained. 10: Stop the counter after completing output in the current period. 11: Reserved **When TCCST = "1"** 00: Continuous output 01: Continuous output 10: One-time output 11: Reserved

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

TCCST	Start/stop the timer.	0: Stop 1: Start

Note 1: The TC0CR1 and TC0CR2 registers should not be rewritten after a timer start (when TCCST is "1").
Note 2: Before attempting to modify the TC0CR1 or TC0CR2, clear TCCST and then check that CNTBF = 0 to determine that the timer is stopped.
Note 3: The TCCST bit only causes the timer to start or stop; it does not indicate the current operating state of the counter. Its value does not change automatically when counting starts or stops
Note 4: In command start and capture mode or command start and trigger start mode, writing 1 to TCCST causes the timer to restart immediately. It means that rewriting any bit other than TCCST in the TC0CR3 after a command start causes the rewriting of TCCST, resulting in the timer being restarted (PPG output is started from the initial state). When TCCST is set to 1, rewriting the TC0CR3 (Using a bit manipulation or LD instruction) clears the counter and restarts the timer.
Note 5: TC0CR2<EMGR> is always read as 0 even after 1 is written.
Note 6: Data registers are not updated by merely modifying the output mode with TC0CR2<TCCOUT>. After modifying the output mode, reconfigure data registers TC0DRA to TC0DRE. Ensure that the data registers are written in an appropriate order because they are not enabled until the upper byte of the TC0DRC is written.
Note 7: When a read instruction is executed on TC0CR3, bits 7 and 6 are read as "0".

Dead Time1 Setup Register AH

TC0DRAH (0x0E9C)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRAH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Dead Time1 Setup Register AL

TC0DRAL (0x0E9B)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRAL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 1 Setup Register BH

TC0DRBH (0x0E9E)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRBH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 1 Setup Register BL

TC0DRBL (0x0E9D)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRBL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Period Setup Register CH

TC0DRCH (0x0EA0)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRCH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Period Setup Register CL

TC0DRCL (0x0E9F)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRCL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Dead Time2 Setup Register DH

TC0DRDH (0x0EA2)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRDH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Dead Time2 Setup Register DL

TC0DRDL (0x0EA1)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRDL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 2 Setup Register EH

TC0DREH (0x0EA4)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DREH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 2 Setup Register EL

TC0DREL (0x0EA3)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DREL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1: Data registers TC0DRA to TC0DRE have double-stage configuration, consisting of a data register that stores data written by

an instruction and a compare register to be compared with the counter.

Note 2: When writing data to data registers TC0DRA to TC0DRE, first write the lower byte and then the upper byte.

Note 3: Unused bits (Bits 10 to 15) in the upper bytes of data registers TC0DRA to TC0DRE are not assigned specific register functions. These bits are always read as 0 even when a 1 is written.

Note 4: Values read from data registers TC0DRA to TC0DRE may differ from the actual PPG output waveforms due to their double-stage configuration.

Note 5: Data registers are not updated by merely modifying the output mode with TC0CR2<TCCOUT>. After modifying the output mode, reconfigure data registers TC0DRA to TC0DRE. Ensure that the data registers are written in an appropriate order because they are not enabled until the upper byte of the TC0DRC is written.

Rising-edge Capture Value Register AH

TC0CAPAH (0x0EA6)	15	14	13	12	11	10	9	8
Bit Symbol	TC0CAPAH							
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*

Rising-edge Capture Value Register AL

TC0CAPAL (0x0EA5)	7	6	5	4	3	2	1	0
Bit Symbol	TC0CAPAL							
Read/Write	R	R	R	R	R	R	R	R/W
After reset	*	*	*	*	*	*	*	*

Rising-edge Capture Value Register BH

TC0CAPBH (0x0EA8)	15	14	13	12	11	10	9	8
Bit Symbol	TC0CAPBH							
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*

Rising-edge Capture Value Register BL

TC0CAPBL (0x0EA7)	7	6	5	4	3	2	1	0
Bit Symbol	TC0CAPBL							
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*

Note 1: Capture registers (TC0CAPA and TC0CAPB) must be read in the following order: Lower byte of the TC0CAPA, upper byte of the TC0CAPA, lower byte of the TC0CAPB, upper byte of the TC0CAPB.
Note 2: The next captured data is not updated by reading the TC0CAPA only. The TC0CAPB must also be read.
Note 3: It is possible to read the TC0CAPB only. Read the lower byte first.
Note 4: If a capture edge is not detected within a period, the previous capture value is maintained in the next period.
Note 5: If more than one capture edge is detected within a period, the capture value for the edge detected last is valid in the next period.
Note 6: When a read instruction is executed on TC0CAPA and TC0CAPB, bits 15 to 10 are read as "0".

10.6.3 Low Power Consumption Function

Timer counter C0 has the low power consumption register (POFFCR0) that saves power consumption when the timer is not used.

Setting POFFCR0<TCC0EN> to "0" disables the basic clock supply to timer counter C0 to save power. Note that this makes the timer unusable. Setting POFFCR0<TCC0EN> to "1" enables the basic clock supply to timer counter C0 and allows the timer to operate.

After reset, POFFCR0<TCC0EN> is initialized to "0", and this makes the timer unusable. When using the timer for the first time, be sure to set POFFCR0<TCC0EN> to "1" in the initial setting of the program (before the timer control register is operated).
Do not change POFFCR0<TCC0EN> to "0" during the timer operation. Otherwise timer counter C0 may operate unexpectedly.

10.6.4 Configuring Control and Data Registers

Configure control and data registers in the following order:

1. Configure mode settings: TC0CR1, TC0CR2
2. Configure data registers (Dead time, pulse width):TC0DRA, TC0DRB, TC0DRD, TC0DRE (only those required for selected mode)
3. Configure data registers (Period): TC0DRC
4. Configure timer start/stop: TC0CR3

・ Data registers have double-stage configuration, consisting of a data register that stores data written by an instruction and a compare register to be compared with the counter.
・ Data stored in a data register is processed according to the output mode specified in the TC0CR2<TCCOUT>,transferred to the compare register, and then used for comparison with the up counter.
・ Data registers required for the specified output mode are used for data register processing and transfer to the compare register. Ensure that the output mode is specified in the TC0CR2<TCCOUT> before configuring data registers.
・ Writing data to the upper byte of the TC0DRC causes a data transfer request to be issued for data in data registers TC0DRA to TC0DRE. If a counter match or clear occurs while that request is valid, the data is transferred to the compare register and becomes valid for comparison.
・ If a data register is written more than once within a period, the data in the data register that was set when the upper byte of the TC0DRC was written is valid as data for the next period. The data in the data register written last in the first period will be valid for the period that follows the next period.

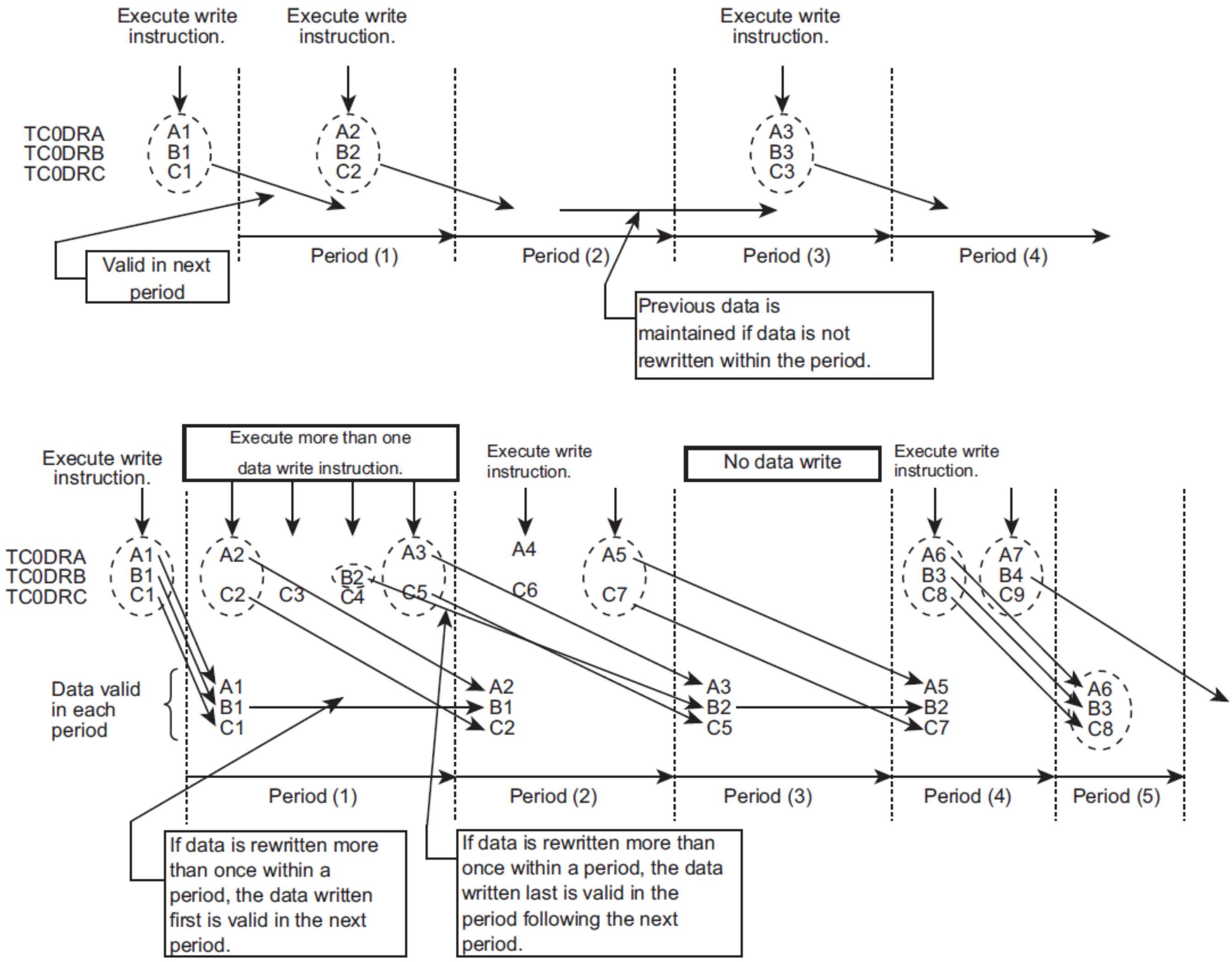

Execute write instruction.
Execute write instruction.
Execute write instruction.
TC0DRA
TC0DRB
TC0DRC
A1 B1 C1
A2 B2 C2
A3 B3 C3
Valid in next period
Period (1)
Period (2)
Period (3)
Period (4)
Previous data is maintained if data is not rewritten within the period.
Execute write instruction.
Execute more than one data write instruction.
Execute write instruction.
No data write
Execute write instruction.
TC0DRA
TC0DRB
TC0DRC
A1 B1 C1
A2 C2 C3 B2 C4 A3 C5
A4 C6
A5 C7
A6 B3 C8
A7 B4 C9
Data valid in each period
A1 B1 C1
A2 B1 C2
A3 B2 C5
A5 B2 C7
A6 B3 C8
Period (1)
Period (2)
Period (3)
Period (4)
Period (5)
If data is rewritten more than once within a period, the data written first is valid in the next period.
If data is rewritten more than once within a period, the data written last is valid in the period following the next period.

Figure 10.27 Example Configuration of control/data registers (1)

10.6.5 Features

10.6.5.1 Programmable pulse generator output (PPG output)

The PPGC01B and PPGC02B pins provide PPG outputs. The output waveform mode for PPG outputs is specified with TC0CR2<TCCOUT> and their waveforms are controlled by comparing the contents of the 10-bit up counter with the data set in data registers (TC0DRA to TC0DRE). Three output waveform modes are available: 50% duty mode, variable duty mode, and PPGC01B/PPGC02B independent mode.

50 % duty mode

Description

With a period specified in the TC0DRC, the PPGC01B and PPGC02B pins provide waveforms having a pulse width (Active duration) that equals a half the period.

The PPGC01B output is active at the beginning of a period and becomes inactive at half the period. The PPGC02B output is inactive at the beginning of a period, becomes active at half the period, and remains active until the end of the period.

If a dead time is specified in the TC0DRA, the pulse width (Active duration) is shortened by the dead time.

Register Settings

TC0CR2<TCCOUT> = "11", TC0DRA = "dead time", TC0DRC = "period"

Valid range for data register values

・ Period:

0x002 ≤ TC0DRC ≤ 0x400

(Writing 0x400 to TC0DRC results in 0x000 being read from it)

When the value set in the TC0DRC is an odd number, the PPGC02 pulse width is one count longer than the PPGC01 pulse width.

・ Dead time TC0DRA:

0x000 ≤ TC0DRA < (TC0DRC ÷ 2)

To specify no dead time, set the TC0DRA to 0x000.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

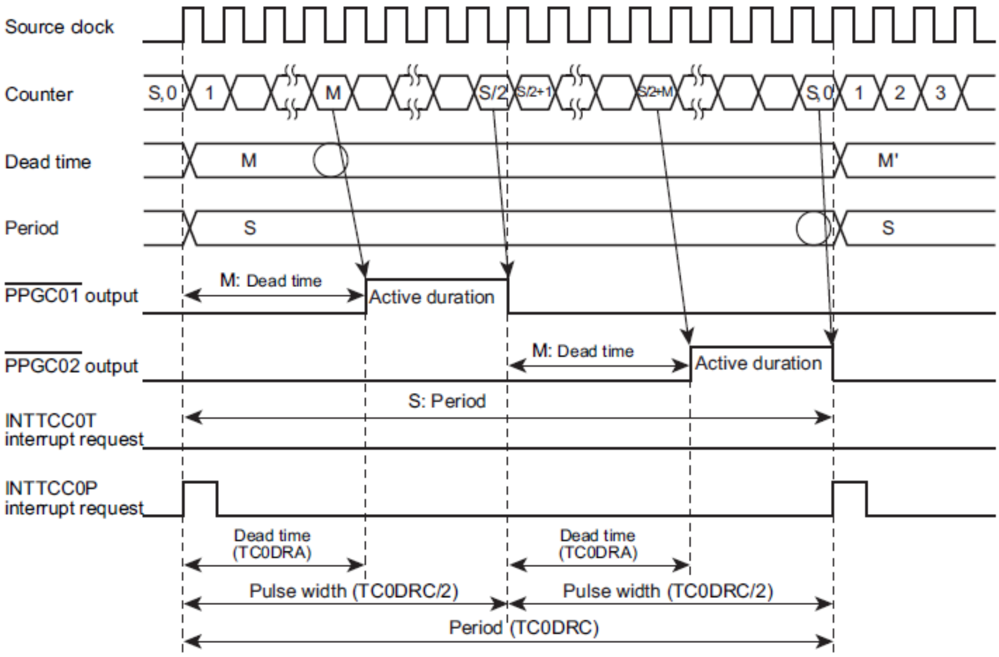


Figure 10.28 Example operation in 50% duty mode: command and capture start, positive logic, continuous output

Variable duty mode

Description

With a period specified in the TC0DRC and a pulse width in the TC0DRB, the PPGC01B pin provides a waveform having the specified pulse width while the PPGC02B pin provides a waveform having a pulse width that equals (TC0DRC . TC0DRB).

The PPGC01B output is active at the beginning of a period, remains active during the pulse width specified in the TC0DRB, after which it is inactive until the end of the period. The PPGC02B output is inactive at the beginning of a period, remains inactive during the pulse width specified in the TC0DRB,after which it is active until the end of the period, that is, during the pulse width of (TC0DRC .TC0DRB).

If a dead time is specified in the TC0DRA, the pulse width (Active duration) is shortened by the dead time.

Register Settings

TC0CR2<TCCOUT> = “10”
TC0DRA = “dead time”, TC0DRB = “pulse width”, TC0DRC = “period”

Valid range for data register values

- Period:

$0x002 \leq TC0DRB + TC0DRA < TC0DRC \leq 0x400$

(Writing 0x400 to TC0DRC results in 0x000 being read from it.)

- Pulse width:

$0x001 \leq TC0DRB < TC0DRC$

- Dead time:

$0x000 \leq TC0DRA < TC0DRB$

$0x000 \leq TC0DRA < (TC0DRC - TC0DRB)$

(To specify no dead time, set the TC0DRA to 0x000.)

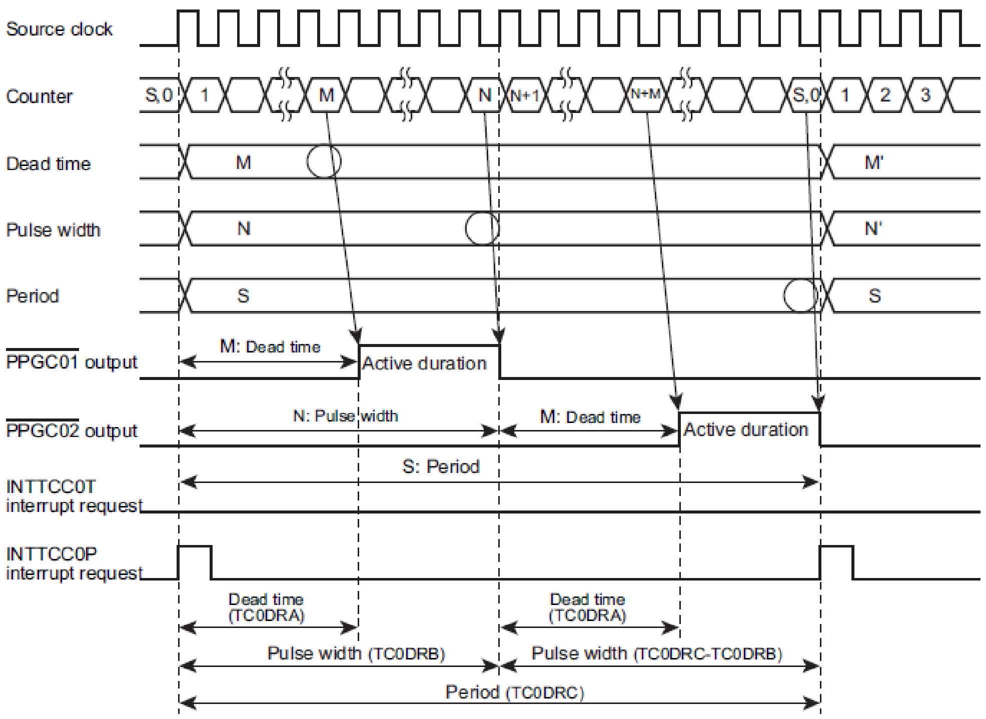


Figure 10.29 Example Operation in Variable Duty Mode: Command and Capture Start, Positive Logic, Continuous Output

PPGC01B/PPGC02B independent mode

Description

For the PPGC01B output, specify the dead time in the TC0DRA and pulse width in the TC0DRB. For the PPGC02B output, specify the dead time in the TC0DRD and pulse width in the TC0DRE. With a common period specified in the TC0DRC, the PPGC01B and PPGC02B pins provide waveforms having the specified pulse widths.

The PPGC01B output is active at the beginning of a period, remains active during the pulse width specified in the TC0DRB, after which it is inactive until the end of the period. The PPGC02B output is active at the beginning of a period, remains active during the pulse width specified in the TC0DRE, after which it is inactive until the end of the period.

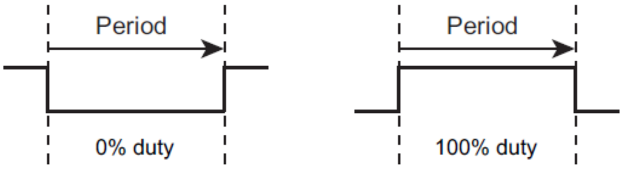
If a dead time is specified in the TC0DRA for the PPGC01B output or in the TC0DRD for the PPGC02B output, the pulse width (Active duration) is shortened by the dead time.

Register Settings
TC0CR2<TCCOUT> = "00", TC0DRC = "period",
TC0DRA = "PPGC01 dead time", TC0DRB = "PPGC01 pulse width",
TC0DRD = "PPGC02 dead time", TC0DRE = "PPGC02 pulse width"

Valid range for data register values
・ Period:
0x002 ≤ TC0DRC ≤ 0x400
(Writing 0x400 to TC0DRC results in 0x000 being read from it.)
・ Pulse width:
0x001 ≤ TC0DRB ≤ 0x400
(Writing 0x400 to TC0DRB results in 0x000 being read from it.)
0x001 ≤ TC0DRE ≤ 0x400
(Writing 0x400 to TC0DRE results in 0x000 being read from it.)
・ Dead time:
0x000 ≤ TC0DRA ≤ 0x3FF, where TC0DRA < TC0DRB ≤ TC0DRC
0x000 ≤ TC0DRD ≤ 0x3FF, where TC0DRD < TC0DRE ≤ TC0DRC
(To specify no dead time, write 0x000)

1. Settings for a duty ratio of 0%
0x002 ≤ TC0DRC ≤ TC0DRA ≤ 0x3FF (PPGC01B output)
0x002 ≤ TC0DRC ≤ TC0DRD ≤ 0x3FF (PPGC02B output)

2. Settings for a duty ratio greater than 0%, up to 100%
0x000 ≤ TC0DRA < TC0DRB ≤ TC0DRC ≤ 0x400 (PPGC01B output)
0x000 ≤ TC0DRD < TC0DRE ≤ TC0DRC ≤ 0x400 (PPGC02B output)



iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

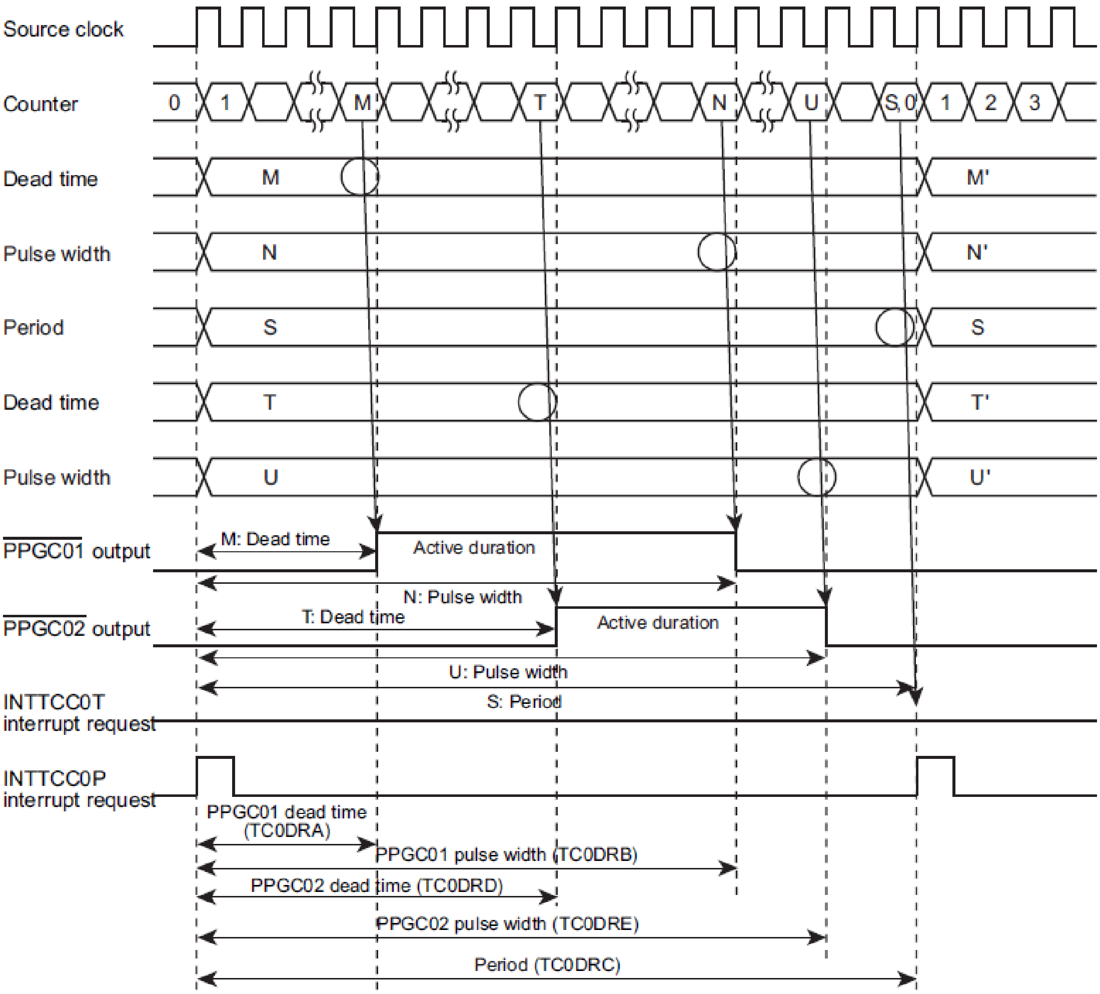


Figure 10.30 Example operation in PPGC01/PPGC02 independent mode: command and capture start, positive logic, continuous output

10.6.5.2 Starting a count

Command start and capture mode (TC0CR2<CSTC>="00"

Description

Writing a 1 to TC0CR3<TCCST> causes the current count to be cleared and the counter to start counting. Once the count has reached a specified period, the counter is cleared. The counter subsequently restarts counting if TC0CR3<STM> specifies continuous mode; it stops counting if TC0CR3<STM> specifies one-time mode.

Writing a 1 to TC0CR3<TCCST> before the count reaches a period causes the counter to be cleared, after which it operates as specified with TC0CR3<STM>.

The count values at the rising and falling edges on the TCC0 pin can be stored in capture registers.

Register Settings

- TC0CR2<CSTC> = “00” Command start and capture mode
- TC0CR3<STM> Continuous/one-time output
- TC0CR3<TCCST> = “1” Starts counting

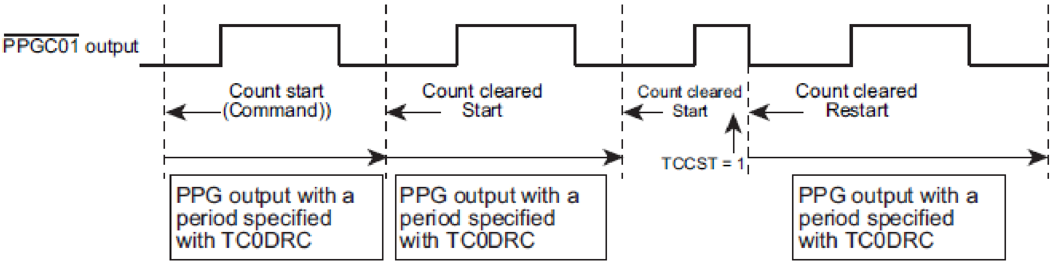

PPGC01 output
Count start (Command))
Count cleared Start
Count cleared Start
Count cleared Restart
TCCST = 1
PPG output with a period specified with TC0DRC
PPG output with a period specified with TC0DRC
PPG output with a period specified with TC0DRC

Figure 10.31 Example operation in command start and capture mode

Command start and trigger start mode (TC0CR2<CSTC>=”01”

Description

Writing a 1 to TC0CR3<TCCST> causes the current count to be cleared and the counter to start counting. The operation is the same as that in command start and capture mode if there is no trigger input on the TCC0 pin. If an edge specified with the start edge selection field (TC0CR1<TRGSEL>) appears on the TCC0 pin, however, the timer starts counting. The counter is cleared and stopped while the TCC0 pin is driven to the specified clear/stop level. If the TCC0 pin is at the clear/stop level when a count start command is issued (1 is written to TC0CR3<TCCST>), counting does not start (INTTCC0P does not occur) until a trigger start edge appears, causing INTTCC0T to occur (A trigger input takes precedence over a command start).

Register Settings

- TC0CR2<CSTC> = “01” Command start and trigger start mode
- TC0CR1<TRGSEL> = Trigger selection
- TC0CR3<STM> Continuous/one-time output
- TC0CR3<TCCST> = “1” Starts counting

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

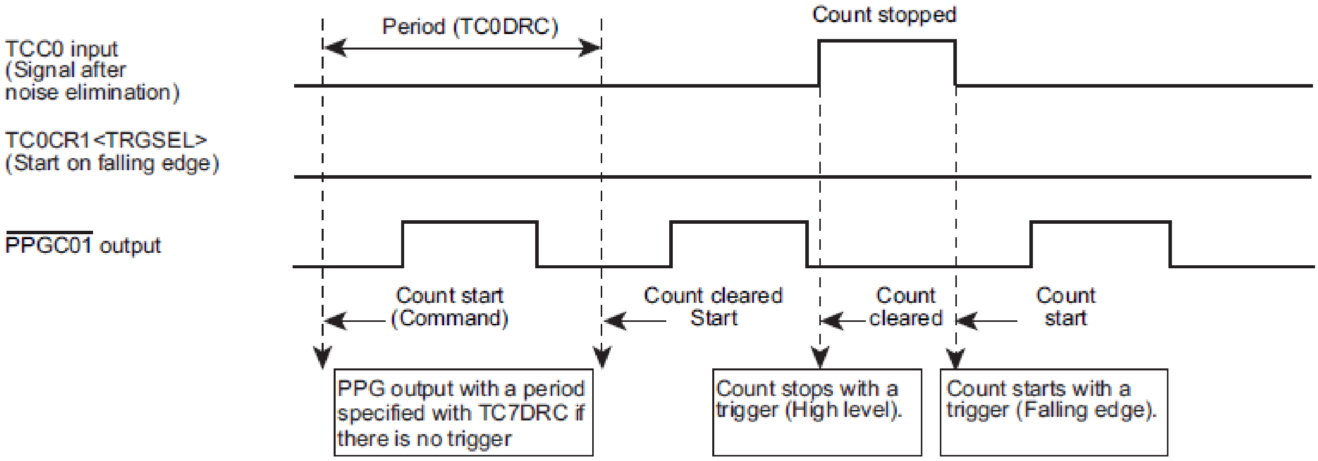


Figure 10.32 Example operation in command start and trigger start mode

Trigger start mode (TC0CR2<CSTC>="10"

Description

If an edge specified with the start edge selection field (TC0CR1<TRGSEL>) appears on the TCC0 pin, the timer starts counting. The counter is cleared and stopped while the TCC0 pin is driven to the specified clear/stop level.

In trigger start mode, writing a 1 to TC0CR3<TCCST> is ignored and does not initialize the PPG output.

Register Settings

- TC0CR2<CSTC> = “10” Trigger start mode
- TC0CR1<TRGSEL> = Trigger selection
- TC0CR3<STM> Continuous/one-time output
- TC0CR3<TCCST> = “1” Starts waiting for a trigger on the TCC0 pin

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

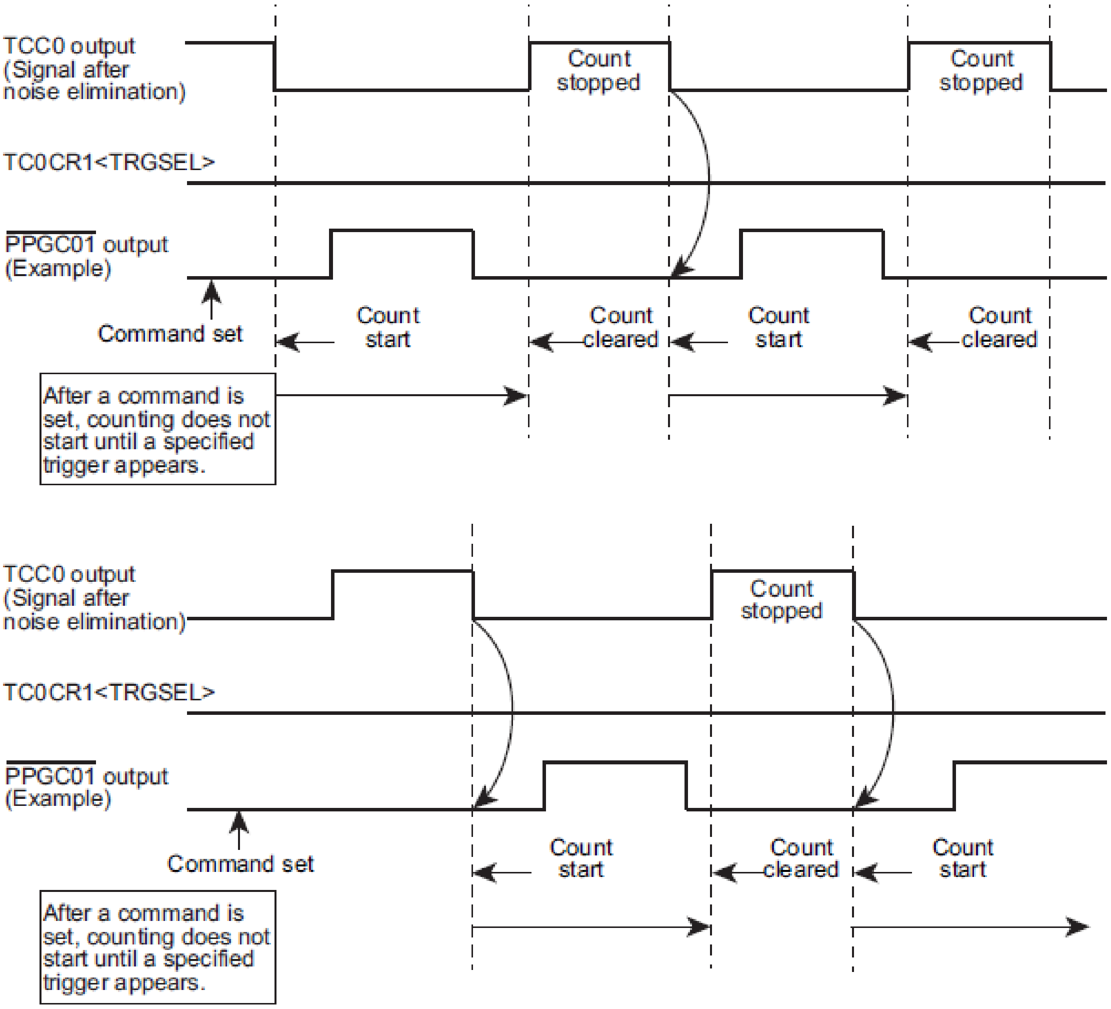


Figure 10.33 Example operation in trigger start mode

10.6.5.3 Trigger capture

Description

When counting starts in command start and capture mode, the count values at the rising and falling edges of the TCC0 pin input are captured and stored in capture registers TC0CAPA and TC0CAPB,respectively.

The captured data is first stored in the capture buffer. At the end of the period, the data is transferred from the capture buffer to the capture register. If a trigger input does not appear within a period, the data captured in the previous period remains in the capture buffer and is transferred to the capture register at the end of the period. If more than one trigger edge is detected within a period, the data captured last is written to the capture register.

Captured data must be read in the following order: Lower byte of capture register A (TC0CAPAL), upper byte of capture register A (TC0CAPAH), lower byte of capture register B (TC0CAPBL), and upper byte of capture register B (TC0CAPBH). Note that reading only the rising-edge captured data (TC0CAPA) does not update the next captured data. The falling-edge captured data (TC0CAPB) must also be read.

An attempt to read a captured value from a register other than the upper byte of the TC0CAPB causes the capture registers to enter protected state, in which captured data cannot be updated. Reading a value from the upper byte of the TC0CAPB cancels that state, re-enabling the updating of captured data (The TC0CAPA and TC0CAPB are read as a single set of operation). Note that the protected state may be still effective immediately after the counter starts. Ensure that a dummy read of capture registers is performed in the first period to cancel the protected state.

The capture feature of the TCC0 assumes that a capture trigger (Rising or falling edge) appears within a period. Captured data is updated (An edge is detected) only when the timer is operating (TC0CR3<TCCST> = 1). If a timer stop command (TC0CR3<TCCST> = 0) is written within a period, captured data will be undefined. Captured data is not updated after a one-time stop command is written. In one-time stop mode, no trigger is accepted after a STOP command is given.

Register Settings

- TC0CR2<CSTC> = “00” Command start and capture mode
- TC0CR3<STM> Continuous/one-time output
- TC0CR3<TCCST> = “1” Starts counting

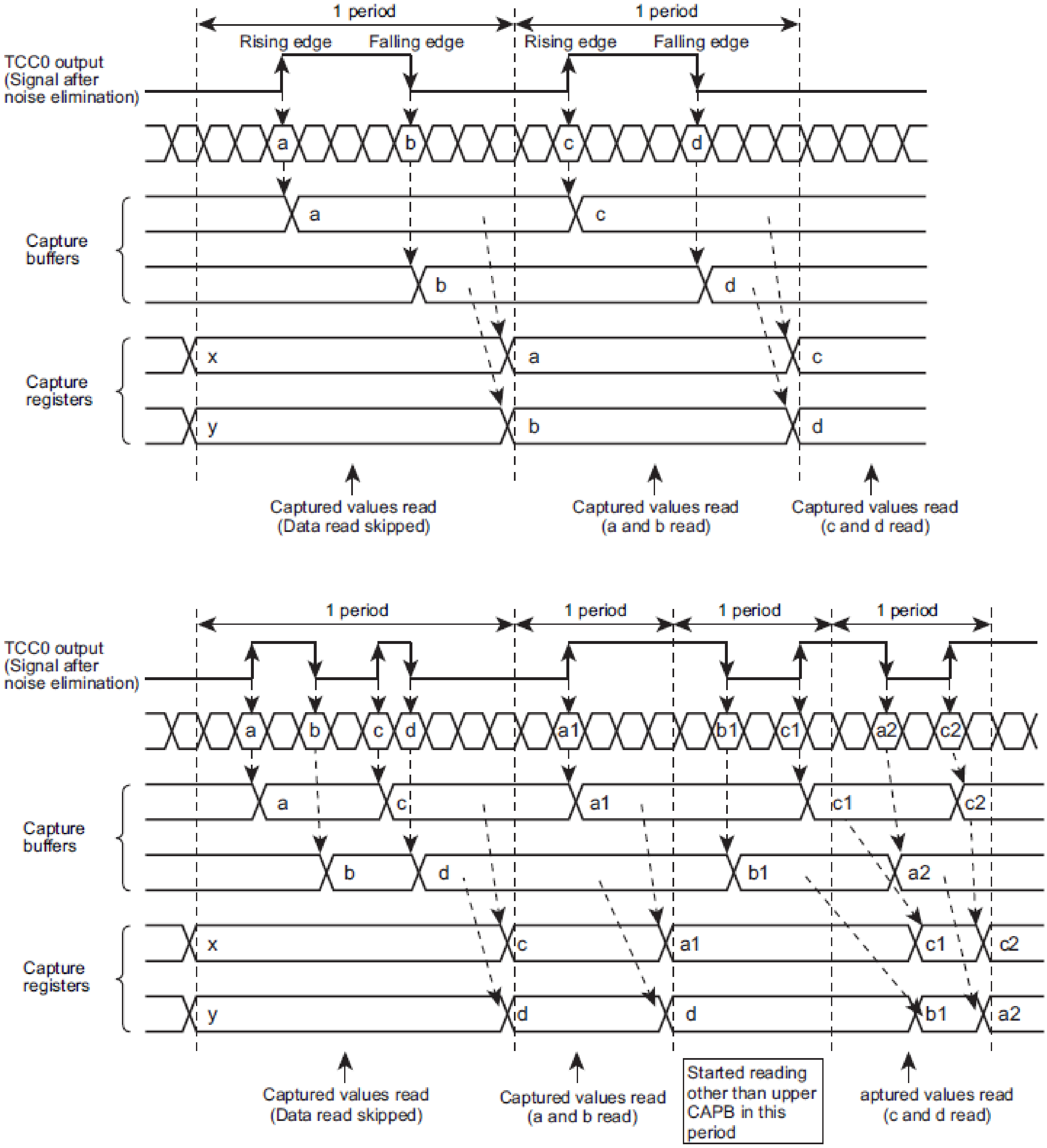


Figure 10.34 Example operation in trigger capture mode

10.6.5.4 Trigger start/stop acceptance mode

Selecting an input signal logic for the TCC0 pin (Trigger input)

The logic for an input trigger signal on the TCC0 pin can be specified using TC0CR1<TRGSEL> .

・ TC0CR1<TRGSEL> = “0”: Counting starts on the falling edge. The counter is cleared and stopped while the TCC0 pin is high.

・ TC0CR1<TRGSEL> = “1”: Counting starts on the rising edge. The counter is cleared and stopped while the TCC0 pin is low.

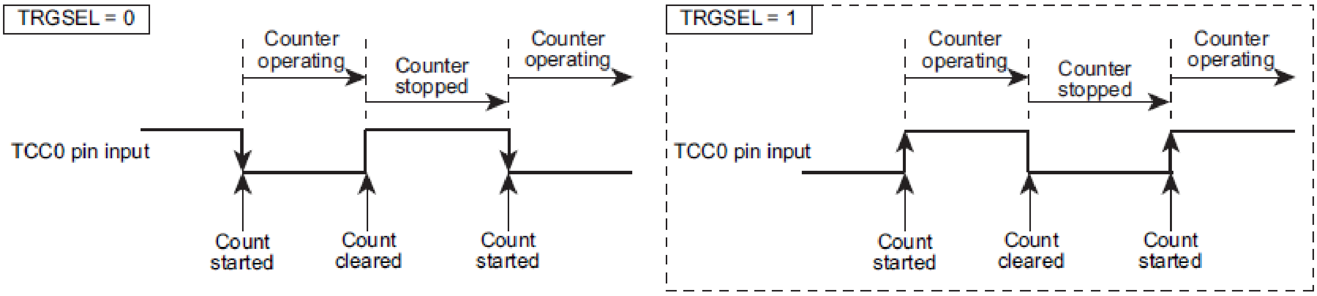


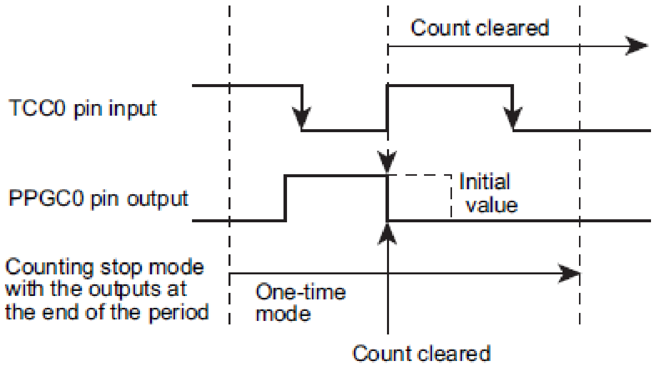
Figure 10.35 Trigger input signal

When TC0CR1<TRGSEL> is set to 0 to select a falling-edge trigger, a falling edge detected on the TCC0 pin causes the counter to start counting and a high level on the TCC0 pin causes the counter to be cleared and the PPG output to be initialized. The counter is stopped while the TCC0 pin input is high.

When TC0CR1<TRGSEL> is set to 1 to select a rising-edge trigger, a rising edge detected on the TCC0 pin causes the counter to start counting and a low level on the TCC0 pin causes the counter to be cleared and the PPG output to be initialized. The counter is stopped while the TCC0 pin input is low.

In one-time stop mode, the counter accepts a stop trigger but does not accept a start trigger (when a stop trigger is accepted within a period, the output is immediately initialized and the counter is stopped).

All triggers (Start and stop) are ignored when the timer is stopped (TC0CR3<TCCST> = 0).



Specifying whether triggers are always accepted or ignored when PPG outputs are active

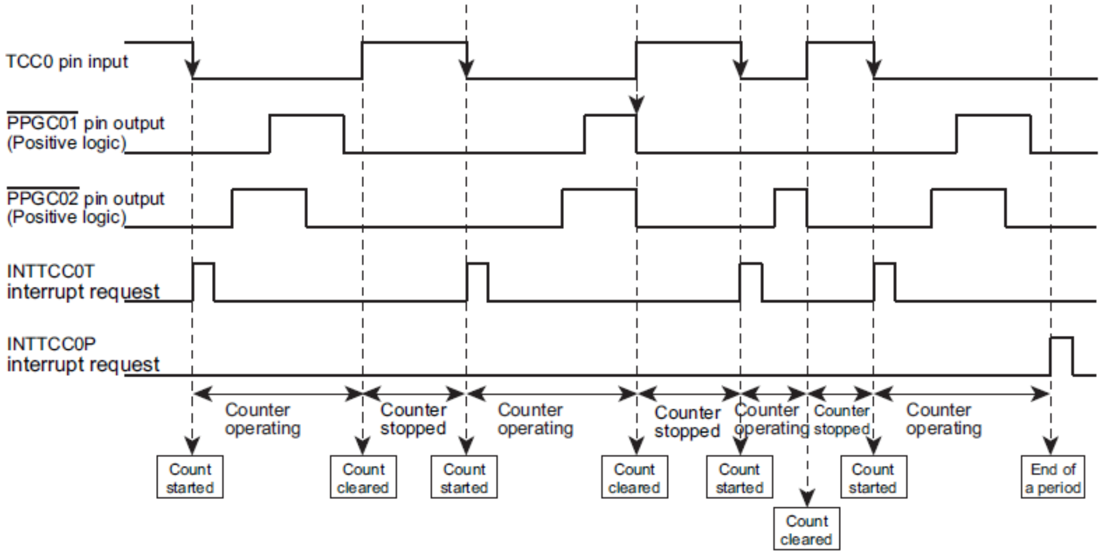
The TC0CR1<TRGAM> specifies whether triggers from the TCC0 pin are always accepted or ignored when the PPG output is active.

・ TC0CR1<TRGAM> = “0”

Triggers from the TCC0 pin are always accepted regardless of whether PPGC01B and PPGC02B outputs are active or inactive. A trigger starts or clears/stops the timer and deactivates PPGC01B and PPGC02B outputs.

・ TC0CR1<TRGAM> = “1”

After TC0CR2<PPGCxOE> setting to "1" ,Triggers from the TCC0 pin are accepted only when PPGC01B and PPGC02B outputs are inactive. A trigger starts or clears/stops the timer. Triggers are ignored when PPGC01B and PPGC02B outputs are active. Triggers are always accept the output of PPGC01B and PPGC02B when TC0CR2<PPGCxOE> is set in "0" (x = 1, 2).



iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Figure 10.36 Start and clear/stop triggers on the TCC0 pin: falling-edge Trigger (Counting stopped at high level), trigger always accepted.

Ignoring triggers when PPG outputs are active

Setting TRGAM to 1 specifies that triggers are ignored when PPG outputs are active; trigger edges detected when PPGC01 and PPGC02 outputs are inactive are accepted and cause the counter to be cleared and stopped. If a trigger is detected when PPGC01 and PPGC02 outputs are active, the counter does not stop immediately but continues counting until the outputs become inactive. If the trigger signal level is a stop level when the outputs become inactive, the counter is cleared/stopped and waits for a next start trigger. If output is enabled for both PPGC01 and PPGC02, triggers are accepted only when both PPGC01 and PPGC02 outputs are inactive.

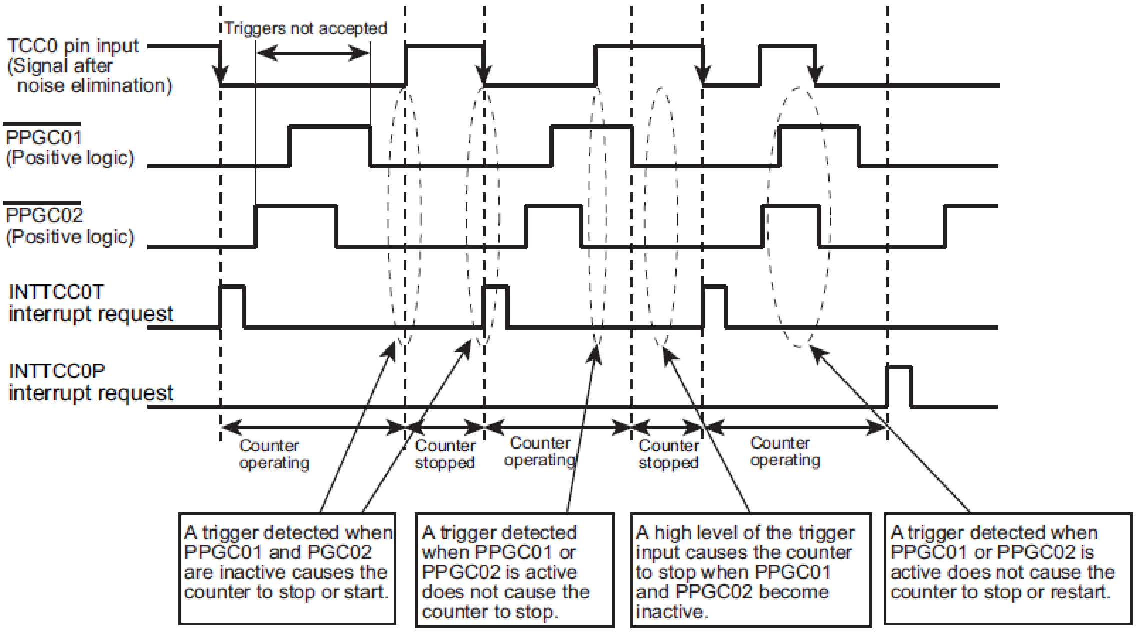

Triggers not accepted
TCC0 pin input (Signal after noise elimination)
PPGC01 (Positive logic)
PPGC02 (Positive logic)
INTTCC0T interrupt request
INTTCC0P interrupt request
Counter operating
Counter stopped
Counter operating
Counter stopped
Counter operating
A trigger detected when PPGC01 and PGC02 are inactive causes the counter to stop or start.
A trigger detected when PPGC01 or PPGC02 is active does not cause the counter to stop.
A high level of the trigger input causes the counter to stop when PPGC01 and PPGC02 become inactive.
A trigger detected when PPGC01 or PPGC02 is active does not cause the counter to stop or restart.

Figure 10.37 Start triggers on the TCC0 pin: falling-edge trigger (counting stopped at high level), triggers ignored when PPG outputs are active.

10.6.5.5 Configuring how the timer stops

Setting TC0CR3<TCCST> to 0 causes the timer to stop with the specified output state according to the setting of TC0CR3<STM>.

Counting stopped with the outputs initialized

When TC0CR3<STM> is set to 00, the counter stops immediately with the PPGC01 and PPGC02 outputs initialized to the values specified with PPGC1INI and PPGC2INI.

Counting stopped with the outputs maintained

When TC0CR3<STM> is set to 01, the counter stops immediately with the current PPGC01 and PPGC02 output states maintained

To restart the counter from the maintained state (TC0CR3<STM> = 01), set TC0CR3<TCCST> to 1.The counter is restarted with the initial output values, specified with PPGC1INI and PPGC2INI.

Counting stopped with the outputs initialized at the end of the period

When TC0CR3<STM> is set to 10, the counter continues counting until the end of the current period and then stops. If a stop trigger is detected before the end of the period, however, the counter stops immediately. TC0CR1 and TC0CR2 must not be rewritten before the counter stops completely.

The TC0CR3<CNTBF> can be read to determine whether the counter has stopped.

10.6.5.6 One-time/ continuous output mode

One-time output mode

Starting the timer (TC0CR3<TCCST> = 1) with TC0CR3<STM> set to 10 specifies one-time output mode. In this mode, the timer stops counting at the end of a period.
For a trigger start, the counter is stopped until a trigger is detected. A specified trigger restarts counting and the counter stops at the end of the period or when a stop trigger is detected, after which it waits for a trigger again.

For a command start, the counter is stopped until TC0CR3<TCCST> is reset to 1.
TC0CR1 and TC0CR2 must not be rewritten before the counter stops completely.
The TC0CR3<CNTBF> can be read to determine whether the counter has stopped.
TC0CR3<TCCST> remains set to 1 after the counter is stopped.
When TC0CR3<TCCST> is set to 1, setting TC0CR3<STM> to 10 clears the counter, which then restarts counting from the beginning in one-time output mode.

Continuous output mode

Starting the timer (TC0CR3<TCCST> = 1) with TC0CR3<STM> set to 00 or 01 specifies continuous output mode. In this mode, the timer outputs specified waveforms continuously.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

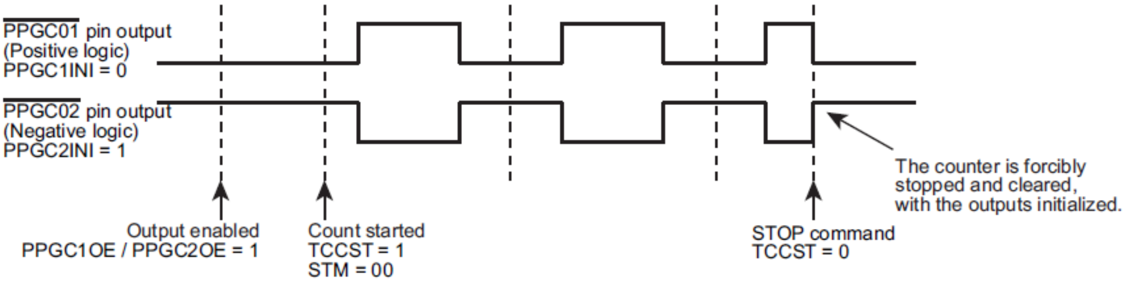


Figure 10.38 Immediately stopping and clearing the counter with the outputs initialized (TC0CR3<STM>="00")

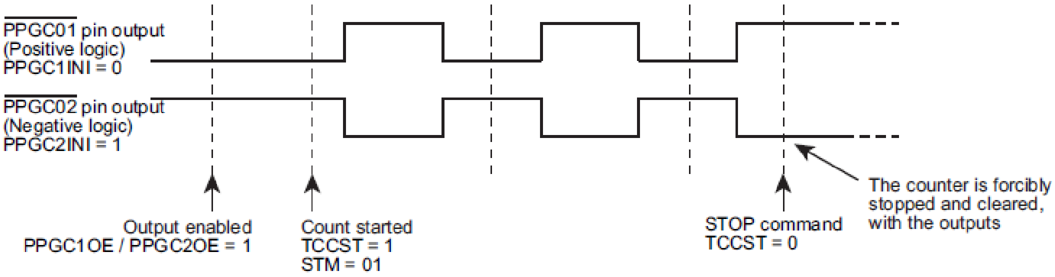


Figure 10.39 Immediately stopping and clearing the counter with the outputs maintained (TC0CR3<STM>="01")

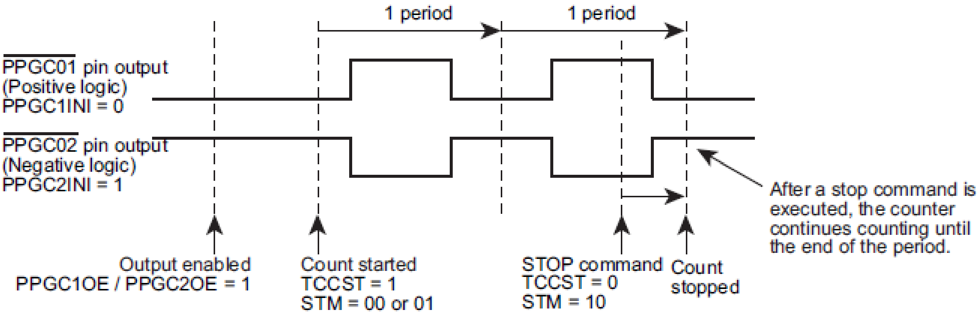


Figure 10.40 Stopping the counter at the end of the period (TC0CR3<STM>="10")

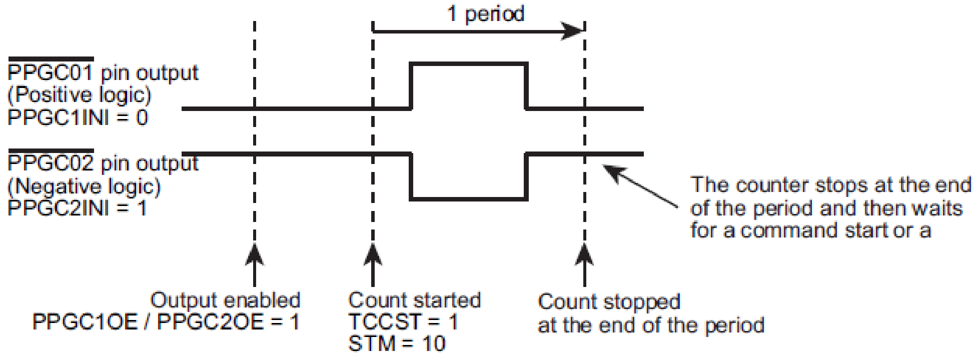


Figure 10.41 Stopping the counter at the end of the period (TC0CR3<STM>="10"), TC0CR3<TCCST>="1" one-time output mode

10.6.5.7 PPG output control (initial value/output logic, enabling /disabling output)

Specifying initial values and output logic for PPG outputs

The TC0CR1<PPGC1INI and PPGC2INI> specify the initial values of PPGC01Band PPGC02B outputs as well as their output logic.

Positive logic output: Setting the bit to 0 specifies that the output is initially low and driven high upon a match between the counter value and specified dead time.

Negative logic output: Setting the bit to 1 specifies that the output is initially high and driven low upon a match between the counter value and specified dead time.

Enabling or disabling PPG outputs

The setting of the I/O port specify whether PPG outputs are enabled or disabled. When outputs are disabled, no PPG waveforms appear while the counter is operating, allowing the PPGC01B and PPGC02B pins to be used as normal input/output pins.

Using the TCC0 as a normal timer/counter

The TCC0 can be used as a normal timer/counter when PPG outputs are disabled using the setting of the I/O port. In that case, use an INTTCC0P interrupt, which occurs upon a match with the value specified in the data register (TC0DRC). Setting count start mode (TC0CR2<CSTC>) in command start and capture mode.

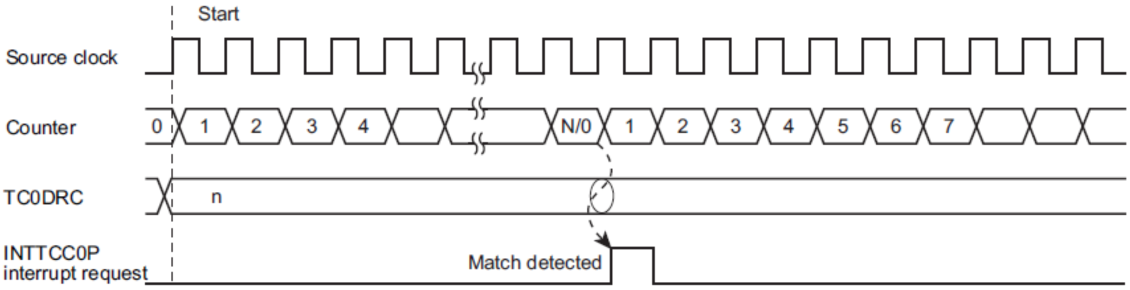


Figure 10.42 Using the TCC0 as a normal timer/counter (when TC0CR3<CSIDIS> is "1")

10.6.5.8 Noise canceller

A digital noise canceller eliminates noise from the input signal on the TCC0 pin.

The digital noise canceller uses a sampling clock of fcgck/4, fcgck/2 or fcgck, as specified with TC0CR1<NCRSEL>, and samples the signal five times. It accepts a level input which is continuous at least over the period of time required for five samplings. Any level input which does not continue over the period of time required for five samplings is canceled as noise.

TC0CR1 <NCRSEL>	Sampling Frequency (Number of Samplings)	Pulse Width Always Assumed as Noise			Pulse Width Always Assumed as Signal		
			at 8 MHz	at 16 MHz		at 8 MHz	at 16 MHz
00	fcgck/4 (5 times)	16/fcgck [s]	2 [μs]	1 [μs]	20/fcgck [s]	2.5 [μs]	1.25 [μs]
01	fcgck/2 (5 times)	8/fcgck [s]	1 [μs]	500 [ns]	10/fcgck [s]	1.25 [μs]	0.625 [μs]
10	fcgck (5 times)	4/fcgck [s]	0.5 [μs]	250 [ns]	5/fcgck [s]	0.625 [μs]	0.3125 [μs]
11	(None)	None	-	-	(1/fcgck)		

Table 10.20 Noise Canceller Settings

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

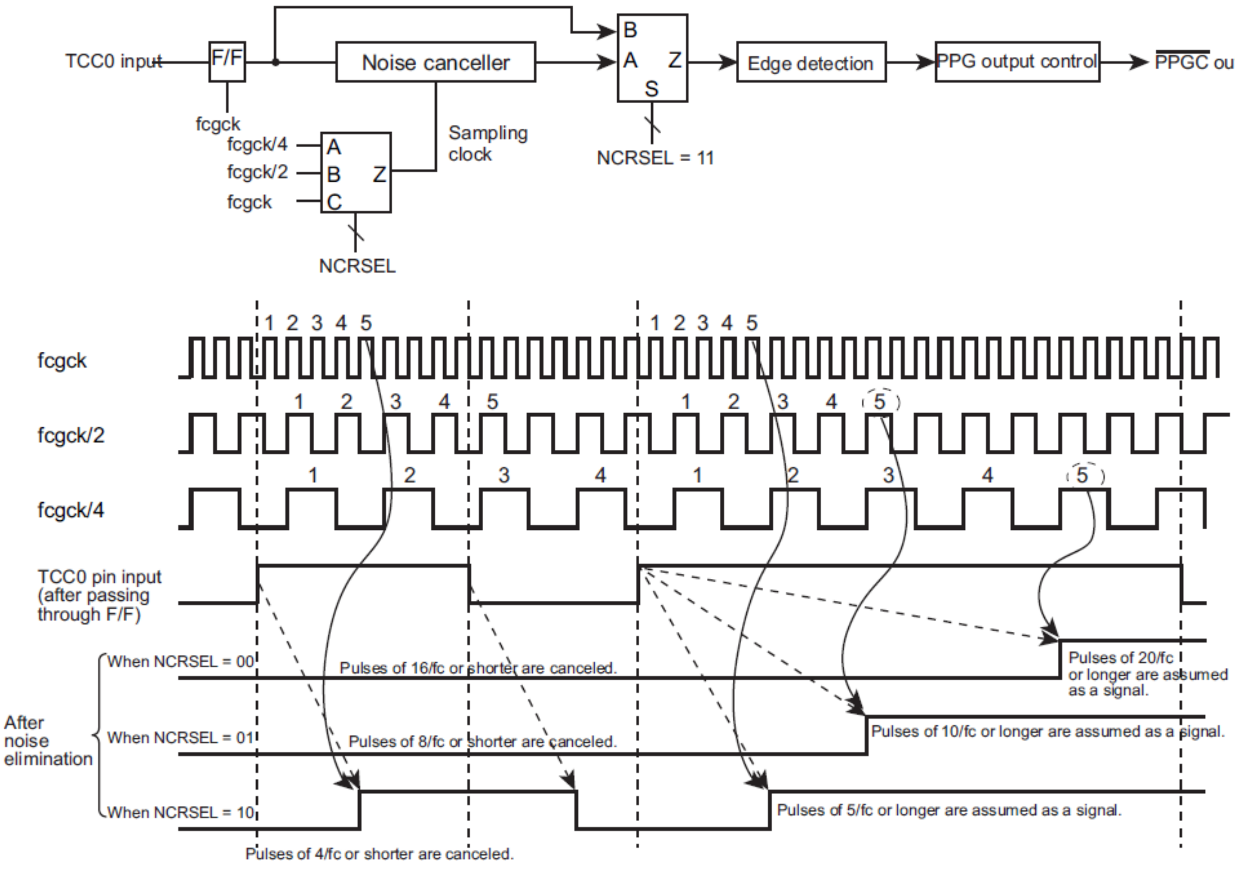

TCC0 input
F/F
fcgck
Noise canceller
B
A
Z
S
Edge detection
PPG output control
PPGC output
NCRSEL = 11
Sampling clock
fcgck/4
fcgck/2
fcgck
A
B
C
Z
NCRSEL
fcgck
fcgck/2
fcgck/4
TCC0 pin input (after passing through F/F)
After noise elimination
When NCRSEL = 00
When NCRSEL = 01
When NCRSEL = 10
Pulses of 16/fc or shorter are canceled.
Pulses of 8/fc or shorter are canceled.
Pulses of 4/fc or shorter are canceled.
Pulses of 20/fc or longer are assumed as a signal.
Pulses of 10/fc or longer are assumed as a signal.
Pulses of 5/fc or longer are assumed as a signal.
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4
1 2 3 4 5

Figure 10.41 Noise canceller operation

・ When TC0CR1<NCRSEL> = 00, a TCC0 input level after passing through the F/F is always canceled if its duration is 16/fcgck [s] or less and always assumed as a signal if its duration is 20/fcgck [s] or greater. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay between 21/fcgck [s] and 24/fcgck [s] before the PPG outputs vary.

・ When TC0CR1<NCRSEL> = 01, a TCC0 input level after passing through the F/F is always canceled if its duration is 8/fcgck [s] or less and always assumed as a signal if its duration is 10/fcgck [s] or greater. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay between 13/fcgck [s] and 14/fcgck [s] before the PPG outputs vary.

・ When TC0CR1<NCRSEL> = 10, a TCC0 input level after passing through the F/F is always canceled if its duration is 4/fcgck [s] or less and always assumed as a signal if its duration is 5/fcgck [s] or greater. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay of 5/fcgck [s] before the PPG outputs vary.

・ When TC0CR1<NCRSEL> = 11, a pulse shorter than 1/fcgck may be assumed as a signal or canceled as noise in the first-stage F/F. Ensure that input signal pulses are longer than 1/fc. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay of 4/fcgck [s] before the PPG outputs vary.

Note 1: If the pin input level changes while the specified noise elimination threshold is being modified, the noise canceller may assume noise as a pulse or cancel a pulse as noise.

Note 2: If noise occurs in synchronization with the internal sampling timing consecutively, it may be assumed as a signal.

Note 3: The signal supplied on the TCC0 pin requires 1/fcgck [s] or less to pass through the F/F.

10.6.5.9 Interruputs

The TCC0 supports three interrupt sources.

1. INTTCC0T (Trigger start interrupt)

A trigger interrupt (INTTCC0T) occurs when the counter starts upon the detection of a trigger edge specified with TC0CR1<TRGST>. This interrupt does not occur with a trigger edge for clearing the count. A trigger edge detected in trigger capture mode does not cause an interrupt. A start trigger causes an interrupt even when the counter is stopped in emergency.

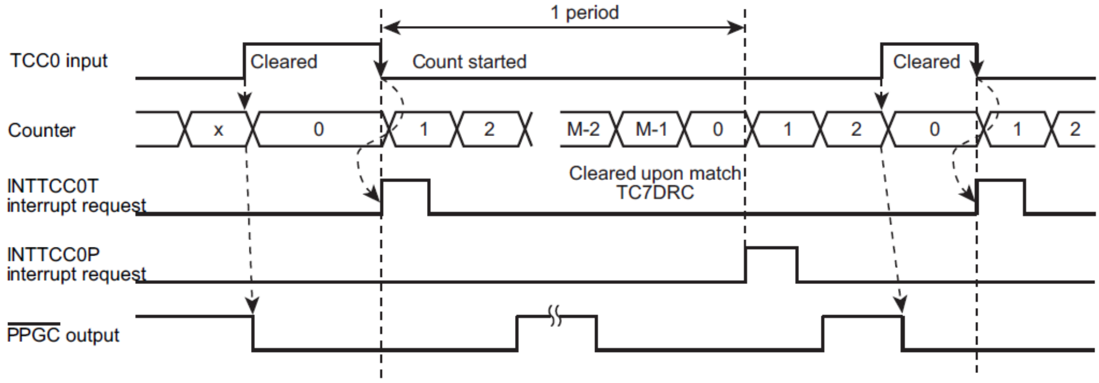


Figure 10.42 Trigger Start Interrupt

2. INTTCC0P (Period interrupt)

A period interrupt (INTTCC0P) occurs when the counter starts with a command and when the counter is cleared with the specified counter period (TC0DRC) reached, that is, at the end of a period. A match with the set period causes an interrupt even when the counter is stopped in emergency.

If a command start is specified (1 is written in TC0CR3<TCCST>) when the TCC0 pin is at a stop level, the counter does not start (INTTCC0P does not occur); a subsequent trigger start edge causes the counter to start and INTTCC0T to occur.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

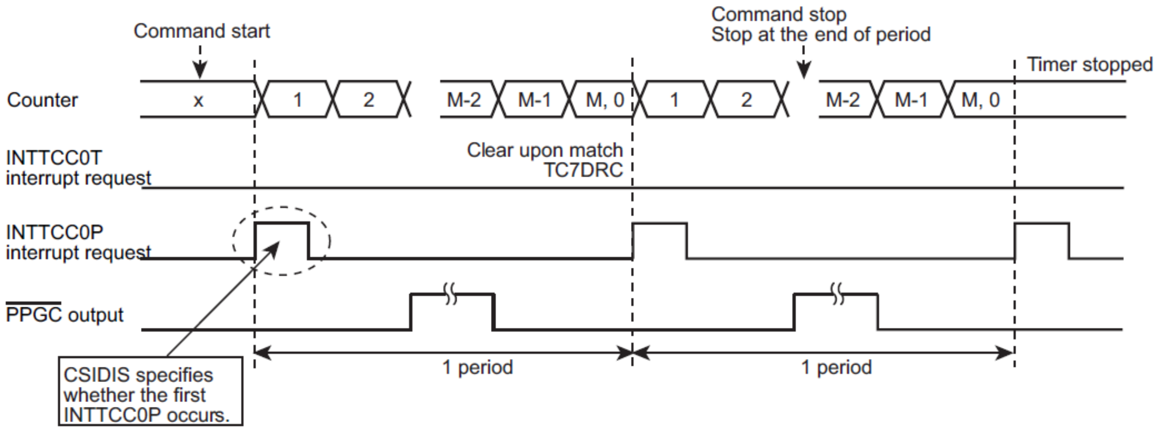


Figure 10.43 Period Interrupt

3. INTEMG0B (Emergency output stop interrupt)

An emergency output stop interrupt (INTEMG0B) occurs when the emergency output stop circuit operates to stop PPG outputs in emergency

10.6.5.10 Emergency PPG output stop feature

Setting TC0CR2<EMGIE> to 1 enables the emergency PPG output stop feature (Enables the EMG0B pin input).

A low level input detected on the EMG0 pin causes an EMG interrupt (INTEMG0) to occur with the PPG waveforms initialized (as specified with PPGC1INI and PPGC2INI). (Emergency PPG output stop) This feature only disables PPG outputs without stopping the counter. Use the EMG interrupt handler routine to stop the timer.

Note: Ensure that a low level on the EMG0 pin continues for at least 4/fcgck [s]. The emergency PPG output stop feature may not operate normally with a low level shorter than 4/fcgck [s]

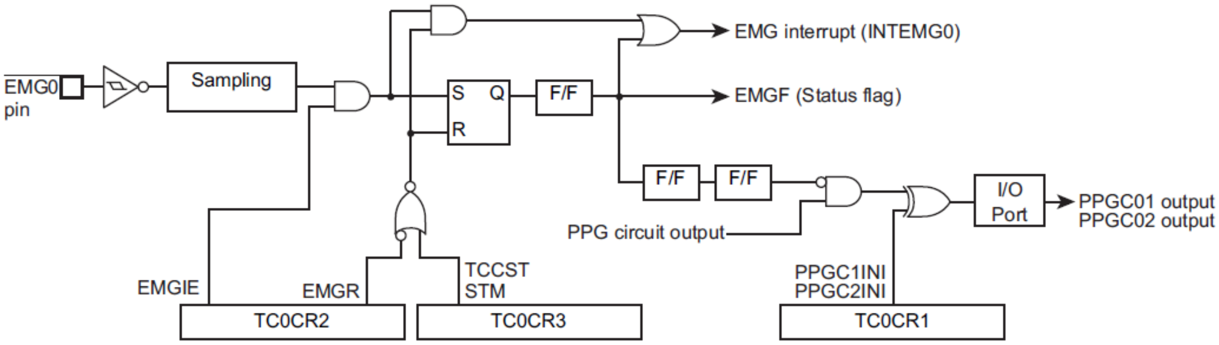


Figure 10.44 EMG0B Pin

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Enabling/disabling input on the EMG0B pin

Setting TC0CR2<EMGIE> to 1 enables input on the EMG0 pin and setting the bit to 0 disables input on the pin. (Initially, TC0CR2<EMGIE> is set to 0, disabling an emergency output stop (EMG0B pin) input.)

The input signal on the EMG0B pin is valid only when its shared port pin is placed in input mode. Ensure that the shared port pin is placed in input mode before attempting to enable the EMG0B pin input.

The EMG0B pin input is sampled using a high-frequency clock. The emergency PPG output stop feature does not operate normally if the high-frequency clock is stopped.

Monitoring the emergency PPG output stop state

When the emergency PPG output stop feature activates, the TC0CR3<EMGF> is set to 1. 1 read from EMGF indicates that PPG outputs are disabled by the emergency PPG output stop feature.

EMG interrupt

An EMG interrupt (INTEMG0) occurs when an emergency PPG output stop input is accepted. To use an INTEMG0 interrupt for some processing, ensure that the interrupt is enabled beforehand.

When the EMG0 pin is low with TC0CT2<EMGIE> set to 1 (EMG0 pin input enabled), an attempt to cancel the emergency PPG output stop state results in an interrupt being generated again, with the emergency PPG output stop state reestablished.

An INTEMG interrupt occurs whenever a stop input is accepted when TC0CR2<EMGIE> = 1, regardless of whether the timer is operating.

Canceling the emergency PPG output stop state

To cancel the emergency PPG output stop state, ensure that the input on the EMG0 pin is high, set TC0CR3<TCCST> to 0 and TC0CR3<STM> to 00 to stop the timer, and then set TC0CR2<EMGR> to 1. Setting EMGR to 1 cancels the stop state only when TCCST = 0 and TC0CR3<STM> = 00; ensure that TC0CR3<TCCST> = 0 and TC0CR3<STM> = 00 before setting TC0CR2<EMGR> to 1. If the input on the EMG0 pin is low and TC0CR2<EMGIE> = 1 when the emergency PPG output stop state is canceled, the timer re-enters the emergency PPG output stop state and an INTEMG interrupt occurs.

Restarting the timer after canceling the emergency PPG output stop state

To restart the timer after canceling the emergency PPG output stop state, reconfigure the control registers (TC0CR1, TC0CR2, TC0CR3) before restarting the timer.

The timer cannot restart in the emergency PPG output stop state. Monitor the emergency PPG output stop state and cancel the state before reconfiguring the control registers to restart the timer. Ensure that the control registers are reconfigured according to the appropriate procedure for configuring timer operation control.

Response time between EMG0B pin input and PPG outputs being initialized

The time between a low level input being detected on the EMG0 pin and the PPG outputs being initialized is up to 4/fcgck [s].

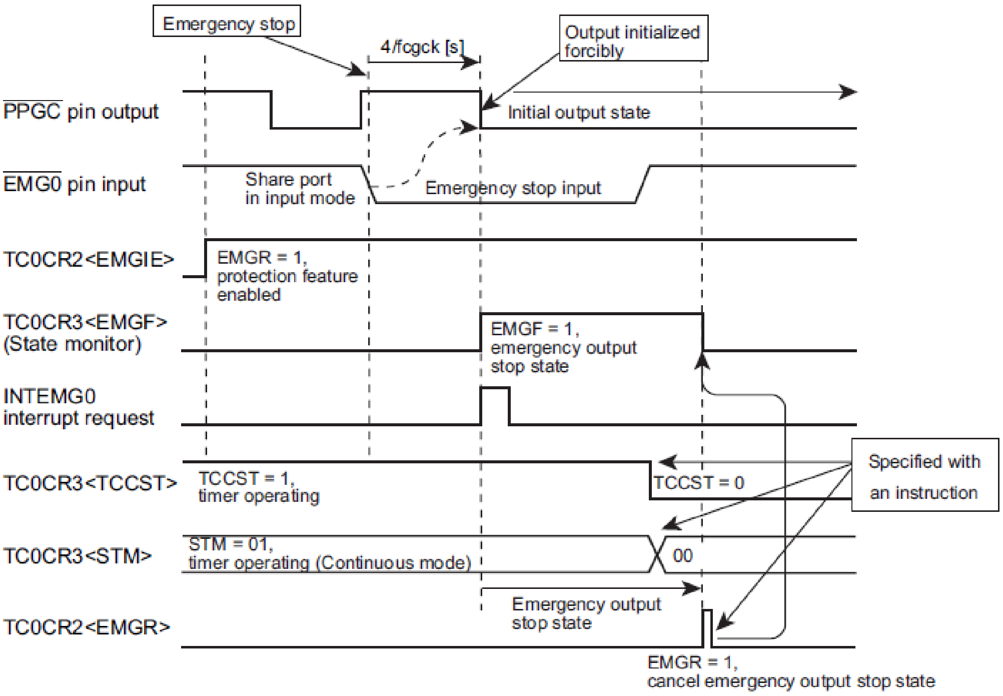

Emergency stop
4/fcgck [s]
Output initialized forcibly
PPGC pin output
Initial output state
EMG0 pin input
Share port in input mode
Emergency stop input
TC0CR2<EMGIE>
EMGR = 1, protection feature enabled
TC0CR3<EMGF> (State monitor)
EMGF = 1, emergency output stop state
INTEMG0 interrupt request
TC0CR3<TCCST>
TCCST = 1, timer operating
TCCST = 0
Specified with an instruction
TC0CR3<STM>
STM = 01, timer operating (Continuous mode)
00
Emergency output stop state
TC0CR2<EMGR>
EMGR = 1, cancel emergency output stop state

Figure 10.44 Timing between EMG0B pin input being detected and PPG outputs being disabled

10.6.5.11 TCC0 operation and MCU operating mode

The TCC0 operates when the microcontroller is placed in NORMAL1, NORMAL2, IDLE1, or IDLE2 mode. If the mode changes from NORMAL or IDLE to STOP, SLOW, or SLEEP while the TCC0 is operating, the TCC0 is initialized and stops operating.

To change the microcontroller operating mode from NORMAL to STOP, SLOW, or SLEEP, ensure that the TCC0 timer is stopped before attempting to execute a mode change instruction.

To change the mode from STOP, SLOW, or SLEEP to NORMAL to restart the TCC0, reconfigure all registers according to the appropriate TCC0 operation procedure.

10.6.5.12 Considerations for Using the development tools of TCC0

When stopped a program by the break of the debugger, TCC0 suspend PPG output, and change to initial value depend on TC0CR1<PPGC1INI and PPGC2INI> setting. When the break is released, PPG

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

output is restarted from the point at which it was suspended. The setting of development tools can continue PPG output during break. For detail refer to the instruction manual of the development tools.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

10.7 16-bit Timer Counter (TCA)

MQ6935 contains 1 channels of high-performance 16-bit timer counters (TCA).

This chapter describes the 16-bit timer counter A0. For 16-bit timer counters A1, replace the SFR addresses and pin names as shown in Table 10.21 and Table 10.22.

	TAxDAL (Address)	TAxDRAH (Address)	TAxDRBL (Address)	TAxDRBH (Address)	TAxMOD (Address)	TAxCR (Address)	TAxSR (Address)	Low power consumption register
Timer counter A0	TA0DRAL (0x002D)	TA0DRAH (0x002E)	TA0DRBL (0x002F)	TA0DRBH (0x0030)	TA0MOD (0x0031)	TA0CR (0x0032)	TA0SR (0x0033)	POFFCR0 <TCA0EN>
Timer counter A1	TA1DRAL (0x0FA8)	TA1DRAH (0x0FA9)	TA1DRBL (0x0FAA)	TA1DRBH (0x0FAB)	TA1MOD (0x0FAC)	TA1CR (0x0FAD)	TA1SR (0x0FAE)	POFFCR0 <TCA1EN>

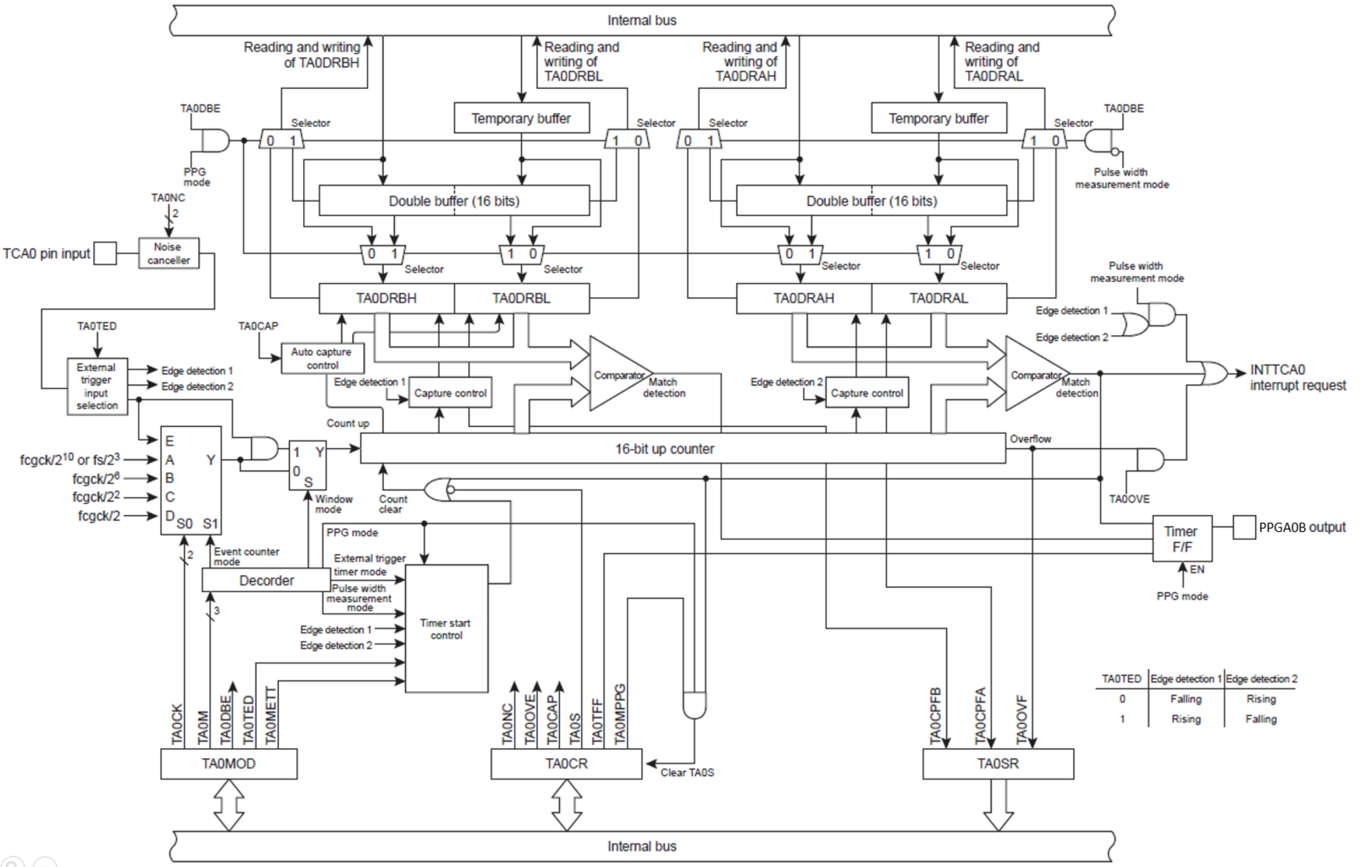
Table 10.21 SFR Address Assignment

	Timer Input Pin	PPG Output Pin
Timer counter A0	TCA0 pin	PPGA0B pin
Timer counter A1	TCA1 pin	PPGA1B pin

Table 10.22 Pin Names

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

10.7.1 Configuration



TA0TED	Edge detection 1	Edge detection 2
0	Falling	Rising
1	Rising	Falling

Figure 10.45 16-bit Timer Counter

10.7.2 Control

Timer Counter A0 is controlled by the low power consumption register (POFFCR0), the timer counter A0 mode register (TA0MOD), the timer counter A0 control register (TA0CR) and two 16-bit timer A0 registers (TA0DRA and TA0DRB).

Low Power Consumption Register 0

POFFCR0 (0x0F74)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCA1EN	TCA0EN
Read/Write	R	R	R/W	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,TC03 enable control	0: Disable 1: Enable
TC001EN	TC00,TC01 enable control	0: Disable 1: Enable
TCC0EN	TCC0 enable control	0: Disable 1: Enable
TCA1EN	TCA1 enable control	0: Disable 1: Enable
TCA0EN	TCA0 enable control	0: Disable 1: Enable

Timer Counter A0 Mode Register

TA0MOD (0x0031)	7	6	5	4	3	2	1	0
Bit Symbol	TA0DBE	TA0TED	TA0MCAP TA0METT	TA0CK		TA0M		
Read/Write	R/W	R/W	R/W	R/W		R/W		
After reset	1	0	0	0	0	0	0	0

<table>
<tr><td>TA0DBE</td><td>Double buffer control</td><td colspan="4">0: Disable the double buffer
1: Enable the double buffer</td></tr>
<tr><td>TA0TED</td><td>External trigger input selection</td><td colspan="4">0: Rising edge / H Level
1: Falling edge / L Level</td></tr>
<tr><td>TA0MCAP</td><td>Pulse width measurement mode control</td><td colspan="4">0: Double edge capture
1: Single edge capture</td></tr>
<tr><td>TA0METT</td><td>External trigger timer mode control</td><td colspan="4">0: Trigger start
1: Trigger start and stop</td></tr>
<tr><td rowspan="6">TA0CK</td><td rowspan="6">Timer counter 1 source clock selection</td><td rowspan="2"></td><td colspan="2">Normal 1/2, IDLE 1/2 mode</td><td rowspan="2">SLOW 1/2 mode
SLEEP 1 mode</td></tr>
<tr><td>SYSCR1
<DV9CK>=0</td><td>SYSCR1
<DV9CK>=1</td></tr>
<tr><td>00:</td><td>fcgck/2^{10}</td><td>fs/2^3</td><td>fs/2^3</td></tr>
<tr><td>01:</td><td>fcgck/2^6</td><td>fcgck/2^6</td><td>-</td></tr>
<tr><td>10:</td><td>fcgck/2^2</td><td>fcgck/2^2</td><td>-</td></tr>
<tr><td>11:</td><td>fcgck/2</td><td>fcgck/2</td><td>-</td></tr>
<tr><td rowspan="8">TA0M</td><td rowspan="8">Timer counter 1 operation mode selection</td><td>000:</td><td colspan="3">Timer mode</td></tr>
<tr><td>001:</td><td colspan="3">Timer mode</td></tr>
<tr><td>010:</td><td colspan="3">Event counter mode</td></tr>
<tr><td>011:</td><td colspan="3">PPG output mode (Software start)</td></tr>
<tr><td>100:</td><td colspan="3">External trigger time mode</td></tr>
<tr><td>101:</td><td colspan="3">Window mode</td></tr>
<tr><td>110:</td><td colspan="3">Pulse width measurement mode</td></tr>
<tr><td>111:</td><td colspan="3">Reserved</td></tr>
</table>

Note 1): fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2): Set TA0MOD in the stopped state (TA0CR <TA0S>="0"). Writing to TA0MOD is invalid during the operation (TA0CR <TA0S>="1").

Timer Counter A0 Control Register

TA0CR (0x0032)	7	6	5	4	3	2	1	0
Bit Symbol	TA0OVE	TA0TFF	TA0NC		-	-	TA0ACAP TA0MPPG	TA0S
Read/Write	R/W	R/W	R/W		R	R	R/W	R/W
After reset	0	1	0	0	0	0	0	0

<table>
<tr><td>TA0OVE</td><td>Overflow interrupt control</td><td colspan="3">0: Generate no INTTCA0 interrupt request when the counter overflow occurs.
1: Generate an INTTCA0 interrupt request when the counter overflow occurs.</td></tr>
<tr><td>TA0TFF</td><td>Timer F/F control</td><td colspan="3">0: Clear
1: Set</td></tr>
<tr><td>TA0MCAP</td><td>Pulse width measurement mode control</td><td colspan="3">0: Double edge capture
1: Single edge capture</td></tr>
<tr><td rowspan="5">TA0NC</td><td rowspan="5">Noise canceller sampling interval setting</td><td></td><td>Normal 1/2 or IDLE 1/2 mode</td><td>SLOW 1/2 mode
SLEEP 1 mode</td></tr>
<tr><td>00:</td><td>No noise canceller</td><td>No noise canceller</td></tr>
<tr><td>01:</td><td>fcgck/2</td><td>-</td></tr>
<tr><td>10:</td><td>fcgck/2^2</td><td>-</td></tr>
<tr><td>11:</td><td>fcgck/2^8</td><td>fs/2</td></tr>
<tr><td>TA0ACAP</td><td>Auto capture function</td><td colspan="3">0: Disable the auto capture
1 Enable the auto capture</td></tr>
<tr><td>TA0MPPG</td><td>PPG output control</td><td colspan="3">0: Continuous
1: One-shot</td></tr>
<tr><td>TA0S</td><td>Timer counter A start control</td><td colspan="3">0: Stop and counter clear
1: Start</td></tr>
</table>

Note 1): The auto capture can be used only in the timer, event counter, external trigger timer and window modes.

Note 2): Set TA0TFF, TA0OVE and TA0NC in the stopped state (TA0S="0"). Writing is invalid during the operation (TA0S="1").

Note 3): When the STOP mode is started, the start control (TA0S) is automatically cleared to "0" and the timer stops. Set TA0S again to use the timer counter after the release of the STOP mode.

Note 4): When a read instruction is executed on TA0CR, bits 3 and 2 are read as "0".

Note 5): Do not set TA0NC to "01" or "10" when the SLOW 1/2 or SLEEP 1 mode is used. Setting TA0NC to "01" or "10" stops the noise canceller and no signal is input to the timer.

Timer Counter A0 Status Register

TA0SR (0x0033)	7	6	5	4	3	2	1	0
Bit Symbol	TA0OVF	-	-	-	-	-	TA0CPFA	TA0CPFB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

TA0OVF	Overflow flag	0: No overflow has occurred. 1: At least an overflow has occurred.
TA0CPFA	Capture completion flag A	0: No capture operation has been executed. 1: At least a pulse width capture has been executed in the double-edge capture
TA0CPFB	Capture completion flag B	0: No capture operation has been executed. 1: At least a capture operation has been executed in the single-edge capture. At least a pulse duty width capture has been executed in the double-edge capture.

Note 1): TA0OVF, TA0CPFA and TA0CPFB are cleared to "0" automatically after TA0SR is read. Writing to TA0SR is invalid.

Note 2): When a read instruction is executed on TA0SR, bits 6 to 2 are read as "0".

Timer Counter A0 Register AH

TA0DRAH (0x002E)	15	14	13	12	11	10	9	8
Bit Symbol	TA0DRAH							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Timer Counter A0 Register AL

TA0DRAL (0x002D)	7	6	5	4	3	2	1	0
Bit Symbol	TA0DRAL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Timer Counter A0 Register BH

TA0DRBH (0x0030)	15	14	13	12	11	10	9	8
Bit Symbol	TA0DRBH							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Timer Counter A0 Register BL

TA0DRBL (0x002F)	7	6	5	4	3	2	1	0
Bit Symbol	TA0DRBL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	1	1	1	1	1	1	1	1

Note 1): When a write instruction is executed on TA0DRAL (TA0DRBL), the set value does not become effective immediately, but is temporarily stored in the temporary buffer. Subsequently, when a write instruction is executed on the higher-level register, TA0DRAH (TA0DRBH), the 16-bit set values are collectively stored in the double buffer or TA0DRAL/H. When setting data to the timer counter A0 register, be sure to write the data into the lower level register and the higher level in this order.
Note 2): The timer counter A0 register is not writable in the pulse width measurement mode.

10.7.3 Low Power Consumption Function

Timer counter A0 has the low power consumption register (POFFCR0) that saves power consumption when the timer is not used.

Setting POFFCR0<TCA0EN> to "0" disables the basic clock supply to timer counter A0 to save power. Note that this makes the timer unusable. Setting POFFCR0<TCA0EN> to "1" enables the basic clock supply to timer counter A0 and allows the timer to operate.

After reset, POFFCR0<TCA0EN> is initialized to "0", and this makes the timer unusable. When using the timer for the first time, be sure to set POFFCR0<TCA0EN> to "1" in the initial setting of the program (before the timer control register is operated).

Do not change POFFCR0<TCA0EN> to "0" during the timer operation. Otherwise timer counter A0 may operate unexpectedly.

10.7.4 Timer Function

Timer counter A0 has six types of operation modes; timer, external trigger timer, event counter, window, pulse width measurement and programmable pulse generate (PPG) output modes.

10.7.4.1 Timer Mode

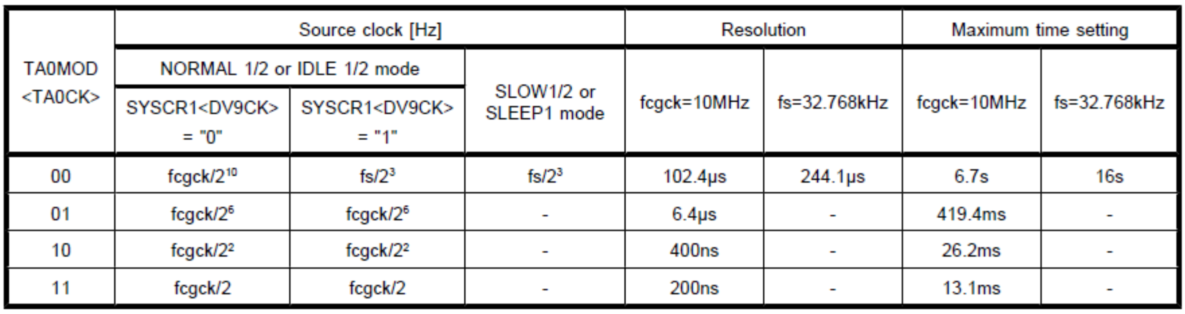
In the timer mode, the up-counter counts up using the internal clock, and interrupts can be generated regularly at specified times.

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "000" or "001" activates the timer mode. Select the source clock at TA0MOD <TA0CK>.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Setting TA0CR <TA0S> to "1" starts the timer operation. After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> becomes invalid. Be sure to complete the required mode settings before starting the timer.



TA0MOD <TA0CK>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
00	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	6.7s	16s
01	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	419.4ms	-
10	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	26.2ms	-
11	fcgck/2	fcgck/2	-	200ns	-	13.1ms	-

Table 10.23 Timer Mode Resolution and Maximum Time Setting

(b) Operation

Setting TA0CR <TA0S> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up-counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting. Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

(c) Auto Capture

The latest contents of the up counter can be taken into timer register B (TA0DRB) by setting TA0CR <TA0ACAP> to "1" (auto capture function). When TA0CR<TA0ACAP> is "1", the current contents of the up counter can be read by reading TA0DRBL. TA0DRBH is loaded at the same time as TA0DRBL is read. Therefore, when reading the captured value, be sure to read TA0DRBL and TA0DRBH in this order. (The capture time is the timing when TA0DRBL is read.) The auto capture function can be used whether the timer is operating or stopped. When the timer is stopped, TA0DRBL is read as "0x00". TA0DRBH keeps the captured value after the timer stops, but it is cleared to "0x00" when TA0DRBL is read while the timer is stopped.

If the timer is started with TA0CR <TA0ACAP> written to "1", the auto capture is enabled immediately after the timer is started.

Note): The value set to TA0CR <TA0ACAP> cannot be changed at the same time as TA0CR <TA0S> is rewritten from "1" to "0". (This setting is invalid.)

(d) Register Buffer Configuration

1. Temporary Buffer

MQ6935 contains an 8-bit temporary buffer. When a write instruction is executed on TA0DRAL, the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TA0DRAH, the set value is stored into the double buffer or TA0DRAH. At the same time, the set

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

value in the temporary bufferis stored into the double buffer or TA0DRAL. (This structure is designed to enable the set values of the lower-level and higher-level registers simultaneously.) Therefore, when setting data to TA0DRA, be sure to write the data into TA0DRAL and TA0DRAH in this order.

2. **Double Buffer**

In the MQ6935, the double buffer can be used by setting TA0CR <TA0DBF>. Setting TA0CR <TA0DBF> to "0" disables the double buffer. Setting TA0CR <TA0DBF> to "1" enables the double buffer.

- When the double buffer is enabled

When a write instruction is executed on TA0DRAH during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L compare the up counter value to the last set values. If the values are matched, an INTTCA0 interrupt request is generated and the double buffer set value is stored in TA0DRAH/L. Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on TA0DRAH/L, the double buffer value (the last set value) is read, rather than the TA0DRAH/L values (the current effective values).

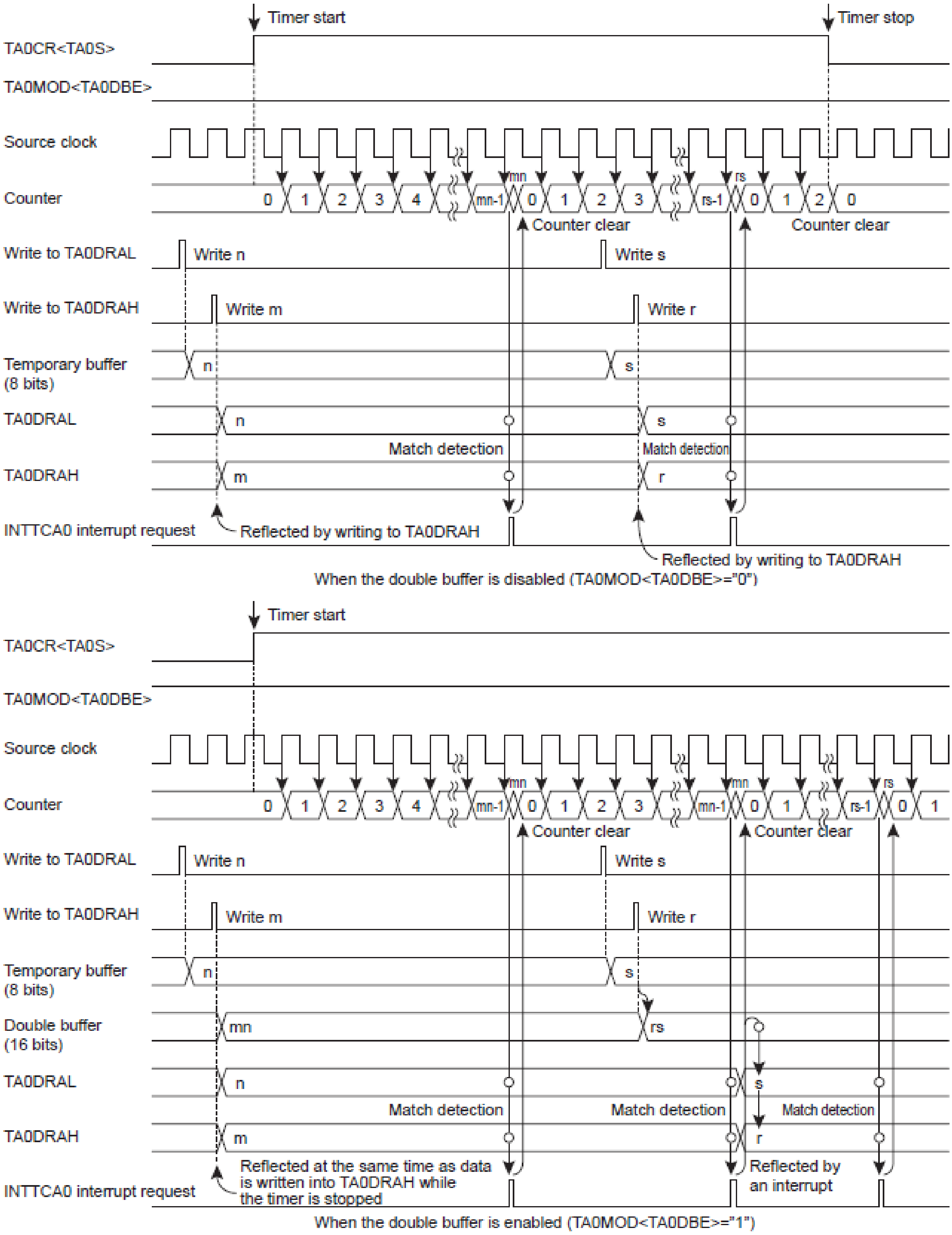
When a write instruction is executed on TA0DRAH/L while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L.

- When the double buffer is disabled

When a write instruction is executed on TA0DRAH during the timer operation, the set value is immediately stored into TA0DRAH/L. Subsequently, the match detection is executed using a new set value.

If the values set to TA0DRAH/L are smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If that is a problem, enable the double buffer.

When a write instruction is executed on TA0DRAH/L while the timer is stopped, the set value is immediately stored into TA0DRAH/L.



When the double buffer is disabled (TA0MOD<TA0DBE>="0")

When the double buffer is enabled (TA0MOD<TA0DBE>="1")

Figure 10.46 Timer Mode Timing Chart

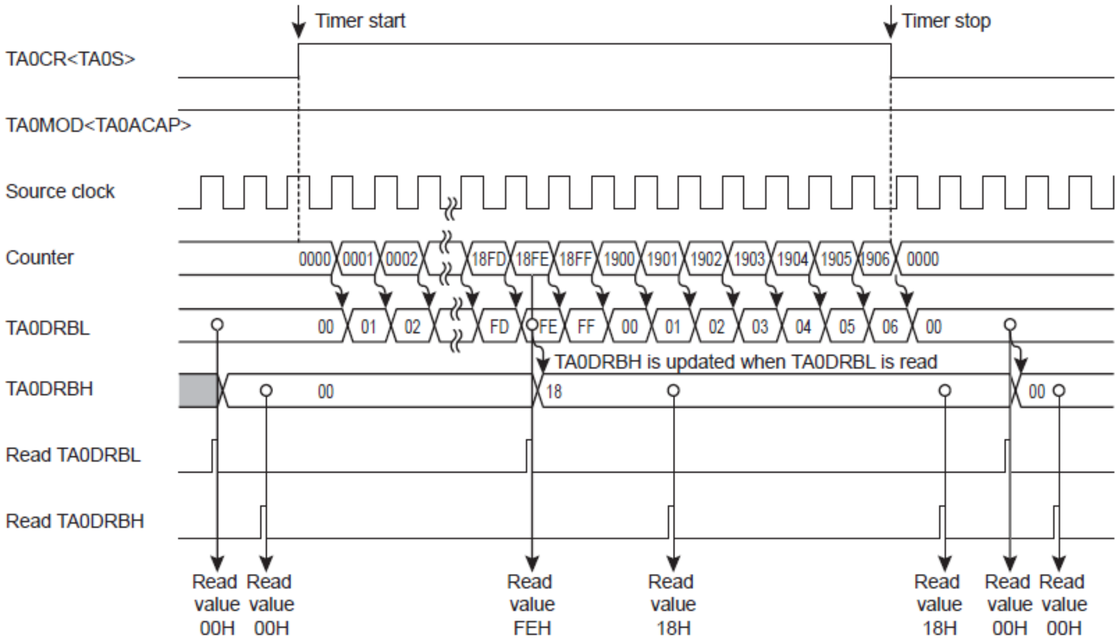


Figure 10.47 Timer Mode Timing Chart (Auto Capture)

10.7.4.2 External Trigger Timer Mode

In the external trigger timer mode, the up counter starts counting when it is triggered by the input to the TCA0 pin.

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "100" activates the external trigger timer mode. Select the source clock at TA0MOD <TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD <TA0TED>. Setting TA0MOD <TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the timer is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments according to the selected source clock. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting.

When TA0MOD <TA0METT> is "1" and the edge opposite to the selected trigger edge is detected, the up counter stops counting and is cleared to "0x0000". Subsequently, when the selected trigger edge is detected, the up counter restarts counting. In this mode, an interrupt request can be generated by detecting that the input pulse exceeds a certain pulse width. If TA0MOD <TA0METT> is "0", the detection of the selected edge and the opposite edge is ignored during the period from the detection of the specified trigger edge and the start of counting through until the match detection.

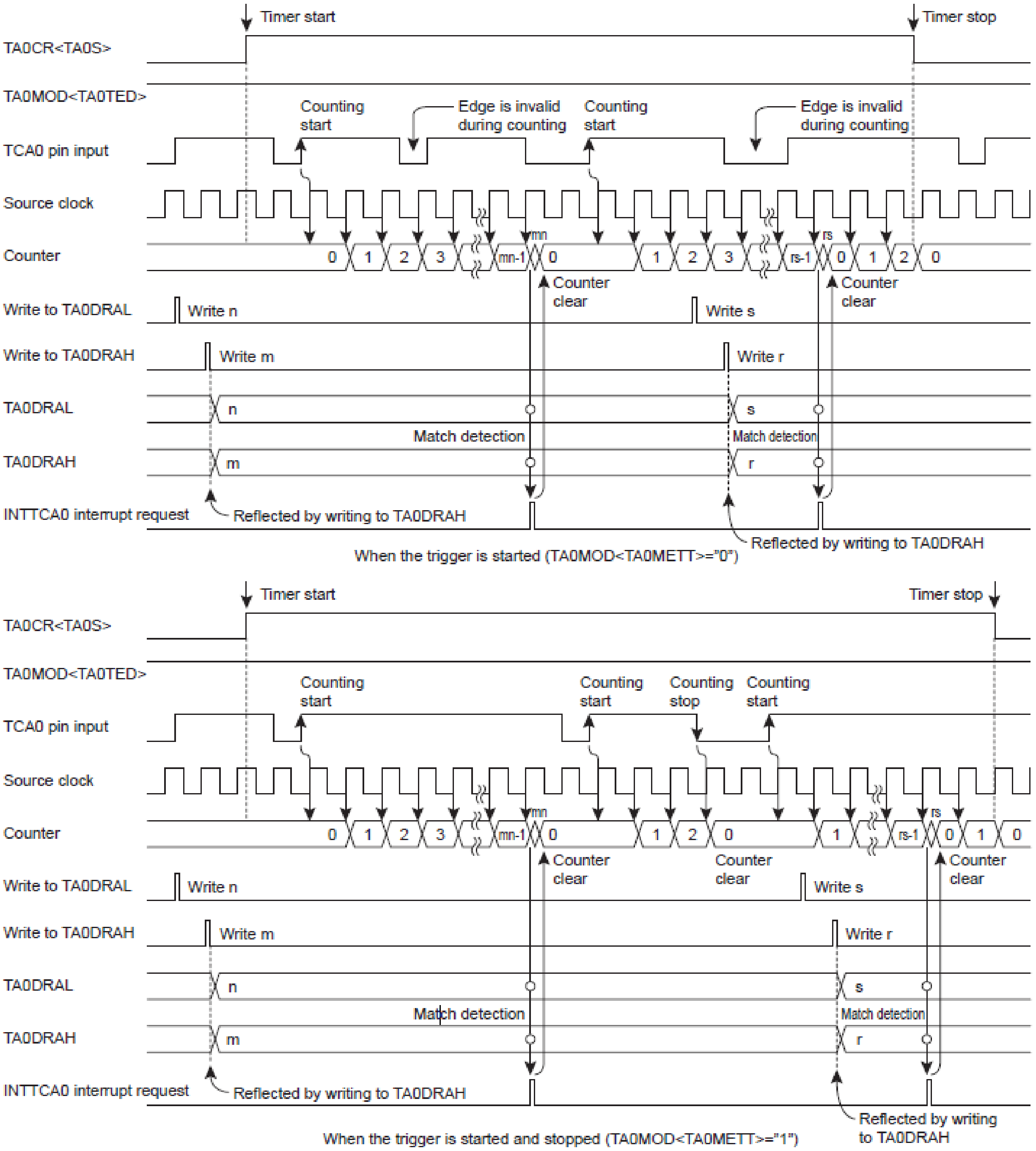
Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

(c) Auto Capture

Refer to "10.7.4.1 (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "10.7.4.1 (d) Register Buffer Configuration ".



When the trigger is started (TA0MOD<TA0METT>="0")

When the trigger is started and stopped (TA0MOD<TA0METT>="1")

Figure 10.48 External Trigger Timer Mode Timing Chart

10.7.4.3 Event Counter Mode

In the event counter mode, the up counter counts up at the edge of the input to the TCA0 pin.

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "010" activates the event counter mode.

Set the trigger edge at the external trigger input selection TA0MOD <TA0TED>. Setting TA0MOD <TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge for counting up.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the event counter mode is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments.

When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting and counts up at each edge of the input to the TCA0 pin. Setting TA0CR <TA0S> to "0" during the operation causes the up counter to stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in the NORMAL 1/2 or IDLE 1/2 mode) or fs/ 2 [Hz] (in the SLOW 1/2 or SLEEP 1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

(c) Auto Capture

Refer to "10.7.4.1 (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "10.7.4.1 (d) Register Buffer Configuration ".

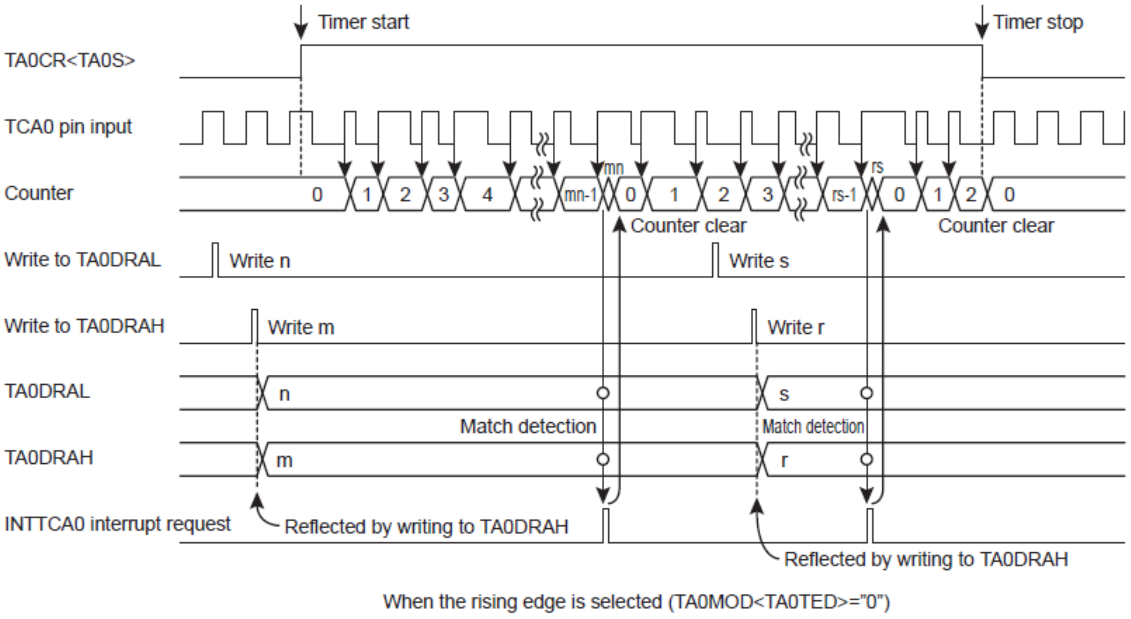


Figure 10.49 Event Counter Mode Timing Chart

10.7.4.4 Window Mode

In the window mode, the up counter counts up at the rising edge of the pulse that is logical anded product of the input pulse to the TCA0 pin (window pulse) and the internal clock.

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "101" activates the window mode. Select the source clock at TA0MOD <TA0CK>.

Select the window pulse level at the trigger edge input selection TA0MOD <TA0TED>. Setting TA0MOD <TA0TED> to "0" enables counting up as long as the window pulse is at the "H" level. Setting TA0MOD <TA0TED> to "1" enables counting up as long as the window pulse is at the "L" level.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the operation is started, when the level selected at TA0MOD <TA0TED> is input to the TCA0 pin, the up counter increments according to the source clock selected at TA0MOD <TA0CK>. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting.

The maximum frequency to be supplied must be slow enough for the program to analyze the count value. Define a frequency pulse that is sufficiently lower than the programmed internal source clock.

Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

(c) Auto Capture

Refer to "10.7.4.1 (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "10.7.4.1 (d) Register Buffer Configuration ".

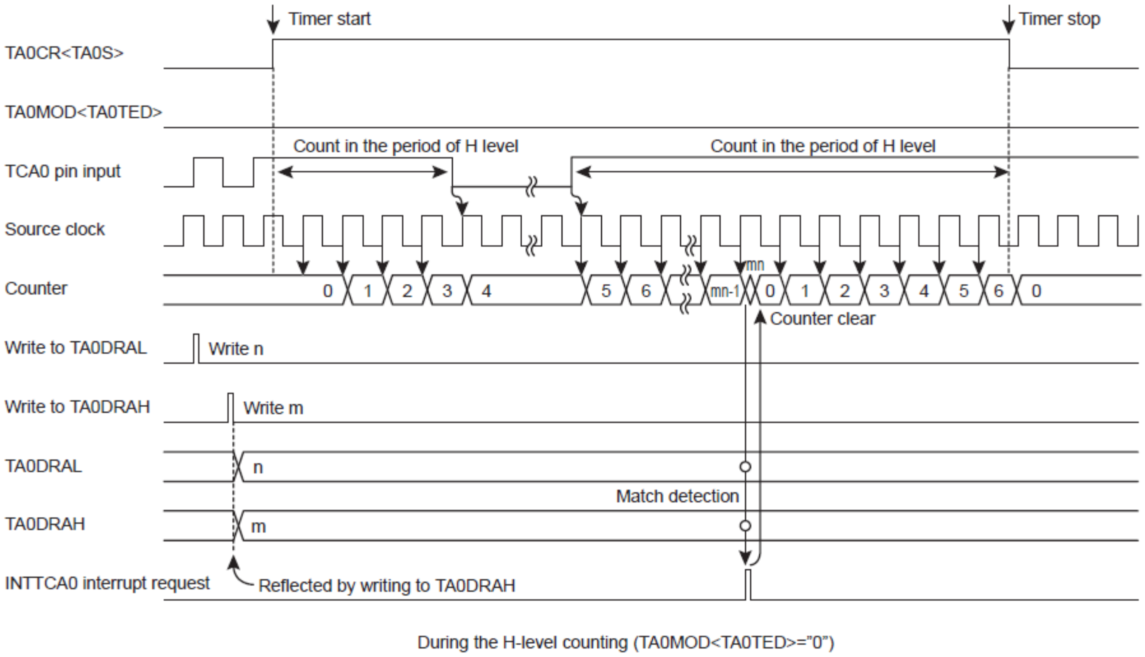

Timer start
Timer stop
TA0CR<TA0S>
TA0MOD<TA0TED>
TCA0 pin input
Count in the period of H level
Count in the period of H level
Source clock
Counter
0 1 2 3 4 5 6 mn-1 mn 0 1 2 3 4 5 6 0
Counter clear
Write to TA0DRAL
Write n
Write to TA0DRAH
Write m
TA0DRAL
n
Match detection
TA0DRAH
m
INTTCA0 interrupt request
Reflected by writing to TA0DRAH
During the H-level counting (TA0MOD<TA0TED>="0")

Figure 10.50 Window Mode Timing Chart

10.7.4.5 Pulse Width Measurement Mode

In the pulse width measurement mode, the up counter starts counting at the rising/falling edge(s) of the input to the TCA0 pin and measures the input pulse width based on the internal clock.

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "110" activates the pulse width measurement mode. Select the source clock at TA0MOD <TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD <TA0TED>. Setting TA0MOD <TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

as a trigger to start the capture.

The operation after capturing is determined by the pulse width measurement mode control TA0MOD <TA0MCAP>. Setting TA0MOD <TA0MCAP> to "0" selects the double-edge capture. Setting TA0MOD <TA0MCAP> to "1" selects the single-edge capture.

The operation to be executed in case of an overflow of the up counter can be selected at the overflow interrupt control TA0CR <TA0OVE>. Setting TA0OVE to "1" makes an INTTCA0 interrupt request occur in case of an overflow. Setting TA0OVE to "0" makes no INTTCA0 interrupt request occur in case of an overflow.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". In this time, TA0DRA and TA0DRB register are initialized to "0x0000". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the timer is started, when the selected trigger edge (start edge) is input to the TCA0 pin, INTTCA0 interrupt request is generated, and then the up counter increments according to the selected source clock. Subsequently, when the edge opposite to the selected edge is detected, the up counter value is captured into TA0DRB, an INTTCA0 interrupt request is generated, and TA0SR <TA0CPFB> is set to "1". Depending on the TA0MOD <TA0MCAP> setting, the operation differs as follows:

1. **Double-edge capture (When TA0MOD <TA0MCAP> is "0"**

   The up counter continues counting up after the edge opposite to the selected edge is detected. Subsequently, when the selected trigger edge is input, the up counter value is captured into TA0DRA, an INTTCA0 interrupt request is generated, and TA0SR <TA0CPFA> is set to "1". At this time, the up counter is cleared to "0x0000".

2. **Single-edge capture (When TA0MOD <TA0MCAP> is "1"**

   The up counter stops counting up and is cleared to "0x0000" when the edge opposite to the selected edge is detected. Subsequently, when the start edge is input, INTTCA0 interrupt request is generated, and then the up counter restarts increment.

When the up counter overflows during capturing, the overflow flag TA0SR <TA0OVF> is set to "1". At this time, an INTTCA0 interrupt request occurs if the overflow interrupt control TA0CR <TA0OVE> is set to "1".

The capture completion flags (TA0SR <TA0CPFA, TA0CPFB> and the overflow flag (TA0SR <TA0OVF>) are cleared to "0" automatically when TA0SR is read.

The captured value must be read from TA0DRB (and also from TA0DRA for the double-edge capture) before the next trigger edge is detected. If the captured value is not read, it becomes undefined. TA0DRA and TA0DRB must be read by using a 16-bit access instruction.

Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

Note): After the timer is started, if the edge opposite to the selected trigger edge is detected first, no capture is executed and no INTTCA0 interrupt request occurs. In this case, the capture starts when the selected trigger edge is detected next.

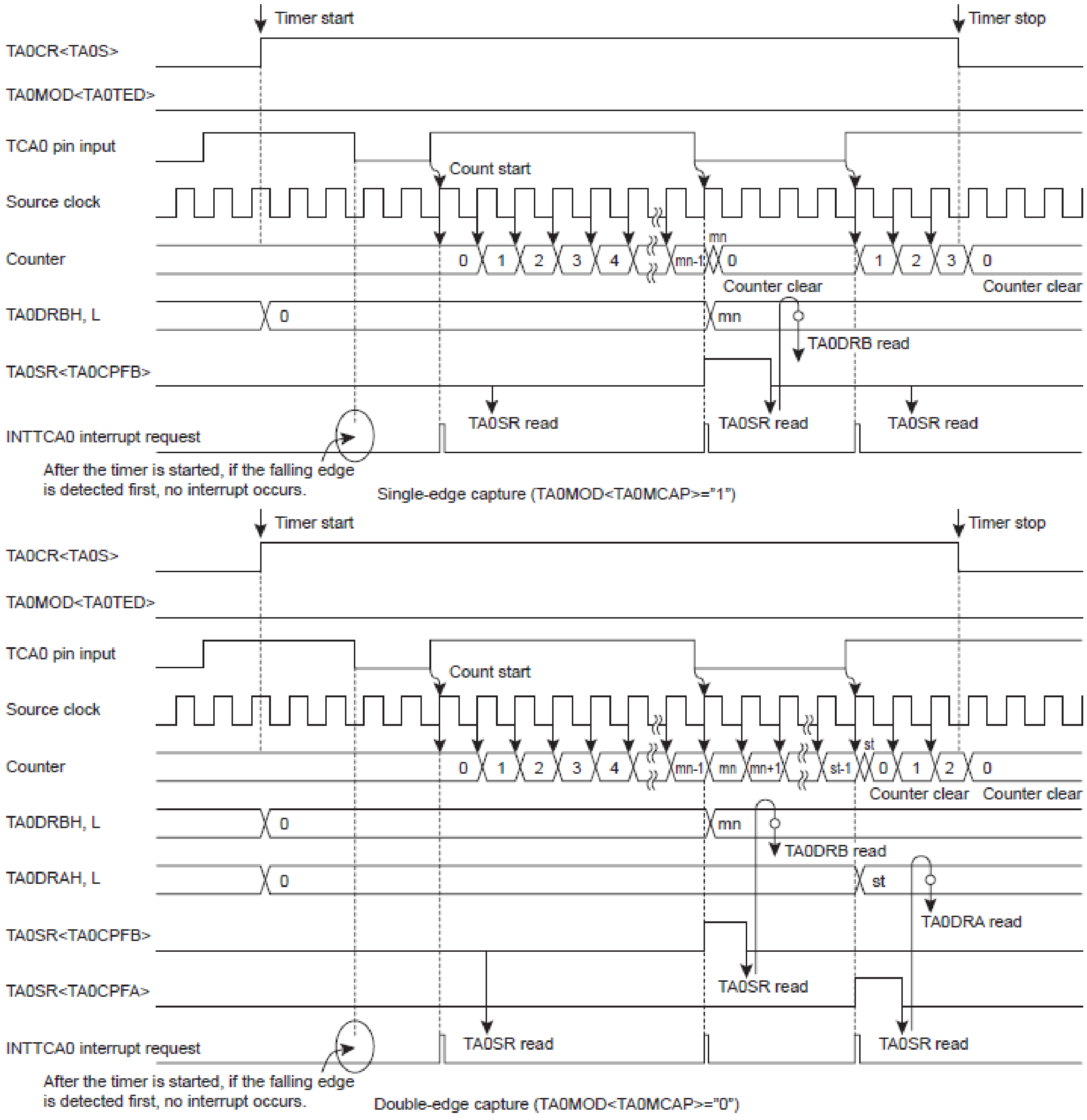


Figure 10.51 Pulse Width Measurement Mode Timing Chart

(c) Capture Process

Figure 10.31 shows an example of the capture process for INTTCA0 interrupt subroutine. The capture edge or overflow state can be easily judged by status register (TA0SR).

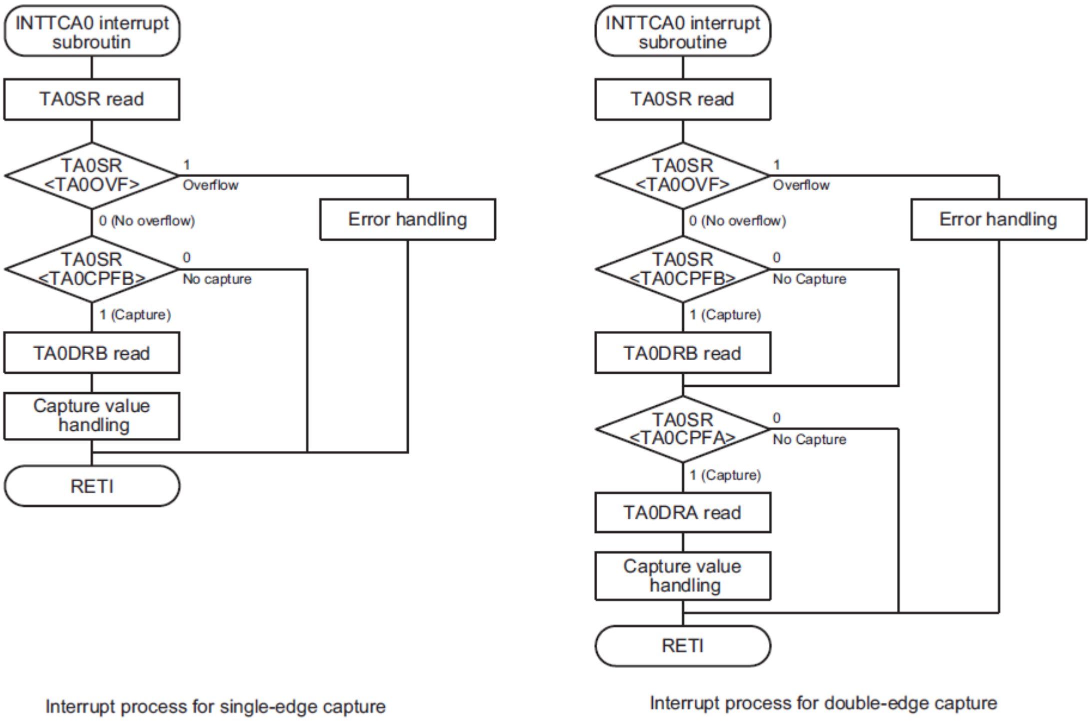


Figure 10.52 Example of Capture Process

10.7.4.6 Programmable pulse generate (PPG) mode

In the PPG output mode, an arbitrary duty pulse is output by two timer registers.

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "011" activates the PPG output mode. Select the source clock at TA0MOD <TA0CK>. Select continuous or one-shot PPG output at TA0CR <TA0MPPG>.

Set the PPG output cycle at TA0DRA and set the time until the output is reversed first at TA0DRB. Be sure to set register values so that TA0DRA is larger than TA0DRB. Note that this mode uses the PPGA0B pin. The PPGA0B pin must be set to the output mode beforehand in port settings.

Set the initial state of the PPGA0B pin at the timer flip-flop TA0CR <TA0TFF>. Setting TA0CR <TA0TFF> to "1" selects the "H" level as the initial state of the PPGA0B pin. Setting TA0CR <TA0TFF> to "0" selects the "L" level as the initial state of the PPGA0B pin.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE, TA0TFF> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the timer is started, the up counter increments.

When a match between the up counter value and the value set to timer register B (TA0DRB) is detected, the PPGA0B pin is changed to the "H" level if TA0CR <TA0TFF> is "0", or the PPGA0B pin is changed to the "L" level if TA0CR <TA0TFF> is "1".

Subsequently, the up counter continues counting. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, the PPGA0B pin is changed to the "L" level if TA0CR <TA0TEFF> is "0", or the PPGA0B pin is changed to the "H" level if TA0CR <TA0TFF> is "1". At this time, an INTTCA0 interrupt request occurs. If the PPG output control TA0CR <TA0MPPG> is set to "1" (one-shot), TA0CR <TA0S> is automatically cleared to "0" and the timer stops.

If TA0CR <TA0MPPG> is set to "0" (continuous), the up counter is cleared to "0x0000" and continues counting and PPG output. When TA0CR <TA0S> is set to "0" (including the auto stop by the one-shot operation) during the PPG output, the PPGA0B pin returns to the level set in TA0CR<TA0TFF>.

TA0CR <TA0MPPG> can be changed during the operation. Changing TA0CR <TA0MPPG> from "1" to "0" during the operation cancels the one-shot operation and enables the continuous operation. Changing TA0CR<TA0MPPG> from "0" to "1" during the operation clears TA0CR<TA0S> to "0" and stops the timer automatically after the current pulse output is completed.

Timer registers A and B can be set to the double buffer. Setting TA0CR <TA0DBF> to "1" enables the double buffer. When the values set to TA0DRA and TA0DRB are changed during the PPG output with the double buffer enabled, the writing to TA0DRA and TA0DRB will not immediately become effective but will become effective when a match between TA0DRA and the up counter is detected. If the double buffer is disabled, the writing to TA0DRA and TA0DRB will become effective immediately. If the written value is smaller than the up counter value, the up counter overflows. After a cycle, the counter match process is executed to reverse the output.

(c) Register Buffer Configuration

1. Temporary Buffer

MQ6935 contains an 8-bit temporary buffer. When a write instruction is executed on TA0DRAL (TA0DRBL), the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TA0DRAH (TA0DRBH), the set value is stored into the double buffer or TA0DRAH (TA0DRBH). At the same time, the set value in the temporary buffer is stored into the double buffer or TA0DRAL (TA0DRBL). (This structure is designed to enable the set values of the lower-level register and the higher-level register simultaneously.) Therefore, when setting data to TA0DRA (TA0DRB), be sure to write the data into TA0DRAL and TA0DRAH (TA0DRBL and TA0DRBH) in this order.

2. **Double Buffer**

In MQ6935, the double buffer can be used by setting TA0CR <TA0DBF>. Setting TA0CR <TA0DBF> to "0" disables the double buffer. Setting TA0CR <TA0DBF> to "1" enables the double buffer.

- When the double buffer is enabled

When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L (TA0DRBH/L) compare the last set values to the counter value.

If a match is detected, an INTTCA0 interrupt request is generated and the double buffer set value is stored into TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on TA0DRAH/L (TA0DRBH/L), the double buffer value (the last set value) is read, not the TA0DRAH/L (TA0DRBH/L) values (the current effective values).

When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L (TA0DRBH/L).

- When the double buffer is disabled

When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is immediately stored in TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

If the values set to TA0DRAH/L (TA0DRBH/L) are smaller than the up counter value, the up counter overflows and the match detection is executed using a new set value. As a result, the output pulse width may be longer than the set time. If that is a problem, enable the double buffer.

When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into TA0DRAH/L (TA0DRBH/L).

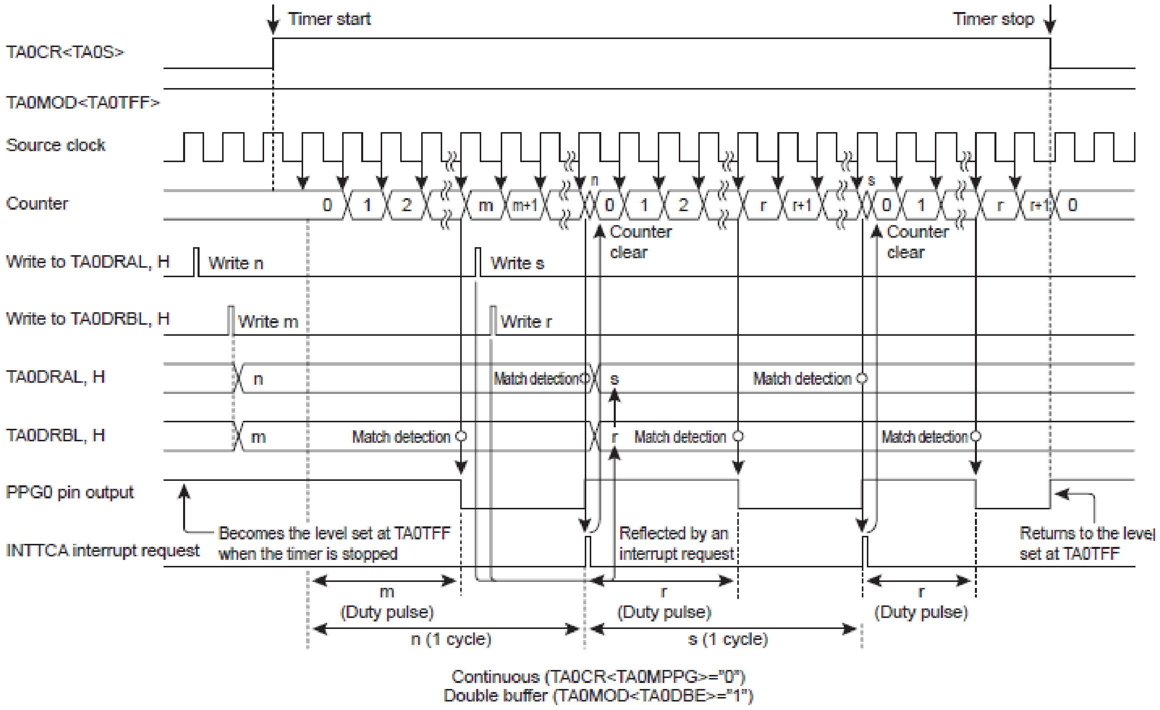


Figure 10.53 PPG Mode Timing Chart - Continuous

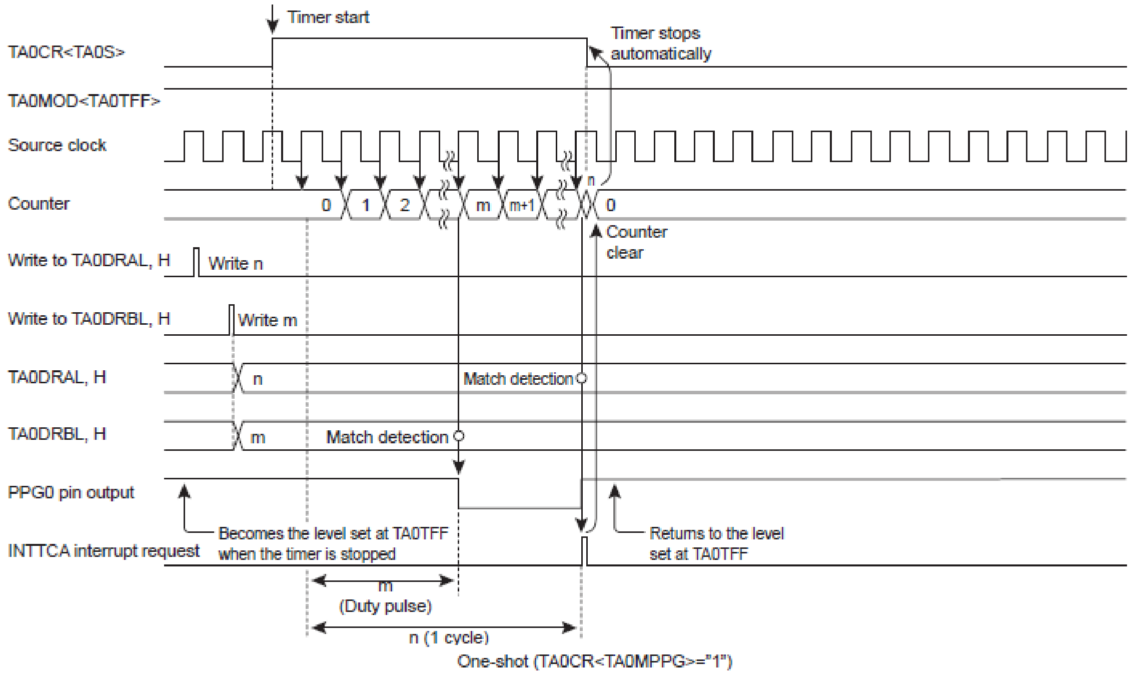


Figure 10.54 PPG Mode Timing Chart - One Shot

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

10.7.5 Noise Canceller

The digital noise canceller can be used in the operation modes that use the TCA0 pin.

When the digital noise canceller is used, the input level is sampled at the sampling intervals set at TA0CR <TA0NC>. When the same level is detected three times consecutively, the level of the input to the timer is changed.

Setting TA0CR <TA0NC> to any values than "00" allows the noise canceller to start operation, regardless of the TA0CR <TA0S> value.

When the noise canceller is used, allow the timer to start after a period of time that is equal to four times the sampling interval after TA0CR <TA0NC> is set has elapsed. This stabilizes the input signal. Set TA0CR <TA0NC> while the timer is stopped (TA0CR <TA0S> = "0"). When TA0CR <TA0S> is "1", writing is ignored.

In the SLOW 1/2 or SLEEP 1 mode, setting TA0CR <TA0NC> to "11" selects fs/2 as the source clock for the operation. Setting TA0CR <TA0NC> to "00" disables the noise canceller. Setting TA0CR <TA0NC> to "01" or "10" disables the TCA0 pin input.

TA0NC	Sampling interval	Time removed as noise	Time regarded as signal
00	None	-	-
01	200 ns (2/fcgck)	600 ns or less	800 ns or more
10	400 ns (4/fcgck)	1.2 µs or less	1.6 µs or more
11	25.6 µs (256/fcgck)	76.8 µs or less	102.4 µs or more

Table 10.24 Noise Cancel Time (fcgck = 10 [MHz])

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

11. Asynchronous Serial Interface (UART)

MQ6935 contains 3 channels of asynchronous serial interfaces (UART).This chapter describes asynchronous serial interface 0 (UART0). For UART1 and UART2, replace the SFR addresses and pin namesas shown in Table 11.1 and Table 11.2.

	UARTxCR1 (Address)	UARTxCR2 (Address)	UARTxDR (Address)	UARTxSR (Address)	RDxBUF (Address)	TDxBUF (Address)
UART0	UART0CR1 (0x001A)	UART0CR2 (0x001B)	UART0DR (0x001C)	UART0SR (0x001D)	RD0BUF (0x001E)	TD0BUF (0x001E)
UART1	UART1CR1 (0x0F54)	UART1CR2 (0x0F55)	UART1DR (0x0F56)	UART1SR (0x0F57)	RD1BUF (0x0F58)	TD1BUF (0x0F58)
UART2	UART2CR1 (0x0F5A)	UART2CR2 (0x0F5B)	UART2DR (0x0F5C)	UART2SR (0x0F5D)	RD2BUF (0x0F5E)	TD2BUF (0x0F5E)

Table 11.1 SFR Address Assignment

	Serial Data Input Pin	Serial Data Output Pin
UART0	RXD0 pin	TXD0 pin
UART1	RXD1 pin	TXD1 pin
UART2	RXD2 pin	TXD2 pin

Table 11.2 Pin Names

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

11.1Configuration

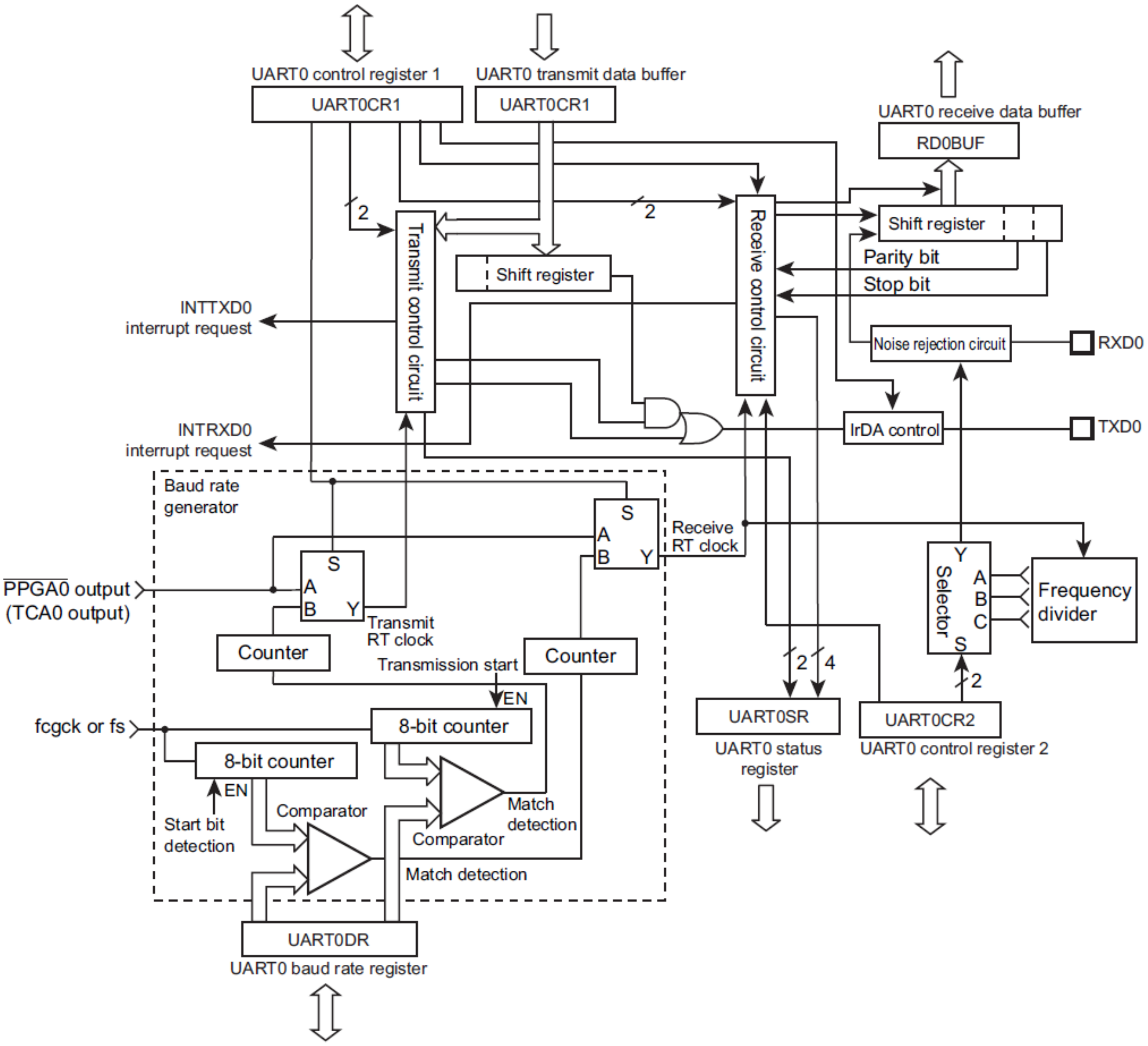


Figure 11.1 Asynchronous Serial Interface (UART)

11.2 Control

UART0 is controlled by the low power consumption registers (POFFCR1), UART0 control registers 1 and 2 (UART0CR1 and UART0CR2) and the UART0 baud rate register (UART0DR). The operating status can be monitored using the UART status register (UART0SR).

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Low Power Consumption Register 1

POFFCR1 (0x0F75)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 control	0: Disable 1: Enable
UART2EN	UART2 control	0: Disable 1: Enable
UART1EN	UART1 control	0: Disable 1: Enable
UART0EN	UART0 control	0: Disable 1: Enable

UART0 Control Register 1

UART0CR1 (0x001A)	7	6	5	4	3	2	1	0
Bit Symbol	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
After reset	0	0	0	0	0	0	0	0

<table>
<tr><td>TXE</td><td>Transmit operation</td><td colspan="3">0: Disable
1: Enable</td></tr>
<tr><td>RXE</td><td>Receive operation</td><td colspan="3">0: Disable
1: Enable</td></tr>
<tr><td>STOPBT</td><td>Transmit stop bit length</td><td colspan="3">0: 1 bit
1: 2 bits</td></tr>
<tr><td>EVEN</td><td>Parity selection</td><td colspan="3">0: Odd-numbered parity
1: Even number parity</td></tr>
<tr><td>PE</td><td>Parity addition</td><td colspan="3">0: No parity
1: Parity added</td></tr>
<tr><td>IRDASEL</td><td>TXD pin output selectin</td><td colspan="3">0: UART output
1: IrDA output</td></tr>
<tr><td rowspan="3">BRG</td><td rowspan="3">Transfer base clock selection</td><td></td><td>When SYSCR2<SYSCK> is "0"</td><td>When SYSCR2<SYSCK> is "1"</td></tr>
<tr><td>0</td><td>fcgck</td><td>fs</td></tr>
<tr><td>1</td><td colspan="2">TCA0 output</td></tr>
</table>

Note 1): fcgck, Gear clock; fs, Low-frequency clock

Note 2): If the TXE or RXE bit is set to "0" during the transmission or receiving of data, the operation is not disabled until the data transfer is completed. At this time, the data stored in the transmit data buffer is discarded.

Note 3): EVEN, PE and BRG settings are common to transmission and receiving.

Note 4): Set RXE and TXE to "0" before changing BRG.

Note 5): When BRG is set to the TCA0 output, the RT clock becomes asynchronous and the start bit of the transmitted/received data may get shorter by a maximum of (UART1DR+1)/(Transfer base clock frequency)[s]. If the pin is not used for the TCA0 output, control the TCA0 output by using the port function control register.

Note 6): To prevent STOPBT, EVEN, PE, IRDASEL and BRG from being changed accidentally during the UART communication, the *register cannot be rewritten during the UART operation. For details, refer to "11.3 Protection to Prevent UART1CR1 and UART1CR2 Registers from Being Changed ".*

Note 7): When the STOP, IDLE0 or SLEEP0 mode is activated, TXE and RXE are cleared to "0" and the UART stops. Other bits keep their values.

UART0 Control Register 2

UART0CR2 (0x001B)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	RTSEL			RXDNC		STOPBR
Read/Write	R	R	R/W			R/W		R/W
After reset	0	0	0	0	0	0	0	0

RTSEL	Selects the number of RT clocks		Odd-numbered bits of transfer frame	Even-numbered bits of transfer frame
		000	16 clocks	16 clocks
		001	16 clocks	17 clocks
		010	15 clocks	15 clocks
		011	15 clocks	16 clocks
		100	17 clocks	17 clocks
		101	Reserved	
		11*	Reserved	
RXDNC	Selects the RXD input noise rejection time (Time of pulses to be removed as noise)	00: No noise rejection 01: 1 x (UART1DR + 1) / (Transfer base clock frequency) [s] 10: 2 x (UART1DR + 1) / (Transfer base clock frequency) [s] 11: 4 x (UART1DR + 1) / (Transfer base clock frequency) [s]		
STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits		

Note 1): When a read instruction is executed on UART1CR2, bits 7 and 6 are read as "0".

Note 2): RTSEL can be set to two kinds of RT clocks for the even- and odd-numbered bits of the transfer frame. For details, refer to "11.7.1 Transfer baud rate calculation method".

Note 3): For details of the RXDNC noise rejection time, refer to "11.9 Received Data Noise Rejection".

Note 4): When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART1CR2 remains unchanged.

Note 5): When STOPBR is set to 2 bits, the first bit of the stop bits (during data receiving) is not checked for a framing error.

Note 6): To prevent RTSEL, RXDNC and STOPBR from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "11.3 Protection to Prevent UART1CR1 and UART1CR2 Registers from Being Changed ".

UART0 Baud Rate Register

UART0DR (0x001C)	7	6	5	4	3	2	1	0
Bit Symbol	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1): Set UART0CR1<RXE> and UART0CR1<TXE> to "0" before changing UART0DR. For the set values, refer to "11.7 Transfer Baud Rate".

Note 2): When UART0CR1<BRG> is set to the TCA0 output, the value set to UART0DR has no meaning.

Note 3): When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0DR remains unchanged.

UART0 Status Register

UART0SR (0x001D)	7	6	5	4	3	2	1	0
Bit Symbol	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

PERR	Parity error flag	0: No parity error 1: Parity error
FERR	Framing error flag	0: No framing error 1: Framing error
OERR	Overrun error flag	0: No overrun error 1: Overrun error
RBSY	Receive busy flag	0: Before receiving or end of receiving 1: On receiving
RBFL	Receive buffer full flag	0: Receive buffer empty 1: Receive buffer full
TBSY	Transmit busy flag	0: Before transmission or end of transmission 1: On transmission
TBFL	Transmit buffer full flag	0: Transmit buffer empty 1: Transmit buffer full

Note 1): TBFL is cleared to "0" automatically after an INTTXD1 interrupt request is generated, and is set to "1" when data is set to TD1BUF.

Note 2): When a read instruction is executed on UART1SR, bit 4 is read as "0".

Note 3): When the STOP, IDLE0 or SLEEP0 mode is activated, each bit of UART1SR is cleared to "0" and the UART stops.

UART0 Receive Data Register

RD0BUF (0x001E)	7	6	5	4	3	2	1	0
Bit Symbol	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Note): When the STOP, IDLE0 or SLEEP0 mode is activated, the RD1BUF values become undefined. If received data is required, read *it before activating the mode.*

UART0 Transmit Data Register

TD0BUF (0x001E)	7	6	5	4	3	2	1	0
Bit Symbol	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
Read/Write	W	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0

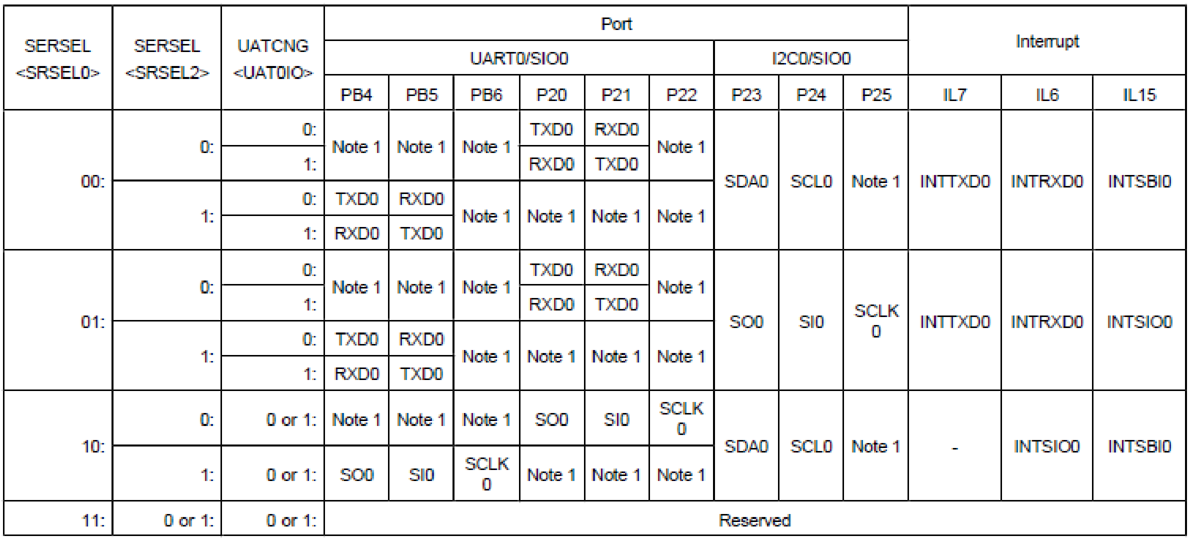
Note): When the STOP, IDLE0 or SLEEP0 mode is activated, the TD1BUF values become undefined.

UART Input/Output Change Control Register

UATCNG (0x0E57)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	UAT2IO	UAT1IO	UAT0IO
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

			RXD pin		TXD pin	
UAT2IO	Select UART2 input/ output port	0:	P27		P26	
		1:	P26		P27	
UAT1IO	Select UART1 input/ output port	0:	P91		P90	
		1:	P90		P91	
UAT0IO	Select UART0 input/ output port		SERSEL <SERSEL2>="0"	SERSEL <SERSEL2>="1"	SERSEL <SERSEL2>="0"	SERSEL <SERSEL2>="1"
		0:	P21	PB5	P20	PB4
		1:	P20	PB4	P21	PB5

Note 1: The operation for changing UATCNG must be executed while the applicable serial interface operations are stopped.



SERSEL <SRSEL0>	SERSEL <SRSEL2>	UATCNG <UAT0IO>	Port									Interrupt		
			UART0/SIO0						I2C0/SIO0					
			PB4	PB5	PB6	P20	P21	P22	P23	P24	P25	IL7	IL6	IL15
00:	0:	0:	Note 1	Note 1	Note 1	TXD0	RXD0	Note 1	SDA0	SCL0	Note 1	INTTXD0	INTRXD0	INTSBI0
		1:				RXD0	TXD0							
	1:	0:	TXD0	RXD0	Note 1	Note 1	Note 1	Note 1						
		1:	RXD0	TXD0										
01:	0:	0:	Note 1	Note 1	Note 1	TXD0	RXD0	Note 1	SO0	SI0	SCLK 0	INTTXD0	INTRXD0	INTSIO0
		1:				RXD0	TXD0							
	1:	0:	TXD0	RXD0	Note 1	Note 1	Note 1	Note 1						
		1:	RXD0	TXD0										
10:	0:	0 or 1:	Note 1	Note 1	Note 1	SO0	SI0	SCLK 0	SDA0	SCL0	Note 1	-	INTSIO0	INTSBI0
	1:	0 or 1:	SO0	SI0	SCLK 0	Note 1	Note 1	Note 1						
11:	0 or 1:	0 or 1:	Reserved											

Note 1: Can be used as a port. (Set the function register (PxFC) to "0".)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

11.3 Low Power Consumption Function

UART0 has a low power consumption register (POFFCR1) that saves power consumption when the UART function is not used.

Setting POFFCR1 <UART0EN> to "0" disables the basic clock supply to UART0 to save power. Note that this renders the UART unusable. Setting POFFCR1 <UART0EN> to "1" enables the basic clock supply to UART0 and renders the UART usable.

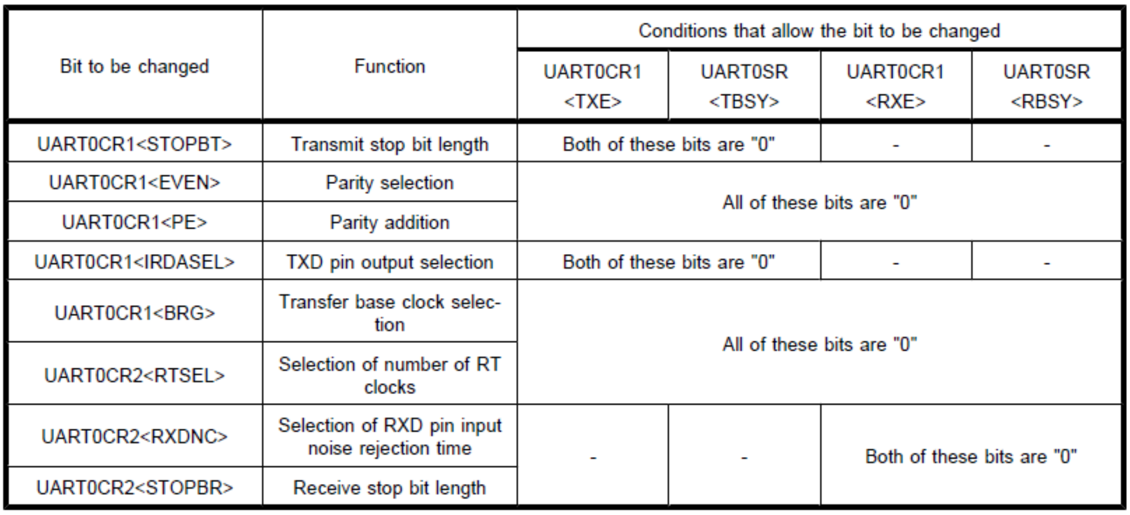
After reset, POFFCR1 <UART0EN> is initialized to "0", and this renders the UART unusable. When using the UART for the first time, be sure to set POFFCR1 <UART0EN> to "1" in the initial setting of the program (before the UART control register is operated).

Do not change POFFCR1 <UART0EN> to "0" during the UART operation, otherwise UART0 may operate unexpectedly.

11.4 Protection of UART0CR1 and UART0CR2 Registers from Being Changed

MQ6935 has a function that protects the registers from being changed so that the UART communication settings (for example, stop bit and parity) are not changed accidentally during the UART operation.

Specific bits of UART0CR1 and UART0CR2 can be changed only under the conditions shown in Table 11.3. If a write instruction is executed on the register when it is protected from being changed, the bits remain unchanged and keep their previous values.



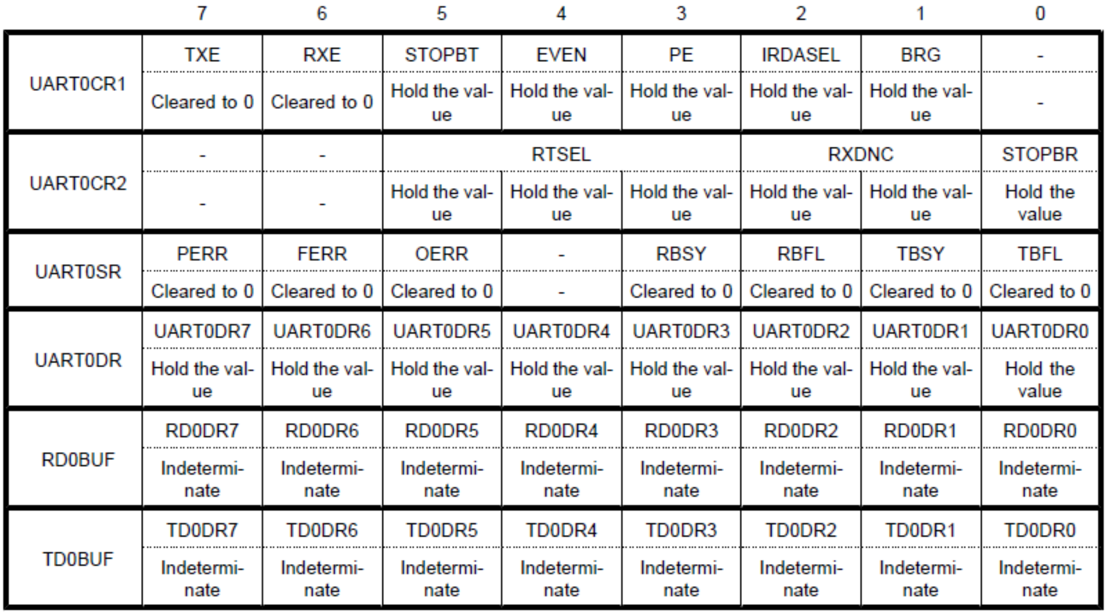
Bit to be changed	Function	Conditions that allow the bit to be changed			
		UART0CR1 <TXE>	UART0SR <TBSY>	UART0CR1 <RXE>	UART0SR <RBSY>
UART0CR1<STOPBT>	Transmit stop bit length	Both of these bits are "0"		-	-
UART0CR1<EVEN>	Parity selection	All of these bits are "0"			
UART0CR1<PE>	Parity addition				
UART0CR1<IRDASEL>	TXD pin output selection	Both of these bits are "0"		-	-
UART0CR1<BRG>	Transfer base clock selection	All of these bits are "0"			
UART0CR2<RTSEL>	Selection of number of RT clocks				
UART0CR2<RXDNC>	Selection of RXD pin input noise rejection time	-	-	Both of these bits are "0"	
UART0CR2<STOPBR>	Receive stop bit length				

Table 11.3 Changing of UART0CR1 and UART0CR2

11.5 Activation of STOP, IDLE0 or SLEEP0 Mode

11.5.1 Transition of Register Status

When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically and each register becomes the status as shown in Table 11.4. For the registers that do not hold their values, make settings again as needed after the operation mode is recovered.

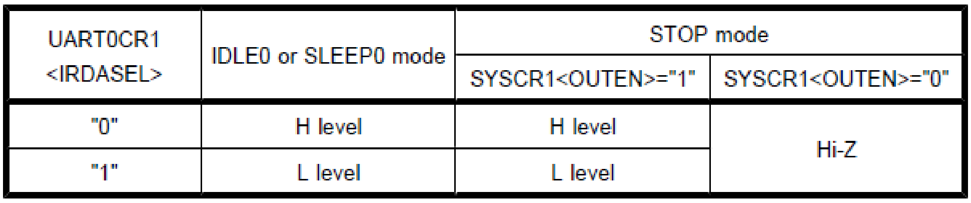


	7	6	5	4	3	2	1	0
UART0CR1	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
	Cleared to 0	Cleared to 0	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	-
UART0CR2	-	-	RTSEL			RXDNC		STOPBR
	-	-	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
UART0SR	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
	Cleared to 0	Cleared to 0	Cleared to 0	-	Cleared to 0	Cleared to 0	Cleared to 0	Cleared to 0
UART0DR	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
RD0BUF	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate
TD0BUF	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate

Table 11.4 Transition of Register Status

11.5.2 Transition of TXD Pin Status

When the IDLE0, SLEEP0 or STOP mode is activated, the TXD pin reverts to the status shown in Table 11.5, whether data is transmitted/received or the operation is stopped.



UART0CR1 <IRDASEL>	IDLE0 or SLEEP0 mode	STOP mode	
		SYSCR1<OUTEN>="1"	SYSCR1<OUTEN>="0"
"0"	H level	H level	Hi-Z
"1"	L level	L level	

Table 11.5 TXD Pin Status When the STOP, IDLE0 or SLEEP0 Mode Is Activated

11.6 Transfer Data Format

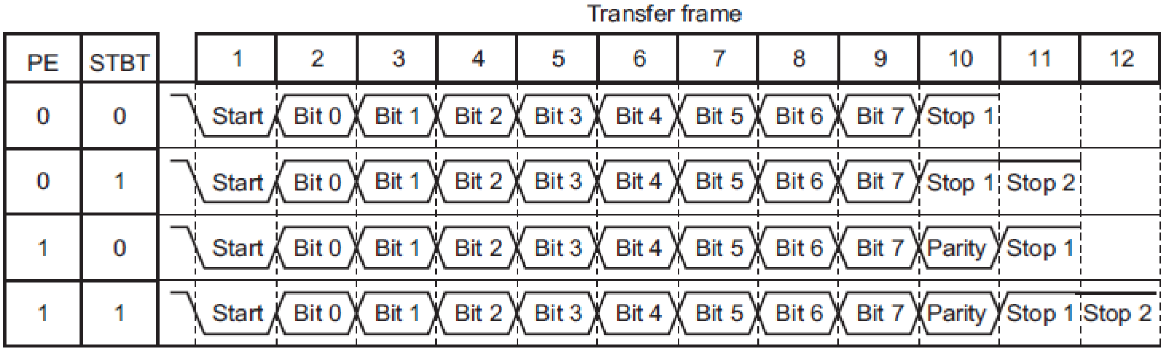
The UART transfers data composed of the following four elements. The data from the start bit to the stop

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

bit is collectively defined as a "transfer frame". The start bit consists of 1 bit (L level) and the data consists of 8 bits. Parity bits are determined by UART1CR1 <PE> that selects the presence or absence of parity and UART1CR1 <EVEN> that selects even- or odd-numbered parity. The bit length of the stop bit can be selected at UART1CR1 <STBT>.

Figure 11.1 shows the transfer data format.

- Start bit (1 bit)
- Data (8 bits)
- Parity bit (selectable from even-numbered, odd-numbered or no parity)
- Stop bit (selectable from 1 bit or 2 bits)



Transfer frame

PE	STBT	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

Figure 11.2 Transfer Data Format

11.7 Infrared Data Format Transfer Mode

The TXD1 pin can output data in the infrared data format (IrDA) by the setting of the IrDA output control register. Setting UART1CR1 <IRDASEL> to "1" allows the TXD1 pin to output data in the infrared data format.

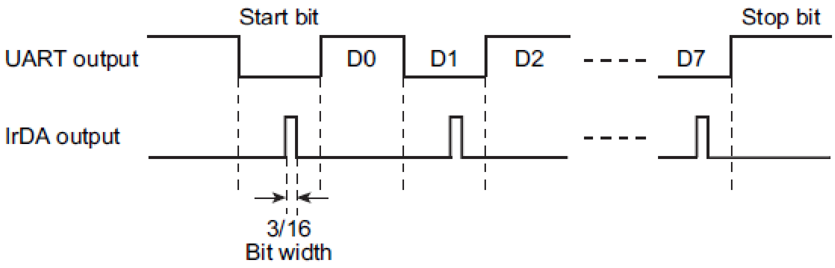


Figure 11.3 Example of Infrared Data Format (Comparison between Normal Output and Ir-DA Output)

11.8 Transfer Baud Rate

The transfer baud rate of UART is set by UART1CR1 <BRG>, UART1DR and UART1CR2<RTSEL>. The settings of UART1DR and UART1CR2 <RTSEL> for general baud rates and operating frequencies are shown below. For independent calculation of transfer baud rates, refer to "11.8.1 Transfer baud rate calculation method".

Basic baud rate[baud]	Register	Operating frequency				
		16MHz	8MHz	4MHz	2MHz	1MHz
128000	UART0DR[7:0]	0x07	0x03	0x01	0x00	-
	RTSEL[2:0]	0y011	0y011	0y011	0y011	-
	Error	(+0.81%)	(+0.81%)	(+0.81%)	(+0.81%)	-
115200	UART0DR[7:0]	0x08	0x03	0x01	0x00	-
	RTSEL[2:0]	0y011	0y100	0y100	0y100	-
	Error	(-0.44%)	(+2.12%)	(+2.12%)	(+2.12%)	-
76800	UART0DR[7:0]	0x0C	0x06	0x02	-	-
	RTSEL[2:0]	0y000	0y010	0y100	-	-
	Error	(+0.16%)	(-0.79%)	(+2.12%)	-	-
62500	UART0DR[7:0]	0x0F	0x07	0x03	0x01	0x00
	RTSEL[2:0]	0y000	0y000	0y000	0y000	0y000
	Error	0%	0%	0%	0%	0%
57600	UART0DR[7:0]	0x11	0x08	0x03	0x01	0x00
	RTSEL[2:0]	0y011	0y011	0y100	0y100	0y100
	Error	(-0.44%)	(-0.44%)	(+2.12%)	(+2.12%)	(+2.12%)
38400	UART0DR[7:0]	0x19	0x0C	0x06	0x02	-
	RTSEL[2:0]	0y000	0y000	0y010	0y100	-
	Error	(+0.16%)	(+0.16%)	(-0.79%)	(+2.12%)	-
19200	UART0DR[7:0]	0x30	0x19	0x0C	0x06	0x02
	RTSEL[2:0]	0y100	0y000	0y000	0y010	0y100
	Error	(+0.04%)	(+0.16%)	(+0.16%)	(-0.79%)	(+2.12%)
9600	UART0DR[7:0]	0x64	0x33	0x19	0x0C	0x06
	RTSEL[2:0]	0y001	0y000	0y000	0y000	0y010
	Error	(+0.01%)	(+0.16%)	(+0.16%)	(+0.16%)	(-0.79%)
4800	UART0DR[7:0]	0xC9	0x67	0x33	0x19	0x0C
	RTSEL[2:0]	0y001	0y000	0y000	0y000	0y000
	Error	(+0.01%)	(+0.16%)	(+0.16%)	(+0.16%)	(+0.16%)
2400	UART0DR[7:0]	-	0xCF	0x67	0x33	0x19
	RTSEL[2:0]	-	0y000	0y000	0y000	0y000
	Error	-	(+0.16%)	(+0.16%)	(+0.16%)	(+0.16%)
1200	UART0DR[7:0]	-	-	0xCF	0x67	0x33
	RTSEL[2:0]	-	-	0y000	0y000	0y000
	Error	-	-	(+0.16%)	(+0.16%)	(+0.16%)

Table 11.6 Set Values of UART0DR and UART0CR2<RTSEL> for Transfer Baud Rates (fcgck=10 to 1MHz, UART0CR2 <RXDNC>=0y00)

Basic baud rate[baud]	Register	Operating frequency
		32.768kHz
300	UART0DR[7:0]	0x06
	RTSEL[2:0]	0y011
	Error	(+0.67%)
150	UART0DR[7:0]	0x0D
	RTSEL[2:0]	0y011
	Error	(+0.67%)
134	UART0DR[7:0]	0x0E
	RTSEL[2:0]	0y001
	Error	(-1.20%)
110	UART0DR[7:0]	0x11
	RTSEL[2:0]	0y001
	Error	(+0.30%)
75	UART0DR[7:0]	0x1C
	RTSEL[2:0]	0y010
	Error	(+0.44%)

Table 11.7 Set Values of UART0DR and UART0CR2<RTSEL> for Transfer Baud Rates (fs=32.768 kHz, UART0CR2<RXDNC>=0y00)

Note 1: The overall error from the basic baud rate must be within ±3%. Even if the overall error is within ±3%, the communication may fail due to factors such as frequency errors in external controllers (for example, a personal computer) and oscillators and the load capacity of the communication pin.

11.8.1 Transfer Baud Rate Calculation Method

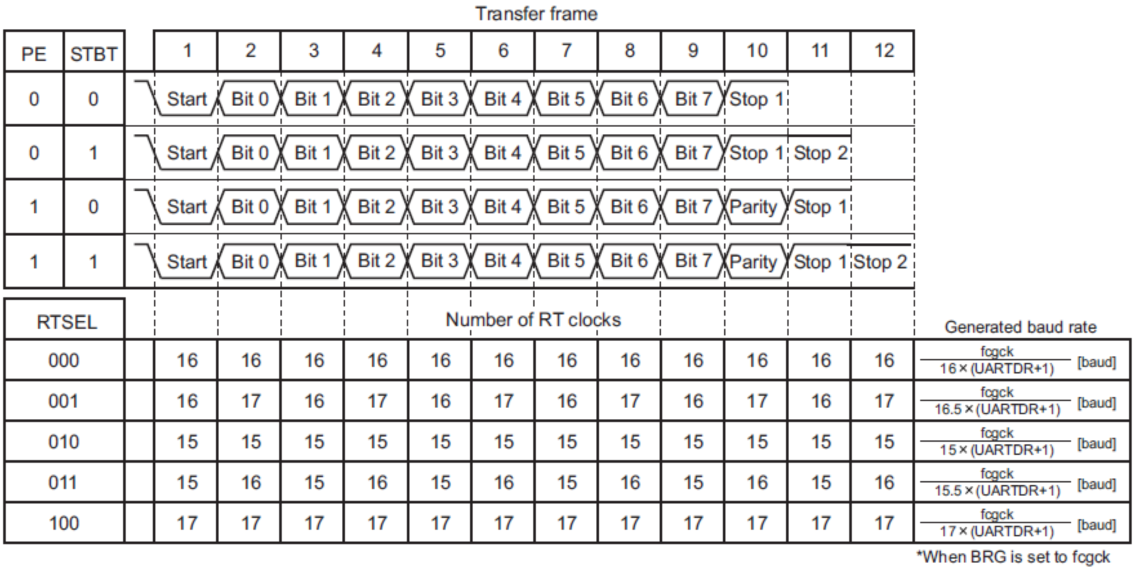
11.8.1.1 Bit width adjustment using UART0CR2<RTSEL>

The bit width of transmitted/received data can be finely adjusted by changing UART1CR2 <RTSEL>. The number of RT clocks per bit can be changed in a range of 15 to 17 clocks by changing UART1CR2<RTSEL>. The RT clock is the transfer base clock, which is the pulses obtained by counting the clock selected at UART1CR1<BRG> the number of times of (UART1DR set value) + 1. Especially, when UART1CR2 <RTSEL> is set to "0y001" or "0y011", two types of RT clocks alternate at each bit, so that the pseudo baud rates of RT × 15.5 clocks and RT × 16.5 clocks can be generated. The number of RT clocks per bit of transfer frame is shown in Figure 11.3.

For example, when fcgck is 4 [MHz], UART1CR2<RTSEL> is set to "0y000" and UART1DR is set to "0x19", the baud rate calculated using the formula in Figure 11.3 is expressed as:fcgck / (16 × (UART1DR + 1)

= 9615 [baud]

These settings generate a baud rate close to 9600 [baud] (+0.16%).



Transfer frame

PE	STBT	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

Number of RT clocks

RTSEL	1	2	3	4	5	6	7	8	9	10	11	12	Generated baud rate
000	16	16	16	16	16	16	16	16	16	16	16	16	$\frac{fcgck}{16\times(UARTDR+1)}$ [baud]
001	16	17	16	17	16	17	16	17	16	17	16	17	$\frac{fcgck}{16.5\times(UARTDR+1)}$ [baud]
010	15	15	15	15	15	15	15	15	15	15	15	15	$\frac{fcgck}{15\times(UARTDR+1)}$ [baud]
011	15	16	15	16	15	16	15	16	15	16	15	16	$\frac{fcgck}{15.5\times(UARTDR+1)}$ [baud]
100	17	17	17	17	17	17	17	17	17	17	17	17	$\frac{fcgck}{17\times(UARTDR+1)}$ [baud]

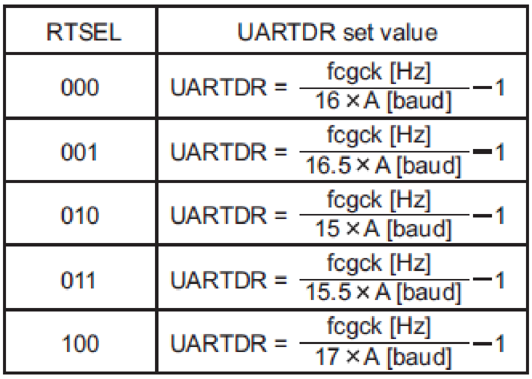
*When BRG is set to fcgck

Figure 11.4 Fine Adjustment of Baud Rate Clock Using UART11R2 <RTSEL>

11.8.1.2 Calculation of Set Values of UART0CR2 <RTSEL> and UART0DR

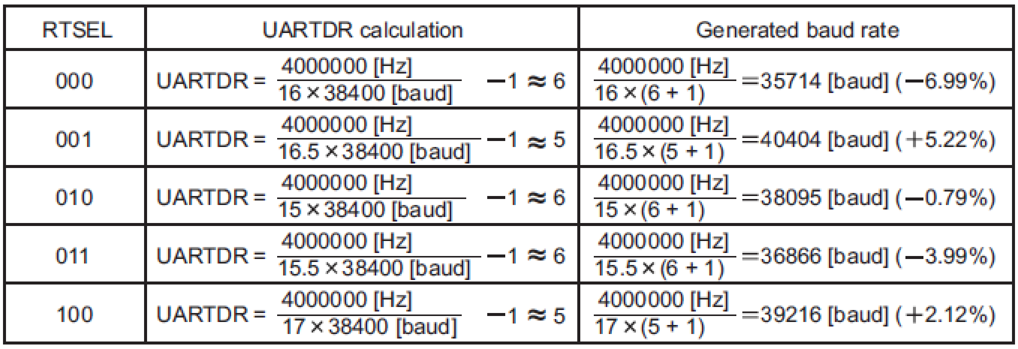
The set value of UART0DR for an operating frequency and baud rate can be calculated using the calculation formula shown in Figure 11.4. For example, to generate a basic baud rate of 38400 [baud] with fcgck=4 [MHz], calculate the set value of UART0DR for each setting of UART0CR2 <RTSEL> and compensate the calculated value to a positive number to obtain the generated baud rate as shown in Figure 11.5. Basically, select the set value of UART0CR2 <RTSEL> that has the smallest baud rate error from among the generated baud rates. In Figure 11.5, the setting of UART0CR2 <RTSEL>="0y010" has the smallest error among the calculated baud rates, and thus the generated baud rate is 38095 [baud] (−0.79%) against the basic baud rate of 38400 [baud].

Note): The error from the basic baud rate should be accurate to within ±3%. Even if the error is within ±3%, the communication may fail due to factors such as frequency errors of external controllers (for example, a personal computer) and oscillators and the load capacity of the communication pin.



RTSEL	UARTDR set value
000	$\text{UARTDR} = \frac{\text{fcgck [Hz]}}{16 \times \text{A [baud]}} - 1$
001	$\text{UARTDR} = \frac{\text{fcgck [Hz]}}{16.5 \times \text{A [baud]}} - 1$
010	$\text{UARTDR} = \frac{\text{fcgck [Hz]}}{15 \times \text{A [baud]}} - 1$
011	$\text{UARTDR} = \frac{\text{fcgck [Hz]}}{15.5 \times \text{A [baud]}} - 1$
100	$\text{UARTDR} = \frac{\text{fcgck [Hz]}}{17 \times \text{A [baud]}} - 1$

Table 11.7 UART0DR Calculation Method (When BRG Is Set to fcgck)



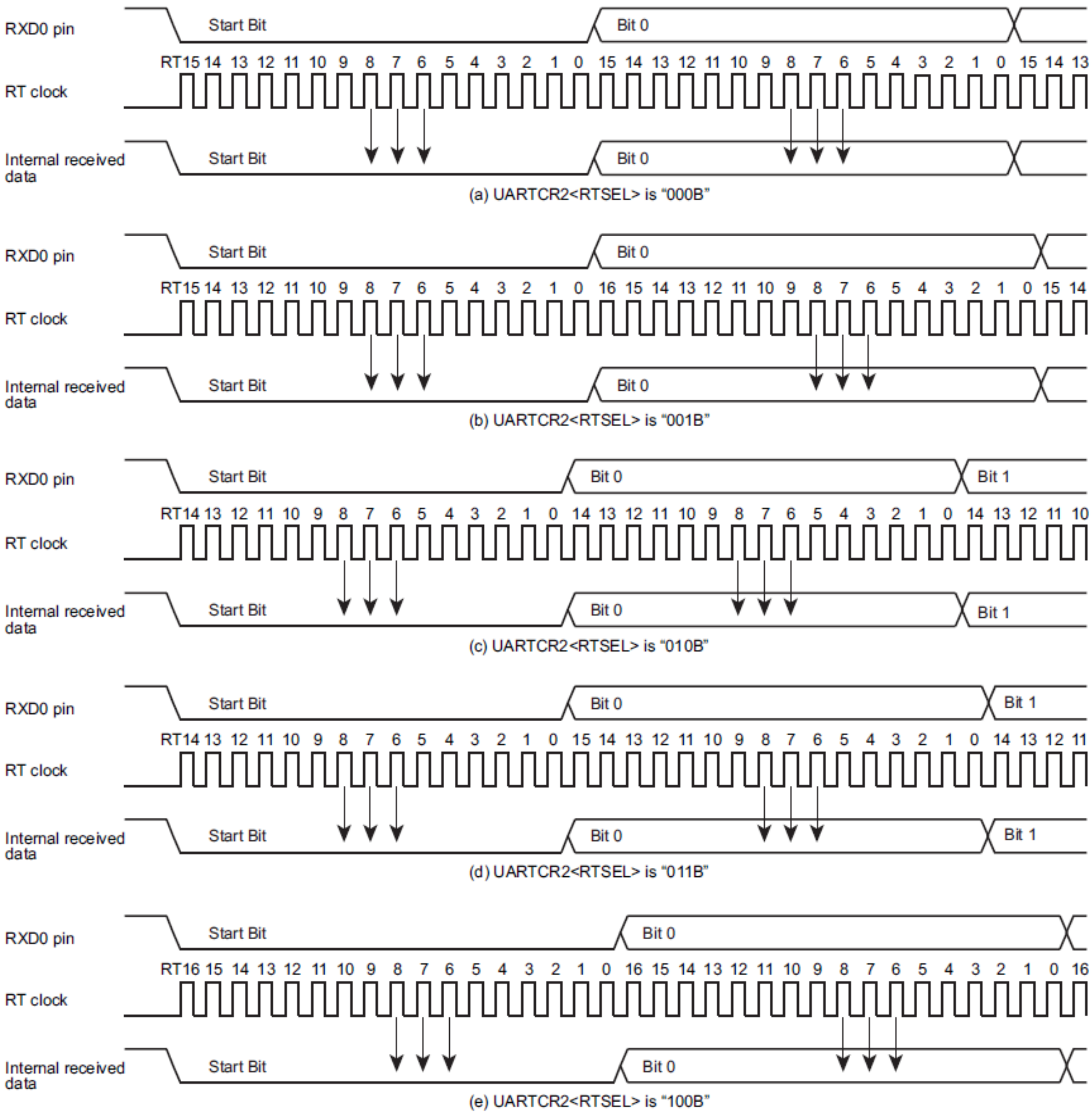
RTSEL	UARTDR calculation	Generated baud rate
000	$\text{UARTDR} = \frac{4000000\text{ [Hz]}}{16 \times 38400\text{ [baud]}} - 1 \approx 6$	$\frac{4000000\text{ [Hz]}}{16 \times (6+1)} = 35714\text{ [baud] } (-6.99\%)$
001	$\text{UARTDR} = \frac{4000000\text{ [Hz]}}{16.5 \times 38400\text{ [baud]}} - 1 \approx 5$	$\frac{4000000\text{ [Hz]}}{16.5 \times (5+1)} = 40404\text{ [baud] } (+5.22\%)$
010	$\text{UARTDR} = \frac{4000000\text{ [Hz]}}{15 \times 38400\text{ [baud]}} - 1 \approx 6$	$\frac{4000000\text{ [Hz]}}{15 \times (6+1)} = 38095\text{ [baud] } (-0.79\%)$
011	$\text{UARTDR} = \frac{4000000\text{ [Hz]}}{15.5 \times 38400\text{ [baud]}} - 1 \approx 6$	$\frac{4000000\text{ [Hz]}}{15.5 \times (6+1)} = 36866\text{ [baud] } (-3.99\%)$
100	$\text{UARTDR} = \frac{4000000\text{ [Hz]}}{17 \times 38400\text{ [baud]}} - 1 \approx 5$	$\frac{4000000\text{ [Hz]}}{17 \times (5+1)} = 39216\text{ [baud] } (+2.12\%)$

Table 11.8 Example of UART0DR Calculation

11.9 Data Sampling Method

The UART receive control circuit starts RT clock counting when it detects a falling edge of the input pulses to the RXD1 pin. 15 to 17 RT clocks are counted per bit and each clock is expressed as RTn (n=16 to 0). In a bit that has 17 RT clocks, RT16 to RT0 are counted. In a bit that has 16 RT clocks, RT15 to RT0 are counted. In a bit that has 15 RT clocks, RT14 to RT0 are counted (Decrement). During counting of RT8 to RT6, the UART receive control circuit samples the input pulses to the RXD1 pin to make a majority decision. The same level detected twice or more from among three samplings is processed as the data for the bit.

The number of RT clocks can be changed in a range of 15 to 17 by setting UART1CR2 <RTSEL>. However, sampling is always executed in RT8 to RT6, even if the number of RT clocks is changed (Figure 11.6).


RXD0 pin
Start Bit
Bit 0
Bit 1
RT clock
RT15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13
RT15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14
RT14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 14 13 12 11 10
RT14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 14 13 12 11
RT16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 16
Internal received data

(a) UARTCR2<RTSEL> is "000B"

(b) UARTCR2<RTSEL> is "001B"

(c) UARTCR2<RTSEL> is "010B"

(d) UARTCR2<RTSEL> is "011B"

(e) UARTCR2<RTSEL> is "100B"

Figure 11.5 Data Sampling in Each Case of UARTCR2 <RTSEL>

If "1" is detected in sampling of the start bit, for example, due to the influence of noise, RT clock counting stops and the data receiving is suspended. Subsequently, when a falling edge is detected in the input pulses to the RXD1 pin, RT clock counting restarts and the data receiving restarts with the start bit.

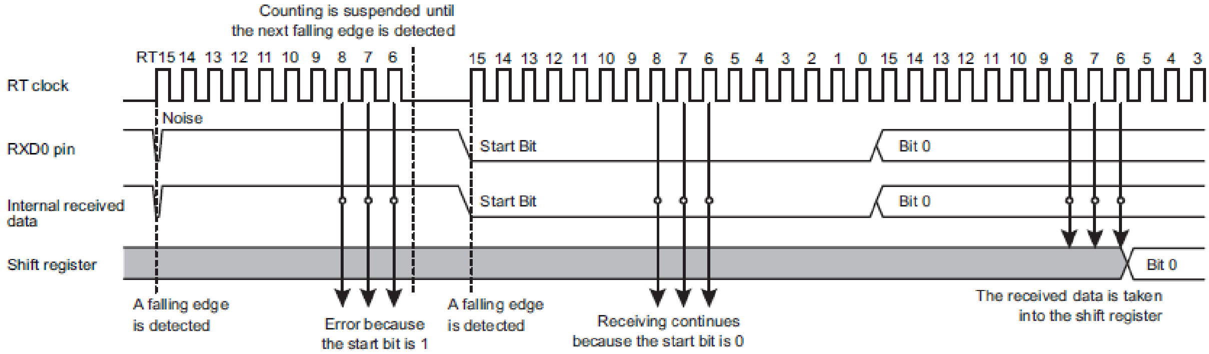

Counting is suspended until the next falling edge is detected
RT 15 14 13 12 11 10 9 8 7 6
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8 7 6 5 4 3
RT clock
Noise
RXD0 pin
Start Bit
Bit 0
Internal received data
Start Bit
Bit 0
Shift register
Bit 0
A falling edge is detected
Error because the start bit is 1
A falling edge is detected
Receiving continues because the start bit is 0
The received data is taken into the shift register

Figure 11.6 Start Bit Sampling

11.10Received Data Noise Rejection

When noise rejection is enabled at UART1CR2 <RXDNC>, the time of pulses to be regarded as signals is as shown in Table as below.

RXDNC	Noise rejection time [s]	Time of pulses to be regarded as signals
00	No noise rejection	-
01	(UART0DR+1)/(Transfer base clock frequency)	2 × (UART0DR+1)/(Transfer base clock frequency)
10	2 × (UART0DR+1)/(Transfer base clock frequency)	4 × (UART0DR+1)/(Transfer base clock frequency)
11	4 × (UART0DR+1)/(Transfer base clock frequency)	8 × (UART0DR+1)/(Transfer base clock frequency)

Table 11.9 Received Data Noise Rejection Time

Note): The transfer base clock frequency is the clock frequency selected at UARTCR1 <BRG>.

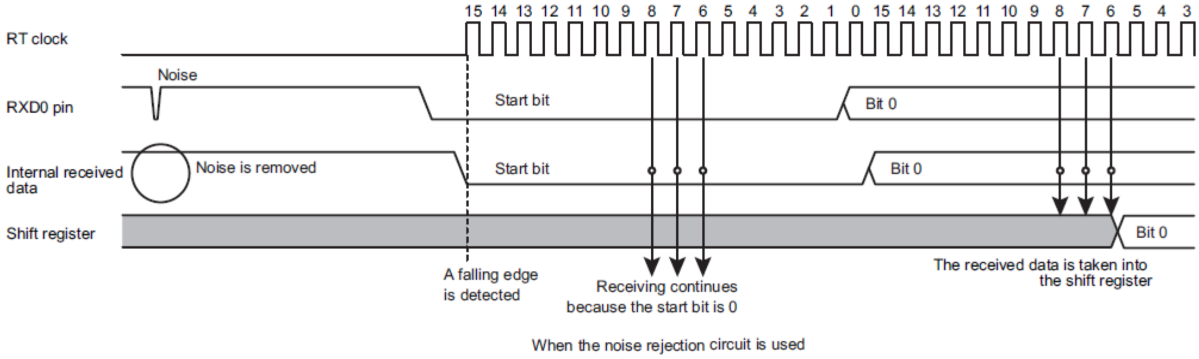


Figure 11.7 Received Data Noise Rejection

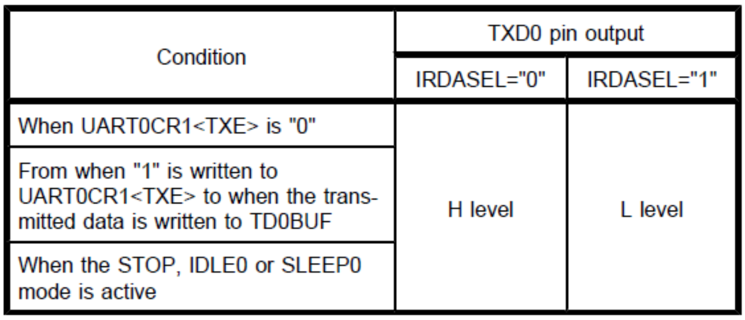
iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

11.11 Transmit/Receive Operation

11.11.1 Data Transmit Operation

Set UART0CR1 <TXE> to "1". Check UART1SR <TBFL> = "0", and then write data into TD0BUF (transmit data buffer). Writing data into TD0BUF sets UART0SR<TBFL> to "1", transfers the data to the transmit shift register, and outputs the data sequentially from the TXD0 pin. The data output includes a start bit, stop bits whose number is specified in UART0CR1 <STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART0CR1 <BRG>, UART0CR2 <RTSEL> and UART0DR. When data transmission starts, the transmit buffer full flag UART0SR <TBFL> is cleared to "0" and an INTTXD0 interrupt request is generated.

Note 1): After data is written into TD0BUF, if new data is written into TD0BUF before the previous data is transferred to the shift register, the new data is written over the previous data and is transferred to the shift register.
Note 2): Under the conditions shown in Table 11.7, the TXD0 pin output is fixed at the L or H level according to the setting of UART0CR1 <IRDASEL>.



Condition	TXD0 pin output	
	IRDASEL="0"	IRDASEL="1"
When UART0CR1<TXE> is "0"	H level	L level
From when "1" is written to UART0CR1<TXE> to when the transmitted data is written to TD0BUF		
When the STOP, IDLE0 or SLEEP0 mode is active		

Table 11.10 TXD0 Pin Output

11.11.2 Data Receive Operation

Set UART0CR1 <RXE> to "1". When data is received via the RXD0 pin, the received data is transferred to RD0BUF (receive data buffer). At this time, the transmitted data includes a start bit, stop bit(s) and a parity bit if parity addition is specified. When the stop bit(s) are received, data only is extracted and transferred to RD0BUF (receive data buffer). Then the receive buffer full flag UART0SR <RBFL> is set and an INTRXD0 interrupt request is generated. Set the data transfer baud rate using UART0CR1 <BRG>, UART0CR2 <RTSEL> and UART0DR.

If an overrun error occurs when data is received, the data is not transferred to RD0BUF (receive data buffer) but discarded; data in the RD0BUF is not affected.

11.12 Status Flag

11.12.1 Parity Error

When the parity determined using the receive data bits differs from the received parity bit, the parity error flag UART1SR <PERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR <PERR> is "1" when UART0SR is read, UART0SR <PERR> will be cleared to "0" when RD0BUF is read subsequently. (The RD0BUF read value becomes undefined.)

If UART0SR <PERR> is set to "1" after UART0SR is read, UART0SR <PERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <PERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

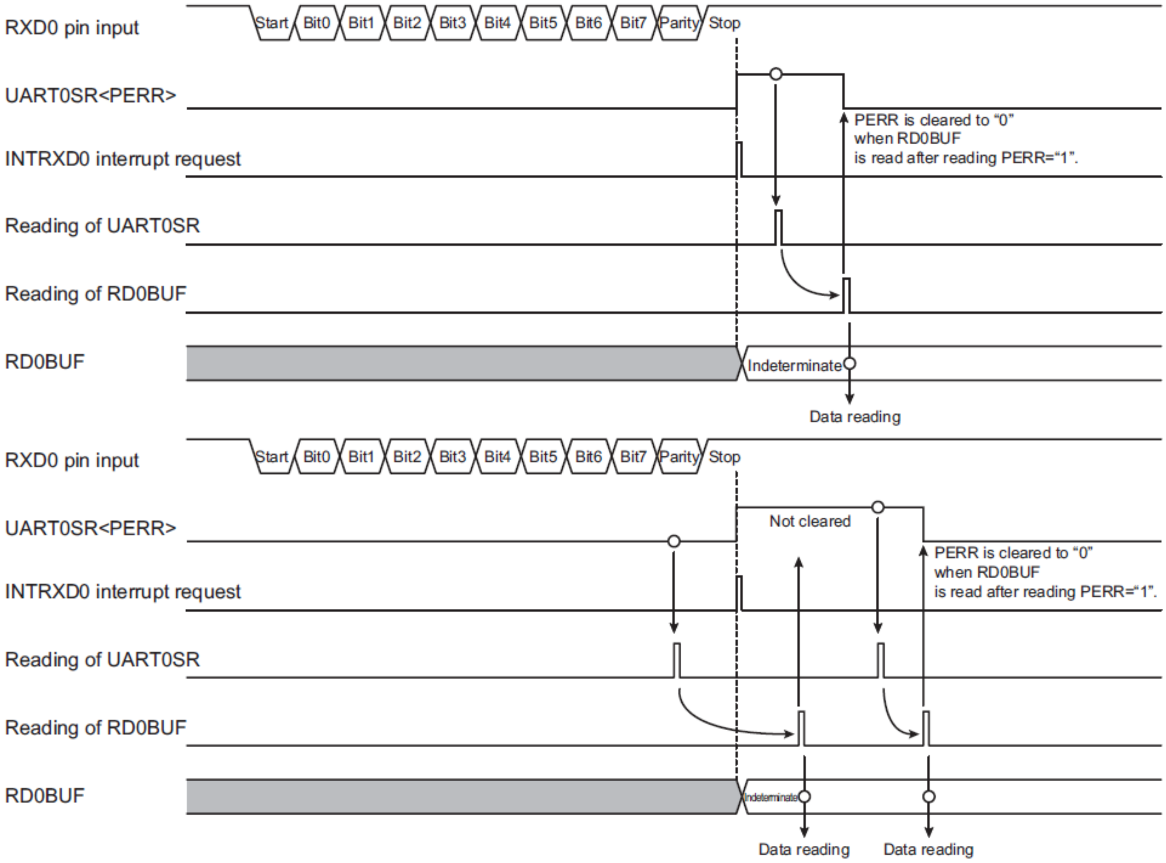


Figure 11.8 Occurrence of Parity Error

11.12.2 Framing Error

If the internal and external baud rates differ or "0" is sampled as the stop bit of received data due to the influence of noise on the RXD0 pin, the framing error flag UART0SR <FERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

If UART0SR <FERR> is "1" when UART0SR is read, UART0SR <FERR> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR <FERR> is set to "1" after UART0SR is read, UART0SR <FERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <FERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

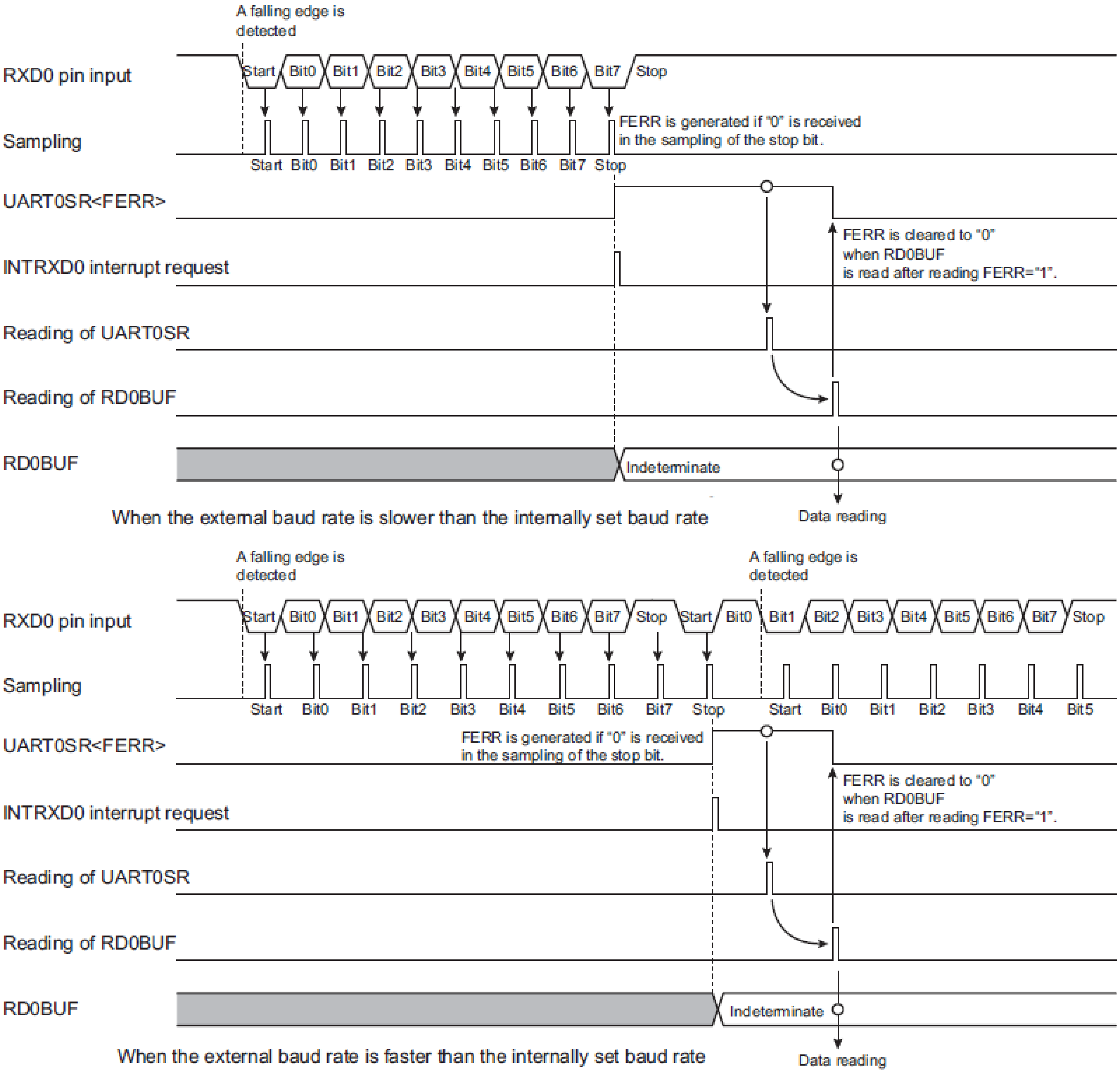

A falling edge is detected
RXD0 pin input
Start Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Stop
Sampling
FERR is generated if "0" is received in the sampling of the stop bit.
UART0SR<FERR>
FERR is cleared to "0" when RD0BUF is read after reading FERR="1".
INTRXD0 interrupt request
Reading of UART0SR
Reading of RD0BUF
RD0BUF
Indeterminate
Data reading
When the external baud rate is slower than the internally set baud rate
A falling edge is detected
A falling edge is detected
RXD0 pin input
Sampling
UART0SR<FERR>
FERR is generated if "0" is received in the sampling of the stop bit.
FERR is cleared to "0" when RD0BUF is read after reading FERR="1".
INTRXD0 interrupt request
Reading of UART0SR
Reading of RD0BUF
RD0BUF
Indeterminate
Data reading
When the external baud rate is faster than the internally set baud rate

Figure 11.9 Occurrence of Framing Error

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

11.12.3 Overrun Error

If receiving of all data bits is completed before the previous received data is read from RD0BUF, the overrun error flag UART0SR <OERR> is set to "1" and an INTRXD0 interrupt request is generated. The data received at the occurrence of the overrun error is discarded and the previous received data is maintained. Subsequently, if data is received while UART0SR <OERR> is still "1", no INTRXD0 interrupt request is generated, and the received data is discarded.

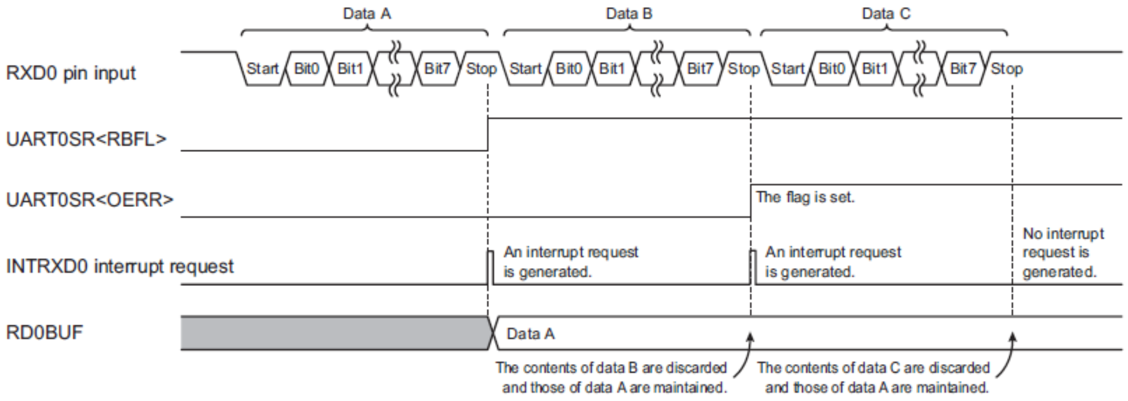
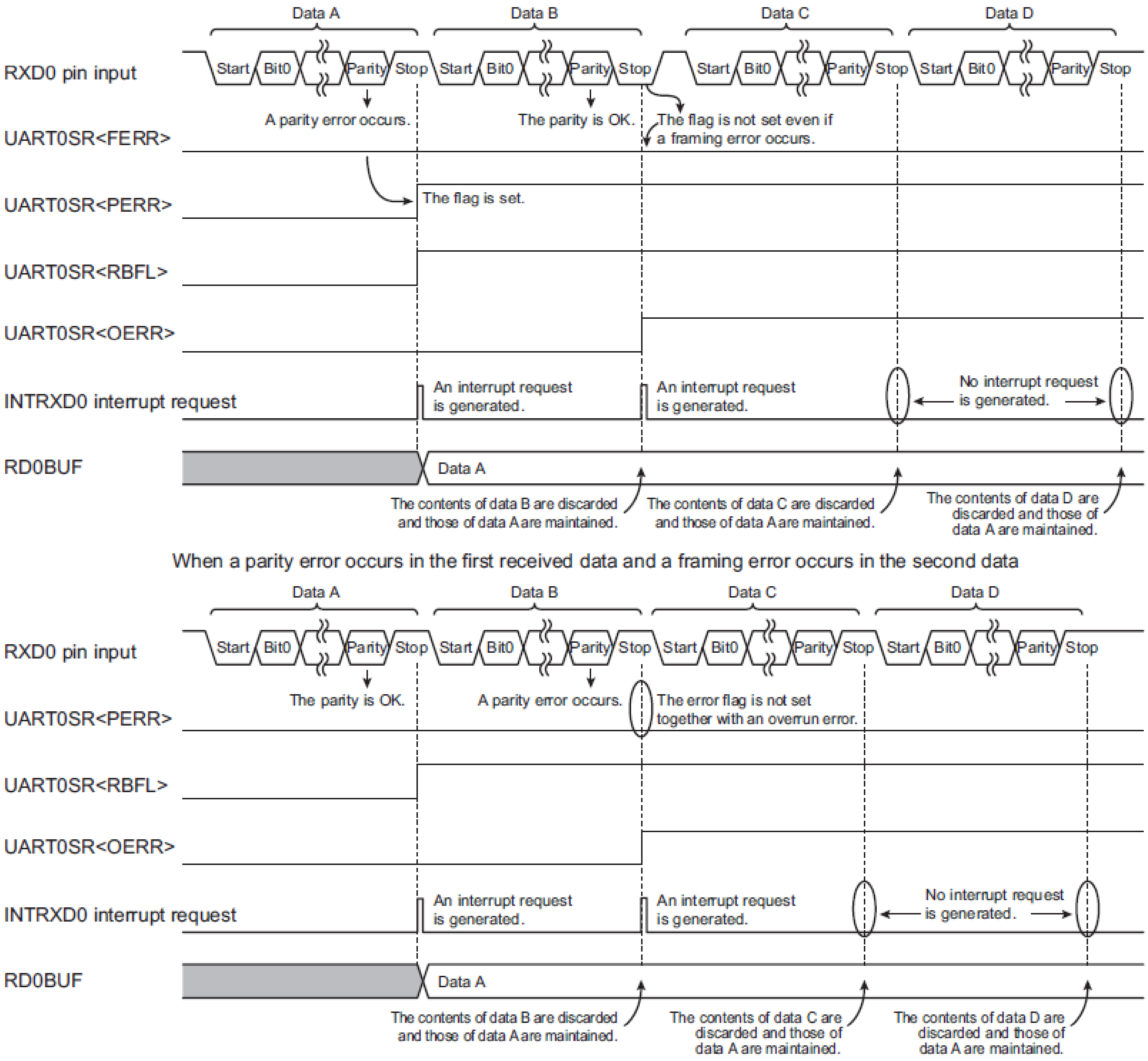

Data A
Data B
Data C
RXD0 pin input
Start
Bit0
Bit1
Bit7
Stop
UART0SR<RBFL>
UART0SR<OERR>
The flag is set.
INTRXD0 interrupt request
An interrupt request is generated.
An interrupt request is generated.
No interrupt request is generated.
RD0BUF
Data A
The contents of data B are discarded and those of data A are maintained.
The contents of data C are discarded and those of data A are maintained.

Figure 11.10 Generation of INTRXD0 Interrupt Request

Note that parity or framing errors in the discarded received data cannot be detected. (These error flags are not set.) That is to say, if these errors are detected together with an overrun error during the reading of UART0SR, they have occurred in the previous received data (the data stored in RD0BUF).

If UART0SR <OERR> is "1" when UART0SR is read, UART0SR <OERR> will be cleared to "0" whenRD1BUF is read subsequently.

If UART0SR <OERR> is set to "1" after UART0SR is read, UART0SR <OERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <OERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.


Data A
Data B
Data C
Data D
RXD0 pin input
Start
Bit0
Parity
Stop
A parity error occurs.
The parity is OK.
The flag is not set even if a framing error occurs.
UART0SR<FERR>
UART0SR<PERR>
The flag is set.
UART0SR<RBFL>
UART0SR<OERR>
INTRXD0 interrupt request
An interrupt request is generated.
An interrupt request is generated.
No interrupt request is generated.
RD0BUF
Data A
The contents of data B are discarded and those of data A are maintained.
The contents of data C are discarded and those of data A are maintained.
The contents of data D are discarded and those of data A are maintained.

When a parity error occurs in the first received data and a framing error occurs in the second data

Data A
Data B
Data C
Data D
RXD0 pin input
Start
Bit0
Parity
Stop
The parity is OK.
A parity error occurs.
The error flag is not set together with an overrun error.
UART0SR<PERR>
UART0SR<RBFL>
UART0SR<OERR>
INTRXD0 interrupt request
An interrupt request is generated.
An interrupt request is generated.
No interrupt request is generated.
RD0BUF
Data A
The contents of data B are discarded and those of data A are maintained.
The contents of data C are discarded and those of data A are maintained.
The contents of data D are discarded and those of data A are maintained.

When a parity error occurs in the second received data

Figure 11.11 Framing/Parity Error Flags When an Overrun Error Occurs

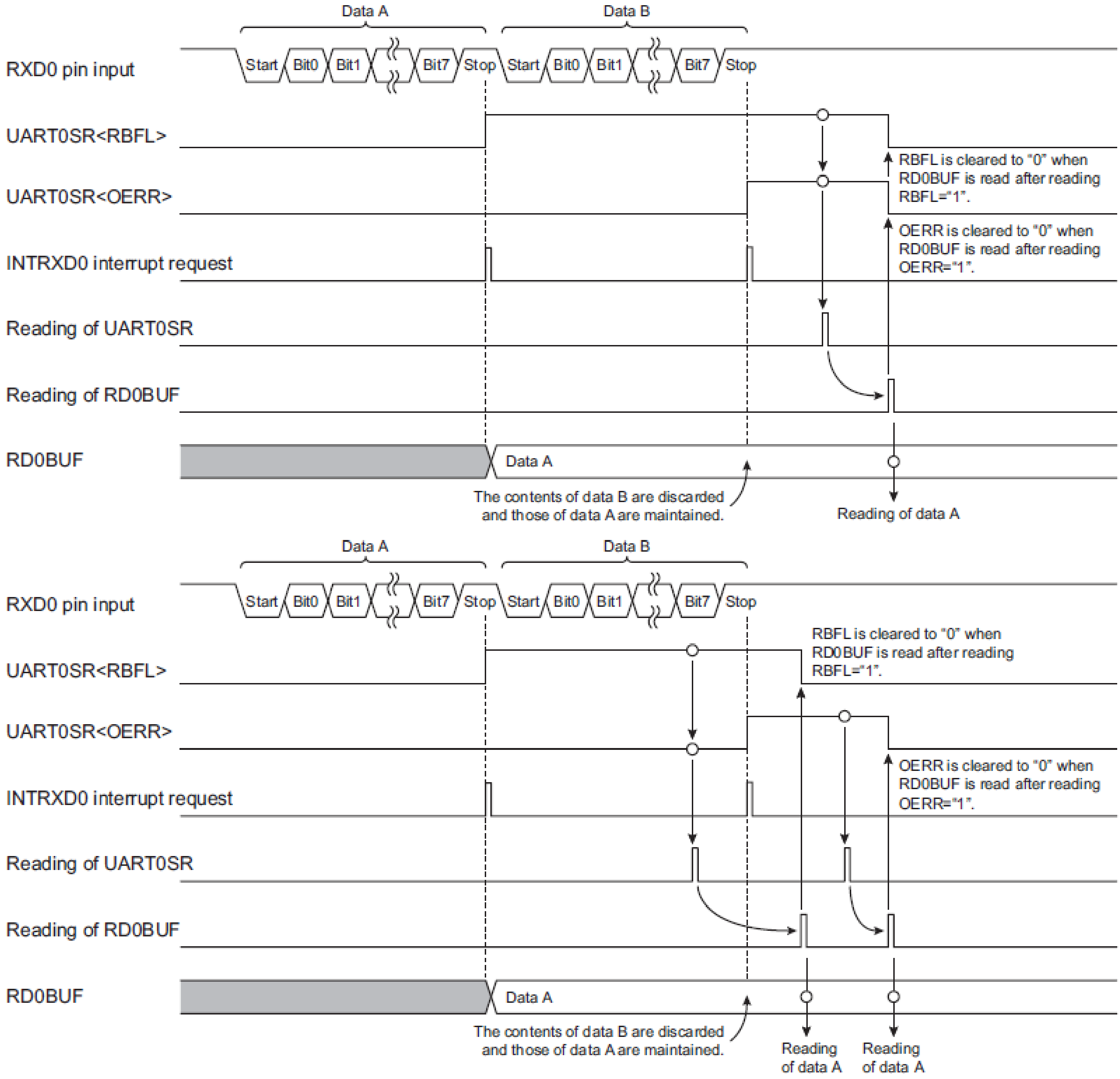

Data A
Data B
RXD0 pin input
Start
Bit0
Bit1
Bit7
Stop
UART0SR<RBFL>
UART0SR<OERR>
INTRXD0 interrupt request
Reading of UART0SR
Reading of RD0BUF
RD0BUF
RBFL is cleared to "0" when RD0BUF is read after reading RBFL="1".
OERR is cleared to "0" when RD0BUF is read after reading OERR="1".
The contents of data B are discarded and those of data A are maintained.
Reading of data A
Reading of data A
Reading of data A

Figure 11.12 Clearance of Overrun Error Flag

11.12.4 Receive Data Buffer Full

Loading the received data in RD0BUF sets UART0SR <RBFL> to "1".

If UART0SR <RBFL> is "1" when UART0SR is read, UART0SR <RBFL> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR <RBFL> is set to "1" after UART0SR is read, UART0SR <RBFL> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <RBFL> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

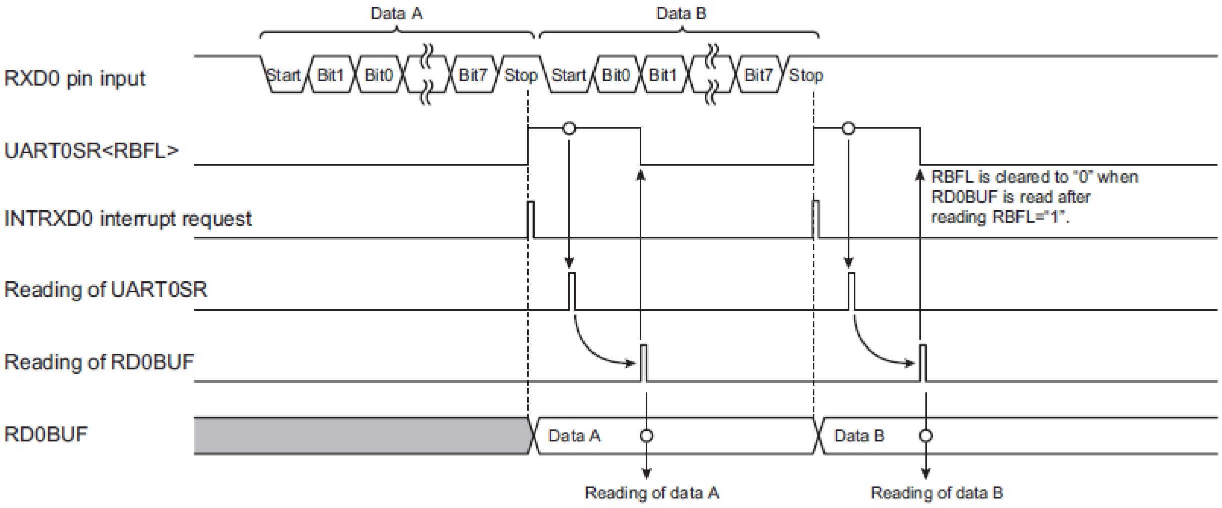


Figure 11.13 Occurrence of Receive Data Buffer Full

11.12.5 Transmit Busy Flag

If transmission is completed with no waiting data in TD0BUF (when UART0SR <TBFL>="0"), UART0SR <TBSY> is cleared to "0". When transmission is restarted after data is written into TD0BUF, UART0SR <TBSY> is set to "1". At this time, an INTTXD0 interrupt request is generated.

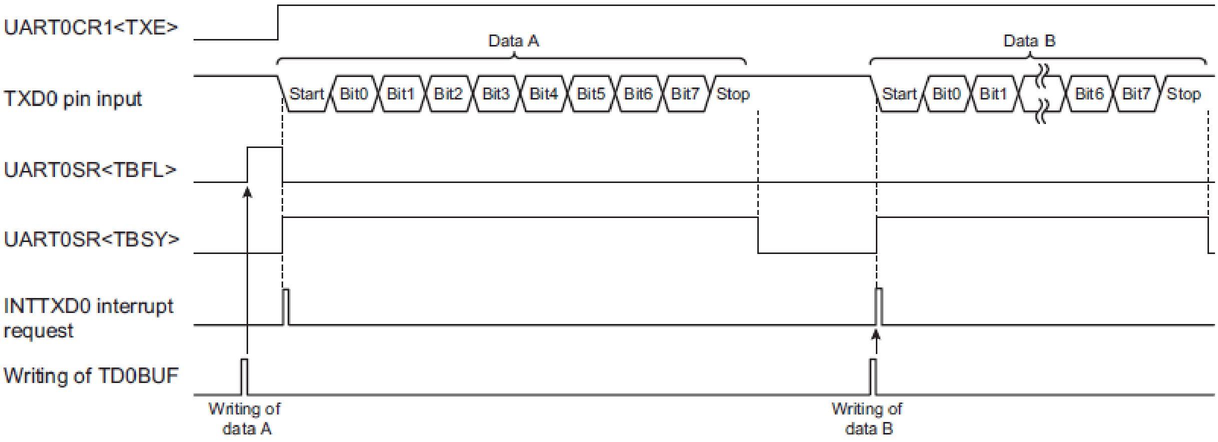


Figure 11.14 Transmit Busy Flag and Occurrence of Transmit Buffer Full

11.12.6 Transmit Buffer Full

When TD0BUF has no data, or when data in TD0BUF is transferred to the transmit shift register and transmission is started, UART0SR <TBFL> is cleared to "0". At this time, an INTTXD0 interrupt request is generated.

Writing data into TD0BUF sets UART0SR <TBFL> to "1".

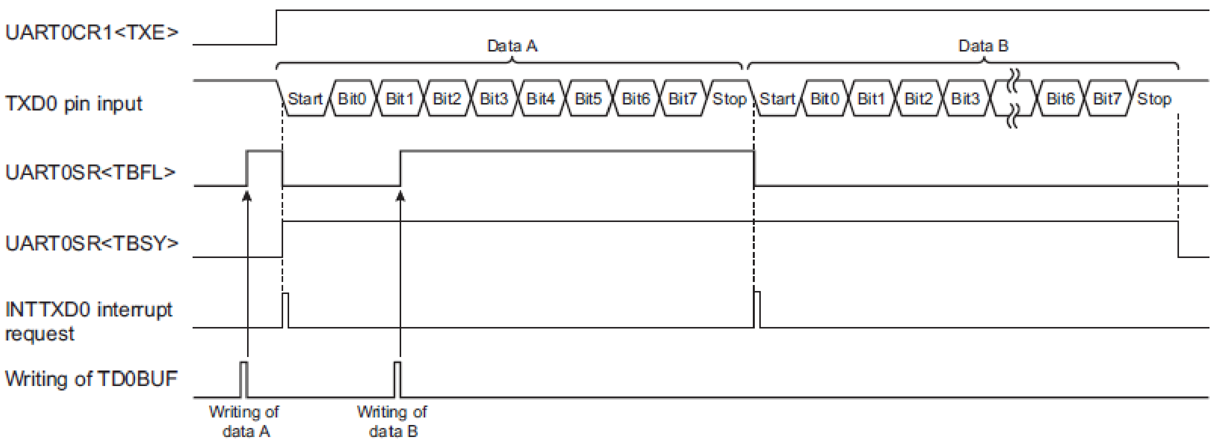


Figure 11.15 Occurrence of Transmit Buffer Full

11.13 Receiving Process

The figure shows an example of the receiving process. Details of flag judgments in the processing are shown in Table as below .

If any framing error or parity error is detected, the received data has erroneous value(s). Execute the error handling, for example, by discarding the received data read from RD0BUF and receiving the data again.

If any overrun error is detected, the receiving of one or more pieces of data is unfinished. It is impossible to determine the number of pieces of data that could not be received. Execute the error handling, for example, by receiving data again from the beginning of the transfer. Basically, an overrun error occurs when the internal software processing cannot follow the data transfer speed. It is recommended to slow the transfer baud rate or modify the software to execute flow control.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

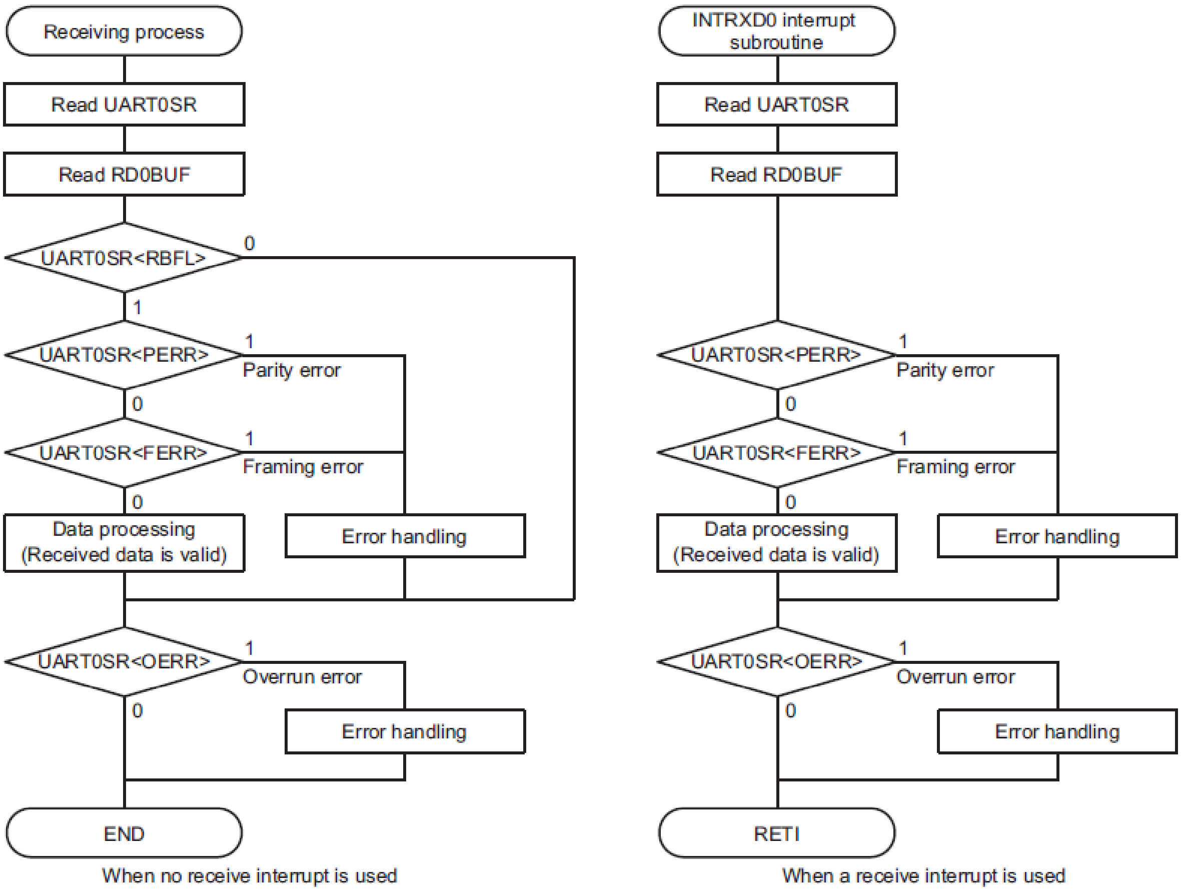
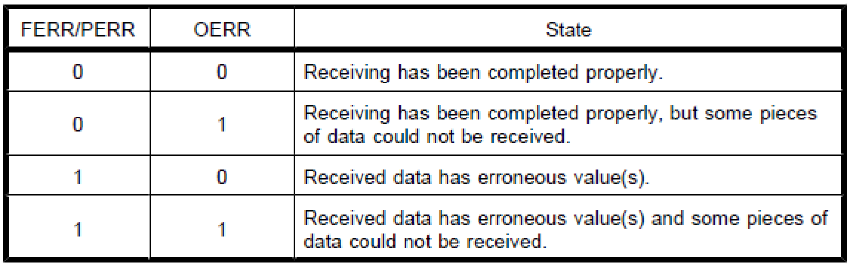


Figure 11.16 Example of Receiving Process

Note): If multiple interrupts are used in the INTRXD0 interrupt subroutine, the interrupt should be enabled after reading UART0SR and RD0BUF.

RBFL	FERR/PERR	OERR	State
0	-	0	Data has not been received yet.
0	-	1	Some pieces of data could not be received during the previous data receiving process (Receiving of next data is completed in the period from when UART0SR is read to when RD0BUF is read in the previous data receiving process.)
1	0	0	Receiving has been completed properly.
1	0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	1	0	Received data has erroneous value(s).
1	1	1	Received data has erroneous value(s) and some pieces of data could not be received.

Table 11.11 Flag Judgments When No Receive Interrupt Is Used



FERR/PERR	OERR	State
0	0	Receiving has been completed properly.
0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	0	Received data has erroneous value(s).
1	1	Received data has erroneous value(s) and some pieces of data could not be received.

Table 11.12 Flag Judgments When a Receive Interrupt Is Used

11.14 AC Properties

11.14.1 IrDA properties

(VSS = 0 V, Topr = −40 to 85°C)

Item	Condition	Min	Typ	Max.	Unit
TXD output pulse time (RT clock × (3/16))	Transfer baud rate = 2400 bps	-	78.13	-	µs
	Transfer baud rate = 9600 bps	-	19.53	-	
	Transfer baud rate = 19200 bps	-	9.77	-	
	Transfer baud rate = 38400 bps	-	4.88	-	
	Transfer baud rate = 57600 bps	-	3.26	-	
	Transfer baud rate = 115200 bps	-	1.63	-	

12. Flash Memory

MQ6935 has flash memory of 16384 bytes. A write and erase to be performed on flash memory can be controlled in the MCU mode, and Serial PROM mode.

<u>MCU mode :</u> In MCU mode, the flash memory is accessed by the CPU control, and the flash memory can be executed the erasing and writing without affecting the operations of a running application. Therefore, this mode is used for software debugging and firmware change after shipment of the MQ6935.

<u>Serial PROM mode :</u> In serial PROM mode, the flash memory is accessed by the CPU control. Use of the serial interface (UART and SIO) enables the flash memory to be controlled by the small number of pins. The MQ6935 used in serial PROM mode supports on-board programming, which enables users to program flash memory after the microcontroller is mounted on a user board.

In MCU mode and serial PROM mode, flash memory control registers (FLSCR1 and FLSCR2) are used to control the flash memory. This chapter describes how to access the flash memory using the MCU mode, and serial PROM mode.

12.1 Flash Memory Control

The flash memory is controlled by the flash memory control register 1 (FLSCR1), flash memory control register 2 (FLSCR2), and flash memory standby control register (FLSSTB).

Flash Memory Control Register 1

FLSCR1 (0x0FD0)	7	6	5	4	3	2	1	0
Bit Symbol	FLSMD			BAREA	FAREA		-	-
Read/Write	R/W			R/W	R/W		R/W	R/W
After reset	0	1	0	0	0		0	0

FLSMD	Flash memory command sequence and toggle control	010: Disable command sequence and toggle execution 101: Enable command sequence and toggle execution Others: Reserved
BAREA	BOOTROM mapping control	MCU mode: 0: Hide BOOTROM 1: Show BOOTROM Serial PROM mode 0: - 1: Show BOOTROM
FAREA	Flash memory area select control	00: Assign the data area 0xC000 through 0xFFFF to the data area 0xC000 through 0xFFFF (standard mapping). 01: Reserved 10:Assign the code area 0xC000 through 0xFFFF to the data area 0xC000 through 0xFFFF. 11: Reserved

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Note 1: It is prohibited to make a setting in "Reserved".

Note 2: The flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

Note 3: FLSMD must be set to either "0y010" or "0y101".

Flash Memory Control Register 2

FLSCR2 (0x0FD1)	7	6	5	4	3	2	1	0
Bit Symbol	CR1EN							
Read/Write	W							
After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable / disable control	0xD5: Enable a change in the FLSCR1 setting Others: Reserved

Note]: If "0xD5" is set on FLSCR2<CR1EN> with FLSCR1<FLSMD> set to "101", the flash memory goes into an active state, and MCU consumes the same amount of current as it does during a read.

Flash Memory Control Register 1 Monitor

FLSCRM (0x0FD1)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	FLSMDM	BAREAM	FAREAM		ROMSELM	
Read/Write	R	R	R	R	R		R	
After reset	0	0	0	0	0	0	0	0

FLSMDM	Monitoring of FLSCR1 <FLSMD> status	0: FLSCR1 <FLSMD>="101" setting disabled 1: FLSCR1 <FLSMD>="101" setting enabled
BAREAM	Monitoring of FLSCR1<BAREA> status	Value of currently enabled FLSCR1<BAREA>
FAREAM	Monitoring of FLSCR1<FAREA> status	Value of currently enabled FLSCR1<FAREA>
ROMSELM	Monitoring of FLSCR1<ROMSEL> status	Value of currently enabled FLSCR1<ROMSEL>

Note 1: FLSCRM is the register that checks the value of the shift register of the flash memory control register 1.

Note 2: FLSMDM turns into "1" only if FLSMD="101" becomes effective.

Note 3: If an instruction to read FLSCRM is executed, "0" is read from bits 7 and 6.

Note 4: In serial PROM mode, "1" is always read from BAREAM..

Flash Memory Standby Control Register

FLSSTB (0x0FD2)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	-	FSTB
Read/Write	R	R	R	R	R	R	R	W
After reset	0	0	0	0	0	0	0	0

FSTB	Flash memory standby control	0: Disable flash memory standby 1: Enable flash memory standby

Note 1): A value can be written to FSTB only by using a program that resides in RAM. A value written using a program residing in the flash memory will be invalidated.

Note 2): If FSTB is set to "1", do not execute instructions to fetch or read data from or write data to the flash memory. If they are executed, a flash standby reset will occur.

Note 3): If an instruction to read FLSSTB is executed, "0" is read from bits 7 through 0.

Port Input Control Register (only work in serial PROM mode)

SPCR (0x0FD3)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	PIN1	PIN0
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

PIN1	Port input control (SCLK0 pin) in serial PROM mode	In serial PROM mode 0:Port input disabled 1:Port input enabled In MCU mode 0: Input enabled for all ports 1: Nonfunctional whatever settings are made "0" is read
PIN0	Port input control (except RXD0, TXD0 and SCLK0) in serial PROM mode	In serial PROM mode 0:Port input disabled 1:Port input enabled In MCU mode 0: Input enabled for all ports 1: Nonfunctional whatever settings are made "0" is read

Note 1): A read or write can be performed on the SPCR register only in serial PROM mode. If a write is performed on this register in MCU mode, the port input control does not function. If a read is performed on the SPCR register in MCU mode, "0" is read from bits 7 through 0.

Note 2): All I/O ports are controlled by PIN0, except the ports RXD0, TXD0 and SCLK0 which are used in serial PROM mode. By using PIN1, the SCLK0 pin can be configured separately from other pins.

12.2 Flash Memory Functions

12.2.1 Flash Memory Command Sequence and Toggle Control (FLSCR1 <FLSMD>)

To prevent inadvertent writes to the flash memory due to program error or microcontroller malfunction, the execution of the flash memory command sequence and the toggle operation can be disabled (the flash memory can be write protected) by making an appropriate control register setting (write protect). To enable the execution of the command sequence and the toggle operation, set FLSCR1<FLSMD> to "0y101", and then set "0xD5" on FLSCR2<CR1EN>. To disable the execution of the command sequence, set FLSCR1<FLSMD> to "0y010", and then set "0xD5" on FLSCR2<CR1EN>. If the command sequence or the toggle operation is executed with the execution of the command sequence and the toggle operation set to "disable", the executed command sequence or toggle operation takes no effect. After a reset, FLSCR1<FLSMD> is initialized to "0y010" to disable the execution of the command sequence. FLSCR1<FLSMD> should normally be set to "0y010" except when a write or erase is to be performed on the flash memory.

Note 1): If "0xD5" is set on FLSCR2<CR1EN> with FLSCR1<FLSMD> set to "101", the flash memory goes into an active state, and MCU consumes the same amount of current as it does during a read.

Note 2): If FLSCR1<FLSMD> is set to "disable", subsequent commands (write instructions) generated are rejected but a command sequence being executed is not initialized.

If you want to set FLSCR1<FLSMD> to "disable", you must finish all command sequences and verify that the flash memory is ready to be read.

12.2.2 Flash memory area switching (FLSCR1<FAREA>)

To perform an erase or write on the flash memory, a memory transfer instruction (command sequence) must be executed. If a memory transfer instruction is used to read or write data, a read or write can be performed only on the data area. To perform an erase or write on the code area, therefore, part of the code area must be temporarily switched to the data area. This switching between data and code areas is performed by making the appropriate FLSCR1<FAREA> setting.

By setting "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<FAREA> to "10", 0xC000 through 0xFFFF (AREA C1) in the code area is mapped to 0xC000 through 0xFFFF (AREA D1) in the data area.

To restore the flash memory to the initial state of mapping, set FLSCR1<FAREA> to "00", and then set "0xD5" on FLSCR2<CR1EN>.

All flash memory areas can be accessed by performing the appropriate steps described above and then executing the memory transfer instruction on 0xC000 through 0xFFFF (AREA D1) in the data area.

0xC000 through 0xFFFF (AREA D1) in the data area and 0xC000 through 0xFFFF (AREA C1) in the code area are mirror areas; these two areas refer to the same physical address in memory. Therefore, an erase or write must be performed on one of these two mirror areas. For example, If a write is

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

performed on 0xC000 in the data area with FLSCR1<FAREA> set to "10" after performing a write on 0xC000 in the data area with FLSCR1<FAREA> set to "00", data is overwritten. To write data to the flash memory that already has data written to it, existing data must first be erased from the flash memory by performing a sector erase or chip erase, and then data must be written.

Additionally, access to areas to which memory is not assigned should be avoided by executing an instruction or specifying such an area by using jump or call instructions.

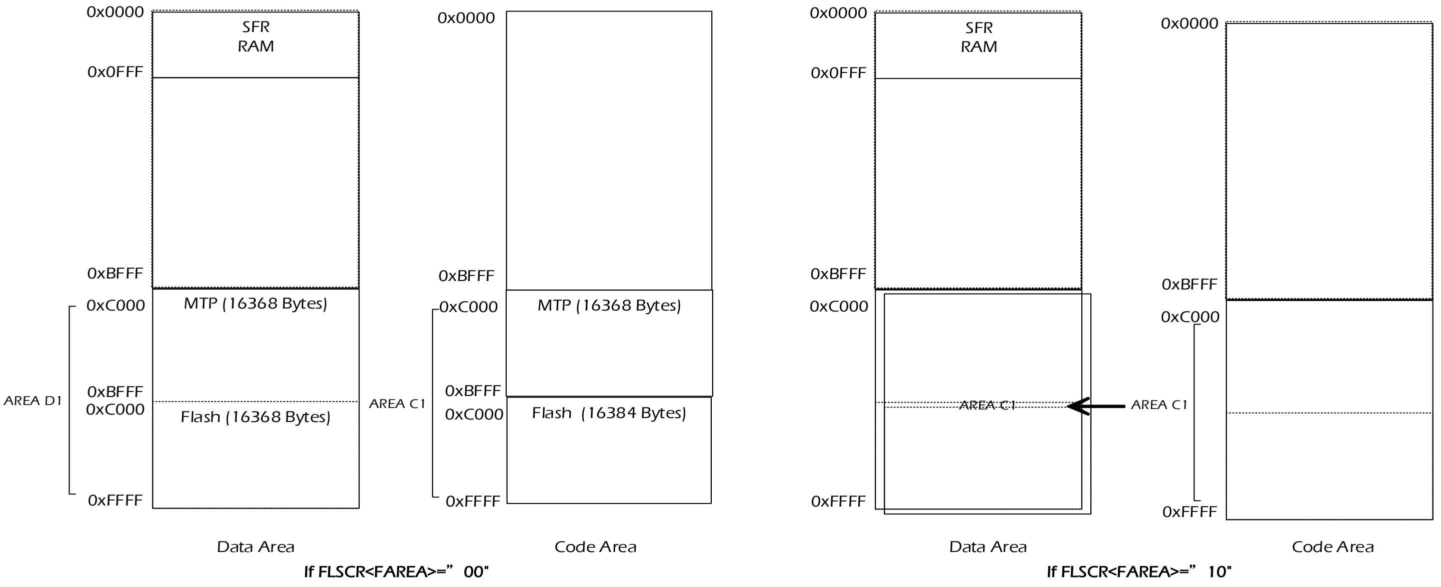


Figure 12.1 Area Switching Using the FLSCR1<FAREA> Setting

12.2.3 RAM area switching (33<RAREA>)

If "0xD4" is set on SYSCR4 after SYSCR3<RAREA> is set to "1" in MCU mode, RAM is mapped to the code area. To restore the RAM area to the initial state of mapping, set SYSCR3<RAREA> to "0", and then set "0xD4" on SYSCR4.

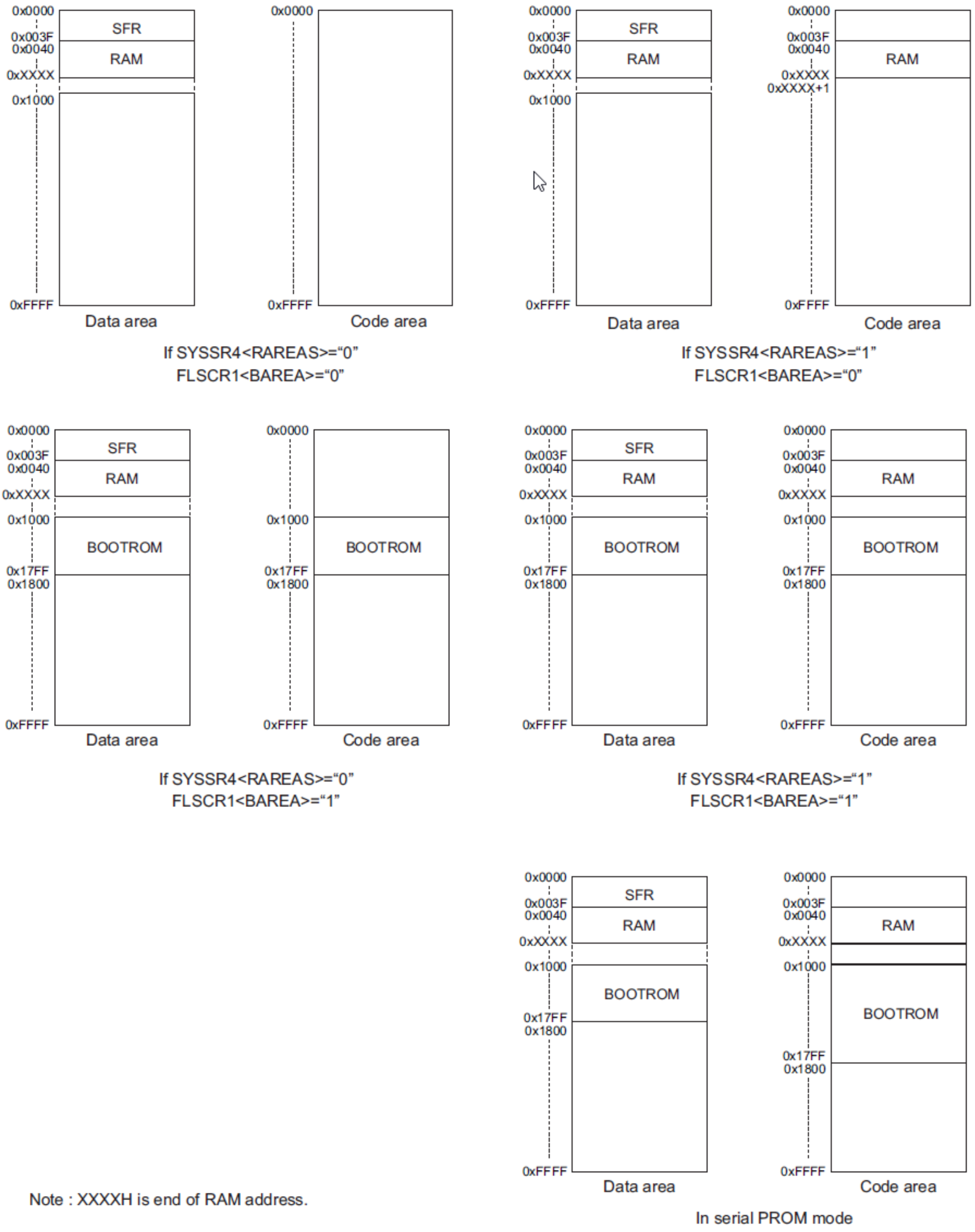
In serial PROM mode, RAM is mapped to the code area, irrespective of the SYSCR3<RAREA> setting.

12.2.4 BOOTROM area switching (FLSCR1<BAREA>)

If "0xD5" is set on FLSCR2<CR1EN> after FLSCR1<BAREA> is set to "1" in MCU mode, 0x1000 through 0x17FF in the code and data areas is masked by flash memory, and 2K-byte (first half of 4KB) BOOTROM is mapped. If you do not want to map BOOTROM, set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "0".

A set of codes for programming flash memory in serial PROM mode are built into BOOTROM, and a support program (API) for performing an erase or write on flash memory in a simple manner is also built into one part in the BOOTROM area. Therefore, by calling a subroutine in the support program after BOOTROM is mapped, it is possible to erase, write and read flash memory easily.

In serial PROM mode, BOOTROM is mapped to 0x1000 through 0x17FF in the data area and 0x1000 through 0x1FFF in the code area, irrespective of the FLSCR1<BAREA> setting. BAREA is always "1", and the set BAREA value remains unchanged, even if data is written. "1" is always read from BAREA.



If SYSSR4<RAREAS>="0"
FLSCR1<BAREA>="0"

If SYSSR4<RAREAS>="1"
FLSCR1<BAREA>="0"

If SYSSR4<RAREAS>="0"
FLSCR1<BAREA>="1"

If SYSSR4<RAREAS>="1"
FLSCR1<BAREA>="1"

In serial PROM mode

Note : XXXXH is end of RAM address.

Figure 12.2 Show/Hide Switching for BOOTROM and RAM

12.2.5 Flash Memory Standby Control (FLSSTB <FSTB>)

FLSSTB<FSTB> is the register provided to maintain the compatibility with the previous product version. It must normally be set to "0". In using FLSSTB<FSTB> built into the MCU, the following point should be noted: FLSSTB<FSTB> can be configured only by using a program allocated to RAM. If it is configured by using a program allocated to the flash memory, the configured value will be invalidated and does not take effect.

To access the flash memory again after setting FLSSTB<FSTB> to "1", set FLSSTB<FSTB> to "0" by using a program allocated to RAM. If the flash memory is accessed with FLSSTB<FSTB> set to "1," a flash standby reset will occur.

If an interrupt occurs when the interrupt vector is assigned to the flash memory area (SYSCR3<RVCTR> = "0" is effective), FSTB is automatically initialized to "0", and then the interrupt vector of the flash memory area is read. If an interrupt occurs when the interrupt vector is assigned to the RAM area (SYSCR3<RVCTR> = "1" is effective), FSTB is not cleared to "0", and then the interrupt vector of the RAM area is read. In this case, the RAM area should be designated as a referential address of interrupt vector. If the flash memory area is designated as a referential address of interrupt vector, a flash standby reset occurs after an interrupt is generated.

12.2.6 Port Input Control Register (SPCR<PIN0,PIN1>

In serial PROM mode, the input levels of all ports, except the ports RXD0 and TXD0 used in serial PROM mode, are physically fixed after a reset is released. This is designed to prevent a penetration current from flowing through unused ports (port inputs and functional peripheral inputs, which are also used as ports, are disabled). To access the flash memory using the RAM loader mode and a method other than the UART, therefore, port inputs must be set to "enable". To enable the SCLK0 port input, set SPCR<PIN1> to "1". To enable port inputs other than RXD0, TXD0 and SCLK0 port inputs, set SPCR<PIN0> to "1".

In MCU mode, the SPCR register does not function.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

12.3 Command Sequence

In MCU mode and serial PROM mode, the command sequence consists of following commands (JEDEC compatible), as shown in Table 12.1.

Command Sequence		1st Bus Writer Cycle		2nd Bus Writer Cycle		3rd Bus Writer Cycle		4th Bus Writer Cycle		5th Bus Writer Cycle		6th Bus Writer Cycle	
		Add	Data	Add	Data	Add	Data	Add	Data	Add	Data	Add	Data
1	Byte Program	0x#555	0xAA	0x#AAA	0x55	0x#555	0xA0	BA (Note 1)	Data (Note 1)	-	-	-	–
2	Sector Erase(Partial erase in units of 1KB bytes)	0x#555	0xAA	0x#AAA	0x55	0x#555	0x80	0x#555	0xAA	0x#AAA	0x55	SA (Note 2)	0x30
3	Chip Erase (All erase)	0x#555	0xAA	0x#AAA	0x55	0x#555	0x80	0x#555	0xAA	0x#AAA	0x55	0x#555	0x10
4	Product ID Entry	0x#555	0xAA	0x#AAA	0x55	0x#555	0x90	-	-	-	-	-	-
5	Product ID Exit	0xXX	0xF0										
6	Security Program	0x#555	0xAA	0x#AAA	0x55	0x#555	0xA5	0xFF7F	0x00				

Table12.1 Command Sequence

Note 1: Specify the address and data to be written (Refer to Table 12-2 about BA).

Note 2: The area to be erased is specified with the upper 5 bits of the address (Refer to Table 12-3 about SA).

Note 3: Do not start the STOP, IDLE0, IDLE1, IDLE2, SLEEP1 or SLEEP0 mode while a command sequence is being executed or a task specified in a command sequence is being executed (write, erase or ID entry).

Note 4: # ; 0x8 through 0xF should be specified as the upper 4bits of the address. Usually, it is recommended that 0xF is specified.

Note 5: XXX ; Don't care

12.3.1 Byte Program

This command writes the flash memory in units of one byte. The address and data to be written are specified in the 4th bus write cycle. The range of addresses that can be specified is shown in Table 12-2. For example, to write data to 0xC000 in the data area, set FLSCR1<FAREA> to "0y00", set "0xD5" on FLSCR2<CR1EN>, and then specify 0xC000 as an address in the 4th bus write cycle. The time needed to write each byte is 40 μs maximum. The next command sequence cannot be executed if an ongoing write operation is not completed. To check the completion of the write operation, perform read operations twice on the same address in the flash memory, and perform polling until the same data is read from the flash memory. During the write operation, bit 6 is reversed each time a read is performed.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Note 1: To rewrite data to addresses in the flash memory where data (including 0xFF) is already written, make sure that you erase the existing data by performing a sector erase or chip erase before writing data.

Note 2: The data and code areas become mirror areas. As you access these areas, you are brought to the same physical address in memory. When performing a Byte Program, make sure that you write data to either of these two areas, not both.

Note 3: Do not perform a Byte Program on areas other than those shown in Table 12-2.

Write Area		FLSCR1 <FAREA>	Address specified by instruction (Address of 4th bus write cycle)
AREA D1 (Data area)	0x8000 through 0xFFFF	00	0x8000 through 0xFFFF
AREA C1 (Code area)	0x8000 through 0xFFFF	10	0x8000 through 0xFFFF

Table 12.2　Range of addresses specifiable (BA)

12.3.2　Sector Erase (1K Byte Partial Erase)

This command erases the flash memory in units of 1 kbytes. The flash memory area to be erased is specified by the upper 5 bits of the 6th bus write cycle address. The range of addresses that can be specified is shown in Table 12-3. For example, to erase 1 kbytes from 0xC000 through 0xC3FF in the code area, set FLSCR1<FAREA> to "0y10", set "0xD5" on FLSCR2<CR1EN>, and then specify either 0xC000 or 0xC3FF as the 6th bus write cycle. The sector erase command is effective only in MCU and serial PROM modes, and it cannot be used in parallel PROM mode.

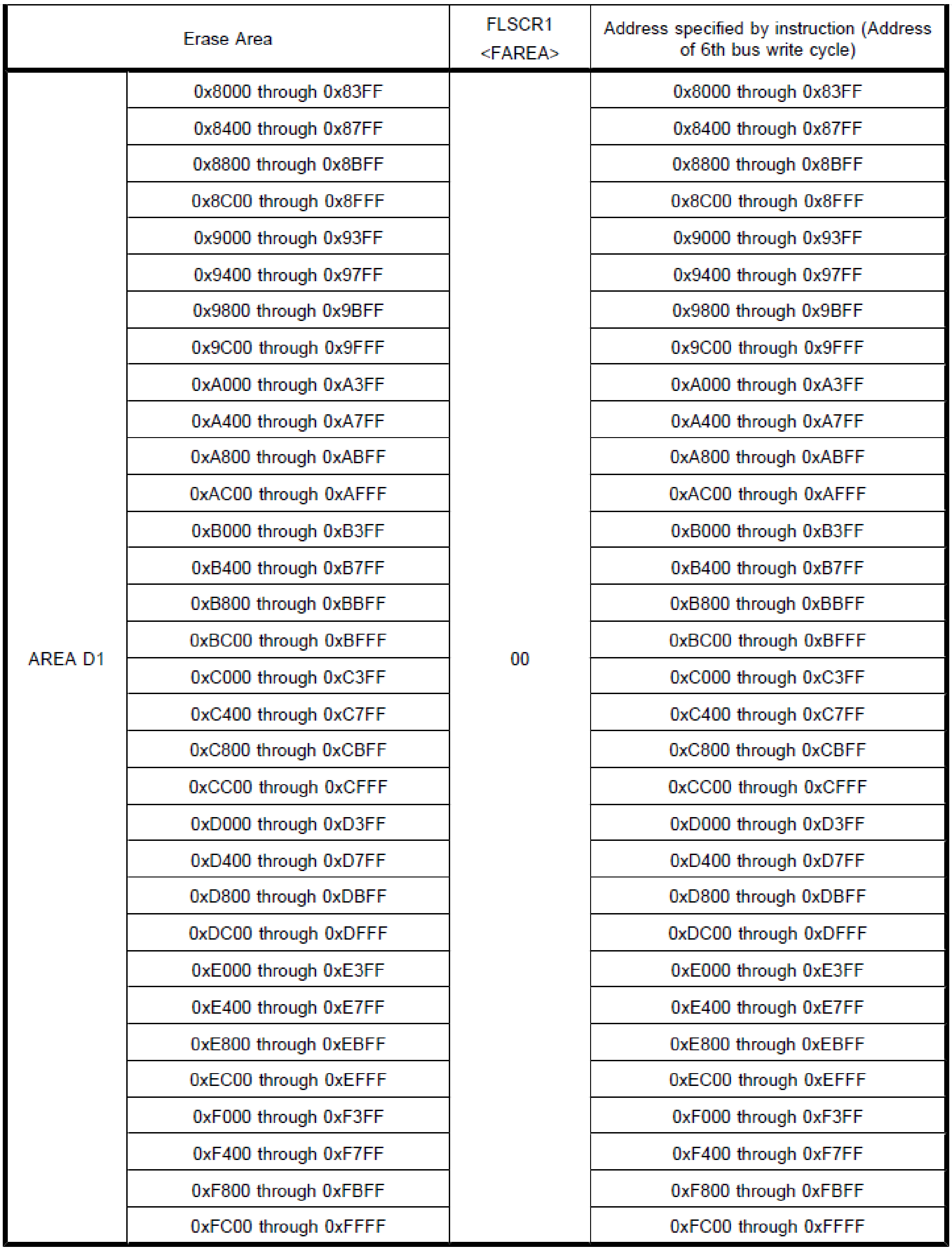
The time needed to erase 1 kbytes is 40 ms maximum. The next command sequence cannot be executed if an ongoing erase operation is not completed. To check the completion of the erase operation, perform read operations twice on the same address in the flash memory, and perform polling until the same data is read from the flash memory. During the erase operation, bit 6 is reversed each time a read is performed.

Data in the erased area is 0xFF.

Note 1: The data and code areas become mirror areas. As you access these areas, you are brought to the same physical address in memory. When performing a sector erase, make sure that you erase data from either of these two areas, not both.

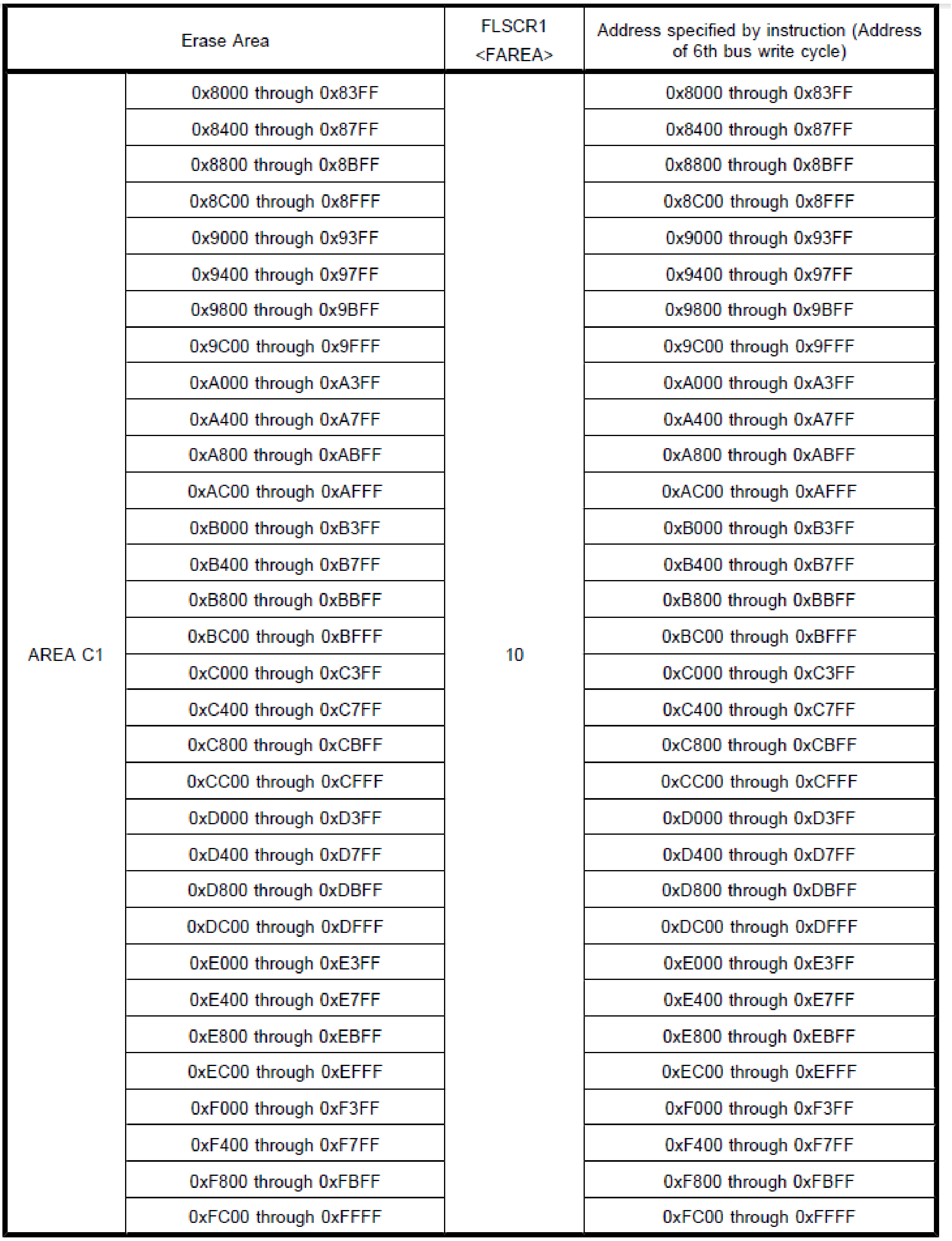
Note 2: Do not perform a sector erase on areas other than those shown in Table 12-3.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.



Erase Area		FLSCR1 <FAREA>	Address specified by instruction (Address of 6th bus write cycle)
AREA D1	0x8000 through 0x83FF	00	0x8000 through 0x83FF
	0x8400 through 0x87FF		0x8400 through 0x87FF
	0x8800 through 0x8BFF		0x8800 through 0x8BFF
	0x8C00 through 0x8FFF		0x8C00 through 0x8FFF
	0x9000 through 0x93FF		0x9000 through 0x93FF
	0x9400 through 0x97FF		0x9400 through 0x97FF
	0x9800 through 0x9BFF		0x9800 through 0x9BFF
	0x9C00 through 0x9FFF		0x9C00 through 0x9FFF
	0xA000 through 0xA3FF		0xA000 through 0xA3FF
	0xA400 through 0xA7FF		0xA400 through 0xA7FF
	0xA800 through 0xABFF		0xA800 through 0xABFF
	0xAC00 through 0xAFFF		0xAC00 through 0xAFFF
	0xB000 through 0xB3FF		0xB000 through 0xB3FF
	0xB400 through 0xB7FF		0xB400 through 0xB7FF
	0xB800 through 0xBBFF		0xB800 through 0xBBFF
	0xBC00 through 0xBFFF		0xBC00 through 0xBFFF
	0xC000 through 0xC3FF		0xC000 through 0xC3FF
	0xC400 through 0xC7FF		0xC400 through 0xC7FF
	0xC800 through 0xCBFF		0xC800 through 0xCBFF
	0xCC00 through 0xCFFF		0xCC00 through 0xCFFF
	0xD000 through 0xD3FF		0xD000 through 0xD3FF
	0xD400 through 0xD7FF		0xD400 through 0xD7FF
	0xD800 through 0xDBFF		0xD800 through 0xDBFF
	0xDC00 through 0xDFFF		0xDC00 through 0xDFFF
	0xE000 through 0xE3FF		0xE000 through 0xE3FF
	0xE400 through 0xE7FF		0xE400 through 0xE7FF
	0xE800 through 0xEBFF		0xE800 through 0xEBFF
	0xEC00 through 0xEFFF		0xEC00 through 0xEFFF
	0xF000 through 0xF3FF		0xF000 through 0xF3FF
	0xF400 through 0xF7FF		0xF400 through 0xF7FF
	0xF800 through 0xFBFF		0xF800 through 0xFBFF
	0xFC00 through 0xFFFF		0xFC00 through 0xFFFF

Table 12.3 Range of address specifiable (Area D1)



Erase Area		FLSCR1 <FAREA>	Address specified by instruction (Address of 6th bus write cycle)
AREA C1	0x8000 through 0x83FF	10	0x8000 through 0x83FF
	0x8400 through 0x87FF		0x8400 through 0x87FF
	0x8800 through 0x8BFF		0x8800 through 0x8BFF
	0x8C00 through 0x8FFF		0x8C00 through 0x8FFF
	0x9000 through 0x93FF		0x9000 through 0x93FF
	0x9400 through 0x97FF		0x9400 through 0x97FF
	0x9800 through 0x9BFF		0x9800 through 0x9BFF
	0x9C00 through 0x9FFF		0x9C00 through 0x9FFF
	0xA000 through 0xA3FF		0xA000 through 0xA3FF
	0xA400 through 0xA7FF		0xA400 through 0xA7FF
	0xA800 through 0xABFF		0xA800 through 0xABFF
	0xAC00 through 0xAFFF		0xAC00 through 0xAFFF
	0xB000 through 0xB3FF		0xB000 through 0xB3FF
	0xB400 through 0xB7FF		0xB400 through 0xB7FF
	0xB800 through 0xBBFF		0xB800 through 0xBBFF
	0xBC00 through 0xBFFF		0xBC00 through 0xBFFF
	0xC000 through 0xC3FF		0xC000 through 0xC3FF
	0xC400 through 0xC7FF		0xC400 through 0xC7FF
	0xC800 through 0xCBFF		0xC800 through 0xCBFF
	0xCC00 through 0xCFFF		0xCC00 through 0xCFFF
	0xD000 through 0xD3FF		0xD000 through 0xD3FF
	0xD400 through 0xD7FF		0xD400 through 0xD7FF
	0xD800 through 0xDBFF		0xD800 through 0xDBFF
	0xDC00 through 0xDFFF		0xDC00 through 0xDFFF
	0xE000 through 0xE3FF		0xE000 through 0xE3FF
	0xE400 through 0xE7FF		0xE400 through 0xE7FF
	0xE800 through 0xEBFF		0xE800 through 0xEBFF
	0xEC00 through 0xEFFF		0xEC00 through 0xEFFF
	0xF000 through 0xF3FF		0xF000 through 0xF3FF
	0xF400 through 0xF7FF		0xF400 through 0xF7FF
	0xF800 through 0xFBFF		0xF800 through 0xFBFF
	0xFC00 through 0xFFFF		0xFC00 through 0xFFFF

Table 12.4 Range of address specifiable (Area C1)

12.3.3 Chip erase (all erase)

This command erases the entire flash memory.

The time needed to erase it is 40 ms maximum. The next command sequence cannot be executed if an ongoing erase operation is not completed. To check the completion of the erase operation, perform read operations twice on the same address in the flash memory, and perform polling until the same data is read from the flash memory. During the erase operation, bit 6 is reversed each time a read is performed

Data in the erased area is 0xFF.

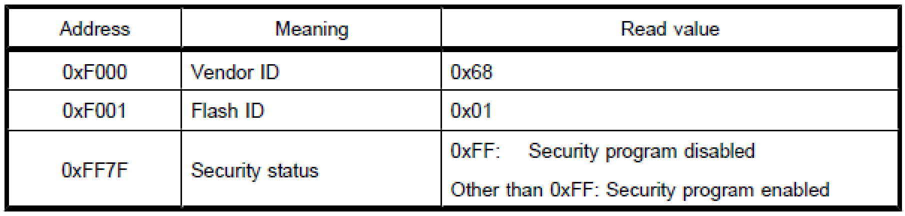
12.3.4 Security program

If the security program is enabled, the flash memory is write and read protected in parallel PROM mode, and the flash memory overwrite command and the RAM loader command cannot be executed in serial PROM mode.

To disable the security program, the chip erase must be performed. To check whether the security program is enabled or disabled, read 0xFF7F in product ID mode. Refer to Table 12-4 for further details. The time needed to enable or disable the security program is 40 μs maximum. The next command sequence cannot be executed until the security program setting is completed. To check the completion of the security program setting, perform read operations twice on the same address in the flash memory, and perform polling until the same data is read. When the security program setting is being made, bit 6 is reversed each time a read is performed.

12.3.5 Product ID Entry

This command activates the product ID mode. If an instruction to read the flash/MTP memory is executed in Product ID mode, the vendor ID, flash/MTP ID and security status can be read from the flash/MTP memory.



Address	Meaning	Read value
0xF000	Vendor ID	0x68
0xF001	Flash ID	0x01
0xFF7F	Security status	0xFF: Security program disabled Other than 0xFF: Security program enabled

Table12.5 Values to Be Read in Product ID Mode

12.3.6 Product ID Exit

This command is used to exit the Product ID mode.

12.4 Toggle Bit (D6)

After the flash memory write and the chip erase, the value of the 6th bit (D6) in data read by a read operation is reversed each time a read is performed. This bit reversal can be used as a software mechanism for checking the completion of each operation. Normally, perform read operations twice on the same address in the flash memory, and perform polling until the same data is read from the flash memory.

After the flash memory write, the chip erase, and the security program command sequence are executed, the toggle bit read by the first read operation is always "1".

Note 1: If FLSCR1<FLSMD> is set to "disable", the toggle bit is not reversed.

Note 2: Do not read the toggle bit by using a 16-bit transfer instruction. If the toggle bit is read using a 16-bit transfer instruction, the toggle bit does not function properly.

Note 3: Because the instruction cycle is longer than the write time in SLOW mode, the value is not reversed, even if the toggle bit is read right after the Byte Program is performed.

12.5 Access to the Flash Memory Area

A read or a program fetch cannot be performed on the whole of the flash memory area if data is being written to the flash memory, if data in flash memory is being erased or if a security setting is being made in the flash memory. When performing these operation on the flash memory area, the flash memory cannot be directly accessed by using a program in the flash memory; the flash memory must be accessed using a program in the BOOTROM area or the RAM area.

Data can be written to and read from the flash memory area in units of one byte. Data in the flash memory can be erased in units of 1 kbytes, and all data in the flash memory can be erased at one stroke. A read can be performed using one memory transfer instruction. A write or erase, however, must be performed using more than one memory transfer instruction because the command sequence method is used. For information on the command sequence, refer to Table 12-1.

Note 1: To allow a program to resume control on the flash memory area that is rewritten, it is recommended that you let the program jump (return) after verifying that the program has been written properly.

Note 2: Do not reset the MCU (including a reset generated due to internal factors) when data is being written to the flash memory, data is being erased from the flash memory or the security command is being executed. If a reset occurs, there is the

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

possibility that data in the flash memory may be rewritten to an unexpected value.

12.5.1 Flash Memory Control in MCU Mode

In MCU mode, a write can be performed on the flash memory by executing a control program in RAM or using a support program (API) provided inside BOOTROM.

12.5.1.1 How to write to the flash memory by transferring a control program to the RAM area

This section describes how to execute a control program in RAM in MCU mode. A control program to be executed in RAM must be acquired and stored in the flash memory or it must be imported from an outside source through a communication pin. (The following procedure assumes that a program copy is provided inside the flash memory.)

Steps 1 through 5 and 11 shown below concern the control by a program in the flash memory, and other steps concern the control by a program transferred to RAM. The following procedure is linked with a program example to be described later.

1. Set the interrupt master enable flag to "disable (DI)" (IMF ← "0").

2. Transfer the write control program to RAM.

3. Establish the non-maskable interrupt vector in the RAM area.

4. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "1", set "0xD4" on SYSCR4. Then allocate RAM to the code area, and switch the vector area to the RAM area.

5. Invoke the erase processing program in the RAM area by generating a CALL instruction.

6. Set FLSCR1<FLSMD> to "0y101", and specify the area to be erased by making the appropriate FLSCR1<FAREA> setting. (Make the appropriate FLSCR1<ROMSEL> setting, as necessary.) Then set "0xD5" on FLSCR2<CR1EN>.

7. Execute the erase command sequence.

8. Perform a read on the same address in the flash memory twice consecutively. (Repeat this step until the read values become the same.)

9. After setting FLSCR1<FLSMD> to "0y010" and FLSCR1<FAREA> to "0y00", set "0xD5" on FLSCR2<CR1EN>. (This disables the execution of the command sequence and returns FAREA to the initial state of mapping.)

10. Generate the RET instruction to return to the flash memory.

11. Invoke the write program in the RAM area by generating a CALL instruction.

12. Set FLSCR1<FLSMD> to "0y101", and make the appropriate FLSCR1<FAREA> setting to specify the area (area erased by performing step 7 above) on which a write is to be performed. (Make the appropriate FLSCR1<ROMSEL> setting, as necessary.) Then set "0xD5" on FLSCR2<CR1EN>.

13. Execute the write command sequence.

14. Perform a read on the same address in the flash memory twice consecutively. (Repeat this step until the read values become the same.)

15. After setting FLSCR1<FLSMD> to "0y010" and FLSCR1<FAREA> to "0y00", set "0xD5" on FLSCR2<CR1EN>. (This disables the execution of the command sequence and returns FAREA to the initial state of mapping.)

16. Generate the RET instruction to return to the flash memory.

17. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "0", set "0xD4" on SYSCR4. Then release RAM allocation for the code area, and switch the vector area to the flash area.

Note 1: Before writing data to the flash memory from the RAM area in MCU mode, the vector area must be switched to the RAM area by using SYSCR3<RVCTR>, data must be written to the vector addresses (INTUNDEF, INTSWI: 0x01F8 to 0x01F9, INTWDT: 0x01FC to 0x01FD) that correspond to non-maskable interrupts, and the interrupt subroutine (RAM area) must be defined. This allows you to trap the errors that may occur due to an unexpected non-maskable interrupt during a write. If SYSCR3<RVCTR> is set in the flash memory area and if an unexpected interrupt occurs during a write, a malfunction may occur because the vector area in the flash memory cannot be read properly.

Note 2: Before using a certain interrupt in MCU mode, the vector address corresponding to that interrupt and the interrupt service routine must be established inside the RAM area. In this case, the non-maskable interrupt setting must be made, as explained in Note 1.

Note 3: Before jumping from the flash memory to the RAM area, RAM must be allocated to the code area by making the appropriate SYSCR3<RAREA> setting (setting made in step 4 in the procedure described on the previous page). Example: Case in which a program is transferred to RAM, a sector erase is performed on 0xE000 through 0xEFFF in the code area, and then 0x3F data is written to 0xE500. If non-maskable interrupts (INTSWI, INTUNDEF or INTWDT) occur, system clock reset is generated.

```
                           cRAMStartAdd equ 0x0200     ; RAM start address
main section code abs = 0xF000
                           DI                          ; Disable interrupts(step 1)
; #### Transfer the program to RAM #### (step 2)
                           LD       HL,cRAMStartAdd
                           LD       IX,sRAMprogStart
sRAMLOOP:                  LD       A,(IX)             ; Transfer the program from
                                                       ; sRAMprogStart to
                           LD       (HL),A              ; sRAMprogEnd to cRAMStartAdd.
                           INC      HL
                           INC      IX
                           CMP      IX,sRAMprogEnd
                           J        NZ,sRAMLOOP
; #### Set a nonmaskable interrupt vector inside the RAM area #### (step 3)
                           LD       HL,0x01FC           ; Set INTUNDEF and INTSWI
                                                        interrupt ;vectors
                           LDW      (HL),sINTSWI - sRAMprogStart + cRAMStartAdd
                           LD       HL,0x01F8           ; Set INTWDT interrupt vector
                           LDW      (HL),sINTWDT - sRAMprogStart + cRAMStartAdd
; #### Allocate RAM to the code area. Switch the vector area to RAM #### (step 4)
                           LD       (SYSCR3),0x06       ; Set RAREA and RVCTR to "1"
                           LD       (SYSCR4),0xD4       ; Enable Code
; #### Sector erase and write process ####
                           LD       HL,0xF555           ; Variable for command sequence
                           LD       DE,0xFAAA           ; Variable for command sequence
; Sector erase process (step 5)
                           LD       C,0x00              ; Set upper addresses
                           CALL     sSectorErase - sRAMprogStart + cRAMStartAdd

; Write process (step 11)
                           LD       C,0x00
                           LD       IX,0xE500
                           LD       B,0x3F
                           CALL     sByteProgram - sRAMprogStart + cRAMStartAdd
                                                        ; Write process (0xE500)
; #### Allocate RAM to the code area. Switch the
vector area to RAM #### (step 17)
                           LD       (SYSCR3),0x00       ; Set RAREA and RVCTR to "0"
                           LD       (SYSCR4),0xD4       ; Enable Code
; #### Execute the next main program ####
                           :        :                   ; Execute the main program
                           J        XXXXX
; #### Program to be executed in RAM ####
sRAMprogStart:
                           NOP                          ; Fail-safe process
                           NOP
                           NOP
                           NOP
                           NOP
                           LD       (SYSCR2),0x10       ; Generate system clock reset
sSectorErase:              CALL     sAddConv - sRAMprogStart + cRAMStartAdd
                                                        ; Address conversion process
; Sector erase process
(step 7)
                           LD       (HL),E              ; 1st Bus Write Cycle (note 1)
                           LD             (DE),L        ; 2nd Bus Write Cycle (note 1)
                           LD            (HL),0x80      ; 3rd Bus Write Cycle (note 1)
                           LD       (HL),E              ; 4th Bus Write Cycle (note 1)
                           LD       (DE),L              ; 5th Bus Write Cycle (note 1)
                           LD       (IX),0x30           ; 6th Bus Write Cycle (note 1)
                           J        sRAMopEnd
; Write process (step 13)
sByteProgram:              CALL     sAddConv - sRAMprogStart + cRAMStartAdd
                                                       ; Address conversion process
                           LD       (HL),E             ; 1st Bus Write Cycle (note 1)
                           LD       (DE),L             ; 2nd Bus Write Cycle (note 1)
                           LD           (HL),0xA0      ; 3rd Bus Write Cycle (Note 1)
                           LD       (IX),B             ; 4th Bus Write Cycle (note 1)
; End process
sRAMopEnd:                 NOP                         ; (note 2)
                           NOP                         ; (note 2)
                           NOP                         ; (note 2)
sLOOP1:                    LD       A,(IX)
                           CMP      A,(IX)
                           J        NZ,sLOOP1          ; Loop until the read values become the
                                                       same
                           LD       (FLSCR1),0x40      ; Disable the execution of
                                                       ; command ;sequence
                                                       ; (steps 9 and 15)
                           LD       (FLSCR2),0xD5      ; Reflect the FLSCR1 setting
                           RET                         ; Return to flash memory
```

```
; Address conversion process (steps 6 and 12)
sAddConv:                 LD       WA,IX
                          SWAP     C
                          AND      C,0x10
                          SWAP     W
                          AND      W,0x08
                          OR       C,W
                          XOR      C,0x08
                          SHRC     C
                          OR       C,0xA0
                          LD       (FLSCR1),C          ; Enable the execution of command
                                                       ; sequence Make the FAREA setting.
                          LD       (FLSCR2),0xD5       ; Reflect the FLSCR1 setting
                          LD       WA,IX
                          TEST     C.3
                          J        Z,sAddConvEnd
                          OR       W,0x80
                          LD       IX,WA
sAddConvEnd:              RET
;Interrupt subroutine
sINTWDT:
sINTSWI:                  LD       IX,0xF000
                          LD       A,(IX)
                          CMP      A,(IX)
                          J        NZ,sINTWDT          ; Loop until the read values become the
                                                       ; same
                          LD       (SYSCR2),0x10       ; Generate system clock reset
                          RETN
sRAMprogEnd:              NOP
```

Note 1: In using a write instruction in the xxx bus write cycle, make sure that you use a write instruction of more than three machine cycles or arrange write instructions in such a way that they are generated at intervals of three or more machine cycles. If a 16-bit transfer instruction is used or if write instructions are executed at intervals of two machine cycles, the flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Note 2: If a read of the flash memory (toggle operation) is to be performed after a write instruction is generated in the xth bus write cycle, instructions must be arranged in such a way that they are generated at intervals of three or more machine cycles; machine cycles are counted from when the last xth bus write cycle is generated to when each instruction is generated. Three NOP instructions are normally used. If the interval between instructions is short, the toggle bit does not operation correctly.

12.5.1.2 How to read data from flash memory

To read data from flash memory, execute transfer instruction for memory. It is possible to read the corresponding individual data (include data of code area) to each address in flash memory, if FLSCR1<FAREA> and FLSCR2 is selected properly.

Example: Case in which data is read from 0xF000 in the code area and stored at 0x98 in RAM

```
    LD             (FLSCR1),0xA8         ; Select AREA C1

    LD             (FLSCR2),0xD5         ; Reflect the FLSCR1 setting

    LD             A,(0xF000)            ; Read data from 0xF000

    LD             (0x98),A              ; Store data at 0x98
```

```
        LD          (FLSCR1),0x40       ; Select AREA D0
        LD          (FLSCR2),0xD5       ; Reflect the FLSCR1 setting
```

12.5.1.3 How to set the security program by using a support program (API) of BOOTROM

1. Transfer the subroutine program of nonmaskable interrupt (INTSWI, INTWDT) to RAM.
2. Establish the nonmaskable interrupt vector in the RAM area.
3. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "1", set "0xD4" on SYSCR4. Then allocate RAM to the code area, and switch the vector area to the RAM area.
4. Set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "1".
5. Set "0xD5" to A register as enable code.
6. Set "0x00" to C register.
7. Call address (0x1016). (After processing, security program state returns to A register.)
8. If A register is not "0xFF", jump to sSKIP because security program is already set.
9. Set "0xD5" to A register as enable code.
10. Set "0x00" to C register.
11. Call address (0x1016). (Security program is performed.)
12. Set "0xD5" to FLSCR2 after setting FLSCR1<BAREA> to "0".
13. Set "0xD4" to SYSCR4 after setting SYSCR3<RAREA, RVCTR> to "0".

Example: Whether the security program is enabled or disabled is checked. If it is disabled, it is

```
.BTWrite            equ 0x1010                ; Write data to the flash memory
.BTEraseSec         equ 0x1012                ; Sector Erase
.BTEraseChip        equ 0x1014                ; Chip Erase
.BTGetRP            equ 0x1016                ; Check the status of the security program
.BTSetRP            equ 0x1018                ; Enable the security program



cRAMStartAdd        equ 0x0200                ; RAM start address
main section code abs = 0xF000
; #### Transfer the program to RAM ####
LD                  HL,cRAMStartAdd
LD                  IX,sRAMprogStart
LD                  A,(IX)                    ; Transfer the program from sRAMprogStart to
LD                  (HL),A                    ; sRAMprogEnd to cRAMStartAdd.
INC                 HL
INC                 IX
CMP                 IX,sRAMprogEnd
J                   NZ,sRAMLOOP
; #### Set a nonmaskable interrupt vector inside the RAM area ####
```

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

```
LD                  HL,0x01FC                  ; Set INTUNDEF and INTSWI interrupt vectors
LDW                 (HL),sINTSWI - sRAMprogStart + cRAMStartAdd
LD                  HL,0x01F8                  ; Set INTWDT interrupt vector
LDW                 (HL),sINTWDT - sRAMprogStart + cRAMStartAdd
; #### Allocate RAM to the code area. Switch the vector area to RAM ####
LD                  (SYSCR3),0x06              ; Set RAREA and RVCTR to "1"
LD                  (SYSCR3),0xD4              ; Enable Code
   ; #### Allocate BOOTROM to the data/code area ####
LD                  (FLSCR1),0x50              ; Set BAREA to "1"
LD                  (FLSCR2),0xD5              ; Reflect the FLSCR1 setting
; #### Check the status of the security program ####
LD                  A,0xD5                     ; Enable Code
LD                  C,0x00                     ; Set 0x00 (note 1)
CALL                (.BTGetRP)                 ; Check the status of the security program
CMP                 A,0xFF
J                   NZ,sSKIP                   ; Go to sSKIP if the security program is enabled
; #### Security program enable process (API) ####
LD                  A,0xD5                     ; Enable Code
LD                                 C,0x00      ; Set 0x00 (note 1)
CALL                (.BTSetRP)                 ; Enable the security program
LD                  (FLSCR1),0x40              ; Set BAREA to "0"
LD                  (FLSCR2),0xD5
LD                  (SYSCR3),0x00              ; Set RAREA and RVCTR to "0"
LD                  (SYSCR4),0xD4                 ; Enable Code
:                   :
J                   XXXX
; #### Program to be executed in RAM ####
be executed in RAM ####
sRAMprogStart:

; Interrupt subroutine
LD                  IX,0xF000
LD                  A,(IX)
CMP                 A,(IX)
J                   NZ,sINTWDT                 ; Loop until the read values become the same
LD                  (SYSCR2),0x10              ; Generate system clock reset
RETN
NOP
```

12.5.2 Flash Memory Control in Serial PROM Mode

The serial PROM mode is used to access the flash memory by using a control program provided in the BOOTROM area. Since almost all operations relating to access to the flash memory can be controlled simply using data supplied through the serial interface (UART or SIO), it is not necessary to operate the control register for the user. For details of the serial PROM mode, see "Serial PROM Mode".

To access the flash memory in serial PROM mode by using a user-specific program or peripheral functions other than UART and SIO, it is necessary to execute a control program in the RAM area by using the RAM loader command of the serial PROM mode.

12.5.2.1 How to Transfer and Writer a Control Program to the RAM Area in RAM Loader Mode of the Serial PROM Mode

How to execute a control program in the RAM area in serial PROM mode is described below. A control program to be executed in the RAM area must be generated in the Intel-Hex format and be transferred using the RAM loader of the serial PROM mode.

Steps 1 and 2 shown below are controlled by a program in the BOOTROM, and other steps are controlled by a program transferred to the RAM area. The following procedure is linked with a program example to be explained later.

1. Transfer the write control program to the RAM area in RAM loader mode.
2. Jump to the RAM area.
3. Set a non-maskable interrupt vector in the RAM area.
4. Set FLSCR1<FLSMD> to "0y101", and specify the area to be erased by making the appropriate FLSCR1<FAREA> setting. (Make the appropriate FLSCR1<ROMSEL> setting as required.)
Then set "0xD5" on FLSCR2<CR1EN>.
5. Execute the erase command sequence.
6. Read the same flash memory address twice consecutively. (Repeat step 6 until the read values become the same.)
7. Specify the area (area erased in step 5 above) to which data is written by making the appropriate FLSCR1<FAREA> setting. (Make the appropriate FLSCR1<ROMSEL> setting as required.) Then set "0xD5" on FLSCR2<CR1EN>.
8. Execute the write command sequence.
9. Read the same flash memory address twice consecutively. (Repeat step 9 until the read values become the same.)
10. Set FLSCR1<FLSMD> to "0y010", and then set "0xD5" on FLSCR2<CR1EN> (to disable the execution of the command sequence).

Note 1: If the RAM loader is used in serial PROM mode, the BOOTROM disables (DI) a maskable interrupt, and the interrupt vector area is designated as a RAM area (SYSCR3<RVCTR>="1"). Considering that a non-maskable interrupt may be generated

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

unexpectedly, it is recommended that vector addresses corresponding these interrupts (INTUNDEF, INTSWI: 0x01F8 to 0x01F9, WDT: 0x01FC to 0x01FD) be established and that an interrupt service routine be defined inside the RAM area.

Note 2: If a certain interrupt is used in the RAM loader program, a vector address corresponding to that interrupt and the interrupt service routine must be established inside the RAM area. In this case, it is recommended that a nonmaskable interrupt be handled as explained in Note 1.

Note 3: Do not set SYSCR3<RVCTR> to "0" by using the RAM loader program. If an interrupt occurs with SYSCR3<RVCTR> set to "0", the BOOTROM area is referenced as a vector address and, therefore, the program will not function properly.

```
main section code abs = 0x0100
; #### Set a nonmaskable interrupt vector inside the RAM area #### (step 3)
                                     LD        HL,0x01FC      ; Set INTUNDEF and INTSWI
                                                              ; interrupt vectors
                                     LDW       (HL),sINTSWI   ; LDW (HL),sINTSWI
                                     LD        HL,0x01F8      ; LD HL,0x01F8 ; Set INTWDT
                                                              ; interrupt vector
                                     LDW       (HL),sINTWD
; #### Sector erase and write
process ####
                                     LD        HL,0xF555      ; Variable for command sequence
                                     LD        DE,0xFAAA      ; Variable for command sequence
; Sector erase process (step 5)
                                     LD        C,0x00         ; Set upper address
; Write process (step 8)
                                     LD        C,0x00         ; Set upper address
                                     LD        IX,0xE500      ; Set middle and lower
                                                              ; addresses
                                     LD        B,0x3F         ; Data to be written
                                     CALL      sByteProgram   ; Write process (0xE500)
; #### Execute the next main program ####
                                     :         :              ; Execute the main program
                                     J         XXXXX
; #### Program to be executed in RAM ####
sSectorErase:                        CALL      sAddConv       ; Address conversion process
; Sector erase process
                                     LD        (HL),E         ; 1st Bus Write Cycle (note 1)
                                     LD        (DE),L         ; 2nd Bus Write Cycle (note 1)
                                     LD        (HL),0x80      ; 3rd Bus Write Cycle (note 1)
                                     LD        (HL),E         ; 4th Bus Write Cycle (note 1)
                                     LD        (DE),L         ; 5th Bus Write Cycle (note 1)
                                     LD        (IX),0x30      ; 6th Bus Write Cycle (note 1)
                                     J         sRAMopEnd
; Write process
sByteProgram:                        CALL      sAddConv       ; Convert address
                                     LD        (HL),E         ; 1st Bus Write Cycle (note 1)
                                     LD        (DE),L         ; 2nd Bus Write Cycle (note 1)
                                     LD        (HL),0xA0      ; 3rd Bus Write Cycle (note 1)
                                     LD        (IX),B         ; 4th Bus Write Cycle (note 1)
; End process
sRAMopEnd                            NOP                      ; (note 2)
                                     NOP                      ; (note 2)
                                     NOP                      ; (note 2)
sLOOP1:                              LD        A,(IX)         ; (step 6,9)
                                     CMP       A,(IX)
                                     J         NZ,sLOOP1      ; Loop until the read values
                                                              ; become the same
                                     LD        (FLSCR1),0x40  ; Disable the execution of
                                                              ; command sequence (step 10)
                                     LD        (FLSCR2),0xD5  ; Reflect the FLSCR1 setting
                                     RET                      ; Return to the program in RAM
; Convert address (steps 4 and 7)
   sAddConv:                         LD        WA,IX
                                     SWAP      C
                                     AND       C,0x10
                                     SWAP      W
```

```
                                    AND       W,0x08
                                    OR        C,W
                                    XOR       C,0x08
                                    SHRC      C
                                    OR        C,0xA0
                                    LD        (FLSCR1),C     ; Enable the execution of
                                                             ; command sequence. Make the
                                                             ; FAREA setting.

                                    LD        (FLSCR2),0xD5  ; Reflect the FLSCR1 setting
                                    LD        WA,IX
                                    TEST      C.3
                                    J         Z,sAddConvEnd
                                    OR        W,0x80
                                    LD        IX,WA
sAddConvEnd:                        RET
      ; Interrupt subroutine
sINTWDT:
sINTSWI:                            LD        IX,0xF000
                                    LD        A,(IX)
                                    CMP       A,(IX)
                                    J         NZ,sINTWDT     ; Loop until the read
                                                             ; values ;become the same
                                    LD        (SYSCR2),0x10  ; Generate system clock reset
                                    RETN
```

Note 1: In using a write instruction in the xxx bus write cycle, make sure that you use a write instruction of more than three machine cycles or arrange write instructions in such a way that they are generated at intervals of three or more machine cycles. If a 16-bit transfer instruction is used or if write instructions are executed at intervals of two machine cycles, the flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Note 2: If a read of the flash memory (toggle operation) is to be performed after a write instruction is generated in the xth bus write cycle, instructions must be arranged in such a way that they are generated at intervals of three or more machine cycles; machine cycles are counted from when the last xth bus write cycle is generated to when each instruction is generated. Three NOP instructions are normally used. If the interval between instructions is short, the toggle bit does not operation correctly.

12.6 API (Application Programming Interface)

The BOOTROM has a support program (API) which contains a special subroutine for erasing or writing on the flash memory. After mapping of the BOOTROM, it allows easy erasing or writing on the flash memory by only calling the subroutine in BOOTROM.

Address	Contents	Using Stack	Working Register	Argument		Return Value	
				Register	Setting Value	Register	Setting Value
0x1010 (.BTWrite)	Writing the data to specified address of one byte.	7bytes	WA BC DE IX	WA	Specify the address to be written.	-	-
				C	0x00		
				E	Specify the data to be written.		
				(SP-)	0xD5 (Enable Code)		
0x1014 (.BTEraseChip)	Executing the Chip Erase.	6bytes	WA BC DE IX	A	0xD5 (Enable Code)	-	-
				C	0x00		
0x1016 (.BTGetRP)	Getting the status of Security Program.	6bytes	WA BC DE IX	A	0xD5 (Enable Code)	A	0xFF: Security Program disabled. Others: Security Program enabled.
				C	0x00		
0x1018 (.BTSetRP)	Setting the Security Program.	6bytes	WA BC DE IX	A	0xD5 (Enable Code)		
				C	0x00	-	-
0x101E (.BTCalcUART)	Calculating the setting for UART (Baud rate) from the captured value by timer counter.	4bytes	WA BC DE IX IY	WA	Captured value by timer counter	W	Setting value for RTSEL
				C	The number of bit for calculation.	A	Setteing value for UARTDR

Table 12.6 List of API

Note 1: Because working registers (general-purpose registers) are rewritten in the support program, the contents of general-purpose registers should be saved before calling the support program.

Note 2: While the support program is executed, a maximum 7 bytes are used as stack which does not include the stack used by interrupts. Therefore, be sure to reserve a stack area beforehand.

Note 3: Each API works properly without the setting Enable Code (0xD5) as argument. However, it is recommended to set the Enable Code (0xD5) to keep compatibility in the family products.

12.6.1 .BTWrite

Data in E register is written into the address specified by WA register. C register should be written 0x00 and (SP-) should be written Enable Code (0xD5) before calling the subroutine.

12.6.2 .BTEraseChip

All flash memory area is erased. C register should be written 0x00 and A register should be written Enable Code (0xD5) before calling the subroutine.

12.6.3 .BTGetSP

The security status of flash memory can be read out. C register should be written 0x00 and A register should be written Enable Code (0xD5) before calling the subroutine.

After completion of the execution, API returns A register with the contents of 0xFF7F (security status) in Product ID as return value.

12.6.4 .BTSetSP

The setting of security program can be executed by this API. C register should be written 0x00 and A register should be written Enable Code (0xD5) before calling the subroutine.

12.6.5 .BTCalcuUART

This API calculates the proper setting for baud rate of UART from the value of C and WA register. Generally, 8 bits data (0x80) of UART is captured by 16-bit timer counter which is set to pulse width measurement mode. In this case, the timer counter input pin should be assigned to RXD pin. And stores the captured value in WA register. Be sure to select fcgck/2 as the source clock for 16-bit timer counter and capture the length of 8 bits. C register should be written 0x08.

RXD pin can be used as TCA pin. To capture the value, please select the pin as TCA pin by SERSEL< TCA0SEL> temporarily. After capturing, be sure to resume the pin to RXD pin. The possible value for WA register as argument is from 0x0020 to 0x3BFF. In the return value of WA register, bits 5, 4 and 3 are suitable value for UARTCR2<RTSEL> and the return value of A register is a suitable data for UARTDR. The API sets bits 7, 6, 2, 1 and 0 to "0" as return value. Therefore, set the proper value for UARTCR2<RXDNC, STOPBR>. If the contents of WA register is out of the area 0x0020 to 0x3BFF, the API returns WA register with 0xFFFF as return value.

Note 1: If the captured value of WA register is little even though the value is within 0x0020 to 0x3BFF, the proper setting may not be gotten.

The following procedure shows example how to calculate the baud rate for UART in MCU mode by using support program.

1. By serial interface selection control register SERSEL<TCA0SEL>, assign TCA pin to RXD pin.

2. Set 16-bit timer counter to pulse width measurement mode. And set falling edge/L level as an externa trigger and select fcgck/2 as the source clock.

3. Receive data (0x80) via RXD pin and capture it by 16-bit timer counter. In this case, enabling of UART is no need.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

4. Write the captured value into WA register. Because general-purpose registers (DE, BC, IX, IY) are rewritten in the support program, the contents of these registers should be saved before calling the support program.

5. Set the interrupt master enable flag to "disable (DI)" (IMF ← "0").

6. Set "0xD5" on FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "1".

7. Set "0x08" to C register as the number of bit.

8. Call address (0x101E).

9. Set bits 5, 4 and 3 of W register into UARTCR2<RTSEL> and set the contents of A register to UARTDR. If the value of WA register is 0xFFFF which indicates an error of calculation, retry the execution from receiving data (0x80).

10. Set "0xD5" to FLSCR2 after setting FLSCR1<BAREA> to "0".

Note 1: If general-purpose registers (WA, BC, DE, IX, IY) are used in non-maskable interrupt subroutine, occurring of non-maskable interrupt may cause unexpected result.

Note 2: With success of calculation, this API returns "0" into bits 7, 6, 2, 1 and 0 as return value. Therefore, set proper value for these bits to set UART0CR2<RXDNC> and UART0CR2<STOPBR>.

Example: Captures the low width of 8 bit value via RXD pin by 16-bit timer counter which is set to the pulse width measurement mode. And calculates baud rate for UART from the captured value.

```
.BTCalcUART      equ 0x101E          ; Calculating the setting for UART (Baud rate)
CalcUART secion code abs = 0xF000
; #### Assign TCA input to RXD pin ####
LD               (SERSEL),0x40       ; Assign TCA0 pin to RXD pin
; #### Receive data (0x80) from a master device ####
LD               (TA0MOD),0x5E       ; Set the pulse width measurement mode and select falling
                                     ; edge/L level for external trigger
                                     ; Select fcgck/2 as the surce clock
LD               (TA0CR),0x01        ; Timer start
```

Receives data (0x80) via TCA0 pin

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

```
LD              WA,(TA0DRL)             ; Write the captured data into WA register
LD              (TA0CR), 0x00           ; Timer stop
DI
; #### Allocate BOOTROM to the data/code area ####
LD              (FLSCR1),0x50           ; Set BAREA to "1"
LD              (FLSCR2),0xD5           ; Reflect the FLSCR1 setting
; #### Calculation for UART setting (API) ####
LD              C,0x08                  ; The number of bit length (8 bit)
CALL            (.BTCalcUART)           ; Calculate UART setting
CMP             W, 0xFF
J               Z, sTimerStart          ; Return to sTimerStart if W register equals 0xFF
; #### Setting the calculated result to UART registers ####
LD              (UARTCR2),W             ; Set RTSEL
LD              (UARTDR), A             ; Set UARTDR
; #### End process ####
LD              (FLSCR1),0x40           ; Set BAREA to "0"
LD              (FLSCR2),0xD5           ; Reflect the FLSCR1 setting
```

13 Serial PROM Mode

The MQ6935 has a 4K-byte BOOTROM (Mask ROM) for programming to flash memory. BOOTROM is available in serial PROM mode. The serial PROM mode is controlled by RXD0/SI0 pins, TXD0/SO0 pins, MODE pin, and RESETB pin. In serial PROM mode, communication is performed via the UART or SIO.

Parameter	Min	Max	Unit
Power supply voltage	2.7	5.5	V
High frequency	1	16	MHz

Table 13.1 Operating Range in Serial PROM Mode

13.1 Serial PROM Mode Setting

13.1.1 Serial PROM mode control pins

To execute on-board programming, activate the serial PROM mode. Table13.2 shows the pin setting used to activate the serial PROM mode.

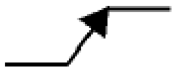
Pin	Setting
RXD0 / SI0 / P21 pin	H Level
TXD0 / SO0 / P20 pin	H Level
MODE, RESETB pin	

Table 13.2 Serial PROM Mode Setting

Note: Before you activate the serial PROM mode, you must set the RXD0/SI0/P21 and TXD0/SO0/P20 pins to high (H) level by using a pull-up resistor.

Pin name (in serial PROM mode)	input/ output	Function		Pin name (in MCU mode)
TXD0 / SO0	Output	Serial PROM mode control/serial data output	Note1	TXD0 / SO0 / P20
RXD0 / SI0	Input	Serial PROM mode control/serial data input		RXD0 / SI0 / P21
RESETB	Input	Serial PROM mode control		RESETB
MODE	Input	Serial PROM mode control		MODE
SCLK0	Input	Serial clock input (if SIO is used) These ports are in the high-impedance state in the serial PROM mode. If the UART is used, the port input is physically fixed to a specified input level in order to prevent a penetration current. To enable the port input, the SPCR<PIN1> must be set to "1" by operating the RAM loader control program.		SCLK0
VDD	Power supply	2.7 V to 5.5 V		
VAREF / AVDD	Power supply	Connect to VDD.		
VSS	Power supply	0 V		
Input/output port other than RXD0 and TXD0	Input/ output	These ports are in the high-impedance state in the serial PROM mode. The port input is physically fixed to a specified input level in order to prevent a penetration current (the port input is disabled). To enable the port input, the SPCR<PIN0> must be set to "1" by operating the RAM loader control program.		
XIN	Input	Connect a resonator to make these pins self-oscillate.		
XOUT	Output	Connect a resonator to make these pins self-oscillate.		

Table 13.3 Pin Functions in Serial PROM Mode

Note 1: If other parts are mounted on a user board, they may interfere with data being communicated through these communication pins during on-board programming. It is recommended that these parts be somehow isolated to prevent the pins from being affected.

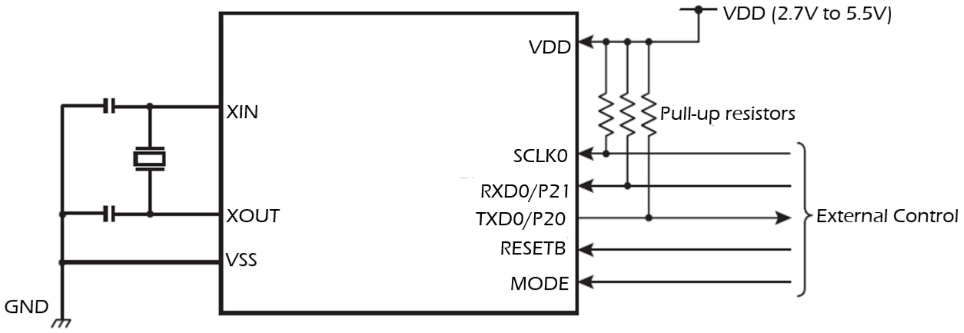


Figure13.1 Serial PROM Mode Pin Setting

Note 1: In the case of access using the UART, the control of the SCLK0 pin is unnecessary.

Note 2: For information on other pin settings, refer to "Table 23-3 Pin Functions in Serial PROM Mode".

13.2 Activating the Serial PROM Mode

Activate the serial PROM mode by performing the following procedure.

1. Supply power to the VDD pin.
2. Set the RESETB and MODE pins to low.
3. Set the RXD0/SI0/P21 and TXD0/SO0/P20 pins to high.
4. Wait until the power supply and clock oscillation stabilize.
5. Set the RESETB and MODE pins from low to high.
6. Input the matching data 0x86 or 0x30 to the RXD0/SI0/P21 pins after the setup period has elapsed.

13.3 Interface Specifications

The serial PROM mode supports two communication methods: UART and SIO. The communication method is selected based on the first serial data value received after a reset. To execute an on-board program, the communication format of the external controller (personal computer, microcontroller, etc.) must be set as described below.

13.3.1 SIO communication

- Transfer rate: 250 kbps (Max.)
- Data length: 8 bits
- Slave (external clock)
- Hardware flow control (SO0 pin)

If the MQ6935 receives serial data "0x30" after a reset, it starts the SIO communication.

In the SIO communication, the MQ6935 functions as a slave device. Therefore, the external controller must supply the MQ6935 with a serial clock (SCLK0 pin) for synchronization. If the MQ6935 is not outputting serial data, it controls the hardware flow by using the SO0 pin. If internal data processing is not completed yet, though data has been received, the SO0 pin outputs the L level. If internal data processing has progressed to a near-completion state or if it has been completed, the SO0 pin outputs the H level. The external controller must check the status of the SO0 pin before it starts to supply a serial clock.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

13.3.2 UART communication

- Baud rate: 9600 to 128000 bps (automatic detection)
- Data length: 8 bits (LSB first)
- Parity bit: None
- STOP bit: 1 bit

If the MQ6935 receives serial data "0x86" after a reset, it starts the UART communication. It also measures the pulse width of the received data (0x86), and automatically establishes the reference baud rate.

In all subsequent data communication transactions, this reference baud rate is used. Usable baud rates differ depending on the operating frequency and are shown in Table 13.4. However, there is the possibility of data communication not working properly, even if a baud rate shown in Table 13.4 is used, because data communication is affected by frequency errors of a resonator of the external controller (personal computer, etc.), the load capacity of a communication pin, and various other factors.

	9600 bps	19200 bps	38400 bps	57600 bps	115200 bps	128000 bps
16MHz	V	V	V	V	V	V
10MHz	V	V	V	V	V	V
8MHz	V	V	V	V	V	V
7.3728MHz	V	V	V	V	V	-
6.144MHz	V	V	V	-	-	V
6MHz	V	V	V	V	V	V
5MHz	V	V	V	-	-	-
4.9152MHz	V	V	V	V	-	-
4.19MHz	V	V	V	-	-	V
4MHz	V	V	V	V	V	V
2MHz	V	V	V	V	-	-
1MHz	V	V	-	V	-	-

Note :"V" means a usable baud rate. "-" means an unusable baud rate.

Table 13.4 Usable Baud Rates as a General Guideline

13.4 Memory Mapping

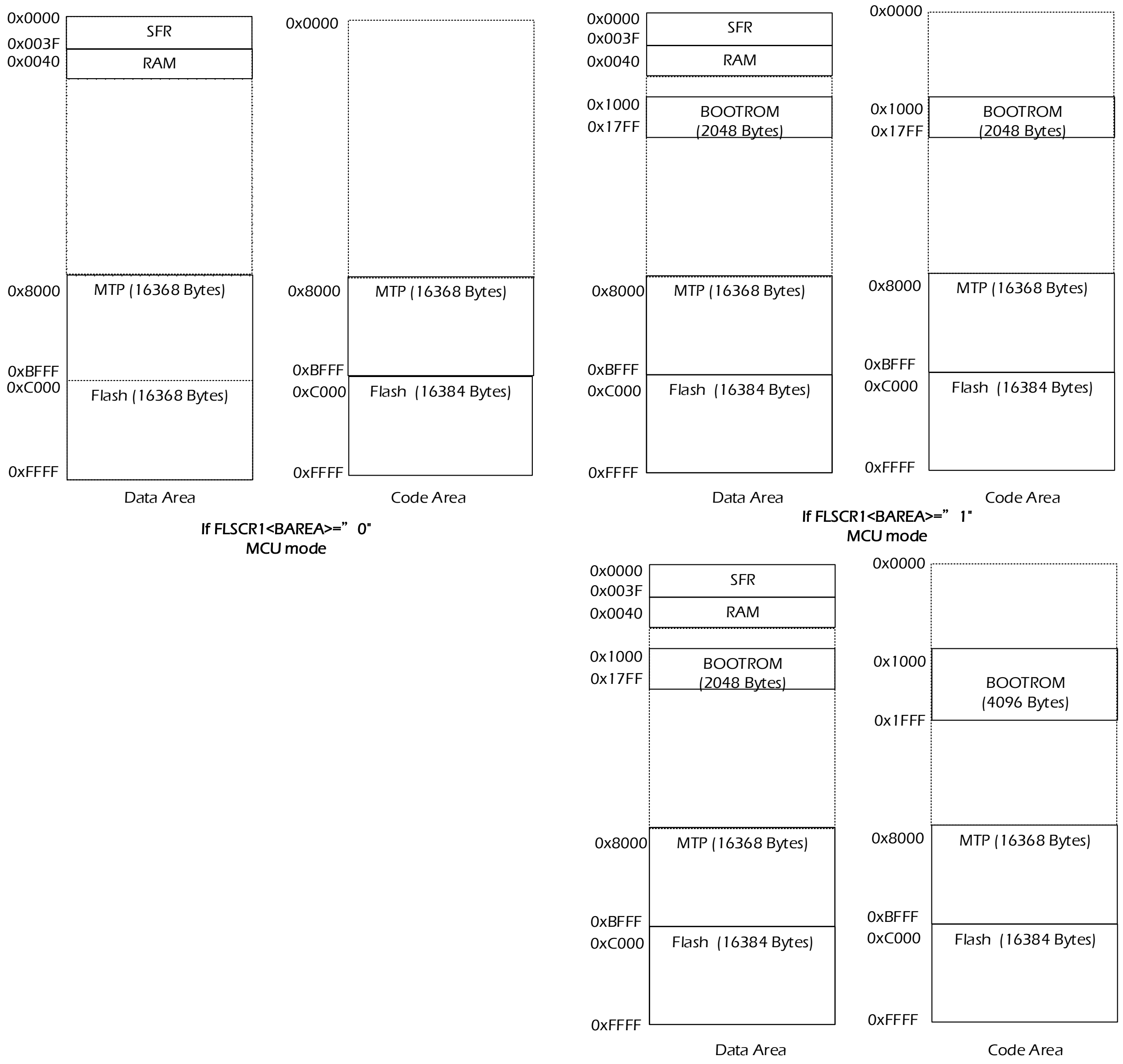
In serial PROM mode, the BOOTROM (mask ROM) is mapped to the 0x1000 through 0x17FF in the data area and 0x1000 through 0x1FFF in the code area respectively.

To write data to or erase data from flash memory by using the RAM loader command (hereafter called the 0x60 command) and an original program, data write or erase operations must be performed while switching between areas by using the flash memory control registers (FLSCR1 and 2). For information on how to specify addresses, refer to Flash Memory.

When the command to write data to flash memory (hereafter called the 0x30 command) or the command to erase data from flash memory (hereafter called the 0xF0 command) is executed, BOOTROM

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

automatically converts addresses. Therefore, as the address of flash memory, specify an address equivalent to that specified in MCU mode (if FLSCR1<BAREA>="0"), namely, 0x8000 through 0xFFFF.



If FLSCR1<BAREA>=" 0"
MCU mode

If FLSCR1<BAREA>=" 1"
MCU mode

If Serial PROM mode

Figure 13.2 Memory Mapping

13.5 Operation Commands

In serial PROM mode, the commands shown in Table 13. 5 are used. After a reset is released, the MQ6935 goes into a standby state and awaits the arrival of matching data 1 (0x86 or 0x30).

Command data	Operation command	Description
0x86 or 0x30	Setup (matching data 1, 2)	After a reset is released, the serial PROM mode always starts operation with this command. If matching data 1 is 0x86, communication starts in the UART format. If matching data 1 is 0x30, communication starts in the SIO format.
0xF0	Flash memory erase	Data in the flash memory area (address 0x8000 through 0xFFFF) can be erased.
0x30	Flash memory write	Data can be written to the flash memory area (address 0x8000 through 0xFFFF).
0x40	Flash memory read	Data can be read from the flash memory area (address 0x8000 through 0xFFFF).
0x60	RAM loader	Data can be written to a specified RAM area (address 0x0060 through 0x083F).
0x90	Flash memory SUM output	0xFF check data and 2-byte checksums of the entire flash memory area (address 0x8000 through 0xFFFF) are output in descending order (from upper to lower bytes).
0xC0	Product ID code output	Product ID codes are output.
0xC3	Flash memory status output	The security program status and other status codes are output.
0xFA	Flash memory security setting	The security program setting is enabled.

Table 13.5 Operation Command in Serial PROM Mode

Each command lists as below:

1. Flash memory erase command
Chip Erase (total erase of flash memory) can be used to erase the data in flash memory. Data in the erased area is 0xFF. To disable the security program setting, execute the flash erase command of Chip Erase. Before erasing the data in flash memory, the MQ6935 performs password authentication except where a product is a blank product or EPFC_OP is 0xFF. If a password is not authenticated, the flash memory erase command is not executed.

2. Flash memory write command
Data can be written in single-byte units to a specified address in flash memory. Provision the external controller so that it transmits data to write as binary data in the Intel Hex format. If errors do not occur until the end record is reached, the MQ6935 calculates checksums in the entire flash memory area (0x8000 through 0xFFFF), and returns the calculation results. If the security program is enabled, the flash memory

write command cannot be executed. In this case, execute Chip Erase beforehand by using the flash memory erase command. Before executing the flash memory write command, the MQ6935 performs password authentication except where a product is a blank product. If a password is not authenticated, the flash memory write command is not executed.

3. Flash memory read command
Data can be read from a specified address in flash memory in single-byte units. Provision the external controller so that it transmits the address in memory where a read starts, as well as the number of bytes. After outputting the number of data equal to the number of bytes, the MQ6935 calculates the checksums of the output data, and returns the calculation results. If the security program is enabled, the flash memory read command cannot be executed. In this case, execute Chip Erase beforehand by using the flash memory erase command. Before executing the flash memory read command, the MQ6935 performs password authentication except where a product is blank. If a password is not authenticated, the flash memory read command is not executed.

4. RAM loader command
The RAM loader transfers the Intel Hex format data sent by the external controller to the built-in RAM. If it completes the data transfer normally, it calculates the checksums, transmits the calculation results, jumps to the RAM address specified by the first data record, and starts to execute the user program. If the security program is enabled, the RAM loader command is not executed. In this case, execute Chip Erase beforehand by using the flash memory erase command. Before executing the RAM loader command, the MCU performs password authentication except where a product is blank. If a password is not authenticated, the RAM loader command is not executed.

5. Flash memory SUM output command
Checksums in the entire flash memory area (0x8000 through 0xFFFF) are calculated, and the calculation results are returned.

6. Product ID code output command
This is a code output used to identify a product. The output code consists of information on the ROM area and on the RAM area respectively. The external controller reads this code to identify the product to which data is to be written.

7.Flash memory status output command
The status of 0xFFE0 through 0xFFFF and that of the security program are output. The external controller reads this code to identify the status of flash memory.

8. Flash memory security setting command
This command is used to prohibit the reading or writing of data in flash memory in parallel mode. In serial PROM mode, the flash memory write command and RAM loader command are prohibited. To disable the flash memory security program, execute Chip Erase by using the flash memory erase command.

13.5.1 Flash memory erase command (0xF0)

Table 13.6 shows the flash memory erase commands.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0xF0)	Baud rate after adjustment	-
	6th byte	-	Baud rate after adjustment	OK: Echo back data (0xF0) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (note 1)
	7th byte	Password count storage address bit 23 to 16	Baud rate after adjustment	-
	8th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	9th byte	Password count storage address bit 15 to 08	Baud rate after adjustment	-
	10th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	11th byte	Password count storage address bit 07 to 00	Baud rate after adjustment	-
	12th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	13th byte	Password comparison start address bit 23 to 16	Baud rate after adjustment	-
	14th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	15th byte	Password comparison start address bit 15 to 08	Baud rate after adjustment	
	16th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	17th byte	Password comparison start address bit 07 to 00	Baud rate after adjustment	-
	18th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	19th byte :	Password string	Baud rate after adjustment	-
	mth byte	-	Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	n-th - 2	Erase area specification	Baud rate after adjustment	-

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

	byte			
	n-th - 1 byte	-	Baud rate after adjustment	OK: Checksum (upper byte) Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: Checksum (lower byte) Error: No data transmitted
	n-th + 1 byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.6 Flash Memory Erase Commands

Note 1: "0x** × 3" means that the device goes into an idle state after transmitting 3 bytes of 0x**.

Note 2: Do not transmit a password string if 0xFFFA of a flash memory is 0xFF, or blank product. (However, the password count storage address and the password comparison start address must be transmitted.)

Note 3: If a value less than 0x20 is transmitted at the n-th - 2 byte (execution of Sector Erase) and if 0xFFFA of flash memory is 0xFF, the product goes into an idle state.

Note 4: When a password error occurs, the product stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the product by using the RESET pin, and restart the serial PROM mode.

Note 5: If a communication error occurs during the transfer of a password address or a password string, the product stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

The flash memory erase command is used to specify an area in flash memory to be erased at n-th-2 byte. If data of more than 0x20 is specified, Chip Erase (total erasure of flash memory) is executed, and the security program in flash memory is disabled. Therefore, to disable the security program in flash memory, execute Chip Erase.

Erase area specification data (data at n-th-2 bytes)

	7	6	5	4	3	2	1	0
Bit Symbol	ERASEC							

ERASEC	Erase area start address	0x00	0x0000 - 0x83FF
		0x01	0x8400 - 0x87FF
		0x02	0x8800 - 0x8BFF
		0x03	0x8C00 - 0x8FFF
		0x04	0x9000 - 0x93FF
		0x05	0x9400 - 0x97FF
		0x06	0x9800 - 0x9BFF
		0x07	0x9C00 - 0x9FFF
		0x08	0xA000 - 0xA3FF
		0x09	0xA400 - 0xA7FF
		0x0A	0xA800 - 0xABFF
		0x0B	0xAC00 - 0xAFFF
		0x0C	0xB000 - 0xB3FF
		0x0D	0xB400 - 0xB7FF
		0x0E	0xB800 - 0xBBFF
		0x0F	0xBC00 - 0xBFFF
		0x10	0xC000 - 0xC3FF
		0x11	0xC400 - 0xC7FF
		0x12	0xC800 - 0xCBFF
		0x13	0xCC00 - 0xCFFF
		0x14	0xD000 - 0xD3FF
		0x15	0xD400 - 0xD7FF
		0x16	0xD800 - 0xDBFF

		0x17 0xDC00 - 0xDFFF 0x18 0xE000 - 0xE3FF 0x19 0xE400 - 0xE7FF 0x1A 0xE800 - 0xEBFF 0x1B 0xEC00 - 0xEFFF 0x1C 0xF000 - 0xF3FF 0x1D 0xF400 - 0xF7FF 0x1E 0xF800 - 0xFBFF 0x1F 0xFC00 - 0xFFFF 0x20 or more Chip Erase (erasure of the entire area)

Note 1: If Sector Erase is performed on an area where flash memory does not exist, the MCU stops communication, and goes into an idle state.

Note 2: If Reserved data is transmitted, the MCU stops communication, and goes into an idle state.

13.5.2 Flash memory write command (operation command: 0x30)

Table 13.7 shows the transfer formats of flash memory write commands.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0x30)	Baud rate after adjustment	OK: Echo back data (0x30)
	6th byte	-	Baud rate after adjustment	- Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3
	7th byte	Password count storage address bit 23 to 16	Baud rate after adjustment	-
	8th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	9th byte	Password count storage address bit 15 to 08	Baud rate after adjustment	-
	10th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	11th byte	Password count storage address bit 07 to 00	Baud rate after adjustment	-
	12th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	13th byte	Password comparison start address bit 23 to 16	Baud rate after adjustment	-
	14th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted

	15th byte	Password comparison start address bit 15 to 08	Baud rate after adjustment	-
	16th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	17th byte	Password comparison start address bit 07 to 00	Baud rate after adjustment	-
	18th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	19th byte :	Password string	Baud rate after adjustment	
	mth byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 1 byte . n-th -3 byte	Intel Hex format (binary)	Baud rate after adjustment	- -
	n-th - 2 byte	-	Baud rate after adjustment	OK: 0x55 Overwrite detect : 0xAA
	n-th - 1 byte	-	Baud rate after adjustment	OK: Checksum(high) Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: Checksum (lower byte) Error: No data transmitted
	n-th + 1 byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.7 Transfer Formats of Flash Memory Write Commands

Note 1: "0x** × 3" means that the device goes into an idle state after transmitting 3 bytes of 0x**.

Note 2: If the area 0xFFE0 through 0xFFFF is all 0xFF, password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even for a blank product. If the password count storage address and/or password comparison start address is/are incorrect, a password error occurs, the product stops communication, and it goes into an idle state. Therefore, if a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 3: If the security program is enabled in flash memory or if a password error occurs, the product stops communication, and goes into an idle state. Therefore, if a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 4: If a communication error occurs during the transfer of a password address or a password string, the product stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 5: If all data in flash memory are the same data, make sure that you never write data to the address 0xFFE0 through 0xFFFF. If data is written to this address, a password error occurs, and the subsequent operations cannot be performed.

Note 6: The n-th-2 byte is a flag for detecting an overwrite. If memory contents at an address where data is to be written are other than 0xFF, the n-th-2 byte is 0xAA (data is not written to this address, and the data write routine is skipped). The checksum at the n-th-1 byte or n-th byte is calculated based on data in which data in memory areas where data was not written are included. Therefore, if an overwrite is detected, the checksum of transmitted data does not match that at the n-th-1 byte or n-th byte.

13.5.3 Flash Memory Read Command (operation command: 0x40)

Table 13.8 shows the transfer formats of flash memory write commands.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0x40)	Baud rate after adjustment	-
	6th byte	-	Baud rate after adjustment	OK: Echo back data (0x40) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3
	7th byte	Password count storage address bit 23 to 16	Baud rate after adjustment	-
	8th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	9th byte	Password count storage address bit 15 to 08	Baud rate after adjustment	-
	10th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	11th byte	Password count storage address bit 07 to 00	Baud rate after adjustment	-
	12th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	13th byte	Password comparison start address bit 23 to 16	Baud rate after adjustment	-
	14th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	15th byte	Password comparison start address bit 15 to 08	Baud rate after adjustment	-
	16th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	17th byte	Password comparison start address bit 07 to 00	Baud rate after adjustment	-
	18th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	19th byte ：	Password string	Baud rate after adjustment	-
	mth byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

	m-th + 1 byte	Read start address bit 23 to 16	Baud rate after adjustment	-
	m-th + 2 byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 3 byte	Read start address bit 15 to 08	Baud rate after adjustment	-
	m-th + 4byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 5byte	Read start address bit 07 to 00	Baud rate after adjustment	-
	m-th + 6 byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 7 byte	Number of bytes to read 23 to16	Baud rate after adjustment	-
	m-th + 8 byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 9 byte	Number of bytes to read 15 to 08	Baud rate after adjustment	-
	m-th + 10 byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 11 byte	Number of bytes to read 07 to 00	Baud rate after adjustment	-
	m-th + 12 byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	m-th + 13 byte . n-th - 2 byte		Baud rate after adjustment Baud rate after adjustment	Memory data Memory data
	n-th - 1 byte	-	Baud rate after adjustment	OK: Checksum(high) Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: Checksum(low) Error: No data transmitted
	n-th + 1 byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.8 Transfer Formats of the Flash Memory Read Command

Note 1: "0x** × 3" means that the device goes into an idle state after transmitting 3 bytes of 0x**.

Note 2: If the area 0xFFE0 through 0xFFFF is all 0xFF, password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even for a blank product. If the password count storage address and/or password comparison start address are/is incorrect, a password error occurs; the product stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the product by using the RESET pin, and restart the serial PROM mode.

Note 3: If the security program is enabled in flash memory or if a password error occurs, the product stops communication, and goes into an idle state. Therefore, if a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 4: If a communication error occurs during the transfer of a password address or a password string, the product stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 5: If the number of bytes received at the m-th + 4 byte, m-th + 5 byte or m-th +6 byte is more than 0x000000 or the size of internal memory, the product stops communication and goes into an idle state.

13.5.4 RAM Loader Command (operation command: 0x60)

Table 13.9 shows the transfer formats of RAM loader command.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0x60)	Baud rate after adjustment	-
	7th byte	Password count storage address 23 to 16	Baud rate after adjustment	-
	8th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	9th byte	Password count storage address 15 to 08	Baud rate after adjustment	-
	10 th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	11 th byte	Password count storage address 07 to 00	Baud rate after adjustment	-
	12 th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	13 th byte	Password comparison start address 23 to 16	Baud rate after adjustment	-
	14 th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	15 th byte	Password comparison start address 15 to 08	Baud rate after adjustment	-
	16 th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	17 th byte	Password comparison start address 07 to 00	Baud rate after adjustment	-
	18 th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	19 th byte : : mth byte	Password string	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	mth+xx byte :	Intel Hex format (binary)	Baud rate after adjustment	-
	nth -2 byte		Baud rate after adjustment	-
	nth -1 byte	-	Baud rate after adjustment	OK: Checksum(high) Error: No data transmitted
	nth byte	-	Baud rate after adjustment	OK: Checksum(low) Error: No data transmitted

Table 13.9 Transfer Formats of the RAM Loader Command

Note 1: "0x** × 3" means that the device goes into an idle state after transmitting 3 bytes of 0x**.

Note 2: If the area 0xFFE0 through 0xFFFF is all 0xFF, password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even for a blank product. If the password count storage address and/or password comparison start address are/is incorrect, a password error occurs; the MCU stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the MCU by using the RESETB pin, and restart the serial PROM mode.

Note 3: After sending a password string, do not send the end record only. If the MCU receives the end record after receiving a password string, it may malfunction.

Note 4: If the security program is enabled in flash memory or if a password error occurs, the MCU stops communication, and goes into an idle state. Therefore, if a password error occurs, initialize the MCU by using the RESETB pin, and restart the serial PROM mode.

Note 5: If a communication error occurs during the transfer of a password address or a password string, the MCU stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the MCU by using the RESETB pin, and restart the serial PROM mode.

13.5.5 Flash Memory SUM Output Command (operation command: 0x90)

Table 13.10 shows the transfer formats of flash memory SUM output command.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0x90)	Baud rate after adjustment	-
	6th byte	-	Baud rate after adjustment	OK: Echo back data (0x90) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3
	7th byte	-	Baud rate after adjustment	0x55 : - 0xAA: All data are 0xFF.
	8th byte	-	Baud rate after adjustment	OK: Checksum(high) Error: No data transmitted
	9th byte	-	Baud rate after adjustment	OK: Checksum(low) Error: No data transmitted
	10th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.10 Transfer Formats of the Flash Memory SUM Output Command

Note 1: "0x** × 3" means that the device goes into an idle state after transmitting 3 bytes of 0x**.

Note 2: If data to be included in the checksum are all 0xFF, the 7th byte becomes 0xAA. If any one piece of data to be included in the checksum is other than 0xFF, the 7th byte becomes 0x55.

13.5.6 Product ID code output command (operation command: 0xC0)

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0xC0)	Baud rate after adjustment	-
	6th byte		Baud rate after adjustment	OK: Echo back data (0xC0) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3
	7th byte		Baud rate after adjustment	0x3A: Start mark
	8th byte		Baud rate after adjustment	0x13: Number of transfer data (from 9th to 27th bytes)
	9th byte		Baud rate after adjustment	0x03: Length of address (3 bytes)
	10 th byte		Baud rate after adjustment	0xFD: Reserved
	11 th byte		Baud rate after adjustment	0x00: Reserved
	12 th byte		Baud rate after adjustment	0x00: Reserved
	13 th byte		Baud rate after adjustment	0x00: Reserved
	14 th byte		Baud rate after adjustment	0x80: ROM size code
	15 th byte		Baud rate after adjustment	0x01: ROM block count (1 block)
	16 th byte		Baud rate after adjustment	0x00: First address of ROM (upper byte)
	17 th byte		Baud rate after adjustment	0x80: First address of ROM (middle byte)
	18 th byte		Baud rate after adjustment	0x00: First address of ROM (lower byte)
	19 th byte		Baud rate after adjustment	0x00:End address of ROM (upper byte)
	20 th byte		Baud rate after adjustment	0xFF: End address of ROM (middle byte)
	21st byte		Baud rate after adjustment	0XFF: End address of ROM (lower byte)
	22nd byte		Baud rate after adjustment	0X00: First address of RAM (upper byte)
	23rd byte		Baud rate after adjustment	0x00: First address of RAM (middle byte)
	24th byte		Baud rate after adjustment	0x60: First address of RAM (lower byte)
	25th byte		Baud rate after adjustment	0x00: End address of RAM (upper byte)
	26th byte		Baud rate after adjustment	0x08: End address of RAM (middle byte)
	27th byte		Baud rate after adjustment	0x3F: End address of RAM (lower byte)
	28th byte		Baud rate after adjustment	0xYY: YYH : Checksum of transfer data (complement of 2 of the sum total from 9th through 27th bytes)
	29th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.11Transfer Formats of Product ID code output command

Note 1: "0x** × 3" means that the device goes into an idle state after transmitting 3 bytes of 0x**.

Note 2: 16th through 21st bytes show the range of addresses in flash memory where data can be written.

Note 3: 22nd through 27th bytes show the flash memory area and RAM area that can be used by the RAM loader. Because the range of addresses shown here does not include the work area used by BOOTROM, it is smaller than the size of a RAM built into an actual product.

ROM Size Code(14th Byte)

	7	6	5	4	3	2	1	0
Bit Symbol	ROMSIZE					"0"	"0"	"0"

The specified value (1000 0000)

ROMSIZE	Data on the flash memory size	00010 : 4Kbytes 00100 : 8Kbytes 01000 : 16Kbytes 10000 : 32Kbytes 11000 : 48Kbytes 11110 : 60Kbytes 10001 : 96Kbytes 11111 : 124Kbytes (Read Only)

13.5.7 Flash Memory Status Output Command (0xC3)

Table 13.10 shows the transfer formats of flash memory status output command.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0xC3)	Baud rate after adjustment	-
	6th byte	-	Baud rate after adjustment	OK: Echo back data (0xC3) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3
	7th byte		Baud rate after adjustment	0x3A :Start mark
	8th byte		Baud rate after adjustment	0x04: Byte cpunt (from 9th through 12th bytes)
	9th byte		Baud rate after adjustment	0x00 to 0x7F: Status code1
	10th byte		Baud rate after adjustment	0x00:Reserved
	11th byte		Baud rate after adjustment	0x00:Reserved
	12th byte		Baud rate after adjustment	0x00:Reserved
	13th byte		Baud rate after adjustment	Checksum (complement of 2 ot the sum total from 9th through 12th bytes)
	14th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.12 Flash Memory Status Output Commands

Note 1: "xxH × 3" means that the device goes into an idle state after transmitting 3 bytes of xxH.

The flash memory status code is 7-byte data. It shows the status of the flash memory security program and that of the address from 0xFFE0 to 0xFFFF.

Data	Description	In the case of MQ6935
1st	Start mark	0x3A
2nd	Number of transfer data (4 bytes from 3rd through 6th bytes)	0x04
3rd	Status code	0x00 through 0x1F (see information below)
4th	Reserved	0x00
5th	Reserved	0x00
6th	Reserved	0x00
7th	Checksum of transfer data (complement of 2 of the sum total of 3rd through 6th bytes)	If 3rd data is 0x00: 0x00 If 3rd data is 0x01: 0xFF If 3rd data is 0x02: 0xFE If 3rd data is 0x03: 0xFD

Table 13.13 Flash Memory Status Code

Status code 1

	7	6	5	4	3	2	1	0	
Bit Symbol					EPFC	DAFC	RPENA	BLANK	Initial value (**** ****)

EPFC	Password string judgment when the flash memory erase command is executed (status of 0xFFFA)	0: To skip the judgment of a password string (to judge PNSA and PCSA only) 1: To judge a password string, PNSA, and PCSA
DAFC	Security program check of the onchip debugging function (OCD) (status of 0xFFFB)	0: To skip the security program check at the start of OCD 1: To perform the security program check at the start of OCD
RPENA	Status of the flash memory security program	0: Status in which the security program is disabled 1: Status in which the security program is enabled
BLANK	Status of 0xFFE0 through 0xFFFF	0: If data in the area 0xFFE0 through 0xFFFF are all 0xFF 1: If data in the area 0xFFE0 through 0xFFFF are other than 0xFF

Restrictions are placed on the execution of some operation commands, depending on the contents of the status code 1. Detailed information on this is shown in the table below. If the security program is enabled, three commands cannot be executed: the flash memory write command, RAM loader mode command, and Sector Erase command. To execute these commands, Chip Erase must be performed on flash memory before they are executed.

RPENA	BLAN	EPFC	DAFC	Flash memory overwrite command, flash memory read command, and RAM loader command	Flash memory SUM output command, product ID output command, and status output command	Flash memory erase command		Flash memory security setting command
						Chip erase	Sector erase	
0	0	0	0	V	V	V	X	X

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

1	0	0	0	X	V	V	X	X
0	1	0	*	Pass	V	V	x	Pass
		1	*	Pass	V	Pass	Pass	Pass
1	1	0	*	X	V	V	X	Pass
		1	*	x	v	Pass	X	Pass

Note: V : A command can be executed.

Pass: A password is required to execute a command.

×: A command cannot be executed.

(After a command is echoed back, the MCU stops communication, and goes into an idle state.)

13.5.8 Flash Memory Security Setting Command (0xFA)

Table 13.11 shows the transfer formats of flash memory security setting command.

	Transfer byte	Transfer data from the external controller to MQ6935	Baud rate	Transfer data from MQ6935 to the external controller
BOOT ROM	1st byte	Matching data 1 (0x86 or 0x30)	Automatic adjustment	- (Automatic baud rate adjustment)
	2nd byte	-	Baud rate after adjustment	OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte	Matching data 2 (0x79 or 0xCF)	Baud rate after adjustment	-
	4th byte	-	Baud rate after adjustment	OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte	Operation command data (0xFA)	Baud rate after adjustment	-
	6th byte	-	Baud rate after adjustment	OK: Echo back data (0xFA) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3
	7th byte	Password count storage address 23 to 16	Baud rate after adjustment	-
	8th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	9th byte	Password count storage address 15 to 08	Baud rate after adjustment	-
	10th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	11th byte	Password count storage address 07 to 00	Baud rate after adjustment	-
	12th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	13th byte	Password comparison start address 23 to 16	Baud rate after adjustment	-
	14th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	15th byte	Password comparison start address 15 to 08	Baud rate after adjustment	-
	16th byte		Baud rate after adjustment	OK: No data transmitted Error: No data transmitted
	17th byte	Password comparison start address 07 to 00	Baud rate after adjustment	-
	18th byte		Baud rate after adjustment	OK: No data transmitted

				Error: No data transmitted
	19th byte :	Password string	Baud rate after adjustment	-
	m-th byte	-	Baud rate after adjustment	OK: 0xFB Error: No data transmitted
	n-th byte	-	Baud rate after adjustment	OK: 0xFB Error: No data transmitted
	n-th +1 byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Table 13.14 Flash Memory Security Setting Command

Note 1: "xxH × 3" means that the device goes into an idle state after transmitting 3 bytes of xxH.

Note 2: If the flash memory security setting command is executed for a blank product or if a password error occurs for a nonblank product, the product stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 3: If a communication error occurs during the transfer of a password address or password string, the product stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the product by using the RESETB pin, and restart the serial PROM mode.

Note 4: If the flash memory security is not enabled, it becomes possible to read ROM data freely in parallel PROM mode. Make sure that you enable the flash memory security in mass production.

13.6 Error Code

Table 13.12 shows the error codes that the MQ6935 transmits when it detects errors. If a password error occurs, the MQ6935 does not transmit an error code.

Data transmitted	Meaning of error data
0x63, 0x63, 0x63	Operation command error
0xA1, 0xA1, 0xA1	Framing error in the received data
0xA3, 0xA3, 0xA3	Overrun error in the received data

Table 13.15 Error Codes

13.7 Checksum

For the following operation commands, a checksum is returned to verify the appropriateness of the result of command execution:

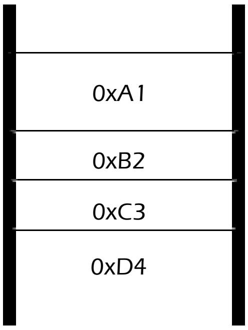
- Flash memory erase command (0xF0)
- Flash memory write command (0x30)
- Flash memory SUM output command (0x90)
- Flash memory read command (0x40)
- Flash memory status output command (0xC3)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

13.7.1 Calculation Method

The checksum is calculated with the sum of all bytes, and the obtained result is returned as a word. The data is read in single-byte units, and the calculated result is returned as a word.

Example:



If the data to be calculated consists of four bytes as the left, the checksum of the data is as follows:

0xA1 + 0xB2 + 0xC3 + 0xD4 = 0x02EA

SUM (HIGH)= 0x02

SUM (LOW)= 0xEA

13.7.2 Calculation data

Operation command	Calculation data	Description
Flash memory erase command	All data in the erased area of flash memory (whole or part of flash memory)	In the case of the chip erase, an entire area of the flash memory is used. When the sector erase is executed, only the erased area is used to calculate the checksum.
Flash memory write command	Data in the entire area of flash memory	Even if a part of the flash memory is written, the checksum of the entire flash memory area (0x8000 to 0xFFFF) is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Flash memory SUM output command	Data in the entire area of flash memory	
Flash memory read command	Data in the read area of flash memory	
Flash memory status output command	9th through 12th bytes of transferred data	For details, refer to Table "Table 13.10 Flash Memory Status Output Commands".

Table 13.16 Data for which a Checksum Is Calculated

13.8 Security

In serial PROM mode, two security functions are provided to prohibit illegal memory access attempts by a third party: password and security program functions.

13.8.1 Passwords

A password is one of the security functions, and can be used when the IC operates in serial PROM mode or when the on-chip debugging function (hereafter called OCD) is used. Specifically, a password can be established by using data (part of user memory) in flash memory. If a password is established, a password authentication process must be performed to execute the flash memory read command, flash memory write command, and other operation commands. In the case of the OCD, the password authentication process is required prior to the start of the OCD system.

In parallel PROM mode, there are no access-related restrictions using a password. To establish the access-related restrictions that work in both serial and parallel PROM modes, the security program must be set to an appropriate setting.

13.8.1.1 How a password can be specified

With the IC, any piece of data in flash memory (8 or more consecutive bytes) can be specified as a password. A password thus specified is authenticated by comparing a password string transmitted by the external controller with the memory data string of MCU where the password is specified. The area where a password can be specified is 0x8000 through 0xFEFF in flash memory.

13.8.1.2 Password structure

A password consists of three components: PNSA, PCSA, and a password string. Figure 23-4 shows the password structure (example of a transmitted password).

・PNSA (password count storage address)
A 3-byte address is specified in the area 0x8000 through 0xFEFF. The memory data of a specified address is the number of bytes of a password string. If the memory data is less than 0x07 or if an address is outside the specified address range, a password error occurs. The memory data specified here is defined as N.

・PCSA (password comparison start address)
A 3-byte address is specified in the area 0x8000 through 0xFEFF-N. An address thus specified is the starting address to be used to compare with a password string. If an address is outside the specified address range, a password error occurs.

・Password string

Data of 8 bytes to 255 bytes (=N) must be specified as a password string. Memory data and a password string are compared by a specified number "N" of bytes; a comparison starts at an address specified by PCSA. If there is a mismatch as a result of this comparison or if data of 3 or more consecutive bytes is specified, a password error occurs, and the IC goes into a idle state. In this idle state, external devices cannot communicate with the IC. To resume communication, the IC must be restarted in serial PROM mode by using the reset pin.

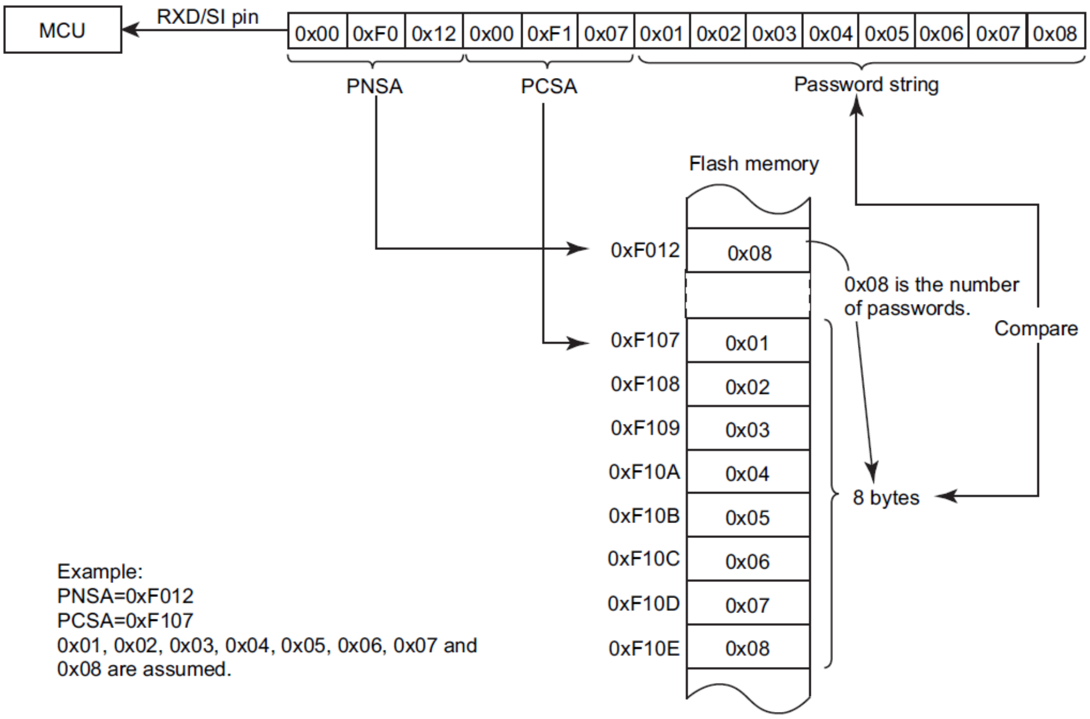


Figure 13.3 Password Structure (Example of a Password Transmitted)

13.8.1.3 Password setting, cancellation and authentication

・Password setting

Because a password is created by using part of a user program, a special password setting routine is unnecessary. A password can be set by simply writing a program to flash memory.

・Password cancellation

To cancel a password, Chip Erase (all erase) must be performed on flash memory. A password is canceled when flash memory is all initialized to 0xFF.

・Password authentication

If there is data other than 0xFF in any one byte of data written to the address 0xFFE0 through 0xFFFF of the IC a product is considered a non-blank product, and password authentication is required to execute an operation command. In this password authentication process, PNSA, PCSA and a password string are used. An operation command is executed only if a password has been successfully authenticated. If a password is unsuccessfully authenticated, the IC goes into an idle state.

If all data written to the address 0xFFE0 through 0xFFFF are 0xFF, a product is considered blank, and no password authentication is performed. To execute some special operation commands, however, PNSA and PCSA are still required (a password string is not required) even if a product is blank. In this case, the addresses defined in Table 13-14 must be selected as PNSA and PCSA.

Whether a product is blank or non-blank can be confirmed by executing the status output command. The operation commands that require PNSA and PCSA (password string) for them to be executed are as follows:

- Flash memory erase command (0xF0)
- Flash memory write command (0x30)
- Flash memory read command (0x40)
- RAM loader command (0x60)
- Flash memory security setting command (0xFA)

13.8.1.4 Password values and setting range

A password must be set in accordance with the conditions shown in Table 23-21. If a password created without meeting these conditions is used, a password error occurs. In this case, the IC does not transmit data and goes into an idle state.

Password	Blank Product (note 1)	Non-blank product
PNSA (password count storage address)	0x8000 ≦PNSA ≦0xFEFF	0x8000 ≦PCSA ≦0xFEFF
PCSA (password comparison start address)	0x8000 ≦PCSA ≦0xFEFF	0x8000 ≦PCSA ≦0xFF00-N
N (password count)	Don’t care	8≦N
Password string	Note required (note3 and 4)	Required (note2)

Table 13.17 Password Values and Setting Range

Note 1: When addresses from 0xFFE0 through 0xFFFF are filled with "0xFF", the product is recognized as a blank product.

Note 2: The data including the same consecutive data (three or more bytes) cannot be used as a password. (A password error occurs during password authentication. The IC does not transmit any data and goes into an idle state.)

Note 3: In flash memory writing mode or RAM loader mode, the blank product receives the Intel Hex format data immediately after receiving PCSA; it does not receive password strings. In this case, the subsequent processing is performed correctly because the IC keeps ignoring incoming data until the start mark (0x3A ":") in the Intel Hex format is detected, even if the external controller transmits the dummy password string. However, if the dummy password string contains "0x3A", it is detected as the start mark erroneously, and the microcontroller enters the halt mode. If this causes a problem, do not transmit the dummy password strings.

Note 4: In executing the flash memory erase command, do not transmit a password string to a blank product.

13.8.2 Security program

The security program can be used in parallel and serial PROM modes and for OCD. It has a special memory for protection, and a special command is required to make this protection setting. If the security program is enabled, the reading or writing of flash memory in parallel PROM mode is prohibited. In serial PROM mode, the read and write of flash memory and other operation commands cannot be used. In performing OCD, two options about system startup are provided: prohibiting the system startup by using an option code and starting the system by password authentication.

13.8.2.1 How the security program functions

With the IC, you can control the read of flash memory by writing protection-related information to a specially-designed memory. Because protection-related information is written to this specially-designed memory, no user memory resource are required.

13.8.2.2 Enabling or disabling the security program

・Enabling the security program

To enable the security program, execute the flash memory security setting command.

・Disabling the security program

To disable the security program, execute Chip Erase of the flash memory erase command.

13.8.3 Option codes

If a specified option code is placed at a specified address inside the interrupt vector area, whether password string authentication is performed or not when executing the flash memory erase command and whether the security program is checked or not when starting OCD can be designated.

- Erase password free code EPFC_OP (0xFFFA)

If changes are frequently made to a program during software development, there are cases in which a password may get lost. In this case, you can cancel the password string authentication of the flash memory erase command (0xF0) by setting the erase password free code (EPFC_OP). EPFC_OP is assigned to 0xFFFA in the vector area. Allocate 0xFF to this EPFC_OP to cancel the password string of the flash memory erase command (0xF0).

It is recommended that the password string authentication of the flash memory erase command (0xF0) be enabled during mass production by allocating data other than 0xFF to EPFC_OP. Only Chip Erase can cancel the password string authentication by using the flash memory erase command. If Sector Erase is executed with EPFC_OP set to 0xFF, the IC goes into an idle state. Commands other than the flash memory erase command cannot cancel the password string authentication.

- OCD security program free code DAFC_OP (0xFFFB)

With the IC, you can enable the security program to prevent illegal access attempts by a third party. If the security program is enabled, restrictions are imposed on operation commands related to memory access, and the startup of OCD.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

The security program should be usually enabled at the time of shipment. If there is the possibility that the OCD may be used by keeping the contents of memory intact, it is possible to directly start the OCD by setting the OCD security program free code (DAFC_OP) and thereby skipping the security program check (the password string authentication, however, is still required).

DAFC_OP is assigned to 0xFFFB in the vector area. To skip the security program check at the startup of the OCD, assign 0xFF to DAFC_OP. In this case, the security program check is not performed, and the OCD can be started by performing only the password string authentication.

If DAFC_OP is not 0xFF, whether the OCD can be used or not is determined by the status of the security program. If the OCD is started with the security program enabled, the IC stops communication and goes into an idle state. To use the OCD when the IC is in this idle state, Chip Erase must be executed for flash memory by using the flash memory erase command (0xF0). If the security program is disabled, the OCD can be started by performing only the password string authentication.

Symbol	Function	Address	Set value
EPFC_OP	Password string authentication when the flash memory erase command is executed	0xFFFA	0xFF : The password string authentication is skipped (only PNSA and PCSA are authenticated). Other than 0xFF: The password string, PNSA, and PCSA are authenticated.
DAFC_OP	Security program check when the OCD is started	0xFFFB	0xFF: The security program check is skipped. Other than 0xFF: The security program check is performed.

Table 23.18 Option Codes

Example: Case in which the password authentication and OCD security program authentication are disabled. Vector Section romdata abs = 0xFFFA

```
DB          0xFF                ;Cancel the password string during the erase operation (EPFC_OP)
DB          0xFF                ;Permit access when OCD is started (DAFC_OP)
```

13.8.4 Recommended settings

Table 13-16 shows the option codes and recommended security program settings.

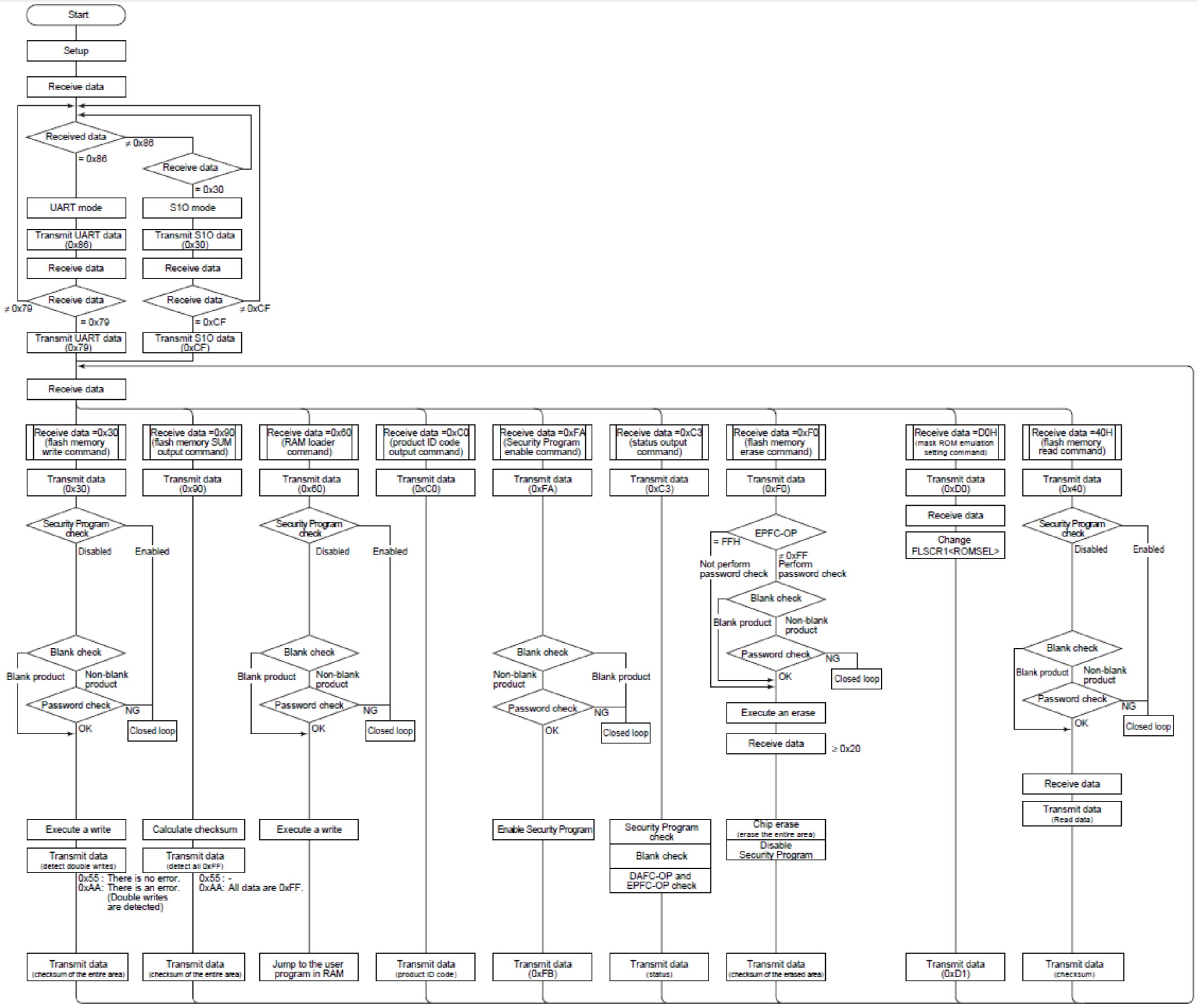
	Device status			Serial PROM mode		Parallel PROM mode		OCD
	EPFC_OP OCD (0xFFFA)	DAFC_OP (0xFFFB)	Security Program	Memory read	Erase	Memory read	Erase	
At the time of debugging during software development	0xFF	0xFF	Disable	Password string required	Possible	Possible	Possible	Can be used
In mass production	0xFF	0xFF	Enable	Impossible	Possible	Impossible	Possible	Can be used
		Other than 0xFF						Cannot be used
	Other than 0xFF	0xFF			Password string required			Can be used
		Other than 0xFF						Cannot be used

Table 13.19 Option Codes and Recommended Security Program Settings

Note 1: In parallel PROM mode, Chip Erase can be performed irrespective of the option code setting.

Note 2: If the security program is not enabled in parallel PROM mode, ROM data can be read with no restrictions. Make sure that in parallel PROM mode, you always enable the security program to protect ROM data.

13.9 Flow Chart


Start
Setup
Receive data
Received data
≠ 0x86
= 0x86
Receive data
= 0x30
UART mode
S1O mode
Transmit UART data (0x86)
Transmit S1O data (0x30)
Receive data
Receive data
Receive data
Receive data
≠ 0x79
≠ 0xCF
= 0x79
= 0xCF
Transmit UART data (0x79)
Transmit S1O data (0xCF)
Receive data
Receive data =0x30 (flash memory write command)
Receive data =0x90 (flash memory SUM output command)
Receive data =0x60 (RAM loader command)
Receive data =0xC0 (product ID code output command)
Receive data =0xFA (Security Program enable command)
Receive data =0xC3 (status output command)
Receive data =0xF0 (flash memory erase command)
Receive data =D0H (mask ROM emulation setting command)
Receive data =40H (flash memory read command)
Transmit data (0x30)
Transmit data (0x90)
Transmit data (0x60)
Transmit data (0xC0)
Transmit data (0xFA)
Transmit data (0xC3)
Transmit data (0xF0)
Transmit data (0xD0)
Transmit data (0x40)
Security Program check
Disabled
Enabled
Blank check
Blank product
Non-blank product
Password check
NG
OK
Closed loop
EPFC-OP
= FFH
Not perform password check
≠ 0xFF Perform password check
Receive data
Change FLSCR1<ROMSEL>
Execute an erase
Receive data
≥ 0x20
Execute a write
Calculate checksum
Execute a write
Enable Security Program
Security Program check
Blank check
DAFC-OP and EPFC-OP check
Chip erase (erase the entire area)
Disable Security Program
Receive data
Transmit data (Read data)
Transmit data (detect double writes)
Transmit data (detect all 0xFF)
0x55 : There is no error.
0xAA: There is an error. (Double writes are detected)
0x55 : -
0xAA: All data are 0xFF.
Transmit data (checksum of the entire area)
Transmit data (checksum of the entire area)
Jump to the user program in RAM
Transmit data (product ID code)
Transmit data (0xFB)
Transmit data (status)
Transmit data (checksum of the erased area)
Transmit data (0xD1)
Transmit data (checksum)

13.10 Intel Hex Format (Binary)

For the following two commands, the Intel Hex format is used in part of the transfer format:
- Flash memory write command (0x30)
- RAM loader command (0x60)

For information on the definition of the Intel Hex format, refer to Table as below.
Data is in binary form. The start mark ":" must be transmitted as binary data of 0x3A.

	(1)	(2)	(3)	(4)	(5)	(6)
	Start mark	Data length (1 byte)	Offset address (2 bytes)	Record type (1 byte)	Data	Checksum (1 byte)
Data record (record type = 00)	3A	Number of data in a data field	Starting byte storage Address *Specified using bigendian	00	Data (1 to 255 bytes)	(2) Data length (3) Offset address (4) Record type (5) Data Complement of 2 of the sum total of the above
End record (record type = 01)	3A	00	00 00	01	None	(2) Data length (3) Offset address (4) Record type Complement of 2 of the sum total of the above
Extended record (record type = 02)	3A	02	00 00	02	Segment address (2 bytes) *Specified using bigendian	(2) Data length (3) Offset address (4) Record type (5) Segment address Complement of 2 of the sum total of the above

Table 23.20 Definition of the Intel Hex Format

1. After receiving the checksum of each data record, the MCU goes into a wait state and awaits the arrival of the start mark (0x3A ":") of the next data record. Although the external controller transmits data other than 0x3A between records, the MCU ignores such data when it is in this wait state.

2. The external controller must be provisioned so that after it transmits the checksum of end record, it goes into a wait state and does not transmit any data until the arrival of 3-byte data (overwrite detection, upper and lower bytes of the checksum). (3-byte data is used if the flash memory write command is used. If the RAM loader command is used, the external controller awaits the arrival of 2-byte data, or upper and lower bytes of the checksum.)

3. If a receiving error or Intel Hex format error occurs, the MCU goes into an idle state without returning an error code to the external controller. The Intel Hex format error occurs in the following cases:
- If the record type is other than 00h, 01h, or 02h
- If a checksum error of the Intel Hex format occurs
- If the data length of an extended record (record type = 0x02) is not 0x02
- If the MCU receives the data record after receiving an extended record (record type = 0x02) whose segment address is more than 0x2000
- I the data length of the end record (record type = 0x01) is not 0x00
- If the offset address of an extended record (record type = 0x02) is not 0x0000

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

14 Serial Bus Interface(SBI)/ I2C

MQ6935 contains 1 channels of serial bus interface(SBI).

The serial bus interface supports serial communication conforming to the I2C bus standards. It has clock synchronization and arbitration functions, and supports the multi-master in which multiple masters are connected on a bus. It also supports the unique free data format.

14.1 Communication Format

14.1.1 I2C bus

The I2C bus is connected to devices via the SDA0 and SCL0 pins and can communicate with multiple devices.

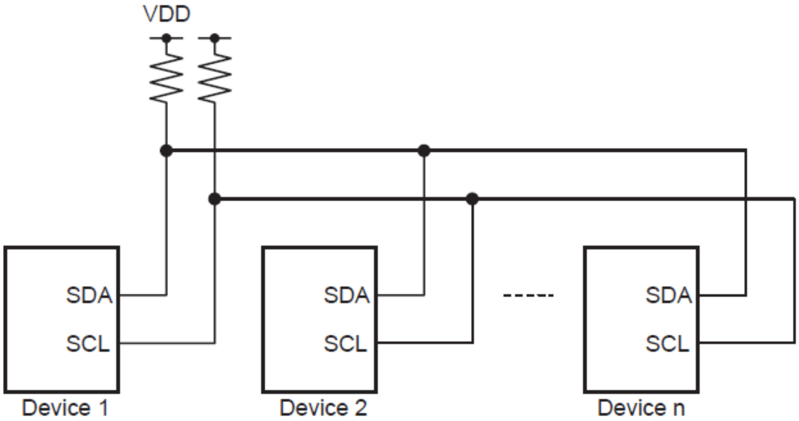


Figure 14.1 Device Connections

Communications are implemented between a master and slave.

The master transmits the start condition, the slave addresses, the direction bit and the stop condition to the slave(s) connected to the bus, and transmits and receives data. The slave detects these conditions transmitted from the master by the hardware, and transmits and receives data. The data format of the I2C bus that can communicate via the serial bus interface is shown in the figure as below.

The serial bus interface does not support the following functions among those specified by the I2C bus standards:

1. Start byte
2. 10-bit addressing
3. SDA and SCL pins falling edge slope control

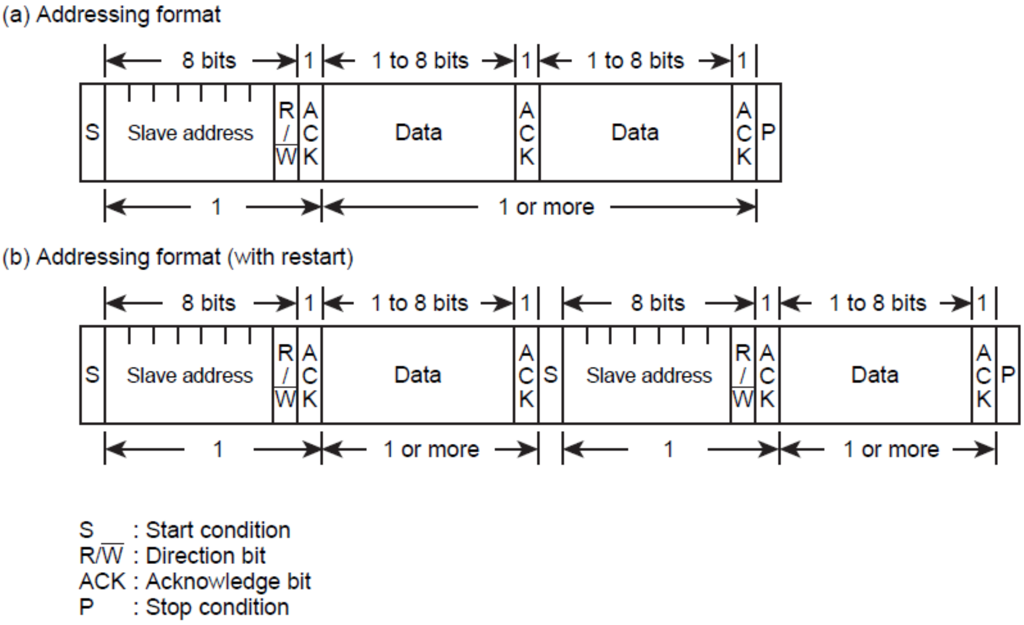


Figure14.2 Data format if I2C bus

14.1.2 Free data format

The free data format is for communication between a master and slave.
In the free data format, the slave address and the direction bit are processed as data.

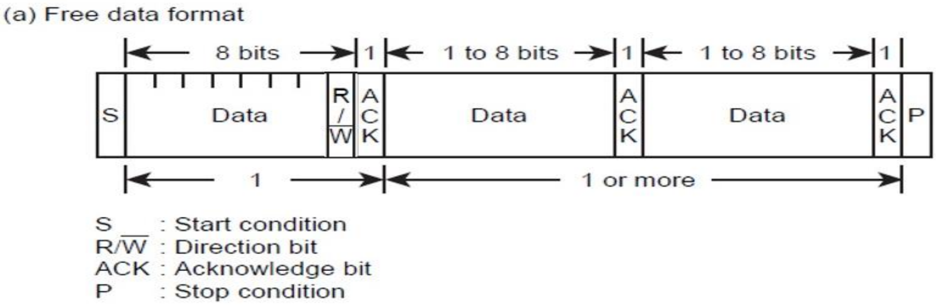


Figure 14.3 Free Data Format

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

14.2 Configuration

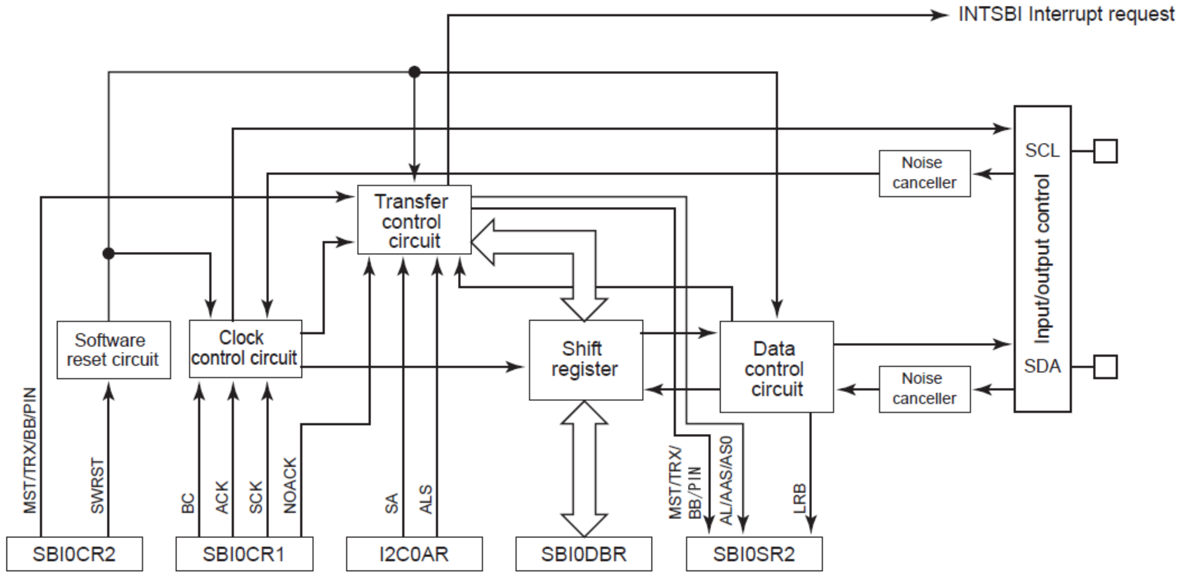


Figure 14.4 Serial Bus Interface0 (SBI0)

14.3 Control

The following registers are used to control the serial bus interface and monitor the operation status.

- Serial bus interface control register 1 ((SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface status register 2 (SBI0SR2)
- Serial bus interface data buffer register (SBI0BR)
- I2C bus address register (I2C0AR)

In addition, the serial bus interface has low power consumption registers that save power when the serial bus interface is not being used.

Low Power Consumption Register 1

POFFCR1 (0x0F75)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 control	0: Disable 1: Enable
UART2EN	UART2 control	0: Disable 1: Enable
UART1EN	UART1 control	0: Disable 1: Enable

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

UART0EN	UART0 control	0: Disable 1: Enable

Note: When SBI0EN is cleared to "0", the clock supply to the serial bus interface is stopped. At this time, the data written to the serial bus interface control registers is invalid. When the serial bus interface is used, set SBI0EN to "1" and then write the data to the serial bus interface control registers.

Serial bus interface control register 1

SBI0CR1 (0x0022)	7	6	5	4	3	2	1	0
Bit Symbol	BC			ACK	NOACK	SCK		
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

BC	Number of data bits	BC	ACK=0		ACK=1	
			Number of clocks for data transfer	Number of data bits	Number of clocks for data transfer	Number of data bits
		000	8	8	9	8
		001	1	1	2	1
		010	2	2	3	2
		011	3	3	4	3
		100	4	4	5	4
		101	5	5	6	5
		110	6	6	7	6
		111	7	7	8	7
ACK	Generation and counting of the clocks for an acknowledge signal	ACK	Master mode		Slave mode	
		0:	Not generating the clocks for an acknowledge signal. Generate an interrupt request when the data transfer is finished (non-acknowledgement mode)		Generate an interrupt request when the data transfer is finished (non-acknowledgement mode)	
		1:	Generate the clocks for an acknowledge signal and an interrupt request when the data transfer is finished (acknowledgement mode)		Count the clocks for an acknowledge signal and generate an interrupt request when the data transfer is finished (acknowledgement mode)	
NOACK	Enables/disables the slave address match detection and the GENERAL CALL detection	NOACK	Master mode		Slave mode	
		0:	Don't Care		Enable the slave address match detection and the GENERAL CALL detection	
		1:	Don't Care		Disable the slave address match detection and the GENERAL CALL detection	
SCK	HIGH and LOW periods of the serial clock in the master mode Time before the release of the SCL pin in the slave mode	SCK	t_{HIGH}(m/fcgck)	t_{LOW}(n/fcgck)	fscl@fcgck=8MHz	
			m	n		
		000	9	12	381KHz	
		001	11	14	320KHz	
		010	15	18	242KHz	
		011	23	26	163KHz	
		100	39	42	99KHz	
		101	71	74	55KHz	

		110	*135*	*138*	*29KHz*
		111	*263*	*266*	*15KHz*

Note 1: fcgck = Gear clock [Hz], fs = Low-frequency clock [Hz] 。

Note2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Note 5: When fcgck is 4MHz, SCK should be not set to 0y000, 0y001 or 0y010 because it is not possible to satisfy the bus specification of fast mode.

Serial bus interface control register2

SBI0CR2 (0x0023)	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	SBIM	-	SWRST	
Read/Write	W	W	W	W	W	R	W	W
After reset	0	0	0	1	0	0	0	0

MST	Master/slave selection	0:Slave 1:Master
TRX	Transmitter/receiver selection	0:Receiver 1:Transmitter
BB	Start/stop generation	0:Generate the stop condition(when MST、TRX and PIN are"1") 1: Generate the start condition (when MST、TRX and PIN are"1")
PIN	Cancel interrupt service request	0:- (cannot clear this bit by software) 1: Cancel interrupt service request
SBIM	Serial bus interface operation mode register	0: Port mode 1:Serial bus interface mode
SWRST	Software reset start bit	The software reset starts by first writing "10" and next writing"01"

Note 1: When SBI0CR2<SBIM> is "0", no value can be written to SBI0CR2 except SBI0CR2<SBIM>. Before writing values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

Note 2: Don't change the contents of the registers, except SBI0CR2<SWRST>, when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: Make sure that the port is in a high state before switching the port mode to the serial bus interface mode. Make sure that the bus is free before switching the serial bus interface mode to the port mode.

Note 4: SBI0CR2 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 5: After a software reset is generated, all the bits of SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 6: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Serial bus interface status register2

SBI0SR2 (0x0023)	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	1	0	0	0	*

MST	Master/slave selection status monitor	0:Slave 1:Master
TRX	Transmitter/receiver selection status monitor	0:Receiver 1:Transmitter
BB	Bus status monitor	0:Bus free 1: Bus busy
PIN	Interrupt service requests status monitor	0:Requesting interrupt service 1:Releasing interrupt service
AL	Arbitration lost detection monitor	0: - 1:Aritration lost detected
AAS	Slave address match detection monitor	0: - 1:Detect slave address match or "GENERAL CALL"
AD0	"GENERAL CALL" detection monitor	0: - 1: Detect "GENERAL CALL"
LRB	Last received bit monitor	0: Last received bit is"0" 1: Last received bit is"1"

*Note 1: * Unstable*

Note 2: When SBI0CR2<SBIM> becomes "0", SBI0SR is initialized.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

I^{2}C bus address register

I2C0AR (0x0024)	7	6	5	4	3	2	1	0
Bit Symbol	SA							ALS
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

SA	Slave address setting	Slave address in the slave mode
ALS	Communication format selection	0: I^{2}C bus mode 1: Free data format

Note 1: Don't set I2C0AR<SA> to "0x00". If it is set to "0x00", the slave address is deemed to be matched when the I2C bus standard start byte ("0x01") is received in the slave mode.

Note 2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Serial bus interface data buffer register

SBI0DBR (0x0025)	7	6	5	4	3	2	1	0
Bit Symbol	SBI0DBR							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1 : Write the transmit data beginning with the most significant bit (bit 7).

Note 2: SBI0DBR has individual writing and reading buffers, and written data cannot be read out. Therefore, SBI0DBR must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 3:: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 4: To set SBI0CR2<PIN> to "1" by writing the dummy data to SBI0DBR, write 0x00. Writing any data other than 0x00 causes an improper value in the subsequently received data.

Note 5: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

14.4 Functions

14.4.1 Low power consumption function

The serial bus interface has a low power consumption register (POFFCR1) that saves power when the serial bus interface is not being used.

Setting POFFCR1<SBI0EN> to "0" disables the basic clock supply to the serial bus interface to save power. Note that this makes the serial bus interface unusable. Setting POFFCR1<SBI0EN> to "1" enables the basic clock supply to the serial bus interface and makes external interrupts usable.

After reset, POFFCR1<SBI0EN> is initialized to "0", and this makes the serial bus interface unusable. When using the serial bus interface for the first time, be sure to set POFFCR1<SBI0EN> to "1" in the initial setting of the program (before the serial bus interface control registers are operated).

Do not change POFFCR1<SBI0EN> to "0" during the serial bus interface operation, otherwise serial bus interface may operate unexpectedly.

14.4.2 Selecting the slave address match detection and the GENERAL CALL detection

SBI0CR1<NOACK> enables and disables the slave address match detection and the GENERAL CALL detection in the slave mode.

C

learing SBI0CR1<NOACK> to "0" enables the slave address match detection and the GENERAL CALL detection.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections.

The slave addresses and "GENERAL CALL" sent from the master are ignored. No acknowledgement is returned and no interrupt request is generated.
In the master mode, SBI0CR1<NOACK> is ignored and has no influence on the operation.

Note: If SBI0CR1<NOACK> is cleared to "0" during data transfer in the slave mode, it remains at "1" and returns an acknowledge signal of data transfer.

14.4.3 Selecting the number of clocks for data transfer and selecting the acknowledgement or non-acknowledgement mode

1-word data transfer consists of data and an acknowledge signal. When the data transfer is finished, an interrupt request is generated.

SBI0CR1<BC> is used to select the number of bits of data to be transmitted/received subsequently.
The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

The master device generates the clocks for an acknowledge signal and outputs an acknowledge signal in the receiver mode. The slave device counts the clocks for an acknowledge signal and outputs an acknowledge signal in the receiver mode.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

The master device does not generate the clocks for an acknowledge signal. The slave device does not count the clocks for an acknowledge signal.

14.4.3.1 Number of clock for data transfer

The number of clocks for data transfer is set by using SBI0CR1<BC> and SBI0CR1<ACK>.
The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

In the acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, generates the clocks for an acknowledge signal, and generates an interrupt request.
The slave device counts the clocks that correspond to the data bits, counts the clocks for an acknowledge signal, and generates an interrupt request.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".
In the non-acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, and generates an interrupt request.
The slave device counts the clocks that correspond to the data bits, and generates an interrupt request.

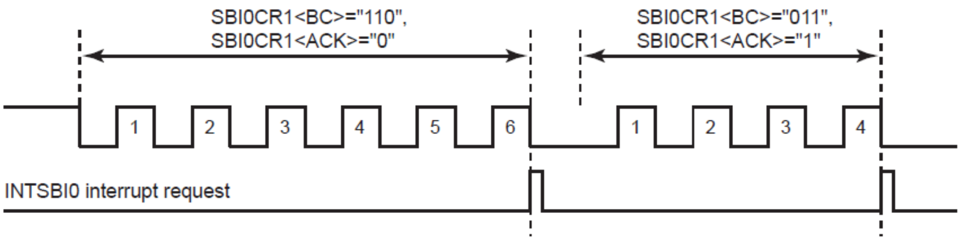
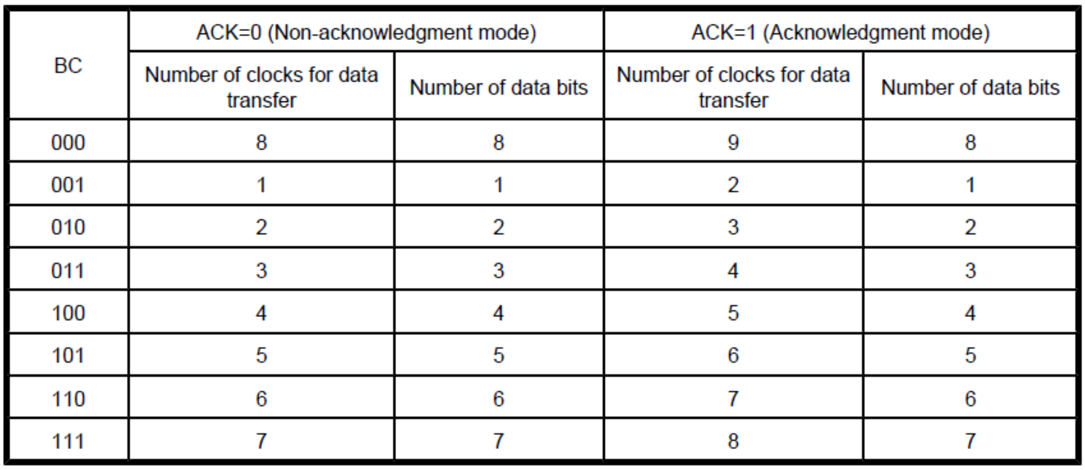


Figure 14.5 Number of clocks for Data transfer and SBI0CR1<BC> and SBI0CR1<ACK>

The relationship between the number of clocks for data transfer and SBI0CR1<BC> and SBI0CR1<ACK> is shown in Table 14.1



BC	ACK=0 (Non-acknowledgment mode)		ACK=1 (Acknowledgment mode)	
	Number of clocks for data transfer	Number of data bits	Number of clocks for data transfer	Number of data bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Table 14.1 Relationship between the Number of Clocks for Data Transfer and SBI0CR1<BC> and SBI0CR1<ACK>

BC is cleared to "000" by the start condition. Therefore, the slave address and the direction bit are always transferred in 8-bit units. In other cases, BC keeps the set value.

Note: SBI0CR1<ACK> must be set before transmitting or receiving a slave address. When SBI0CR1<ACK> is cleared, the slave address match detection and the direction bit detection are not executed properly.

14.4.3.2 Output of an acknowledge signal

In the acknowledgment mode, the SDA0 pin changes as follows during the period of the clocks for an acknowledge signal.

(a) In the master mode

In the transmitter mode, the SDA0 pin is released to receive an acknowledge signal from the receiver during the period of the clocks for an acknowledge signal. In the receiver mode, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated during the period of the clocks for an acknowledge signal.

(b) In the slave mode

When a match between the received slave address and the slave address set to I2C0AR<SA> is detected or when a GENERAL CALL is received, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated during the period of the clocks for an acknowledge signal.

During the data transfer after the slave address match is detected or a "GENERAL CALL" is received in the transmitter mode, the SDA0 pin is released to receive an acknowledge signal from the receiver during the period of the clocks for an acknowledge signal.

In the receiver mode, the SDA0 pin is pulled down to the low level and an acknowledge signal

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

is generated. Table 19-2 shows the states of the SCL0 and SDA0 pins in the acknowledgment mode.

Note: In the non-acknowledgment mode, the clocks for an acknowledge signal are not generated or counted, and thus no acknowledge signal is output.

Mode	Pin	Condition	Transmitter	Receiver
Master	SCL0	-	Add the clocks for an acknowledge signal.	Add the clocks for an acknowledge signal
	SDA0	-	Release the pin to receive an acknowledge signal	Output the low level as an acknowledge signal to the pin
Slave	SCL0	-	Count the clocks for an acknowledge signal	Count the clocks for an acknowledge signal
	SDA0	When the slave address match is detected or a "GENERAL CALL" is received	-	Output the low level as an acknowledge signal to the pin
		During transfer after the slave address match is detected or a "GENERAL CALL" is received	Release the pin to receive an acknowledge signal	Output the low level as an acknowledge signal to the pin

Table 14.2 States of the SCL0 and SDA0 pins in the acknowledgment mode

14.4.4 Serial clock

14.4.4.1 Clock source

SBI0CR1<SCK> is used to set the HIGH and LOW periods of the serial clock to be output in the master mode.

SCK	t_{HIGH}(m/fcgck)	t_{LOW}(n/fcgck)
	m	n
000:	9	12
001:	11	14
010:	15	18
011:	23	26
100:	39	42
101:	71	74
110:	135	138
111:	263	266

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

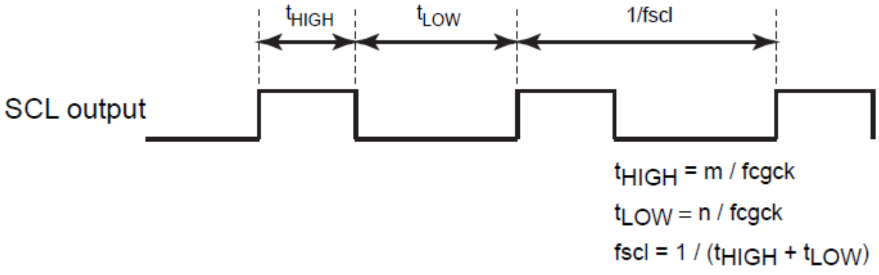


Figure14.6 SCL output

Note: There are cases where the HIGH period differs from tHIGH selected at SBI0CR1<SCK> when the rising edge of the SCL pin becomes blunt due to the load capacity of the bus.

In the master mode, the hold time when the start condition is generated is tHIGH [s] and the setup time when the stop condition is generated is t_{HIGH} [s].

When SBI0CR2<PIN> is set to "1" in the slave mode, the time that elapses before the release of the SCL pin is t_{LOW} [s].

In both the master and slave modes, the high level period must be 3/fcgck[s] or longer and the low level period must be 5/fcgck[s] or longer for the externally input clock, regardless of the SBI0CR1<SCK> setting.

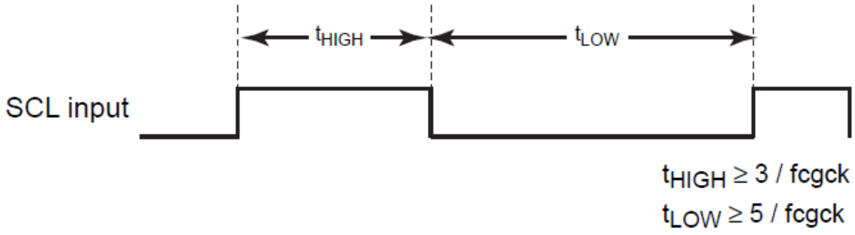


Figure14.7 SCL input

14.4.4.2 Clock synchronization

In the I2C bus, due to the structure of the pin, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate the clock pulse of another master device which generates a high-level clock pulse. Therefore, the master outputting the high level must detect this to correspond to it.

The serial bus interface circuit has a clock synchronization function. This function ensures normal transfer even if there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

a bus.

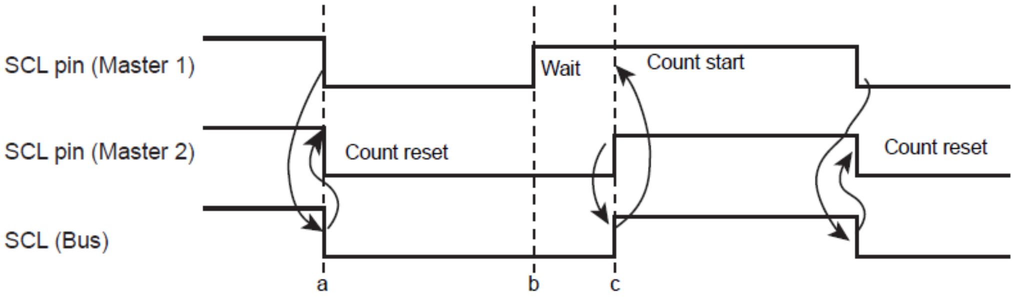


Figure14.8 Example of clock synchronization

As Master 1 pulls down the SCL pin to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point "b" and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point "c" and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level. Then, the master, which has finished the counting a clock pulse in the high level, pulls down the SCL pin to the low level.

The clock pulse on the bus is determined by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

14.4.5 Master/slave selection

To set a master device, SBI0CR2<MST> should be set to "1".

To set a slave device, SBI0CR2<MST> should be cleared to "0". When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<MST> is cleared to "0" by the hardware.

14.4.6 Transmitter/receiver selection

To set the device as a transmitter, SBI0CR2<TRX> should be set to "1". To set the device as a receiver, SBI0CR2<TRX> should be cleared to "0".

For the I2C bus data transfer in the slave mode, SBI0CR2<TRX> is set to "1" by the hardware if the direction bit (R/W) sent from the master device is "1", and is cleared to "0" if the bit is "0".

In the master mode, after an acknowledge signal is returned from the slave device, SBI0CR2<TRX> is cleared to "0" by hardware if a transmitted direction bit is "1", and is set to "1" by hardware if it is "0".

When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<TRX> is cleared to "0" by the hardware. The table shows SBI0CR2<TRX> changing conditions in each mode and SBI0CR2<TRX> value after changing.

Note:: When SBI0CR1<NOACK> is "1", the slave address match detection and the GENERAL CALL detection are disabled, and thus SBI0CR2<TRX> remains unchanged.

Mode	Direction bit	Changing condition	TRX after changing
Slave mode	"0"	A received slave address is the same as the value set to I2C0AR<SA>	"0"
	"1"		"1"
Master mode	"0"	ACK signal is returned	"1"
	"1"		"0"

Table 14.3 SBI0CR1<TRX> Operation in Each Mode

When the serial bus interface circuit operates in the free data format, a slave address and a direction bit are not recognized. They are handled as data just after generating the start condition. SBI0CR2<TRX> is not changed by the hardware.

14.4.7 Start/stop condition generation

When SBI0SR2<BB> is "0", a slave address and a direction bit which are set to the SBI0DBR are output on a bus after generating a start condition by writing "1" to SBI0CR2 <MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN>. It is necessary to set SBI0CR1<ACK> to "1" before generating the start condition.

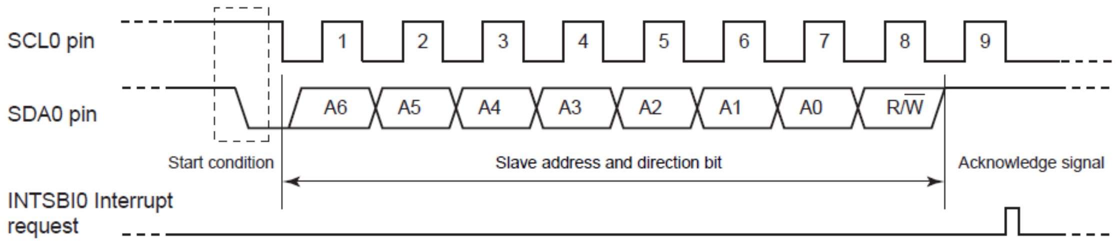


Figure14.9 Generating the start condition and a slave address

When SBI0CR2<BB> is "1", the sequence of generating the stop condition on the bus is started by writing "1" to SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<PIN> and writing "0" to SBI0CR2<BB>.

When a stop condition is generated. The SCL line on a bus is pulled down to the low level by another device, a stop condition is generated after releasing the SCL line.

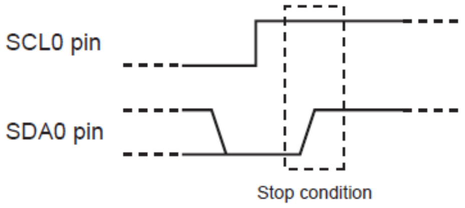


Figure14.10 Stop condition generation

The bus condition can be indicated by reading the contents of SBI0SR2<BB>. SBI0SR2<BB> is set to "1" when the start condition on the bus is detected (Bus Busy State) and is cleared to "0" when the stop condition is detected (Bus Free State).

14.4.8 Interrupt service request and release

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by SBI0CR1<BC> and SBI0CR1<ACK> is complete, a serial bus interface interrupt request (INTSBI0) is generated.

In the slave mode, a serial bus interface interrupt request (INTSBI0) is generated when the above and following conditions are satisfied:

- At the end of the acknowledge signal when the received slave address matches to the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge signal when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of the slave address or receiving of "GENERALCALL"

When a serial bus interface interrupt request occurs, SBI0CR2<PIN> is cleared to "0". During the time that SBI0CR2<PIN> is "0", the SCL0 pin is pulled down to the low level.

Writing data to SBI0DBR sets SBI0CR2<PIN> to "1". The time from SBI0CR2<PIN> being set to "1" until the SBI0 pin is released takes t_{LOW}. Although SBI0CR2<PIN> can be set to "1" by the software, SBI0CR2<PIN> can not be cleared to "0" by the software.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

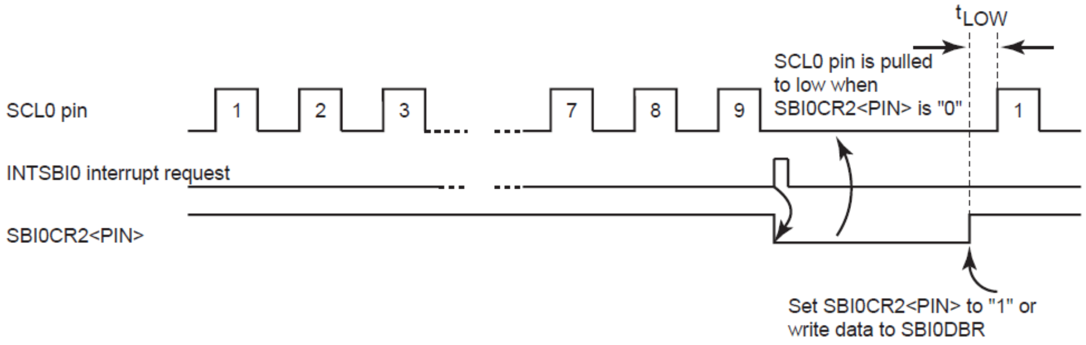


Figure14.11 SBI0CR2<PIN> and SCL0 pin

14.4.9 Setting of serial bus interface mode

SBI0CR2<SBIM> is used to set serial bus interface mode.

Setting SBI0CR2<SBIM> to "1" selects the serial bus interface mode. Setting it to "0" selects the port mode. Set SBI0CR2<SBIM> to "1" in order to set serial bus interface mode. Before setting of serial bus interface mode, confirm serial bus interface pins in a high level, and then, write "1" to SBI0CR2<SBIM>. And switch a port mode after confirming that a bus is free and set SBI0CR2<SBIM> to "0".

Note: When SBI0CR2<SBIM> is "0", no data can be written to SBI0CR2 except SBI0CR2<SBIM>. Before setting values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

14.4.10 Software reset

The serial bus interface circuit has a software reset function that initializes the serial bus interface circuit. If the serial bus interface circuit locks up, for example, due to noise, it can be initialized by using this function. A software reset is generated by writing "10" and then "01" to SBI0CR2<SWRST>.

After a software reset is generated, the serial bus interface circuit is initialized and all the bits of SBI0CR2 register, except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR<SA> and SBI0SR2 registers, are initialized.

14.4.11 Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I2C bus. The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Master 2 output the same data until point "a". After that, when Master 1 outputs "1" and Master 2 outputs "0", since the SDA line of a bus is wired AND, the SDA line is pulled down to the low level by Master 2. When the SCL line of a bus is pulled-up at point "b", the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

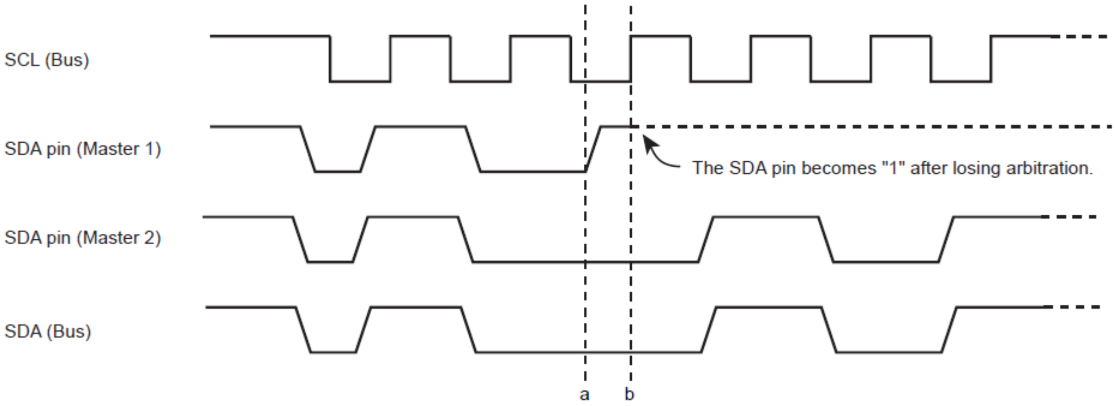


Figure14.12 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and SBI0SR2<AL> is set to "1".

When SBI0SR2<AL> is set to "1", SBI0CR2<MST> and SBI0CR2<TRX> are cleared to "0" and the mode is switched to a slave receiver mode. Thus, the serial bus interface circuit stops output of clock pulses during data transfer after the SBI0SR2<AL> is set to "1". After the data transfer is completed, SBICR2<PIN> is cleared to "0" and the SCL pin is pulled down to the low level.

SBI0SR2<AL> is cleared to "0" by writing data to the SBI0DBR, reading data from the SBI0DBR or writing data to the SBI0CR2.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

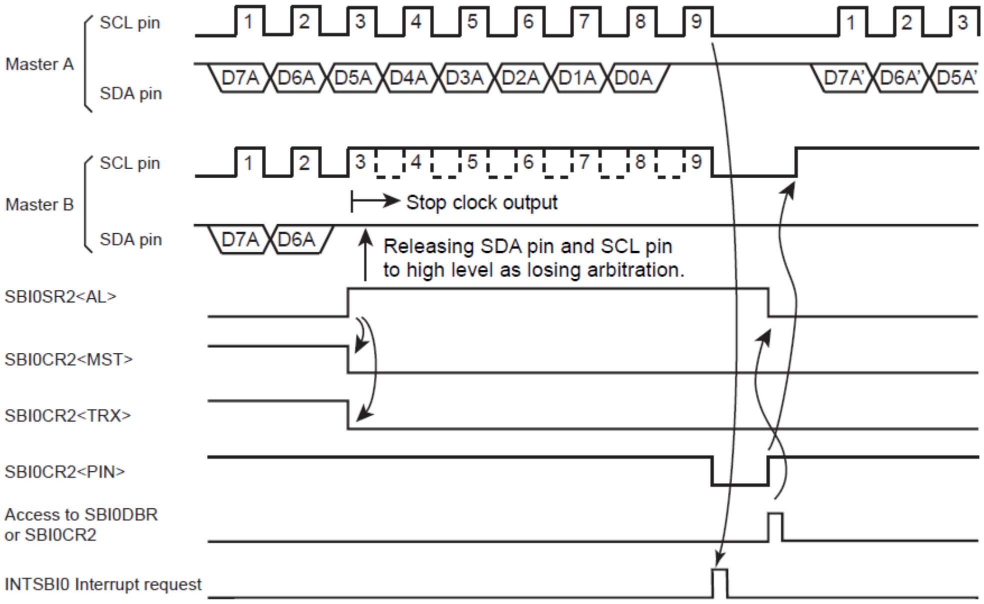

Master A
SCL pin
SDA pin
1 2 3 4 5 6 7 8 9 1 2 3
D7A D6A D5A D4A D3A D2A D1A D0A D7A' D6A' D5A'
Master B
SCL pin
SDA pin
1 2 3 4 5 6 7 8 9
D7A D6A
Stop clock output
Releasing SDA pin and SCL pin to high level as losing arbitration.
SBI0SR2<AL>
SBI0CR2<MST>
SBI0CR2<TRX>
SBI0CR2<PIN>
Access to SBI0DBR or SBI0CR2
INTSBI0 Interrupt request

Figure14.13 Example when Master B is a serial bus interface circuit

14.4.12 Slave address match detection monitor

In the slave mode, SBI0SR2<AAS> is set to "1" when the received data is "GENERAL CALL" or the received data matches the slave address setting by I2C0AR<SA> with SBI0CR1<NOACK> set at "0" and the I2C bus mode is active (I2C0AR<ALS>="0").

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AAS> remains at "0" even if a "GENERAL CALL" is received or the same slave address as the I2C0AR<SA> set value is received.

When a serial bus interface circuit operates in the free data format (I2C0AR<ALS>= "1"), SBI0SR2<AAS> is set to "1" after receiving the first 1-word of data. SBI0SR2<AAS> is cleared to "0" by writing data to the SBI0DBR or reading data from the SBI0DBR.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

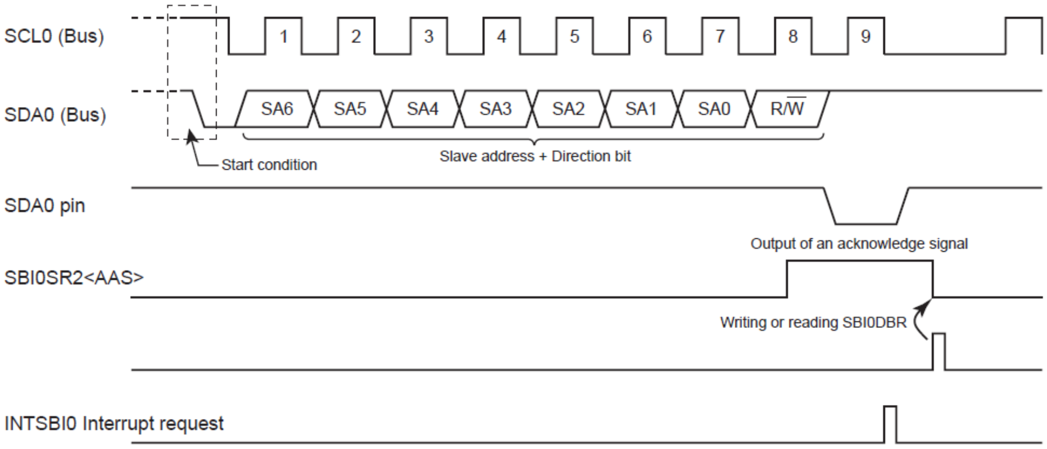

SCL0 (Bus)
1
2
3
4
5
6
7
8
9
SDA0 (Bus)
SA6
SA5
SA4
SA3
SA2
SA1
SA0
R/W
Start condition
Slave address + Direction bit
SDA0 pin
Output of an acknowledge signal
SBI0SR2<AAS>
Writing or reading SBI0DBR
INTSBI0 Interrupt request

Figure14.14 Change in the slave address match detection monitor

14.4.13 GENERAL CALL detection monitor

SBI0SR2<AD0> is set to "1" when SBI0CR1<NOACK> is "0" and GENERAL CALL (all 8-bit received data is "0" immediately after a start condition) in a slave mode.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AD0> remains at "0" even if a "GENERAL CALL" is received.

SBI0SR2<AD0> is cleared to "0" when a start or stop condition is detected on a bus.

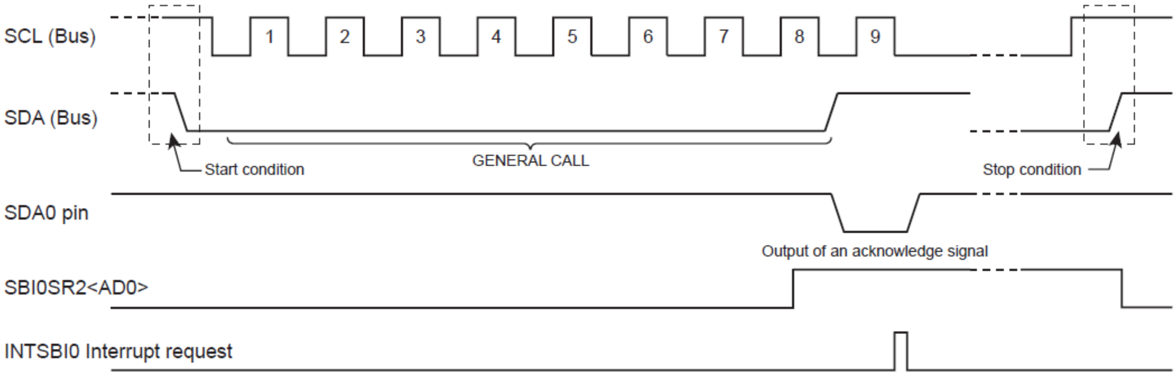


Figure14.15 Changes in the GENERAL CALL detection monitor

14.4.14 Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to SBI0SR2<LRB>.

In the acknowledge mode, immediately after an interrupt request is generated, an acknowledge signal is read by reading the contents of SBI0SR2<LRB>.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

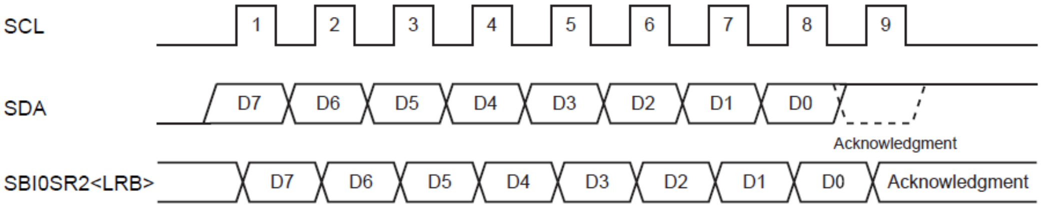


Figure14.16 Changes in the Last received bit monitor

14.4.15 Slave address and address recognition mode specification

When the serial bus interface circuit is used in the I2C bus mode, clear I2C0AR<ALS> to "0", and set I2C0AR<SA> to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set I2C0AR<ALS> to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after the start condition.

14.5 Data transfer of I2C Bus

14.5.1 Device initialization

Set POFFCR1<SBI0EN> to "1".

After confirming that the serial bus interface pin is high level, set SBI0CR2<SBIM> to "1" to select the serial bus interface mode. Set SBI0CR1<ACK> to "1", SBI0CR1<NOACK> to "0" and SBI0CR1<BC> to "000" to count the number of clocks for an acknowledge signal, to enable the slave address match detection and the GENERAL CALL detection, and set the data length to 8 bits. Set t_{HIGH} and t_{LOW} at SBI0CR1<SCK>.

Set a slave address at I2C0AR<SA> and set I2C0AR<ALS> to "0" to select the I2C bus mode. Finally, set SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0", SBI0CR2<PIN> to "1" and SBI0CR2<SWRST> to "00" for specifying the default setting to a slave receiver mode.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data cannot be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

Example : Initialize a device

```
LD      A,(P2PRD)           ;Checks whether the serial bus interface pin is at the high level
AND     A, 0x18
CMP     A, 0x18
JR      NZ, CHK_PORT
LD      (SBI0CR2),0x18      ;Selects the serial bus interface mode
LD      (SBI0CR1),0x16      ; Selects the acknowledgment mode and sets SBI0CR1<SCK> to "110"
LD      (I2C0AR),0xa0       ; Sets the slave address to 1010000 and selects the I2C bus mode
LD      (SBI0CR2),0x18      ; Selects the slave receiver mode
```

14.5.2 Start condition and slave address generation

Confirm a bus free status (SBI0SR2<BB>="0").

Set SBI0CR1<ACK> to "1" and specify a slave address and a direction bit to be transmitted to the SBI0DBR.

By writing "1" to SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN>, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBI0DBR are output. The time from generating the START condition until the falling SBI0 pin takes t_{HIGH}.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

An interrupt request occurs at the 9th falling edge of a SCL clock cycle, and SBI0CR2<PIN> is cleared to "0". The SCL0 pin is pulled down to the low level while SBI0CR2<PIN> is "0". When an interrupt request occurs, SBI0CR2<TRX> changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1: Do not write a slave address to the SBI0DBR while data is transferred. If data is written to the SBI0DBR, data to be output may be destroyed.

Note 2: The bus free state must be confirmed by software within 98.0 µs (the shortest transmitting time according to the standard mode I2C bus standard) or 23.7µs (the shortest transmitting time according to the fast mode I2C bus standard) after setting of the slave address to be output. Only when the bus free state is confirmed, set "1" to SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> to generate the start conditions. If the writing of slave address and setting of SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> doesn't finish within 98.0µs or 23.7µs, the other masters may start the transferring and the slave address data written in SBI0DBR may be broken.

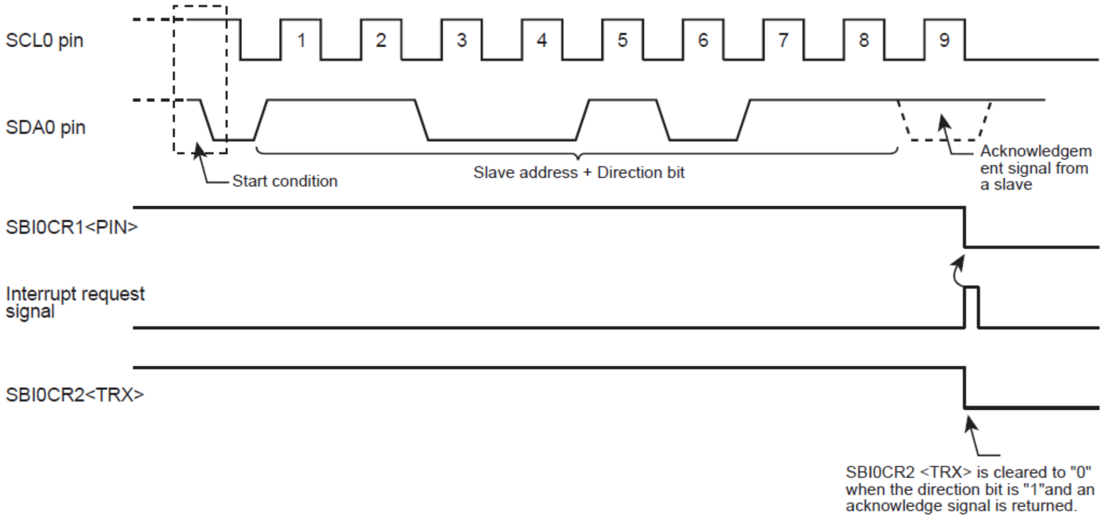

SCL0 pin
1
2
3
4
5
6
7
8
9
SDA0 pin
Start condition
Slave address + Direction bit
Acknowledgem ent signal from a slave
SBI0CR1<PIN>
Interrupt request signal
SBI0CR2<TRX>
SBI0CR2 <TRX> is cleared to "0" when the direction bit is "1"and an acknowledge signal is returned.

Figure14.17 Generating the start condition and the slave address

Example : Generate the start condition

```
TEST      (SBI0SR2).BB        ; Confirms that the bus is free
JR        F, CHK_BB
LD        (SBI0DBR), 0xcb     ; The transmission slave address 0x65 and the direction bit "1"
LD        (SBI0CR2), 0xf8     ; Write "1" to SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

14.5.3 1-word data transfer

Check SBI0SR2<MST> by the interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

14.5.3.1 SBI0SR2<MST> is"1" (Master mode)

Check SBI0SR2<TRX> and determine whether the mode is a transmitter or receiver.

(a) When SBI0SR2<TRX> is "1" (Transmitter mode)

Check SBI0SR2<LRB>. When SBI0SR2<LRB> is "1", a receiver does not request data. Implement the process to generate a stop condition (described later) and terminate data transfer.
When SBI0SR2<LRB> is "0", the receiver requests subsequent data. When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC> again, set SBI0CR1<ACK> to "1", and write the transmitted data to SBI0DBR.

After writing the data, SBI0CR2<PIN> becomes "1", a serial clock pulse is generated for transferring the subsequent 1-word data from the SCL0 pin, and then the 1-word data is transmitted from the SDA0 pin.

After the data is transmitted, an interrupt request occurs. SBI0CR2<PIN> become "0" and the SCL0 pin is set to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the SBI0SR2<LRB> checking above.

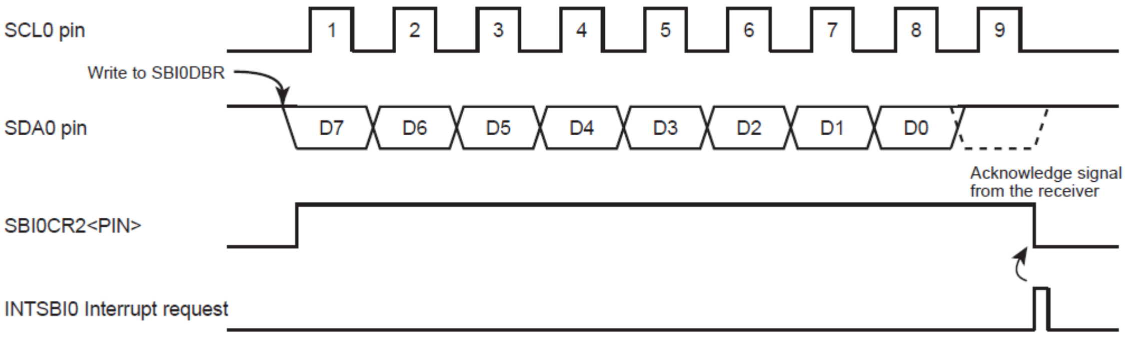


Figure14.18 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1"

(b) When SBI0SR2<TRX> is"0"(Receiver mode)

When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC> again. Set SBI0CR1< ACK> to "1" and read the received data from the SBI0DBR (Reading data is undefined immediately after a slave address is sent).

After the data is read, SBI0CR2<PIN> becomes "1" by writing the dummy data (0x00) to the SBI0DBR. The serial bus interface circuit outputs a serial clock pulse to the SCL0 pin to transfer the subsequent 1-word data and sets the SDA0 pin to "0" at the acknowledge signal timing. An interrupt request occurs and SBI0CR2<PIN> becomes "0". Then a serial bus interface circuit outputs a clock pulse for 1-word data transfer and the acknowledge signal by writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" after reading the received data.

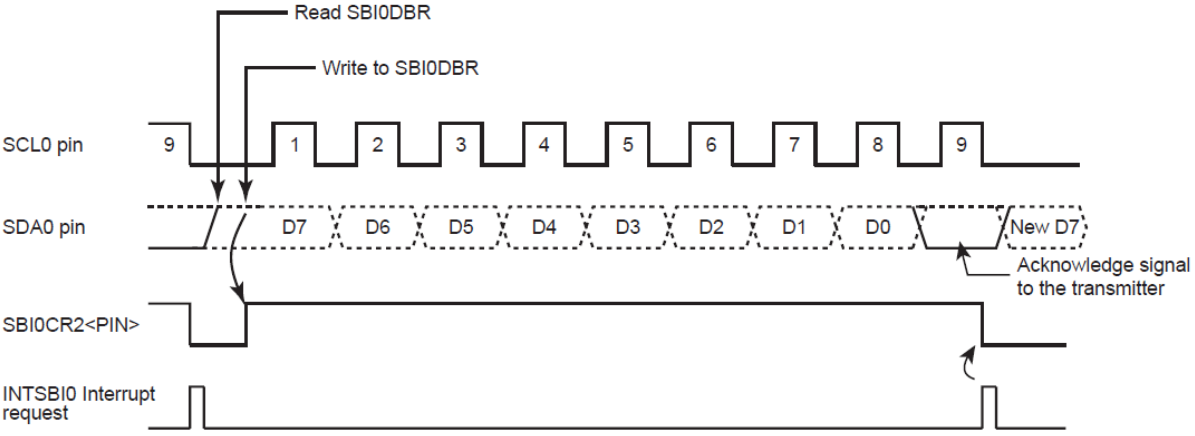


Figure14.19 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1

To make the transmitter terminate transmission, execute following procedure before receiving a last data.

1. Read the received data.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "000".
3. To set SBI0CR2<PIN> to "1", write a dummy data (0x00) to SBI0DBR.

Transfer 1-word data in which no clock is generated for an acknowledge signal by setting SBI0CR2<PIN> to "1". Next, execute following procedure.

1. Read the received data.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "001".
3. To set SBI0CR2<PIN> to "1", write a dummy data (0x00) to SBI0DBR.

Transfer 1-bit data by setting SBI0CR1<PIN> to "1".

In this case, since the master device is a receiver, the SDA line on a bus keeps the high level. The transmitter receives the high-level signal as a negative acknowledge signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generate the stop condition to terminate data transfer.

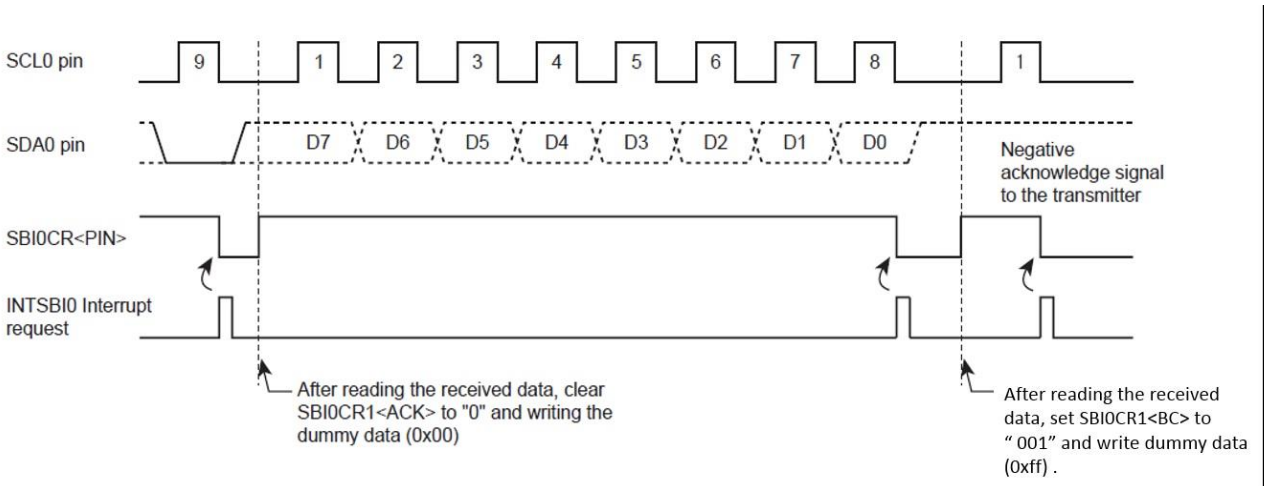

SCL0 pin
9
1
2
3
4
5
6
7
8
1
SDA0 pin
D7
D6
D5
D4
D3
D2
D1
D0
Negative acknowledge signal to the transmitter
SBI0CR<PIN>
INTSBI0 Interrupt request
After reading the received data, clear SBI0CR1<ACK> to "0" and writing the dummy data (0x00)
After reading the received data, set SBI0CR1<BC> to " 001" and write dummy data (0xff) .

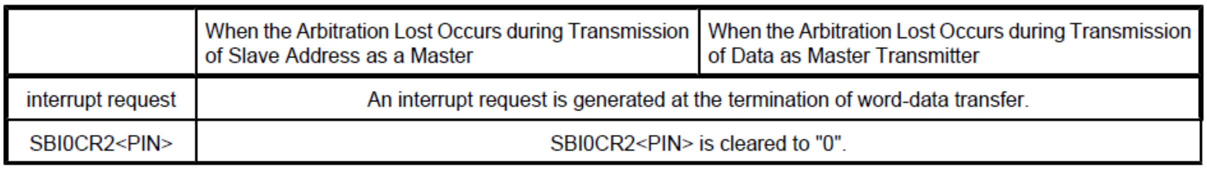
Figure14.20 Termination of Data Transfer in the Master Receiver Mode

14.5.3.2 SBI0SR2<MST> is ”0”(Slave mode)

In the slave mode, a serial bus interface circuit operates either in the normal slave mode or in the slave mode after losing arbitration.
In the slave mode, the conditions of generating the serial bus interface interrupt request (INTSBI0) are follows:

- At the end of the acknowledge signal when the received slave address matches the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge signal when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of slave address or receiving of "GENERAL CALL"

The serial bus interface circuit changes to the slave mode if arbitration is lost in the master mode. And an interrupt request occurs when the word data transfer terminates after losing arbitration. The generation of the interrupt request and the behavior of SBI0CR2<PIN> after losing arbitration are shown in Table14.4。



	When the Arbitration Lost Occurs during Transmission of Slave Address as a Master	When the Arbitration Lost Occurs during Transmission of Data as Master Transmitter
interrupt request	An interrupt request is generated at the termination of word-data transfer.	
SBI0CR2<PIN>	SBI0CR2<PIN> is cleared to "0".	

Table 14.4 The behavior of an interrupt request and SBI0CR2<PIN> after losing arbitration

When an interrupt request occurs, SBI0CR2<PIN> is reset to "0", and the SCL0 pin is set to the low level. Either writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" releases the SCL0 pin after taking t_{LOW}. Check SBI0SR2<AL>, SBI0SR2<TRX>, SBI0SR2<AAS> and SBI0SR2<AD0> and implement processes according to conditions listed in table 14.5.

SBI0SR2<TRX>	SBI0SR2<AL>	SBI0SR2<AAS>	SBI0SR2<AD0>	Conditions	Process
1	1	1	0	The serial bus interface circuit loses arbitration when transmitting a slave address, and receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to SBI0CR1<BC> and write the transmitted data to the SBI0DBR.
	0	1	0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the slave transmitter mode, the serial bus interface circuit finishes the transmission of 1-word data	Check SBI0SR2<LRB>. If it is set to "1", set SBI0CR2<PIN> to "1" since the receiver does not request subsequent data. Then, clear SBI0CR2<TRX> to "0" to release the bus. If SBI0SR2<LRB> is set to "0", set the number of bits in 1 word to SBI0CR1<BC> and write the transmitted data to SBI0DBR since the receiver requests subsequent data.
0	1	1	1/0	The serial bus interface circuit loses arbitration when transmitting a slave address, and receives a slave address of which the value of the direction bit sent from another master is "0" or receives a "GENERAL CALL".	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1", or write "1" to SBI0CR2<PIN>.
		0	0	The serial bus interface circuit loses arbitration when transmitting a slave address or data, and terminates transferring the word data.	The serial bus interface circuit is changed to the slave mode. Write the dummy data (0x00) to the SBI0DBR to clear SBI0SR2<AL> to "0" and set SBI0CR2<PIN> to "1".
	0	1	1/0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "0" or receives "GENERAL CALL".	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1", or write "1" to SBI0CR2<PIN>.
		0	1/0	In the slave receiver mode, the serial bus interface circuit terminates the receipt of 1-word data.	Set the number of bits in 1-word to SBI0CR1<BC>, read the received data from the SBI0DBR and write the dummy data (0x00).

Table 14.5 Operation in the slave mode

Note: In the slave mode, if the slave address set in I2C0AR<SA> is "0x00", a START Byte "0x01" in I2C bus standard is received, the device detects slave address match and SBI0CR2<TRX> is set to "1". Do not set I2C0AR<SA> to "0x00".

14.5.4 Stop condition generation

When SBI0CR2<BB> is "1", a sequence of generating a stop condition is started by setting "1" to SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<PIN> and clearing SBI0CR2<BB> to "0". Do not modify the contents of SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> until a stop condition is generated on a bus.

When a SCL line on a bus is pulled down by other devices, a serial bus interface circuit generates a stop condition after a SCL line is released. The time from the releasing SCL line until the generating the STOP condition takes t_{HIGH}.

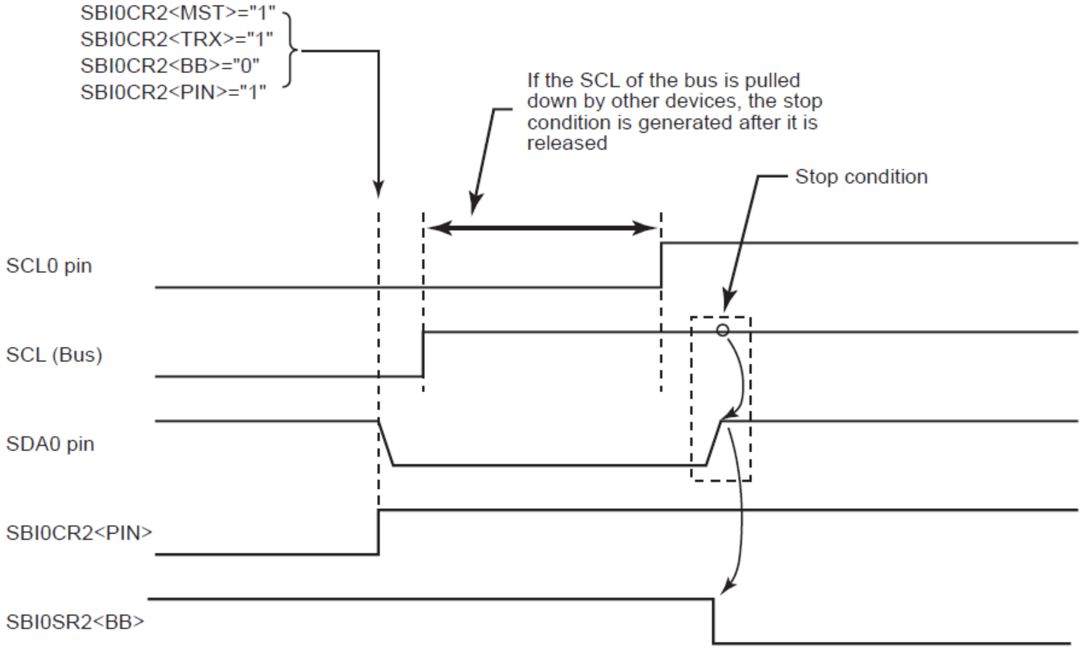


Figure 14.21 Stop Condition Generation

Example : Generate the stop condition

```
LD        (SBI0CR2), 0xD8       ; Sets SBI0CR2<MST>, <TRX> and <PIN> to "1" and SBI0CR2<BB> to "0"
TEST      (SBI0SR2).BB          ; Waits until the bus is set free
JR        T, CHK_BB
```

14.5.5 Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart the serial bus interface circuit.

Clear SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0" and set SBI0CR2 <PIN> to "1". The

SDA0 pin retains the high level and the SCL0 pin is released.

Since this is not a stop condition, the bus is assumed to be in a busy state from other devices. Check SBI0SR2<BB> until it becomes "0" to check that the SCL0 pin of the serial bus interface circuit is released. Check SBI0SR2<LRB> until it becomes "1" to check that the SCL line on the bus is not pulled down to the low level by other devices.

After confirming that the bus stays in a free state, generate a start condition in the procedure "Start condition and slave address generation".
In order to meet the setup time at a restart, take at least 4.7μs of waiting time by the software in the standard mode I2C bus standard or at least 0.6μs of waiting time in the fast mode I2C bus standard from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note: When the master is in the receiver mode, it is necessary to stop the data transmission from the slave device before the STOP condition is generated. To stop the transmission, the master device make the slave device receiving a negative acknowledge. Therefore, SBI0SR2<LRB> is "1" before generating the Restart and it can not be confirmed that SCL line is not pulled down by other devices. Please confirm the SCL line state by reading the port.

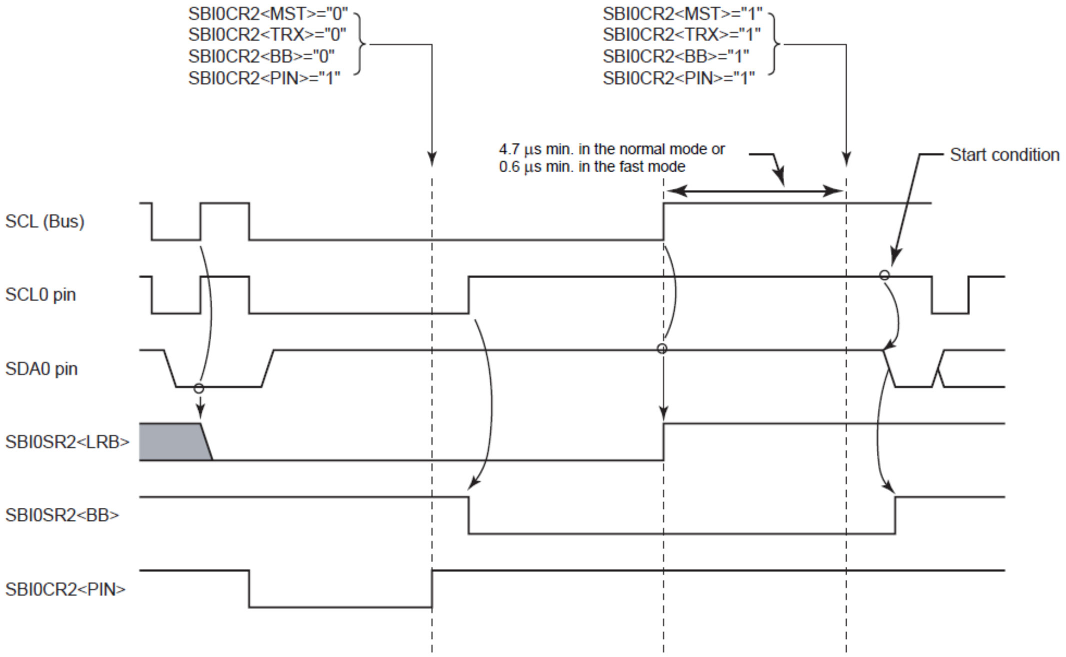


Figure 14.22 Timing diagram when restarting

Example : Generate a restart

```
LD        (SBI0CR2), 0x18       ; Sets SBI0CR2<MST>, <TRX> and <BB> to "0" and SBI0CR2<PIN> to "1"
TEST      (SBI0SR2).BB          ; Waits until SBI0SR2<BB> becomes "0"
JR        T, CHK_BB
TEST      (SBI0SR2).LRB         ; Waits until SBI0SR2<LRB> becomes "1"
JR        F, CHK_LRB
          -
          -                     ; Wait time process by the software
          -
LD        (SBI0CR2), 0xf8       ; Sets SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

14.6 AC Specifications

The operating mode (fast or standard) mode should be selected suitable for frequency of fcgck. For these operating mode, refer to the following table.

Parameter	Symbol	Standard mode		Fast mode		Unit
		MIN.	MAX.	MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	fcgck / (m+n)	0	fcgck / (m+n)	kHz
Hold time (re)start condition. This period is followed by generation of the first clock pulse.	$t_{HD;STA}$	m / fcgck	-	m / fcgck	-	μs
Low-level period of SCL clock (output)	t_{LOW}	n / fcgck	-	n / fcgck	-	μs
High-level period of SCL clock (output)	t_{HIGH}	m / fcgck	-	m / fcgck	-	μs
Low-level period of SCL clock (input)	t_{LOW}	5 / fcgck	-	5 / fcgck	-	μs
High-level period of SCL clock (input)	t_{HIGH}	3 / fcgck	-	3 / fcgck	-	μs
Restart condition setup time	$t_{SU;STA}$	Depends on the software	-	Depends on the software	-	μs
Data hold time	$t_{HD;DAT}$	0	5 / fcgck	0	5 / fcgck	μs
Data setup time	$t_{SU;DAT}$	250	-	100	-	ns
Rising time of SDA and SCL signals	t_r	-	1000	-	300	ns
Falling time of SDA and SCL signals	t_f	-	300	-	300	ns
Stop condition setup time	$t_{SU;STO}$	m / fcgck	-	m / fcgck	-	μs
Bus free time between the stop condition and the start condition	t_{BUF}	Depends on the software	-	Depends on the software	-	μs
Time before rising of SCL after SBICR2<PIN> is changed from "0" to "1"	$t_{SU;SCL}$	n / fcgck	-	n / fcgck	-	μs

Table 14.6 AC Specifications (Circuit Output Timing)

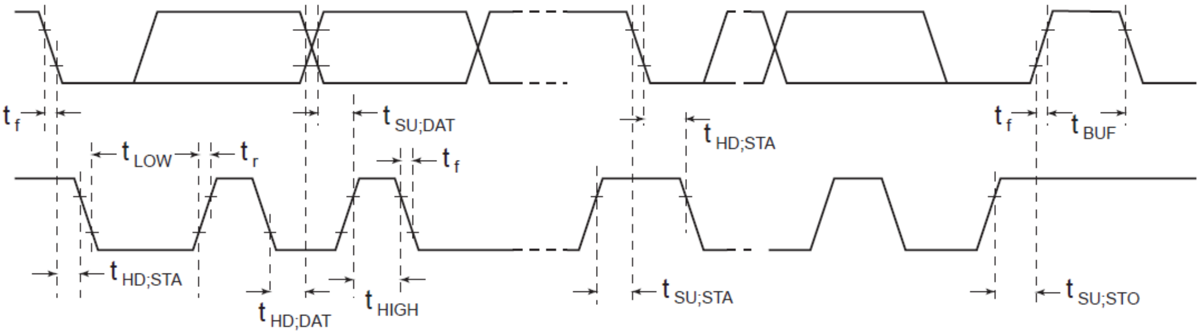


Figure14.23 Definition of Timing (No.1)

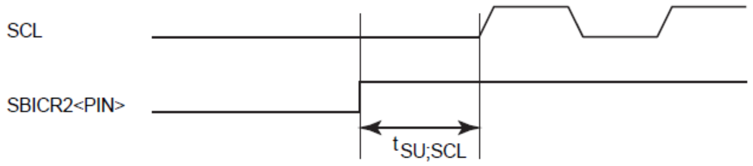


Figure 14.24 Definition of Timing (No.2)

15 Synchronous Serial Interface (SIO)

MQ6935 contains 1 channel of 8-bit serial interface of the clock synchronization type.

	SIOxCR (Address)	SIOxSR (Address)	SIOxBUF (Address)
SIO0	SIO0CR (0x001F)	SIO0SR (0x0020)	SIO0BUF (0x0021)

Table 15.1 SFR Address Assignment

	Serial clock Input/output pin	Serial data Input pin	Serial data output pin
SIO0	SCLK0 pin	SI0 pin	SO0 pin

Table 15.2 Pin Names

15.1 Configuration

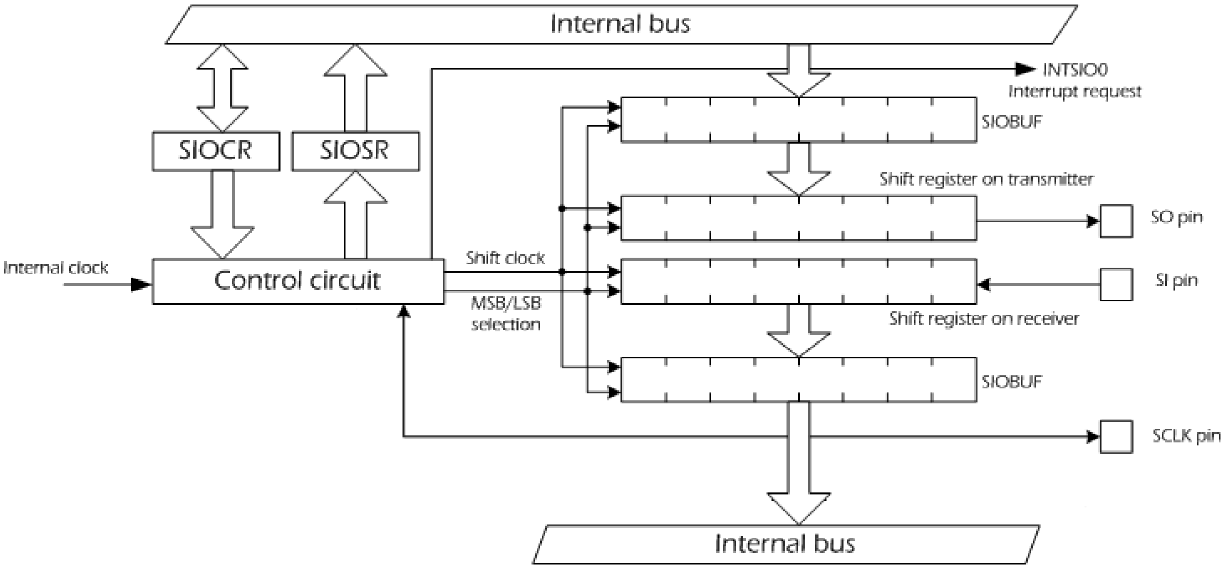


Figure15.1 Serial Interface

Note: The serial interface input/output pins are also used as the I/O ports. The I/O port register settings are required to use these pins for a serial interface.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

15.2 Control

The synchronous serial interface SIO0 is controlled by the low power consumption registers (POFFCR2)、the serial interface data buffer register (SIO0BUF), the serial interface control register (SIO0CR) and the serial interface status register(SIO0SR).

Low power consumption register 2

POFFCR2 (0x0F76)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	RTCEN	-	-	-	-	SIO0EN
Read/Write	R/W	R/W	R/W	R	R	R	R	R/W
After reset	0	0	0	0	0	0	0	0

RTCEN	RTC control	0: Disable 1: Enable
SIO0EN	SIO0 control	0: Disable 1: Enable

Serial interface buffer register

SIO0BUF (0x0021)	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF							
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Serial interface buffer register

SIO0BUF (0x0021)	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF							
Read/Write	W	W	W	W	W	W	W	W
After reset	1	1	1	1	1	1	1	1

Note: SIO0BUF is the data buffer for both transmission and reception. The last received data is read each time SIO0BUF us read. If SIO0BUF has never received data, it is read as "0". When data is written into it, the data is treated as the transmit data.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Serial interface control register

SIO0CR (0x001F)	7	6	5	4	3	2	1	0
Bit Symbol	SIOEDG	SIOCKS			SIODIR	SIOS	SIOM	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

<table>
<tr><td>SIOEDG</td><td>Transfer edge selection</td><td colspan="3">0:Receive data at a rising edge and transmit data at a falling edge
1:Transmit data at a rising edge and receive data at a falling edge.</td></tr>
<tr><td rowspan="9">SIOCKS</td><td rowspan="9">Serial clock selection[Hz]</td><td></td><td>NORMAL1/2 or IDLE 1/2 mode</td><td>SLOW1/2 or SLEEP 1 mode</td></tr>
<tr><td>000</td><td>fcgck/2^9</td><td>-</td></tr>
<tr><td>001</td><td>fcgck/2^6</td><td>-</td></tr>
<tr><td>010</td><td>fcgck/2^5</td><td>-</td></tr>
<tr><td>011</td><td>fcgck/2^4</td><td>-</td></tr>
<tr><td>100</td><td>fcgck/2^3</td><td>-</td></tr>
<tr><td>101</td><td>fcgck/2^2</td><td>-</td></tr>
<tr><td>110</td><td>fcgck/2</td><td>fs/2^3</td></tr>
<tr><td>111</td><td colspan="2">Ext. clock input</td></tr>
<tr><td>SIODIR</td><td>Transfer format (MSB/LSB)</td><td colspan="3">0: LSB first (transfer from bit 0)
1: MSB first (transfer from bit 7)</td></tr>
<tr><td>SIOS</td><td>Transfer operation start/stop instruction</td><td colspan="3">0:Operation stop (reserved stop)
1:Operation start</td></tr>
<tr><td>SIOM</td><td>Transfer mode selection and operation</td><td colspan="3">00:Operation stop (forced stop)
01: 8-bit transmit mode
10:8-bit receive mode
11:8-bit transmit and receive mode</td></tr>
</table>

Note1: fcgck is Gear clock (Hz)，fs is low-frequency clock (Hz)。

Note2: After the operation is started (writing "1" to SIOS), writing to SIOEDG, SIOCKS and SUIDUR is invalid until SIO0SR<SIOF> becomes"0". (SIOEDGE, SIOCKS and SIODIR can be changed at the same time as changing SIOS from "0" to"1")

Note3: After the operation is started (writing "1" to SIOS), no values other than "00" can be written to SIOM until SIOF becomes "0" (if a value from "01" to "11" is written to SIOM, it is ignored). The transfer mode cannot be changed during the operation.

Note4: SIOS remains at "0", if "1" is written to SIOS when SIOM is "00" (operation stop).

Note5: When SIO is used in SLOW1/2 or SLEEP1 mode, be sure to set SIOCKS to "110". If SIOCKs is set to any other value SIO will not operate. When SIO is used in SLOW1/2 or SLEEP1 mode, execute communications with SIOCKS="110" in advance or change SIOCKs after SIO is stopped.

Note6: When STOP, IDLE0 or SLEEP0 mode is activated, SIOM is automatically cleared to "00" and SIO stops the operation. Meanwhile, SIOS is cleared to "0". However, the values set for SIOEDG, SIOCKS and SIODIR are maintained.

Serial interface status register

SIO0SR (0x0020)	7	6	5	4	3	2	1	0
Bit Symbol	SIOF	SEF	OERR	RENDB	UERR	TBFL	-	-
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

SIOF	Serial transfer operation status monitor	0:Transfer not in progress 1:Transfer in progress
SEF	Shift operation status monitor	0:Shift operation not in progress 1:Shift operation in progress
OERR	Receive overrun error flag	0:No overrun error has occurred 1:At least one overrun error has occurred
RENDB	Receive completion flag	0: No data has been received since the last received data was read out 1: At least one data receive operation has been executed
UERR	Transmit underrun error flag	0: No transmit underrun error has occurred 1:At least one transmit underrun error has occurred
TBFL	Transmit buffer full flag	0:The transmit buffer is empty 1:The transmit buffer has the data that has not yet been transmitted

Note1: The OERR and UERR flags are cleared by reading SIO0SR.

Note2: The REND flag is cleared by reading SIO0BUF.

Note3: Writing "00" to SIO0CR<SIOM> clears all the bits of SIO0SR to "0", whether the serial interface is operating or not. When STOP, IDLE0 or SLEEP0 mode is activated, SIOM is automatically cleared ti "00" and all the bits of SIO0SR are cleared to "0"

Note 4: Bit 1 to 0 of SIO0SR are read"0".

15.3 Low power consumption function

Serial interface 0 has the low power consumption registers (POFFCR2) that save power when the serial interface is not being used.

Setting POFFCR2<SIO0EN> to "0" disables the basic clock supply to serial interface 0 to save power. Note that this renders the serial interface unusable. Setting POFFCR2<SIO0EN> to "1" enables the basic clock supply to serial interface 0 and allows the serial interface to operate.

After reset, POFFCR2<SIO0EN> are initialized to "0", and this renders the serial interface unusable. When using the serial interface for the first time, be sure to set POFFCR2<<SIO0EN> to "1" in the initial setting of the program (before the serial interface control registers are operated).

During the serial interface operation, do not change POFFCR2<SIO0EN> to "0". Otherwise serial interface 0 may operate unexpectedly.

15.4 Functions

15.4.1 Transfer format

The transfer format can be set to either MSB or LSB first by SIO0CR<SIODIR>. Setting SIOCR<SIODIR> to "0" selects LSB first as the transfer format. In this case the serial data is transferred in sequence from the least significant bit.

Setting SIO0CR<SIODIR> to "1" selects MSB first as the transfer format. In this case, the serial data is transferred in sequence from the most significant bit.

15.4.2 Serial clock

The serial clock can be selected by using SIO0CR <SIOCK>. Setting SIO0CR<SIOCKS> to "000"to110" selects the internal clock as the serial clock. In this case, the serial clock is output from the SCLK0 pin. The serial data is transferred in synchronization with the edge of the SCLK0 pin output.

Setting SIO0CR<SIOCKS> to"111" selects an external clock as the serial clock. In this case, an external serial clock must be input to the SCLK0 pin. The serial data is transferred in synchronization with the edge of the external clock.

The serial data transfer edge can be selected for both the external and internal clocks. For details, refer to "15.4.3 Transfer edge selection".

SIO0CR <SIOCKS>	Serial clock [Hz]		fcgck=4MHz		fcgck=8MHz		fcgck=10MHz		fs=32.768kHz	
	NORMAL 1/2 or IDLE 1/2 mode	SLOW1/2 or SLEEP1 mode	1-bit time(us)	Baud rate (bps)	1-bit time(us)	Baud rate (bps)	1-bit time(us)	Baud rate (bps)	1-bit time(us)	Baud rate (bps)
000	fcgck/2^9	-	128	7.813k	64	15.625k	51.2	19.531k	-	-
001	fcgck/2^6	-	16	62.5k	8	125k	6.4	156.25k	-	-
010	fcgck/2^5	-	8	125k	4	250k	3.2	312.5k	-	-
011	fcgck/2^4	-	4	250k	2	500k	1.6	625k	-	-
100	fcgck/2^3	-	2	500k	1	1M	0.8	1.25M	-	-
101	fcgck/2^2	-	1	1M	0.5	2M	0.4	2.5M	-	-
110	fcgck/2	fs/2^3	0.5	2M	0.25	4M	0.2	5M	244	4k

Table 15.3 Transfer Baud Rate

15.4.3 Transfer edge selection

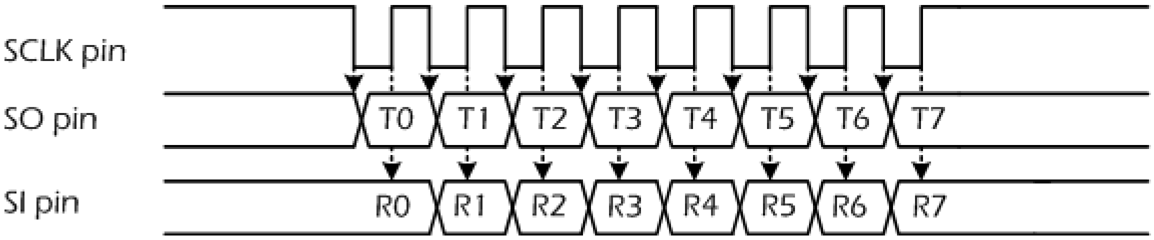
The serial data transfer edge can be selected by using SIOCR<SIOEDG>.

SIO0CR<SIOEDG>	Data transmission	Data reception
0	Falling edge	Rising edge
1	Rising edge	Falling edge

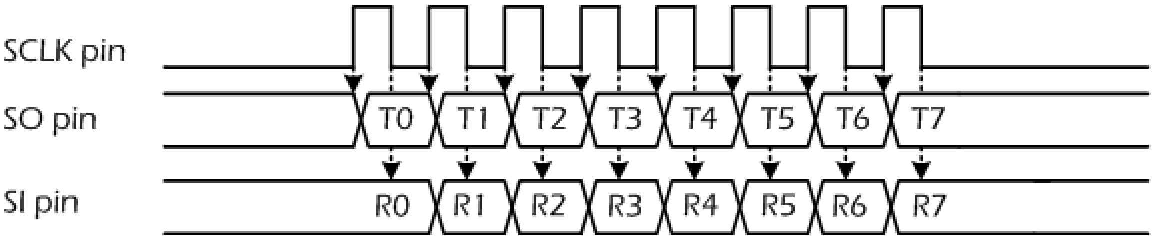
Table 15.4 Transfer Edge Selection

When SIOCR<SIOEDG> is 0，the data is transmitted in synchronization with the falling edge of the clock and the data is received in synchronization with the rising edge of the clock.

When SIOCR<SIOEDG> is”1”，the data is transmitted in synchronization with the rising edge of the clock and the data is received in synchronization with the falling edge of the clock.



SIOCR<SIOEDG> = "0"



SIOCR<SIOEDG> = "1"

Figure 15.2 Transfer Edge

Note: When an external clock input is used, 4/fcgck or longer is needed between the receive edge at the 8th bit and the transfer edge at the first bit of the next transfer.

15.5 Transfer Modes

15.5.1 8-bit transmit mode

Setting SIO0CR<SIOM> to"01",to select 8-bit transmit mode.

15.5.1.1 Setting

Before starting the transmit operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR <SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to"111".

Setting SIO0CR<SIOM>to" 01"，to select the 8-bit transmit mode.

The transmit operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR <SIOF> is1. Make these setting while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

15.5.1.2 Starting the transmit operation

The transmit operation is started by writing data to SIOBUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and then transmitted as the serial data from the SO0 pin according to the settings of SIO0CR<SIOEDG, SIOCKs an SIODIR>. The serial data becomes undefined if the transmit operation is started without writing any transmit data to SIO0BUF.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin. By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1" and an INTSIO0interrupt request is generated. SIO0SR <SEF> is cleared to "0" when the 8th bit of the serial data is output.

15.5.1.3 Transmit buffer and shift operation

If data is written to SIO0BUF when the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0". If data is written to SIO0BUF when some data remains in the shift register, SIO0SR<TBFL> is set to "1". If new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

15.5.1.4 Operation on completion of transmission

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

The operation on completion of the data transmission varies depending on the operating clock and the state of SIO0SR<TBFL>.

(a) When the internal clock is used and SIO0SR<TBFL> is"0"

When the data transmission is completed, the SCLK0 pin becomes the initial state and the SO0 pin becomes the "H" level. SIO0SR<SEF> remains at "0". When the internal clock is used, the serial clock and data output is stopped until the next transmit data is written into SIO0BUF (automatic wait).

When the subsequent data is written into SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the serial clock, and the transmit operation is restarted. An INTSIO0 interrupt request is generated at the restart of the transmit operation.

(b)When an external clock is used and SIO0SR<TBFL> is "0"

When the data transmission is completed, the SO pin keeps last output value. When an external serial clock is input to the SCLK0 pin after completion of the data transmission, an undefined value is transmitted and the transmit underrun error flag SIO0SR<UERR> is set to "1".

If a transmit underrun error occurs, data must not be written to SIO0BUF during the transmission of an undefined value. It is recommended to finish the transmit operation by setting SIO0CR<SIOS> to "0" or force the transmit operation to stop by setting SIO0CR<SIOM> to "00".

The transmit underrun error flag SIO0SR<UERR> is cleared by reading SIO0SR.

(c) When an internal or external clock is used an SIO0SR<TBFL> is "1"

When the data transmission is completed, SIO0SR<TBFL> is cleared to "0". The data in SIO0BUF Is transferred to the shift register and the transmission of subsequent data is started. At this time, SIO0SR<SEF> is set to "1" and an INTSIO0 interrupt request is generated.

15.5.1.5 Stopping the transmit operation

Set SIO0CR<SIOS> to "0" to stop the transmit operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the transmit operation is stopped immediately and an INTSIO0 interrupt request is generated. When SIO0SR<SEF> is "1", the transmit operation is stopped after all the data in the shift register is transmitted (reserved stop). At this time, an INTSIO0 interrupt request is generated again.

When the transmit operation is completed, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

If the internal clock has been used, the SO0 pin automatically returns to the "H" level. If an external

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the "J" level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR <SEF> value. The SO0pin becomes the "H" level. If the internal clock is selected, the SCLK0 pin returns to the initial level.

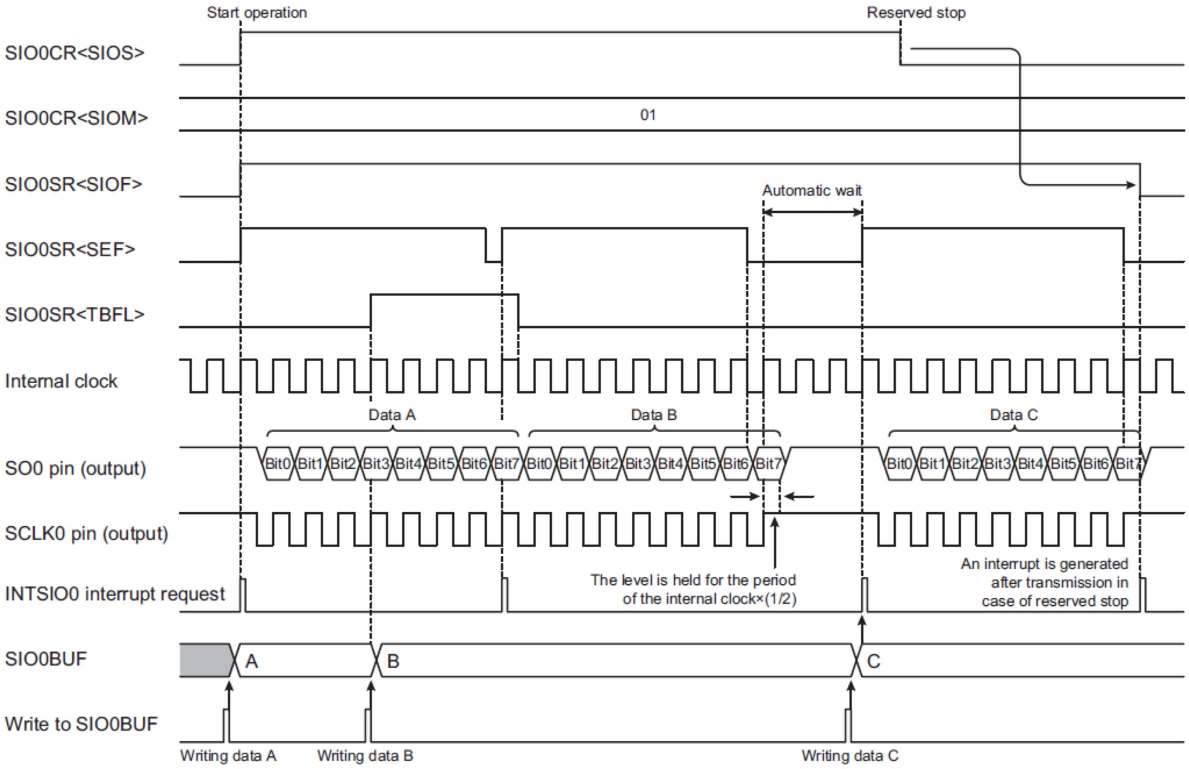


Figure 15.3 8-bit Transmit Mode (Internal Clock and Reserved Stop)

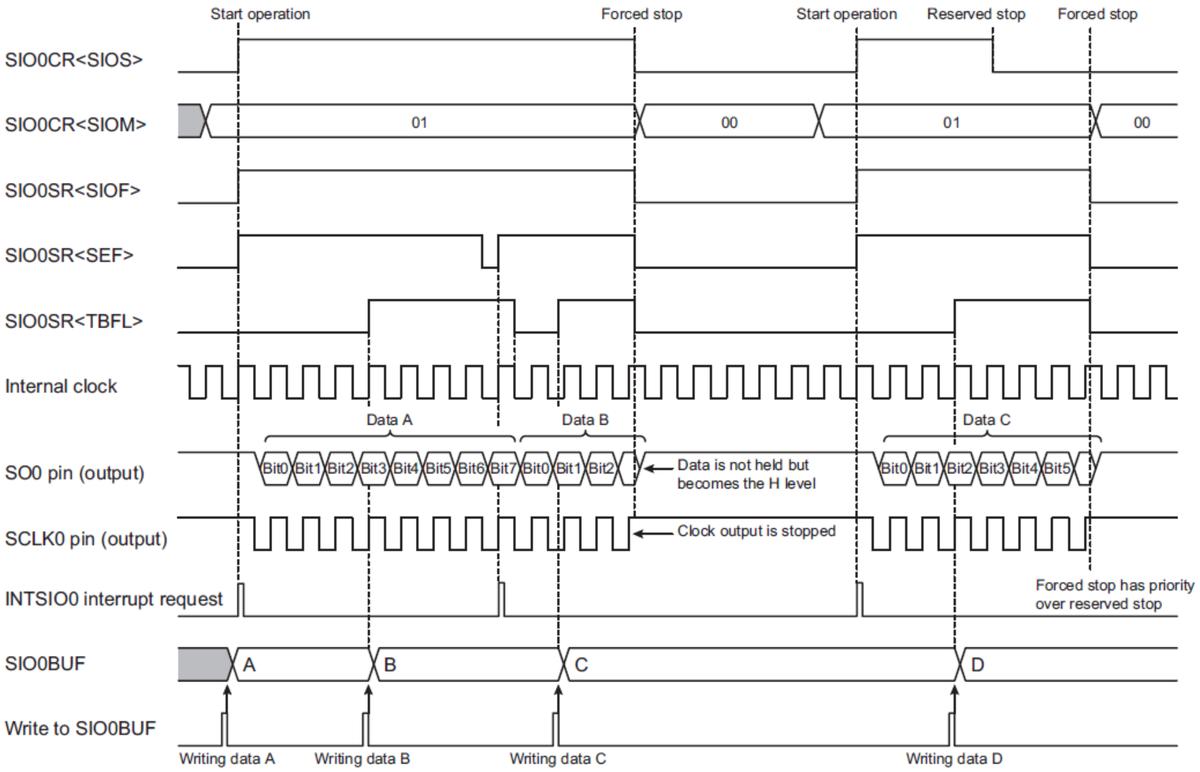


Figure 15.4 8-bit Transmit mode(Internal clock and forced stop)

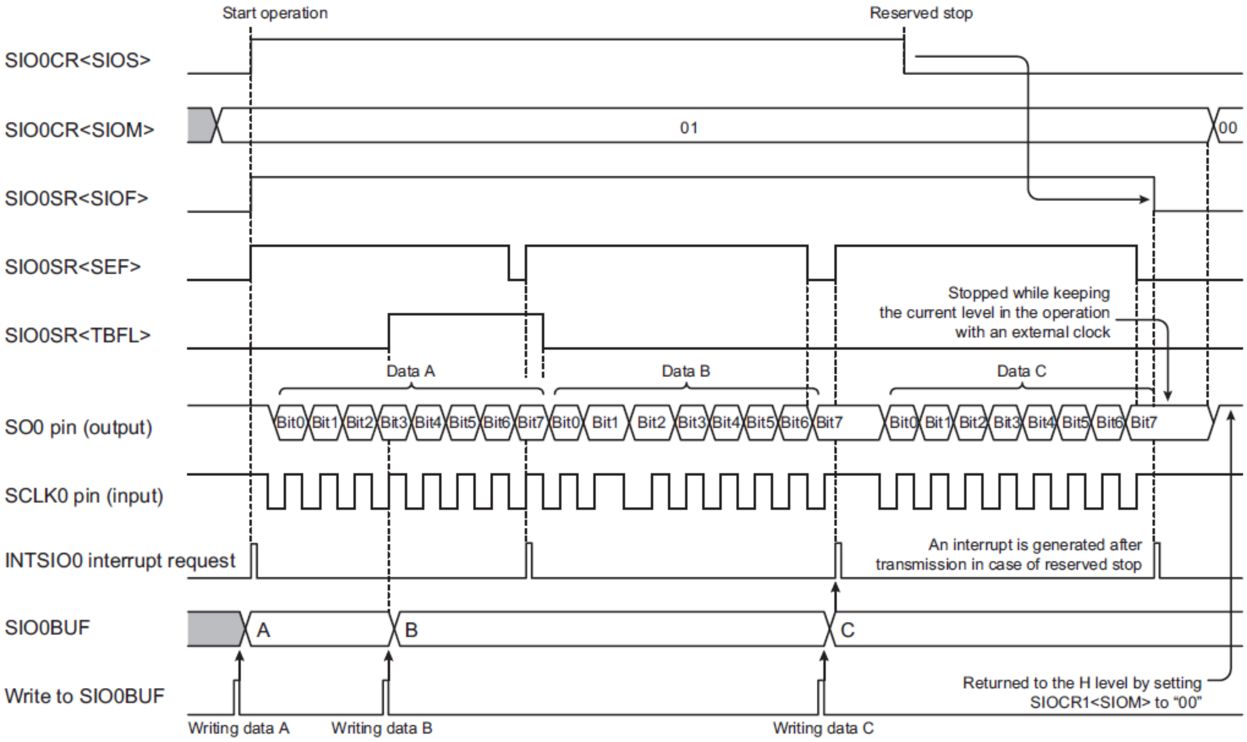

Start operation
Reserved stop
SIO0CR<SIOS>
SIO0CR<SIOM>
01
00
SIO0SR<SIOF>
SIO0SR<SEF>
SIO0SR<TBFL>
Stopped while keeping the current level in the operation with an external clock
Data A
Data B
Data C
SO0 pin (output)
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
SCLK0 pin (input)
INTSIO0 interrupt request
An interrupt is generated after transmission in case of reserved stop
SIO0BUF
A
B
C
Write to SIO0BUF
Returned to the H level by setting SIOCR1<SIOM> to "00"
Writing data A
Writing data B
Writing data C

Figure 15.5 8-bit Transmit mode (External clock and reserved stop)

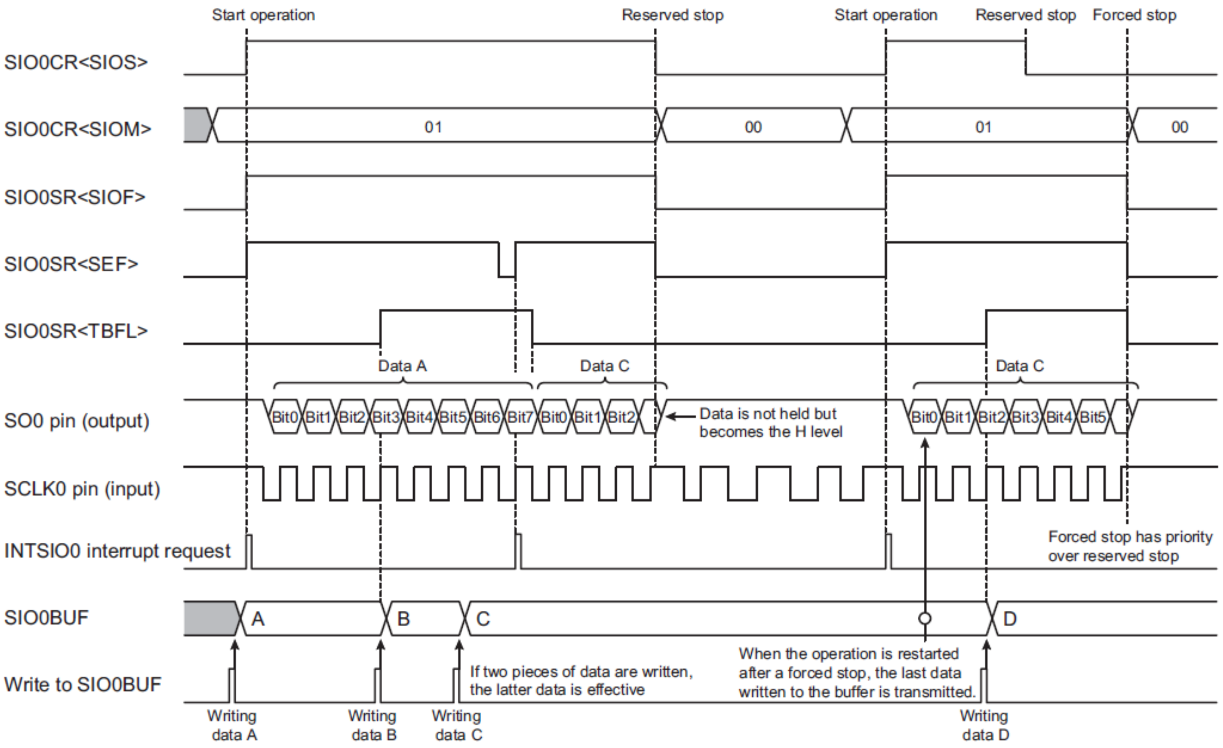

Start operation
Reserved stop
Start operation
Reserved stop
Forced stop
SIO0CR<SIOS>
SIO0CR<SIOM>
01
00
01
00
SIO0SR<SIOF>
SIO0SR<SEF>
SIO0SR<TBFL>
Data A
Data C
Data C
SO0 pin (output)
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Bit0 Bit1 Bit2
Data is not held but becomes the H level
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5
SCLK0 pin (input)
INTSIO0 interrupt request
Forced stop has priority over reserved stop
SIO0BUF
A
B
C
D
Write to SIO0BUF
If two pieces of data are written, the latter data is effective
When the operation is restarted after a forced stop, the last data written to the buffer is transmitted.
Writing data A
Writing data B
Writing data C
Writing data D

Figure 15.6 8-bit Transmit mode (External clock and forced stop)

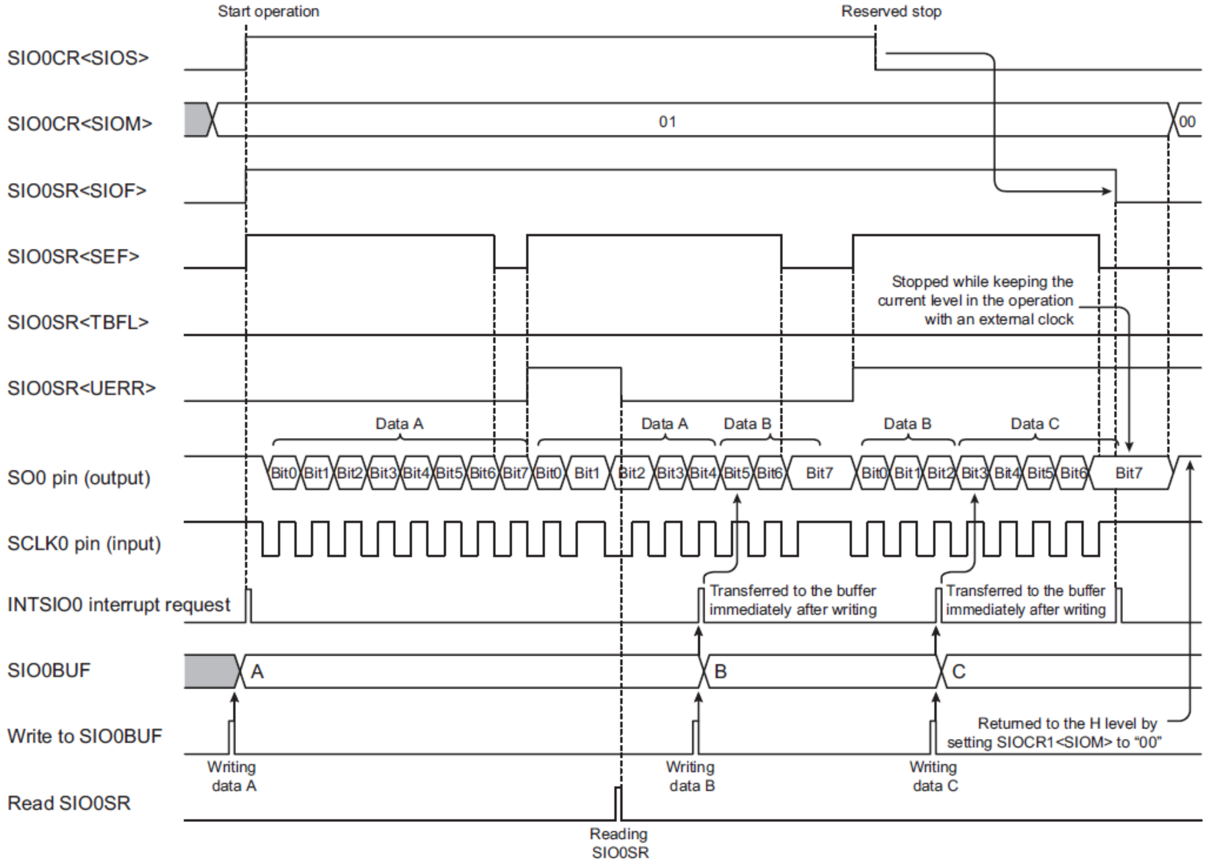

Start operation
Reserved stop
SIO0CR<SIOS>
SIO0CR<SIOM>
01
00
SIO0SR<SIOF>
SIO0SR<SEF>
SIO0SR<TBFL>
Stopped while keeping the current level in the operation with an external clock
SIO0SR<UERR>
Data A
Data A
Data B
Data B
Data C
SO0 pin (output)
Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7 Bit0 Bit1 Bit2 Bit3 Bit4 Bit5 Bit6 Bit7
SCLK0 pin (input)
INTSIO0 interrupt request
Transferred to the buffer immediately after writing
Transferred to the buffer immediately after writing
SIO0BUF
A
B
C
Write to SIO0BUF
Returned to the H level by setting SIOCR1<SIOM> to "00"
Writing data A
Writing data B
Writing data C
Read SIO0SR
Reading SIO0SR

Figure 15.7 8-bit Transmit mode (External clock and occurrence of transmit underrun error)

15.5.2 8-bit Receive Mode

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to”10”.

15.5.2.1 Setting

Before starting the receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SUICKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SUICKS>. To use an external clock as the serial clock, set SIO0CR<SOCKS> to “111”.

The 8-bit Receive mode is selected by setting SIO0CR<SIOM> to “10” . Reception is started by setting SIO0CR<SIOS> to “1”.

Writing data to SIO0CR<SIOEDG,SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is “1”. Make these settings while the serial communication is stopped . While the serial communication is in progress<SIO0SR<SIOF>=”1”) , only writing “00” to SIO0CR<SIOM> or writing “0” to SIO0CR<SIOS> is valid.

15.5.2.2 Starting the receive operation

Reception is started by setting SIO0CR<SIOS> to “1”. External serial data is taken into the shift register from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKs and SIODIR>.

Internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.
I
By setting SIO0CR<SIOS> to “1”, SIO0SR<SIOF and SEF> are automatically set to “1”.

15.5.2.3 Operation on completion of reception

When the data reception is complete, the data is transferred from the shift register to SIOBUF and an INTSIO0 interrupt request is generated. The receive completion flag SIO0SR<REND> is set to “1”.

In the operation with the internal clock, the serial clock output is stopped until the receive data is read from SIO0BUF(automatic wait). At this time, SIO0SR<SEF> is set to “0”. By reading the receive data from SIO0BUF, SIO0SR<SEF> is set to “1”, the serial clock output is restarted and the receive operation continues.

In the operation with an external clock, data can be continuously received without reading the received data from SIO0BUF. In this case, data must be red from SIO0BUF before the subsequent data has been fully received. If the subsequent data is received completely before reading data from SIO0BUF, the overrun error flag SIO0SR<OERR> is set to “1”. When an overrun error has occurrence of an overrun error is discarded, and SIO0BUF holds the data value received before the occurrence of the overrun error.

SIO0SR<REND> is cleared to “0” by reading data from SIO0BUF. SIO0SR<OERR> is cleared by reading SIO0SR.

15.5.2.4 Stopping the receive operation

Set SIO0CR<SIOS> to "0" to stop the receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data has been completely received (reserved stop). At this time, an INTSIO0 interrupt request is generated. After the operation has stopped completely, SIO0SR<SIOF and SEF> are cleared to "0". Other
SIO0SR registers keep their values.

The receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. If the internal clock is selected, the SCLK0 pin returns to the initial level.

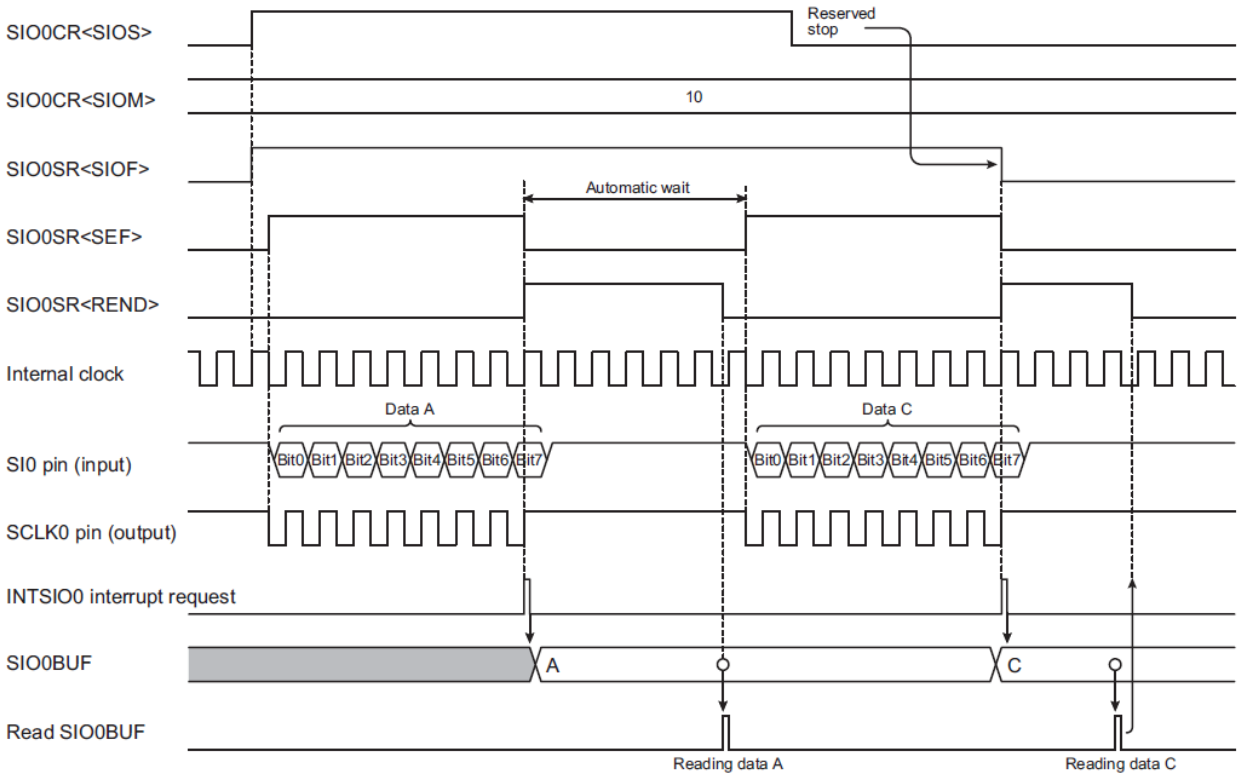


Figure 15.8 8-bit Receive Mode (Internal Clock and Reserved Stop)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

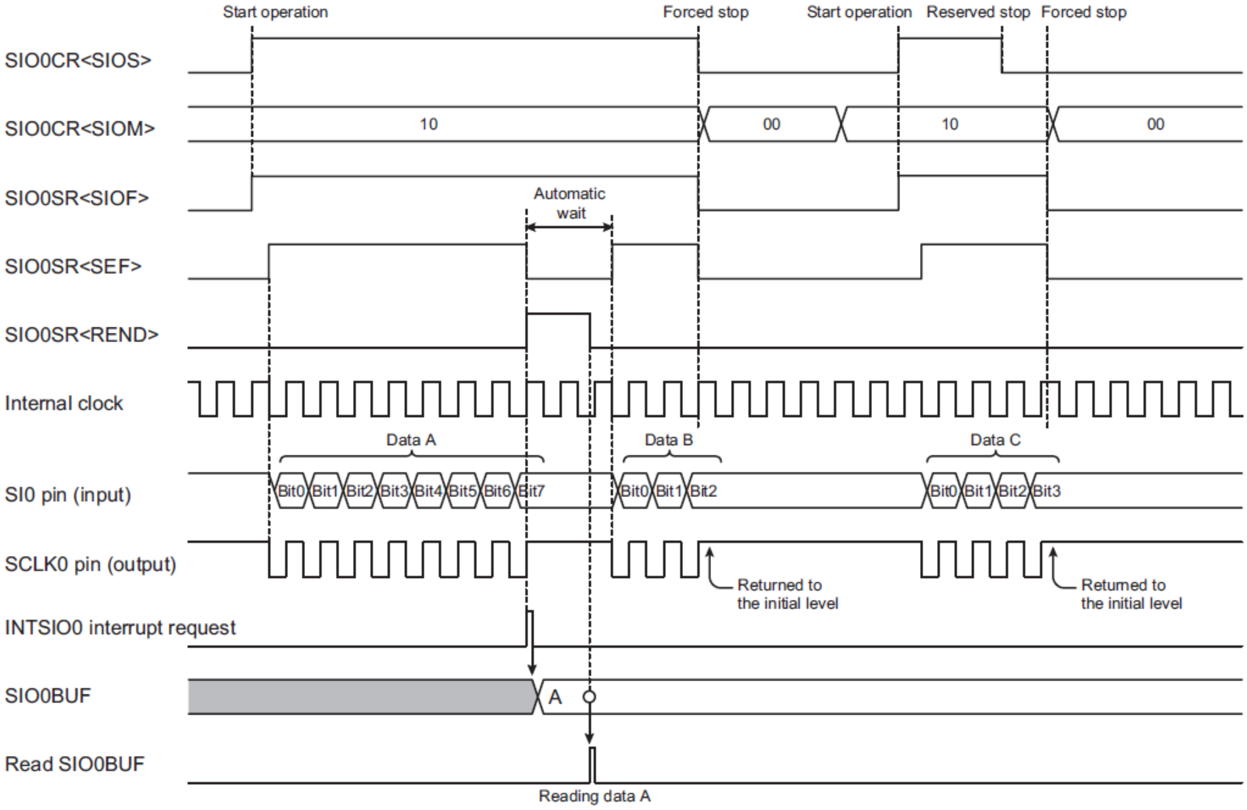


Figure 15.9 8-bit Receive Mode (Internal Clock and Forced Stop)

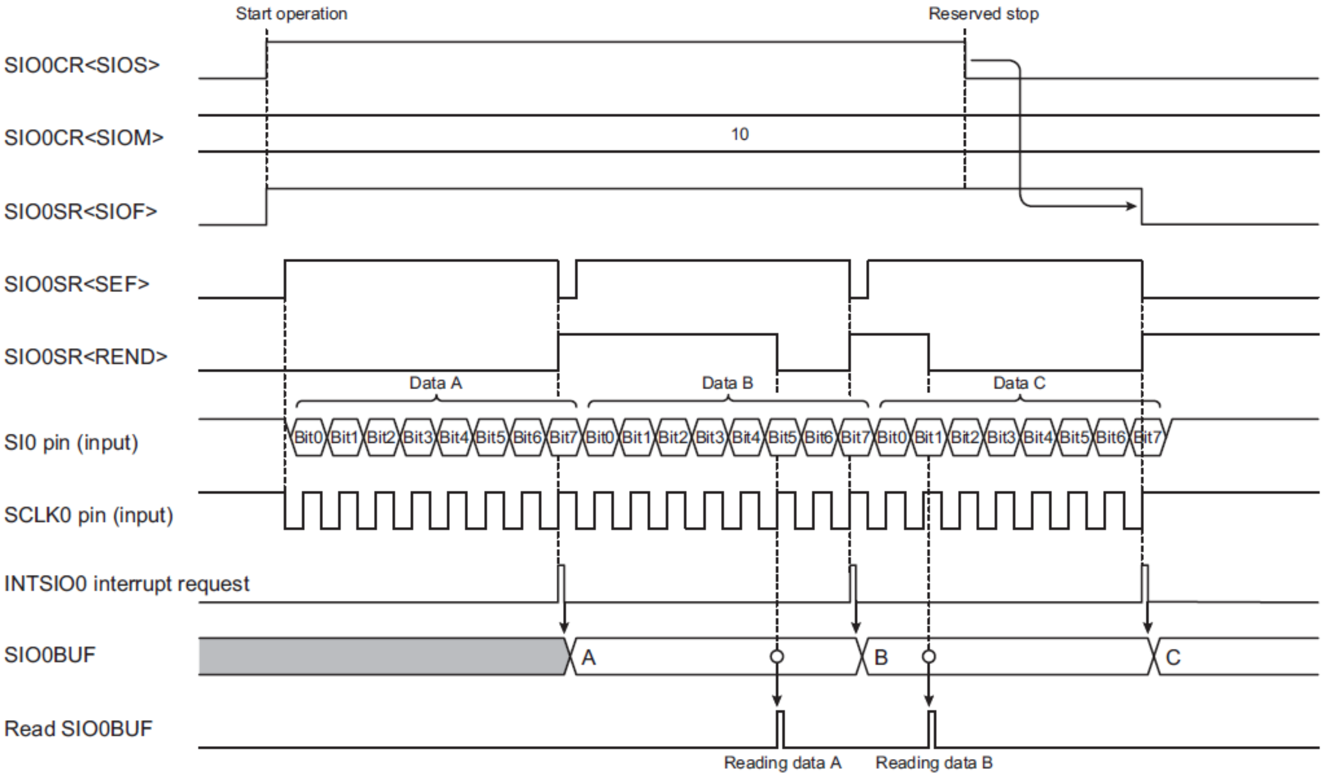


Figure 15.10 8-bit Receive Mode (External Clock and Reserved Stop)

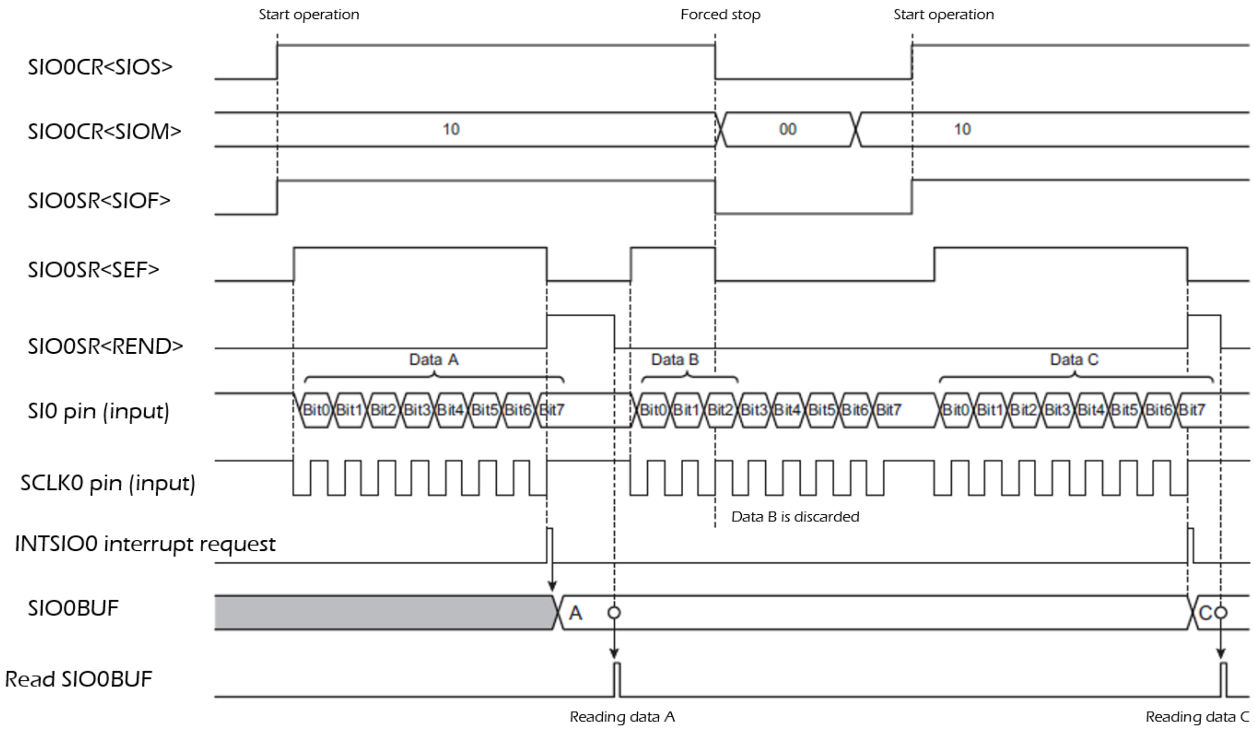


Figure 15.11 8-bit Receive Mode(External Clock and Forced Stop)

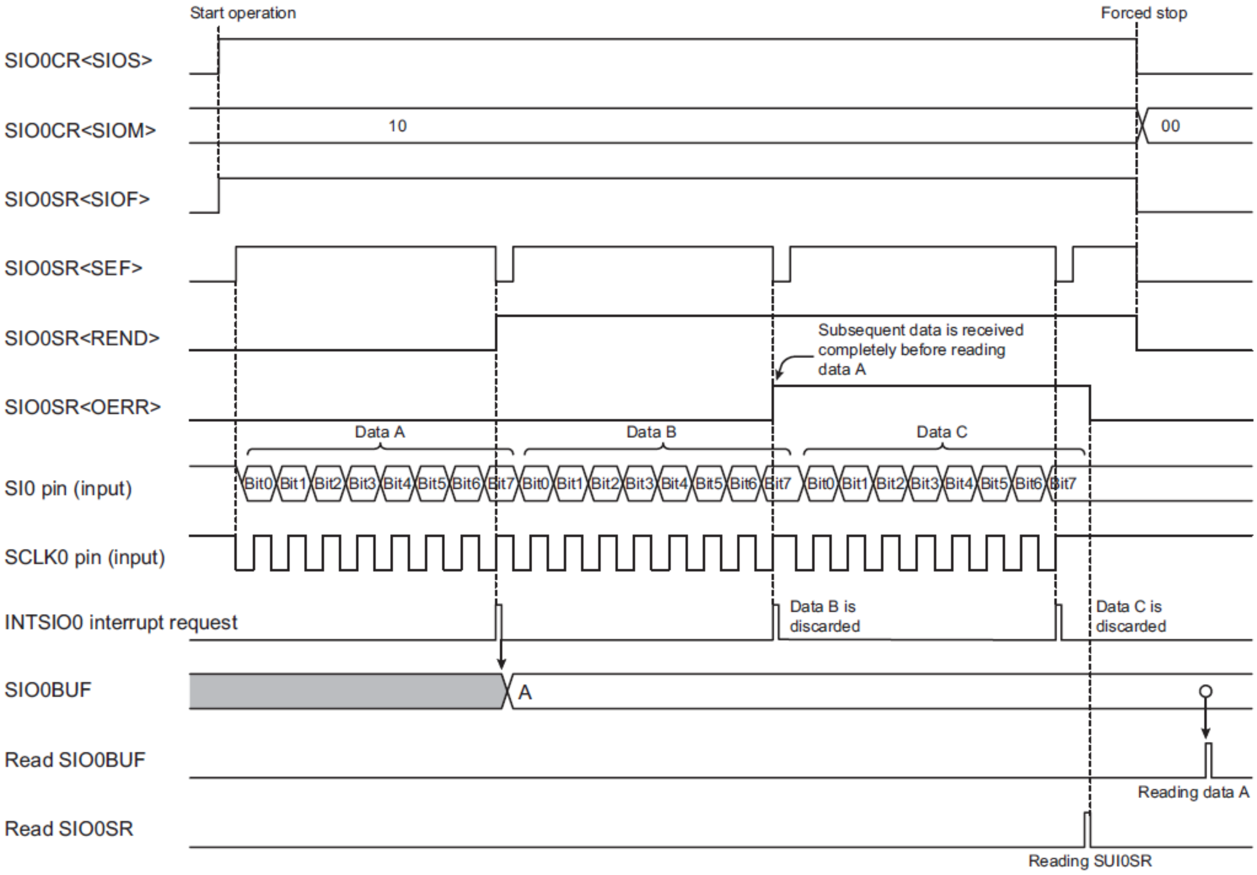


Figure 15.12 8-bit Receive Mode(External Clock and Occurrence of Overrun Error)

15.5.3 8-bit Transmit/receive mode

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

15.5.3.1 Setting

Before starting the transmit/receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

The transmit/receive operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIOCR<SIOS> is valid.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

15.5.3.2 Starting the transmit/receive operation

The transmit/receive operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and the serial data is transmitted from the SO0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>. At the same time, the serial data is received from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

The transmit data becomes undefined if the transmit/receive operation is started without writing any transmit data to SIO0BUF.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1" and an INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of data is received.

15.5.3.3 Transmit buffer and shift operation

If any data is written to SIO0BUF when the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

If any data is written to SIO0BUF when some data remains in the shift register, SIO0SR<TBFL> is set to "1". If new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

15.5.3.4 Operation on completion of transmission/ reception

When the data transmission/reception is completed, SIO0SR<REND> is set to "1" and an INTSIO0 interrupt request is generated. The operation varies depending on the operating clock.

(a) When the internal clock is used

If SIO0SR<TBFL> is "1", it is cleared to "0" and the transmit/receive operation continues. If SIO0SR<REND> is already "1", SIO0SR<OERR> is set to "1".

If SIO0SR<TBFL> is "0", the transmit/receive operation is aborted. The SCLK0 pin becomes the initial state and the SO0 pin becomes the "H" level. SIO0SR<SEF> remains at "0". When the subsequent data is written to SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the clock and the transmit/receive operation is restarted. To confirm the receive data, read it from SIO0BUF before writing data to SIO0BUF.

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

(b) When external clock is used

The transmit/receive operation continues. If the external serial clock is input without writing any data to SIO0BUF, the last data value set to SIO0BUF is re-transmitted. At this time, the transmit underrun error flag SIO0SR<UERR> is set to "1".

When the next 8-bit data is received completely before SIO0BUF is read, or in the state of SIO0SR<REND>="1", SIO0SR<OERR> is set to "1".

15.5.3.5 Stopping the transmit/receive operation

Set SIO0CR<SIOS> to "0" to stop the transmit/receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data is received completely. At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

If the internal clock has been used, the SO0 pin automatically returns to the "H" level. If an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the "H" level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit/receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the "H" level. If the internal clock is selected, the SCLK0 pin returns to the initial level.

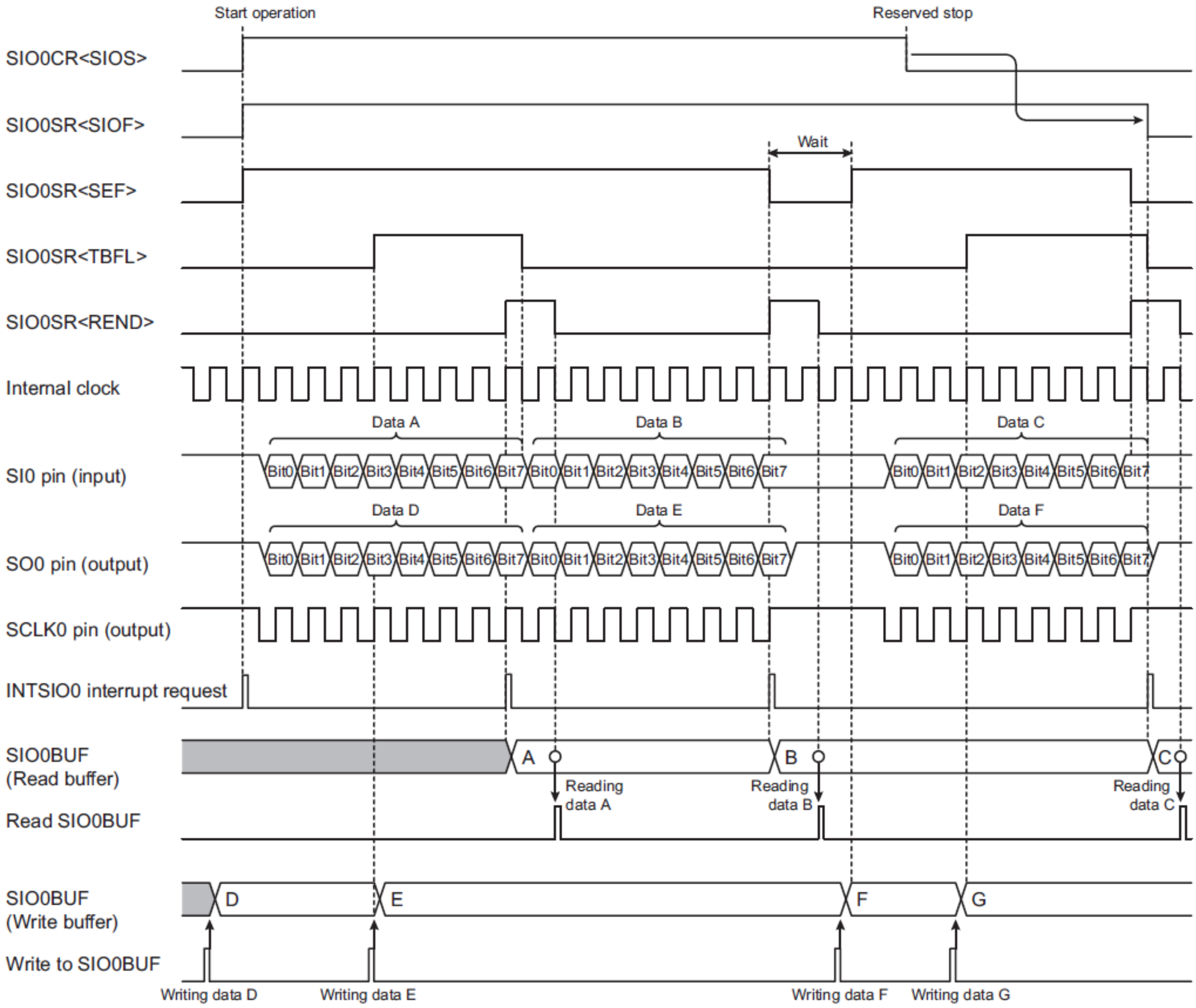


Figure 15.13 8-bit Transmit Receive mode(Internal clock and reserved stop)

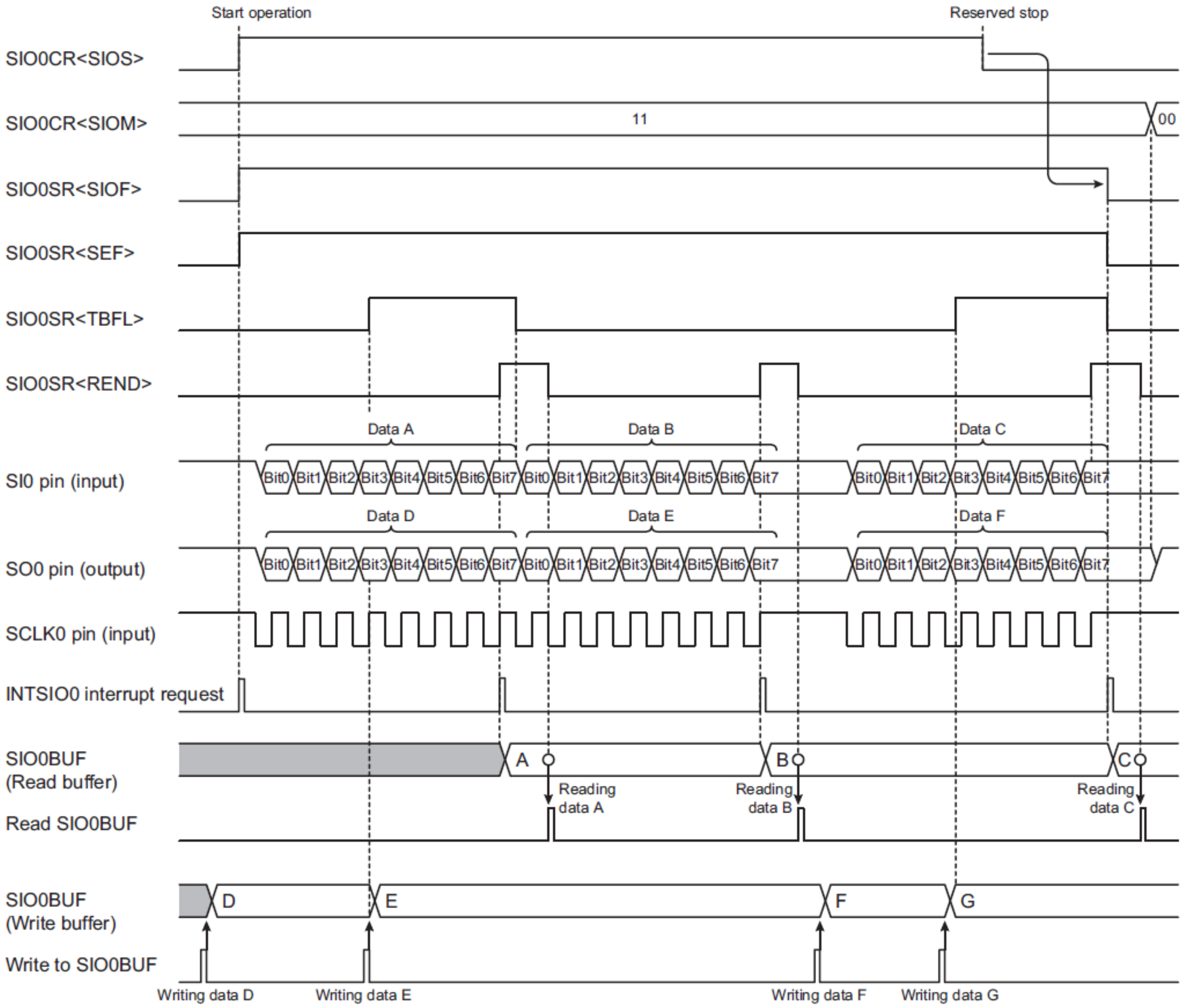


Figure 15.14 8-bit transmit/receive mode(external clock and reserved stop)

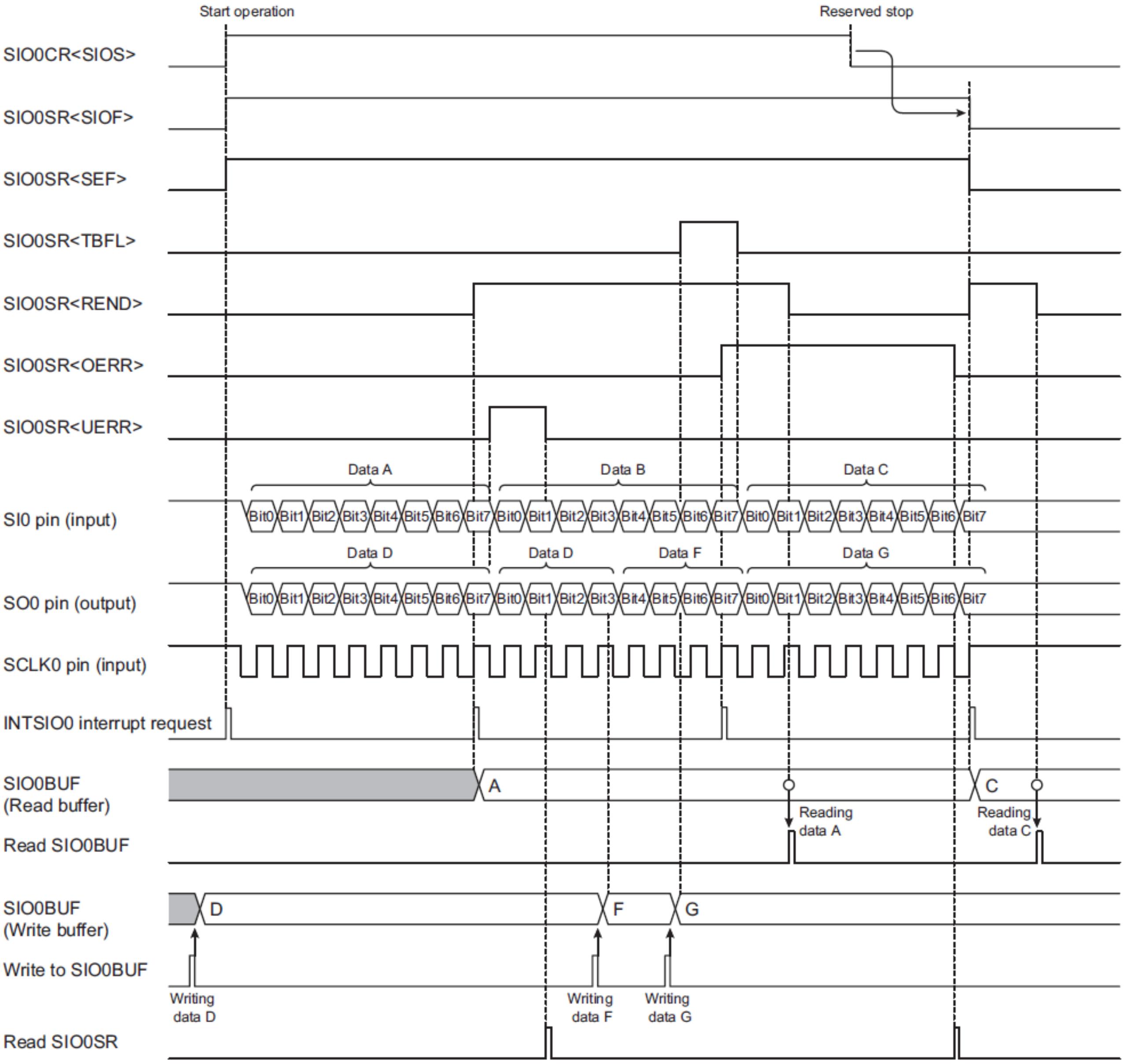


Figure 15.15 8-bit Transmit/receive mode(External clock, occurrence of transmit underrun error and occurrence of overrun error)

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

15.6 AC Characteristic

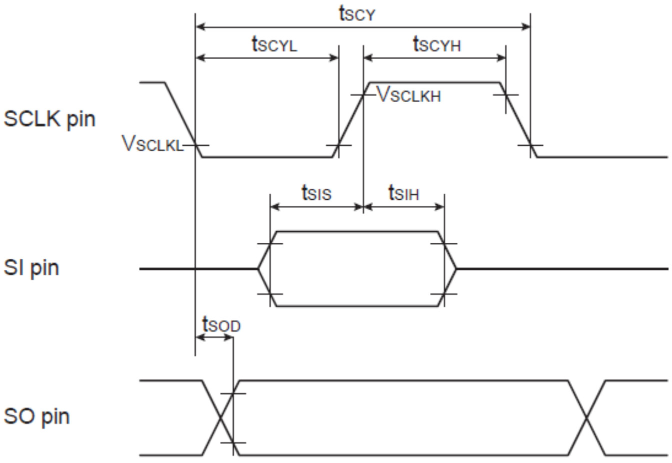
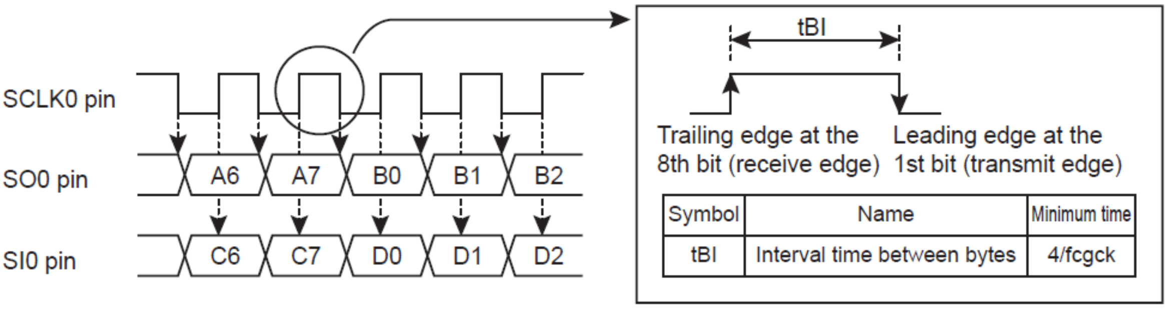


Figure 15.16 AC characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 V - 5.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
SCLK cycle time	t_{SCY}	Internal clock operation SO pin and SCLK pin load capacity=100 pF	2 / fcgck	-	-	ns
SCLK "L" pulse width	t_{SCYL}		1 / fcgck − 25	-	-	
SCLK "H" pulse width	t_{SCYH}		1 / fcgck − 15	-	-	
SI input setup time	t_{SIS}		60	-	-	
SI input hold time	t_{SIH}		35	-	-	
SO output delay time	t_{SOD}		−50	-	50	
SCLK cycle time	t_{SCY}	External clock operation SO pin and SCLK pin load capacity=100 pF	2 / fcgck	-	-	
SCLK "L" pulse width	t_{SCYL}		1 / fcgck	-	-	
SCLK "H" pulse width	t_{SCYH}		1 / fcgck	-	-	
SI input setup time	t_{SIS}		50	-	-	
SI input hold time	t_{SIH}		50	-	-	
SO output delay time	t_{SOD}		0	-	60	
SCLK low-level input voltage	t_{SCLKL}		0	-	V_{DD} × 0.30	V
SCLK high-level input voltage	t_{SCLKH}		V_{DD} × 0.70	-	V_{DD}	


SCLK0 pin
SO0 pin
SI0 pin
A6
A7
B0
B1
B2
C6
C7
D0
D1
D2
tBI
Trailing edge at the 8th bit (receive edge)
Leading edge at the 1st bit (transmit edge)

Symbol	Name	Minimum time
tBI	Interval time between bytes	4/fcgck

Figure 15.17 Interval time between bytes

Appendix A. In-System Programming Function (ISP)

MQ6935 has an in-system programming (ISP) function. Using a combination of this function and iMQ on-chip debug emulator MQ-Link, the user is able to perform software debugging in the on-board environment. This emulator can be operated from a debugger installed on a PC so that the emulation and debugging functions of an application program can be used to modify a program or for other purposes.

This chapter describes the control pins needed to use the ISP function and how a target system is connected.

Control Pins

The ISP function uses two pins for communication and four pins for power supply, reset and mode control. The pins used for the on-chip debug function are shown in Table A-1.

Ports P20 and P21 are used as communication control pins of the ISP function. If the on-Chip Debug Emulator is used, therefore, Ports P20 and P21 cannot be debugged as port pins or UART0 and SIO0 pins. However, because the UART0 and SIO0 functions can be assigned to other ports by using SERSEL<SRSEL2>, these communication functions can also be used during on-chip debug operation. For details, refer to the section of I/O ports.

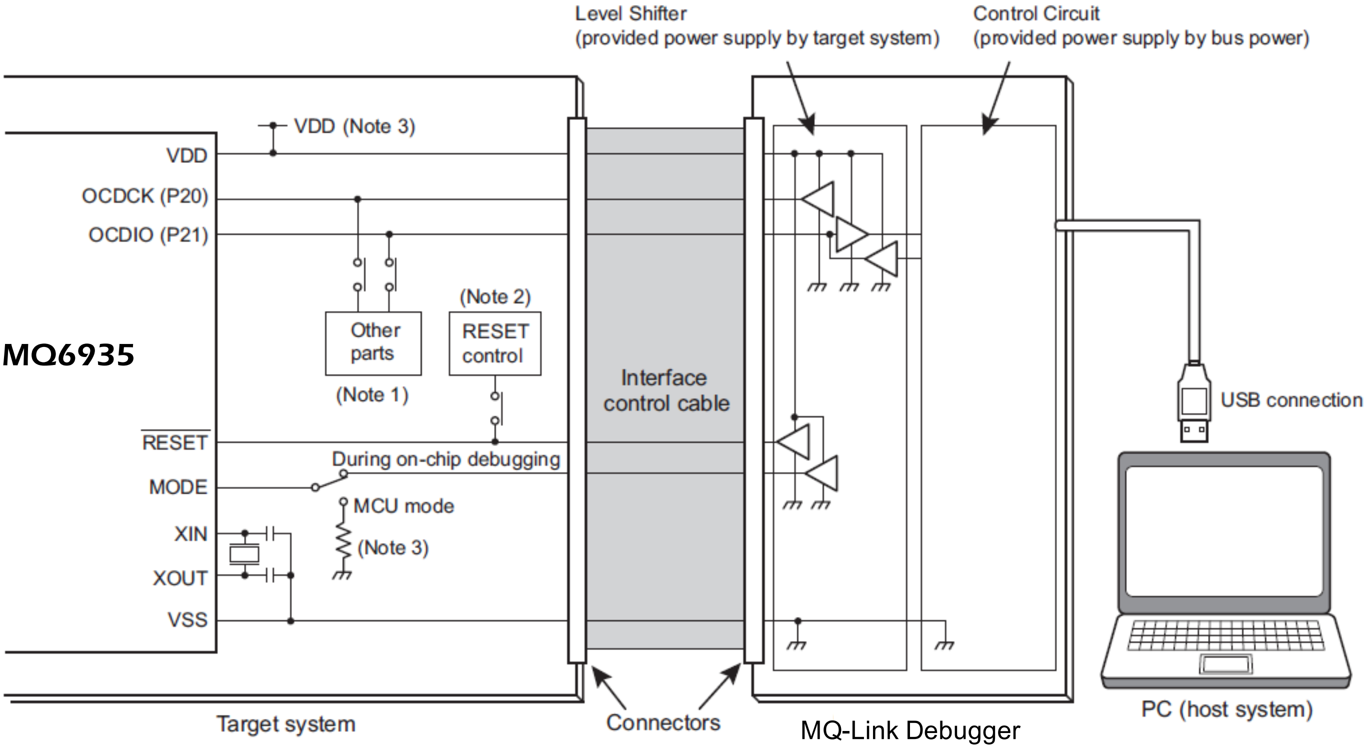
Pin Name (during ISP function)	Input/Output	Function		Pin Name (in MCU mode)
OCDCK	Input	Communication control pin (clock control)	(Note 1)	P20/TXD0/SO0
OCDIO	I/O	Communication control pin (data control)		P21/RXD0/SI0
RESETB	Input	Reset control pin		RESETB
MODE	Input	Mode control pin		Mode
VDD	Power Supply	5.0V		
VSS	Power Supply	0V		
Input and output ports other than P20 and P21	I/O	Can be used for an application in a target system		
XIN	Input	To be connected to and oscillator to put these pins in a state of self-oscillation		
XOUT	Output			

TableA.1 Pins Used for ISP Function

How to Connect MQ-Link Debugger to a Target System

To use the ISP function, the specific pins on a target system must be connected to the MQ-Link debugging system. MQ-Link can be connected to a target system via an interface control cable. iMQ provides a connector for this interface control cable as an accessory tool. Mounting this connector on a target system will make it easier to use the ISP function.

The connection between the MQ-Link and a target system is shown in Figure A.1.



FigureA.1 How to Connect MQ-Link Debugger to a Target System

Note 1): If the reset control circuit on an application board affects the control of the ISP function, it must be disconnected using a jumper, switch, etc.

Note 2): During the ISP function, the power supply of MQ6935 on target system is provided by MQ-Link debugger directly. After finishing ISP function, MQ6935 can use the original power supply on target system.

Note 3): For details of MQ-Link, please refer to "iMQ i87-IDE User Manual".

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.

Appendix B. Product Number Information

Example： MQ 69 35 LA 044 H V L R

iMQ

Product Series

Sub Series

Package Type

Code	Package Type	Code	Package Type
ST	SOT23	SD	SDIP
SP	SOP	LQ	LQFP 7x7
MS	MSOP	LA	LQFP 10x10
SS	SSOP	LE	LQFP 14x14
DP	PDIP	N4	QFN 4x4
TS	TSOP	N5	QFN 5x5
DS	TSSOP		

Pin Count

Code	Pin Count.	Code	Pin Count.
005	5	032	32
006	6	036	36
008	8	040	40
010	10	044	44
014	14	048	48
016	16	064	64
020	20	080	80
024	24	096	96
028	28	100	100

Program Flash

Data Flash

RAM

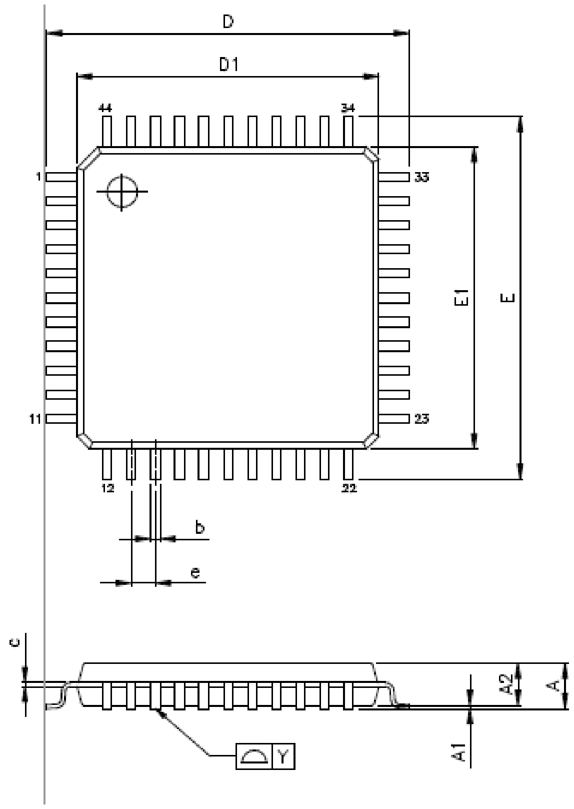
Code	Program Flash/ Data Flash/ RAM Size	Code	Program Flash/ Data Flash/ RAM Size
A	128 Bytes	K	24K Bytes
B	256 Bytes	M	32K Bytes
E	512 Bytes	N	40K Bytes
J	1K Bytes	P	48K Bytes
L	2K Bytes	S	64K Bytes
T	4K Bytes	U	96K Bytes
G	8K Bytes	W	128K Bytes
C	12K Bytes	V	N/A
H	16K Bytes		

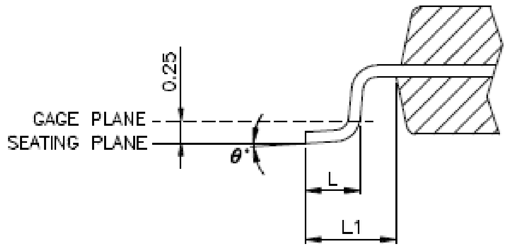
Operating Temp.

Code	Operating Temp.
R	-40~85°C
X	-40~105°C
T	-40~125°C

Appendix C. Package Dimensions

Product No.: MQ6935LA044HVLR





Symbol	MILLMETER		
	MIN	NOM	MAX
A	1.45	1.55	1.65
A1	0.015	–	0.21
A2	1.30	1.40	1.50
b	0.25	0.35	0.45
c	0.09	0.15	0.20
D	–	12	–
D1	–	10	–
E	–	12	–
E1	–	10	
e	–	0.8	–
L	0.42	–	0.75
L1	–	1	–
θ	0	–	10°
y	–	0.10	–

iMQ reserves the right to change the information in this document without prior notice. Please contact iMQ to obtain the latest version of product specification before placing your order. Use of iMQ devices in life support is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless iMQ from any and all damages, claims, suits or expenses resulting from such use.