# MQ6812/MQ6821/MQ6822
# Datasheet V1.1

# 1. Change History

| Version | Approved Date | Description |
|---|---|---|
| V 1.0 | 2018/5/3 | New release. |
| V1.1 | 2021/2/3 | 1. Update "CH2.3 Pin Assignment" , update MQ6822 pin-assignment and figure 2.2.<br>2. Update "Appendex C. Package Information" QFN 32(4x4) |

# 2. Product Overview

## 2.1 Features

### ◆ General information

- Wide operating voltage : 2.0V ~ 5.5V (2.6V ~ 5.5V, when LCD enable)
- Operating temperature: -40°C ~ 85°C
- i87 8-bit MCU core

### ◆ Memory

- 16K x 8 program Flash memory (endurance 100K times)
- 512 x 8 data memory RAM
- 64 x 8 information block

### ◆ I/O

- 28 bi-direction I/Os、2 Hi-driving I/O.
- Two 35mA LED driving (P80/P81)、other 27 I/O are 15mA output (excluding P10)
- Max. six 10-bit PPG output
- One16-bit PPG output
- Max. 16 external wake-up pin
- Max. 27 programble pull-up and pull-down I/Os
- 2 UART transfer/receive pins (details description please refer chapter 2.4)

### ◆ Instruction cycles

- Instruction cycle can set as 1/1、1/2、1/4、1/8 fc

### ◆ 1ch of UART

### ◆ 1sets of SIO

### ◆ 1ch of I²C/SIO

### ◆ 21 interrupt sources

- Max. 18 internal interrupt
- Max 3 external interrupt

### ◆ Clock Sources

- External crystal or internal oscillator
- Support 1MHz~16MHz, or 32kHz external crystal
- Internal high oscillator frequency 16MHz
- Internal Low oscillator frequency 24KHz

### ◆ LCD

- Max. LCD driver up to 8 COM X 12 SEG (DMA)
- Eight LCD modes: Static,1/8 duty(1/4 bias,1/3 bias,1/2 bias),1/4 duty(1/3 bias),1/3 duty (1/3 bias, 1/2 bias),1/2 duty(1/2 bias).

### ◆ Timer/Counter information

- Max. six 10-bit Timer (TCQ，with capture function)
  - TCQ00、TCQ02、TCQ04 with double buffer
  - TCQ01、TCQ03、TCQ05 without double buffer
- One 16-bit Timer (TCA，with capture function)
- Time Base Timer(TBT)
- Watch Dog Timer(WDT)
- Watch Dog Timer 2 (WDT2)
- Warm-up Counter (WUC)
- Real Time Clock (RTC)
- 8-bit divider output (DVO)

### ◆ Low Power Operation Modes

- 8 low power operation modes: STOP, SLOW1,SLOW2,IDLE0IDLE1,IDLE2SLEEP0,SLEEP1

### ◆ 2set 8-level voltage detect

### ◆ 10-bit AD converter(ADC)

- Max. 9 external ADC input
- One internal 1/4 VDD battery measure input pin
- Three ADC internal referenc voltage : 4V,3V,2V
- One external ADC referece voltage, voltage range:2.0~5.5V.

◆ In-system programming (ISP)

◆ Package information
- LQFP (7x7)/QFN 32 pin
- SSOP28/SOP28 pin
- SOP/SSOP/DIP20 pin

| Product no. | MQ6812LQ032HAER | MQ6812N4032HAER |
|---|---|---|
| Pin cunt./IOs | 32<br>(30) | 32<br>(30) |
| Operating voltage | 2.0~5.5V | 2.0~5.5V |
| Operating temp. | -40~85C | -40~85C |
| Ext. interrupt | 16 | 16 |
| Flash | 16K Bytes | 16K Bytes |
| RAM | 512 Bytes | 512 Bytes |
| ADC | 10-bit x 9-CH<br>(1/4 VDD, internal,<br>external)*1 | 10-bit x 9-CH<br>(1/4 VDD, internal,<br>external)*1 |
| LCD | 8x12 | 8x12 |
| Interrupt | External: 3<br>Internal: 18 | External: 3<br>Internal: 18 |
| Internal oscillator | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C |
| External crystal | 1~16MHz or<br>32768Hz | 1~16MHz or<br>32768Hz |
| Timer/<br>Counter | 10bit x 6<br>16bit x 1 | 10bit x 6<br>16bit x 1 |
| | WDT,TBT,<br>RTC,WUC | WDT,TBT,<br>RTC,WUC |
| PWM/PPG | 10bit x 6<br>16bit x 1 | 10bit x 6<br>16bit x 1 |
| LVD | 8 level<br>(+/- 0.1V)*2 | 8 level<br>(+/- 0.1V)*2 |
| Communication | UART x 1,<br>SIOx1, I2Cx1 | UART x 1,<br>SIOx1, I2Cx1 |
| ISP | Yes | YEs |
| Package type. | LQFP32 | QFN32 |

*1: "1/4 VDD" means there is one internal 1/4 VDD battery measure input pin. "Internal " means there is internal referenc voltage (4V,3V,2V). " External" means there one ne external ADC referece voltage
*2: There is 2 set of LVD, and each set is 4 level voltage, max. accuracy is +/- 0.1V.

| Product no. | MQ6812SP028HAER | MQ6822SP028HAER | MQ6812SS028HAER |
|---|---|---|---|
| Pin cunt./IOs | 28<br>(26) | 28<br>(26) | 28<br>(26) |
| Operating voltage | 2.0~5.5V | 2.0~5.5V | 2.0~5.5V |
| Operating temp. | -40~85C | -40~85C | -40~85C |
| Ext. interrupt | 13 | 16 | 13 |
| Flash | 16K Bytes | 16K Bytes | 16K Bytes |
| RAM | 512 Bytes | 512 Bytes | 512 Bytes |
| ADC | 10-bit x 9-CH<br>(1/4 VDD, internal, external)*1 | 10-bit x 9-CH<br>(1/4 VDD, internal, external)*1 | 10-bit x 9-CH<br>(1/4 VDD, internal, external)*1 |
| LCD | 8x8 | 8x12 | 8x8 |
| Interrupt | External: 2<br>Internal: 19 | External: 3<br>Internal: 15 | External: 2<br>Internal: 19 |
| Internal oscillator | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C |
| External crystal | 1~16MHz or<br>32768Hz | 1~16MHz or<br>32768Hz | 1~16MHz or<br>32768Hz |
| Timer/ Counter | 10bit x 6<br>16bit x 1 | 10bit x 2<br>16bit x 1 | 10bit x 6<br>16bit x 1 |
| | WDT,TBT,<br>RTC,WUC | WDT,TBT,<br>RTC,WUC | WDT,TBT,<br>RTC,WUC |
| PWM/PPG | 10bit x 6<br>16bit x 1 | 10bit x 2<br>16bit x 1 | 10bit x 6<br>16bit x 1 |
| LVD | 8 level<br>(+/- 0.1V)*2 | 8 level<br>(+/- 0.1V)*2 | 8 level<br>(+/- 0.1V)*2 |
| Communication | UART x 1,<br>SIOx1, $I^2C$x1 | UART x 1,<br>SIOx1, $I^2C$x1 | UART x 1,<br>SIOx1, $I^2C$x1 |
| ISP | Yes | Yes | Yes |
| Package type. | SOP28 | SOP28 | SSOP28 |

*1: "1/4 VDD" means there is one internal 1/4 VDD battery measure input pin. "Internal " means there is internal referenc voltage (4V,3V,2V). " External" means there one ne external ADC referece voltage

*2: There is 2 set of LVD, and each set is 4 level voltage, max. accuracy is +/- 0.1V.

| Product no. | MQ6821SP020HAER | MQ6821SS020HAER | MQ6821DP020HAER |
|---|---|---|---|
| Pin cunt./IOs | 20<br>(18) | 20<br>(18) | 20<br>(18) |
| Operating voltage | 2.0~5.5V | 2.0~5.5V | 2.0~5.5V |
| Operating temp. | -40~85C | -40~85C | -40~85C |
| Ext. interrupt | 8 | 8 | 8 |
| Flash | 16K Bytes | 16K Bytes | 16K Bytes |
| RAM | 512 Bytes | 512 Bytes | 512 Bytes |
| ADC | 10-bit x 6-CH<br>(1/4 VDD, internal,<br>external)*1 | 10-bit x6-CH<br>(1/4 VDD, internal,<br>external)*1 | 10-bit x6-CH<br>(1/4 VDD, internal,<br>external)*1 |
| Interrupt | External: 2<br>Internal: 17 | External: 2<br>Internal: 17 | External: 2<br>Internal: 17 |
| Internal oscillator | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C | 16MHz<br>+/- 1% @ 25C<br>+/- 2% @ 0~85C<br>+/- 3% @ -40~85C |
| External crystal | 1~16MHz or 32768Hz | 1~16MHz or<br>32768Hz | 1~16MHz or<br>32768Hz |
| Timer/<br>Counter | 10bit x 4<br>16bit x 1 | 10bit x 4<br>16bit x 1 | 10bit x 4<br>16bit x 1 |
| | WDT,TBT,<br>RTC,WUC | WDT,TBT,<br>RTC,WUC | WDT,TBT,<br>RTC,WUC |
| PWM/PPG | 10bit x 4<br>16bit x 1 | 10bit x 4<br>16bit x 1 | 10bit x 4<br>16bit x 1 |
| LVD | 8 level<br>(+/- 0.1V)*2 | 8 level<br>(+/- 0.1V)*2 | 8 level<br>(+/- 0.1V)*2 |
| Communication | UART x 1,SIO x1 | UART x 1,SIOx1 | UART x 1,SIOx1 |
| ISP | Yes | Yes | Yes |
| Package type. | SOP20 | SSOP20 | DIP20 |

*1: "1/4 VDD" means there is one internal 1/4 VDD battery measure input pin. "Internal " means there is internal referenc voltage (4V,3V,2V). " External" means there one ne external ADC referece voltage

*2: There is 2 set of LVD, and each set is 4 level voltage, max. accuracy is +/- 0.1V.

## 2.2 Block Diagram

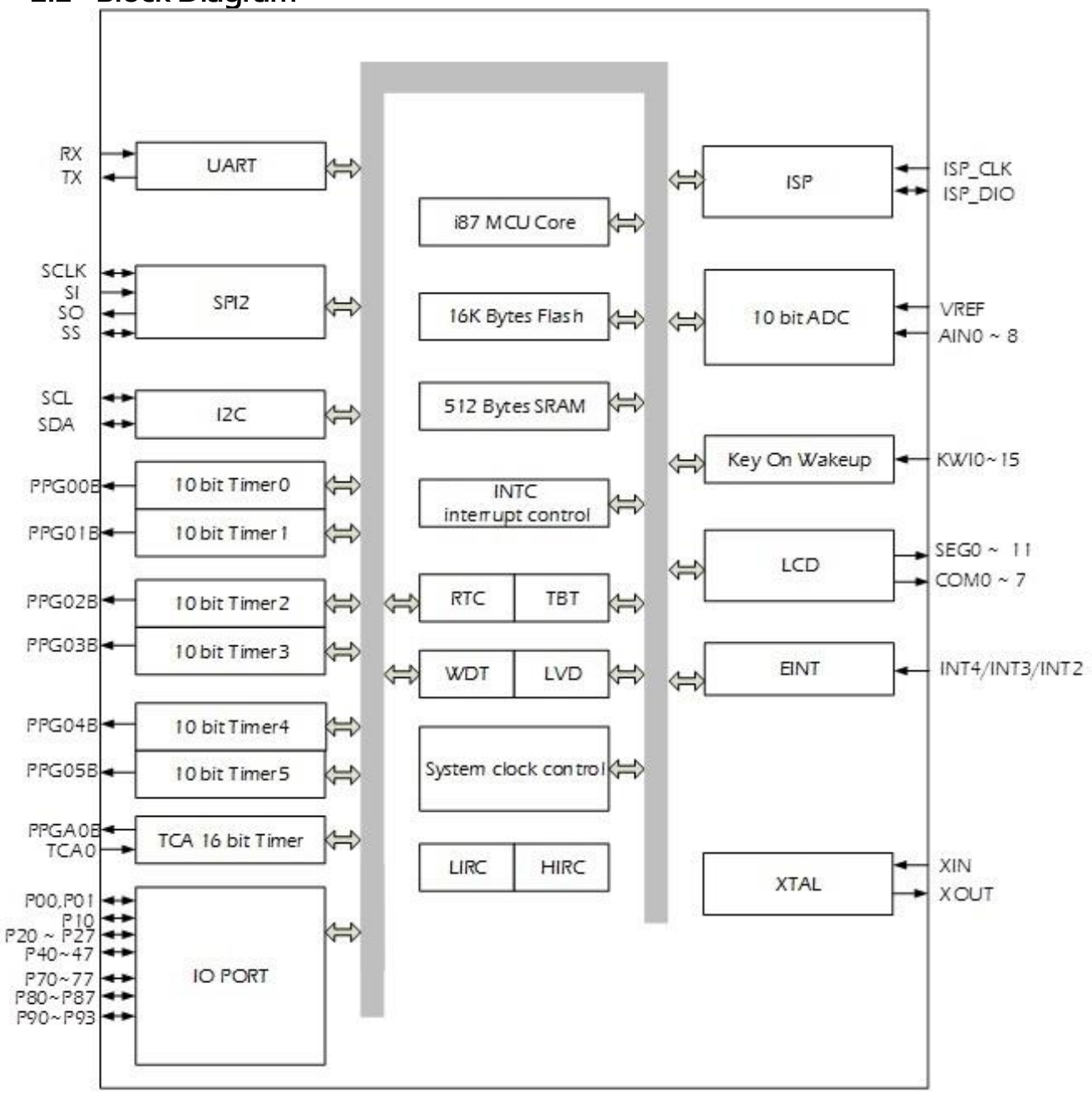
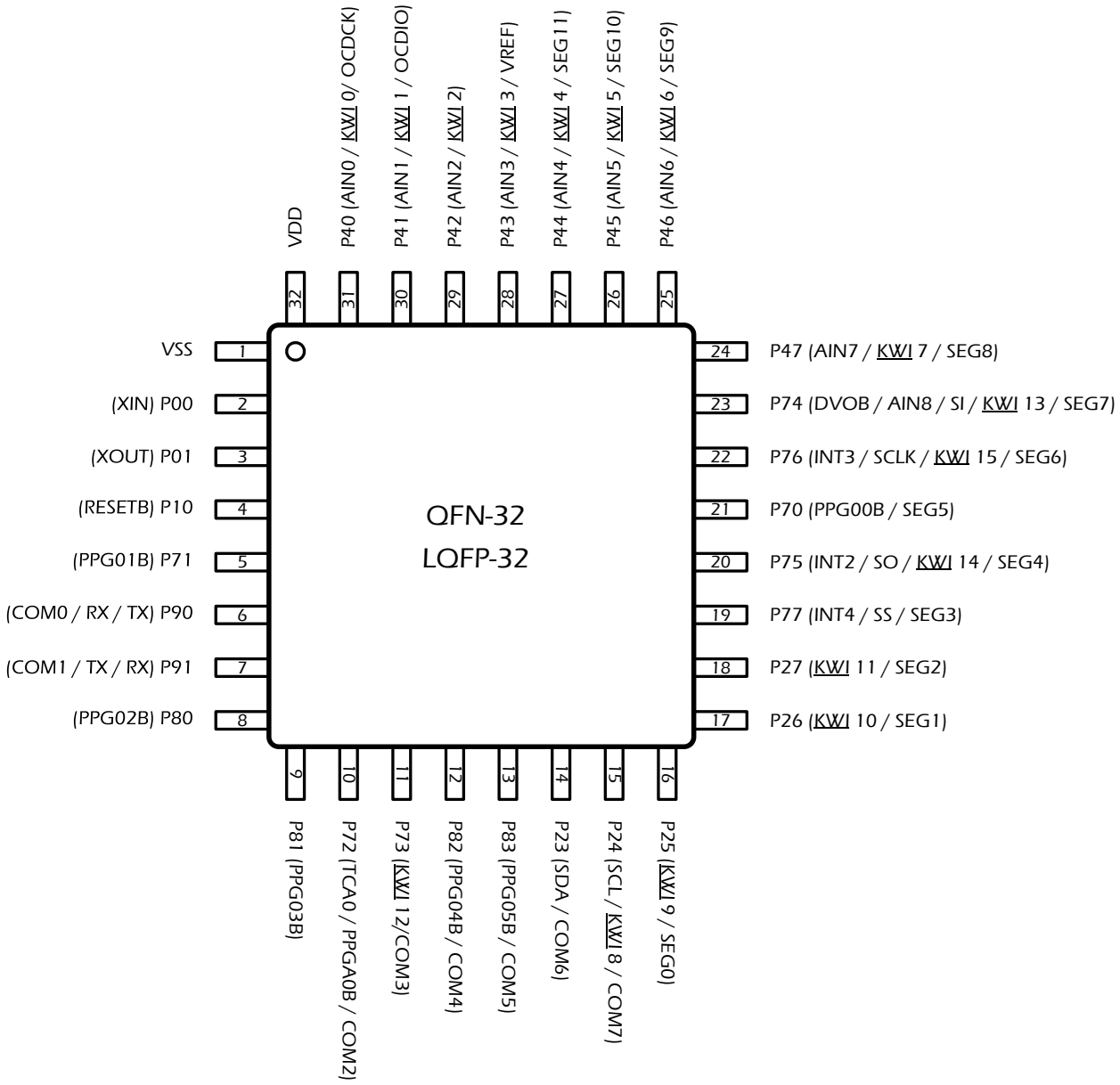


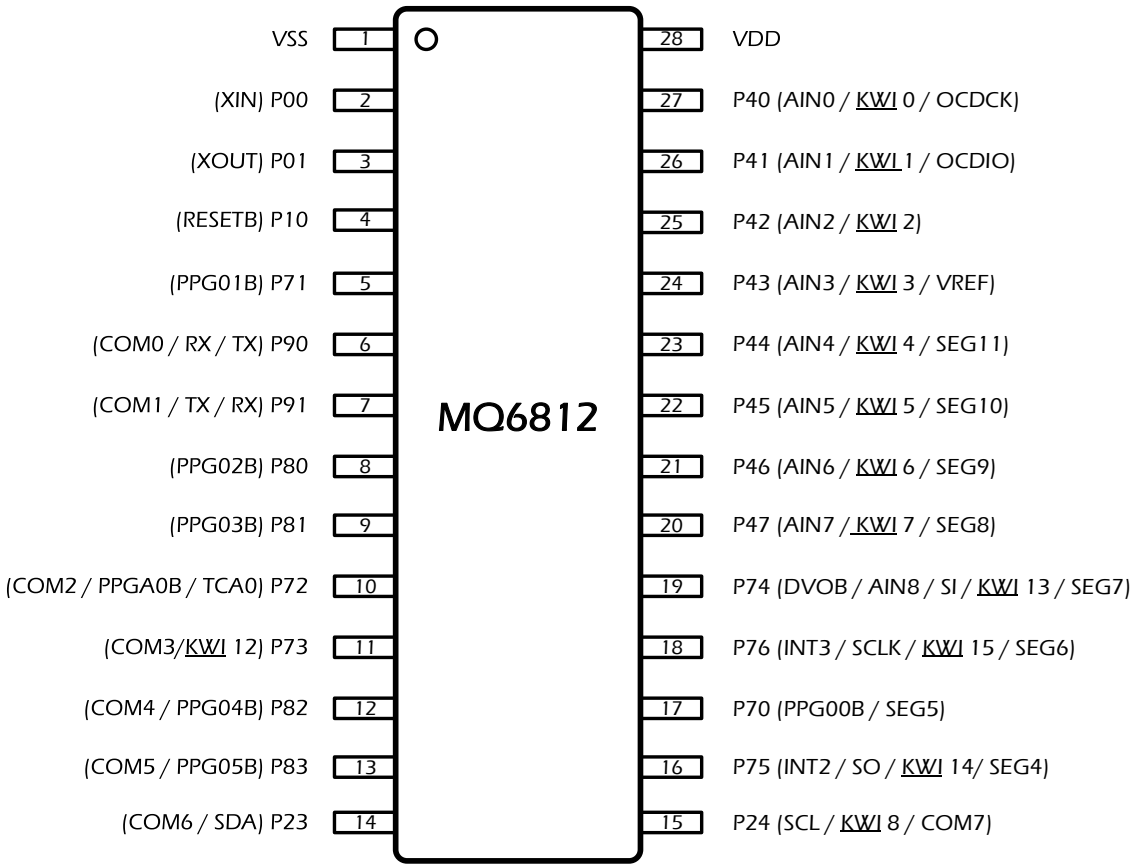

Figure 2.1 MQ6812/MQ6821 block diagram

## 2.3 Pin Assignment

QFN-32 / LQFP-32 (LCD: 8COM x 12 SEG)
Product No.: MQ6812LQ032HAER / MQ6812N4032HAER

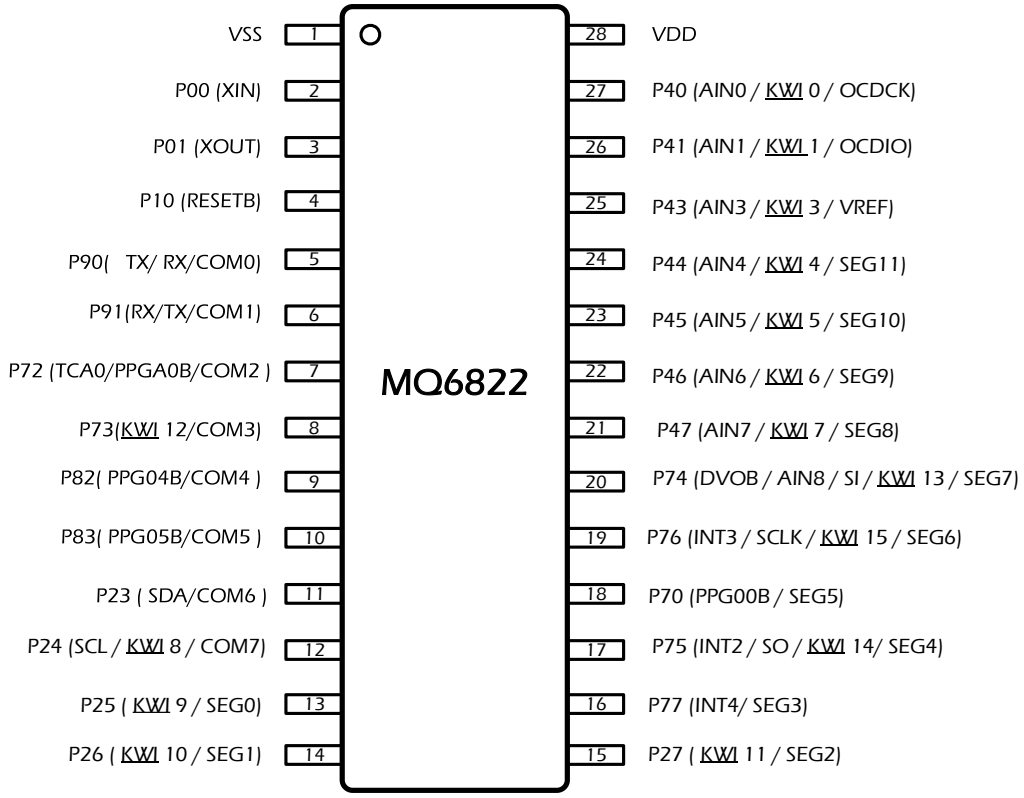


SOP-28/SSOP-28 (LCD: 8COM x 8 SEG)
Product No.:MQ6812SP028HAER /MQ6812SS028HAER

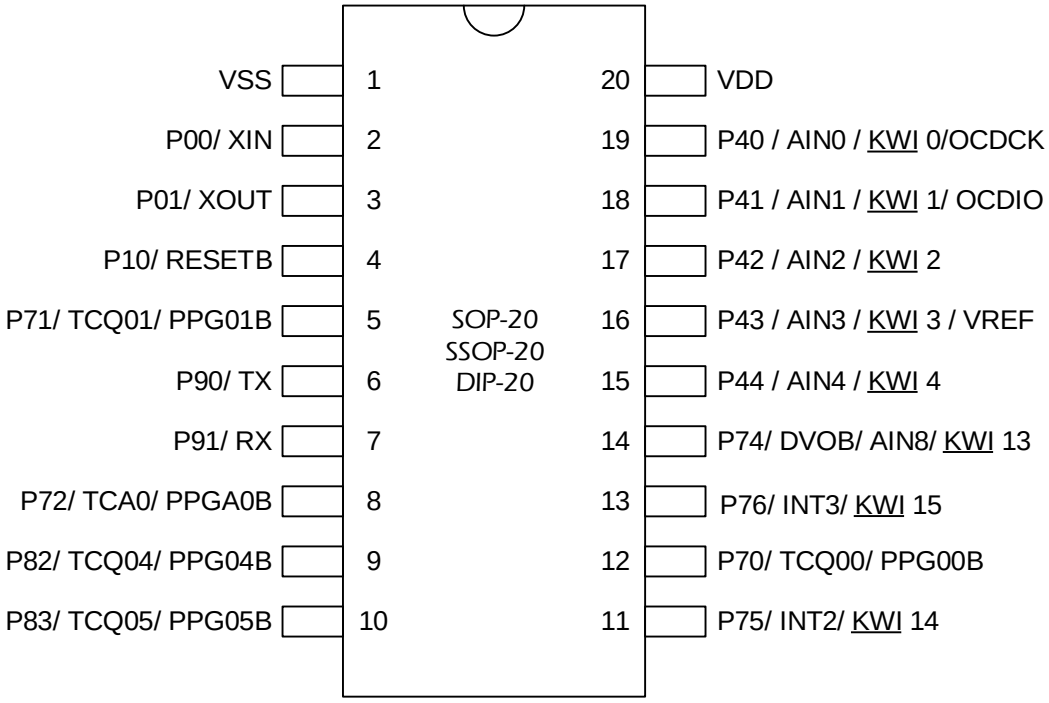


SOP-28 封装型态引脚配置(LCD: 8COM x 12 SEG)

产品型号 MQ6822SP028HAER



SOP-20 / SSOP-20 / DIP-20
Product No.: MQ6821SP020HAER/ MQ6821SS020HAER/ MQ6821DP020HAER



| 32pin No. | Pin Name | LCD (Note 7) | I/O Type | | Function Description |
|---|---|---|---|---|---|
| 1 | VSS | – | Power | | Negative power supply/ ground, |
| 2<br>3 | P00/XIN<br>P01/XOUT | – | I/O | Pull-up<br>Pull-down<br>Ext. crystal<br>(High/Low) | P00 and P01 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>XIN and XOUT are pin-shared with P00 and P01 respectively, and are connected to a high/low frequency external crystal for system clock. |
| 4 | P10/RESETB | – | I/O | Pull-up(note6) | P10 is bi-directional I/O pins, which are software configurable to be with pull-up resistors.<br>RESETB is pin-shared with P10, which is low-active. |
| 5 | P71/ PPG01B | – | I/O | Pull-up<br>Pull-down | P71 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>8-bit timer/counter pin is pin-shared with P71. |
| 6<br>7 | P90/TXD1/RXD1<br>P91/RXD1/TXD1 | COM0<br>COM1 | I/O | Pull-up<br>Pull-down<br>UART<br>LCD | P90, P91 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>UART pin TXD0/ RXD0, LCD driving pin COM0/ COM1,arepin-shared with P90, P91 respectively. |
| 8<br>9 | P80/ PPG02B<br>P81/ PPG03B | | I/O | Open drain | P80 and P81 are 35mA current driving output.<br>10-bit timer/counter TCQ02 and TCQ03 are pin-shared with P80, P81 respecttivly. |
| 10 | P72/TCA0/PPGA0B | COM2 | I/O | Pull-up<br>Pull-down<br>LCD | P72 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>16-bit timer pin TCA0/ PPGA0B,and LCD driving pin is pin-shared with P72. |
| 11 | P73/KWI 12<br>(note 1, note 6) | COM3 | I/O | Pull-up<br>Pull-down<br>Wake up<br>LCD | P73 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>Wakeup input pin KWI 12 and LCD driving pin COM3 is pin-shared with P73. |
| 12<br>13 | P82/PPG04B<br>P83/PPG05B | COM4<br>COM5 | I/O | Pull-up<br>Pull-down<br>LCD | P82 and P83 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>10-bit timer/counter TCQ04、 TCQ05 ,and LCD driving pin COM4、COM5 are pin-shared with P82 and P83. |
| 14 | P23/SDA | COM6 | I/O | Pull-up<br>Pull-down<br>$I^2C$<br>LCD | P23 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>$I^2C$ pin SDA and LCD driving pin COM6 is pin-shared with P23.。 |
| 15 | P24/SCL./KWI 8 (note 1) | COM7 | I/O | Pull-up<br>Pull-down<br>$I^2C$<br>Wake up<br>LCD | P24 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>$I^2C$ pin SCL and wakeup input pin KWI 8,and LCD driving pin COM7/ SEG31 is pin-shared with P24.。 |

| 32pin No. | Pin Name | LCD (Note 7) | I/O Type | | Function Description |
|---|---|---|---|---|---|
| 16 | P25/KWI 9 (note 1) | SEG0 | I/O | Pull-up<br>Pull-down<br>Wake up<br>LCD | P25 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>Wakeup input pin KWI 9 and LCD driving pin SEG0 is pin-shared with P25. |
| 17<br>18 | P26/KWI 10<br>P27/KWI 11<br>(注1) | SEG1<br>SEG2 | I/O | Pull-up<br>Pull-down<br>Wake up<br>LCD | P26 and P27 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>Wakeup input pin KWI 10, KWI 11 ,and LCD driving pin SEG1, SEG2 are pin-shared with P26, P27 respectiely. |
| 19 | P77/INT4/SS | SEG3 | I/O | Pull-up<br>Pull-down<br>Ext. interrupt<br>SIO<br>LCD | P77 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>External interrupt input pin INT4 and (SIO)SS, and LCD driving pin SEG3 is pin-shared with P77. |
| 20 | P75/INT2/ SO/ KWI 14<br>(note 1) | SEG4 | I/O | Pull-up<br>Pull-down<br>Ext. interrupt<br>SIO<br>Wake up<br>LCD | P75 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>External interrupt input pin INT2, SIO pin SO, wakeup input pin KWI 14 and LCD driving SEG4 is pin-shared with P75. |
| 21 | P70/ PPG00B | SEG5 | I/O | Pull-up<br>Pull-down<br>LCD | P70 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>8-bit timer/counter pin PPG01B and LCD driving pin SEG5 is pin-shared with P70. |
| 22 | P76/INT3/SCLK/KWI 15<br>(注1) | SEG6 | I/O | Pull-up<br>Pull-down<br>SIO<br>Wake up<br>LCD | P76 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>SIO pin SCLK and wakeup input pin KWI 15 and LCD driving pin SEG6 is pin-shared with P76. |
| 23 | P74/DVOB/AIN8/SI/KWI 13<br>(note1,note 2, note 8) | SEG7 | I/O | Pull-up<br>Pull-down<br>ADC input<br>SIO<br>Wakeup<br>LCD | P74 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input pin AIN8 and SIO pin SI, DVO, wakeup input pin KWI 13, and LCD driving pin SEG7 is pin-shared with P74. |
| 24<br>25<br>26<br>27 | P47/AIN7/KWI7<br>P46/AIN6/KWI6<br>P45/AIN5/KWI5<br>P44/AIN4/KWI4<br>(note1,note2) | SEG8<br>SEG9<br>SEG10<br>SEG11 | I/O | Pull-up<br>Pull-down<br>ADC input<br>Wakeup<br>LCD | P47,P46,P45, and P44 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input AIN7, AIN6,AIN5,AIN4 and wakeup pin KWI 7, KWI 6, KWI 5, KWI 4 are pin-shared with P47, P46,P45 and P44 respectively. |
| 28 | P43/AIN3/KWI 3/VREF<br>(note1, note2) | -- | I/O | Pull-up<br>Pull-down<br>ADC input<br>Wakeup | P43 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input pin AIN3 and wakeup pin KWI 3, VREF is pin-shared with P43. |

| 32pin No. | Pin Name | LCD (Note 7) | I/O Type | | Function Description |
|---|---|---|---|---|---|
| 29<br>30<br>31 | P42/AIN2/KWI 2<br>P41/AIN1/KWI 1/OCDIO<br>P40/AIN0/KWI 0/OCDCK<br>(note1,note2) | – | I/O | Pull-up<br>Pull-down<br>ADC input<br>Wakeup<br>OCDE | P42,P41,and P40 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors. ADC input AIN2, AIN1,AIN0 and wakeup pin KWI 2、KWI 1、KWI 0 are pin-shared with P42, P41 and P40 respectively. |
| 32 | VDD | | Power | | Positive power supply |

| 20pin No. | Pin Name | I/O Type | | Function Description |
|---|---|---|---|---|
| 1 | VSS | Power | | Negative power supply/ ground, |
| 2<br>3 | P00/XIN<br>P01/XOUT | I/O | Pull-up<br>Pull-down<br>Ext. crystal (High/Low) | P00 and P01 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>XIN and XOUT are pin-shared with P00 and P01 respectively, and are connected to a high/low frequency external crystal for system clock. |
| 4 | P10/RESETB | I/O | Pull-up(note6) | P10 is bi-directional I/O pins, which are software configurable to be with pull-up resistors.<br>RESETB is pin-shared with P10, which is low-active. |
| 5 | P71/ TCQ01/PPG01B | I/O | Pull-up<br>Pull-down | P71 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>8-bit timer/counter pin is pin-shared with P71. |
| 6<br>7 | P90/TXD1/RXD1<br>P91/RXD1/TXD1 | I/O | Pull-up<br>Pull-down<br>UART<br>LCD | P90, P91 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>UART pin TXD0/ RXD0, LCD driving pin COM0/ COM1,arepin-shared with P90, P91 respectively. |
| 8 | P72/TCA0/PPGA0B | I/O | Pull-up<br>Pull-down<br>LCD | P72 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>16-bit timer pin TCA0/ PPGA0B,and LCD driving pin is pin-shared with P72. |
| 9<br>10 | P82/PPG04B<br>P83/PPG05B | I/O | Pull-up<br>Pull-down<br>LCD | P82 and P83 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>10-bit timer/counter TCQ04、 TCQ05 ,and LCD driving pin COM4、COM5 are pin-shared with P82 and P83. |
| 11 | P75/INT2/ SO/ KWI 14<br>(note 1) | I/O | Pull-up<br>Pull-down<br>Ext. interrupt<br>SIO<br>Wake up<br>LCD | P75 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>External interrupt input pin INT2, SIO pin SO, wakeup input pin KWI 14 and LCD driving SEG4 is pin-shared with P75. |
| 12 | P70/ PPG00B | I/O | Pull-up<br>Pull-down<br>LCD | P70 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>8-bit timer/counter pin PPG01B and LCD driving pin SEG5 is pin-shared with P70. |
| 13 | P76/INT3/SCLK/KWI 15<br>(note 1) | I/O | Pull-up<br>Pull-down<br>SIO<br>Wake up<br>LCD | P76 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>SIO pin SCLK and wakeup input pin KWI 15 and LCD driving pin SEG6 is pin-shared with P76. |

| 20pin No. | Pin Name | I/O Type | | Function Description |
|---|---|---|---|---|
| 14 | P74/DVOB/AIN8/SI/KWI 13 (note1,note 2, note 8) | I/O | Pull-up<br>Pull-down<br>ADC input<br>SIO<br>Wakeup | P74 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input pin AIN8 and SIO pin SI, DVO, wakeup input pin KWI 13, and LCD driving pin SEG7 is pin-shared with P74. |
| 15 | P44/AIN4/KWI 4 (note1,note2) | I/O | Pull-up<br>Pull-down<br>ADC input<br>Wakeup | P44 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input pin AIN4 and wakeup pin KWI 4 is pin-shared with P43. |
| 16 | P43/AIN3/KWI 3/VREF (note1, note2) | I/O | Pull-up<br>Pull-down<br>ADC input<br>Wakeup | P43 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input pin AIN3 and wakeup pin KWI 3, VREF is pin-shared with P43. |
| 17<br>18<br>19 | P42/AIN2/KWI 2<br>P41/AIN1/KWI 1/OCDIO<br>P40/AIN0/KWI 0/OCDCK<br>(note1,note2) | I/O | Pull-up<br>Pull-down<br>ADC input<br>Wakeup<br>OCDE | P42,P41,and P40 are bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors.<br>ADC input AIN2, AIN1,AIN0 and wakeup pin KWI 2、KWI 1、KWI 0 are pin-shared with P42, P41 and P40 respectively. |
| 20 | VDD | Power | | Positive power supply |

*Note 1：KWI 0~KWI 15 can define to system wake up input pin. KWI 0~KWI7 are pin-shared with P40~P47, and KWI 8~KWI 15 are pin-shared with P24~P27, 73~P76 respectively.*

*Note 2：AIN0~AIN8 and AIN13~AIN15 are10-bit ADC input pins.*

*Note 3: OCD pin are VSS, VDD, P10, P40 and P41. Please note, under OCDE mode, P10,P40 and P40 can not execute the pin function.*

*Note4: LCD enablem please set LCD driving pin(COM and SEG) as input mode. And keep LCD operating voltage same with VDD of MCU.*

*Note5 : If using P74 as ADC input(AIN8), and also enable other ADC input pin(AIN), it will cost extra power in stop mode. Please avoid P74 as AIN function(AIN8) if possible.*

*Note6: The I/O pins which need to be pull-high or pull-low should be connected to VDD or GND through a resistor (resistance >100ohm). To connect I/O pins to VDDor GND directly should be avoided.*

Figure 2.2 is MQ6812/MQ6822/MQ6821 suggestion external reference circuit，as below 3 parts:

1. Add 10uF and 0.1uF capacitance parallel connection to VDD and VSS. In order to avoid power surge or noise effect, it can also enhance MCU EFTB performance. The capacitance position should be close to VDD and VSS pin as close as possible.

2. Use ADC function, please series a 100ohm and 1nF capacitance to groud. This can filter noise.

3. Connecting 0.1uF capacitor where is near IC, and add (10uF +0.1 uF) capacitor closed to the power jack are recommended. Use ADC and ADC external reference voltage, please add 10uF and 0.1uF capacitance parallel connection,this can filter noise.

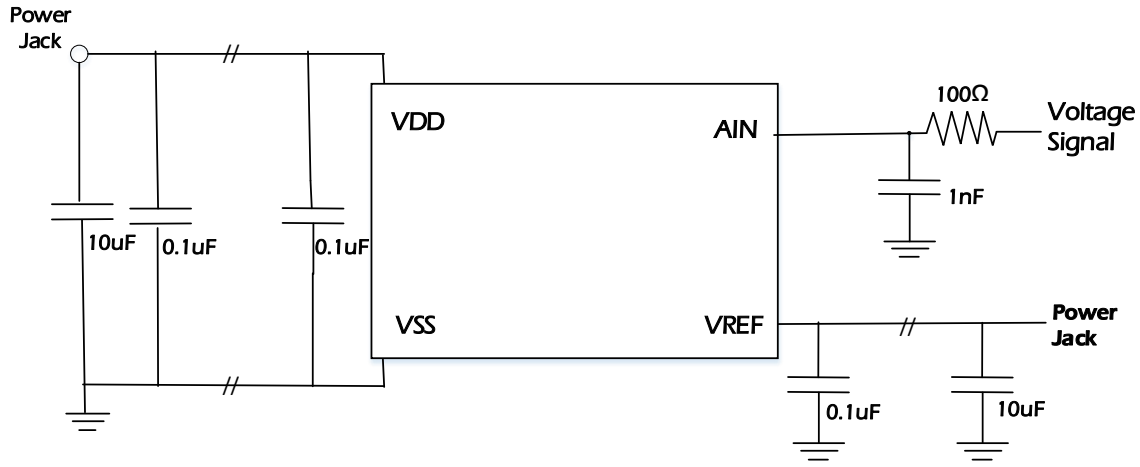



Figure 2.2 recommended external circuit

# 3. Electronic Characteristics

## 3.1 Absolute Maximum Ratings

The absolute maximum ratings are rated value 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 | Max. | Unit |
|---|---|---|---|---|
| Operating voltage | $V_{DD}$ | | -0.3 to 6.0 | V |
| Input voltage | $V_{IN}$ | 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) | $I_{OUT1}$ | P10 (IOL) | 15 | mA |
| | $I_{OUT2}$ | All I/O pins ,excluding P10 (IOL) | 40 | |
| | $I_{OUT3}$ | All I/O pins (IOH) | -15 | |
| | $I_{OUT4}$ | P80/P81 LED driving port) | 60 | |
| Output current (total) | $\Sigma I_{OUT1}$ | All I/O pins (IOL) | 120 | mA |
| | $\Sigma I_{OUT2}$ | All I/O pins (IOH) | 60 | |
| 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 | Typ. | Max | Unit |
|---|---|---|---|---|---|---|---|---|
| Supply voltage | | $V_{DD}$ | LCD disable | 所有工作模式 | 2.0 | - | 5.5 | V |
| | | | LCD enable | | 2.6 | - | 5.5 | V |
| Input high level | | $V_{IH}$ | All I/O pins | | $V_{DD}$x0.75 | - | $V_{DD}$ | V |
| Input low level | | $V_{IL}$ | All I/O pins | | 0 | - | $V_{DD}$x0.25 | V |
| Clock Frequency | Ext. crystal (high) | $f_C$ | XIN, XOUT | $V_{DD}$ = 2.0 to 5.5V | 1.0 | - | 16 | MHz |
| | Ext. crystal (low) | $f_{CL}$ | XIN, XOUT (32.768KHz) | | 30.0 | 32.768 | 34.0 | KHz |
| | Internal crystal (high) | $f_{OSC}$ | FSCTRL<FOSCCKS>="01" - 16MHz | $V_{DD}$ = 2.0 to 5.5V 25°C | -1% | 16.00 | +1% | MHz |
| | | | | $V_{DD}$ = 2.0 to 5.5V 0°C ~85°C | -2% | | +2% | MHz |
| | | | | $V_{DD}$ = 2.0 to 5.5V -40°C ~85°C | -3% | | +3% | MHz |
| | Internal crystal (low) | $f_{OSCL}$ | 24KHz | $V_{DD}$ = 2.0 to 5.5V | - | 24 | - | KHz |
| | System clock | $f_{CGCK}$ | Set CGCR <FCGCKSEL> | $V_{DD}$ = 2.0 to 5.5V | 0.125 | - | 16 | MHz |

## 3.3 DC Characteristics

($V_{SS}$ = 0V, $T_{OPR}$ = -40 to 85°C)

| Parameter | Symbol | Pins | Condition | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|---|
| Hysteresis voltage | $V_{HS}$ | All I/O pins | $V_{DD}$ = 5.5V $V_{IN}$ = 5.5V / 0V | – | 0.9 | – | V |
| Input current | $I_{IN}$ | All I/O pins | | – | – | ±2 | μA |
| Pull-up resistance | $R_{UP}$ | All I/O pins、excluding P80/P81、P10 RESETB disalbe | | 30 | 50 | 70 | KΩ |
| Pull-down resistance | $R_{DN}$ | All I/O pins、excluding P10/P80/P81 | $V_{DD}$ / $V_{IN}$ = 5.5V | 27.5 | 55 | 88 | KΩ |
| | | | $V_{DD}$ / $V_{IN}$ = 2.0V | – | 200 | – | KΩ |
| Output leakage current | $I_{OL1}$ | P10 | $V_{DD}$ = 5.5V $V_{OL}$ = 0.55V | 3.0 | 5.0 | – | mA |
| | $I_{OL2}$ | All I/O pins、excluding P10/P80/P81 | | 9.0 | 15.0 | – | mA |
| | $I_{OL3}$ | P80/P81 | | 21.0 | 35.0 | - | mA |
| | $I_{OH1}$ | All I/O pins、excluding P80/P81 | $V_{DD}$ = 5.5V $V_{OH}$ = 4.95V | 3.0 | 5.0 | – | mA |
| | $I_{OH2}$ | P80/P81 | | 9.0 | 15.0 | - | mA |

*Note2 : Data in "Typ." column is at $T_{OPR}$ = 25°C, $V_{DD}$ = 5.0V, unless otherwise stated.*

($V_{SS}$ = 0V, $T_{OPR}$ = 25℃)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|
| Operating current in NORMAL 1, 2 modes | $I_{DD}$ | $V_{DD}$ = 5.5V<br>$f_{cgck}$ = 16.0 MHz<br>$f_S$ = 24 KHz | – | 3.8 | 4.4 | mA |
| Operating current in IDLE0, 1, 2 modes | | | – | 2.2 | 2.6 | |
| Operating current in NORMAL 1, 2 modes | | $V_{DD}$ = 5.5V<br>$f_{cgck}$ = 8.0 MHz<br>$f_S$ = 24 KHz | – | 3.1 | 3.7 | |
| Operating current in IDLE0, 1, 2 modes | | | – | 2.0 | 2.4 | |
| Operating current in SLOW1 modes | | $V_{DD}$ = 3.0V<br>$f_S$ = 24 KHz | – | 32 | 45 | μA |
| Operating current in SLEEP1 modes | | | – | 19 | 28 | |
| Operating current in SLEEP0 modes | | | – | 19 | 28 | |
| Operating current in STOP modes | | $V_{DD}$ = 5.5V | – | 8 | 12 | |

*Note1：Data in "Typ." column is at $T_{OPR}$ = 25℃, $V_{DD}$ = 5.0V, unless otherwise stated.*

*Note2：Each supply current in SLOW2 mode is equivalent to that in IDLE0, IDLE1 and IDLE2 modes.*

($V_{SS}$ = 0V, $T_{OPR}$ = -40 to 85℃)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|
| Operating current in NORMAL 1, 2 modes | $I_{DD}$ | $V_{DD}$ = 5.5V<br>$f_{cgck}$ = 16.0 MHz<br>$f_S$ = 24 KHz | – | 3.8 | 4.8 | mA |
| Operating current in IDLE0, 1, 2 modes | | | – | 2.2 | 3.0 | |
| Operating current in NORMAL 1, 2 modes | | $V_{DD}$ = 5.5V<br>$f_{cgck}$ = 8.0 MHz<br>$f_S$ = 24 KHz | – | 3.1 | 4.1 | |
| Operating current in IDLE0, 1, 2 modes | | | – | 2.0 | 2.8 | |
| Operating current in SLOW1 modes | | $V_{DD}$ = 3.0V<br>$f_S$ = 24 KHz | – | 32 | 148 | μA |
| Operating current in SLEEP1 modes | | | – | 19 | 129 | |
| Operating current in SLEEP0 modes | | | – | 19 | 128 | |
| Operating current in STOP modes | | $V_{DD}$ = 5.5V | – | 8 | 92 | |

*Note1：Data in "Typ." column is at $T_{OPR}$ = 25℃, $V_{DD}$ = 5.0V, unless otherwise stated.*

*Note2：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 | $V_{REF}$ | – | – | – | $V_{DD}$ | V |
| Analog input voltage range | $V_{AIN}$ | – | $V_{SS}$ | – | $V_{DD}$ | V |
| Conversion Time | | fcgck = 2MHz<br>ADCCR2 <ACK> = "000" | – | 16.0 | – | μs |
| Differential Nonlinearity Error (DNL) | | – | – | – | ±2.0 | LSB |
| Integral Nonlinearity Error (INL) | | – | – | – | ±2.0 | LSB |
| Zero Point Error | | – | – | – | ±2.0 | LSB |
| Full Scale Error | | – | – | – | ±2.0 | LSB |
| Total Error | | – | – | – | ±2.0 | LSB |

($V_{SS}$ = 0V, 2.0V ≦ $V_{DD}$< 2.7V, $T_{OPR}$ = 25°C)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|
| Analog Reference Voltage | $V_{REF}$ | – | – | – | $V_{DD}$ | V |
| Analog input voltage range | $V_{AIN}$ | – | $V_{SS}$ | – | $V_{DD}$ | V |
| Conversion Time | | fcgck = 2MHz<br>ADCCR2 <ACK> = "001" | – | 32.0 | – | μs |
| Differential Nonlinearity Error (DNL) | | – | – | – | ±4.0 | LSB |
| Integral Nonlinearity Error (INL) | | – | – | – | ±4.0 | LSB |
| Zero Point Error | | – | – | – | ±4.0 | LSB |
| Full Scale Error | | – | – | – | ±4.0 | LSB |
| Total Error | | – | – | – | ±4.0 | LSB |

*Note1： 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 $V_{REF}$ 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.*

## 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 write to flash memory | | – | – | 100,000 | times |
| Flash memory write time | | – | – | 40 | μs |
| Flash memory erase time | chip erase | – | – | 40 | ms |
| | sector erase (1 sector = 128 Bytes) | – | – | 5 | |

## 3.6 LCD Characteristics

($V_{SS}$ = 0V, $T_{OPR}$ = 25°C)

| Parameter | Symbol | Condition | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|
| LCD operating volateg | $V_{DD}$ | | 2.6 | – | 5.5 | V |
| LCD bias output 1 | $V_{L1}$ | 1/4 bias LCD, $V_{DD}$ = 5.0V | 3.19 | 3.75 | 4.31 | V |
| | | 1/3 bias LCD, $V_{DD}$ = 5.0V | 2.83 | 3.33 | 3.83 | V |
| | | 1/2 bias LCD, $V_{DD}$ = 5.0V | 2.12 | 2.50 | 2.88 | V |
| LCD bias output 2 | $V_{L2}$ | 1/42 bias LCD, $V_{DD}$ = 5.0V | 2.12 | 2.50 | 2.88 | V |
| | | 1/32 bias LCD, $V_{DD}$ = 5.0V | 1.42 | 1.67 | 1.92 | V |
| LCD bias output 3 | $V_{L3}$ | 1/4 bias LCD, $V_{DD}$ = 5.0V | 1.06 | 1.25 | 1.44 | V |

# 4. Central Processing Unit (CPU)

## 4.1 General Concept

MQ6812/MQ6821 adopts i87 8-bit MCU core with embedded 16KB program flash memory, and 128 x8 bit data flash. The introduction of the powerful central processing unit (CPU) can be divided into two major parts: (1) Addressing Space of Program / Data Memory and Special Function Registers (SFR), (2) Operation Modes.

## 4.2 Addressing Space

Figure 4.1 shows the addressing space of MQ6812/MQ6821, including SRF1, SRF2, SFR3, RAM and program memory (Flash) memory. Except main 16KB program flash memory, and 128 x8 bit data flash, MQ6812/MQ6821 also provide 64x8 bit info. block(0x7E40 至 0x7E7F).Info block can save important product information, e.g. information of writer.

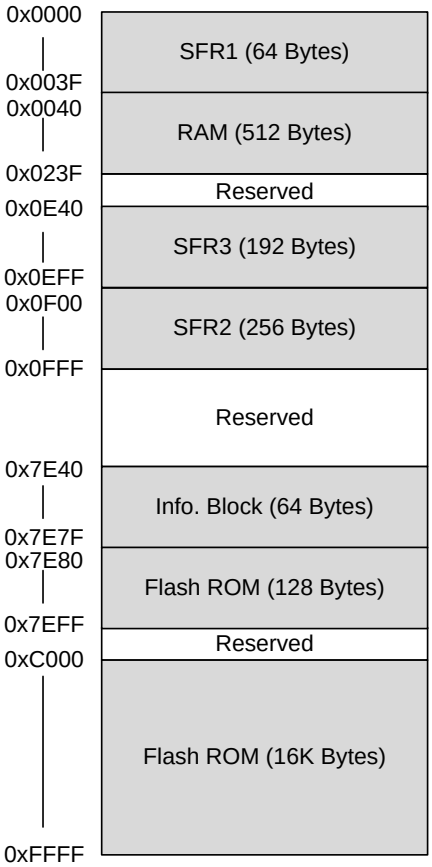



Figure 4.1 Addressing Map of MQ6812/MQ6821

### 4.2.1 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 16K + 128 bytes format which is addressed by the PC and table pointer. The Flash ranges of MQ6812/MQ6821 is from 0xC000 to 0xFFFF (16K bytes) The data flash ranges of MQ6812/MQ6821 is from 0x7E80 to 0x7EFF (128x8) .The 128 x 8 byte area can use for saving user information, e.g. product ID.。

### 4.2.2 RAM Data Memory- RAM

The RAM is mapped to 0x0040 to 0x023F(512x 8 byte) in the data area after reset release. Note that 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.

### 4.2.3 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.

*note: do not read SFR which is system reserved.*

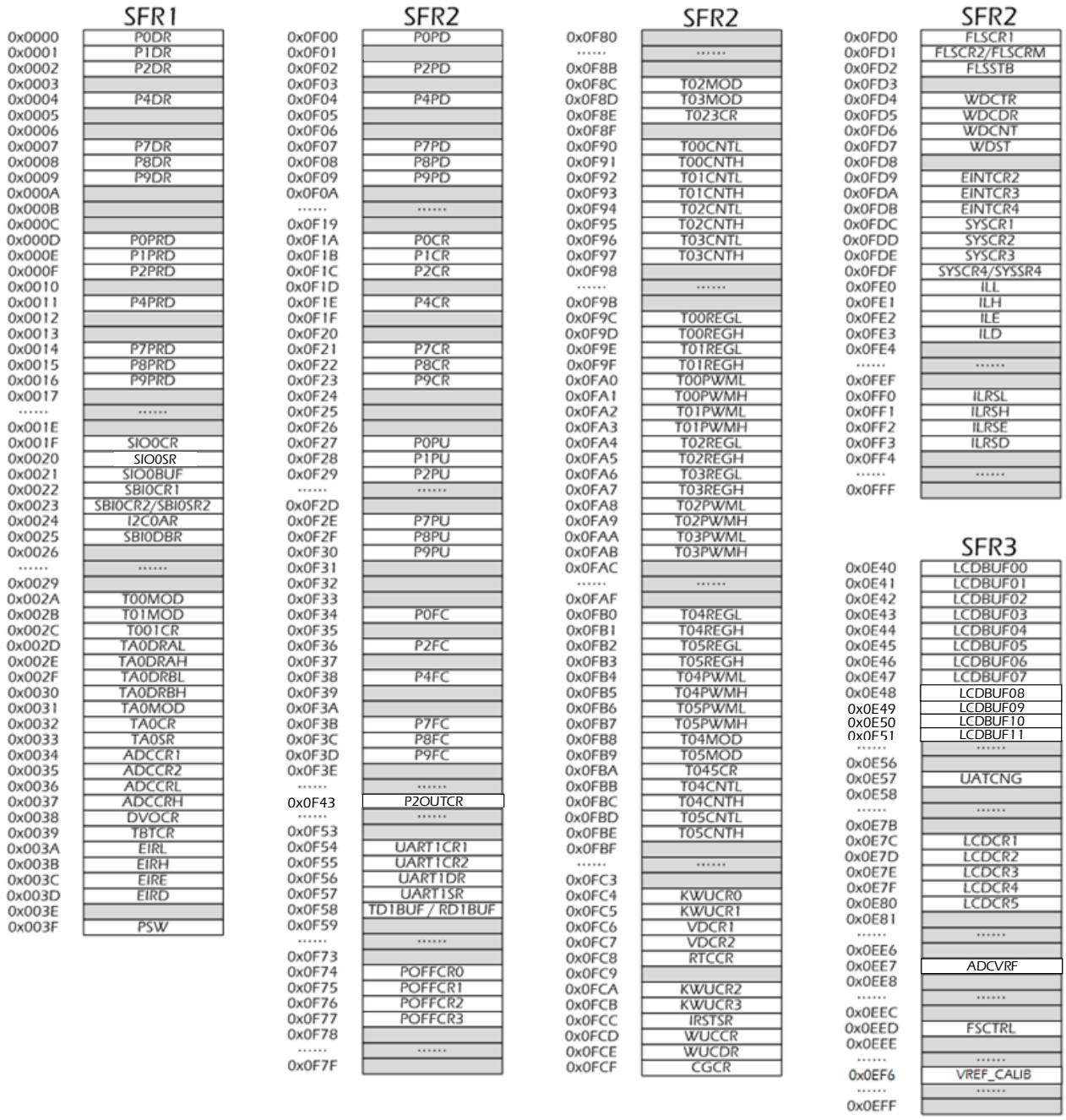



Figure 4.2 SFR1 · SFR2 · SFR3

## 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).

Figure 4.3 shows the operating mode transition diagram.

#### 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 (fc).

(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), 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.

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 this 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 this 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 this 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.

(d) IDLE2 mode

In this 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 this 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 this 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 this 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

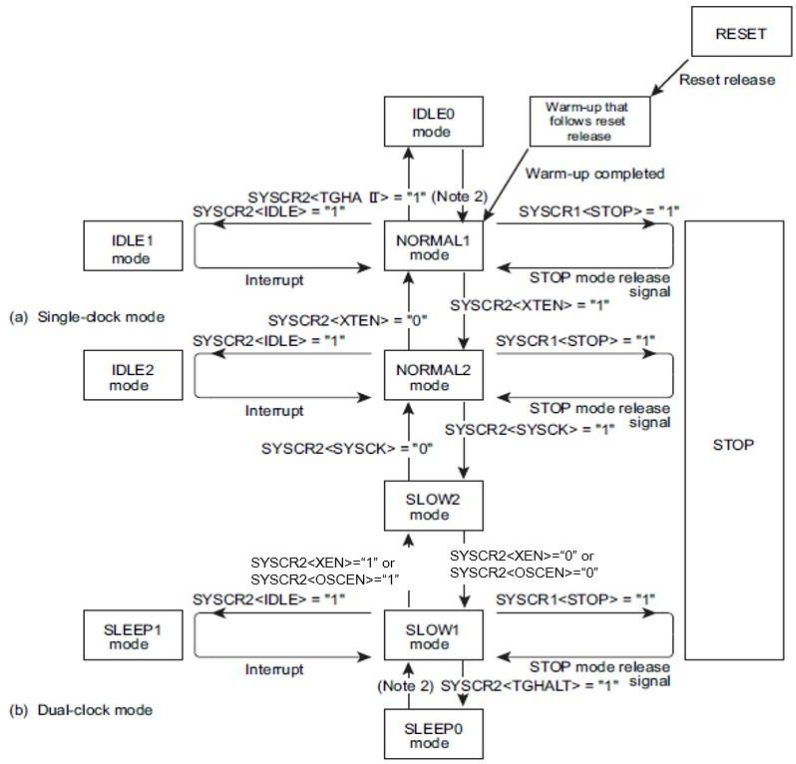



Figure 4.3 Operation Mode Transition Diagram

| Operation mode | | Oscillation circuit | | *CPU* | WDT | TBT | Other peripheral ciruits | Machine cycle time |
|---|---|---|---|---|---|---|---|---|
| | | High frequency | Low frequency | | | | | |
| Single clock | **RESET** | Oscillation | Stop | Reset | Reset | Reset | Reset | 1/fcgck (sec) |
| | **Normal1** | | | Operate | Operate | Operate | Operate | |
| | **IDLE1** | | | Stop | Stop | | | |
| | **IDLE0** | | | | | | Stop | |
| | **STOP** | Stop | | | | Stop | | |
| Dual clock | **Normal2** | Oscillation | Oscillation | Operate with the high frequency | Operate with the high/low frequency | Operate | Operate | 1/fcgck (sec) |
| | **IDLE2** | | | Stop | Stop | | | |
| | **SLOW2** | | | Operate with the low frequency | Operate with the low frequency | | | 4/fs (sec) |
| | **SLOW1** | Stop | | Operate with the low frequency | Operate with the low frequency | | | |
| | **SLEEP1** | | | Stop | Stop | | | |
| | **SLEEP0** | | | | | | Stop | |
| | **STOP** | | Stop | | | Stop | | |

Table 4.1 Operation Modes and Conditions

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

To release the STOP mode by using the STOPB pin, set VDCR2 <VDSS> to "00" or"10". (For details of VDCR2, refer to "5.3 Voltage Detection Circuits")

The STOP mode released by the STOPB 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 (external interrupt input 5) 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.

◆ 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.

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". For release by the key-on wakeup, refer to "3.7 Key-on Wakeup" in iMQ i87 User Manual.

*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 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.<br>The low-frequency clock oscillation circuit stops oscillation. |
| Dual-clock mode | NORMAL2 | High-frequency clock oscillation circuit | Low-frequency clock oscillation cir-cuit | The high-frequency clock oscillation circuit starts oscillation.<br>The low-frequency clock oscillation circuit starts oscillation. |
| | SLOW1 | - | Low-frequency clock oscillation cir-cuit | The high-frequency clock oscillation circuit stops oscillation.<br>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

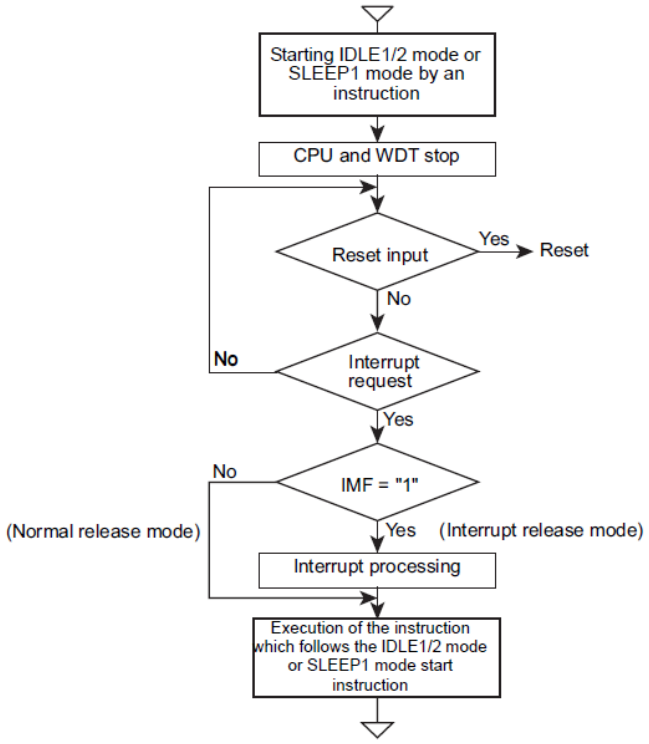



Figure 4.4 IDLE1/2 and SLEEP1 Modes

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*

*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

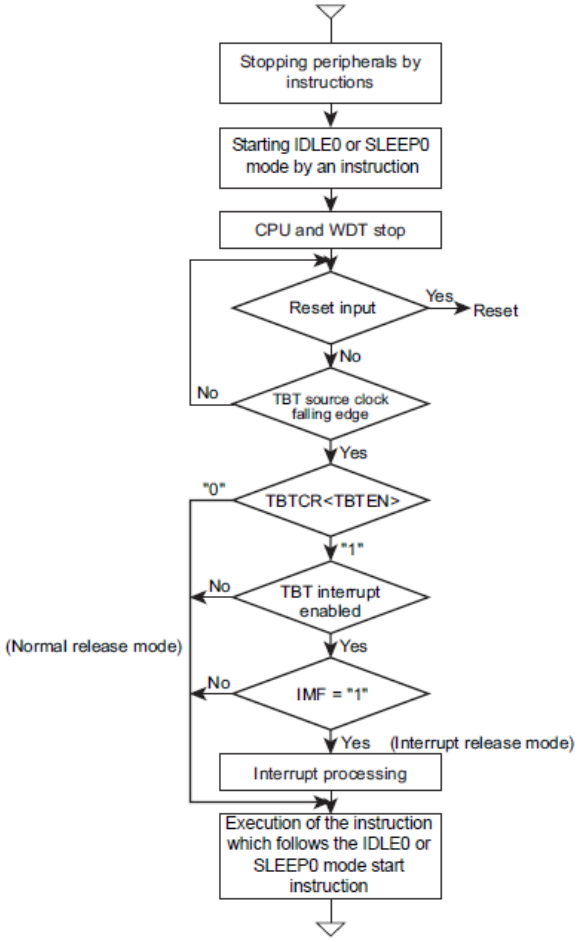



Figure 4.5 IDLE0 and SLEEP0 Modes

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".

(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.

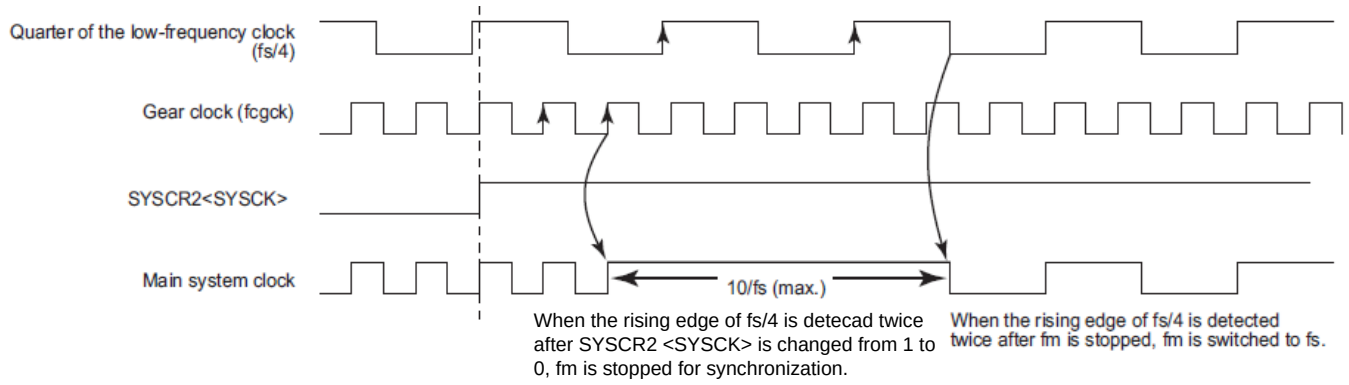



Figure 4.6 Switching of the Main system clock (fm)
(Switching from fcgck to fs/4)

(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.

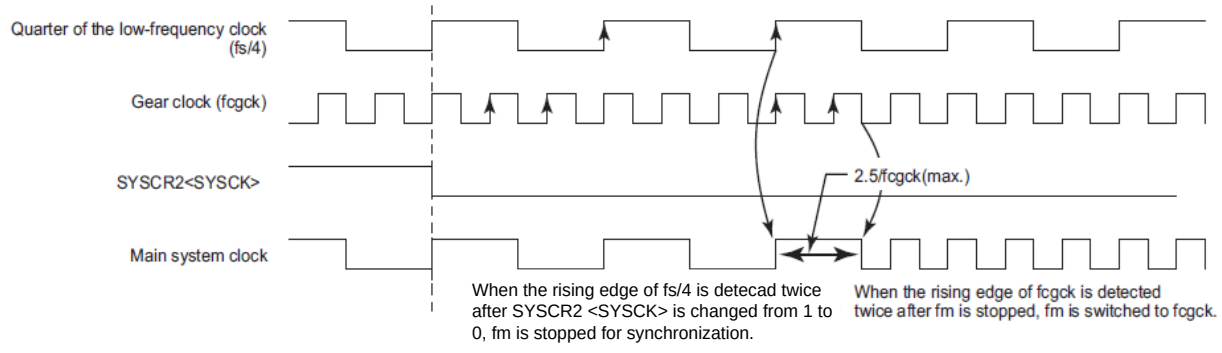



Figure 4.7 Switching of the Main system clock (fm)
(Switching from fs/4 to fcgck)

*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.*

## 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.

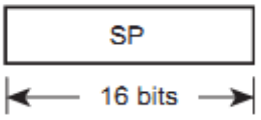



Figure 4.8 Stack Pointer

Figure 4.9 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.

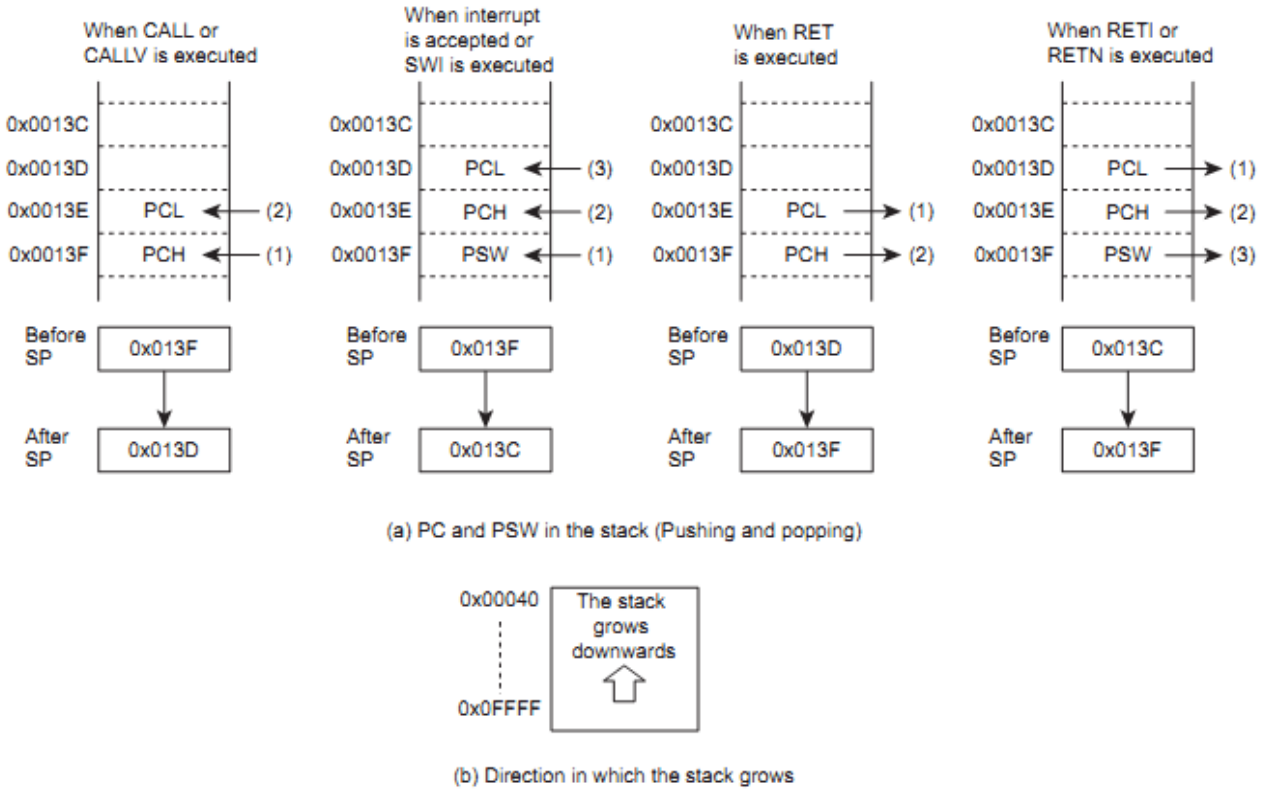


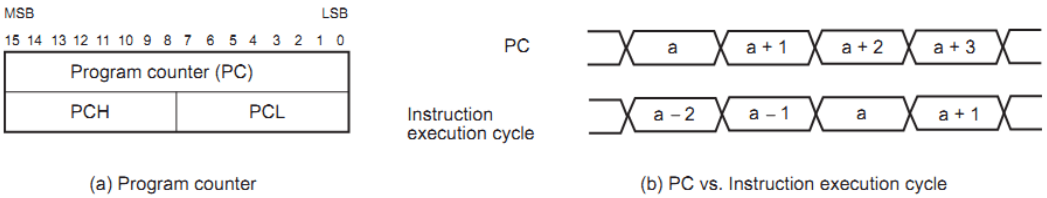

Figure 4.9 Stack

## 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.




(a) Program counter (b) PC vs. Instruction execution cycle

Figure 4.10 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 MQ6812/MQ6821).

## 4.6 General-Purpose Register

MQ6812/MQ6821 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.

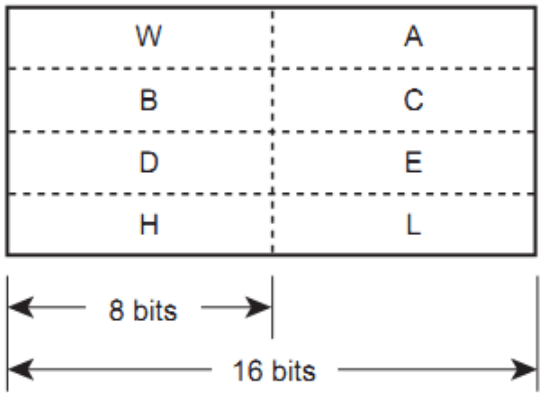



Figure 4.11 General-Purpose Registers

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)

MQ6812/MQ6821 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.

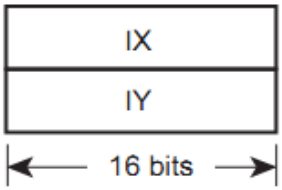



Figure 4.12 16-Bit General-Purpose Registers

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

## 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 PSWclears 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

MQ6812/MQ6821 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)

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 | - | TC045EN | TC023EN | TC001EN | - | - | - | TCA0EN |
| Read/Write | R | R/W | R/W | R/W | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| TC023EN | TC04, 05 enable control | 0: Disable<br>1: Enable |
| TC023EN | TC02, 03 enable control | 0: Disable<br>1: Enable |
| TC001EN | TC00, 01 enable control | 0: Disable<br>1: Enable |
| TCA0EN | TCA0 enable control | 0: Disable<br>1: Enable |

Low Power Consumption Register 1

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

| | | |
|---|---|---|
| SBI0EN | I2C0 control | 0: Disable<br>1: Enable |
| UART1EN | UART1 control | 0: Disable<br>1: Enable |

Low Power Consumption Register 2

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

| | | |
|---|---|---|
| LCDEN | LCD enable control | 0: Disable<br>1: Enable |
| RTCEN | RTC enable control | 0: Disable<br>1: Enable |
| SIO0EN | SIO0 control | 0: Disable<br>1: Enable |

Low Power Consumption Register 3

| POFFCR3 (0x0F77) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | INT4EN | INT3EN | INT2EN | - | - |
| 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 |

| | | |
|---|---|---|
| INT4EN | INT4 Control | 0: Disable<br>1: Enable |
| INT3EN | INT3 Control | 0: Disable<br>1: Enable |
| INT2EN | INT2 Control | 0: Disable<br>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

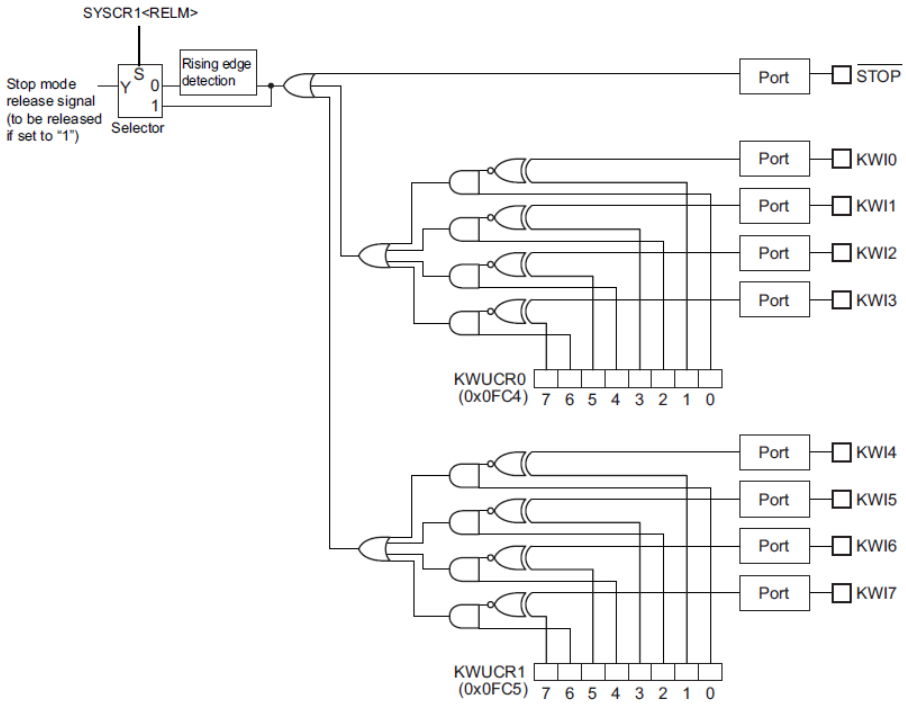



Figure 4.13 Key-on Wakeup Circuit (KWI0 ~ KWI7 for example)

### 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<br>1: High level |
| KW3EN | Input enable / disable control of KWI3 pin | 0: Disable<br>1: Enable |
| KW2LE | STOP mode release level of KWI2 pin | 0: Low level<br>1: High level |
| KW2EN | Input enable / disable control of KWI2 pin | 0: Disable<br>1: Enable |
| KW1LE | STOP mode release level of KWI1 pin | 0: Low level<br>1: High level |
| KW1EN | Input enable / disable control of KWI1 pin | 0: Disable<br>1: Enable |
| KW0LE | STOP mode release level of KWI0 pin | 0: Low level<br>1: High level |
| KW0EN | Input enable / disable control of KWI0 pin | 0: Disable<br>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<br>1: High level |
| KW7EN | Input enable / disable control of KWI7 pin | 0: Disable<br>1: Enable |
| KW6LE | STOP mode release level of KWI6 pin | 0: Low level<br>1: High level |
| KW6EN | Input enable / disable control of KWI6 pin | 0: Disable<br>1: Enable |
| KW5LE | STOP mode release level of KWI5 pin | 0: Low level<br>1: High level |
| KW5EN | Input enable / disable control of KWI5 pin | 0: Disable<br>1: Enable |

| | | |
|---|---|---|
| KW4LE | STOP mode release level of KWI4 pin | 0: Low level<br>1: High level |
| KW4EN | Input enable / disable control of KWI4 pin | 0: Disable<br>1: Enable |

Key-on Wakeup Control Register 2

| KWUCR0 (0x0FCA) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | KW11LE | KW11EN | KW10LE | KW10EN | KW9LE | KW9EN | KW8LE | KW8EN |
| 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 |

| | | |
|---|---|---|
| KW11LE | STOP mode release level of KWI11 pin | 0: Low level<br>1: High level |
| KW11EN | Input enable / disable control of KWI11 pin | 0: Disable<br>1: Enable |
| KW10LE | STOP mode release level of KWI10 pin | 0: Low level<br>1: High level |
| KW10EN | Input enable / disable control of KWI10 pin | 0: Disable<br>1: Enable |
| KW9LE | STOP mode release level of KWI9 pin | 0: Low level<br>1: High level |
| KW9EN | Input enable / disable control of KWI9 pin | 0: Disable<br>1: Enable |
| KW8LE | STOP mode release level of KWI8 pin | 0: Low level<br>1: High level |
| KW8EN | Input enable / disable control of KWI8 pin | 0: Disable<br>1: Enable |

Key-on Wakeup Control Register 3

| KWUCR1 (0x0FCB) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | KW15LE | KW15EN | KW14LE | KW14EN | KW13LE | KW13EN | KW12LE | KW12EN |
| 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 |

| | | |
|---|---|---|
| KW15LE | STOP mode release level of KWI15 pin | 0: Low level<br>1: High level |
| KW15EN | Input enable / disable control of KWI15 pin | 0: Disable<br>1: Enable |
| KW14LE | STOP mode release level of KWI14 pin | 0: Low level<br>1: High level |
| KW14EN | Input enable / disable control of KWI14 pin | 0: Disable<br>1: Enable |
| KW13LE | STOP mode release level of KWI13 pin | 0: Low level<br>1: High level |
| KW13EN | Input enable / disable control of KWI13 pin | 0: Disable<br>1: Enable |

| | | |
|---|---|---|
| KW12LE | STOP mode release level of KWI12 pin | 0: Low level<br>1: High level |
| KW12EN | Input enable / disable control of KWI12 pin | 0: Disable<br>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 15). 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 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.*

# 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 (RESETB,external factor)
2. Power-on reset (POR, internal factor)
3. Voltage detection reset1 (LVD1, internal factor)
4. Voltage detection reset2 (LVD2, internal factor)
5. Watchdog timer reset (WDT ,internal factor)
6. Watchdog timer reset2 (WDT2,internal factor)
7. System clock reset (internal factor)
8. Flash standby reset (internal factor)

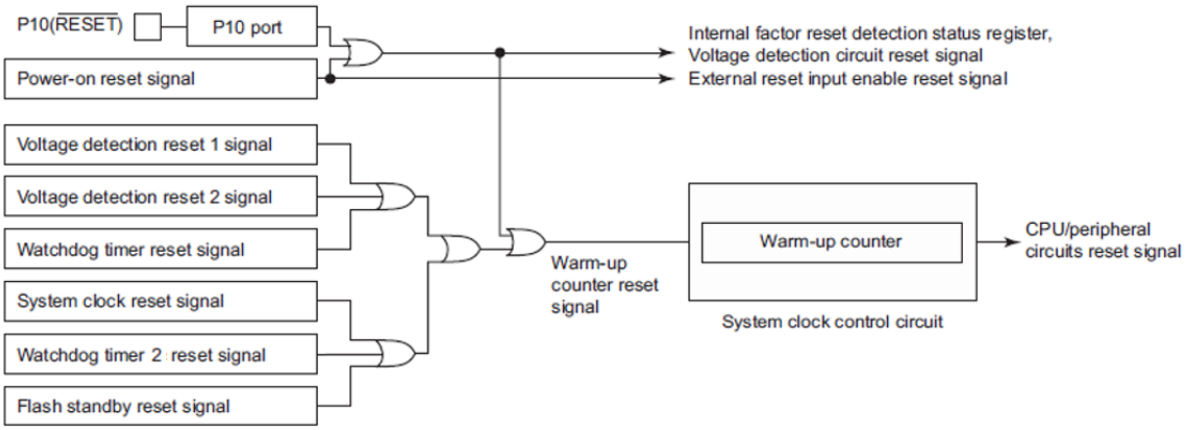



Figure 5.1 Reset Control Circuit

### 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 | - | - | - | - | - | - | - | RSTDIS |
| Read/Write | R | R | R | R | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| RSTDIS | External reset input enable register | 0: Enable the external reset input<br>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".Bit2 to Bit 1 of SYSCR3 are system reseved, do not change the default value.*

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:<br>0x71:<br>Others: | Enable the contents of SYSCR3 <RSTDIS><br>Enable the contents of IRSTSR <FCLR><br>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 | - | - | - | - | - | - | - | RSTDISS |
| Read/Write | R | R | R | R | R | R | R | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| RSTDISS | Status of external reset input enable register | 0: The enabled SYSCR3<RSTDIS> data is "0".<br>1: The disabled SYSCR3< RSTDIS > 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". Bit2 to Bit 1 of SYSCR3 are system reseved.*

Internal Factor Reset Detection Status Register

| IRSTSR (0x0FCC) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | FCLR WDT2RF | FLSRF | - | - | LVD2RF | LVD1RF | SYSRF | WDTRF |
| Read/Write | W | R | - | - | R | R | R | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

*Note 1): IRSTSR is initialized by a power-on 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): IRSTSR <FCLR> is initialized by a power-on reset, or an internal reset factor.*

*Note 4): 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 5): 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 6):Bit 7 of IRSTSR write as FCLR(Flag initialization control) function; Bit 7 of IRSTSR read as WDR2RF(Watch Dog 2 reset flag).*

*Note 7):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 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.

| 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.*

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.

### 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.

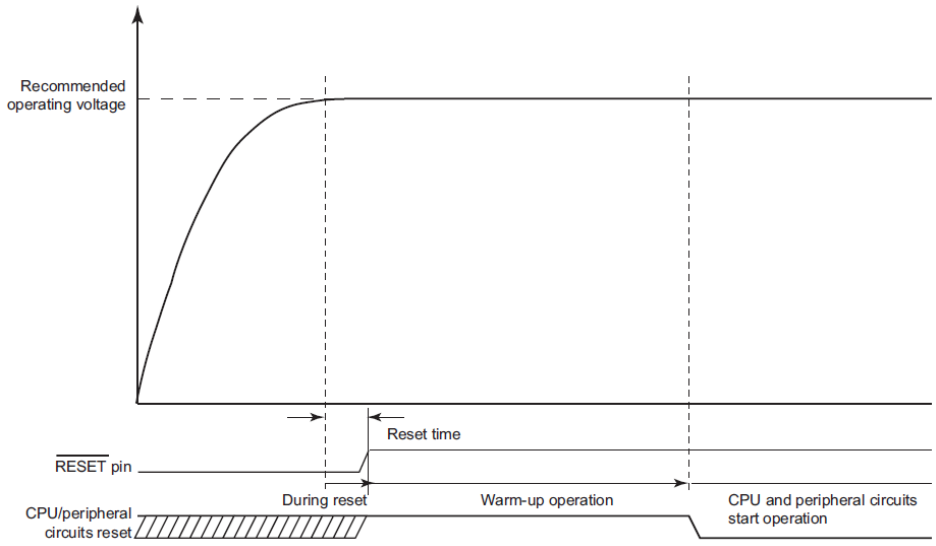



Figure 5.2 External Reset Input (During Power-Up)

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.

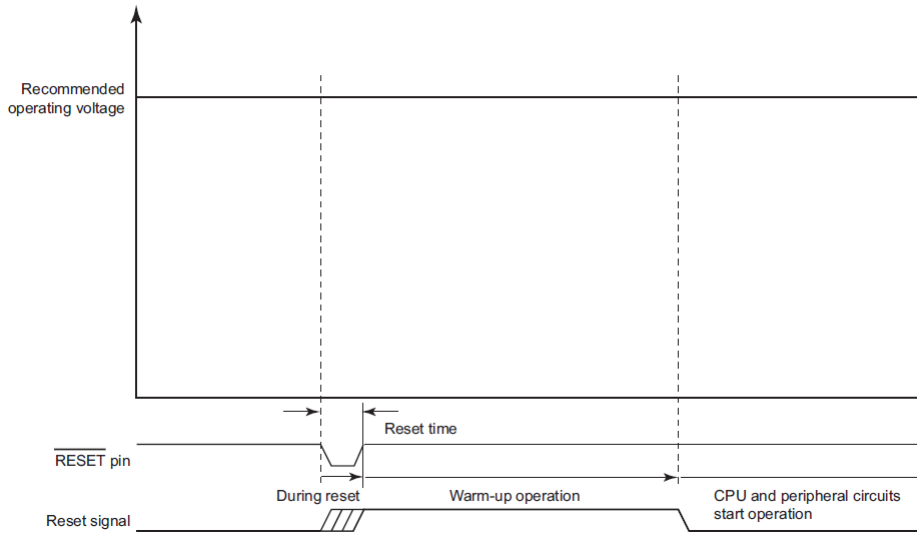



Figure 5.3 External Reset Input (when the power is stabilized)

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. Refer to "5.2 Power-on Reset circuit".

#### 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/ Watchdog Timer 2"

#### 5.1.4.5 Watchdog Timer2 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/ Watchdog Timer 2"

#### 5.1.4.6 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".

#### 5.1.4.7 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".

#### 5.1.4.8 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.*

#### 5.1.4.9 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.

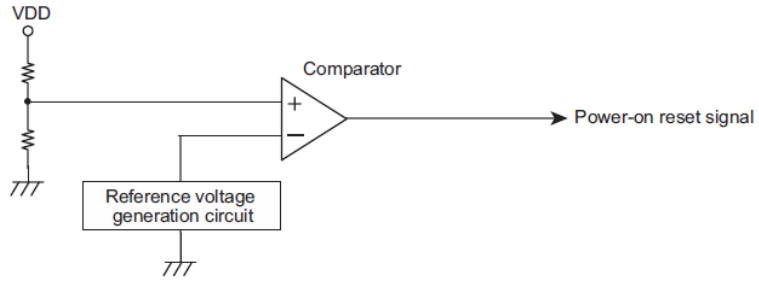



Figure 5.4 Power-on Reset Circuit

### 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.

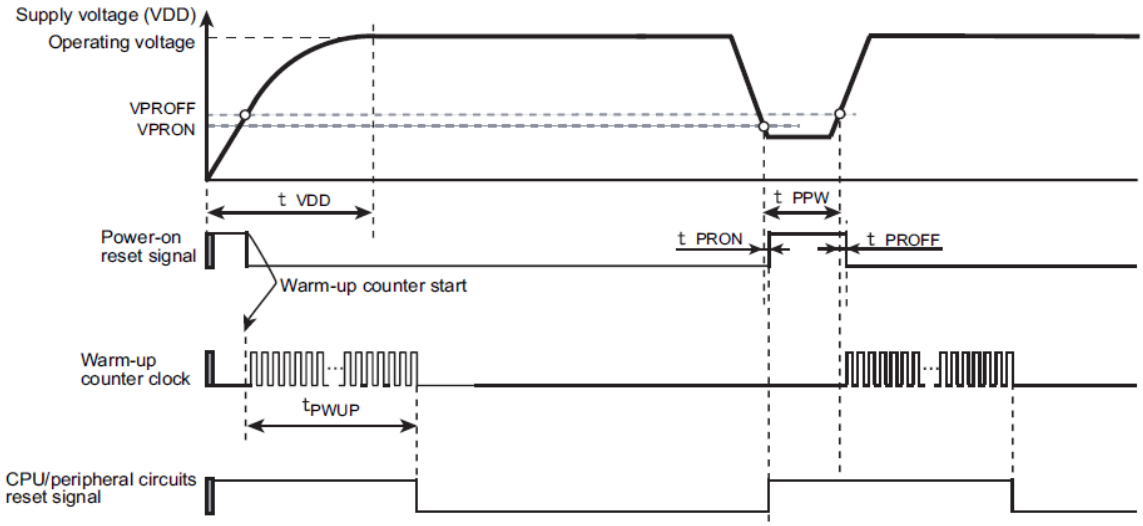



Figure 5.5 Operation Timing of Power-on Reset

*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.*

## 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.*

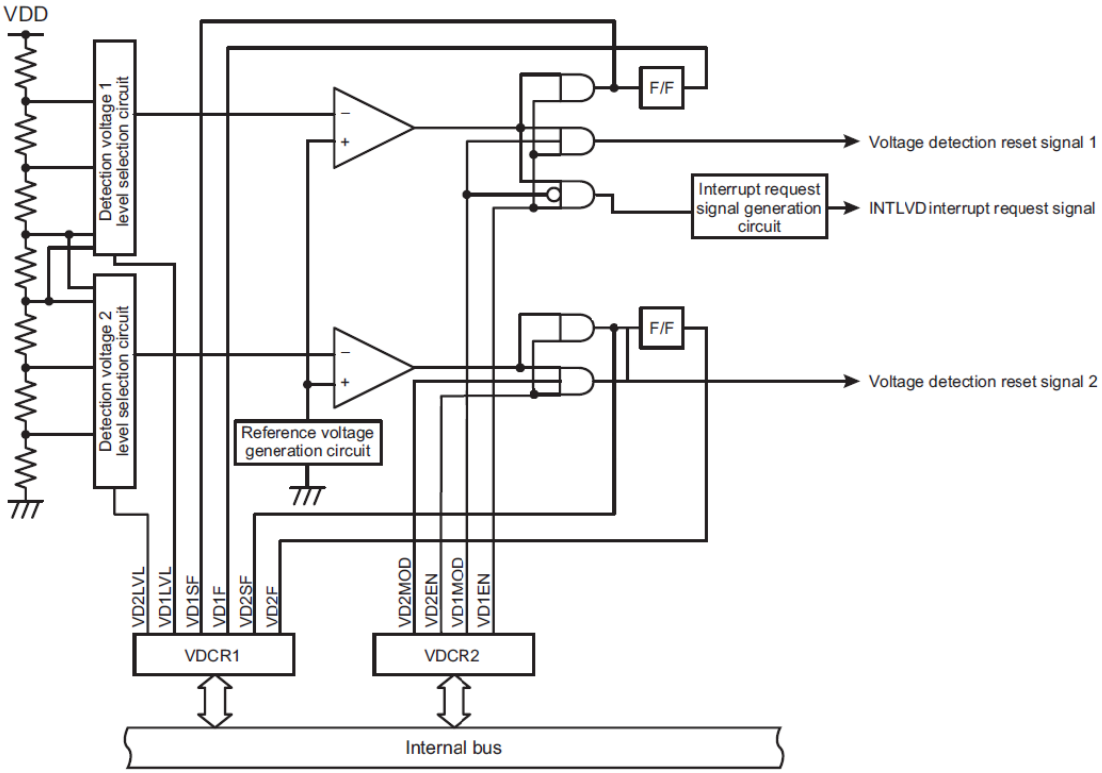



Figure 5.6 Voltage Detection Circuit

### 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 | 1 | 0 | 0 | 0 | 0 | 0 |

| | | | | |
|---|---|---|---|---|
| VD2F | Voltage detection 2 flag (Retain the state when VDD <VD2LVL is detected) | | Read | Writer |
| | | 0<br>1 | 0: VDD ≥ VD2LVL<br>1: VDD < VD2LVL | Clears VD2F to "0"<br>- |
| VD2SF | Voltage detection 2 status flag (Magnitude relation of VDD and VD2LVL when they are read) | 0<br>1 | 0: VDD ≥ VD2LVL<br>1: VDD < VD2LVL | |
| VD2LVL | Selection for detection voltage 2 | 00<br>01<br>10<br>11 | 2.85V +/- 0.15 V<br>2.65V +/- 0.15 V<br>2.35V +/- 0.10 V<br>2.00V +/- 0.10 V | |
| VD1F | Voltage detection 1 flag (Retain the state when VDD <VD1LVL is detected) | | Read | Writer |
| | | 0<br>1 | 0: VDD ≥ VD1LVL<br>1: VDD < VD1LVL | Clears VD1F to "0"<br>- |
| VD1SF | Voltage detection 1 status flag (Magnitude relation of VDD and VD1LVL when they are read) | 0<br>1 | 0: VDD ≥ VD1LVL<br>1: VDD < VD1LVL | |
| VD1LVL | Selection for detection voltage 1 | 00<br>01<br>10<br>11 | 4.50V +/- 0.20 V<br>4.20V +/- 0.20 V<br>3.70V +/- 0.20 V<br>3.15V +/- 0.20 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.*

**Voltage Detection Control Register 2**

| VDCR2 (0x0FC7) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | VD2MOD | VD2EN | VD1MOD | VD1EN |
| Read/Write | R | R | R | R | 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<br>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<br>1: Enable the operation of voltage detection 2 |
| VD1MOD | Select the operation mode of voltage detection 1 | 0: Generate a INTLVD interrupt request signal<br>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<br>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".*

### 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).

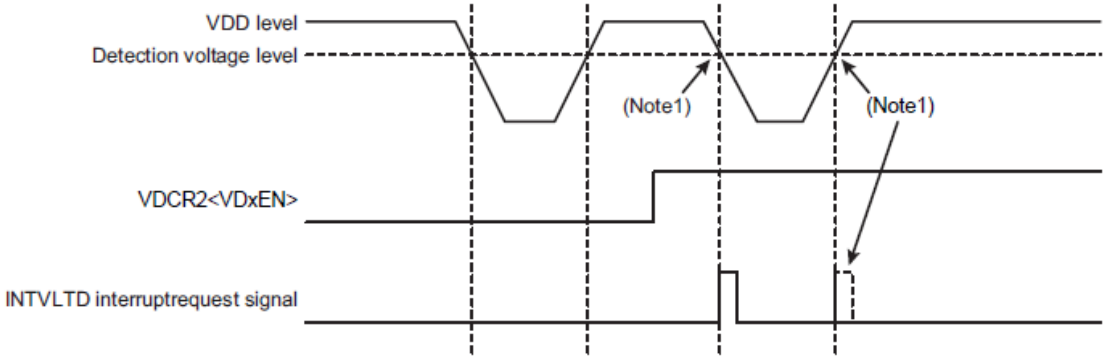



Figure 5.7 Voltage Detection Interrupt Request

**(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).

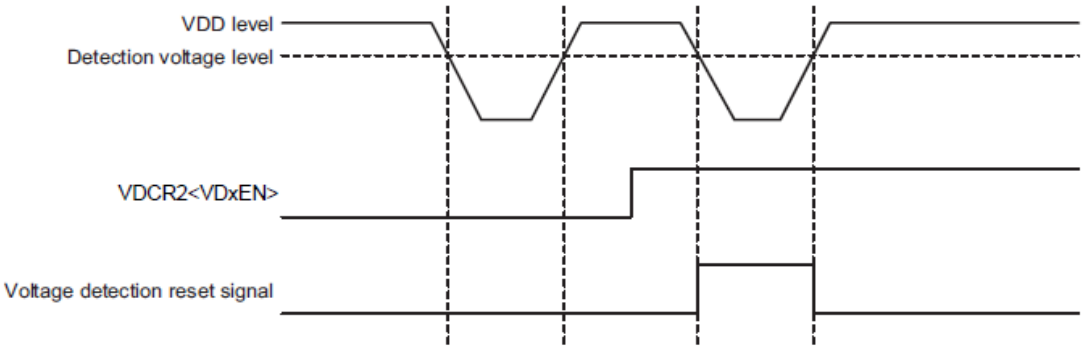



Figure 5.8 Voltage Detection Reset Signal

### 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].*

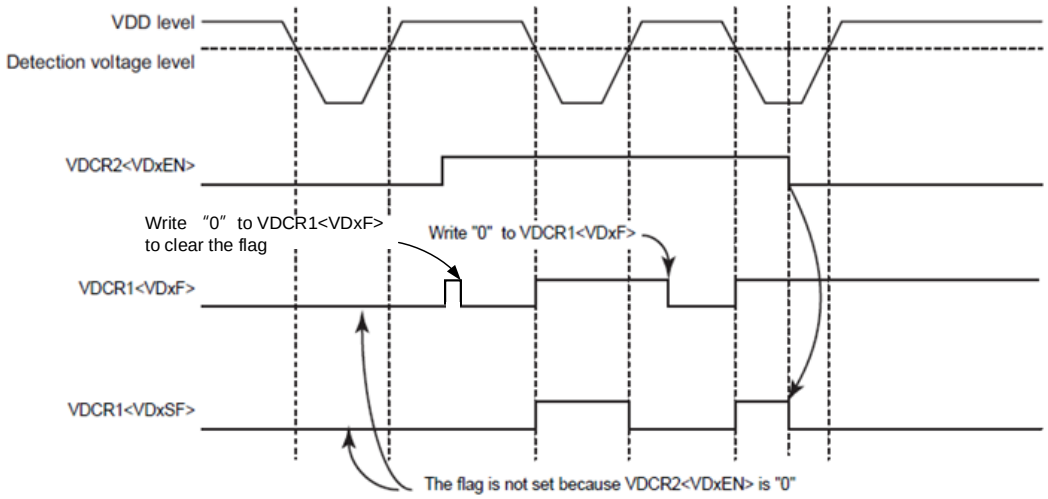



Figure 5.9 Changes in the Voltage Detection Flag and the Voltage Detection Status Flag

## 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.*

#### 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:

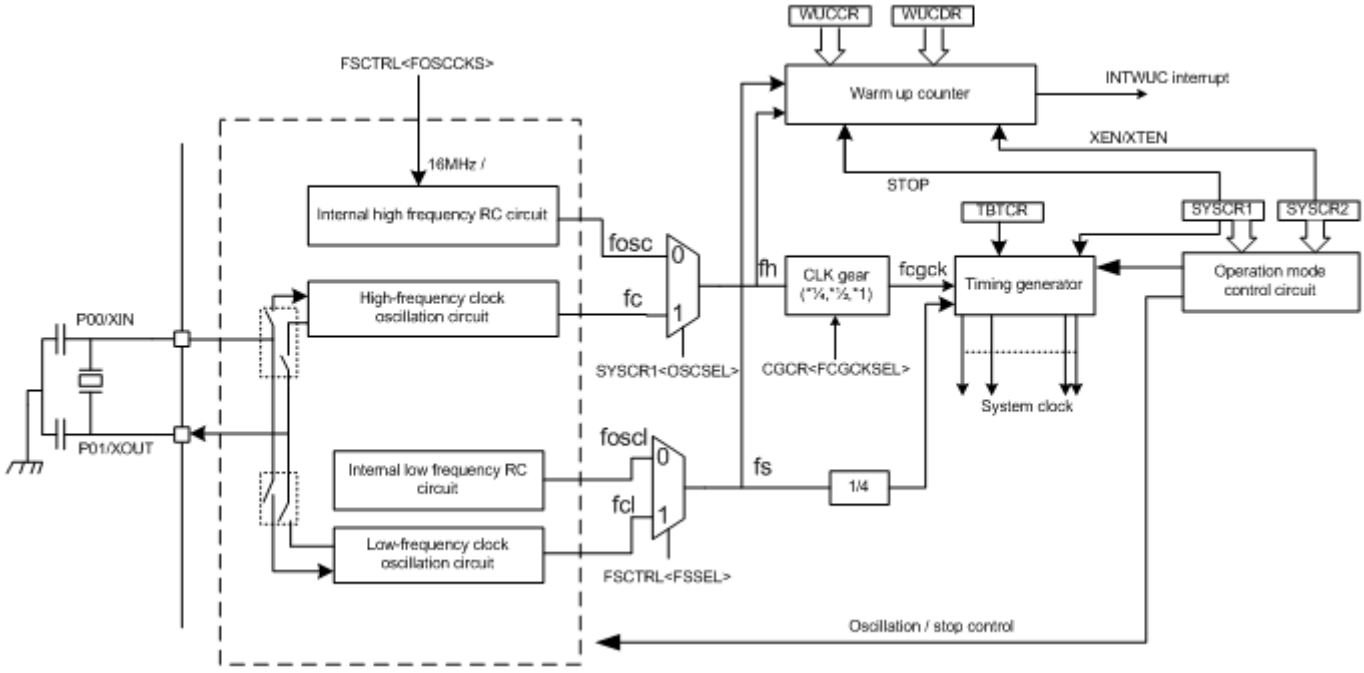
1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Clear VDCR2 <VDxMOD> to "0" to set the operation mode to generate INTLVD interrupt request signals.
3. Clear VDCR2 <VDxEN> to "0" to disable the voltage detection operation.

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

# 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): 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<br>1 | Operate the CPU and the peripheral circuit<br>Stop the CPU and the peripheral circuit (activate the STOP mode) |
| RELM | Select the STOP mode release method | 0<br><br>1 | Edge-sensitive release mode (Release the STOP mode at the rising edge of the STOP mode release signal)<br>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<br>1 | High impedance<br>Output hold |
| DV9CK | Select the input clock to stage 9 of the divider | 0<br>1 | fcgck/$2^9$<br>fs/4 |
| OSCSEL | Select the high-frequency reference clock (fh) | 0<br>1 | Internal high-frequency clock (fosc)<br>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): 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 5): Don't set SYSCK1 <DV9CK> to "1" before the oscillation of the external low-frequency clock oscillation circuit becomes stable.*

*Note 6): 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 7): 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<br>1 | Disable internal high-frequency clock oscillation circuit<br>Enable internal high-frequency clock oscillation circuit |
| XEN | Control the external high-freq. clock (fc) | 0<br>1 | Stop oscillation<br>Continue or start oscillation |
| XTEN | Control the external low-freq. clock (fs) | 0<br>1 | Stop oscillation<br>Continue or start oscillation |
| SYSCK | Select a system clock | 0<br>1 | Gear clock (fcgck) (NORMAL1/2 or IDLE1/2 mode)<br>Low-frequency clock (fs/4) (SLOW1/2 or SLEEP1 mode) |
| IDLE | CPU and WDT control (IDLE1/2 or SLEEP1 mode) | 0<br>1 | Operate the CPU and the WDT<br>Stop the CPU and the WDT (Activate IDLE1/2 or SLEEP1 mode) |
| TGHALT | TG control (IDLE0 or SLEEP0 mode) | 0<br>1 | Enable the clock supply from the TG to all the peripheral circuits<br>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):To Switchthe frequency, first set the new frequency, then turn off the original frequency.*

Internal Clock Control Register 1

| FSCTRL (0x0EED) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | XTAL_SEL | P81_SDR | P80_SDR | PACK_SEL | FOSCCKS | | FSSEL |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W |
| After reset | 0 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 | 1 | 0 |

| | | |
|---|---|---|
| XTAL_SEL<br>P81_SDR<br>P80_SDR<br>PACK_SEL | Chip modifying when recording. | Do not change the value.(Note 2) |
| FOSCCKS | Choose internal high-frequency clock (fosc) | 00: System reserved<br>01: choose internal high-frequency clock 16MHz (default)<br>10: System reserved<br>11: System reserved |
| FSSEL | Choose internal/external-low frequency clock | 0: Choose internal low-frequency clock<br>1: Choose external low-frequency clock |

*Note 1) : FOSCCKS(fosc) internal high frequency(fosc) default is "01"(16MHz).*

*Note2) : Bit6~Bit3 is related to chip modifying when record the chip by writer. Please user use "bit operation" for programming. "SET (FSCTRL).0」" or "CLR (FSCTRL).0" for example. Do not use other bit to change the value of Bit 6~Bit3.*

**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<br>1 | -<br>Clear and stop the counter |
| WUCDIV | Select the frequency division of the warm-up counter source clock | 00<br>01<br>10<br>11 | Sourceclock<br>Sourceclock/2<br>Sourceclock/$2^2$<br>Sourceclock/$2^3$ |
| WUCSEL | Select the warm-up counter source clock | 00<br>01<br>10<br>11 | Select the internal high-frequency clock (fosc)<br>Select the external high-frequency clock (fc)<br>Select the external low-frequency clock (fs)<br>Reserved |

*Note 1): fosc: Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], , foscl: internal low-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<br>01<br>10<br>11 | fcgck = fc / 4<br>fcgck = fc / 2<br>fcgck = fc<br>fcgck = fc / 8 |

*Note 1):fc: 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".*

## 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".

The high-frequency (fc) clock and the low-frequency (fs) clock can easily be obtained by connecting an oscillator between the XIN and XOUT pins.

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.

| 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".*

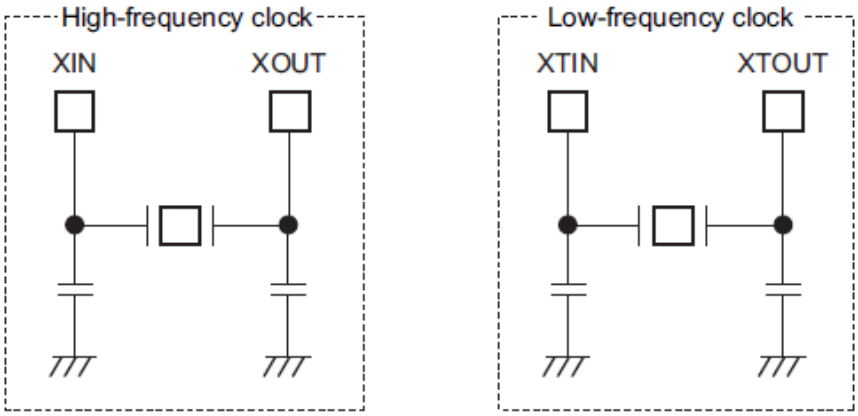



Figure 6.2Examples of Oscillator Connection

*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*

### 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 6.2Gear Clock (fcgck)

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).

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

### 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

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

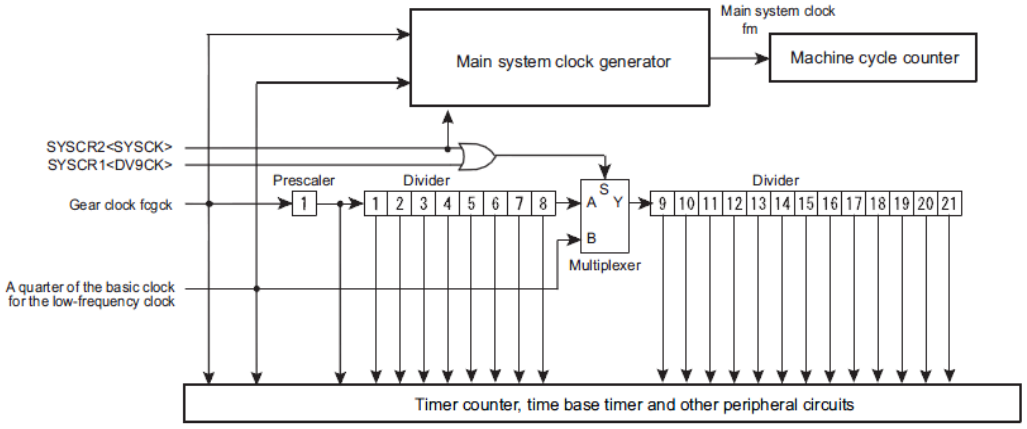



Figure 6.3Configuration of Timing Generator

#### 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).

#### 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.

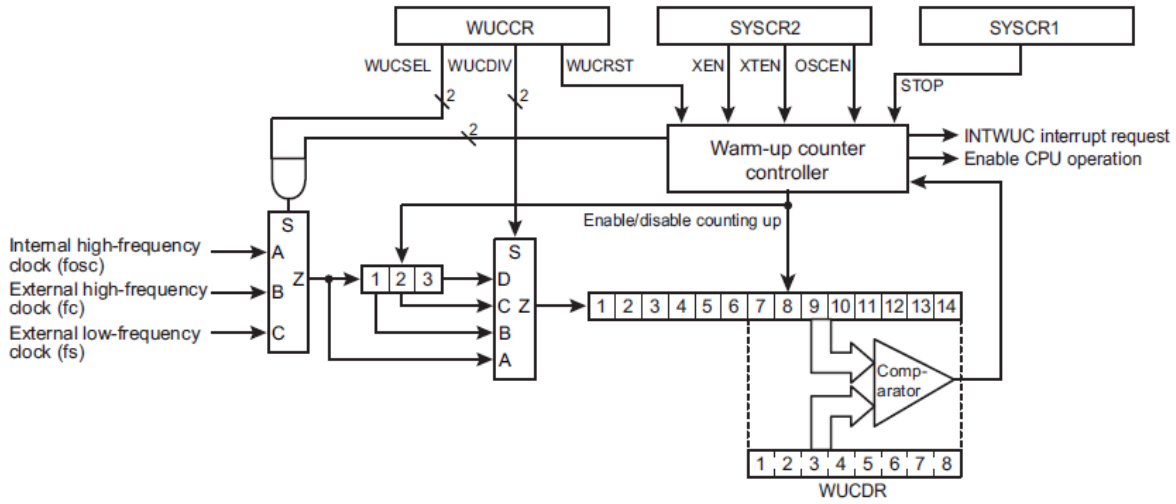



Figure 6.4Warm-up Counter Circuit

## 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 $102\times2^9/fc$ [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> | WUCCD <WUCDIV> | Cpunter 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 × $2^7$/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

MQ6812/MQ6821 has a total of 18 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 |
|---|---|---|---|---|---|
| Internal/External | (Reset) | Non-maskable | - | 0xFFFE | 1 |
| Internal | INTSWI | Non-maskable | - | 0xFFFC | 2 |
| Internal | INTUNDEF | Non-maskable | - | 0xFFFC | 2 |
| Internal | INTWDT | Non-maskable | ILL<IL3> | 0xFFF8 | 2 |
| Internal | INTWUC | IMF AND EIRL<EF4> = 1 | ILL<IL4> | 0xFFF6 | 5 |
| Internal | INTTBT | IMF AND EIRL<EF5> = 1 | ILL<IL5> | 0xFFF4 | 6 |
| Internal | INTSIO0 | IMF AND EIRL<EF6> = 1 | ILL<IL6> | 0xFFF2 | 7 |
| Internal | INTVLTD | IMF AND EIRH<EF9> = 1 | ILH<IL9> | 0xFFEC | 8 |
| Internal | INTADC | IMF AND EIRH<EF10> = 1 | ILH<IL10> | 0xFFEA | 9 |
| Internal | INTRTC | IMF AND EIRH<EF11> = 1 | ILH<IL11> | 0xFFE8 | 10 |
| Internal | INTTC00 | IMF AND EIRH<EF12> = 1 | ILH<IL12> | 0xFFE6 | 11 |
| Internal | INTTC01 | IMF AND EIRH<EF13> = 1 | ILH<IL13> | 0xFFE4 | 12 |
| Internal | INTTCA0 | IMF AND EIRH<EF14> = 1 | ILH<IL14> | 0xFFE2 | 13 |
| Internal | INTSBI0 | IMF AND EIRH<EF15> = 1 | ILH<IL15> | 0xFFE0 | 14 |
| External | INT2 | IMF AND EIRE<EF18> = 1 | ILE<IL18> | 0xFFDA | 15 |
| External | INT3 | IMF AND EIRE<EF19> = 1 | ILE<IL19> | 0xFFD8 | 16 |
| External | INT4 | IMF AND EIRE<EF20> = 1 | ILE<IL20> | 0xFFD6 | 17 |
| Internal | INTRXD1 | IMF AND EIRE<EF22> = 1 | ILE<IL22> | 0xFFD2 | 18 |
| Internal | INTTXD1 | IMF AND EIRE<EF23> = 1 | ILE<IL23> | 0xFFD0 | 19 |
| Internal | INTTC02 | IMF AND EIRD<EF24> = 1 | ILD<IL24> | 0xFFCE | 20 |
| Internal | INTTC03 | IMF AND EIRD<EF25> = 1 | ILD<IL25> | 0xFFCC | 21 |
| Internal | INTTC04 | IMF AND EIRD<EF28> = 1 | ILD<IL28> | 0xFFC6 | 22 |
| Internal | INTTC05 | IMF AND EIRD<EF29> = 1 | ILD<IL29> | 0xFFC4 | 23 |

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/ Watchdog Timer 2".*

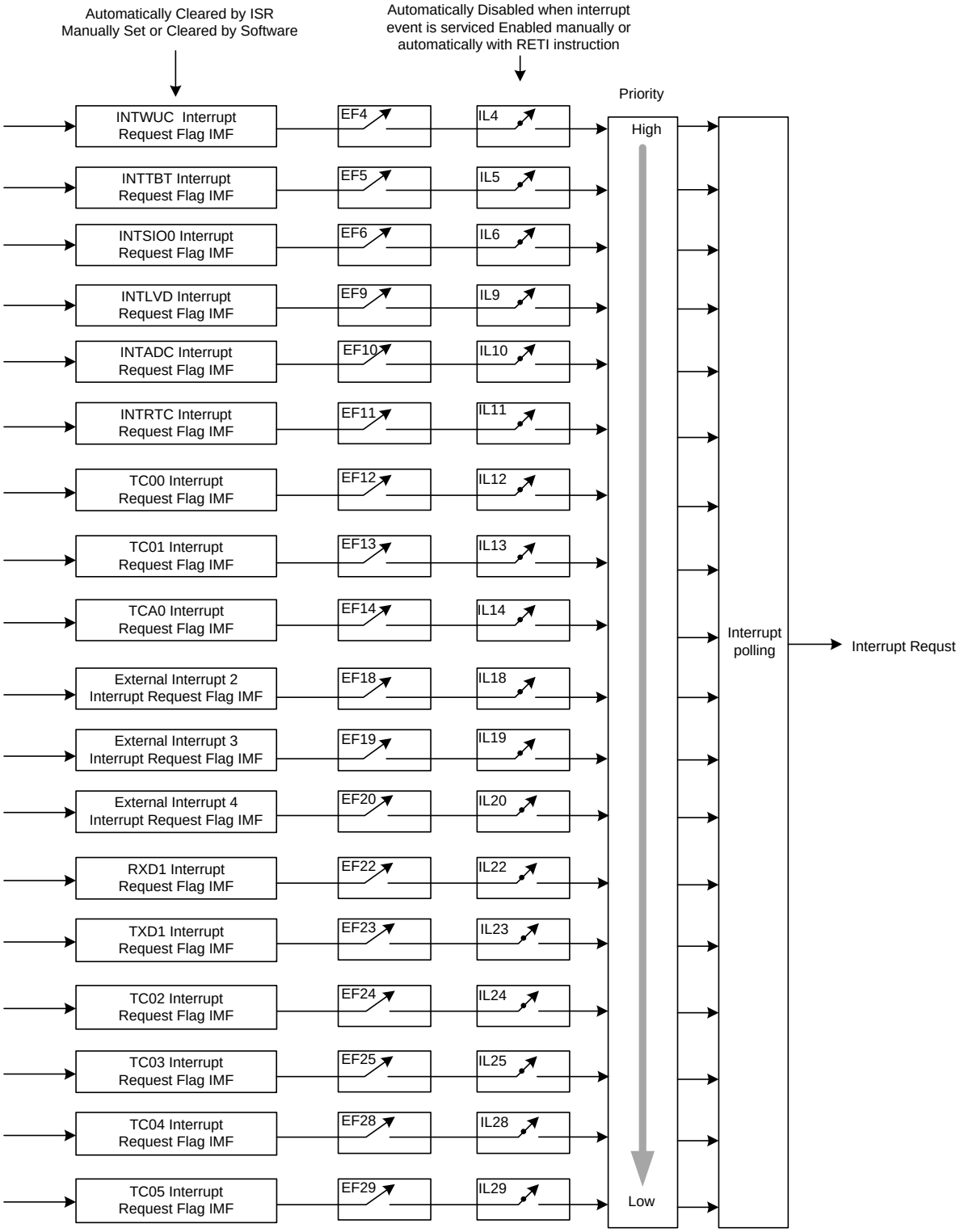



Figure7.1 Interrupts Control Scheme

## 7.1 Interrupts Latches (IL29 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 (EF29 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 | - | IL6 | IL5 | IL4 | IL3 | - | - | - |
| Read/Write | R | R/W | R/W | R/W | R | R | R | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | - | 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 | - |
| 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 |
| Function | INTSBI | INTTCA0 | INTTC01 | INTTC00 | INTRTC | INTADC | INTLVD | - |

Interrupt Latch (ILE)

| ILE (0x0FE2) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | IL23 | IL22 | - | IL20 | IL19 | IL18 | - | - |
| Read/Write | R/W | R/W | R | R/W | R/W | R/W | R | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Function | INTTXD1 | INTRXD1 | - | INT4 | INT3 | INT2 | - | - |
|---|---|---|---|---|---|---|---|---|

Interrupt Latch (ILD)

| ILD (0x0FE3) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | IL29 | IL28 | - | - | IL25 | IL24 |
| Read/Write | R | R | R/W | R/W | R | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | - | - | INTTC05 | INTTC04 | - | - | INTTC03 | INTTC02 |

| | | | Read | Write |
|---|---|---|---|---|
| IL29 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 | No interrupt request | - |
| | | 1 | 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 | - | EF6 | EF5 | EF4 | - | - | - | IMF |
| Read/Write | R | R/W | R/W | R/W | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | - | 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 | - |
| 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 |
| Function | INTSBI | INTTCA0 | INTTC01 | INTTC00 | INTRTC | INTADC | INTLVD | - |

Interrupt Enable Register (EIRE)

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

Interrupt Enable Register (EIRD)

| EIRD (0x003D) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | EF29 | EF28 | - | - | EF25 | EF24 |
| Read/Write | R | R | R/W | R/W | R | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | - | - | INTTC05 | INTTC04 | - | - | INTTC03 | INTTC02 |

| | | | |
|---|---|---|---|
| EF29 to EF4 | Individual Interrupt Enable Flag (specified for each bit) | 0<br>1 | Disable the acceptance of each maskable interrupt<br>Enable the acceptance of each maskable interrupt |
| IMF | Interrupt Master Enable Flag | 0<br>1 | Disable the acceptance of all maskable interrupts<br>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 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.

### 7.5.1 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.

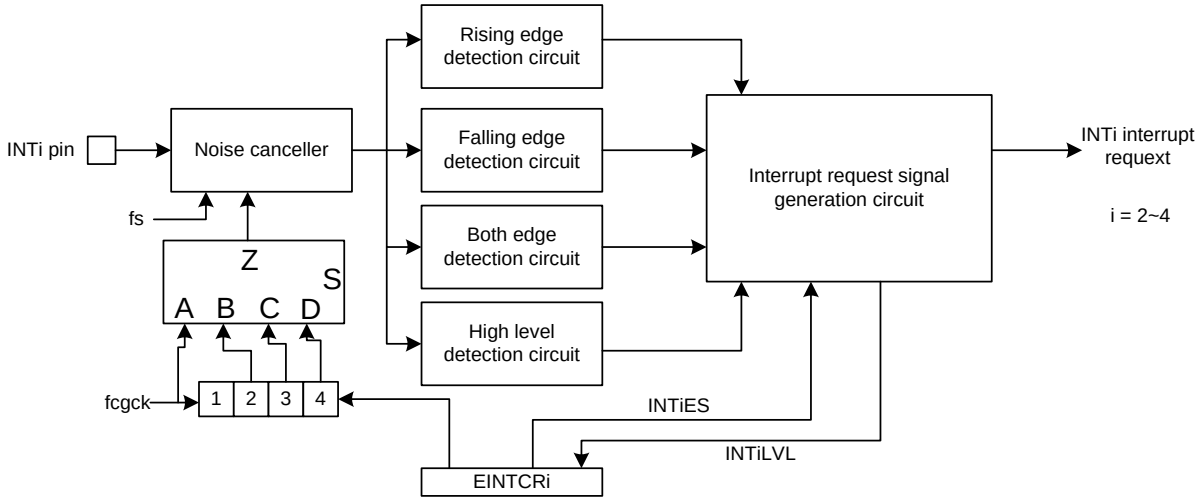



Figure7.3 External Interrupts 2/3/4

### 7.5.2 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 | - | - | - | INT4EN | INT3EN | INT2EN | - | - |
| Read/Write | R | R | R | R/W | R/W | R/W | R | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| **INT4EN** | INT4 Control | 0<br>1 | Disable<br>Enable |
| **INT3EN** | INT3 Control | 0<br>1 | Disable<br>Enable |
| **INT2EN** | INT2 Control | 0<br>1 | Disable<br>Enable |

*Note 1): Clearing INTxEN (x=2 to 4) 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 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 ~Bit 5 and Bit1~Bit0 of POFFSET3 are read as "0".*

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"<br>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<br><br>01: An interrupt request is generated at the falling edge of the noise canceller pass signal<br><br>10: An interrupt request is generated at both edges of the noise canceller pass signal<br><br>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]<br>01: fcgck / $2^2$ [Hz]<br>10: fcgck / $2^3$ [Hz]<br>11: fcgck / $2^4$ [Hz] | 00: fs/4 [Hz]<br>01: fs/4 [Hz]<br>10: fs/4 [Hz]<br>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"<br>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<br><br>01: An interrupt request is generated at the falling edge of the noise canceller pass signal<br><br>10: An interrupt request is generated at both edges of the noise canceller pass signal<br><br>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]<br>01: fcgck / $2^2$ [Hz]<br>10: fcgck / $2^3$ [Hz]<br>11: fcgck / $2^4$ [Hz] | 00: fs/4 [Hz]<br>01: fs/4 [Hz]<br>10: fs/4 [Hz]<br>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".*

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"<br>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<br><br>01: An interrupt request is generated at the falling edge of the noise canceller pass signal<br><br>10: An interrupt request is generated at both edges of the noise canceller pass signal<br><br>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]<br>01: fcgck / $2^2$ [Hz]<br>10: fcgck / $2^3$ [Hz]<br>11: fcgck / $2^4$ [Hz] | 00: fs/4 [Hz]<br>01: fs/4 [Hz]<br>10: fs/4 [Hz]<br>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".*

### 7.5.3 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 |
| INT2 | INT2 | IMF=1<br>EF18=1 | Falling edge<br>Rising edge<br>Both edges | Less than 2/fspl: Noise<br>More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate<br>More than 3/fspl+1/fcgck: Signal | Less than 4/fs: Noise<br>More than 4/fs and less than 8/fs: Indeterminate<br>More than 8/fs: Signal |
| INT3 | INT3 | IMF=1<br>EF19=1 | Falling edge<br>Rising edge<br>Both edges | Less than 2/fspl: Noise<br>More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate<br>More than 3/fspl+1/fcgck: Signal | Less than 4/fs: Noise<br>More than 4/fs and less than 8/fs: Indeterminate<br>More than 8/fs: Signal |
| INT4 | INT4 | IMF=1<br>EF20=1 | Falling edge<br>Rising edge<br>Both edges<br>"H" level | Less than 2/fspl: Noise<br>More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate<br>More than 3/fspl+1/fcgck: Signal | Less than 4/fs: Noise<br>More than 4/fs and less than 8/fs: Indeterminate<br>More than 8/fs: Signal |

Table 7.2 External Interrupts

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

#### 7.5.3.1 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.*

#### 7.5.3.2 External Interrupt 2 to 4

**External interrupt 2/3/4**

External interrupt 2/3/4 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 2/3/4.

| EINTCRx<INTxES> | Detectedat |
|---|---|
| 00 | Rising edge |
| 01 | Falling edge |
| 10 | Both edges |
| 11 | "H" level interrupt |

**Table 7.3Selection of Interrupt Request Generation Edge**

**(b) A Noise Canceller Pass Signal Monitoring Function when Interrupt Request Signals Generated**

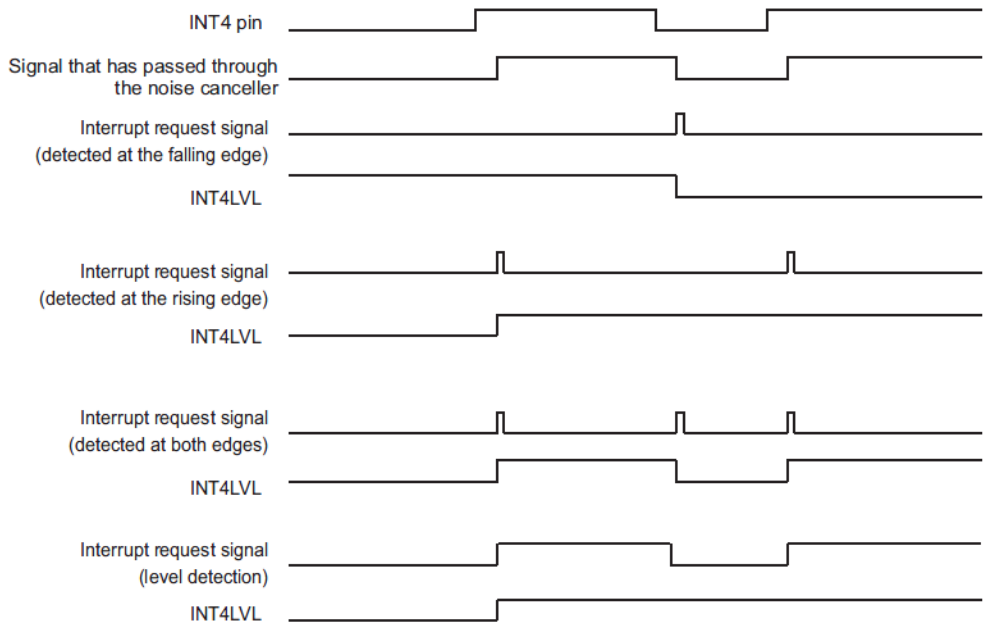



**Figure7.4 Interrupt Request Generation and EINTCRx<INTxLVL> (x = 4)**

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

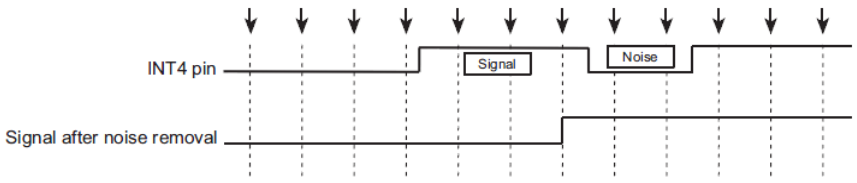



Figure7.5 Noise Cancel Operation(x=4)

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.*

# 8. I/O Ports

MQ6812/MQ6821 has 7 parallel input / output ports (max. 28 I/O pins and 2 input pins) as follows:

| Port Name | Pin Name | No. of Pins | Input/output | Secondary Functions |
|---|---|---|---|---|
| Port P0 | P01 to P00 | 2 | Input / Output | Also used as the high-frequency oscillator connection pin and the low-frequency oscillator connection pin |
| Port P1 | P10 | 1 | Input / Output | Also used as the external reset input, the external interrupt input and the STOP mode release signal input |
| Port P2 | P27to P23 | 5 | Input / Output | Also used as the UART input/output, the I2C input/output , LCD driving pin, and the external interrupt input. |
| Port P4 | P47 to P40 | 8 | Input / Output | Also used as the analog input , LCD pin, and the external interrupt. |
| Port P7 | P77 to P70 | 8 | Input / Output | Also used as the timer counter input/output, DVO output, LCD pin ,SIO input/output, the analog input , the external interrupt and the key-on wakeup input |
| Port P8 | P83 to P80 | 4 | Input / Output (P80/P81:output) | Also used as the timer counter input/output, LCD pin, and LED high driving pin (P80/P81). |
| 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 LCD pin. |

Table 8.1List of I/O Ports

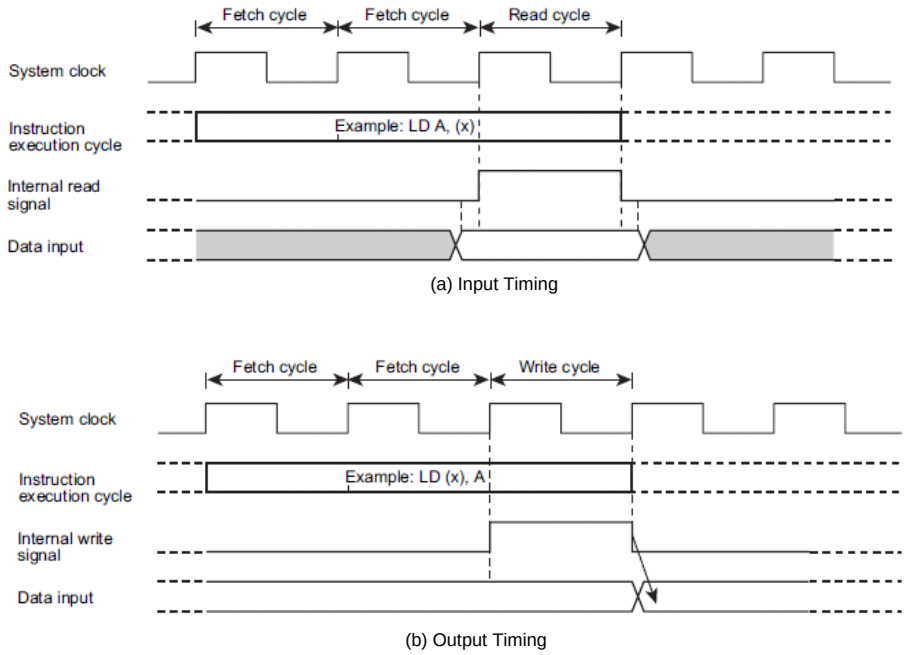



Figure8.1 Input / Output Timing (Example)

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.

### PxPD Register

This register determines whether or not the built-in pull-down resistor is connected when a port is used in the input mode.

## 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 | PxFC | Other Required Settings |
| Port P0 | P01 to P00 | Port input | 0 | 0 | |
| | | Port output | 1 | 0 | |
| | P01 | XOUT | * | Without register | |
| | P00 | XIN | * | 1 | |
| Port P1 | P10 | Port input | 0 | Without register | Note 1 |
| | P10 | Port output | 1 | | Note 1 |
| | P10 | RESETB input | * | | Note 1 |
| Port P2 | P27 to P23 | Port input | 0 | * | |
| | | Port output | 1 | 0 | |
| | P27 to P24 | KWI 11 to KWI 8 | * | * | KWUCR2 |
| | P27 to P25 | SEG2 to SEG0 output | 0 | 0 | P2PU <x> = "0", x = 7~5<br>LCDCR3 (Note 3) |
| | P24 to P23 | COM7 to COM6 output | 0 | 0 | P2PU <x> = "0", x = 4, 3<br>LCDCR5 (Note 3) |
| | | I2C pin SCL/SDA | 1 | 1 | |
| Port 4 | P47 to P40 | Port input | 0 | * | |
| | | Port output | 1 | 0 | |
| | | KWI 7 to KWI 4 | * | * | KWUCR1 |
| | | KWI 3 to KWI 0 | * | * | KWUCR0 |
| | | AIN7 to AIN0 | 0 | 1 | |
| | P47 to P44 | SEG11 to SEG8 output | 0 | 0 | P4PU <x> = "0", x = 7~4<br>LCDCR4 (note 3) |
| | P43 | VREF input | 0 | 1 | |
| Port P7 | P77 to P70 | Port input | 0 | * | |
| | | Port output | 1 | 0 | |
| | P76 to P73 | KWI 15 to KWI 12 | * | * | KWUCR3 |
| | P77 to P74 | SEG3,SEG6,SEG4,SEG7 | 0 | 0 | P7PU <x> = "0", x = 7~4<br>LCDCR3 (not 3) |
| | P77 | INT4 input | 0 | Without register | |
| | | SIO, SS input | 0 | * | |
| | | SIO, SS output | 1 | 1 | |

| Port Name | Pin Name | Function | Register Set Value | | |
|---|---|---|---|---|---|
| | | | PxCR | PxFC | Other Required Settings |
| Port P7 | P76 | INT3 input | 0 | Without register | Note 1 |
| | | SIO, SCLK input | 0 | * | Note 1 |
| | | SIO,SCLK output | 1 | 1 | Note 1 |
| | P75 | INT2 input | 0 | Without register | |
| | | SIO,SO input | 0 | * | |
| | P74 | DVO output | 1 | 1 | |
| | | AIN8 | 0 | 1 | |
| | | SIP,SI output | 1 | 1 | |
| | P72 | TCA0 input | 0 | * | |
| | | PPGA0B output | 1 | 1 | |
| | P71 | PPG01B output | 1 | 1 | |
| | P70 | PPG00B输出 | 1 | 1 | |
| | | SEG5 | 0 | 0 | P7PU <0> = "0"<br>LCDCR3 (note 3) |
| Port P8 | P83 to P80 | Port input | 0 | * | |
| | P83 to P82 | Port output | 1 | 0 | |
| | P83 to P82 | COM5 to COM4 | 0 | 0 | P8PU <x> = "0", x = 3, 2<br>LCDCR5 (note 3) |
| | P83 | PPG05B output | 1 | 1 | |
| | P82 | PPG04B output | 1 | 1 | |
| | P81 | PPG03B output | 1 | 1 | |
| | P80 | PPG02B output | 1 | 1 | |
| Port P9 | P91 to P90 | Port input | 0 | * | |
| | | Port output | 1 | 0 | |
| | | COM1 to COM0 | 0 | 0 | P9PU <x> = "0", x = 1, 0<br>LCDCR5 (note 3) |
| | P91 | RXD1 input | 0 | 0 | UATCNG <UAT1IO>="0" |
| | | TXD1 output | 1 | 1 | UATCNG <UAT1IO>="1" |
| | P90 | TXD1output | 1 | 1 | UATCNG <UAT1IO>="0" |
| | | RXD1input | 0 | 0 | UATCNG <UAT1IO>="1" |

Table 8.2List 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<br>(Operation is the same whether "1" or "0" is selected ) |
| Without register | There is no register that corresponds to the bit |

*Note 3]: When use LCD function, set "POFFCR2<LCDEN >=1" and "LCDCR1<EDSP>=1", and the PxPU register have to set "0".Detail please refer "11. LCD " chapter.*

## 8.3 I/O Port Control Register

### 8.3.1 Port P0 (P01 to P00) Register

Port P0 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 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 | P01 | P00 |
|---|---|---|
| Secondary function | XOUT | XIN |

Table 8.3 Port P0

Port P0 Output Latch Register

| P0DR (0x0000) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | - | - | P01 | P00 |
| 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 P0 Input / Output Control Register

| P0CR (0x0F1A) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | Bit Symbol | - | - | - | - | - | P0CR1 | P0CR0 |
| Read/Write | | Read/Write | R | R | R | R | R | R/W | R/W |
| After reset | | After reset | 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 | | Bit Symbol | - | - | - | - | - | - | P0FC0 |
| Read/Write | | Read/Write | R | R | R | R | R | R | R/W |
| After reset | | After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | | | | Port function |
| | 1 | | | | | | | | XIN (I) |

*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.*

Port P0 Built-in Pull-up Resistor Control Resistor

| P0PU (0x0F27) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| BitSymbol | | - | - | - | - | - | - | P0PU1 | P0PU0 |
| Read/Write | | R | R | R | R | R | R | R/W | R/W |
| Afterreset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | | | The built-in resistor is not connected. | |
| | 1 | | | | | | | The built-in pull-up resistor is connected. (Note) | |

Port P0 Built-in Pull-down Resistor Control Resistor

| P0PD (0x0F00) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| BitSymbol | | - | - | - | - | - | - | P0PD1 | P0PD0 |
| Read/Write | | R | R | R | R | R | R | R/W | R/W |
| Afterreset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | | | The built-in resistor is not connected. | |
| | 1 | | | | | | | *The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected.* | |

*Note): If P0PUx and P0PDx set to "1" at the same time, the port is only connected to pull-up resistor.(x=0,1)*

**Port P0 Input Data Register**

| P0PRD (0x000D) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | - | P0PRD1 | P0PRD0 |
| 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. | |

## 8.3.2 Port P1(P10) Register

Port P1 is a 1-bit input / output por t (P10) 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 | P10 |
|---|---|
| Secondary function | RESETB |

**Table 8.4 Port P1**

**Port P1 Output Latch Register**

| P1DR (0x0001) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| BitSymbol | | - | - | - | - | - | - | - | P10 |
| Read/Write | | R | R | R | R | R | R | R | 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 | | - | - | - | - | - | - | - | P1CR0 |
| Read/Write | | R | R | R | R | R | R | R | R/W |
| Afterreset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | | | | Input mode (port input) |
| | 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 | | - | - | - | - | - | - | - | P1PU0 |
| Read/Write | | R | R | R | R | R | R | R | R/W |
| Afterreset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | | | | The built-in resistor is not connected. |
| | 1 | | | | | | | | 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 |
|---|---|---|---|---|---|---|---|---|
| BitSymbol | - | - | - | - | - | - | - | P1PRD0 |
| Read/Write | R | R | R | R | R | R | R | R |
| Afterreset | 0 | 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".*

### 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.

| Port Name | P27 | P26 | P25 | P24 | P23 |
|---|---|---|---|---|---|
| Secondary function | KWI11 | KWI10 | KW9 | KWI8 SCL | SDA |

Figure 8.4 Port P2

### Port P2 Output Latch Register

| P2DR (0x0002) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | P27 | P26 | P25 | P24 | P23 | - | - | - |
| 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 |
| Function | 0 | Outputs L level when the output mode is selected. | | | | | | | |
| | 1 | Outputs H level when the output mode is selected | | | | | | | |

### Port P2 Input / Output Control Register

| P2CR (0x0F1C) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | P2CR7 | P2CR6 | P2CR5 | P2CR4 | P2CR3 | - | - | - |
| 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 |
| Function | 0: | Input mode (port input) | | | | | | | |
| | | | | | | | | | |
| | 1 | Output mode (port output) | | | | | | | |
| | | | | | SCL (I/O) | SDA (I/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 | | - | - | - | P2FC4 | P2FC3 | - | - | - |
| Read/Write | | R | R | R | R/W | R/W | R | R | R |
| After reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | Port function | | | | | | | |
| | 1 | | | | SCL (I/O) | SDA (I/O) | | | |

### Port P2 Built-in Pull-up Resistor Control Resistor

| P2PU (0x0F29) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | P2PU7 | P2PU6 | P2PU5 | P2PU4 | P2PU3 | - | - | - |
| 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 |
| Function | 0 | The built-in resistor is not connected. | | | | | | | |
| | 1 | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | | | | | | | |

Port P2 Built-in Pull-down Resistor Control Resistor

| P2PD (0x0F02) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | P2PD7 | P2PD6 | P2PD5 | P2PD4 | P2PD3 | - | - | - |
| 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 |
| Function | 0 | The built-in resistor is not connected. | | | | | | | |
| | 1 | The resistor is connected in the input mode only. 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 | - | - | - |
| Read/Write | R | R | R | R | R | R | R | R |
| After reset | * | * | * | * | * | 0 | 0 | 0 |
| 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. ° | | | | | | | |

Port P2 Output Control Register

| P2OUTCR (0x0F43) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | P2OUTCR4 | P2OUTCR3 | - | - | - |
| Read/Write | | R | R | R | R/W | R/W | R | R | R |
| After reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | Port function | | | | | | | |
| | 1 | | | | | | | | |

## 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, LCD Driving, ADC input.

| Port Name | P47 | P46 | P45 | P44 | P43 | P42 | P41 | P40 |
|---|---|---|---|---|---|---|---|---|
| Secondary function | AIN7 KWI7 | AIN6 KWI6 | AIN5 KWI5 | AIN4 KWI4 | AIN3 KWI3 | AIN2 KWI2 | AIN1 KWI1 | AIN0 KWI0 |
| LCD | SEG8 | SEG9 | SEG10 | SEG11 | | | | |

Table 8.5 Port P4

Note) : When use LCD function, set LCD related pin (COM and SEG) as input mode.

### 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) | AIN6 (I) | AIN5 (I) | AIN4 (I) | AIN3 (I)<br>VREF (I) | AIN2 (I) | AIN1 (I) | AIN0 (I) |
| | 1 | Output mode (port output) | | | | | | | |

*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) | AIN4 (I) | AIN3 (I)<br>VREF (I) | AIN2 (I) | AIN1 (I) | AIN0 (I) |

### 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 resistor is not connected. | | | | | | | |
| | 1 | 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 Built-in Pull-down Resistor Control Register**

| P4PD (0x0F04) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| BitSymbol | | P4PD7 | P4PD6 | P4PD5 | P4PD4 | P4PD3 | P4PD2 | P4PD1 | P4PD0 |
| 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 resistor is not connected. | | | | | | | |
| | 1 | The resistor is connected in the input mode only. 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 8.7 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,LCD driving,SIO pins, and the timer counter input/output.

| | P77 | P76 | P75 | P74 | P73 | P72 | P71 | P70 |
|---|---|---|---|---|---|---|---|---|
| Secondary function | INT4<br>SS | INT3<br>KWI15<br>SCLK | INT2<br>KWI14<br>SO | KWI13<br>SI<br>DVOB<br>AIN8 | KWI12 | PPGA0B<br>TCA0 | PPG01B | PPG00B |
| LCD | SEG3 | SEG6 | SEG4 | SEG7 | COM3 | COM2 | | SEG5 |

**Table 8.7Port P7**

Note) : When use LCD function, set LCD related pin (COM and SEG) as input mode.

Port P7 Output Latch Register

| P7DR (0x0007) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| BitSymbol | | 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 |
| 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 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)<br>SS (I) | INT3 (I)<br>SCLK (I) | INT2 (I) | AIN8 (I)<br>SI (I) | | TAC0 (I) | | |
| | 1: | Output mode (port output) | | | | | | | |
| | | SS (O) | SCLK (O) | SO (O) | DVOB (O) | | PPGA0B (O) | PMW01B (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 |
|---|---|---|---|---|---|---|---|---|---|
| BitSymbol | | P7FC7 | P7FC6 | P7FC5 | P7FC4 | - | P7FC2 | P7FC1 | P7FC0 |
| Read/Write | | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W |
| Afterreset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0: | | | | Port function | | | | |
| | 1: | SS(O) | SCLK (O) | SO (O) | DVOB (O)<br>AIN8 (I)<br>SI (I) | | PPGA0B (O) | PPG01B (O) | PPG00B (O) |

Note): When P7CR4 set to "0" , it is AIN8. When P7CR4 set to "1" , it is DVOB.

### Port P7 Built-in Pull-up Resistor Control Resistor

| P7PU (0x0F2E) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | P7PU7 | P7PU6 | P7PU5 | P7PU4 | P7PU3 | P7PU2 | P7PU1 | P7PU0 |
| 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 | The built-in resistor is not connected. | | | | | | | |
| | 1 | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | | | | | | | |

### Port P7 Built-in Pull-down Resistor Control Resistor

| P7PD (0x0F07) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | P7PD7 | P7PD6 | P7PD5 | P7PD4 | P7PD3 | P7PD2 | P7PD1 | P7PD0 |
| Read/Write | | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |
| After reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | The built-in resistor is not connected. | | | | | | | |
| | 1 | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | | | | | | | |

*Note: If set P7PUx to"1" and P7PDx to "1" at the same time, port is only connect to pull-up resistor (x=7 to 0*

### 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. | | | | | | | |

### 8.3.6 Port P8 (P83 to P80) Register

Port P8 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 timer counter input/output, and LCD driving. P80 and P81 are input port only.

| Port Name | P83 | P82 | P81 | P80 |
|---|---|---|---|---|
| Secondary function | PPG05B | PPG04B | PPG03B | PPG02B |
| LCD | COM5 | COM4 | - | - |

**Table 8.8 Port P8**

Note) : When use LCD function, set LCD related pin (COM and SEG) as input mode

**Port P8 Output Latch Register**

| P8DR (0x0008) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | P83 | P82 | P81 | P80 |
| 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 P8 Input / Output Control Register**

| P8CR (0x0F22) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | P8CR3 | P8CR2 | P8CR1 | P8CR0 |
| 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) | | Do not use Input mode. | |
| | 1: | | | | | Output mode (port output) | | | |
| | | | | | | PPG05B (O) | PPG04B (O) | PPG03B (O) | PPG02B (O) |

### Port P8 Function Control Register

| P8FC (0x0F3C) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | P8FC3 | P8FC2 | P8FC1 | P8FC0 |
| 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: | | | | | Port function (P80/P81 is input only) | | | |
| | 1: | | | | | PPG05B (O) | PPG04B (O) | PPG03B (O) | PPG02B (O) |

### Port P8 Built-in Pull-up Resistor Control Resistor

| P8PU (0x0F2F) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | P8PU3 | P8PU2 | - | - |
| Read/Write | | R | R | R | R | R/W | R/W | R | R |
| After reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | The built-in resistor is not connected. | | | |
| | 1 | | | | | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | | | |

### Port P8 Built-in Pull-down Resistor Control Resistor

| P8PD (0x0F08) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | P8PD3 | P8PD2 | - | - |
| Read/Write | | R | R | R | R | R/W | R/W | R | R |
| After reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0 | | | | | The built-in resistor is not connected. | | | |
| | 1 | | | | | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | | | |

*Note: If set P8PUx to"1" and P78Dx to "1" at the same time, port is only connect to pull-up resistor (x=0,1)*

### Port P8 Input Data Register

| P8PRD (0x0015) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | P8PRD3 | P8PRD2 | P8PRD1 | P8PRD0 |
| 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".*

### 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.

| Port Name | P91 | P90 |
|---|---|---|
| Secondary function | RXD1<br>TXD1 | TXD1<br>RXD1 |
| LCD | COM1 | COM0 |

**Table 8.10 Port P9**

Note) : When use LCD function, set LCD related pin (COM and SEG) as input mode

**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. | |

**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) |

UART Input/Output Change Control Register

| UATCNG (0x0E57) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | - | UAT2IO | UAT1IO | - |
| Read/Write | | R | R | R | R | R | R/W | R/W | R |
| After reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Function | 0: | | | | | | Set P90 and P91 as UART function. | Set P90 is TXD; P91 is RXD. | |
| | 1: | | | | | | Set P26 and P27 as UART function.(P26. P27 can not use TCC function at the same time) | Set P90 is RXD; P91 is TXD. | |

*Note): Bit2(UAT2IO) use to choose P90/P91 or P26/P27 as UART pin. If Bit2=0, P90/P91 is UART, and set P90/P91 as TXD1/RXD1 by Bit1 (UAT1IO). If Bit2=1, P26/P27 is UART(P26/P27 cannot use TCC function at this time), and set P26/P27 as TXD1/RXD1 by Bit1 (UAT1IO). The operation for changing UATCNG must be executed while the applicable serial interface operations are stopped.*

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 Built-in Pull-up Resistor Control Resistor

| 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 resistor is not connected. | |
| | 1 | | | | | | | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | |

### Port P9 Built-in Pull-down Resistor Control Resistor

| P9PD (0x0F09) | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | - | - | - | - | - | - | P9PD1 | P9PD0 |
| 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 resistor is not connected. | |
| | 1 | | | | | | | The resistor is connected in the input mode only. Under any other conditions, setting to "1" does not make the resistor connected. | |

*Note: If set P9PUx to"1" and P79Dx to "1" at the same time, port is only connect to pull-up resistor (x=0,1)*

### 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 modet, the contents of the port are read.<br><br>If not, "0" is read. | |

*Note : "*" means "don't care".*

# 9.10-bit AD Converter (ADC)

MQ6812/MQ6821 has a real 10-bit AD converter (ADC), which is a successive approximation type ADC. There are maximum 9 input pins(AIN0 to AIN8), and 1 internal 1/4 VDD battery measure input pin.

## 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.

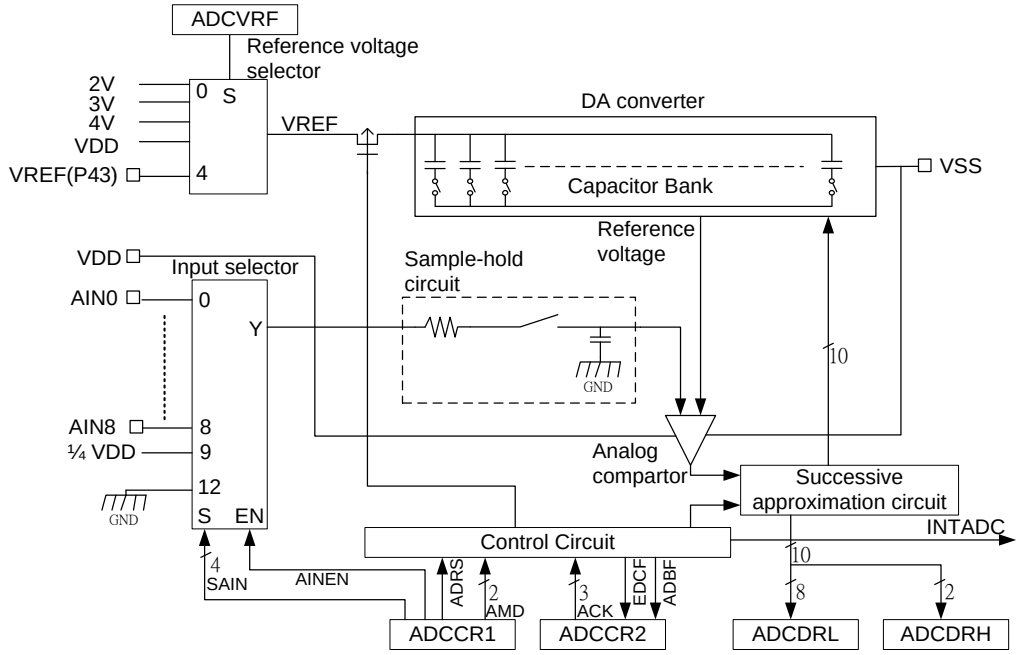



Figure 9.1 10-bit AD Converter

*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:<br>1: | -<br>AD conversion start |
| **AMD** | AD operating mode | 00:<br>01:<br>10:<br>11: | AD operation disable, forcibly stop AD operation<br>Single mode<br>Reserved<br>Repeat mode |
| **AINEN** | Analog input control | 0:<br>1: | Analog input disable<br>Analog input enable |
| **SAIN** | Analog input channel select | 0000<br>0001:<br>0010:<br>0011:<br>0100:<br>0101:<br>0110:<br>0111:<br>1000<br>1001<br>1010<br>1011<br>1100<br>Others | AIN0<br>AIN1<br>AIN2<br>AIN3<br>AIN4<br>AIN5<br>AIN6<br>AIN7<br>AIN8<br>1/4 VDD input<br>Reserved<br>Reserved<br>ground<br>- |

*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).*

*Note 6):When P74 is AIN8, and other AIN is operate in at the same time, it would increase extra power consumption. Avoid using P74 as AIN8, if there are still enough AIN pin.*

AD Converter Control Register 2

| ADCCR2 (0x0035H) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | EOCF | ADBF | - | - | - | 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:<br>1: | Before conversion or during conversion<br>Conversion end |
| **ADBF** | AD conversion BUSY flag | 0:<br>1: | AD conversion being halted<br>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 | 8MHz | 4MHz | 2MHz | 1MHz | 0.5MHz | 0.25MHz |
| 000 | 32/fcgck | - | - | - | 16.0 us | 32.0 us | 64.0 us | 128.0 us |
| 001 | 64/fcgck | - | - | 16.0 us | 32.0 us | 64.0 us | 128.0 us | - |
| 010 | 128/fcgck | - | 16.0 us | 32.0 us | 64.0 us | 128.0 us | - | - |
| 011 | 256/fcgck | 16.0 us | 32.0 us | 64.0 us | 128.0 us | - | - | - |
| 100 | 512/fcgck | 32.0 us | 64.0 us | 128.0 us | - | - | - | - |
| 101 | 1024/fcgck | 64.0 us | 128.0 us | - | - | - | - | - |

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): When using other fcgck, please set ACK to calculate conversion time, please keep conversion time equal or more than 16us.*

*Note 4): 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 Converter Reference Voltage Register

| ADCVRF (0x0EE7) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | VRFSEL | | |
| Read/Write | R | R | R | R | R | R/W | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| VRFSEL | Choose AD Converter Reference Voltage | 000: | ADC internal reference voltage = 2.0V |
| | | 001: | ADC internal reference voltage = 3.0V |
| | | 010: | ADC internal reference voltage = 4.0V |
| | | 011: | ADC internal reference voltage = VDD |
| | | 100: | ADC reference voltage = external reference voltage(VREF) |
| | | 101: | Syetem reserved. |
| | | 110: | Syetem reserved. |
| | | 111: | Syetem reserved. |

### AD internal reference voltage auto caliberation register

| VREF_CALIB (0x0EF6) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | - | VRFCAL | - |
| Read/Write | R | R | R | R | R | R | R/W | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| VRFCAL | ADC internal voltage autocaliberation function | 0: auto calibration disable<br>1: auto calibration enable |

*Note：Detail of auto-calibration function,please refer to appendix D.*

### 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".

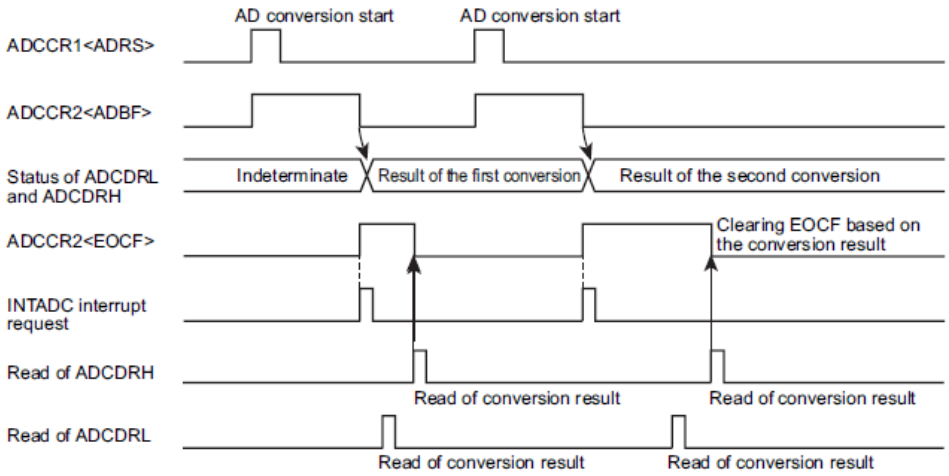



Figure 9.2 Single Mode

*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"*

### 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.

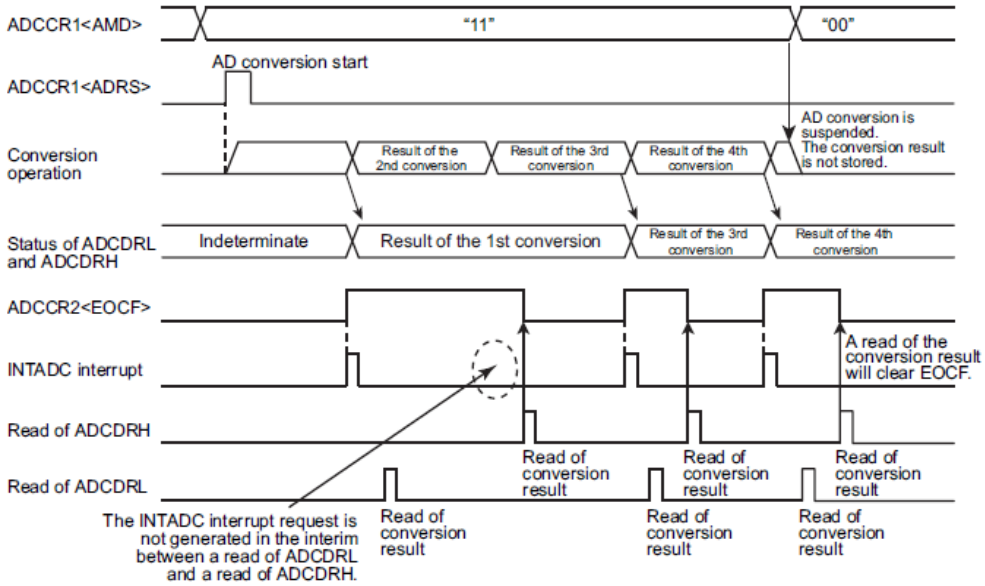



Figure 9.3 Repeat Mode

### 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:
    1. From the AD input channel select (SAIN), select the channel in which AD conversion is to be performed.
    2. Set the analog input control (AINEN) to "Analog input enable".
    3. 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.

## 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.

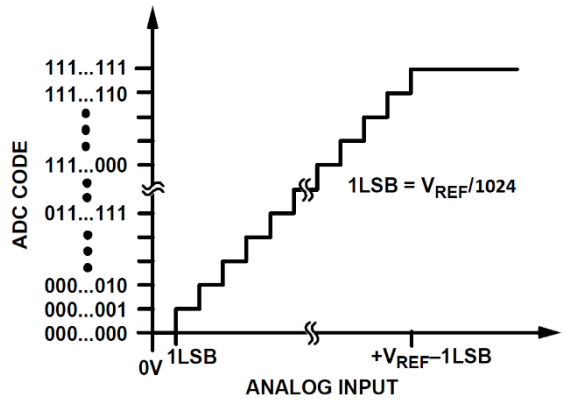



Figure 9.4 Relationships between Analog Input Voltages and AD-converted Values (Typical Values)

## 9.7 Precautions about the AD Converter

### 9.7.1 Analog Input Pin Voltage Range

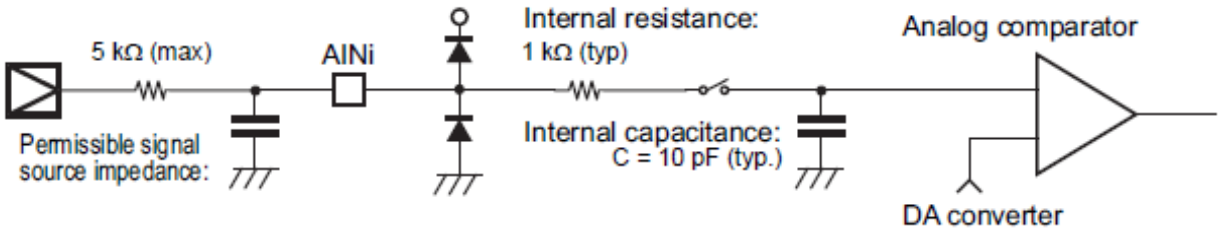
Analog input pins (AIN0 through AIN8) should be used at voltages from VAREF to VSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain, and converted values on other pins will also be affected.

### 9.7.2 Analog Input Pins Used as Input / Output Ports

Analog input pins (AIN0 through AIN8) 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/ Watchdog Timer2 (WDT/WDT2)

Because the WDT/WDT2 of MQ6812/MQ6821 are fixed and cannot be stopped , please clear WDT/WDT2 per 1.5 second at lease.

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.

The watchdog timer2(WDT2) use internal low-frequency clock. The counter value of WDT2 is 0xFFFF,which is fixed. The WDT2 clear function is the same as watchdog timer (WDT).

*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

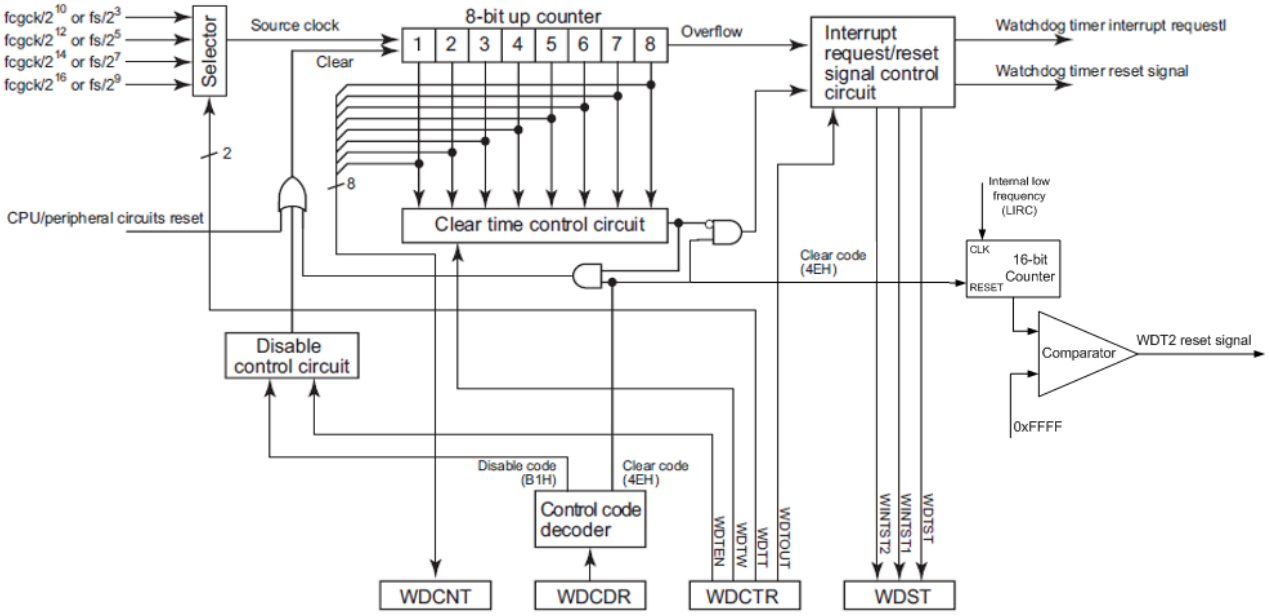



Figure 10.1 Watchdog Timer Configuration

### 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 controlled by the watchdog timer control code register (WDCDR).

The watchdog timer/ watchdog timer2 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 |

| | | | | | |
|---|---|---|---|---|---|
| WDTEN | Enable / disable the watch-dog timer | 0: Disable<br>1: Enable | | | |
| WDTW | Set the clear time of the 8-bit up counter. | 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.<br><br>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.<br><br>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.<br><br>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. | | | |
| WDTT | Set the overflow time of the 8-bit up counter. | | NORMAL mode | | SLOW mode |
| | | | DV9CK=0 | DV9CK=1 | |
| | | 00: | $2^{18}$/fcgck | $2^{11}$/fs | $2^{11}$/fs |
| | | 01: | $2^{20}$/fcgck | $2^{13}$/fs | $2^{13}$/fs |
| | | 10: | $2^{22}$/fcgck | $2^{15}$/fs | $2^{15}$/fs |
| WDTOUT | Select an overflow detection signal of the 8-bit up counter. | 0 : Watchdog timer interrupt request signal<br>1 : Watchdog timer reset request signal | | | |

*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/ the watchdog timer2. (clear code)<br><br>0xB1: Disable the watchdog timer operation and clear the 8-bit up counter when WDCTR <WDTEN> is "0". (disable code)<br><br>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. |

*Note 1): WDCNT is only use for the watchdog timer(WDT). The counter value of watchdog timer2(WDT2) is "0xFFF", cannot be changed by register.*

### 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.<br><br>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.<br><br>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<br>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.*

### 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.

Set the clear code "0x4E" to WDCDR register, would clear the watchdog timer(WDT)/ the watchdog timer2(WDT2) at the same time.

The counter is different between the watchdog timer(WDT) and the watchdog timer2(WDT2).The counter value of the watchdog timer2(WDT2) is 0xFFF, and this values cannot be changed by register.

#### 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.

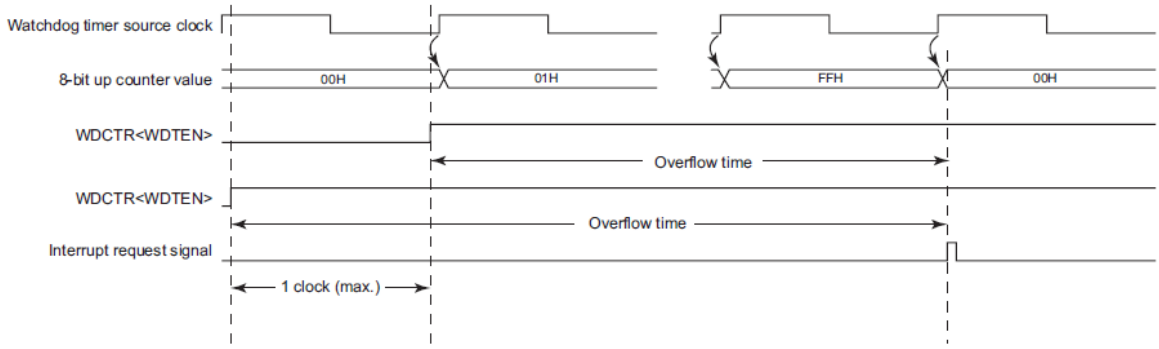



Figure 10.2 WDCTR <WDTEN> Set Timing and Overflow Time

*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.

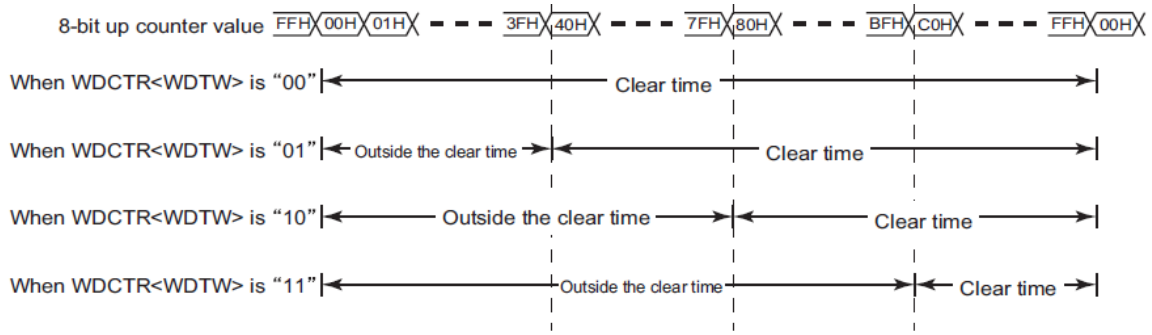



Figure 10.3 WDCTR <WDTW> and the 8-bit Up Counter Clear Time

#### 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.

#### 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.

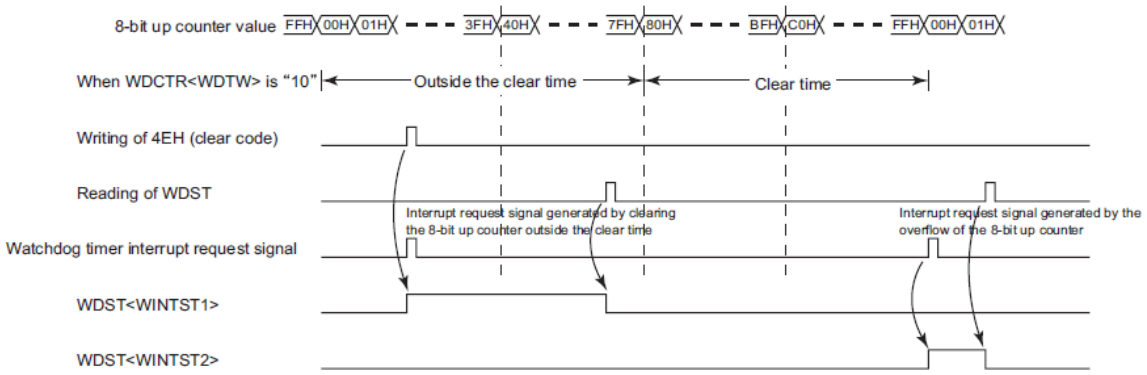



Figure 10.4 Changes in the Watchdog Timer Status

### 10.1.3.8 Read the counter value of WDT2 and the clear the WDT2

The counter is different between the watchdog timer(WDT) and the watchdog timer2(WDT2).The counter value of the watchdog timer2(WDT2) is 0xFFF, and this values cannot be changed by register.

Set the clear code "0x4E" to WDCDR register, would clear the watchdog timer(WDT)/ the watchdog timer2(WDT2) at the same time.

## 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

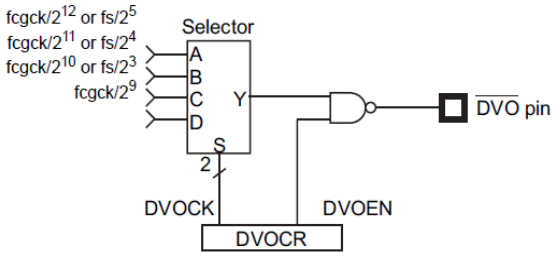



Figure 10.5 Divider Output

### 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 |

| | | | | | |
|---|---|---|---|---|---|
| DVOEN | Enable / disable the divider output | 0: Disable<br>1: Enable | | | |
| DVOCK | Select the divider output frequency<br>Unit: [Hz] | | Normal 1/2, IDLE 1/2 mode | | SLOW 1/2 mode<br>SLEEP 1/2 mode |
| | | | DV9CK=0 | DV9CK=1 | |
| | | 00: | fcgck/$2^{12}$ | fs/$2^5$ | fs/$2^5$ |
| | | 01: | fcgck/$2^{11}$ | fs/$2^4$ | fs/$2^4$ |
| | | 10: | fcgck/$2^{10}$ | fs/$2^3$ | fs/$2^3$ |
| | | 11: | fcgck/$2^9$ | Reserved | Reserved |

*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>.

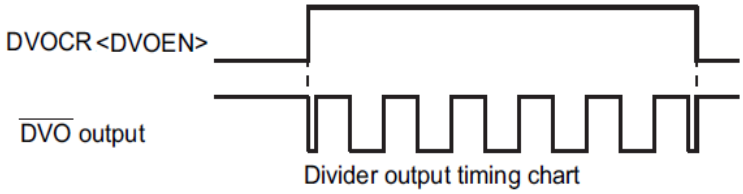



Figure 10.6 Divider Output Timing

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)

## 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

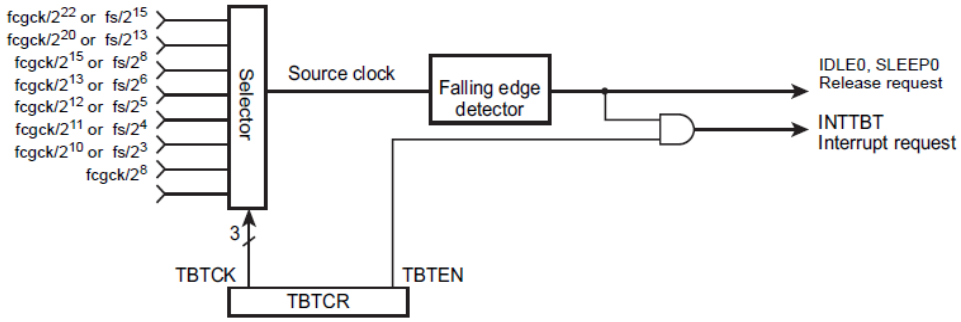



Figure 10.7 Time Base Timer Configuration

### 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<br>1: Enable | | | |
| TBTCK | Select the time base timer interrupt frequency<br>Unit: [Hz] | | Normal 1/2, IDLE 1/2 mode | | SLOW 1/2 mode<br>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>.

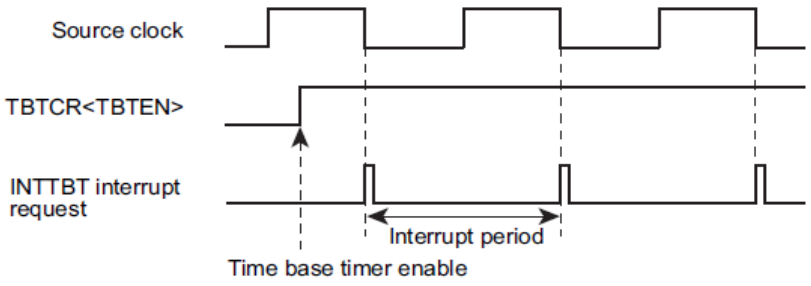



Figure 10.8 Time Base Timer Interrupt

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".

| 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)

## 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

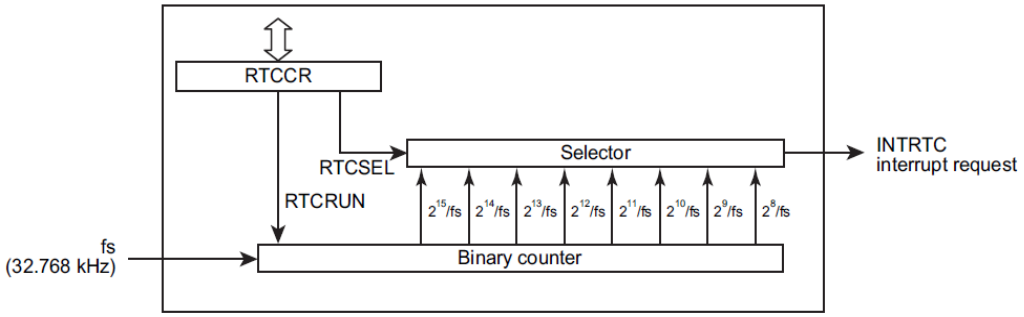



Figure 10.9Real Time Clock

### 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 | LCDEN | - | RTCEN | - | - | - | - | SIO0EN |
| Read/Write | R/W | R | R/W | R | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| LCDEN | LCD control | 0: Disable<br>1: Enable |
| RTCEN | RTC control | 0: Disable<br>1: Enable |
| SIO0EN | SIO0 control | 0: Disable<br>1: Enable |

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 | Selectstheinterruptgenerationinterval | 000:$2^{15}$/fs(1.000[s]@fs=32.768kHz)<br>001:$2^{14}$/fs(0.500[s]@fs=32.768kHz)<br>010:$2^{13}$/fs(0.250[s]@fs=32.768kHz)<br>011: $2^{12}$/fs (125.0[ms]@fs=32.768kHz)<br>100:$2^{11}$/fs(62.50[ms]@fs=32.768kHz)<br>101:$2^{10}$/fs(31.25[ms]@fs=32.768kHz)<br>110: $2^{9}$/fs(15.62[ms]@fs=32.768kHz)<br>111:$2^{8}$/fs(7.81[ms]@fs=32.768kHz) |
| RTCRUN | Enables/disablestherealtimeclockoperation | 0: Disable<br>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 10-bit Timer Counters

MQ6812/MQ6821 contains 6 channels of high-performance 10-bit timer counters 00, 01, 02,03,04 and 05 (TCQ0). Each timer can be used for time measurement and pulse output with a prescribed width.

Please be noted.:

1. Only TCQ00,TCQ02,TCQ04 support double buffer register.
2. PPG pulse output do not support duty as 0% and 100%.

This chapter describes 2 channels of 10-bit timer counters 00 and 01. For 10-bit timer counters 02, 03, and 04,05, replace the SFR addresses and pin names as shown in Table 10.4 and Table 10.5.

**Only TCQ00 (02,04) support double buffer ; TCQ01(03,05) do not support double buffer.**

| | T0xREGL (Address) | T0xREGH (Address) | T0xPWML (Address) | T0xPWMH (Address) | T0xMOD (Address) | T0xxCR (Address) | Low power consumption register |
|---|---|---|---|---|---|---|---|
| Timer counter 00 | T00REGL (0x0F9C) | T00REGH (0x0F9D) | T00PWML (0x0FA0) | T00PWMH (0x0FA1) | T00MOD (0x002A) | T001CR (0x002C) | POFFCR0 <TC001EN> |
| Timer counter 01 | T01REGL (0x0F9E) | T01REGH (0x0F9F) | T01PWML (0x0FA2) | T01PWMH (0x0FA3) | T01MOD (0x002B) | | |
| Timer counter 02 | T02REGL (0x0FA4) | T02REGH (0x0FA5) | T02PWML (0x0FA8) | T02PWMH (0x0FA9) | T02MOD (0x0F8C) | T023CR (0x0F8E) | POFFCR0 <TC023EN> |
| Timer counter 03 | T03REGL (0x0FA6) | T03REGH (0x0FA7) | T03PWML (0x0FAA) | T03PWMH (0x0FAB) | T03MOD (0x0F8D) | | |
| Timer counter 04 | T04REGL (0x0FB0) | T04REGH (0x0FB1) | T04PWML (0x0FB4) | T04PWMH (0x0FB5) | T04MOD (0x0FB8) | T045CR (0x0FBA) | POFFCR0 <TC045EN> |
| Timer counter 05 | T05REGL (0x0FB2) | T05REGH (0x0FB3) | T05PWML (0x0FB6) | T05PWMH (0x0FB7) | T05MOD (0x0FB9) | | |

Table 10.4 SFR Address Assignment

| | PPG Output Pin |
|---|---|
| Timer counter 00 | PPG00B pin |
| Timer counter 01 | PPG01B pin |
| Timer counter 02 | PPG02B pin |
| Timer counter 03 | PPG03B pin |
| Timer counter 04 | PPG04B pin |
| Timer counter 05 | PPG0%B pin |

Table 10.5 Pin Names

## 10.5.1 Control

### 10.5.1.1 Timer Counter 00

The timer counter 00 is controlled by the timer counter 00 mode register (T00MOD) and four 10-bit timer registers (T00REGL/T00REGH and T00PWML/T00PWMH).

**Timer Register 00(L)**

| T00REGL (0x0F9C) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | T00REGL | | | | | | | |
| 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(H)**

| T00REGH (0x0F9D) | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | - | T00REGH | |
| Read/Write | R | R | R | R | R | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Timer Register 00(H)**

| T00PWML (0x0FA0) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | T00PWML | | | | | | | |
| 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(L)**

| T00PWMH (0x0FA1) | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | - | T00PWMH | |
| Read/Write | R | R | R | R | R | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Note): Because PPG pulse output do not support duty as 0% and 100%. Please set T00PWM as 0 < T00PWM < T00REG.*

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 |

| | | | | | |
|---|---|---|---|---|---|
| TFF0 | Timer F/F0 control | 0: Clear<br>1: Set | | | |
| DBE0 | Double buffer control | 0: Disable the double buffer<br>1: Enable the double buffer | | | |
| TCK0 | Operation clock selection | | Normal 1/2, IDLE 1/2 mode | | SLOW 1/2 mode<br>SLEEP 1 mode |
| | | | SYSCR1 <DV9CK>=0 | SYSCR1 <DV9CK>=1 | |
| | | 000: | $fcgck/2^4$ | $fcgck/2^4$ | |
| | | 001: | $fcgck/2^2$ | $fcgck/2^2$ | |
| | | 010: | fcgck/2 | fcgck/2 | |
| | | 011: | fcgck | fcgck | $fs/2^2$ |
| | | 100: | $fcgck/2^4$ | $fcgck/2^4$ | - |
| | | 101: | $fcgck/2^2$ | $fcgck/2^2$ | - |
| | | 110: | fcgck/2 | fcgck/2 | - |
| | | 111: | fcgck | fcgck | $fs/2^2$ |
| | | Other | System reserved | | |
| EIN0 | Selection for using external source clock | 0: Select the internal clock as the source clock.<br>1: Select an external clock a s the source clock. (the falling edge of the TC00 pin) | | | |
| TCM0 | Operation mode selection | 00: | 10-bit timer / event counter modes | | |
| | | 01: | 10-bit timer / event counter modes | | |
| | | 10: | 10-bit programmable pulse generate (PPG) mode | | |
| | | 11: | 10-bit programmable pulse generate (PPG) mode | | |

*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 10-bit timer/event modes, the TFF0 setting is invalid. In this mode, when the 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.*

### 10.5.1.2 Timer Counter 01

Timer counter 01 is controlled by timer counter 01 mode register (T01MOD) and four 10-bit timer registers (T01REGL/ T01REGH and T01PWML/T01PWMH).

Timer Register 01(L)

| T01REGL (0x0F9E) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | T01REGL | | | | | | | |
| 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(H)**

| T01REGH (0x0F9F) | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | - | T01REGH | |
| Read/Write | R | R | R | R | R | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Timer Register 01(L)**

| T01PWML (0x0FA2) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | T01PWML | | | | | | | |
| 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(H)**

| T01PWMH (0x0FA3) | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | - | - | - | T01PWMH | |
| Read/Write | R | R | R | R | R | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Note): Because PPG pulse output do not support duty as 0% and 100%. Please set T00PWM as 0 < T00PWM < T00REG.*

**Timer Counter 01 Mode Register**

| T01MOD (0x002B) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | TFF1 | - (note 5) | TCK1 | | | EIN1 | TCM1 | |
| Read/Write | R/W | R | R/W | R/W | R/W | R | R/W | R/W |
| After reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | | | |
|---|---|---|---|---|---|
| TFF1 | Timer F/F1 control | 0: Clear<br>1: Set | | | |
| TCK1 | Operation clock selection | | Normal 1/2, IDLE 1/2 mode | | SLOW 1/2 mode<br>SLEEP 1 mode |
| | | | SYSCR1 <DV9CK>=0 | SYSCR1 <DV9CK>=1 | |
| | | 100: | fcgck/$2^4$ | fcgck/$2^4$ | - |
| | | 101: | fcgck/$2^2$ | fcgck/$2^2$ | - |
| | | 110: | fcgck/2 | fcgck/2 | - |
| | | 111: | fcgck | fcgck | fs/$2^2$ |
| | | other | System reserved | | |
| EIN1 | Selection for using external source clock | 0: Select the internal clock as the source clock.<br>1: Select an external clock as the source clock.(the falling edge of the TC01 pin) | | | |
| TCM1 | Operation mode selection | 00: | 10-bit timer/eventcountermodes | | |
| | | 01: | 10-bit timer/eventcountermodes | | |
| | | 10: | 10-bit programmable pulse generate (PPG) mode | | |
| | | 11: | 10-bit programmable pulse generate (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 10-bit timer/event modes, the TFF1 setting is invalid. In this mode, when the 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.*

*Note 5): Only TCQ00 (02,04) support double buffer ; TCQ01(03,05) do not support double buffer.*

### 10.5.1.3 Common to Timer Counters 00 and 01

Timer counters 00 and 01(02 and 03/ 04 and 05) 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 | - | TC045EN | TC023EN | TC001EN | - | - | - | TCA0EN |
| Read/Write | R | R/W | R/W | R/W | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| TC045EN | TCQ04, 05 control | 0: Disable<br>1: Enable |
| TC023EN | TCQ02, 03 control | 0: Disable<br>1: Enable |
| TC001EN | TCQ00, 01 control | 0: Disable<br>1: Enable |
| TCA0EN | TCA0 control | 0: Disable<br>1: Enable |

Timer 00 and 01 Control Register

| T001CR (0x002C) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | - | - | - | COMPSL | OUTAND | - | T01RUN | T00RUN |
| Read/Write | R | R | R | R/W | R/W | R | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| COMPSL | Timer 00/01 complementary output control | 0: Output the timer 00 output from the PPG0B pins and the timer 01 output from the PPG1B pins.<br><br>1: Output the timer 00 copmlemetary output from the PPG01B pins and the timer 00 output from the PPG00B pins. |
| OUTAND | Timer 00 and 01 output control | 0: Output the timer 00 output from the PPG00B pins and the timer 01 output from the PPG01B pins.<br><br>1: Output a pulse that is a logical ANDed product of the outputs of timer 00 and 01 from the PPG01B pins. Output the timer 00 output from the PPG00B pins |
| T01RUN | Timer 01 control (note 5) | 0: Stop and clear the timer<br>1: Start |

| | | |
|---|---|---|
| T00RUN | Timer 00 control | 0: Stop and clear the timer<br>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 timers 00 and 01 after STOP mode is released.*

*Note 2): When a read instruction is executed on T001CR, bits 7~5 and Bit2 are read as "0".*

*Note 3): When OUTAND is "1", output is obtained from the PPG1B pins only. There is no timer output to the PPG0B pins. If the PPG0B pins are set as the function output pins in the port setting, the pins always output "H".*

*Note 4): OUTAND and COMPSL 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 COMPSL. OUTAND and COMPSL can be changed at the same time as TC01RUN and TC00RUN are changed from "0" to "1".*

*Note 5): When T01RUN is "1" (start the timer 01), T00RUN should be set as "1".*

### 10.5.1.4 Operation Modes and Usable Source Clocks

| T00MOD <TCK0> | | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|
| Operation mode | | fcgck/$2^4$ | fcgck/$2^2$ | fcgck/2 | fcgck 或 fs/$2^2$ |
| 10-bit timer modes | 10-bit timer | O | O | O | O |
| | 10-bit PPG | O | O | O | O |

Table 10.6 Operation Modes and Usable Source Clocks (NORMAL1/2 and IDLE1/2 Modes)

The operation modes of the 8-bit timers and the usable source clocks are listed below.

*Note 1): ○: Usable, -: Unusable*

*Note 2): 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.*

| T00MOD <TCK0> | | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|
| Operation mode | | – | – | – | fs/$2^2$ |
| 10-bit timer modes | 10-bit timer | – | – | – | O |
| | 10-bit PPG | – | – | – | O |

Table 10.7 Operation Modes and Usable Source Clocks (SLOW1/2 and SLEEP1 Modes)

*Note 1): ○: Usable, -: Unusable*

### 10.5.2 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.

### 10.5.3 Function

The 10-bit modes TC00 and TC01 include three operation modes: 10-bit timer mode, 10-bit programmable pulse generated output (PPG) mode and 10-bit complementary output mode.
(The same to TC02,TC03,TC04 and TC05)

#### 10.5.3.1 10-bit Timer Mode

In the 10-bit timer mode, the up counter counts up using the internal clock, and interrupts can be generated regularly at specified times. The operation of TCQ00 is described below, and the same applies to the operation of TCQ01, TCQ02 ,TCQ03,TCQ04 and TCQ05. (Replace TCQ00- by TCQ0x, x=1~5).

**(a) Setting**

TCQ00 is put into the 10-bit timer mode by setting T00MOD <TCM0> to "00" or "01". Select the source clock at T00MOD <TCK0>. Set the count value to be used for the match detection as an 10-bit value at the timer register T00REGL/ T00REGH.

Set T00MOD <DBE0> to "1" to use the double buffer. Only TCQ00,TCQ02,TCQ04 can use double buffer. TCQ01,TCQ03,TCQ05 cannot use 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 10-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the

T00REGL/TOOREGH 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 T00REGL/T00REGH 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 T00REGL/T00REGH during the timer operation, the set value is initially stored in the double buffer, and T00REGL/T00REGH is not immediately updated. T00REGL/T00REGH 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 T00REGL/T00REGH. Subsequently, the match detection is executed using a new set value.

When a write instruction is executed on T00REGL/T00REGH while the timer is stopped, the set value is immediately stored in both the double buffer and T00REGL/T00REGH.

2. When the Double Buffer is Disabled

When a write instruction is executed on T00REGL/T00REGH during the timer operation, the set value is immediately stored in T00REGL/T00REGH. Subsequently, the match detection is executed using a new set value.

If the value set to T00REGL/T00REGH 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 T00REGL/T00REGH is equal to the up counter value, the match detection is executed immediately after data is written into T00REGL/T00REGH. Therefore, the interrupt request interval may not be an integral multiple of the source clock (Figure 10.11). Under TCQ01,TCQ03,TCQ05 operating, please avoid to change the setting(detail please refer to appendix D). Otherwise, Or use TCQ00,TCQ02,TCQ04 to enable the double buffer.

When a write instruction is executed on T00REGL/T00REGH while the timer is stopped, the set value is immediately stored in T00REGL/T00REGH.

When a read instruction is executed on T00REGL/T00REGH, the last value written into T00REGL/T00REGH is read out, regardless of the T00MOD <DBE0> setting.

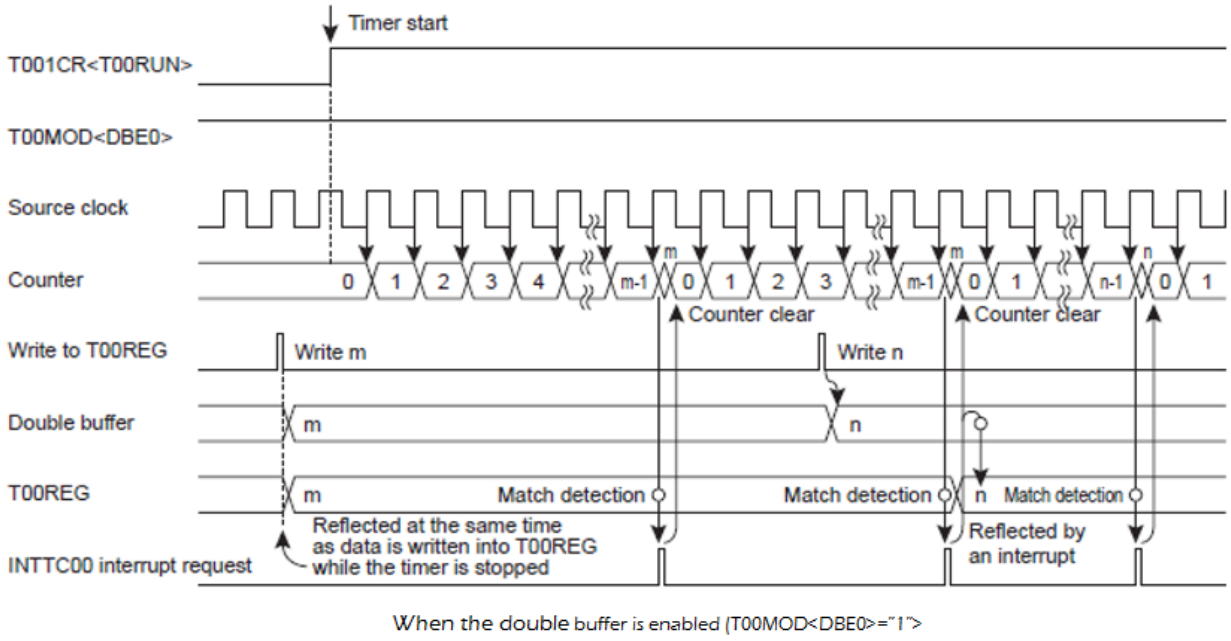
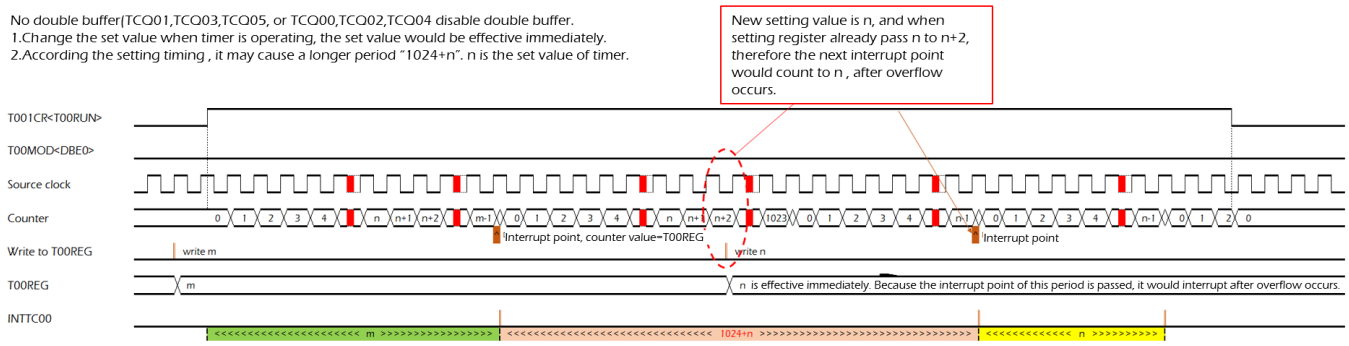
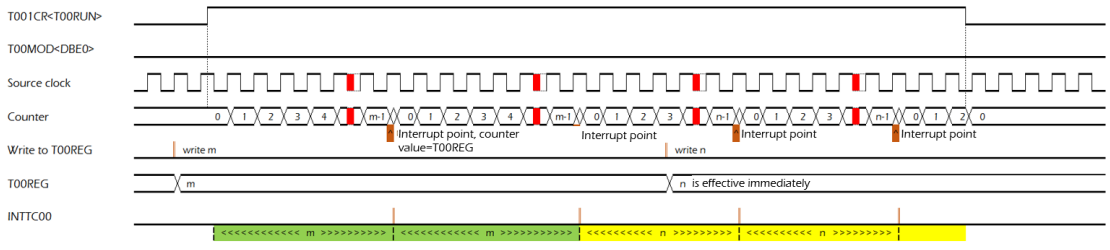



Figure 10.10 Timer Mode Timing Chart (T00MOD<DBE0>="1")

Note: T00REG=T00REGH/L




No double buffer or double buffer disable (T00MOD<DBE0>=0). Set the new value when the set value(n) of this period is passed..




No double buffer or double buffer disable (T00MOD<DBE0>=0). Set the new value when the set value(n) of this period is not over.

Figure 10.11 Timer Mode Timing Chart(T00MOD<DBE0>="0")

Note: T00REG=T00REGH/L

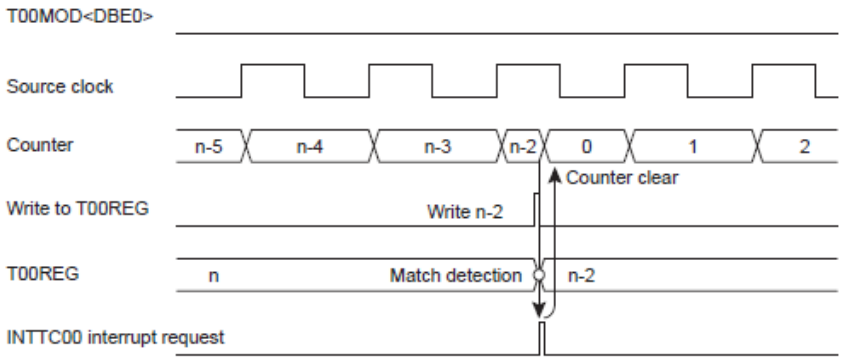



Figure 10.12 Operation When T00REG and the Up Counter Have the Same Value
Note: T00REG=T00REGH/L

| T00MOD <TCK0> | Source clock (Hz) | | | Resolution | | Maximum time setting | |
|---|---|---|---|---|---|---|---|
| | NORMAL 1/2 or IDLE 1/2 mode | | SLOW1/2 or SLEEP1 mode | fcgck=16MHz | fs=32.768KHz | fcgck=16MHz | fs=32.768KHz |
| | SYSCR1 <DV9CK>="0" | SYSCR1 <DV9CK>="1" | | | | | |
| 100 | fcgck/$2^4$ | fcgck/$2^4$ | – | 1us | – | 1.0ms | – |
| 101 | fcgck/$2^2$ | fcgck/$2^2$ | – | 250ns | – | 255.8us | – |
| 110 | fcgck/2 | fcgck/2 | – | 125ns | – | 127.9us | – |
| 111 | fcgck | fcgck | fs/$2^2$ | 62.5ns | 122.1us | 63.9us | 124.9ms |

Table 10.8 10-bit Timer Mode Resolution and Maximum Time Setting

### 10.5.3.2 10-bit Programmable Pulse Generate (PPG) Output Mode

In the 10-bit PPG mode, the pulses with arbitrary duty and cycle are output by using the T00REGL/ T00REGH and T00PWML/ T00PWMH registers.

By setting the T001CR <OUTAND> register, a pulse that is a logical ANDed product of the TCQ00 and TCQ01 outputs can be output to the TCQ01 pin. This function facilitates the generation of remote-controlled waveforms, for example.

The operation of TCQ00 is described below, and the same applies to the operation of TCQ01, and similarly, TCQ02~ TCQ05.

(a) Setting

TCQ00 is put into the 10-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 T00PWML/ T00PWMH and the cycle width at T00REG.

Set T00MOD <DBE0> to "1" to use the double buffer.Only TCQ00,TCQ02,TCQ04 can 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.

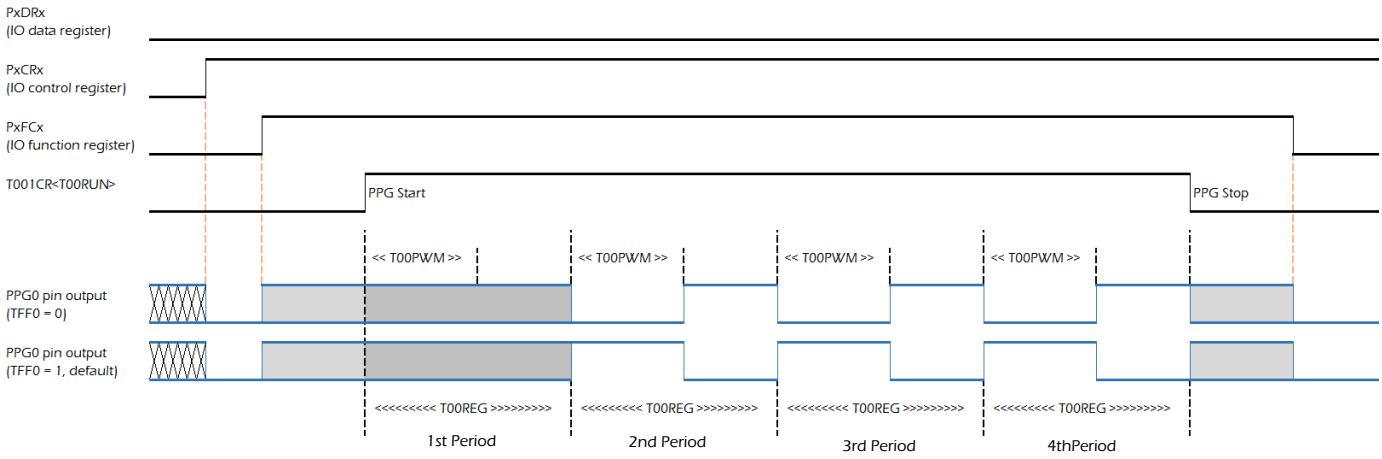



**Figure 10.16 PPG00B Pulse Output**

Note1): PPG output is valid from 2nd period. The level cannot be set by register, please set the pin as I/O, after PPG is stop.(Detail please refer to appendix D)
Note2): T00REG=T00REGH/L , T00PWM=T00PWMH/L

Set the initial state of the PPG0)B pin at T00MOD <TFF0>. Setting T00MOD <TFF0> to "0" selects the "L" level as the initial state of the PPG00B pin. Setting T00MOD <TFF0> to "1" selects the "H" level as the initial state of the PPG00B pin. If the PPG00B 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 PPG00B pin.

Setting the T001CR <OUTAND> bit to "1" allows the PPG00B pin to output a pulse that is a logical ANDed product of the TCQ00 and TCQ01 outputs.

### (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 PPG00B pin changes from the "L" to "H" level. When T00MOD <TFF0> is "1", the PPG00B 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 PPG00B 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". Only TCQ00,TCQ02,TCQ04 can use double buffer.

#### 1. When the Double Buffer is Enabled

When a write instruction is executed on T00PWML/ T00PWMH(T00REGL/ T00REGH) during the timer operation, the set value is first stored in the double buffer, and T00PWML/ T00PWMH(T00REGL/ T00REGH) is not updated immediately. T00PWML/ T00PWMH(T00REGL/ T00REGH) 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 T00PWML/T00PWMH(T00REGL/T00REGH). Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on T00PWML/T00PWMH(T00REGL/T00REGH), the value in the double buffer (the last set value) is read out, not the T00PWML/T00PWMH(T00REGL/T00REGH) value (the currently effective value).

When a write instruction is executed on T00PWML/T00PWMH(T00REGL/T00REGH) while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWML/T00PWMH(T00REGL/T00REGH).

#### 2. When the Double Buffer is Disabled

When a write instruction is executed on T00PWML/T00PWMH(T00REGL/T00REGH) during the timer operation, the set value is immediately stored in T00PWML/T00PWMH(T00REGL/T00REGH). Subsequently, the match detection is executed using a new set value.

If the value set to T00PWML/T00PWMH(T00REGL/T00REGH) is smaller than the up

counter value, the PPG00B pin is not reversed until the up counter overflows and a match detection is executed using a new set value(Figure 10.15). If the value set to T00PWML /T00PWMH (T00REGL/T00REGH)is equal to the up counter value, the match detection is executed immediately after data is written into T00PWML /T00PWMH (T00REGL /T00REGH). Therefore, the timing of changing the PPG00B pin may not be an integral multiple of the source clock (Figure 10.16). When use TCQ01, TCQ03, TCQ05, do not change the set value under operation(detail please refer to appendix D). Or enable double buffer by PPG00B, PPG02B ,PPG04B.

When a write instruction is executed on T00PWML /T00PWMH (T00REGL/T00REGH) while the timer is stopped, the set value is immediately stored in T00PWML /T00PWMH (T00REGL/T00REGH)

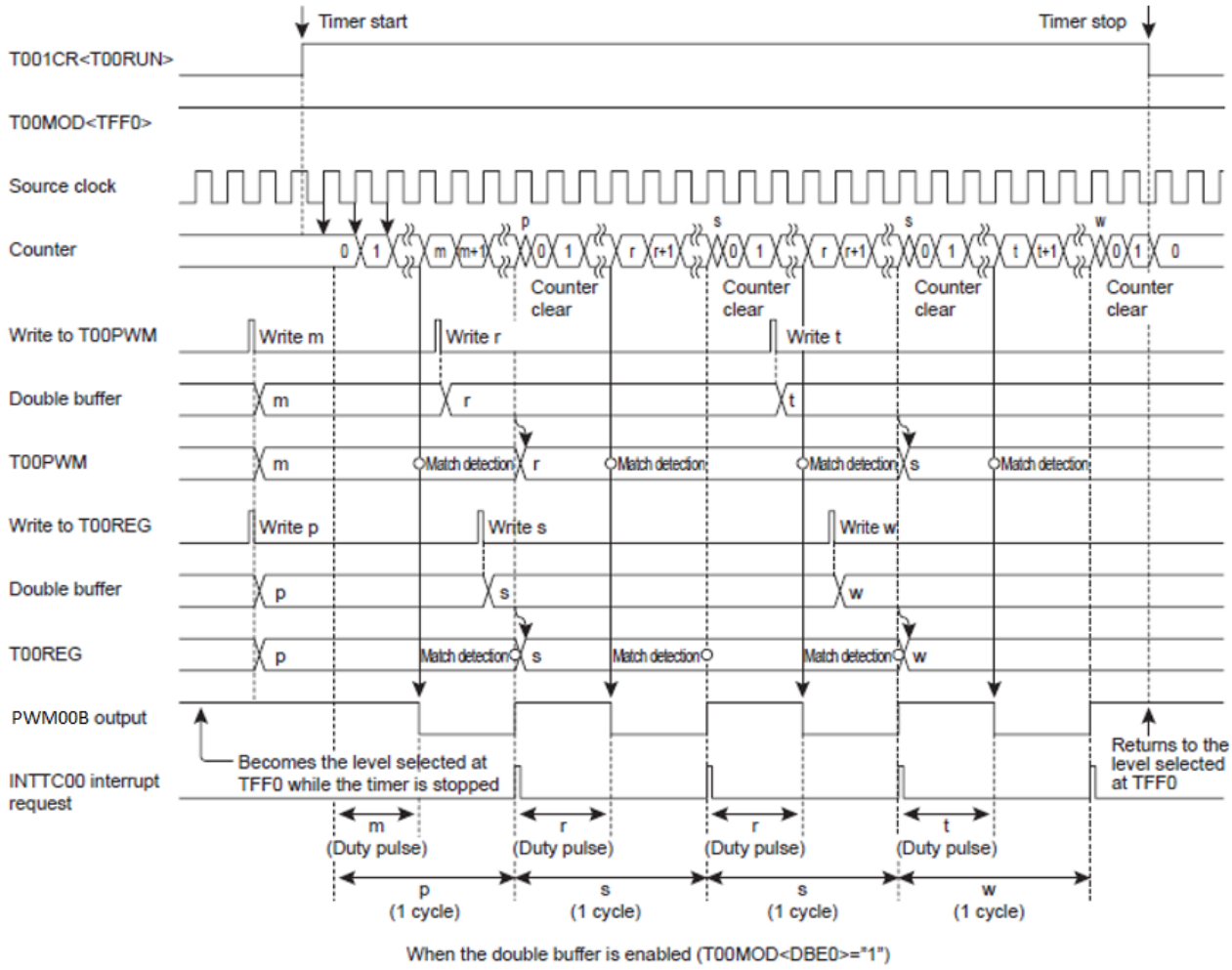
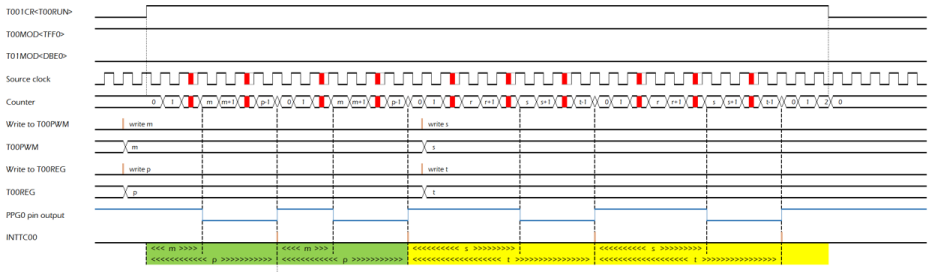



Figure 10.14 10-bit PPG Mode Timing Chart (T00MOD<DBE0>="1")

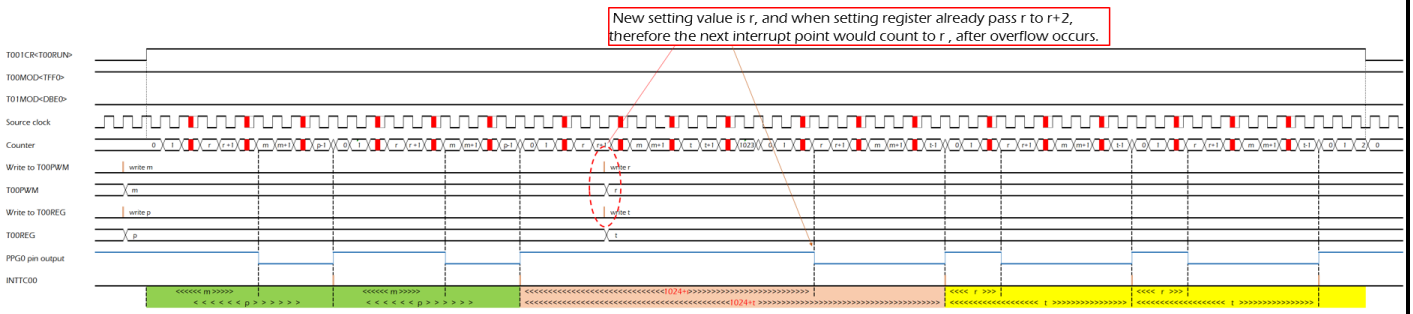
Note1): Only TCQ00,TCQ02,TCQ04 can use double buffer.

Note2): T00REG=T00REGH/L , T00PWM=T00PWMH/L

No double buffer(TCQ01,TCQ03,TCQ05, or TCQ00,TCQ02,TCQ04 disable double buffer.
1.Change the set value when PPG is operating, the set value would be effective immediately.
2.According the setting timing , it may cause a longer period "1024+n". n is the set value of timer.




No double buffer or double buffer disable (T00MOD<DBE0>=0). Set the new value when the set value(r) of this period is not over.




No double buffer or double buffer disable (T00MOD<DBE0>=0). Set the new value when the set value(r) of this period is passed.

**Figure 10.15 10-bit PPG Mode Timing Chart (T00MOD<DBE0>="0")**
Note1): Only TCQ00,TCQ02,TCQ04 can use double buffer.
Note2): T00REG=T00REGH/L , T00PWM=T00PWMH/L

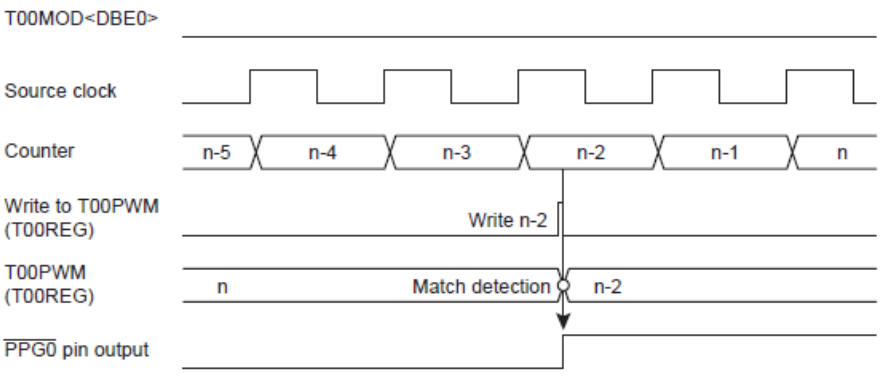



**Figure 10.16 Operation When T00PWML/H (T00REGL/H)and the Up Counter Have the Same Value**
Note): T00REG=T00REGH/L , T00PWM=T00PWMH/L

### 10.5.3.3 10-bit Complementary Programmable Pulse Generate (PPG) Output Mode

In the 10-bit complementary PPG output mode, TCQ00 and TCQ01 would output the PPG complementary signals, the waveform are opposite and non-overlap. Output the timer 00 copmlemetary output from the PPG01B pins and the timer 00 output from the PPG00B pins.

(a) Setting

TCQ00 ahd TCQ01 are put into the 10-bit PPG complementary output mode by setting T01CR <T00RUN> and <T01RUN> to"1".

Set T00MOD<TCM0> to "11" or "10" to select 10-bit PPG complementary output mode. To use the internal clock as the source clock: Set T00MOD <EIN0> to "0" and select the clock at T00MOD <TCK0>.

*Note: In 10-bit PPG complementary output mode, the setting value is the same with T00MOD and T01MOD (x=0,1) ,except <TFFx>.*

When a write instruction is executed on T00PWML/ T00PWMH(T00REGL/ T00REGH) during operation, the set value is first stored in the double buffer, and T00PWML/ T00PWMH(T00REGL/ T00REGH) is not updated immediately. T00PWML/ T00PWMH(T00REGL/ T00REGH) 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 T00PWML/T00PWMH(T00REGL/T00REGH). Subsequently, the match detection is executed using a new set value.

Set the 10-bit counter value correspond to period by T00REGL/T00REGH. Set the 10-bit counter value correspond to operating pulse by T00PWML/T00PWMH. The 10-bit counter value from T00REGL/H show as T00REGL+H; The 10-bit counter value from T00PWML/H show as T00PWML+H. When write the set value to T00PWMH, the set value would be correspond to double buffer or T00REGL+H and T00PWML+H. Set the value of T00REGL/H and T00PWML first, once write the set value to T00PWMH, the setting of all registers would beome effective at the same time.

Set PPG00B initial status by T00MOD<TFF0>. Set T00MOD<TFF)> to "0", the initial statsu of PPG00B is Low.

Set the initial state of the PPG00B pin at T00MOD <TFF0>. Setting T00MOD <TFF0> to "0" selects the "L" level as the initial state of the PPG00B pin. Setting T00MOD <TFF0> to "1" selects the "H" level as the initial state of the PPG00B pin. In the same way, set the initial state of the PPG01B pin at T01MOD <TFF1>. Setting T01MOD <TFF1> to "0" selects the "L" level as the initial state of the PPG01B pin. Setting T01MOD <TFF1> to "1" selects the "H" level as the initial state of the PPG01B 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 internal up counter value and the value set to T01PWML+H is detected, the output of the PPG01B pin is reversed. When T01MOD <TFF1>

is "0", the PPG01B pin changes from the "L" to "H" level. When T01MOD <TFF1> is "1", the PPG01B pin changes from the "H" to "L" level. At this time, an INTTC01 interrupt request is generated.

Subsequently, the up counter continues counting up. When a match between the up counter value and T01REGL+H is detected, the output of the PPG01B pin is reversed again. When T01MOD <TFF1> is "0", the PPG01B pin changes from the "H" to "L" level. When T00MOD <TFF1> is "1", the PPG01B pin changes from the "L" to "H" level. At this time, an INTTC01 interrupt request is generated.

When T001CR <T01RUN> is set to "0" during the operation, the up counter is stopped and cleared to "0x0000". The PPG01B pin returns to the level selected at T01MOD <TFF1>.

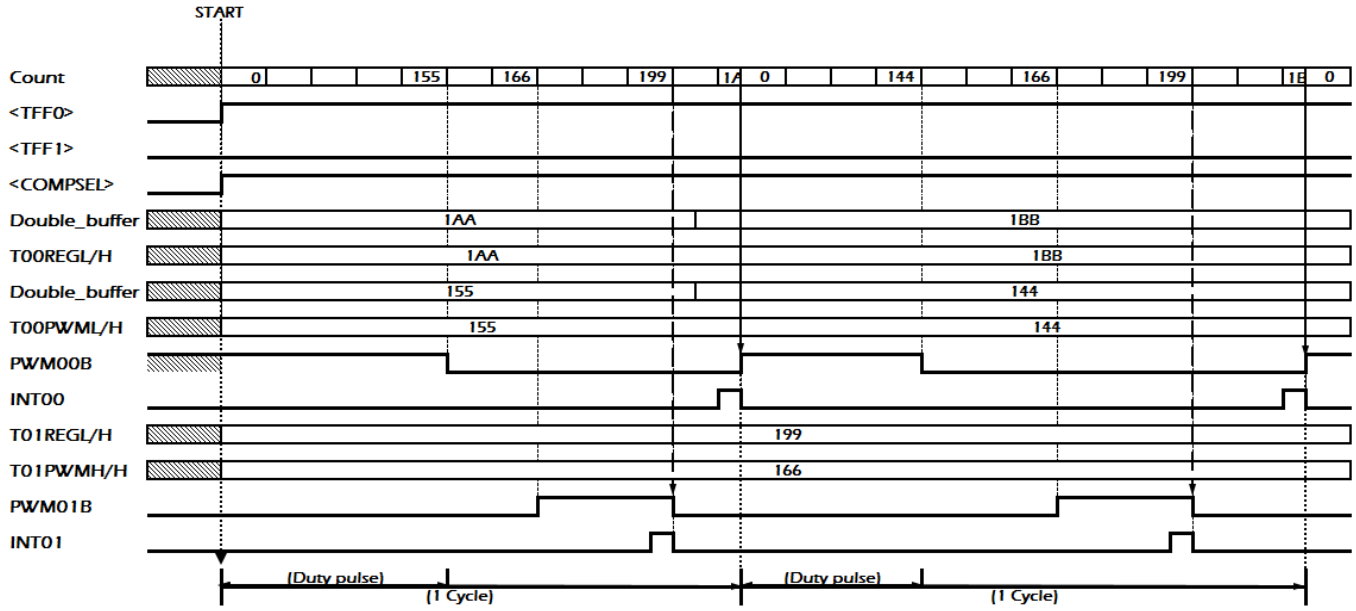



Figure 10.17 10-bit complementary PPG Mode Timing Chart(T00MOD<DBE0>="0")

### (c) Double Buffer

The double buffer can be used for T00PWML/H and T00REGL/H by setting T00MOD <DBE0>. The double buffer is disabled by setting T00MOD <DBE0> to "0" or enabled by setting T00MOD <DBE0> to "1". Only TCO00,TCO02,TCO04 can use double buffer.

#### 1. When the Double Buffer is Enabled

When a write instruction is executed on T00PWML/ T00PWMH first ,then excuted on (T00REGL/ T00REGH) during the timer operation, the set value is first stored in the double buffer, and T00PWML+H and T00REGL+H) is not updated immediately. T00PWML+H and T00REGL+H 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 T00PWML+H and T00REGL+H. Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on T00PWML/T00PWMH first , then excuted on T00REGL/ T00REGH, the value in the double buffer (the last set value) is read out, not the T00PWML/T00PWMH(T00REGL/T00REGH) value (the currently effective value).

When a write instruction is executed on T00PWML/T00PWMH first , then excuted on T00REGL/ T00REGH, while the timer is stopped, the set value is immediately stored in both the double buffer ,T00PWML+H and T00REGL+H. Only TCQ00,TCQ02,TCQ04 can use double buffer.

2. When the Double Buffer is Disabled

When a write instruction is executed on T00PWML/T00PWMH first ,then executed on T00REGL/T00REGH during the timer operation, the set value is immediately stored in T00PWML+H and T00REGL+H. Subsequently, the match detection is executed using a new set value.

If the value set to T00PWML+H or T00REGL+H) is smaller than the up counter value, the PPG00B 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 T00PWML+H or T00REGL+H is equal to the up counter value, the match detection is executed immediately after data is written into T00PWML+H and T00REGL+H. Therefore, the timing of changing the PPG00B pin may not be an integral multiple of the source clock. When use TCQ01, TCQ03, TCQ05, do not change the set value under operation(detail please refer to appendix D). Or enable double buffer by PPG00B, PPG02B ,PPG04B.

When a write instruction is executed on T00REGL, T00REGH, and T00PWML, then executed on T00PWMH while the timer is stopped, the set value is immediately stored in T00PWML+H and T00REGL+T00REGH.

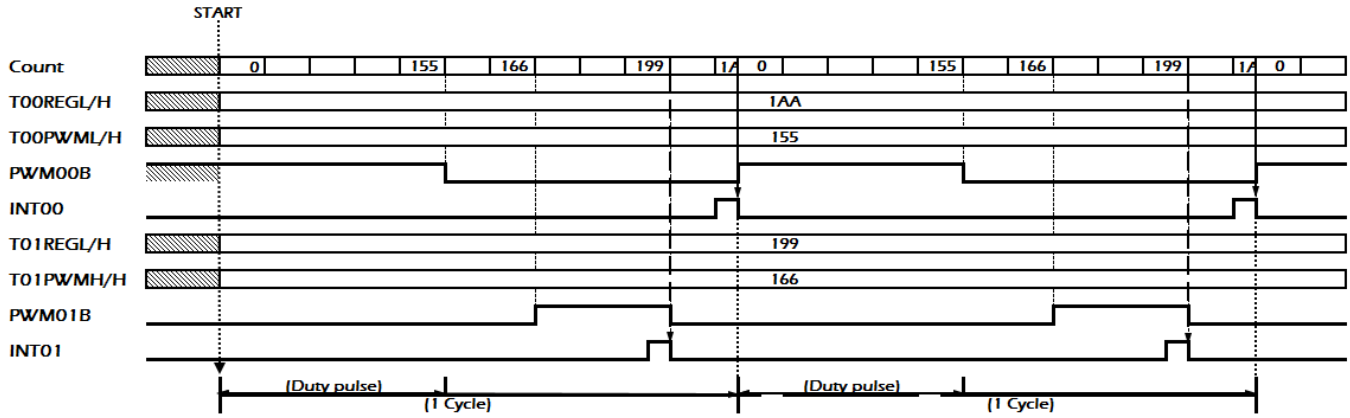



Figure 10.18 10-bit complementary PPG Mode Timing Chart (T00MOD<DBE0>="1")

Note1): Only TCQ00,TCQ02,TCQ04 can use double buffer.

## 10.6 16-bit Timer Counter (TCA)

MQ6812/MQ6821 contains 1 channels of high-performance 16-bit timer counters (TCA0).

This chapter describes the 16-bit timer counter TCA0.

| | 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> |

Table 10.17 SFR Address Assignment

| | Timer Input Pin | PPG Output Pin |
|---|---|---|
| Timer counter A0 | TCA0 pin | PPGA0B pin |

Table 10.18 Pin Names

### 10.6.1 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 | - | TC045EN | TC023EN | TC001EN | - | - | - | TCA0EN |
| Read/Write | R | R/W | R/W | R/W | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| TC045EN | TC045 enable control | 0: Disable<br>1: Enable |
| TC023EN | TC023 enable control | 0: Disable<br>1: Enable |
| TC001EN | TC001 enable control | 0: Disable<br>1: Enable |
| TCA0EN | TCA0 enable control | 0: Disable<br>1: Enable |

Timer Counter A0 Mode Register

| TA0MOD (0x0031) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | TA0DB1 | 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 |

| | | | | | |
|---|---|---|---|---|---|
| TA0DBE | Double buffer control | 0: Disable the double buffer<br>1: Enable the double buffer | | | |
| TA0TED | External trigger input selection | 0: Rising edge / H Level<br>1: Falling edge / L Level | | | |
| TA0MCAP | Pulse width measurement mode control | 0: Double edge capture<br>1: Single edge capture | | | |
| TA0METT | External trigger timer mode control | 0: Trigger start<br>1: Trigger start and stop | | | |
| TA0CK | Timer counter 1 source clock selection | | Normal 1/2, IDLE 1/2 mode | | SLOW 1/2 mode SLEEP 1 mode |
| | | | SYSCR1 <DV9CK>=0 | SYSCR1 <DV9CK>=1 | |
| | | 00: | fcgck/$2^{10}$ | fs/$2^3$ | fs/$2^3$ |
| | | 01: | fcgck/$2^6$ | fcgck/$2^6$ | - |
| | | 10: | fcgck/$2^2$ | fcgck/$2^2$ | - |
| | | 11: | fcgck/2 | fcgck/2 | - |
| TA0M | Timer counter 1 operation mode selection | 000: | Timer mode | | |
| | | 001: | Timer mode | | |
| | | 010: | Event counter mode | | |
| | | 011: | PPG output mode (Software start) | | |
| | | 100: | External trigger time mode | | |
| | | 101: | Window mode | | |
| | | 110: | Pulse width measurement mode | | |
| | | 111: | Reserved | | |

*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 |

| | | | | |
|---|---|---|---|---|
| TA0OVE | Overflow interrupt control | 0: Generate no INTTCA0 interrupt request when the counter overflow occurs.<br>1: Generate an INTTCA0 interrupt request when the counter overflow occurs. | | |
| TA0TFF | Timer F/F control | 0: Clear<br>1: Set | | |
| TA0MCAP | Pulse width measurement mode control | 0: Double edge capture<br>1: Single edge capture | | |
| TA0NC | Noise canceller sampling interval setting | | Normal 1/2 or IDLE 1/2 mode | SLOW 1/2 mode SLEEP 1 mode |

| | | | | |
|---|---|---|---|---|
| | | 00: | No noise canceller | No noise canceller |
| | | 01: | fcgck/2 | - |
| | | 10: | fcgck/$2^2$ | - |
| | | 11: | fcgck/$2^8$ | fs/2 |
| TA0ACAP | Auto capture function | 0: Disable the auto capture<br>1 Enable the auto capture | | |
| TA0MPPG | PPG output control | 0: Continuous<br>1: One-shot | | |
| TA0S | Timer counter A start control | 0: Stop and counter clear<br>1: Start | | |

*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.<br>1: At least an overflow has occurred. |
| TA0CPFA | Capture completion flag A | 0: No capture operation has been executed.<br>1: At least a pulse width capture has been executed in the double-edge capture |
| TA0CPFB | Capture completion flag B | 0: Double edge capture<br>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 |

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.*

### 10.6.2 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.

### 10.6.3 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.

#### 10.6.3.1 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>.

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 10.11 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

MQ6812/MQ6821 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 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 MQ6812/MQ6821, the double buffer can be used by setting TA0MOD <TA0DBE>. Setting TA0CR TA0MOD <TA0DBE> to "0" disables the double buffer. Setting TA0MOD <TA0DBE> to "1" enables the double buffer.

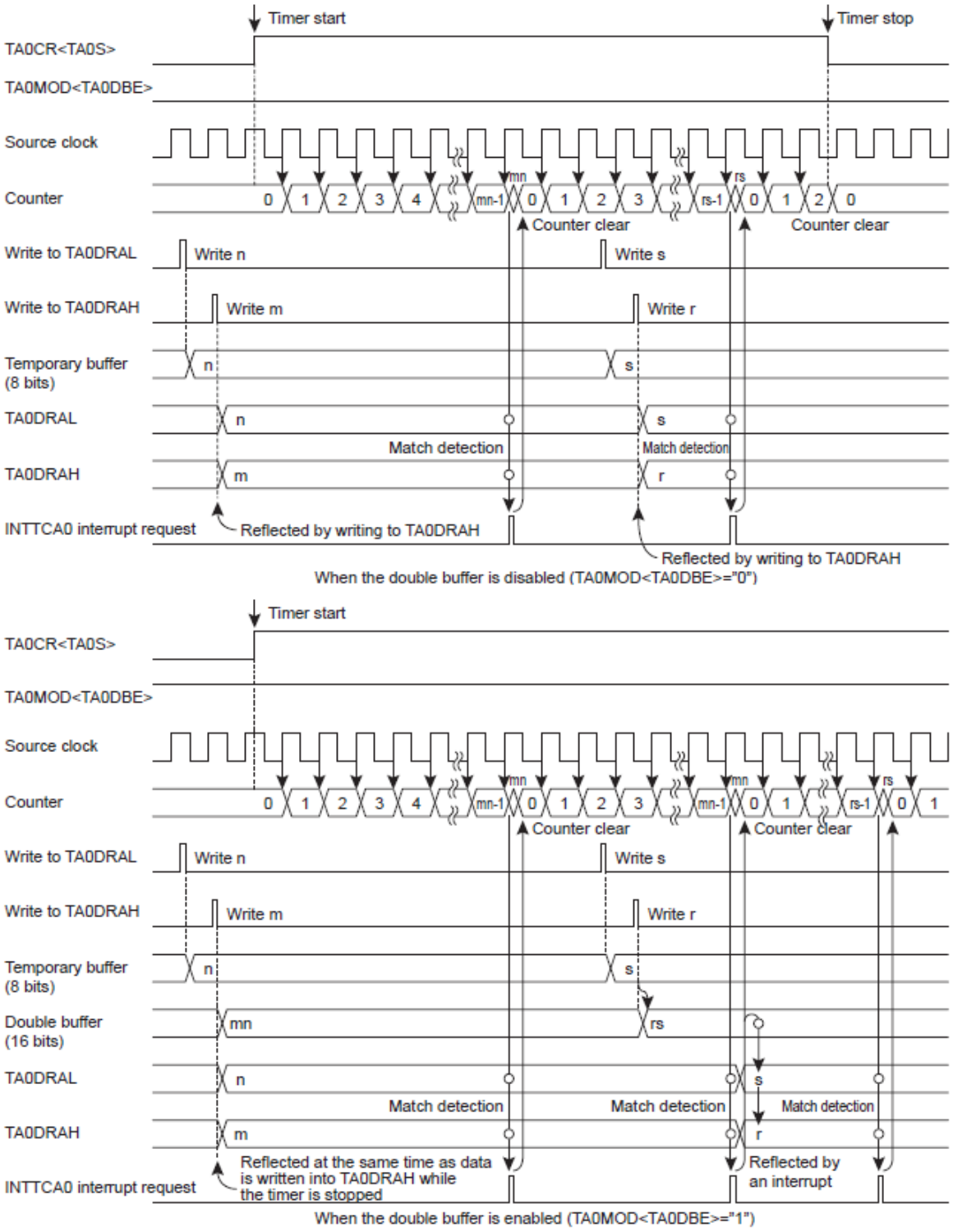



Figure 10.19 Timer Mode Timing Chart

- 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.

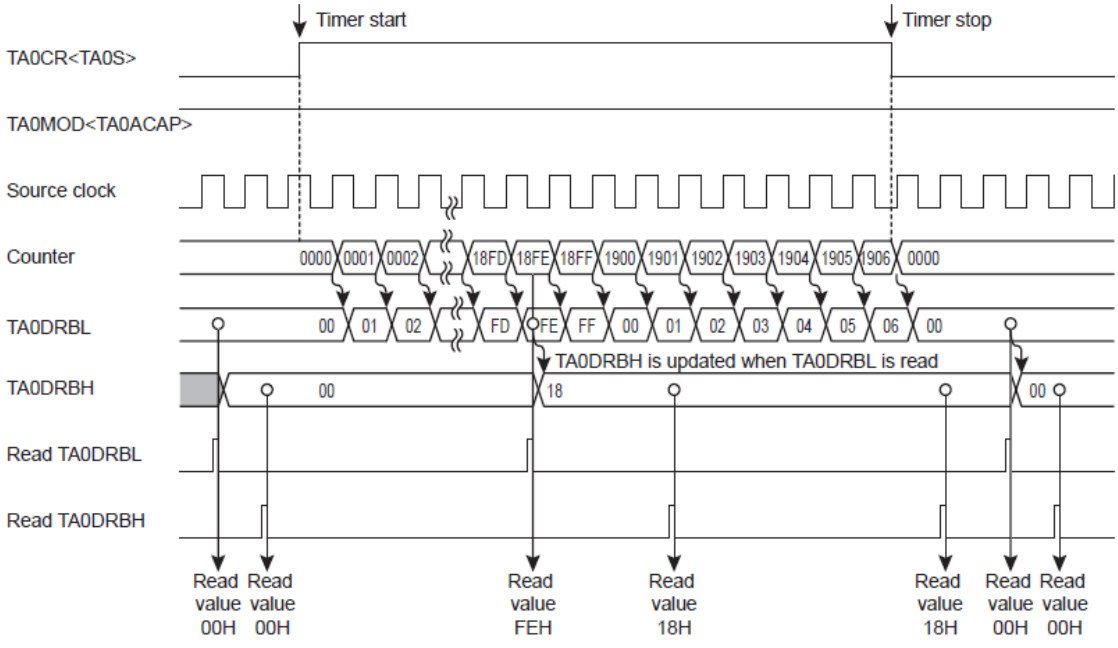


Figure 10.20 Timer Mode Timing Chart (Auto Capture)

### 10.6.3.2 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.

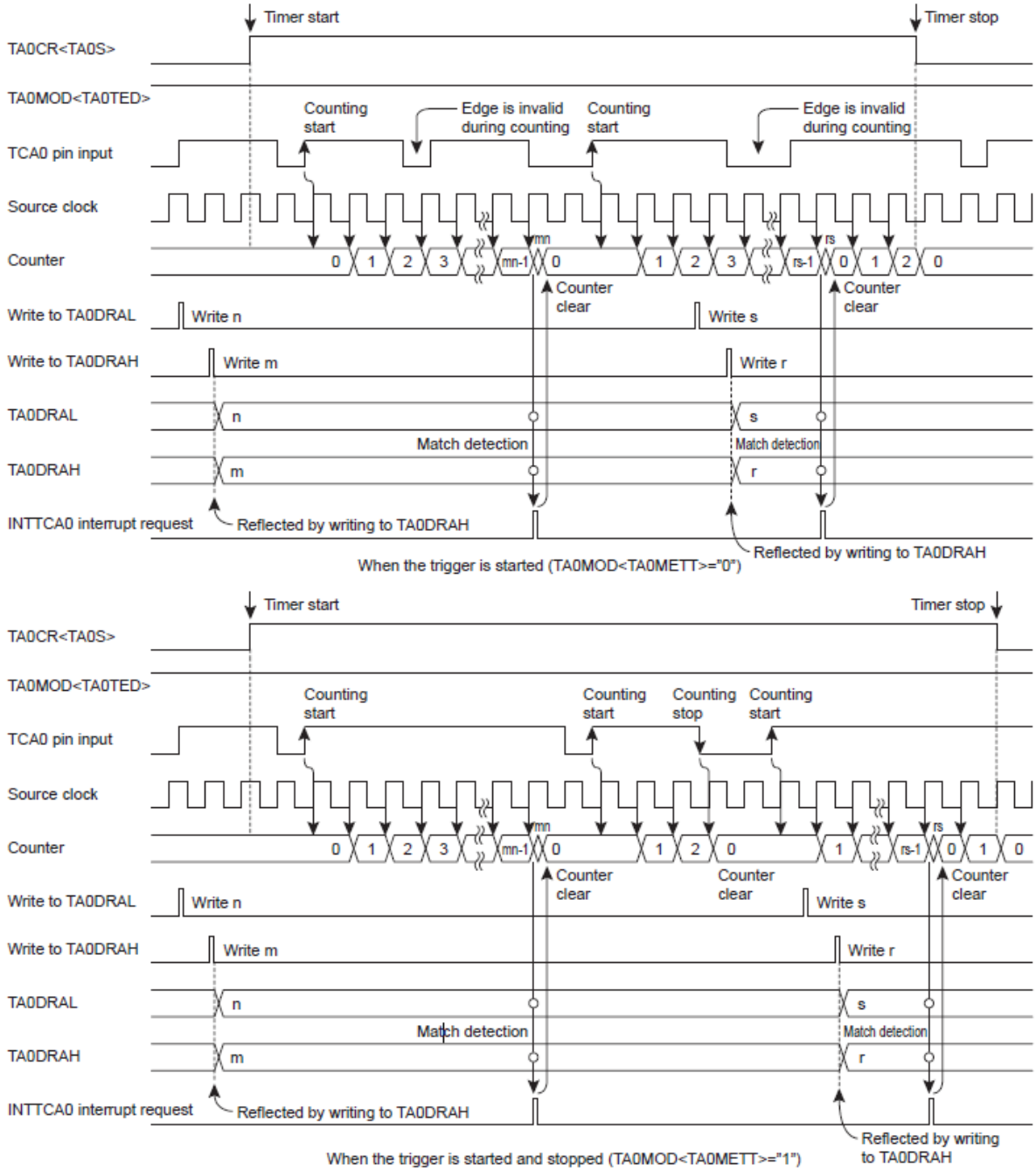



Figure 10.21 External Trigger Timer Mode Timing Chart

(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 TCA0 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 "10.3.3.1 (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "10.3.3.1 (d) Register Buffer Configuration ".

### 10.6.3.3 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 TCA0 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/$2^2$ [Hz] (in the NORMAL 1/2 or IDLE 1/2 mode) or fs/ $2^4$ [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 "10.3.3.1 (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "10.3.3.1 (d) Register Buffer Configuration ".

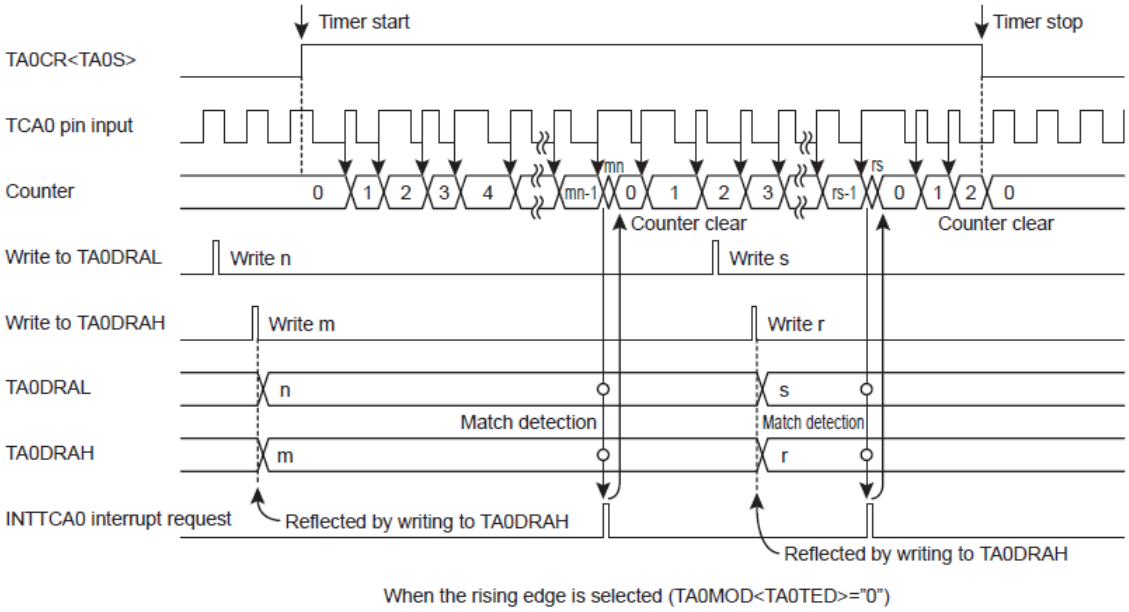



Figure 10.22 Event Counter Mode Timing Chart

### 10.6.3.4 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 TCA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA01CR <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".

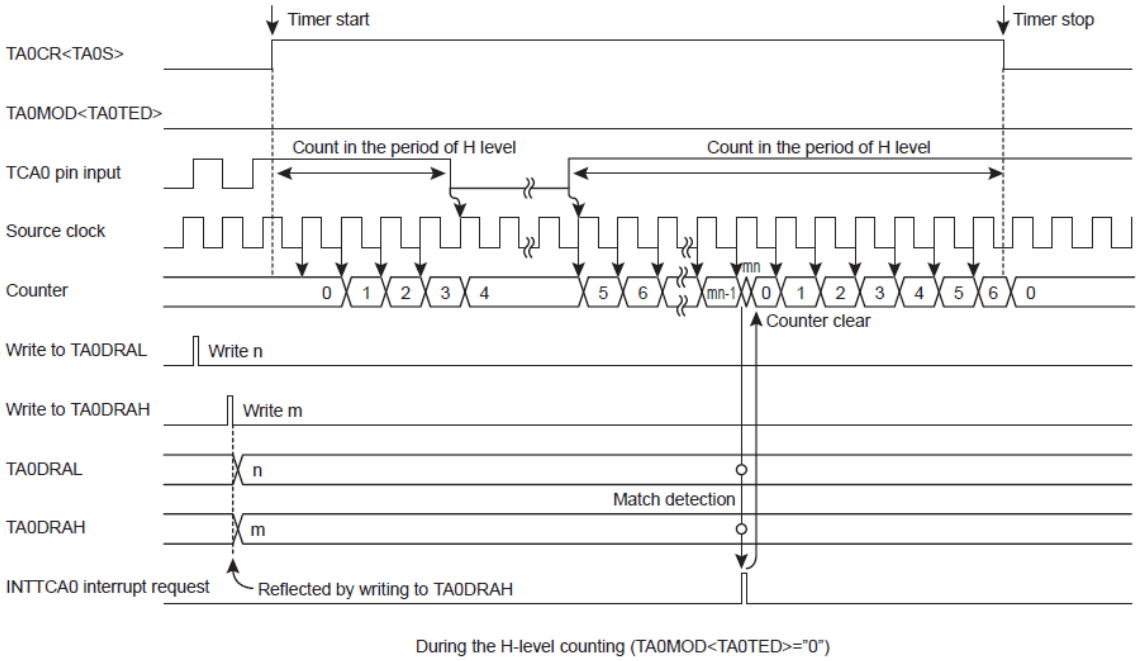



Figure 10.23 Window Mode Timing Chart

### (c) Auto Capture

Refer to "10.3.3.1 (c) Auto Capture".

### (d) Register Buffer Configuration

Refer to "10.3.3.1 (d) Register Buffer Configuration ".

## 10.6.3.5 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 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 TCA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA01CR <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".

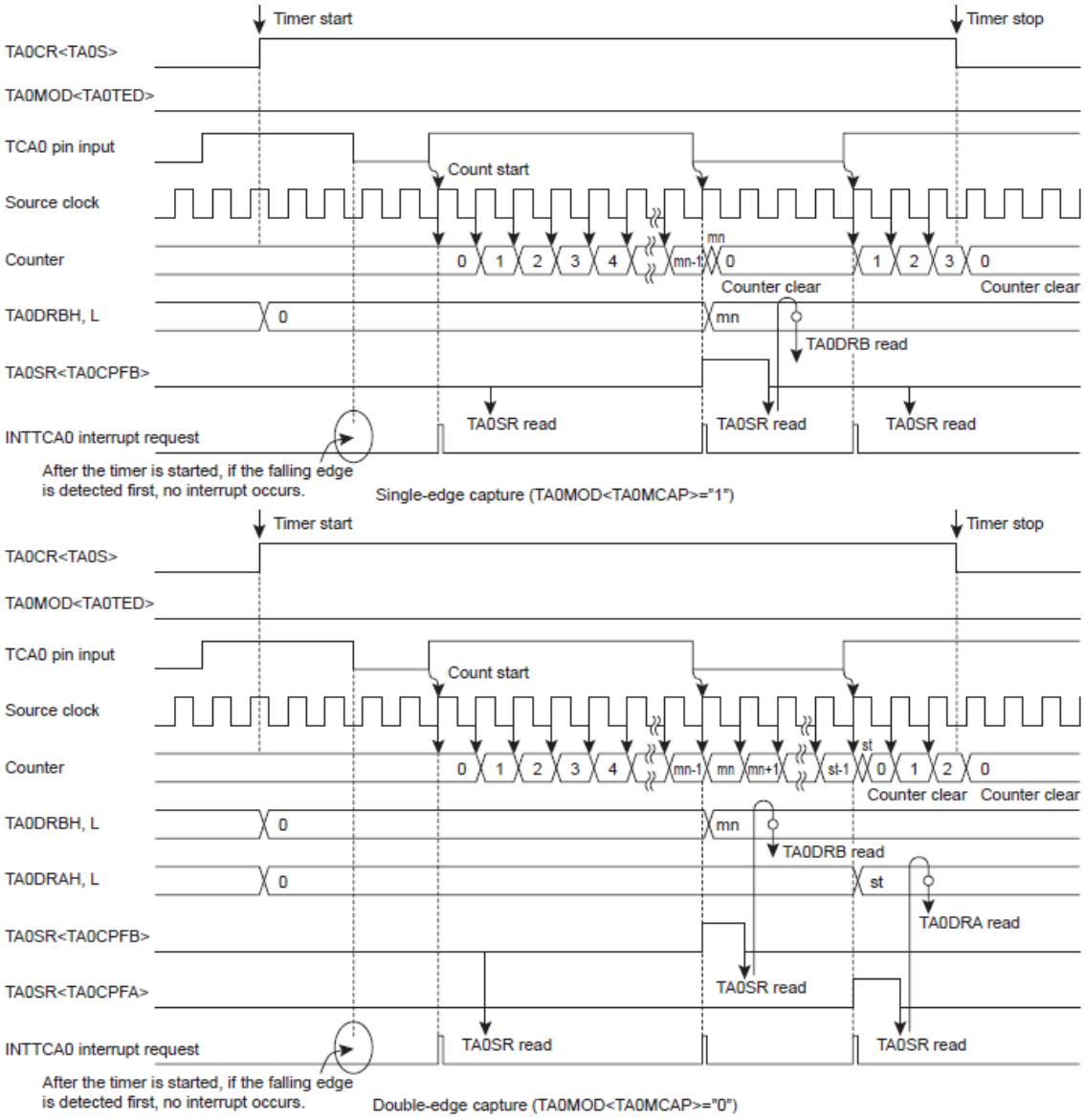



Figure 10.24 Pulse Width Measurement Mode Timing Chart

*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.*

(c) Capture Process

Figure 10.23 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).

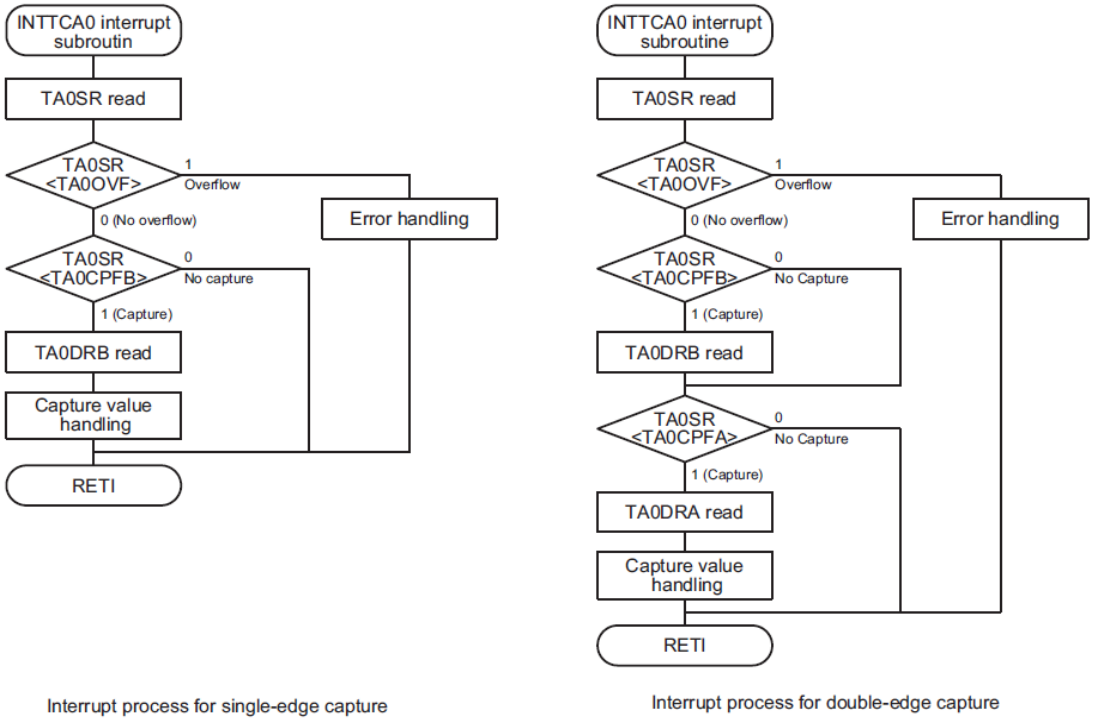



Figure 10.25 Example of Capture Process

### 10.6.3.6 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.

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

MQ6812/MQ6821 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 MQ6812/MQ6821, 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 buffervalue (the last set value) is read, not the TA0DRAH/L (TA0DRBH/L) values (the current effectivevalues).

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).

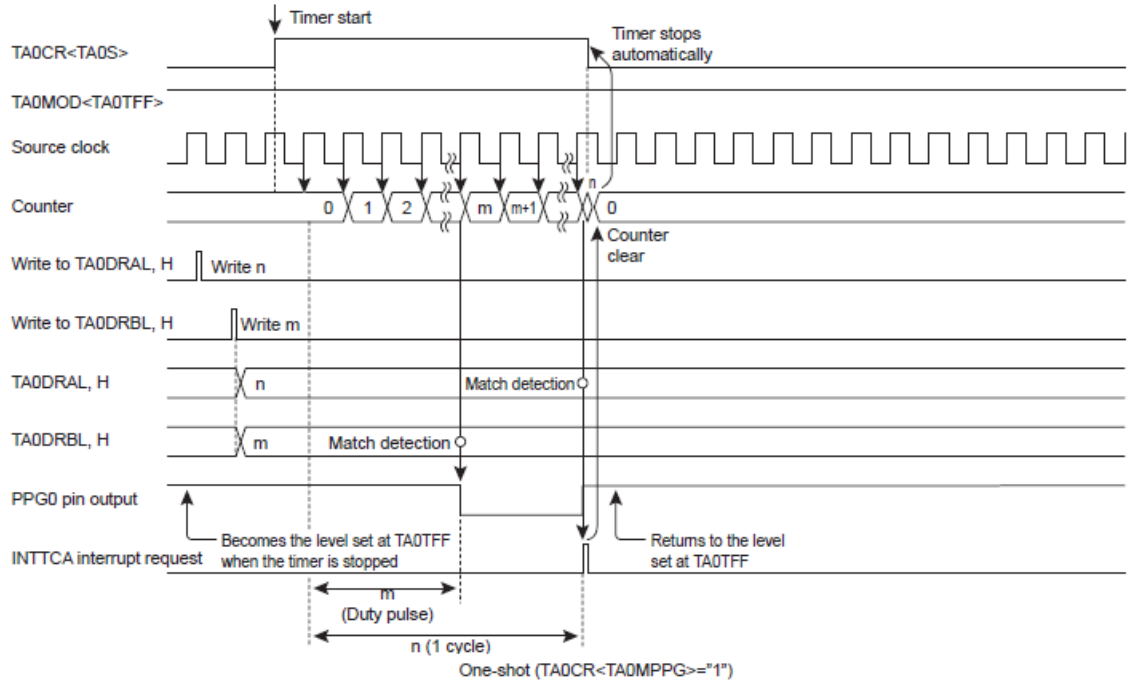



Figure 10.26 PPG Mode Timing Chart - One Shot

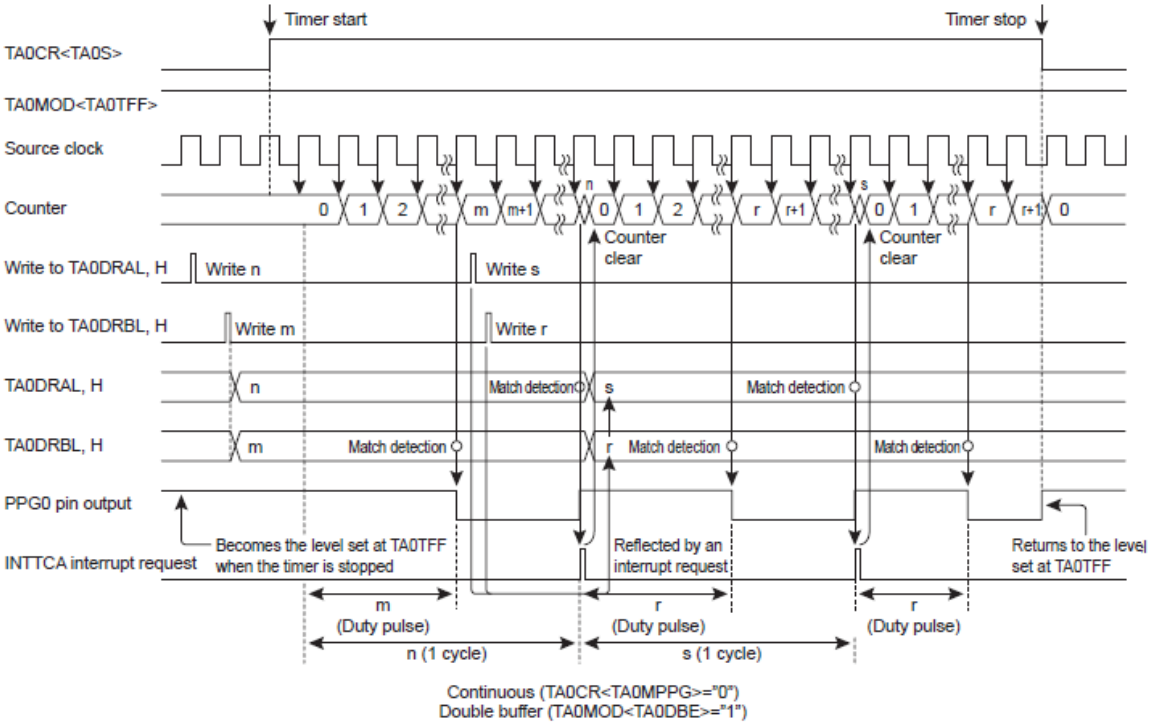



Figure 10.32 PPG Mode Timing Chart - Continuous

### 10.6.4 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.

# 11. LCD

The LCD driver of can directly drive a LCD glass by creating th ac segment and common voltage signals. The LCD related pins :

1. Segment (SEG11 to SEG0)： 12 pins
2. Common( COM7 to COM0)： 8pins

The LCD driver can support 8 types LCD device:

1. 1/8 duty (1/4 bias) LCD，maximum 96 pixels
2.1/8 duty (1/3 bias) LCD，maximum 96 pixels
3. 1/8 duty (1/2 bias) LCD，maximum 96 pixels
4.1/4 duty (1/3 bias) LCD，maximum 48 pixels
5. 1/3 duty (1/3 bias) LCD，maximum 36 pixels
6. 1/3 duty (1/2 bias) LCD，maximum 36 pixels
7. 1/2 duty (1/2 bias) LCD，maximum 24 pixels
8. Static LCD，maximum12 pixels.

*Note ：The VDD sohlud be same as voltage of LCD.*

## 11.1 Configuration

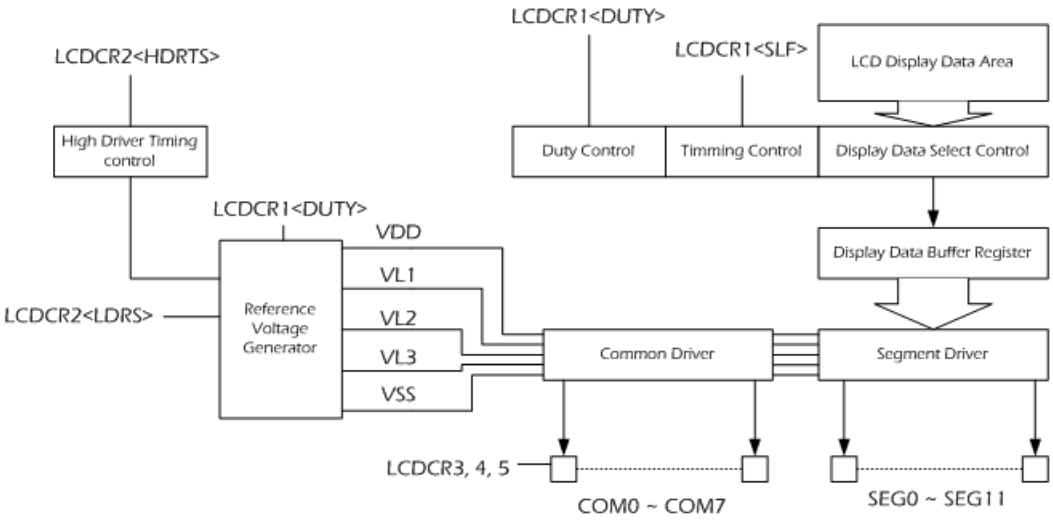



Figure 11.1 LCD driving

## 11.2 Control

LCD consists of the following six registers: low power consumption register (POFFCR2)、LCD control register1(LCDCR1)、LCD control register 2 (LCDCR2)、LCD control register 3 (LCDCR3)、LCD control register 4 (LCDCR4) and LCD control register 5 (LCDCR5).

**Low power consumption register 2**

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

| | | |
|---|---|---|
| LCDEN | LCD control | 0: Disable<br>1: Enable |
| RTCEN | RTC control | 0: Disable<br>1: Enable |
| SIO0EN | SIO0 control | 0: Disable<br>1: Enable |

**LCD control register1**

| LCDCR1 (0x0E7C) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | EDSP | DUTY | | | SLF | | | |
| 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 |

| | | | |
|---|---|---|---|
| EDSP | LCD control | 0: disable<br>1: enable | |
| DUTY | LCD driving select | 000 | 1/4 duty (1/3 bias) |
| | | 001 | 1/3 duty (1/3 bias) |
| | | 010 | 1/3 duty (1/2 bias) |
| | | 011 | 1/2 duty (1/2 bias) |
| | | 100 | Static |
| | | 101 | 1/8 duty (1/4 bias) |
| | | 110 | 1/8 duty (1/3 bias) |
| | | 111 | 1/8 duty (1/2 bias) |
| SLF | Source clock selection | 0000 | $fcgck/2^{18}$ |
| | | 0001 | $fcgck/2^{17}$ |
| | | 0010 | $fcgck/2^{16}$ |
| | | 0011 | $fcgck/2^{15}$ |
| | | 0100 | $fcgck/2^{14}$ |
| | | 0101 | $fcgck/2^{13}$ |
| | | 0110 | $fcgck/2^{12}$ |
| | | 0111 | System reserved |
| | | 1000 | $fs/2^{9}$ |
| | | 1001 | $fs/2^{8}$ |
| | | 1010 | System reserved |
| | | ~ | ~ |
| | | 1111 | System reserved |

*Note 1：fcgck，Gear clock[Hz]，fs，slow clock [Hz]。*

*Note2：In slow 2 mode, do not set SLF to" 0000"~"0110" (fcgck as basic clock). Otherwise segment and common may output waveform with abnormal frame frequenct.*

**LCD control register 2**

| LCDCR2 (0x0E7D) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | HDRTS | | | LDRS | - | - | - | - |
| Read/Write | R/W | R/W | R/W | R/W | R | R | R | R |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | | LCDCR1 <SLF> | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 1000 | 1001 |
| HDRTS | Select high-drive time | 000 | No connected | | | | | | | | |
| | | 001 | $2^{11}$/fcgck | $2^{10}$/fcgck | $2^{9}$/fcgck | $2^{8}$/fcgck | $2^{7}$/fcgck | $2^{6}$/fcgck | $2^{5}$/fcgck | $2^{2}$/fs | 2/fs |
| | | 010 | $2^{12}$/fcgck | $2^{11}$/fcgck | $2^{10}$/fcgck | $2^{9}$/fcgck | $2^{8}$/fcgck | $2^{7}$/fcgck | $2^{6}$/fcgck | $2^{3}$/fs | $2^{2}$/fs |
| | | 011 | $2^{13}$/fcgck | $2^{12}$/fcgck | $2^{11}$/fcgck | $2^{10}$/fcgck | $2^{9}$/fcgck | $2^{8}$/fcgck | $2^{7}$/fcgck | $2^{4}$/fs | $2^{3}$/fs |
| | | 100 | $2^{14}$/fcgck | $2^{13}$/fcgck | $2^{12}$/fcgck | $2^{11}$/fcgck | $2^{10}$/fcgck | $2^{9}$/fcgck | $2^{8}$/fcgck | $2^{5}$/fs | $2^{4}$/fs |
| | | 101 | $2^{15}$/fcgck | $2^{14}$/fcgck | $2^{13}$/fcgck | $2^{12}$/fcgck | $2^{11}$/fcgck | $2^{10}$/fcgck | $2^{9}$/fcgck | $2^{6}$/fs | $2^{5}$/fs |
| | | 110 | Keep connected | | | | | | | | |
| | | 111 | System reserved | | | | | | | | |
| LDRS | Select low-drive current | 0: select low-drive low current<br>1: select low-drive high current | | | | | | | | | |

*Note 1：fcgck，Gear clock [Hz]，fs，slow clock [Hz].*

*Note 2：Bit 3~Bit 0 of LCDCR2 read is "0".*

LCD Control Register3

| LCDCR3 (0x0E7E) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | SEG7_EN | SEG6_EN | SEG5_EN | SEG4_EN | SEG3_EN | SEG2_EN | SEG1_EN | SEG0_EN |
| 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 |

| | | |
|---|---|---|
| SEG7_EN | LCD SEG7 pin function select | 0: P74 is other I/O function<br>1: P74 is SEG7 output |
| SEG6_EN | LCD SEG6 pin function select | 0: P76 is other I/O function<br>1: P76 is SEG6 output |
| SEG5_EN | LCD SEG5 pin function select | 0: P70 is other I/O function<br>1: P70 is SEG5 output |
| SEG4_EN | LCD SEG4 pin function select | 0: P75 is other I/O function<br>1: P75 is SEG4 output |
| SEG3_EN | LCD SEG3 pin function select | 0: P77 is other I/O function<br>1: P77 is SEG3 output |
| SEG2_EN | LCD SEG2 pin function select | 0: P27 is other I/O function<br>1: P27 is SEG2 output |
| SEG1_EN | LCD SEG1 pin function select | 0: P26 is other I/O function<br>1: P26 is SEG1 output |
| SEG0_EN | LCD SEG0 pin function select | 0: P25 is other I/O function<br>1: P25 is SEG0 output |

*Note: When LCD enable, set LCD related pins(COM ans SEG) to input mode.*

LCD Control Register 4

| LCDCR4 (0x0E7F) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | | | | | SEG11_EN | SEG10_EN | SEG9_EN | SEG8_EN |
| Read/Write | R | R | R | R | R/W | R/W | R/W | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| SEG11_EN | LCD SEG11 pin function select | 0: P44 is other I/O function<br>1: P44 is SEG11 output |
| SEG10_EN | LCD SEG10 pin function select | 0: P45 is other I/O function<br>1: P45 is SEG10 output |
| SEG9_EN | LCD SEG9 pin function select | 0: P46 is other I/O function<br>1: P46 is SEG9 output |
| SEG8_EN | LCD SEG8 pin function select | 0: P47 is other I/O function<br>1: P47 is SEG8 output |

*Note: When LCD enable, set LCD related pins(COM ans SEG) to input mode.*

LCD Control Register 5

| LCDCR5 (0x0E80) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | COM7_EN | COM6_EN | COM5_EN | COM4_EN | COM3_EN | COM2_EN | COM1_EN | COM0_EN |
| 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 |

| | | |
|---|---|---|
| COM7_EN | LCD COM7 pin function select | 0: P24 is other I/O function<br>1: P24 is COM7 output |
| COM6_EN | LCD COM6 pin function select | 0: P23 is other I/O function<br>1: P23 is COM6 output |
| COM5_EN | LCD COM5 pin function select | 0: P83 is other I/O function<br>1: P83 is COM5 output |
| COM4_EN | LCD COM4 pin function select | 0: P82 is other I/O function<br>1: P82 is COM4 output |
| COM3_EN | LCD COM3 pin function select | 0: P73 is other I/O function<br>1: P73 is COM3 output |
| COM2_EN | LCD COM2 pin function select | 0: P72 is other I/O function<br>1: P72 is COM2 output |
| COM1_EN | LCD COM1 pin function select | 0: P91 is other I/O function<br>1: P91 is COM1 output |
| COM0_EN | LCD COM0 pin function select | 0: P90 is other I/O function<br>1: P90 is COM0 output |

*Note: When LCD enable, set LCD related pins(COM ans SEG) to input mode.*

## 11.3 Low Power Consumption Function

LCD has the low power consumption register2 (POFFCR2) that saves power consumption when the LCD is not used.

Setting POFFCR2<LCDEN> to "0" disables the basic clock supply to LCD to save power. Note that this makes the LCD function unusable. Setting POFFCR2<LCDEN> to "1" enables the basic clock supply to LCD and allows the LCD function to operate.

After reset, POFFCR2<LCDEN> is initialized to "0", and this makes the LCD unusable. When using the LCD for the first time, be sure to set POFFCR2<LCDEN> to "1" in the initial setting of the program (before the LCD control register is operated).

Do not change POFFCR2<LCDEN> to "0" during the timer operation. Otherwise LCD function may operate unexpectedly.

## 11.4 LCD Driving Function

### 11.4.1 LCD control Register (LCDCR1<EDSP>)

Setting the LCD control register1 LCDCR1 <EDSP> to "1" activates the LCD function, LCD driving enable, and start LCD display. Setting the LCD control register1 LCDCR1 <EDSP> to "0" stop the LCD function, LCD driving disable, and LCD display stop.Table 11.1 LCD driving pin status when LCD enable or disable

| SEGx_EN | LCDCR1<EDSP> | Pin status (GPIO、SEG/COM Outpot) |
|---|---|---|
| 0 | 0 | GPIO |
| 0 | 1 | |
| 1 | 0 | Low level |
| 1 | 1 | Segment output |

| COMx_EN | LCDCR1<EDSP> | Pin status (GPIO、SEG/COM Outpot) |
|---|---|---|
| 0 | 0 | GPIO |
| 0 | 1 | |
| 1 | 0 | Low level |
| 1 | 1 | Common output |

**Table 11.1 LCD driving pin status**

*Note：SEG/COM output no. is "x".*

#### 11.4.1.1 Reset

After reset, LCDCR1 <EDSP> are initialized to "0", and this makes the power of LCD display stop automaticaliy, cut VL1,L2,VL3 bias voltage.The bias voltage output pin would be fixed to low-level at this time. LCD driving pin would be GIPO input pin (High-resistro). Therefore, LCD dispay would become not clear if the external reset time is long.

#### 11.4.1.2 IDLE0、SLEEP0 and STOP mode

When LCDCR1<EDSP> is "1" in IDLE0,SLEEP0 and STOP modes, LCDCR1 <EDSP> would be initialized to "0", LCD display stop. When out of IDLE0,SLEEP0 and STOP modes , reset LCDCR1 <EDSP> to "1" to enable LCD display. r

#### 11.4.1.3 SLOW mode

Enable LCD function in NORMAL 2 and SLOW 1/2 mode, LCDCR1<SLF> is suggested to set as fs ("1000"到"1001"). In this case, do not need to reset the LCDCR1<SLF> when the operation mode change between NORMAL2 and SLOW1/2.

If the LCD clock is fcgck in normal2 mode, LCDCR1<EDSP> have to clear to "0" before change to SLOW2 mode. In SLOW2 mode, change the LCD clock to fs by LCDCR1<SLF>, and set LCDCR1<EDSP> to"1". Similarly, LCDCR1<EDSP> have to clear to "0" before operation mode chage from SLOW2 to NORMAL2. In NORMAL2 mode, change the LCD clock to fcgk by LCDCR1<SLF>, and set LCDCR1<EDSP> to"1".

#### 11.4.1.4 LCD clock (LCDCR1<SLF>)

Before LCDCR1<SLF> set"0000" to"0110", enable high-frequency clock (SYSCR2<XEN>="1" or SYSCR2 <OSCEN>="1") and the oscillation is stable . And then set LCDCR1<EDSP> to"1". Set LCDCR1 <EDSP> to"1"would not enable LCD function when high-frequency clock stop. (Set LCDCR1<EDSP> to"1", LCD would not enable)

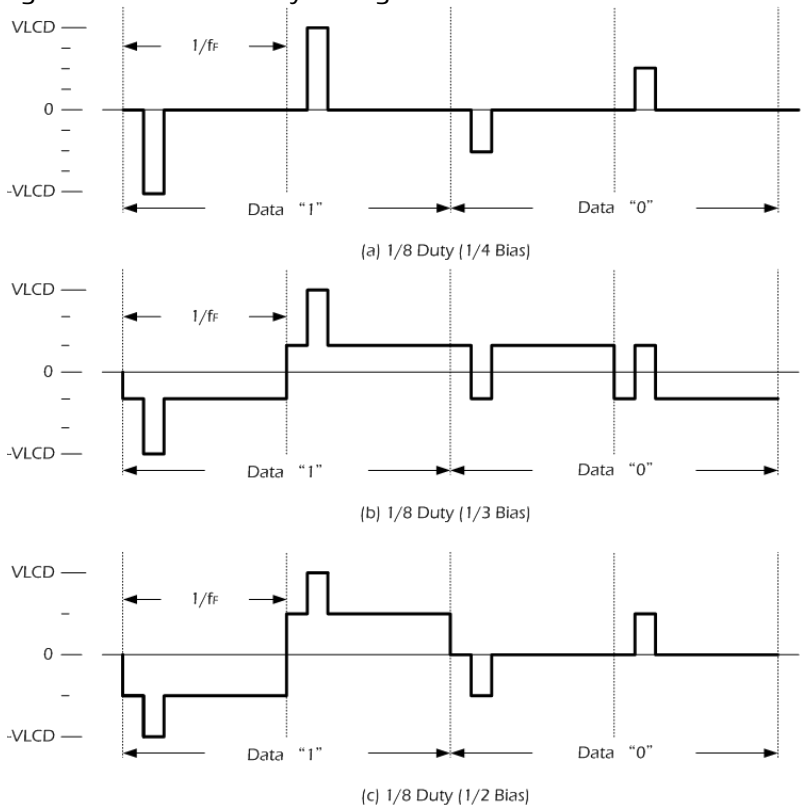
Similarly, before LCDCR1<SLF> set"0000" to"0110", enable low-frequency clock (SYSCR2<XEN>="1") and the oscillation is stable . And then set LCDCR1<EDSP> to"1". Set LCDCR1 <EDSP> to"1"would not enable LCD function when low-frequency clock stop. (Set LCDCR1<EDSP> to"1", LCD would not enable)

#### 11.4.1.5 The LCD drive of POFFCR2

Set POFFCR2<LCDEN> to "0" would stop LCD display , when LCDCR1<EDSP> is "1". Reset POFFCR2 <LCDEN> to "1" would restart LCD display.

### 11.4.2 LCD drive mode (LCDCR1<DUTY>)

To select eight LCD drive modse by setting LCDCR1<DUTY>.




(a) 1/8 Duty (1/4 Bias)

(b) 1/8 Duty (1/3 Bias)

(c) 1/8 Duty (1/2 Bias)

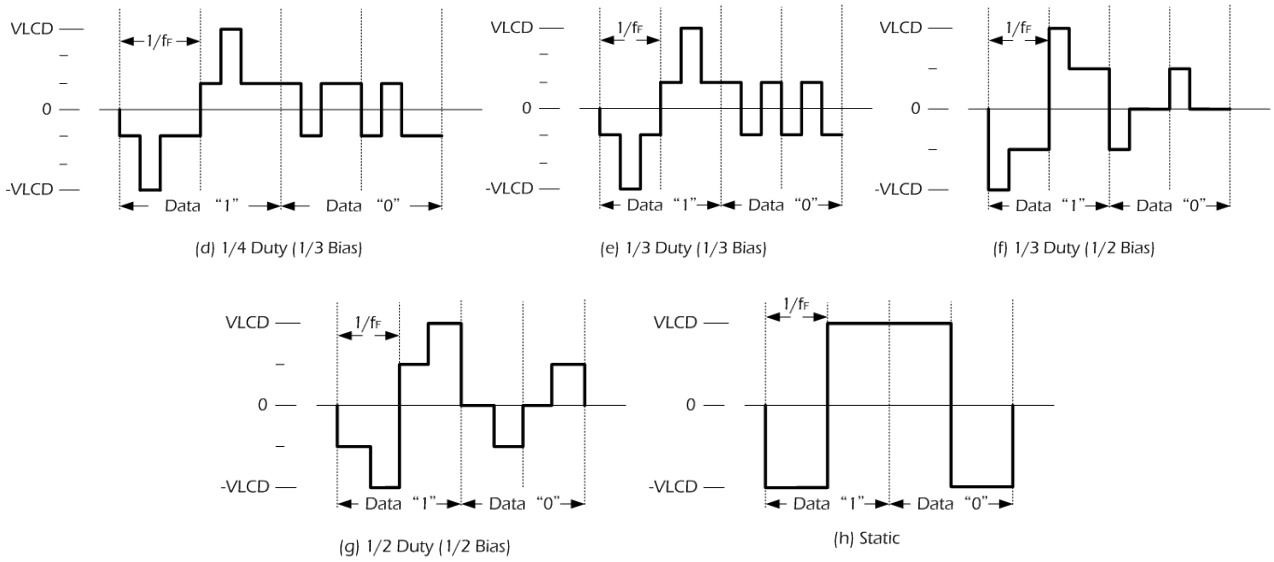



**Figure 11.2 Waveform of LCD display mode (the bias voltage between SEG and COM)**

*Note1：fF is frame frequency*

*Note 2：VLCD is LCD display voltage ( = VDD – VSS)*

### 11.4.3 Frame frequency (LCDCR1<SLF>)

Frame frequency ($f_F$) is decied by drive mode and clock frequenct. As table 11.2, select clock frequency by LCDCR1<SLF>.

| LCDCR1 <SLF> | Clock frequency [Hz] | Frame Frequency [Hz] | | | | |
|---|---|---|---|---|---|---|
| | | 1/8 bias | 1/4 bias | 1/3 bias | 1/2 bias | Static |
| 0000 | $fcgck/2^{18}$ | $(4/8) \times fcgck/2^{18}$ | $fcgck/2^{18}$ | $(4/3) \times fcgck/2^{18}$ | $(4/2) \times fcgck/2^{18}$ | $fcgck/2^{18}$ |
| | (fcgck = 16MHz) | 31 | 61 | 81 | 122 | 61 |
| 0001 | $fcgck/2^{17}$ | $(4/8) \times fcgck/2^{17}$ | $fcgck/2^{17}$ | $(4/3) \times fcgck/2^{17}$ | $(4/2) \times fcgck/2^{17}$ | $fcgck/2^{17}$ |
| | (fcgck = 16MHz) | 61 | 122 | 163 | 244 | 122 |
| | (fcgck = 8MHz) | 31 | 61 | 81 | 122 | 61 |
| 0010 | $fcgck/2^{16}$ | $(4/8) \times fcgck/2^{16}$ | $fcgck/2^{16}$ | $(4/3) \times fcgck/2^{16}$ | $(4/2) \times fcgck/2^{16}$ | $fcgck/2^{16}$ |
| | (fcgck = 8MHz) | 61 | 122 | 163 | 244 | 122 |
| | (fcgck = 4MHz) | 31 | 61 | 81 | 122 | 61 |
| 0011 | $fcgck/2^{15}$ | $(4/8) \times fcgck/2^{15}$ | $fcgck/2^{15}$ | $(4/3) \times fcgck/2^{15}$ | $(4/2) \times fcgck/2^{15}$ | $fcgck/2^{15}$ |
| | (fcgck = 4MHz) | 61 | 122 | 163 | 244 | 122 |
| | (fcgck = 2MHz) | 31 | 61 | 81 | 122 | 61 |
| 0100 | $fcgck/2^{14}$ | $(4/8) \times fcgck/2^{14}$ | $fcgck/2^{14}$ | $(4/3) \times fcgck/2^{14}$ | $(4/2) \times fcgck/2^{14}$ | $fcgck/2^{14}$ |
| | (fcgck = 2MHz) | 61 | 122 | 163 | 244 | 122 |
| | (fcgck = 1MHz) | 31 | 61 | 81 | 122 | 61 |
| 0101 | $fcgck/2^{13}$ | $(4/8) \times fcgck/2^{13}$ | $fcgck/2^{13}$ | $(4/3) \times fcgck/2^{13}$ | $(4/2) \times fcgck/2^{13}$ | $fcgck/2^{13}$ |
| | (fcgck = 1MHz) | 61 | 122 | 163 | 244 | 122 |
| | (fcgck = 0.5MHz) | 31 | 61 | 81 | 122 | 61 |
| 0110 | $fcgck/2^{12}$ | $(4/8) \times fcgck/2^{12}$ | $fcgck/2^{12}$ | $(4/3) \times fcgck/2^{12}$ | $(4/2) \times fcgck/2^{12}$ | $fcgck/2^{12}$ |
| | (fcgck = 0.5MHz) | 61 | 122 | 163 | 244 | 122 |
| | (fcgck = 0.25MHz) | 31 | 61 | 81 | 122 | 61 |
| 1000 | $fs/2^9$ | $(4/8) \times fs/2^9$ | $fs/2^9$ | $(4/3) \times fs/2^9$ | $(4/2) \times fs/2^9$ | $fs/2^9$ |
| | (fs = 32.768 KHz) | 32 | 64 | 85 | 128 | 64 |
| 1001 | $fs/2^8$ | $(4/8) \times fs/2^8$ | $fs/2^8$ | $(4/3) \times fs/2^8$ | $(4/2) \times fs/2^8$ | $fs/2^8$ |
| | (fs = 32.768 KHz) | 64 | 128 | 171 | 256 | 128 |

Table 11.2 Frame frequency

*Note：fcgck，gear clockHz]，fs，slow clock [Hz]。*

### 11.4.4 High-drive time select (LCDCR2<HDRTS>)and Low-drive current select (LCDCR2<LDRS>)

Generally, LCD drive capacity is higher when connect to high-drive circuit longer, and the power consumption is also higher. On the contrary, LCD drive capacity is lower when connect to high-drive

circuit shorter, and the power consumption is also lower. LCD display would become no good(e.g. LCD display becom not clear), if LCD drive capacity is not enough. Select the LCD drive to best the LCD display. Table 11.3 shows the estimation of current of LCDCR2<HDRTS>and LCDCR2 <LDRS> and LCD modes.

| LCDCR2 <HDRTS> | VDD condition | 1/8 duty (1/4 bias) | | 1/4 duty (1/3 bias) | | 1/3 duty (1/3 bias) | | 1/3 duty (1/2 bias) | | unit |
|---|---|---|---|---|---|---|---|---|---|---|
| | | LDRS=1 | LDRS=0 | LDRS=1 | LDRS=0 | LDRS=1 | LDRS=0 | LDRS=1 | LDRS=0 | |
| 000 | High-drive time | 0 % (always keep low-driving current) | | | | | | | | % |
| | VDD = 5V | 15.60 | 9.52 | 10.91 | 6.77 | 10.91 | 6.77 | 5.69 | 3.75 | uA |
| | VDD = 3V | 13.46 | 8.44 | 9.39 | 5.99 | 9.39 | 5.99 | 5.07 | 3.45 | |
| 001 | High-drive time | 3.13 % | | | | | | | | % |
| | VDD = 5V | 18.46 | 12.57 | 13.04 | 9.03 | 13.04 | 9.03 | 6.41 | 4.53 | uA |
| | VDD = 3V | 15.62 | 10.75 | 10.98 | 7.69 | 10.98 | 7.69 | 5.64 | 4.07 | |
| 010 | High-drive time | 6.25 % | | | | | | | | % |
| | VDD = 5V | 21.31 | 15.61 | 15.18 | 11.30 | 15.18 | 11.30 | 7.13 | 5.32 | uA |
| | VDD = 3V | 17.78 | 13.07 | 12.57 | 9.38 | 12.57 | 9.38 | 6.20 | 4.68 | |
| 011 | High-drive time | 12.50 % | | | | | | | | % |
| | VDD = 5V | 27.03 | 21.71 | 19.45 | 15.82 | 19.45 | 15.82 | 8.58 | 6.88 | uA |
| | VDD = 3V | 22.09 | 17.70 | 15.75 | 12.78 | 15.75 | 12.78 | 7.34 | 5.92 | |
| 100 | High-drive time | 25.00 % | | | | | | | | % |
| | VDD = 5V | 38.45 | 33.89 | 27.98 | 24.88 | 27.98 | 24.88 | 11.47 | 10.01 | uA |
| | VDD = 3V | 30.72 | 26.96 | 22.12 | 19.57 | 22.12 | 19.57 | 9.60 | 8.39 | |
| 101 | High-drive time | 50.00 % | | | | | | | | % |
| | VDD = 5V | 61.30 | 58.26 | 45.06 | 42.99 | 45.06 | 42.99 | 17.25 | 16.28 | uA |
| | VDD = 3V | 47l98 | 45.47 | 34.85 | 33.15 | 34.85 | 33.15 | 14.14 | 13.33 | |
| 110 | High-drive time | 100% (always keep low-driving current) | | | | | | | | % |
| | VDD = 5V | 107.00 | 107.00 | 79.20 | 79.20 | 79.20 | 79.20 | 28.80 | 28.80 | uA |
| | VDD = 3V | 82.50 | 82.50 | 60.30 | 60.30 | 60.30 | 60.30 | 23.20 | 23.20 | |

Table 11.3 The List of high-driving current and low-driving current (estimated.)

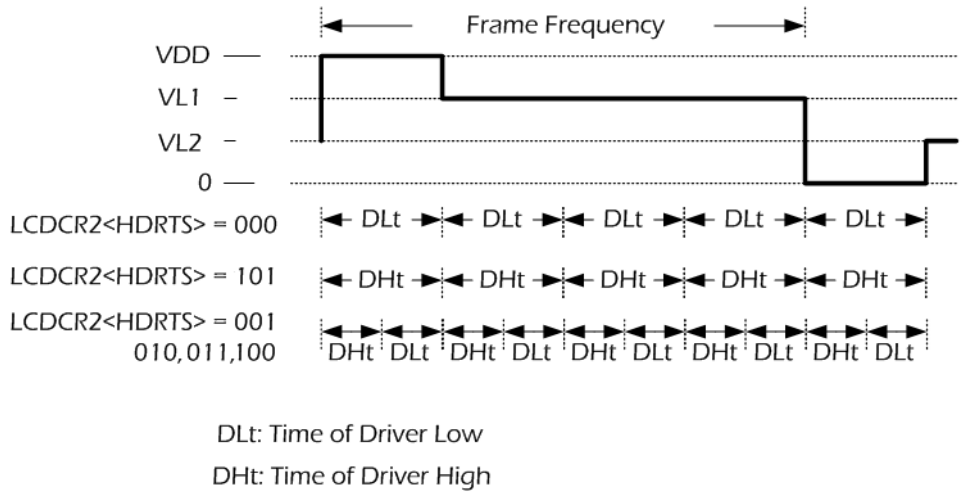



Figure 11.3 Timing sequency for 1/4 duty (1/3 bias) LCD mode

## 11.5 LCD display data

LCD display data store to display data stoarage area (0x0E40 to 0x0E4B, total 12 bytes)

The data in this area would be read by hardware automatically, and then transfer to LCD drive circuit. LCD drive circuit would follow display data to generate segment and common signal. Therefore, the user can change the display pattern easily by modifying the content in display data storage area.

Blink LCD when display data is"1"; disalbe LCD when display data is"0". At reset, the data in display data area(0x0E40 to 0x0E4B) would be initialized to "0".

Becaouse the LCD pixel is different in different LCD modes, the display data byte would be different. The address not used to storage display data or the address not connect to LCD can used to store general users' data (Table 11.4).

| Driving mode | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| 1/8 duty | COM7 | COM6 | COM5 | COM4 | COM3 | COM2 | COM1 | COM0 |
| 1/4 duty | | | | | COM3 | COM2 | COM1 | COM0 |
| 1/3 duty | | | | | | COM2 | COM1 | COM0 |
| 1/2 duty | | | | | | | COM1 | COM0 |
| Static | | | | | | | | COM0 |

Table 11.4 The Address for saving display data

| Registar name (address) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 | Read/ Write | Initial value |
|---|---|---|---|---|---|---|---|---|---|---|
| LCDBUF00 (0x0E40) | SEG0 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF01 (0x0E41) | SEG1 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF02 (0x0E42) | SEG2 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF03 (0x0E43) | SEG3 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF04 (0x0E44) | SEG4 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF05 (0x0E45) | SEG5 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF06 (0x0E46) | SEG6 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF07 (0x0E47) | SEG7 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF08 (0x0E48) | SEG8 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF09 (0x0E49) | SEG9 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF10 (0x0E4A) | SEG10 | | | | | | | | R/W | ( 0000 0000 ) |
| LCDBUF11 (0x0E4B) | SEG11 | | | | | | | | R/W | ( 0000 0000 ) |
| | COM7 | COM6 | COM5 | COM4 | COM3 | COM2 | COM1 | COM0 | | |

Table 11.5 LCD display data area

## 11.6 The Example of LCD

### 11.6.1 Initial

The initial LCD driving function flow chart as figure 11.4

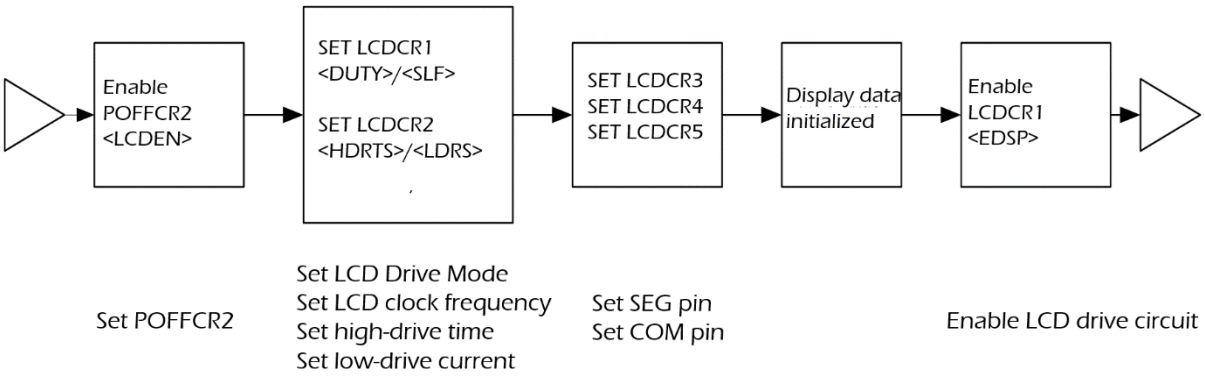



Figure 11.4 Initial LCD driving function folw chart

### 11.6.2 Display data setting

Display data is fixed in falsh, and transfer to display data area by specific instruction.

Figure11.5 is the example of 1/4 duty (1/3 bias) LCD mode. Table 11.6 shows the display data example of 1/4 duty (1/3 bias) LCD


Figure11.5 The example of 1/4 duty (1/3 bias) LCD

| NO. | Display | Display data | NO. | Display | Display data |
|---|---|---|---|---|---|
| 0 |  | SEG0 =xxxx1111<br>SEG1 =xxxx1101 | 5 |  | SEG0 =xxxx0101<br>SEG1 =xxxx1011 |
| 1 |  | SEG0 =xxxx0110<br>SEG1 =xxxx0000 | 6 |  | SEG0 =xxxx0101<br>SEG1 =xxxx1111 |
| 2 |  | SEG0 =xxxx0011<br>SEG1 =xxxx1110 | 7 |  | SEG0 =xxxx0111<br>SEG1 =xxxx0000 |
| 3 |  | SEG0 =xxxx0111<br>SEG1 =xxxx1010 | 8 |  | SEG0 =xxxx0111<br>SEG1 =xxxx1111 |
| 4 |  | SEG0 =xxxx0110<br>SEG1 =xxxx0011 | 9 |  | SEG0 =xxxx0111<br>SEG1 =xxxx1011 |

Table 11.6 The example of display figure "0~9" (1/4 duty, (1/3 bias) LCD)

Figure 11.6 and Table11.7 is the example of 1/2duty (1/2bias) LCD mode. And the example of display figure "0~9".

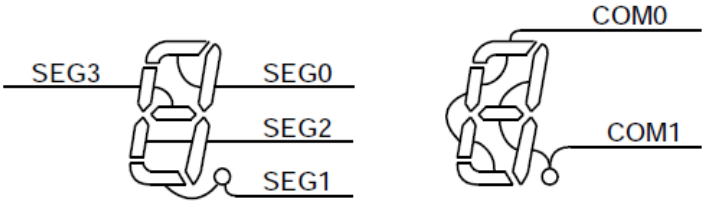



Figure11.6 The example of 1/2 duty (1/2 bias) LCD

| Number | Display data | | | |
|---|---|---|---|---|
| | SEG3 | SEG2 | SEG1 | SEG0 |
| 0 | xxxxxx01 | xxxxxx11 | xxxxxx01 | xxxxxx11 |
| 1 | xxxxxx00 | xxxxxx10 | xxxxxx00 | xxxxxx10 |
| 2 | xxxxxx10 | xxxxxx01 | xxxxxx01 | xxxxxx11 |
| 3 | xxxxxx10 | xxxxxx10 | xxxxxx01 | xxxxxx11 |
| 4 | xxxxxx11 | xxxxxx10 | xxxxxx00 | xxxxxx10 |
| 5 | xxxxxx11 | xxxxxx10 | xxxxxx01 | xxxxxx01 |
| 6 | xxxxxx11 | xxxxxx11 | xxxxxx01 | xxxxxx01 |
| 7 | xxxxxx01 | xxxxxx10 | xxxxxx00 | xxxxxx11 |
| 8 | xxxxxx11 | xxxxxx11 | xxxxxx01 | xxxxxx11 |
| 9 | xxxxxx11 | xxxxxx10 | xxxxxx01 | xxxxxx11 |

Table 11.7 The example of display figure "0~9" (1/2 duty, (1/2 bias) LCD)

### 11.6.3 LCD Drive Example

Figure11.7、figure 11.8 and figure 11.9 are the LCD drive waveform example for 1/4duty (1/3bias), 1/2 duty (1/2 bias) and static LCD separately.

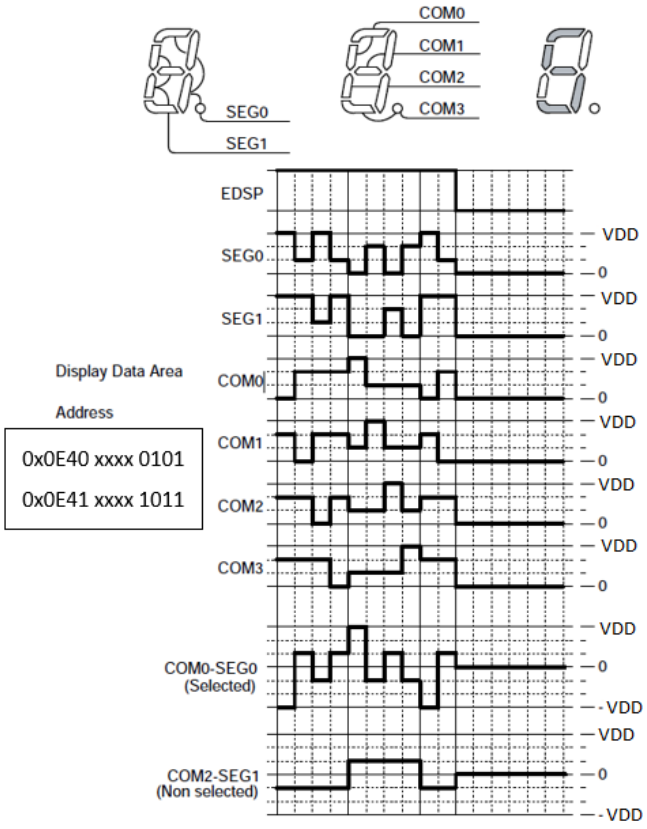



Figure11.7 LCD drive waveform examples for 1/4 duty (1/3 bias) LCD

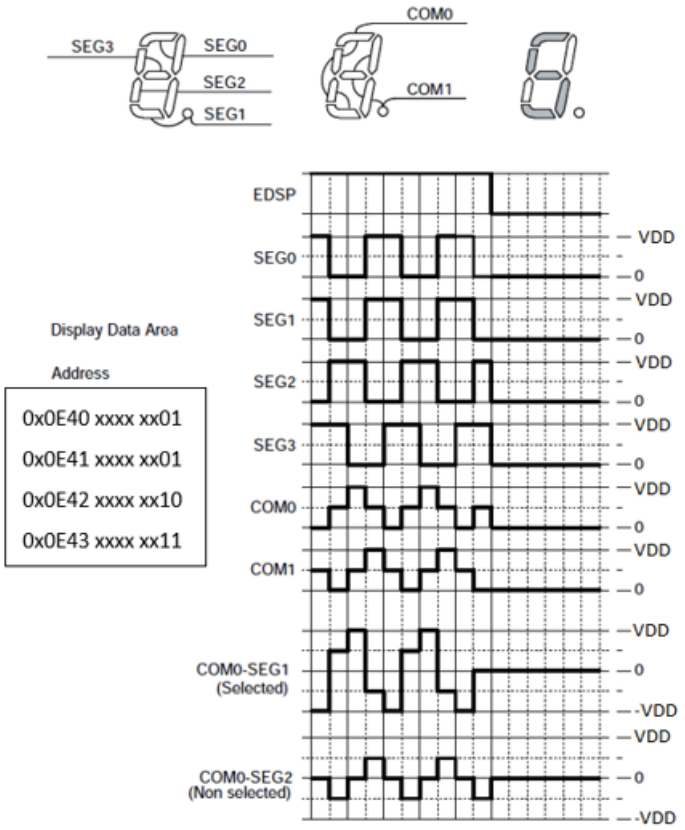



Figure 11.8 LCD drive waveform examples for 1/2duty (1/2bias) LCD

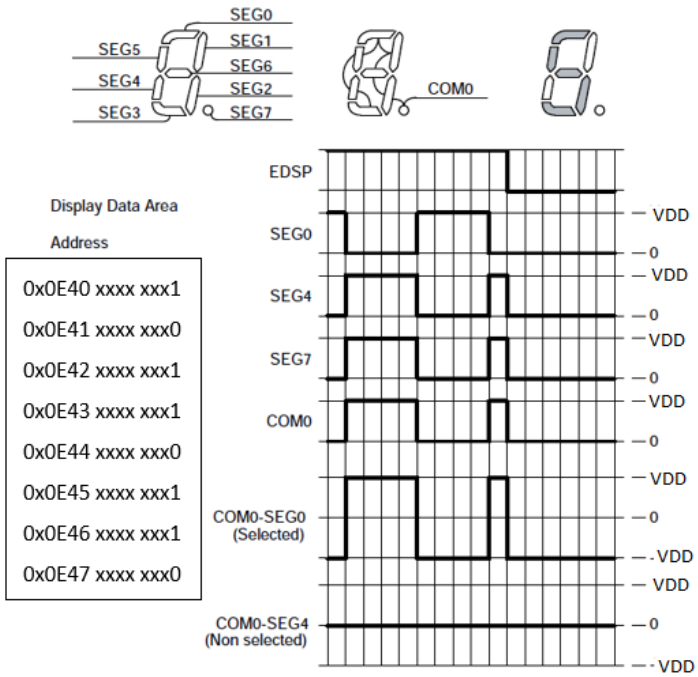



Figure 11.9 LCD drive waveform examples for static LCD

# 12.Asynchronous Serial Interface (UART)

MQ6812/MQ6821 contains 1 channels of asynchronous serial interfaces (UART).This chapter describes asynchronous serial interface 1 (UART1), as shown in Table 12.1 and Table 12.2.

| | UARTxCR1 (Address) | UARTxCR2 (Address) | UARTxDR (Address) | UARTxSR (Address) | RDxBUF (Address) | TDxBUF (Address) |
|---|---|---|---|---|---|---|
| UART1 | UART1CR1 (0x0F54) | UART1CR2 (0x0F55) | UART1DR (0x0F56) | UART1SR (0x0F57) | RD1BUF (0x0F58) | TD1BUF (0x0F58) |

Table 12.1 SFR Address Assignment

| | Serial Data Input Pin | Serial Data Output Pin |
|---|---|---|
| UART1 | RXD1 pin | TXD1 pin |

Table 12.2 Pin Names

## 12.1 Control

UART1 is controlled by the low power consumption registers (POFFCR1), UART1 control registers 1 and 2 (UART1CR1 and UART1CR2) and the UART1 baud rate register (UART1DR). The operating status can be monitored using the UART status register (UART1SR).

Low Power Consumption Register 1

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

| | | |
|---|---|---|
| SBI0EN | I2C0 control | 0: Disable<br>1: Enable |
| UART1EN | UART1 control | 0: Disable<br>1: Enable |

UART1 Control Register 1

| UART1CR1 (0x0F54) | 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 |

| | | | | |
|---|---|---|---|---|
| TXE | Transmit operation | 0: Disable<br>1: Enable | | |
| RXE | Receive operation | 0: Disable<br>1: Enable | | |
| STOPBT | Transmit stop bit length | 0: 1 bit<br>1: 2 bits | | |
| TA0MCAP | Pulse width measurement mode control | 0: Double edge capture<br>1: Single edge capture | | |
| EVEN | Parity selection | 0: Odd-numbered parity<br>1: Even number parity | | |
| PE | Parity addition | 0: No parity<br>1: Parity added | | |
| IRDASEL | TXD pin output selectin | 0: UART output<br>1: IrDA output | | |
| BRG | Transfer base clock selection | | When SYSCR2<SYSCK> is "0" | When SYSCR2<SYSCK> is "1" |
| | | 0 | fcgck | fs |
| | | 1 | TCA0 output | |

*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 "12.3 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.*

### UART1 Control Register 2

| UART1CR2 (0x0F55) | 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<br>01: 1 x (UART1DR + 1) / (Transfer base clock frequency) [s]<br>10: 2 x (UART1DR + 1) / (Transfer base clock frequency) [s]<br>11: 4 x (UART1DR + 1) / (Transfer base clock frequency) [s] | | |
| STOPBR | Receive stop bit length | 0: 1 bit<br>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 "12.7.1 Transfer baud rate calculation method".*

*Note 3): For details of the RXDNC noise rejection time, refer to "12.9 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 "12.3 Protection to Prevent UART1CR1 and UART1CR2 Registers from Being Changed ".*

### UART1 Baud Rate Register

| UART1DR (0x0F56) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | UART1DR7 | UART1DR6 | UART1DR5 | UART1DR4 | UART1DR3 | UART1DR2 | UART1DR1 | UART1DR0 |
| 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 UART1CR1<RXE> and UART1CR1<TXE> to "0" before changing UART1DR. For the set values, refer to "12.7 Transfer Baud Rate".*

*Note 2): When UART1CR1<BRG> is set to the TCA0 output, the value set to UART1DR has no meaning.*

*Note 3): When the STOP, IDLE0 or SLEEP0*

UART1 Status Register

| UAR01SR (0x0F57) | 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<br>1: Parity error |
| RFERR | Framing error flag | 0: No framing error<br>1: Framing error |
| OERR | Overrun error flag | 0: No overrun error<br>1: Overrun error |
| RBSY | Receive busy flag | 0: Before receiving or end of receiving<br>1: On receiving |
| RBFL | Receive buffer full flag | 0: Receive buffer empty<br>1: Receive buffer full |
| TBSY | Transmit busy flag | 0: Before transmission or end of transmission<br>1: On transmission |
| TBFL | Transmit buffer full flag | 0: Transmit buffer empty<br>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.*

UART1 Receive Data Register

| RD1BUF (0x0F58) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | RD1DR7 | RD1DR6 | RD1DR5 | RD1DR4 | RD1DR3 | RD1DR2 | RD1DR1 | RD1DR0 |
| 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.*

UART1 Transmit Data Register

| TD1BUF (0x0F58) | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Bit Symbol | TD1DR7 | TD1DR6 | TD1DR5 | TD1DR4 | TD1DR3 | TD1DR2 | TD1DR1 | TD1DR0 |
| 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.*

## 12.2 Low Power Consumption Function

UART1 has a low power consumption register (POFFCR1) that saves power consumption when the UART function is not used.

Setting POFFCR1 <UART1EN> to "0" disables the basic clock supply to UART1 to save power. Note that this renders the UART unusable. Setting POFFCR1 <UART1EN> to "1" enables the basic clock supply to UART1 and renders the UART usable.

After reset, POFFCR1 <UART1EN> is initialized to "0", and this renders the UART unusable. When using the UART for the first time, be sure to set POFFCR1 <UART1EN> to "1" in the initial setting of the program (before the UART control register is operated).

Do not change POFFCR1 <UART1EN> to "0" during the UART operation, otherwise UART1 may operate unexpectedly.

## 12.3 Protection of UART1CR1 and UART1CR2 Registers from Being Changed

MQ6812/MQ6821 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 UART1CR1 and UART1CR2 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 | | | |
|---|---|---|---|---|---|
| | | UART1CR1<TXE> | UART1SR<TBSY> | UART1CR1<RXE> | UART1SR<RBSY> |
| UART1CR1<STOPBT> | Transmit stop it length | Both of thease bits are "0" | | - | - |
| UART1CR1<EVEN> | Parity selection | All of these bits are "0" | | | |
| UART1CR1<PE> | Parity selection | | | | |
| UART1CR1<IRDASEL> | TXD pin output selection | Both of thease bits are "0" | | - | - |
| UART1CR1<BRG> | Transfer base clock selection | All of these bits are "0" | | | |
| UART1CR2<RTSEL> | Selection of number of RT clocks | | | | |
| UART1CR2<RXDNC> | Selection of RXD pin input noise rejection time | - | - | All of these bits are "0" | |
| UART1CR2<STOPBR> | Receive stop bit length | - | - | All of these bits are "0" | |

Table 12.3 Changing of UART1CR1 and UART1CR2

## 12.4 Activation of STOP, IDLE0 or SLEEP0 Mode

### 12.4.1 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 12.4. 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 |
|---|---|---|---|---|---|---|---|---|
| UART1CR1 | 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 | - |
| UART1CR2 | - | - | RTSEL | | | RXDNC | | STOPBR |
| | - | - | Hold the value | Hold the value | Hold the value | Hold the value | Hold the value | Hold the value |
| UART1SR | 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 |
| UART1DR | UART1DR7 | UART1DR6 | UART1DR5 | UART1DR4 | UART1DR3 | UART1DR2 | UART1DR1 | UART1DR0 |
| | Hold the value | Hold the value | Hold the value | Hold the value | Hold the value | Hold the value | Hold the value | Hold the value |
| RD1BUF | RD1DR7 | RD1DR6 | RD1DR5 | RD1DR4 | RD1DR3 | RD1DR2 | RD1DR1 | RD1DR0 |
| | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate |
| TD1BUF | TD1DR7 | TD1DR6 | TD1DR5 | TD1DR4 | TD1DR3 | TD1DR2 | TD1DR1 | TD1DR0 |
| | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate | Indeterminate |

Table 12.4 Transition of Register Status

### 12.4.2 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.

| UART1CR1 <IRDASEL> | IDEL0 or SLEEP0 mode | STOP mode | |
|---|---|---|---|
| | | SYSCR1<OUTEN>="1" | SYSCR1<OUTEN>="0" |
| "0" | H level | H level | Hi-Z |
| "1" | L level | L level | |

Table 12.5 TXD Pin Status When the STOP, IDLE0 or SLEEP0 Mode Is Activated

## 12.5 Transfer Data Format

The UART transfers data composed of the following four elements. The data from the start bit to the stop

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 12.1 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)

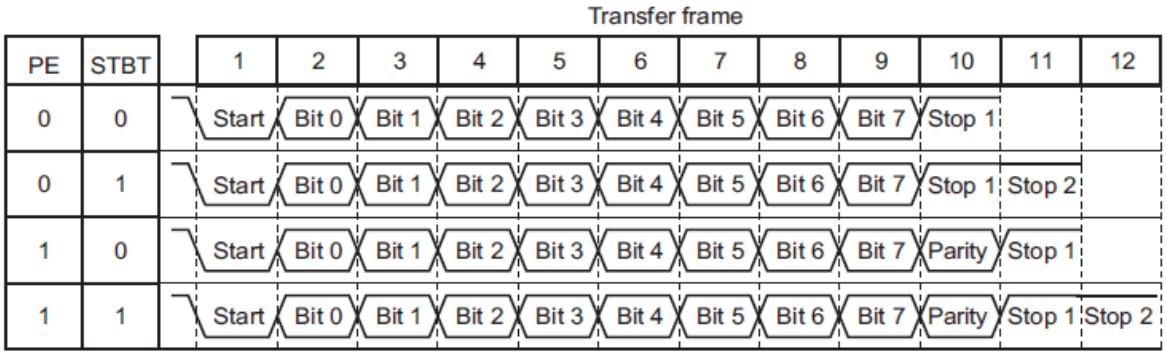



Figure 12.1 Transfer Data Format

## 12.6 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.

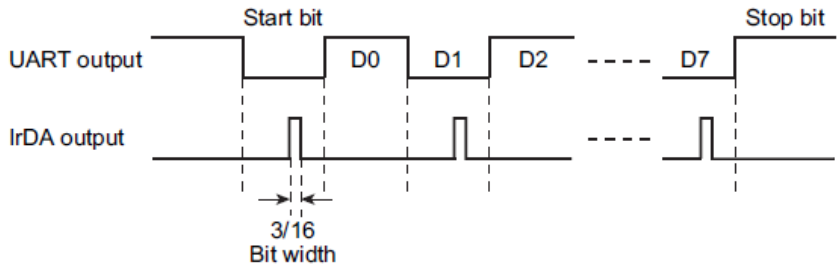



Figure 12.2 Example of Infrared Data Format (Comparison between Normal Output and Ir-DA Output)

## 12.7 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 "11.7.1 Transfer baud rate calculation method".

| Basic baud rate[baud] | Register | Operating frequency | | | | |
|---|---|---|---|---|---|---|
| | | 16MHz | 8MHz | 4MHz | 2MHz | 1MHz |
| 128000 | UART1DR[7:0] | 0x07 | 0x03 | 0x01 | 0x00 | - |
| | RTSEL[2:0] | 0y011 | 0y011 | 0y011 | 0y011 | - |
| | Error | (+0.81%) | (+0.81%) | (+0.81%) | (+0.81%) | - |
| 115200 | UART1DR[7:0] | 0x08 | 0x03 | 0x01 | 0x00 | - |
| | RTSEL[2:0] | 0y011 | 0y100 | 0y100 | 0y100 | - |
| | Error | (-0.44%) | (+2.12%) | (+2.12%) | (+2.12%) | - |
| 76800 | UART1DR[7:0] | 0x0C | 0x06 | 0x02 | - | - |
| | RTSEL[2:0] | 0y000 | 0y010 | 0y100 | - | - |
| | Error | (+0.16%) | (-0.79%) | (+2.12%) | - | - |
| 62500 | UART1DR[7:0] | 0x0F | 0x07 | 0x03 | 0x01 | 0x00 |
| | RTSEL[2:0] | 0y000 | 0y000 | 0y000 | 0y000 | 0y000 |
| | Error | 0% | 0% | 0% | 0% | 0% |
| 57600 | UART1DR[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 | UART1DR[7:0] | 0x19 | 0x0C | 0x06 | 0x02 | - |
| | RTSEL[2:0] | 0y000 | 0y000 | 0y010 | 0y100 | - |
| | Error | (+0.16%) | (+0.16%) | (-0.79%) | (+2.12%) | - |
| 19200 | UART1DR[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 | UART1DR[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 | UART1DR[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 | UART1DR[7:0] | - | 0xCF | 0x67 | 0x33 | 0x19 |
| | RTSEL[2:0] | - | 0y000 | 0y000 | 0y000 | 0y000 |
| | Error | - | (+0.16%) | (+0.16%) | (+0.16%) | (+0.16%) |
| 1200 | UART1DR[7:0] | - | - | 0xCF | 0x67 | 0x33 |
| | RTSEL[2:0] | - | - | 0y000 | 0y000 | 0y000 |
| | Error | - | - | (+0.16%) | (+0.16%) | (+0.16%) |

### 12.7.1 Transfer Baud Rate Calculation Method

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 12.3.

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 |

| RTSEL | Number of RT clocks | | | | | | | | | | | | Generated baud rate |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | |
| 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 12.3 Fine Adjustment of Baud Rate Clock Using UART11R2 <RTSEL>

#### 12.7.1.1 Calculation of Set Values of UART1CR2 <RTSEL> and UART1DR

The set value of UART1DR for an operating frequency and baud rate can be calculated using the calculation formula shown in Figure 12.4. For example, to generate a basic baud rate of 38400 [baud] with fcgck=4 [MHz], calculate the set value of UART1DR for each setting of UART1CR2 <RTSEL> and compensate the calculated value to a positive number to obtain the generated baud rate as shown in Figure 12.5. Basically, select the set value of UART1CR2 <RTSEL> that has the smallest baud rate error from among the generated baud rates. In Figure 12.5, the setting of UART1CR2 <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 | $UARTDR = \frac{fcgck\ [Hz]}{16 \times A\ [baud]} - 1$ |
| 001 | $UARTDR = \frac{fcgck\ [Hz]}{16.5 \times A\ [baud]} - 1$ |
| 010 | $UARTDR = \frac{fcgck\ [Hz]}{15 \times A\ [baud]} - 1$ |
| 011 | $UARTDR = \frac{fcgck\ [Hz]}{15.5 \times A\ [baud]} - 1$ |
| 100 | $UARTDR = \frac{fcgck\ [Hz]}{17 \times A\ [baud]} - 1$ |

Figure 12.4 UART1DR Calculation Method (When BRG Is Set to fcgck)

| RTSEL | UARTDR calculation | Generated baud rate |
|---|---|---|
| 000 | $UARTDR = \frac{4000000\ [Hz]}{16 \times 38400\ [baud]} - 1 \approx 6$ | $\frac{4000000\ [Hz]}{16 \times (6+1)} = 35714\ [baud]\ (-6.99\%)$ |
| 001 | $UARTDR = \frac{4000000\ [Hz]}{16.5 \times 38400\ [baud]} - 1 \approx 5$ | $\frac{4000000\ [Hz]}{16.5 \times (5+1)} = 40404\ [baud]\ (+5.22\%)$ |
| 010 | $UARTDR = \frac{4000000\ [Hz]}{15 \times 38400\ [baud]} - 1 \approx 6$ | $\frac{4000000\ [Hz]}{15 \times (6+1)} = 38095\ [baud]\ (-0.79\%)$ |
| 011 | $UARTDR = \frac{4000000\ [Hz]}{15.5 \times 38400\ [baud]} - 1 \approx 6$ | $\frac{4000000\ [Hz]}{15.5 \times (6+1)} = 36866\ [baud]\ (-3.99\%)$ |
| 100 | $UARTDR = \frac{4000000\ [Hz]}{17 \times 38400\ [baud]} - 1 \approx 5$ | $\frac{4000000\ [Hz]}{17 \times (5+1)} = 39216\ [baud]\ (+2.12\%)$ |

Figure 12.5 Example of UART1DR Calculation

## 12.8 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 12.6).

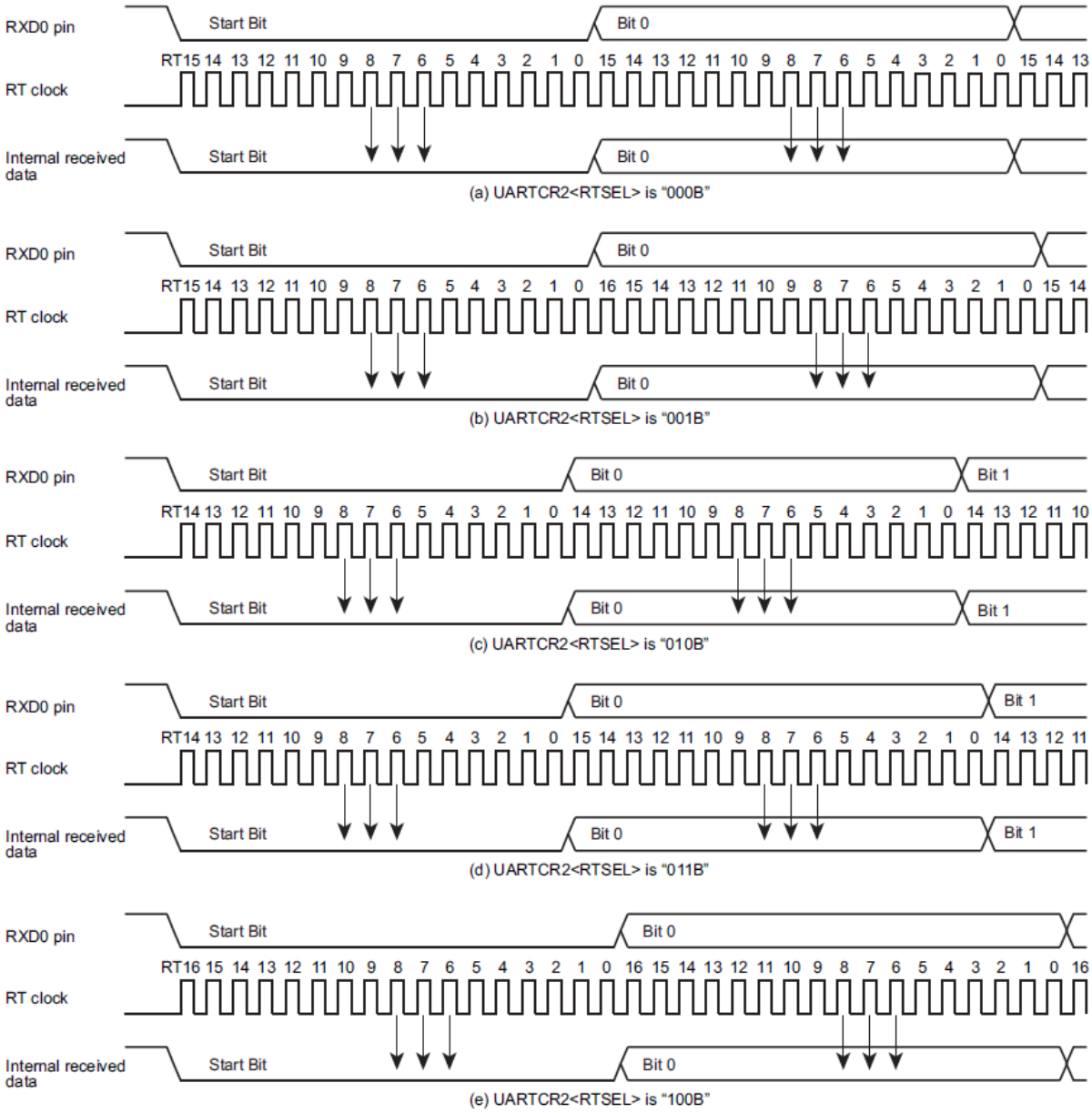



Figure 12.6 Data Sampling in Each Case of UART1CR2 <RTSEL>

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.

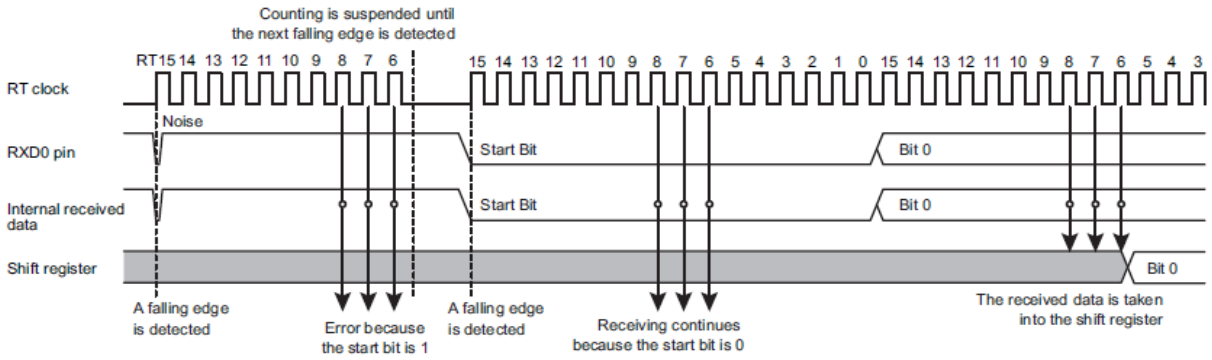



Figure 12.7 Start Bit Sampling

## 12.9 Received 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 12.6.

| RXDNC | Noise rejection time(s) | Time of pulses to be regarded as signals |
|---|---|---|
| 00 | NO noise rejection | - |
| 01 | (UART1DR+1)/(Transfer base clock frequency) | 2x (UART1DR+1)/(Transfer base clock frequency) |
| 10 | 2x (UART1DR+1)/(Transfer base clock frequency) | 4x (UART1DR+1)/(Transfer base clock frequency) |
| 11 | 4x (UART1DR+1)/(Transfer base clock frequency) | 8x (UART1DR+1)/(Transfer base clock frequency) |

Table 12.6 Received Data Noise Rejection Time

*Note): The transfer base clock frequency is the clock frequency selected at UARTCR1 <BRG>.*

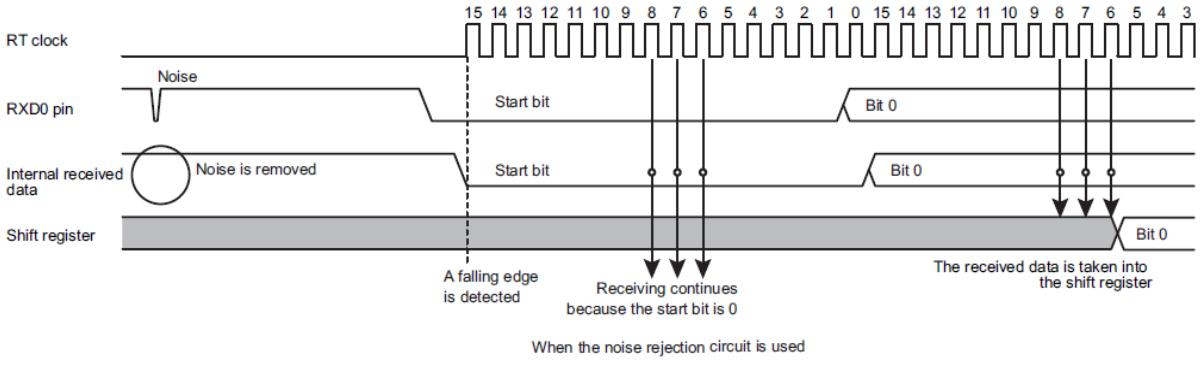



Figure 12.8 Received Data Noise Rejection

## 12.10 Transmit/Receive Operation

### 12.10.1 Data Transmit Operation

Set UART1CR1 <TXE> to "1". Check UART1SR <TBFL> = "0", and then write data into TD1BUF (transmit data buffer). Writing data into TD1BUF sets UART1SR<TBFL> to "1", transfers the data to the transmit shift register, and outputs the data sequentially from the TXD1 pin. The data output includes a start bit, stop bits whose number is specified in UART1CR1 <STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART1CR1 <BRG>, UART1CR2 <RTSEL> and UART1DR. When data transmission starts, the transmit buffer full flag UART1SR <TBFL> is cleared to "0" and an INTTXD1 interrupt request is generated.

*Note 1): After data is written into TD1BUF, if new data is written into TD1BUF 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 12.7, the TXD1 pin output is fixed at the L or H level according to the setting of UART1CR1 <IRDASEL>.*

| Condition | TXD1 pin output | |
|---|---|---|
| | IRDASEL="0" | IRDASEL="1" |
| When UART1CR1<TXE> is "0" | H level | L level |
| From when "1" is written to UART1CR1<TXE> to when the transmitted data is written to TD1BUF | | |
| When the STOP, IDLE0 or SLEEP0 mode is active | | |

Table 11.7 TXD1 Pin Output

### 12.10.2 Data Receive Operation

Set UART1CR1 <RXE> to "1". When data is received via the RXD1 pin, the received data is transferred to RD1BUF (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 RD1BUF (receive data buffer). Then the receive buffer full flag UART1SR <RBFL> is set and an INTRXD1 interrupt request is generated. Set the data transfer baud rate using UART1CR1 <BRG>, UART11R2 <RTSEL> and UART1DR.

If an overrun error occurs when data is received, the data is not transferred to RD1BUF (receive data buffer) but discarded; data in the RD1BUF is not affected.

## 12.11 Transmit/Receive Operation

### 12.11.1 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 INTRXD1 interrupt request is generated.

If UART1SR <PERR> is "1" when UART1SR is read, UART1SR <PERR> will be cleared to "0" when RD1BUF is read subsequently. (The RD1BUF read value becomes undefined.)

If UART1SR <PERR> is set to "1" after UART1SR is read, UART1SR <PERR> will not be cleared to "0" when RD1BUF is read subsequently. In this case, UART1SR <PERR> will be cleared to "0" when UART1SR is read again and RD1BUF is read.

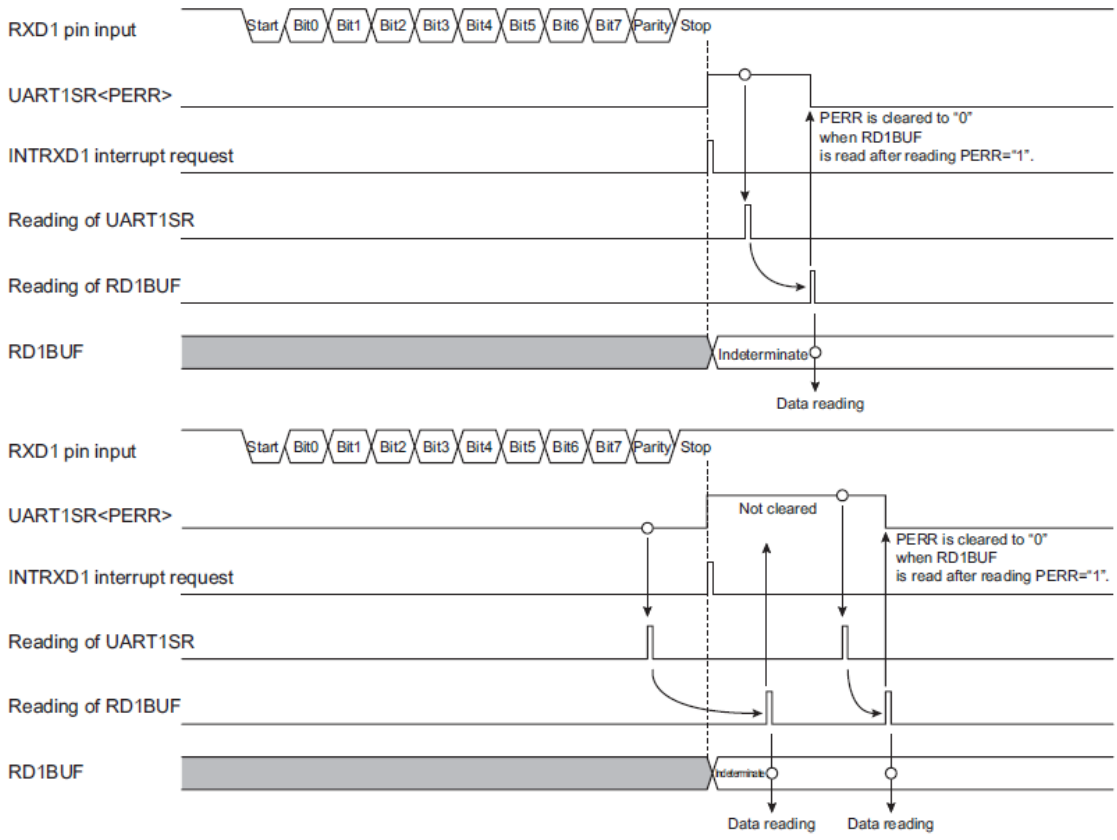



Figure 12.9 Occurrence of Parity Error

### 12.11.2 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 RXD1 pin, the framing error flag UART1SR <FERR> is set to "1". At this

time, an INTRXD1 interrupt request is generated.

If UART1SR <FERR> is "1" when UART1SR is read, UART1SR <FERR> will be cleared to "0" when RD1BUF is read subsequently.

If UART1SR <FERR> is set to "1" after UART1SR is read, UART1SR <FERR> will not be cleared to "0" when RD1BUF is read subsequently. In this case, UART1SR <FERR> will be cleared to "0" when UART1SR is read again and RD1BUF is read.

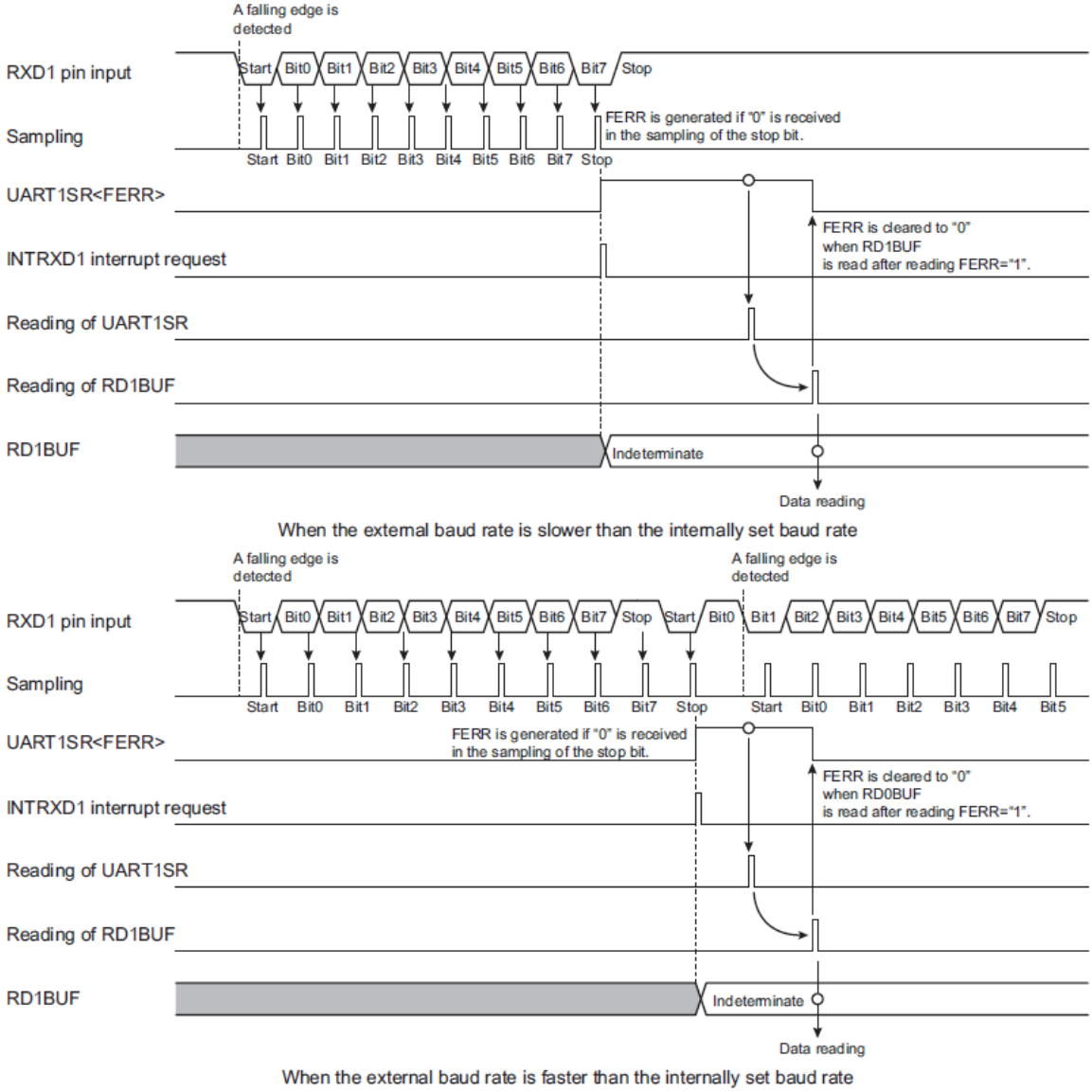



Figure 12.10 Occurrence of Framing Error

### 12.11.3 Overrun Error

If receiving of all data bits is completed before the previous received data is read from RD1BUF, the overrun error flag UART1SR <OERR> is set to "1" and an INTRXD1 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 UART1SR <OERR> is still "1", no INTRXD1 interrupt request is generated, and the received data is discarded. (Figure 12.11)

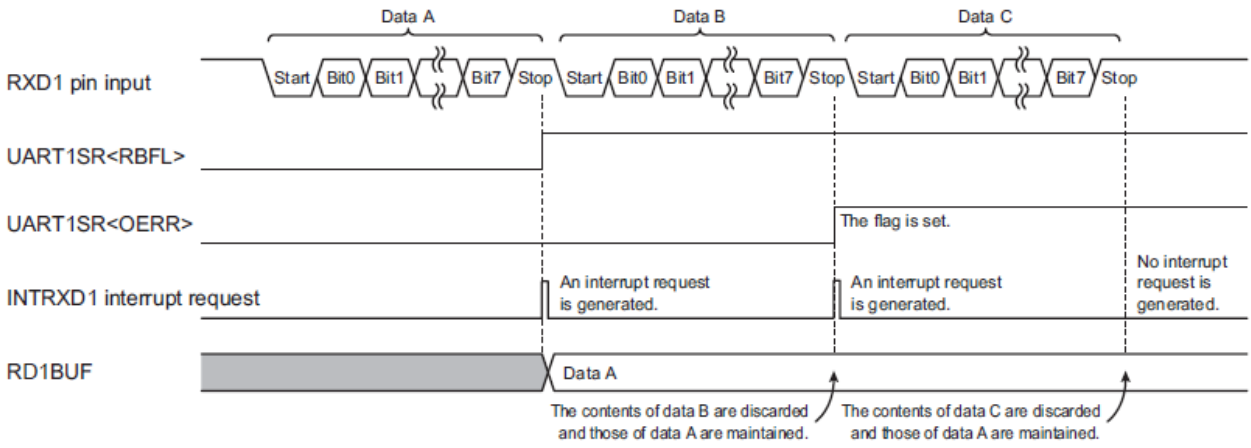



Figure 12.11 Generation of INTRXD1 Interrupt Request

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 UART1SR, they have occurred in the previous received data (the data stored in RD1BUF). (Figure 12.12)

If UART1SR <OERR> is "1" when UART1SR is read, UART1SR <OERR> will be cleared to "0" whenRD1BUF is read subsequently. (Figure 12.13)

If UART1SR <OERR> is set to "1" after UART1SR is read, UART1SR <OERR> will not be cleared to "0" when RD1BUF is read subsequently. In this case, UART1SR <OERR> will be cleared to "0" when UART1SR is read again and RD1BUF is read. (Figure 12.13)

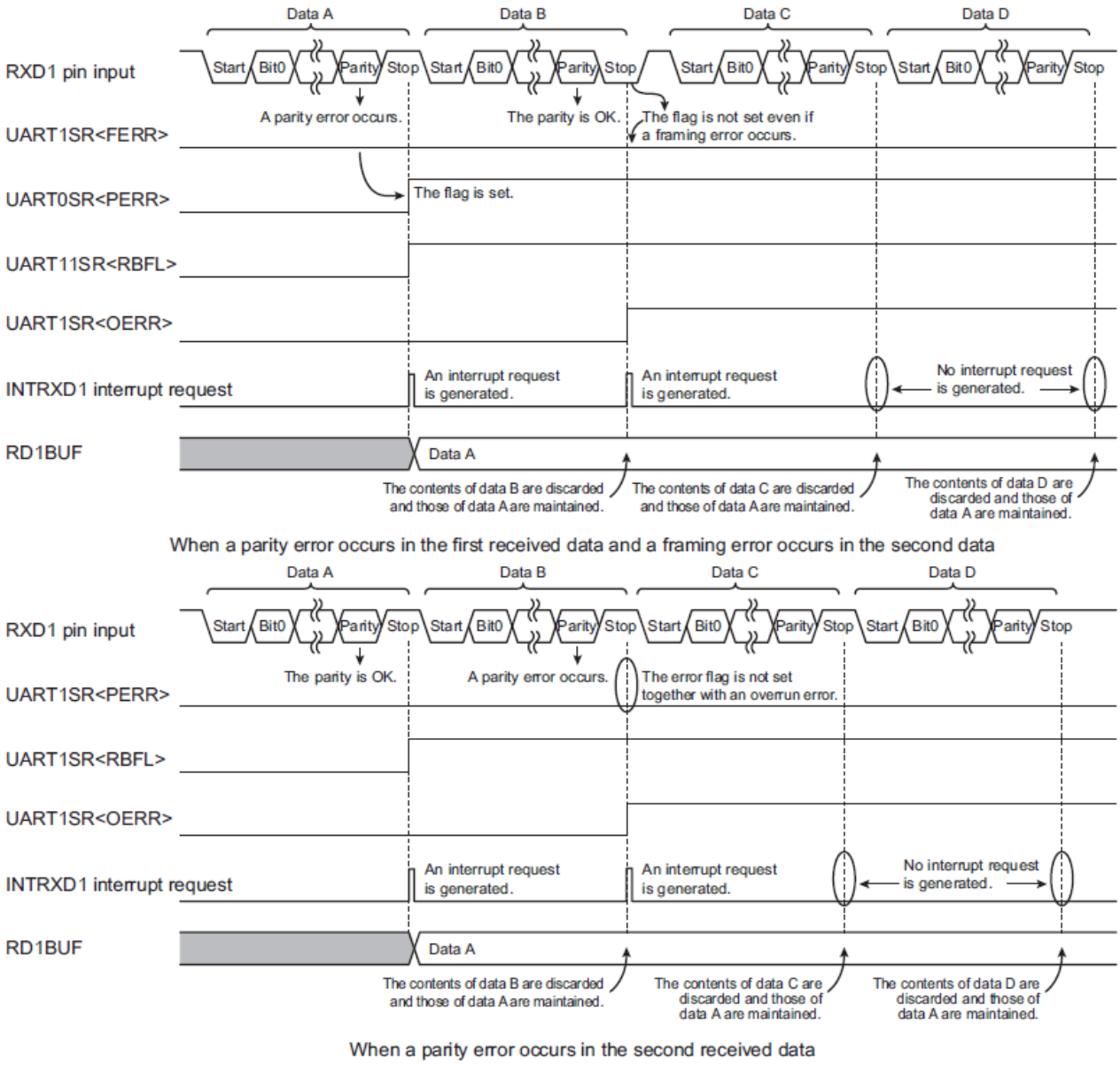



Figure 12.12 Framing/Parity Error Flags When an Overrun Error Occurs

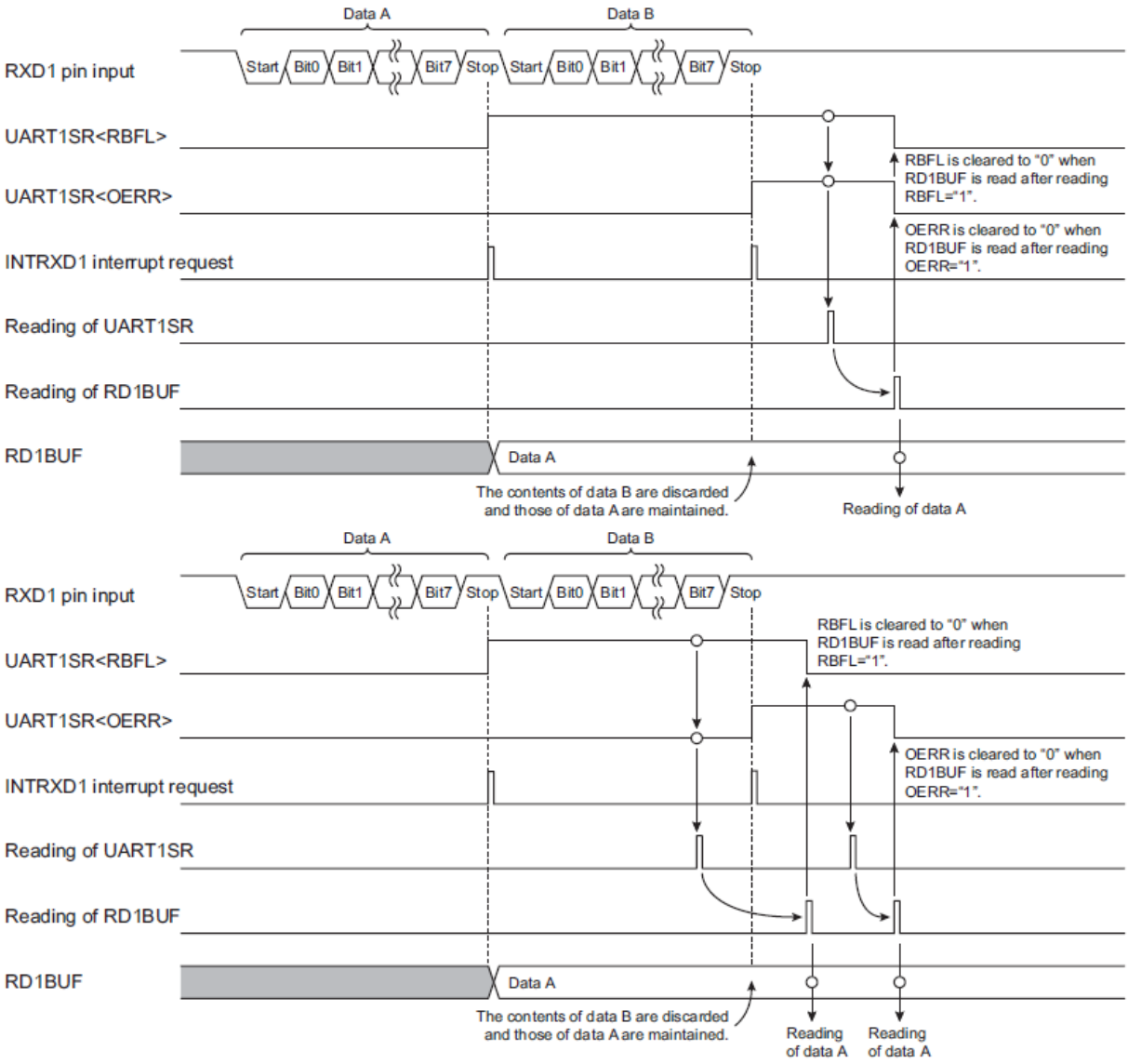



Figure 12.13 Clearance of Overrun Error Flag

### 12.11.4 Recevie Data Buffer Full

Loading the received data in RD1BUF sets UART1SR <RBFL> to "1".

If UART1SR <RBFL> is "1" when UART1SR is read, UART1SR <RBFL> will be cleared to "0" when RD1BUF is read subsequently.

If UART1SR <RBFL> is set to "1" after UART1SR is read, UART1SR <RBFL> will not be cleared to "0" when RD1BUF is read subsequently. In this case, UART1SR <RBFL> will be cleared to "0" when UART1SR is read again and RD1BUF is read.

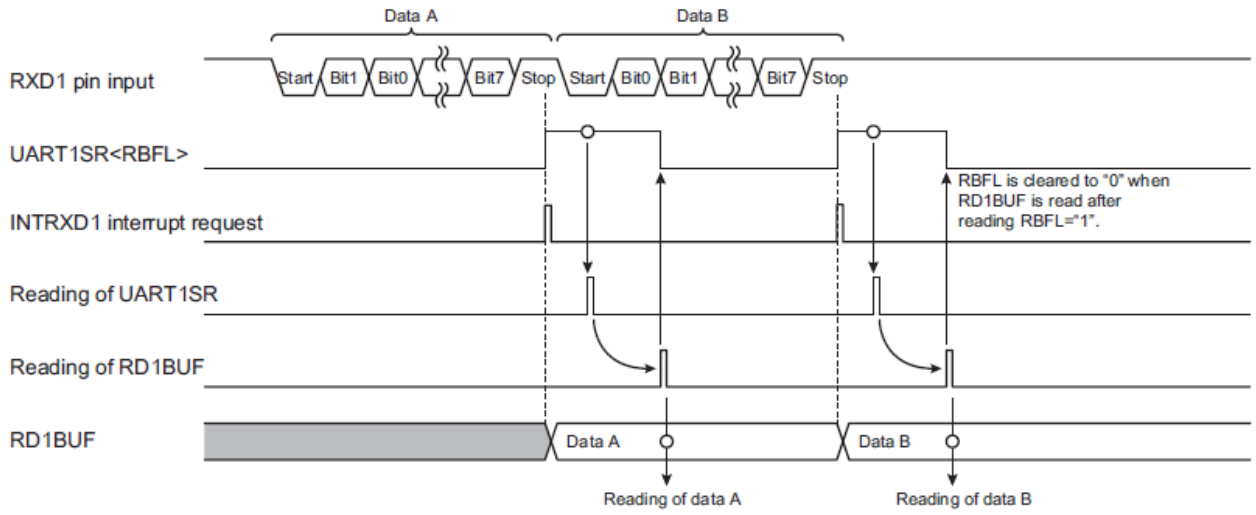



Figure 12.14 Occurrence of Receive Data Buffer Full

### 12.11.5 Transmit Busy Flag

If transmission is completed with no waiting data in TD1BUF (when UART1SR <TBFL>="0"), UART1SR <TBSY> is cleared to "0". When transmission is restarted after data is written into TD1BUF, UART1SR <TBSY> is set to "1". At this time, an INTTXD1 interrupt request is generated.

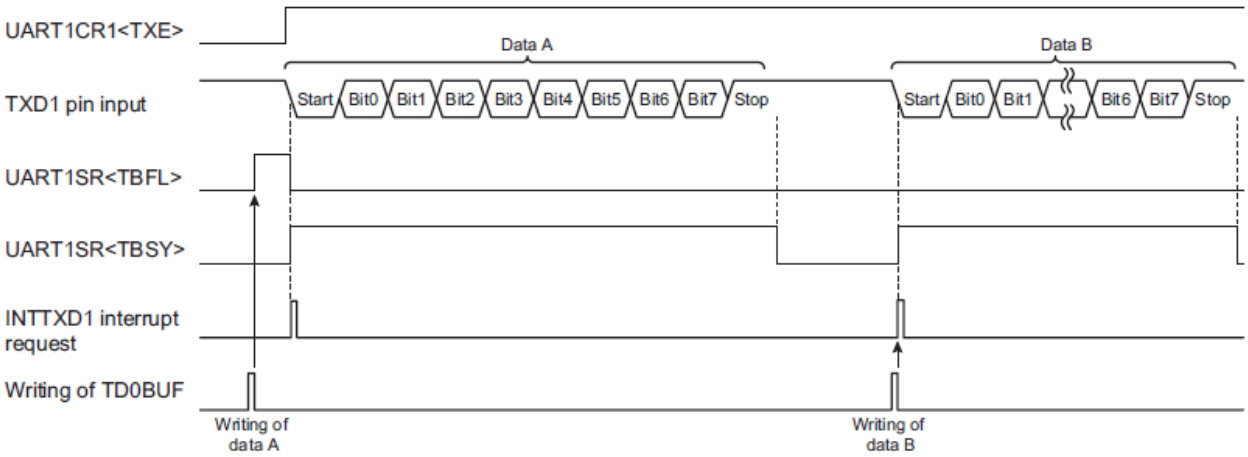



Figure 12.15 Transmit Busy Flag and Occurrence of Transmit Buffer Full

### 12.11.6 Transmit Buffer Full

When TD1BUF has no data, or when data in TD1BUF is transferred to the transmit shift register and transmission is tarted, UART1SR <TBFL> is cleared to "0". At this time, an INTTXD1 interrupt request is generated.

Writing data into TD1BUF sets UART1SR <TBFL> to "1".

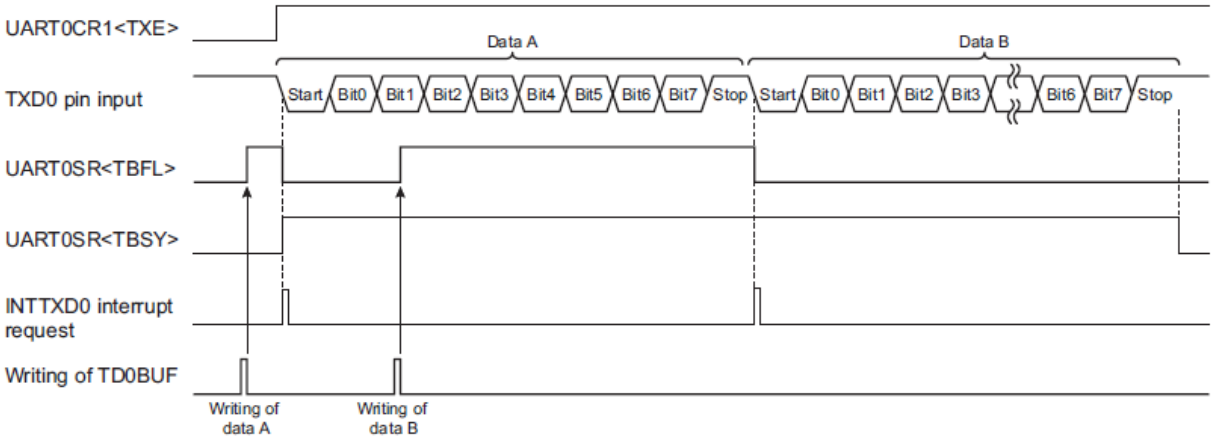



Figure 12.16 Occurrence of Transmit Buffer Full

## 12.12 Receiving Process

Figure 12.17 shows an example of the receiving process. Details of flag judgments in the processing are shown in Table 12.8 and Table 12.9.

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 RD1BUF 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.

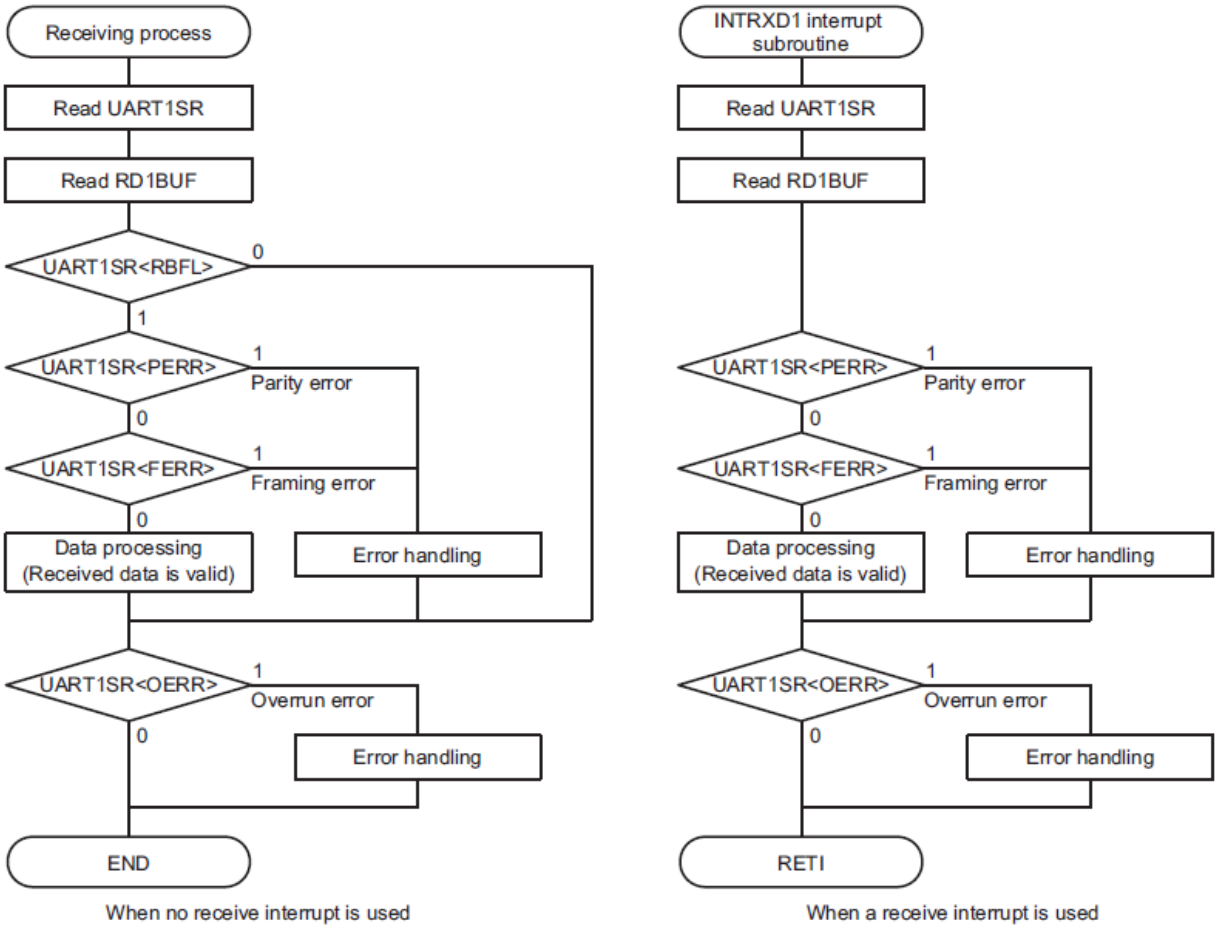



Figure 12.17 Example of Receiving Process

*Note): If multiple interrupts are used in the INTRXD1 interrupt subroutine, the interrupt should be enabled after reading UART1SR and RD1BUF.*

| 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<br>(Receiving of next data is completed in the period from when UART1SR is read to when RD1BUF 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 12.8 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 12.9 Flag Judgments When a Receive Interrupt Is Used

# 13. Flash Memory

MQ6812/MQ6821 has flash memory of 16K x 8 bytes. A write and erase to be performed on flash memory can be controlled in the MCU mode, where 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 MQ6812/MQ6821.

In MCU 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.

## 13.1 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 | | | - | - | - | - | - |
| Read/Write | R/W | | | R | R | R | R | R |
| After reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| | | |
|---|---|---|
| FLSMD | Flash memory command sequence and toggle control | 010: Disable command sequence and toggle execution<br>101: Enable command sequence and toggle execution<br>Others: Reserved |

*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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| CR1EN | FLSCR1 register enable / disable control | 0xD5: Enable a change in the FLSCR1 setting<br>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 | - | - | - | - | - |
| Read/Write | R | R | 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<br>1: FLSCR1 <FLSMD>="101" setting enabled |

*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 7~6 and Bit 4~0..*

## 13.2 Flash Memory Functions

### 13.2.1 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.*

## 13.3 Command Sequence

In MCU mode, the command sequence consists of six commands (JEDEC compatible), as shown in Table 13.1.

| 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 (注 1) | Data (注 1) | 0xF566 | 0xFF | - | |
| 2 | Sector Erase(Partial erase in units of 128 bytes) | 0x#555 | 0xAA | 0x#AAA | 0x55 | 0x#555 | 0x80 | 0x#555 | 0xAA | 0x#AAA | 0x55 | SA (note 2) | 0x20 |
| 3 | Sector Erase(Partial erase in units of 1K bytes) | 0x#555 | 0xAA | 0x#AAA | 0x55 | 0x#555 | 0x80 | 0x#555 | 0xAA | 0x#AAA | 0x55 | SA (note 2) | 0x30 |
| 4 | Chip Erase (All erase) | 0x#555 | 0xAA | 0x#AAA | 0x55 | 0x#555 | 0x80 | 0x#555 | 0xAA | 0x#AAA | 0x55 | 0x#555 | 0x10 |

Table13.1 Command Sequence

*Note 1: Specify the address and data to be written (Refer to Table 13-2 about BA).*

*Note 2: The area to be erased is specified with the upper 5 bits of the address (Refer to Table 13-3 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*

### 13.3.1 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 13-2. 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.

*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.*

### 13.3.2 Sector Erase (128 Byte Partial Erase)

This command erases the flash memory in units of 128 bytes. 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 13-3. For example, to erase 128 byte from 0xE000 through 0xE07F in the code area, set FLSCR1<FAREA> to "101", set "0xD5" on FLSCR2<CR1EN>, and then specify either 0xE000 or 0xE07F as the 6th bus write cycle.

The time needed to erase 128 bytes is 5 ms maximum. The next command sequence cannot be executed if an ongoing erase operation is not completed. During the operation, CPU would enter IDEL 1/2 mode until the erase operation.

Data in the erased area is 0xFF.

### 13.3.3 Entry Product ID mode

This command can enter product ID mode. In product ID mode, excute the read instruction, can read out the information(Vendor ID,Flash ID, Security status). erases the entire flash memory.

| Address | Meaning | Value |
|---|---|---|
| 0xF000 | Vendor ID | 0x68 |
| 0xF001 | Flash ID | 0x12 |
| 0xFF7F | Security Status | 0xFF: Flash data security function disable<br>Excpt 0XFF: Flash data security function enable |

**Table13.2 The meaning of address in product ID mode**

### 13.3.4 Out of Product ID mode

his command can exit product ID mode.

## 13.4 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. During the operation, CPU would enter IDEL 1/2 mode until the erase operation.

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 128 bytes( one sector).

*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 possibility that data in the flash memory may be rewritten to an unexpected value.*

# 14.Serial Bus Interface(SBI)/ I2C

MQ6812/MQ6821 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.

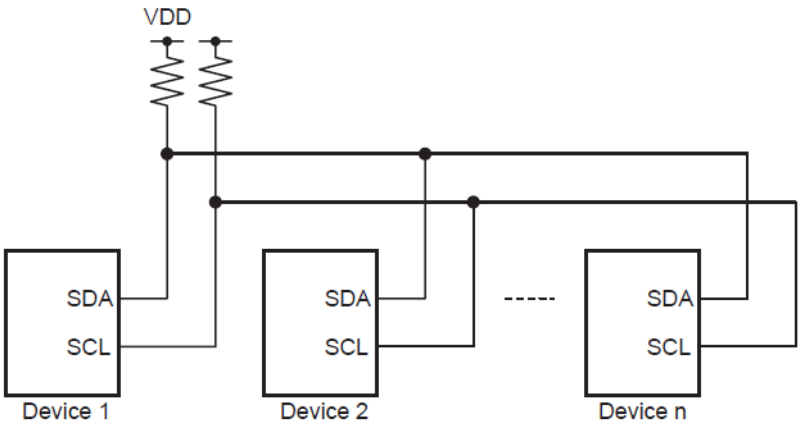



Figure 14.1 Device Connections

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 Figure 14-2.
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

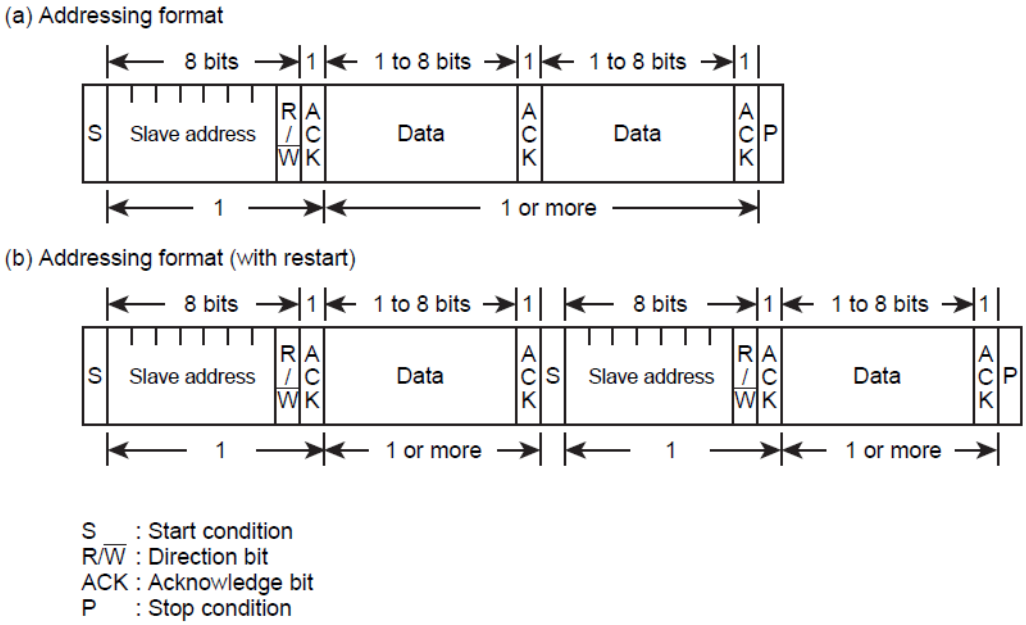



Figure14.2 Data format if I2C bus

### 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.

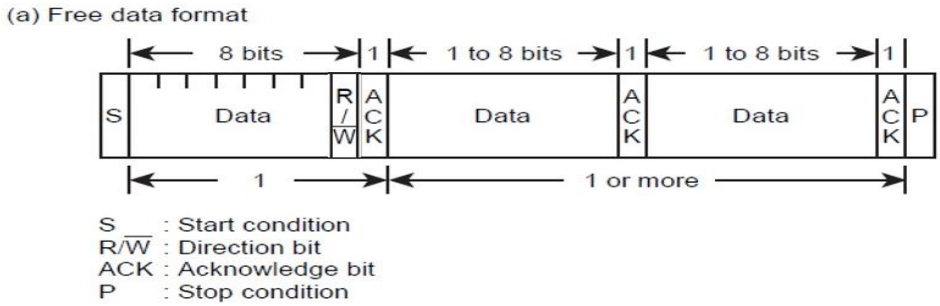



Figure 14.3 Free Data Format

## 14.2 Configuration

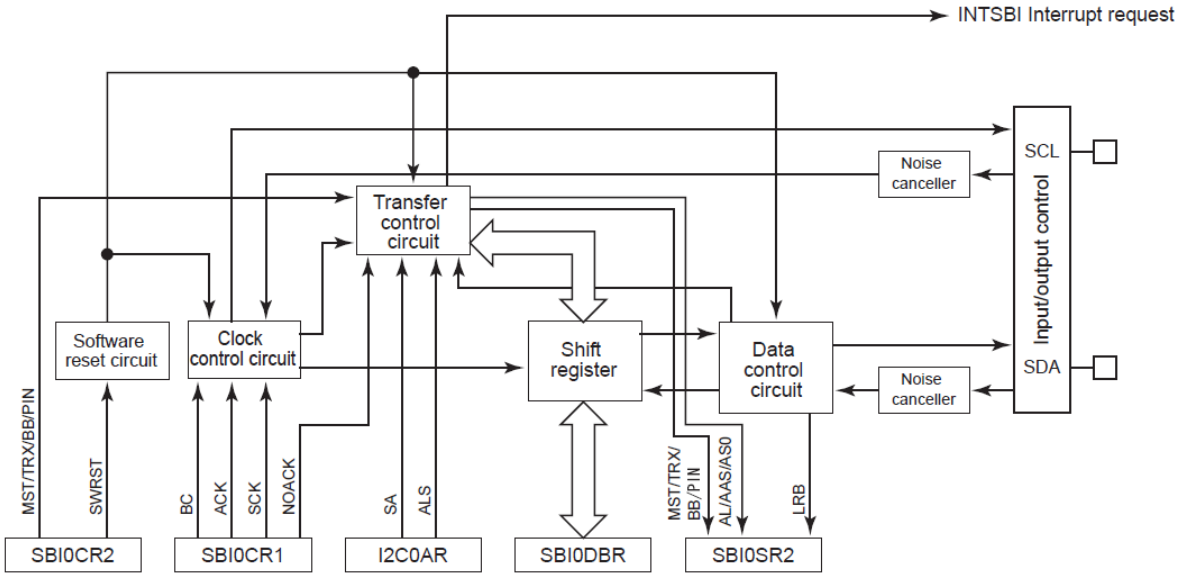



Figure 14.4 Serial Bus Interface0 (SBI0)

## 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)

Other related register

-Low power consumption register1 (POFFCR1)

Set POOFFCR1<SBI0EN> to"1" , before enable SBI/I2C function, unless the SBI/I2C setting is invalid.

-Port P2 Output Control Registe (P2OUTCR)

Set P2OUTCR3 and P2OUTCR4 to"1" , before enable SBI/I2C function

- Port P2 Output Latch Register (P2DR)

Set P2DR to"0" before enable SBI/I2C function, unless the SBI/I2C function would operate unexpectedly.

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 register1

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

| | | |
|---|---|---|
| SBI0EN | $I^2C0$ control | 0:Disable<br>1:Enable |
| UART1EN | UART1 control | 0:Disable<br>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<br>1:Master |
| TRX | Transmitter/receiver selection | 0:Receiver<br>1:Transmitter |
| BB | Start/stop generation | 0:Generate the stop condition(when MST、TRX and PIN are"1")<br>1: Generate the start condition (when MST、TRX and PIN are"1") |
| PIN | Cancel interrupt service request | 0:- (cannot clear this bit by software)<br>1: Cancel interrupt service request |
| SBIM | Serial bus interface operation mode register | 0: Port mode<br>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<br>1:Master |
| TRX | Transmitter/receiver selection status monitor | 0:Receiver<br>1:Transmitter |
| BB | Bus status monitor | 0:Bus free<br>1: Bus busy |
| PIN | Interrupt service requests status monitor | 0:Requesting interrupt service<br>1:Releasing interrupt service |
| AL | Arbitration lost detection monitor | 0: -<br>1:Aritration lost detected |
| AAS | Slave address match detection monitor | 0: -<br>1:Detect slave address match or "GENERAL CALL" |
| AD0 | "GENERAL CALL" detection monitor | 0: -<br>1: Detect "GENERAL CALL" |
| LRB | Last received bit monitor | 0: Last received bit is"0"<br>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.*

I²C 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: I²C bus mode<br>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. Clearing 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.

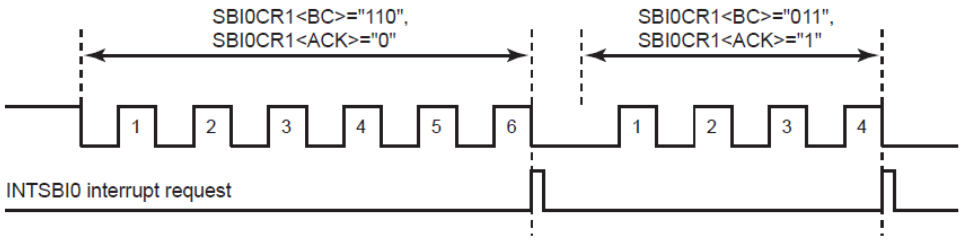



Figure 14.5 Number of clocks for Data transfer and SBI0CR1<BC> and SBI0CR1<ACK>

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 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 |

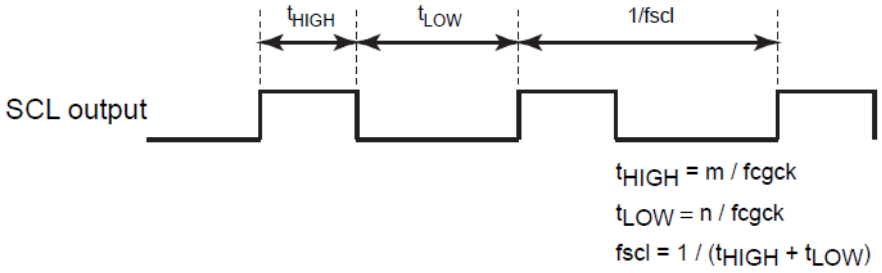



Figure14.6 SCL output

*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.

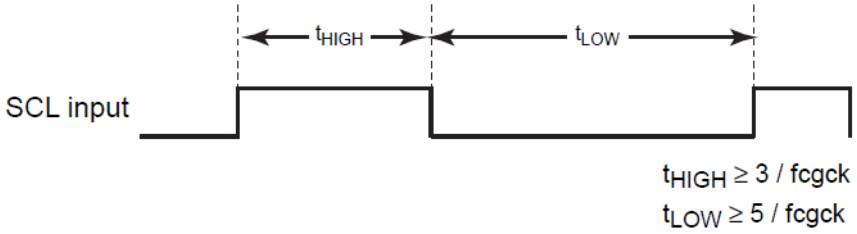



Figure14.7 SCL input

### 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 a bus.

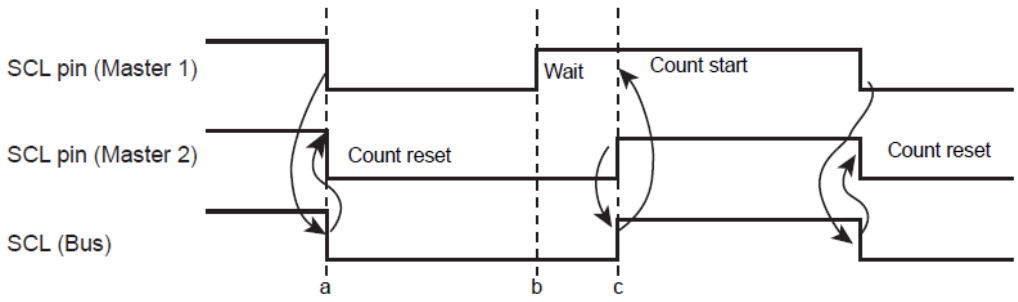



Figure14.8 Example of clock synchronization

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. Table 19-3 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.

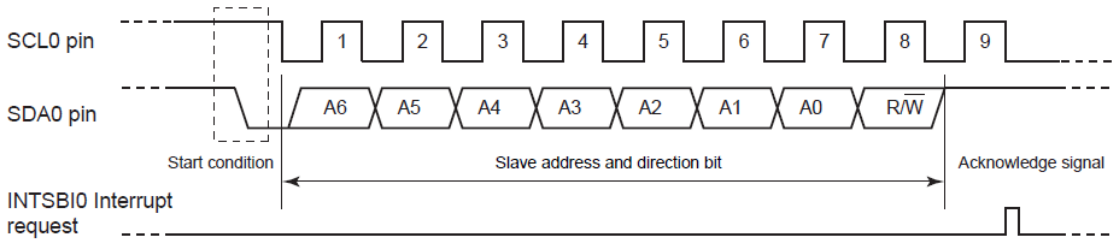



Figure14.9 Generating the start condition and a slave address

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.

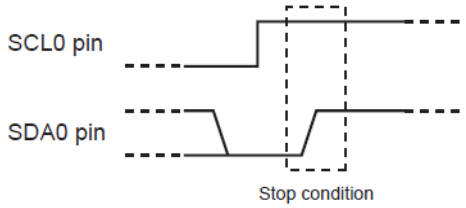



Figure14.10 Stop condition generation

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.

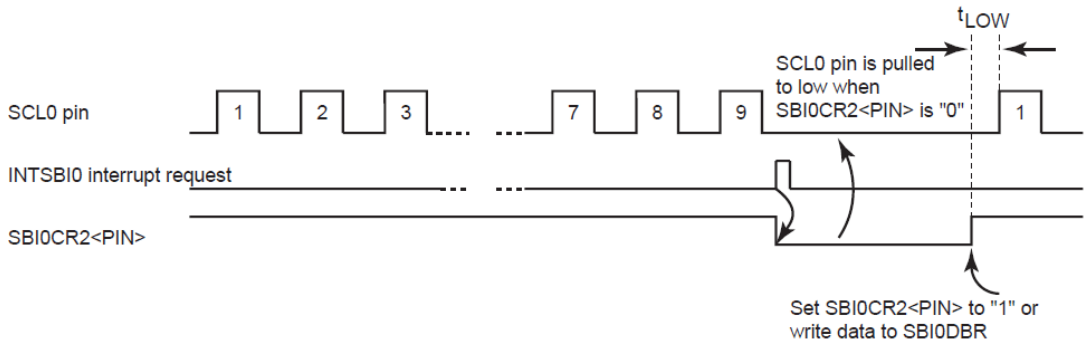



Figure14.11 SBI0CR2<PIN> and SCL0 pin

### 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 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.

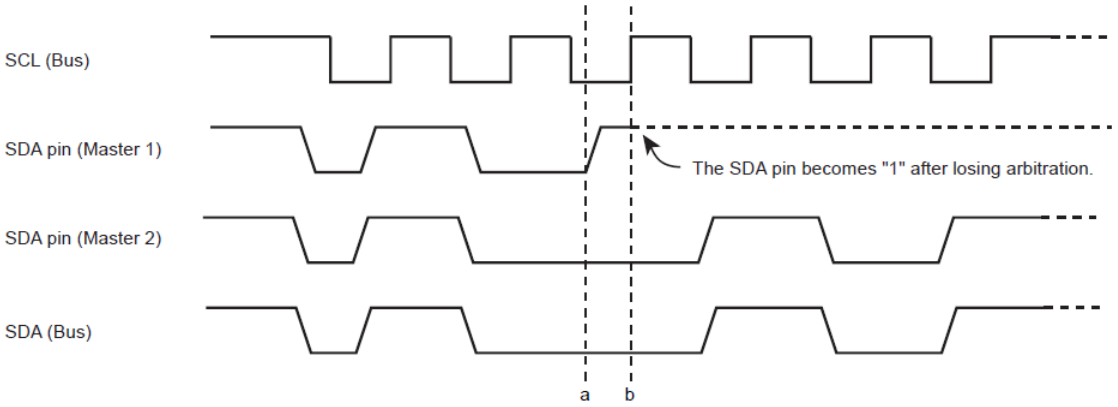



Figure14.12 Arbitration Lost

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.

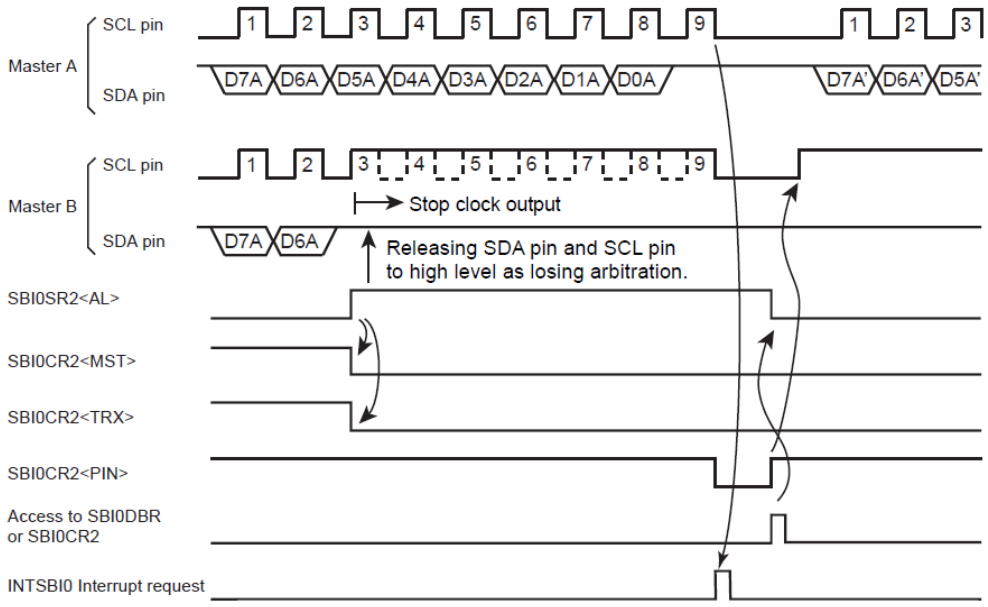



Figure14.13 Example when Master B is a serial bus interface circuit

### 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.

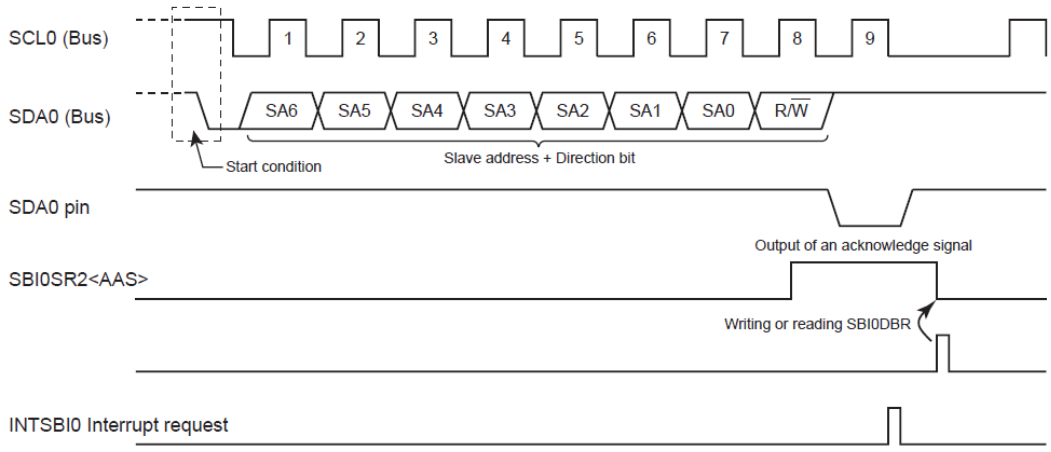



Figure14.14 Change in the slave address match detection monitor

### 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.

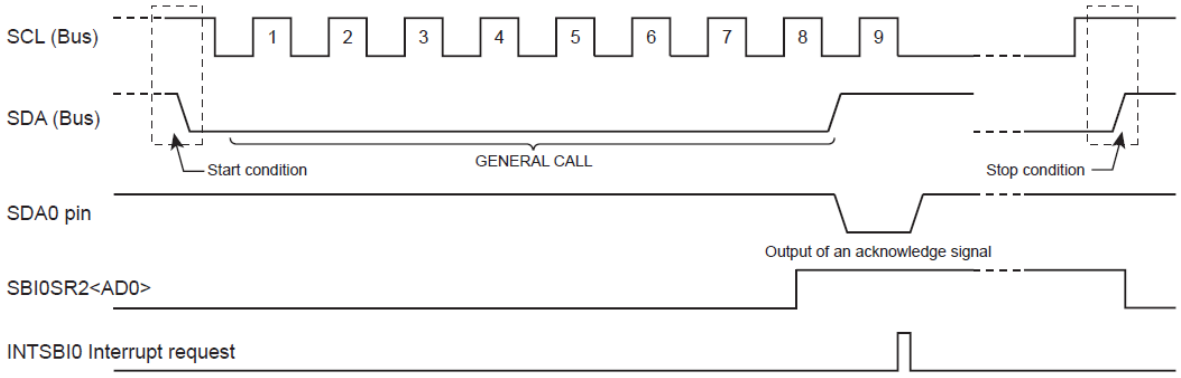



Figure14.15 Changes in the GENERAL CALL detection monitor

### 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>.

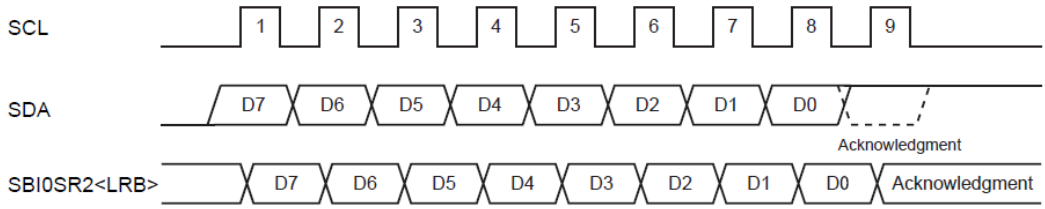



Figure14.16 Changes in the Last received bit monitor

### 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.

*Note1) : 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.*

*Note2): Set the related register before enable SBI/I2C function:*

*-Low power consumption register1 (POFFCR1)*

*Set POOFFCR1<SBI0EN> to"1", before enable SBI/I2C function, unless the SBI/I2C setting is invalid.*

*-Port P2 Output Control Registe (P2OUTCR)*

*Set P2OUTCR3 and P2OUTCR4 to"1", before enable SBI/I2C function*

*- Port P2 Output Latch Register (P2DR)*

*Set P2DR to"0" before enable SBI/I2C function, unless the SBI/I2C function would operate unexpectedly.*

### 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 $t_{HIGH}$.

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.*

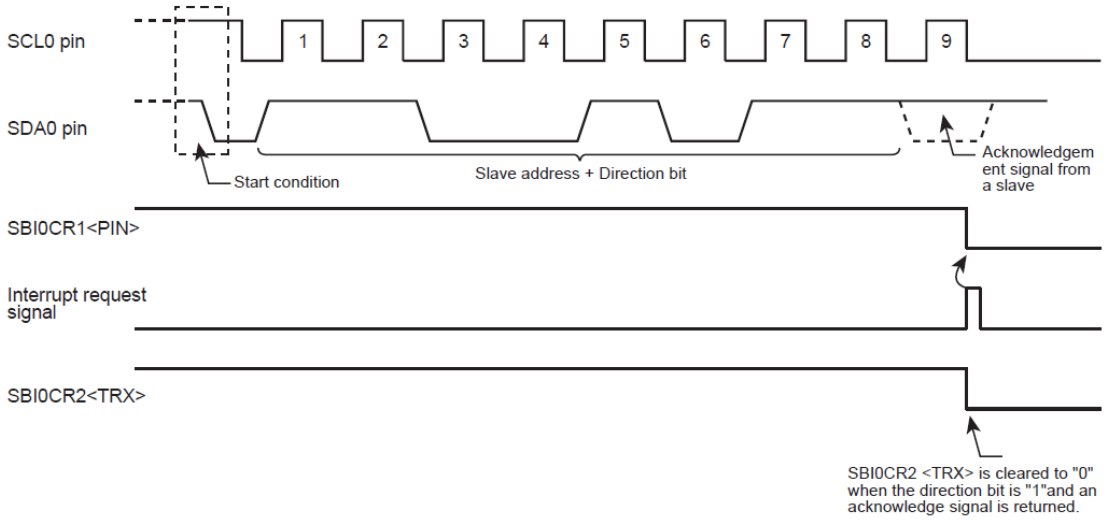



Figure14.17 Generating the start condition and the slave address

### 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 When 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.

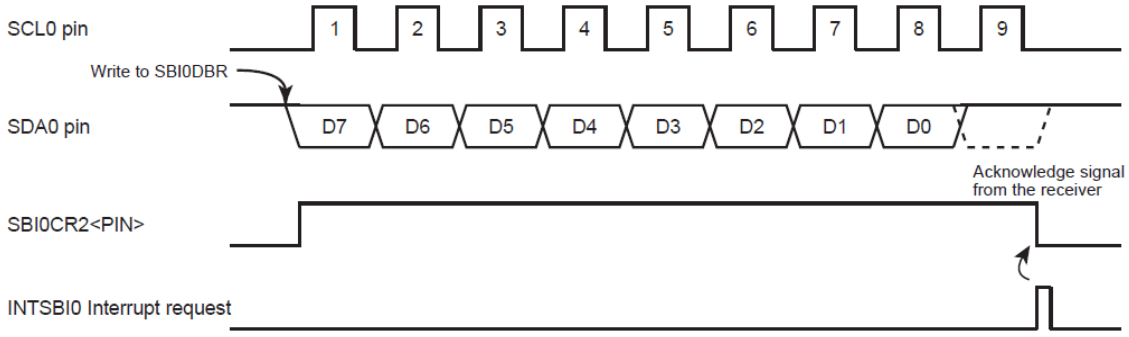



**Figure14.18 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1"**

**(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.

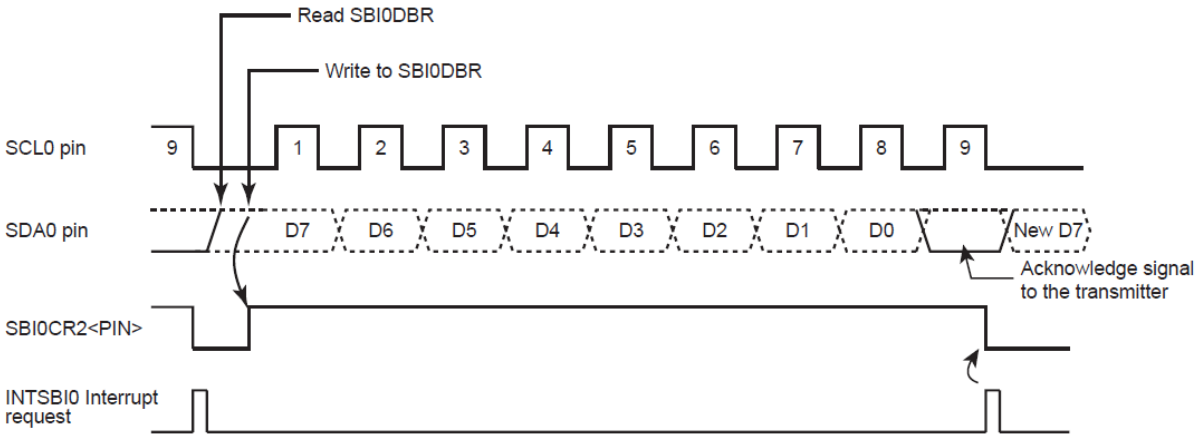



**Figure14.19 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1**

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.

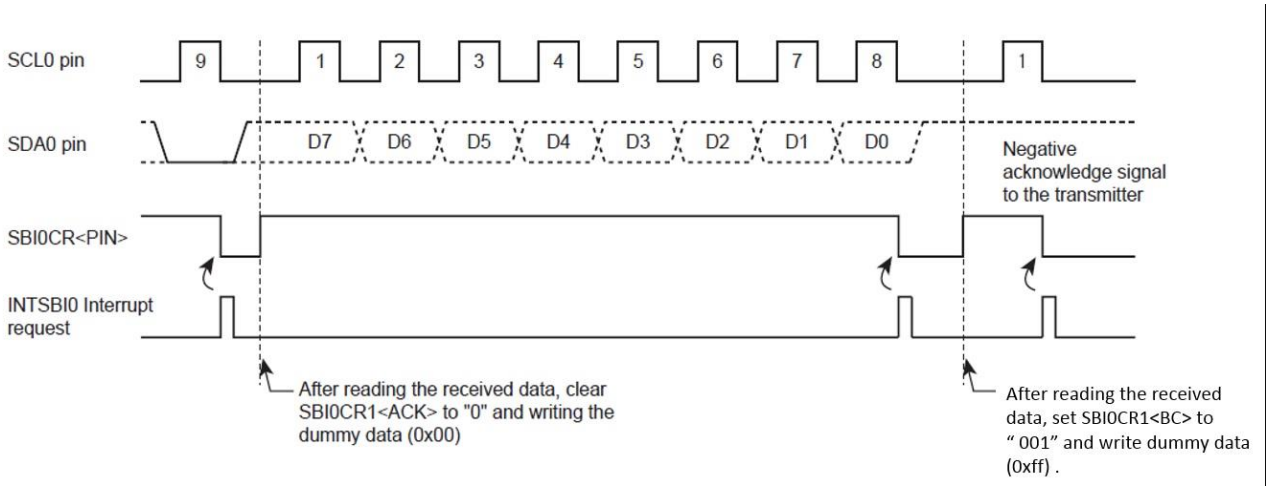



Figure14.20 Termination of Data Transfer in the Master Receiver Mode

#### 14.5.3.2 When 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 $t_{HIGH}$.

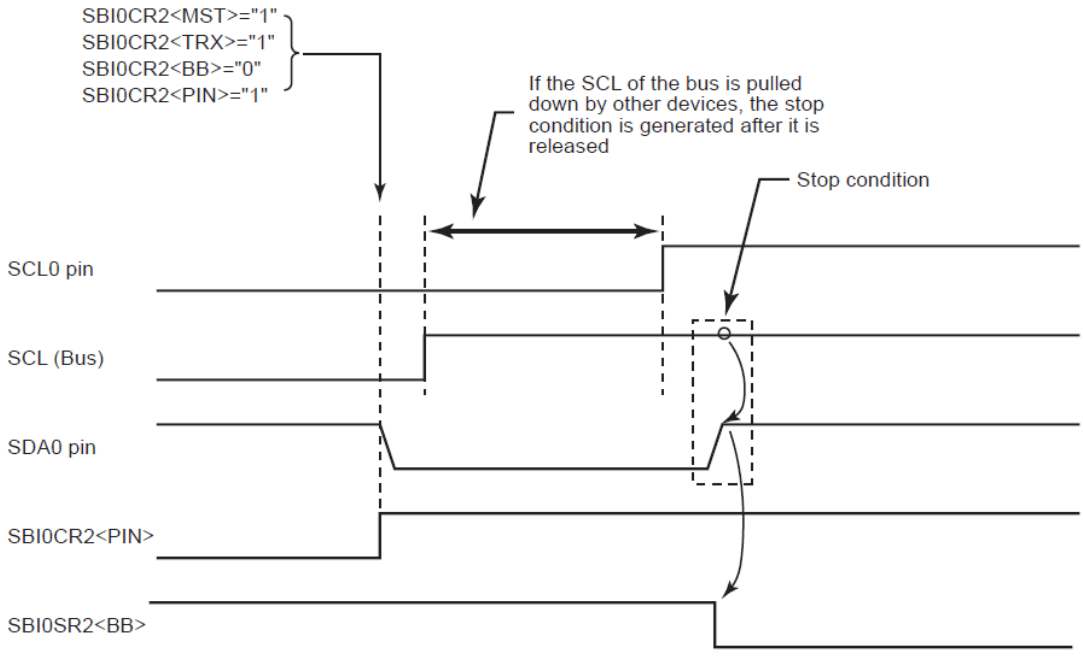



Figure 14.21 Stop Condition Generation

### 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.*

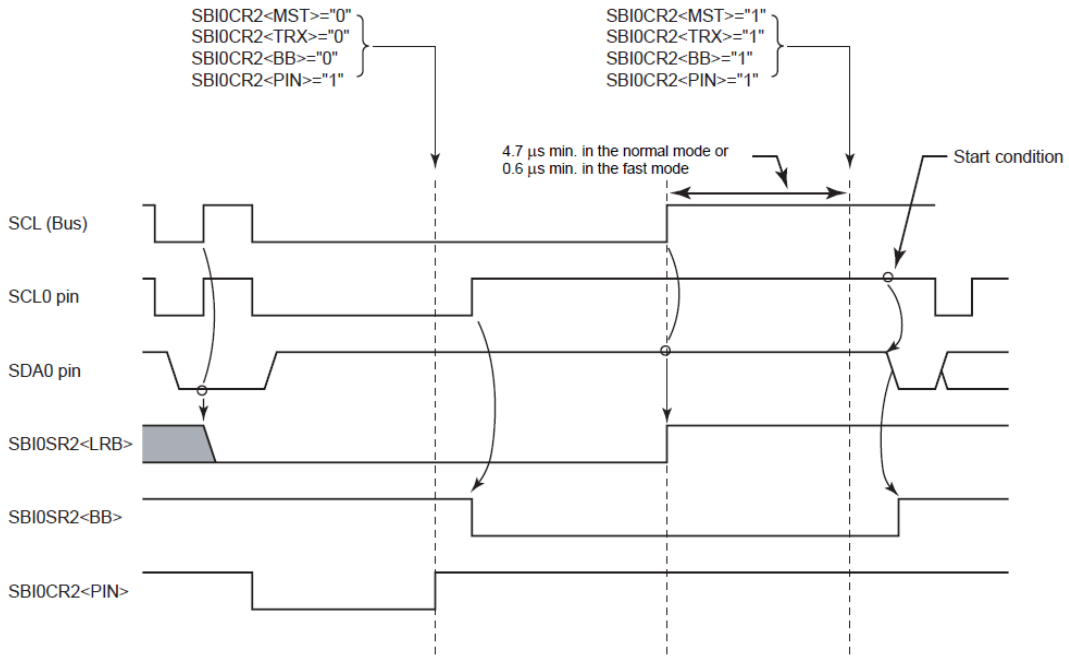



Figure 14.22 Timing diagram when restarting

## 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)

*Note: For m and n, refer to "14.4.4.1 clock source"*

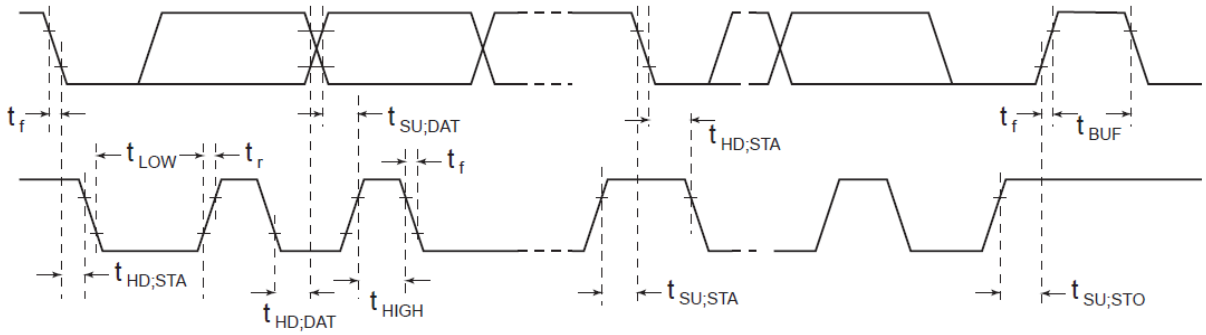



Figure14.23 Definition of Timing (No.1)

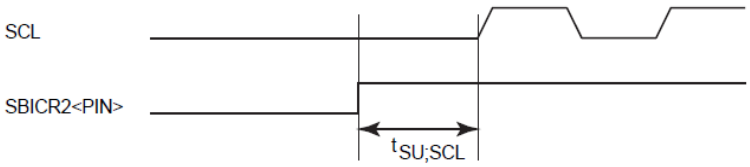



Figure 14.24 Definition of Timing (No.2)

# 15 Synchronous Serial Interface (SIO)

MQ6812/MQ66821 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

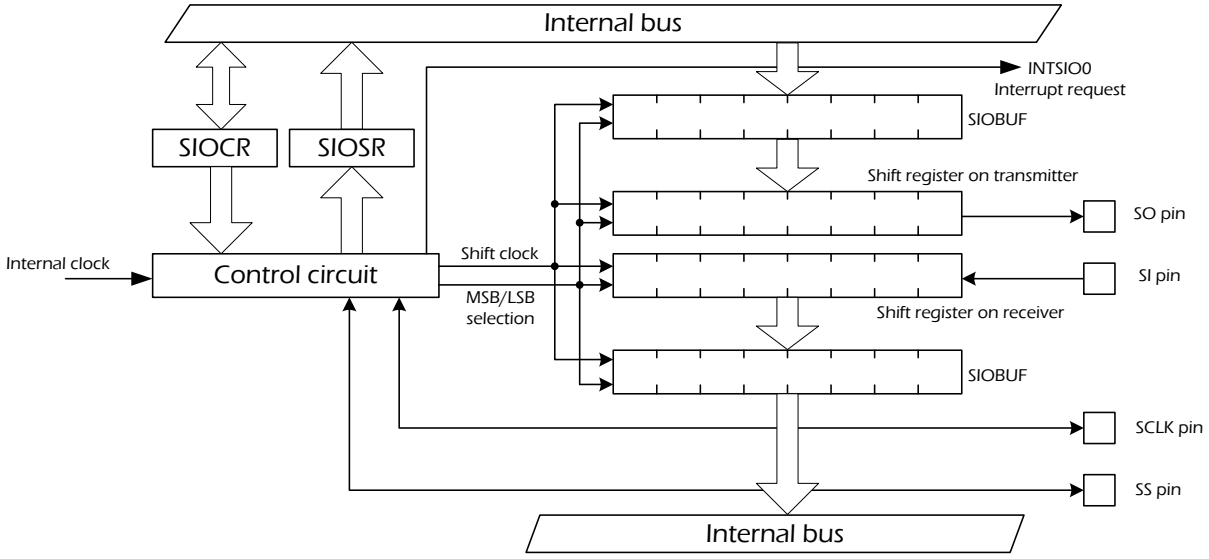



Figure15.1 Serial Interface

*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.*

## 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 | LCDEN | - | RTCEN | - | - | - | - | SIO0EN |
| Read/Write | R/W | R | R/W | R | R | R | R | R/W |
| After reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| LCDEN | LCD control | 0: Disable<br>1: Enable |
| RTCEN | RTC control | 0: Disable<br>1: Enable |
| SIO0EN | SIO0 control | 0: Disable<br>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.*

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 |

| | | | | |
|---|---|---|---|---|
| SIOEDG | Transfer edge selection | 0:Receive data at a rising edge and transmit data at a falling edge<br>1:Transmit data at a rising edge and receive data at a falling edge. | | |
| SIOCKS | Serial clock selection[Hz] | | NORMAL1/2 or IDLE 1/2 mode | SLOW1/2 or SLEEP 1 mode |
| | | 000 | fcgck/$2^9$ | - |
| | | 001 | fcgck/$2^6$ | - |
| | | 010 | fcgck/$2^5$ | - |
| | | 011 | fcgck/$2^4$ | - |
| | | 100 | fcgck/$2^3$ | - |
| | | 101 | fcgck/$2^2$ | - |
| | | 110 | fcgck/2 | fs/$2^3$ |
| | | 111 | Ext. clock input | |
| SIODIR | Transfer format (MSB/LSB) | 0: LSB first (transfer from bit 0)<br>1: MSB first (transfer from bit 7) | | |
| SIOS | Transfer operation start/stop instruction | 0:Operation stop (reserved stop)<br>1:Operation start | | |
| SIOM | Transfer mode selection and operation | 00:Operation stop (forced stop)<br>01: 8-bit transmit mode<br>10:8-bit receive mode<br>11:8-bit transmit and receive mode | | |

*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<br>1:Transfer in progress |
| SEF | Shift operation status monitor | 0:Shift operation not in progress<br>1:Shift operation in progress |
| OERR | Receive overrun error flag | 0:No overrun error has occurred<br>1:At least one overrun error has occurred |
| RENDB | Receive completion flag | 0: At least one data receive operation has been executed<br>1: No data has been received since the last received data was read out |
| UERR | Transmit underrun error flag | 0: No transmit underrun error has occurred<br>1:At least one transmit underrun error has occurred |
| TBFL | Transmit buffer full flag | 0:The transmit buffer is empty<br>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

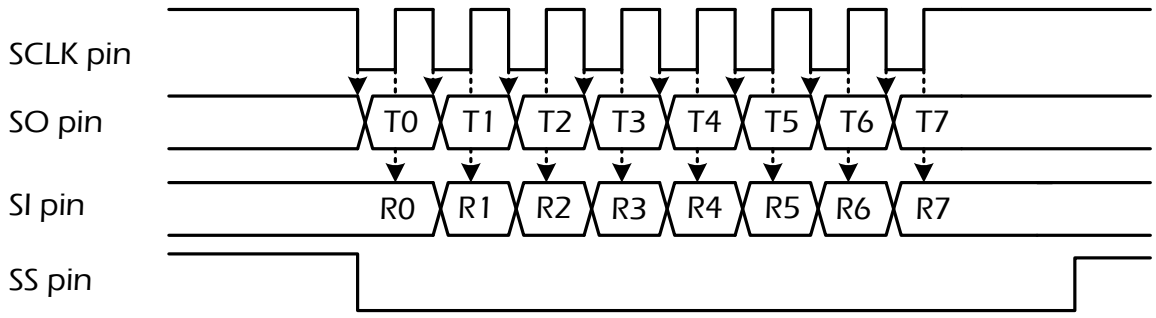
The serial data transfer edge can be selected by using SIOCR<SIOEDG>.

| SIO0CR<SIOEDG> | Data transmission | Data reception |
|---|---|---|
| 0 | Falling edge | Rising edge |
| 1 | Rising edge | Falling edge |

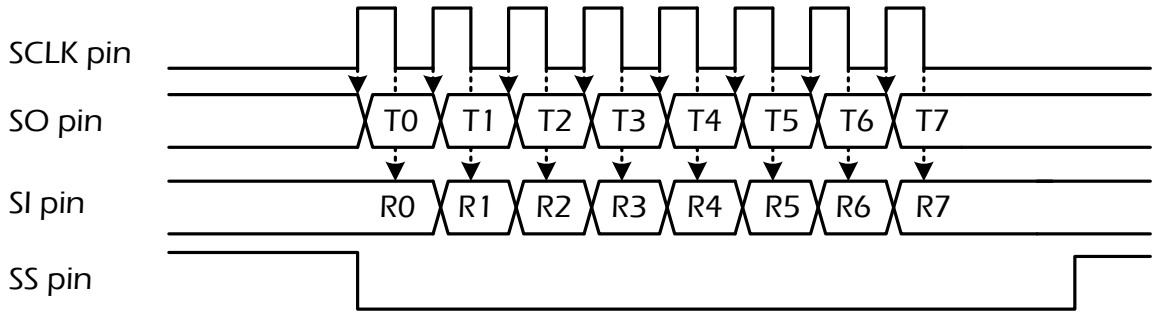
Table 15.4 Transfer Edge Selection

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 8$^{th}$ 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

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 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.

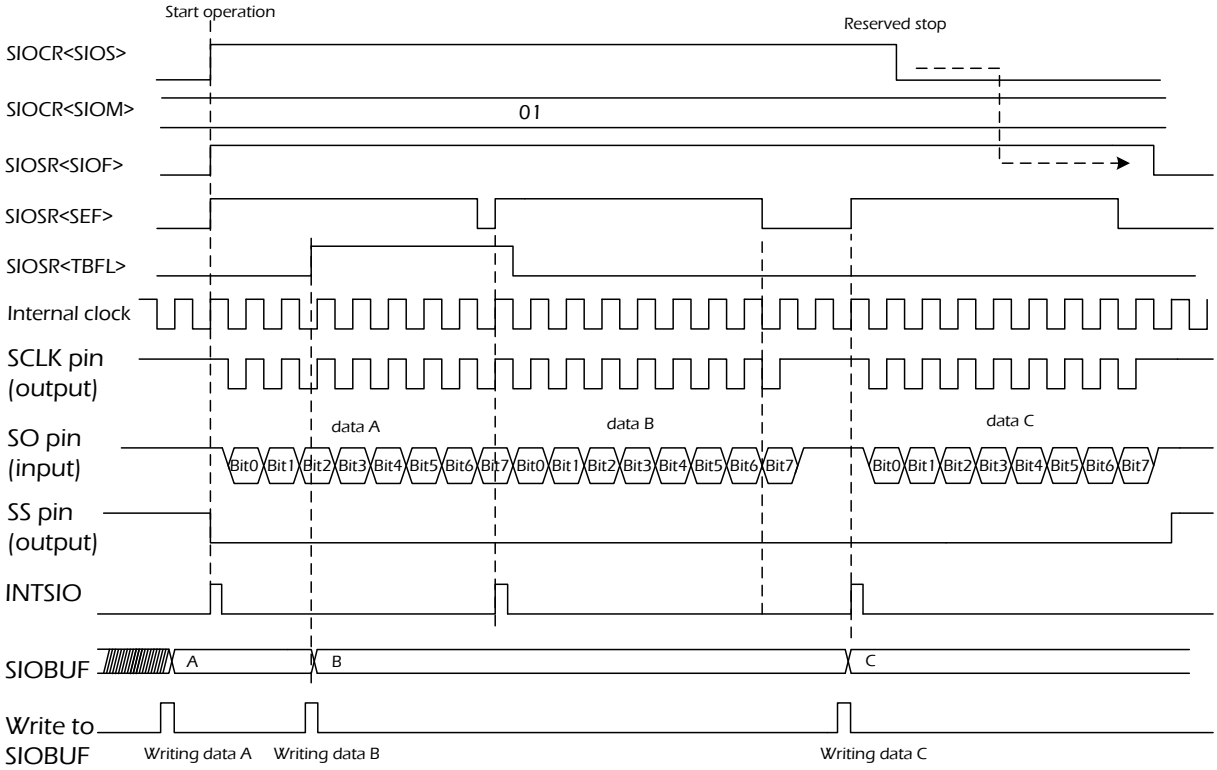
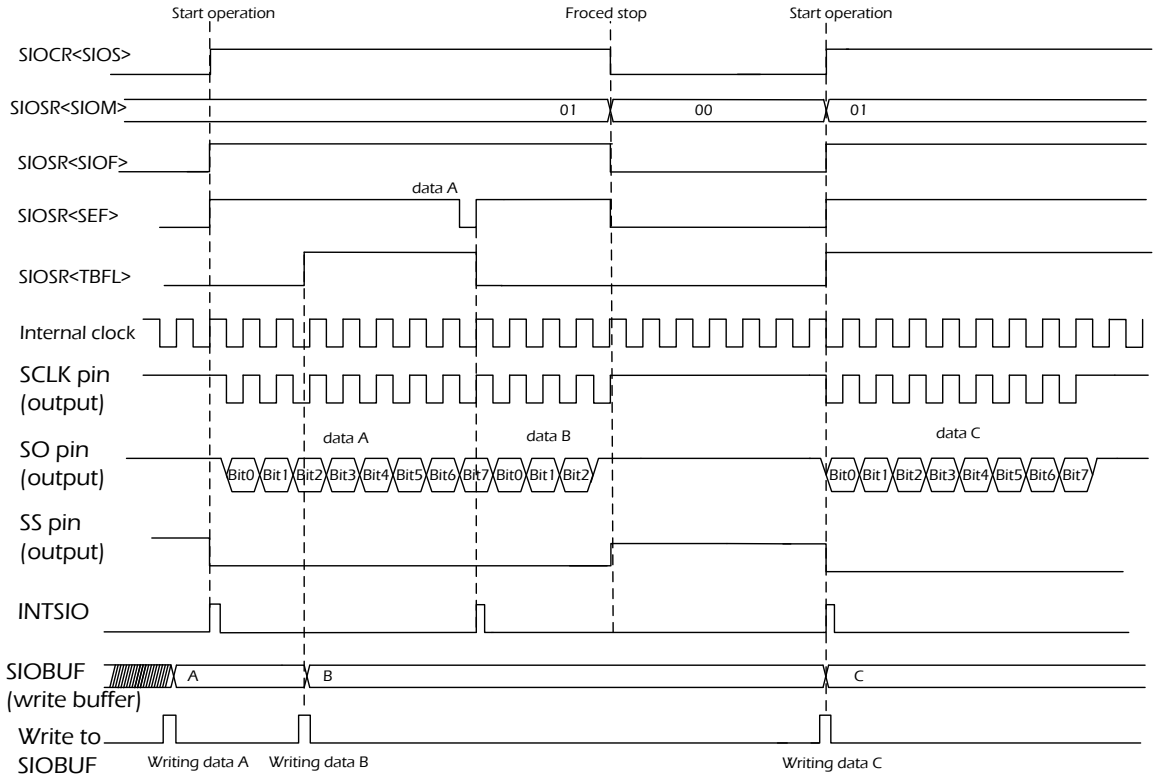



Figure 15.4 8-bit Transmit Mode (Internal Clock and Reserved Stop)




igure 15.5 8-bit Transmit mode(Internal clock and forced stop

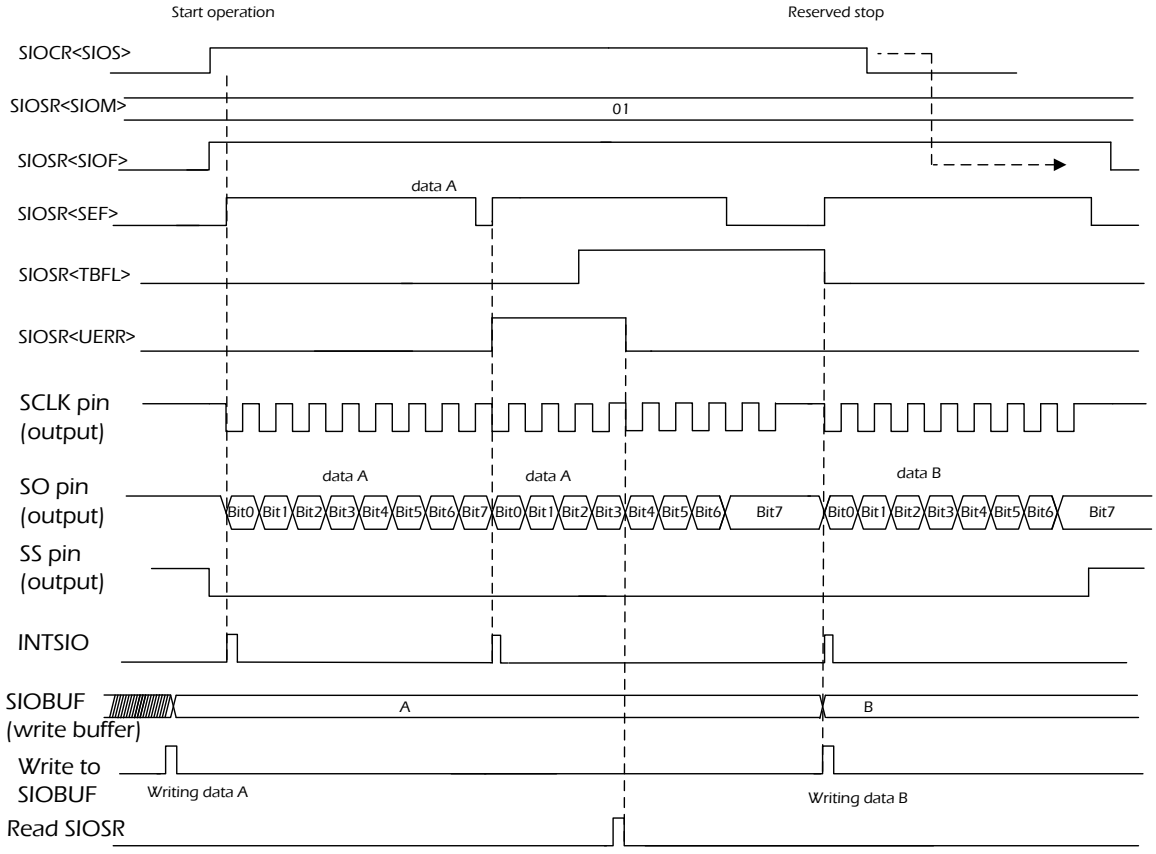



Figure 15.6 8-bit Transmit mode (External clock and occurrence of transmit underrun error)

### 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. 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.

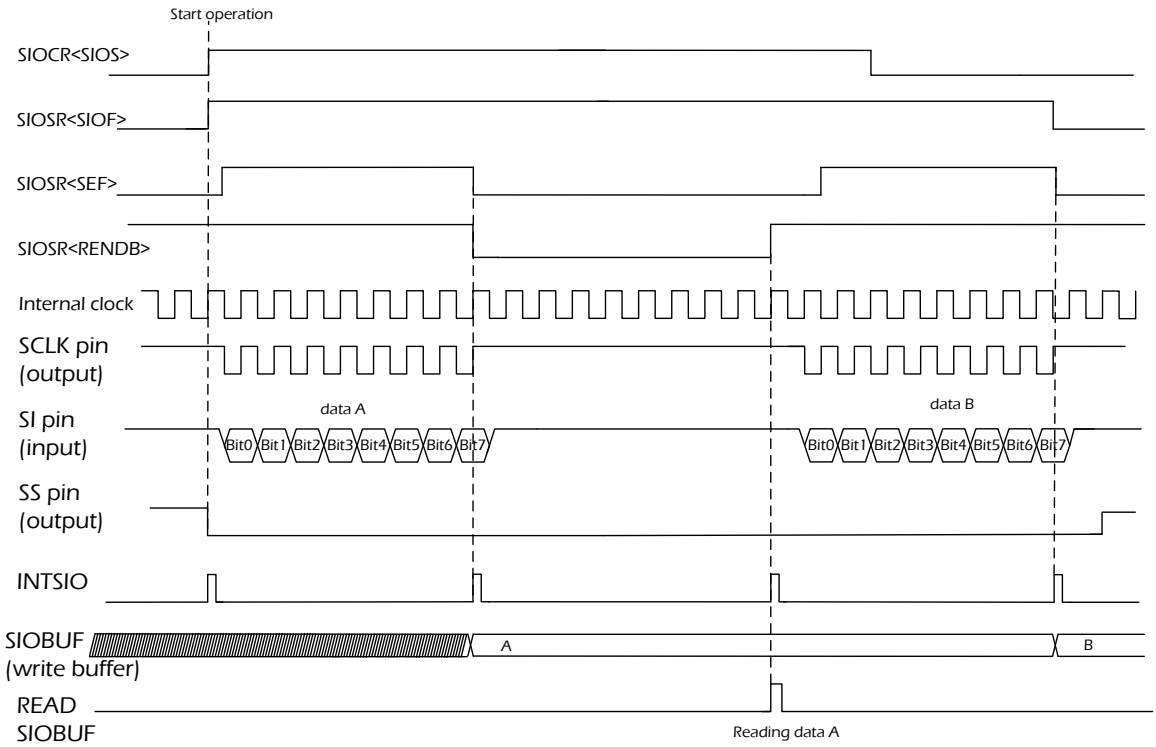



Figure 15.7 8-bit Receive Mode (Internal Clock and Reserved Stop)

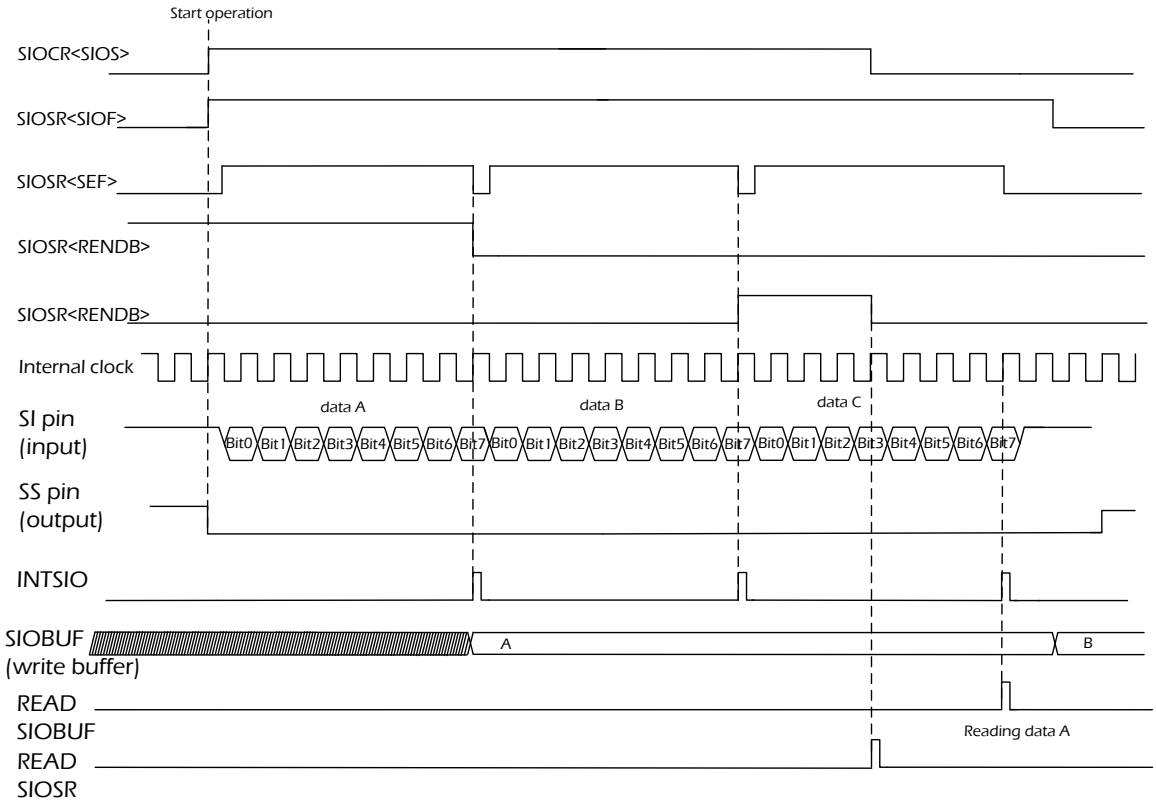



Figure 15.8 8-bit Receive Mode(External Clock and Occurrence of Overrun Error)

## 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.

#### 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.

(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.

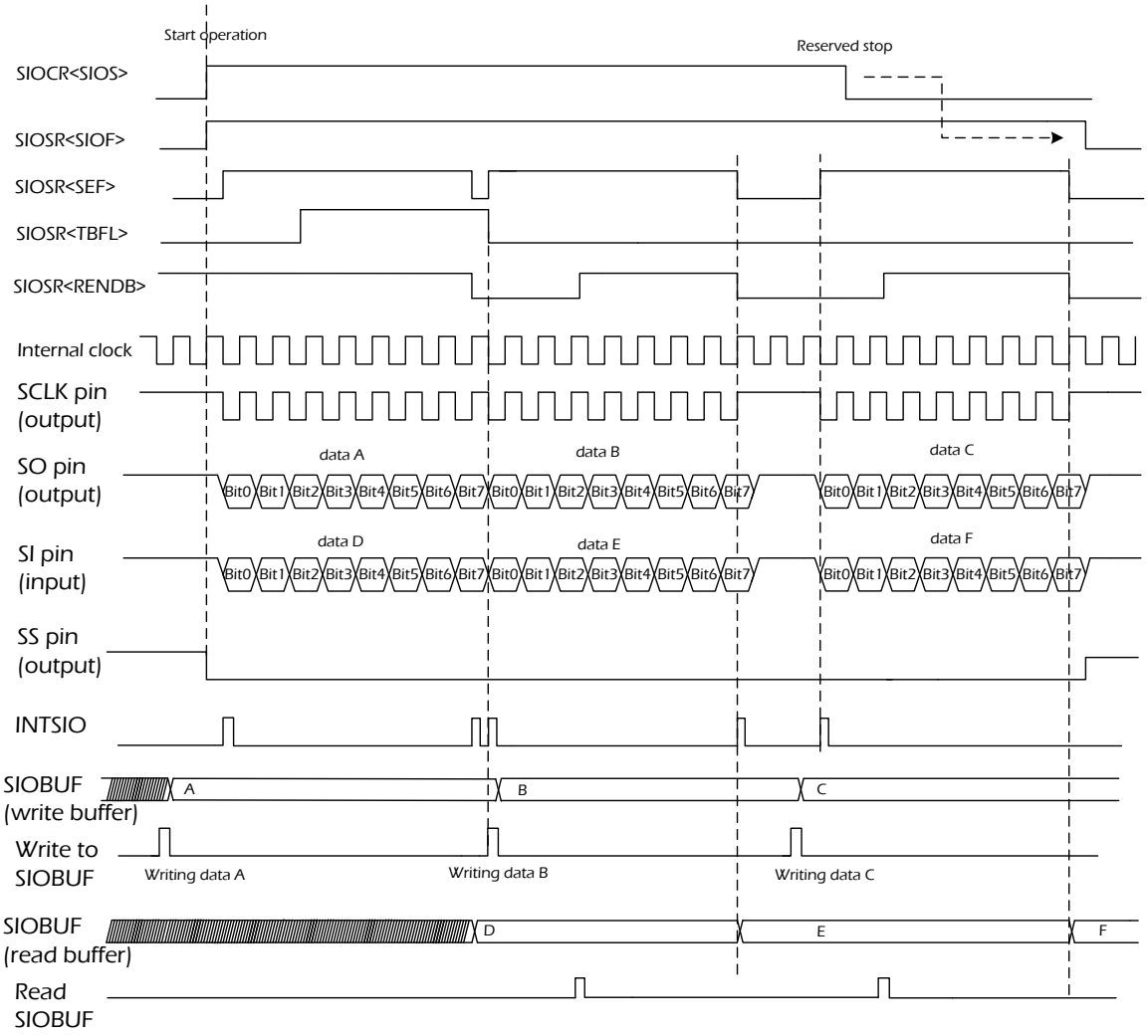



**Figure 15.9 8-bit Transmit/receive mode(External clock, occurrence of transmit underrun error and occurrence of overrun error)**

## 15.6 AC Characteristic

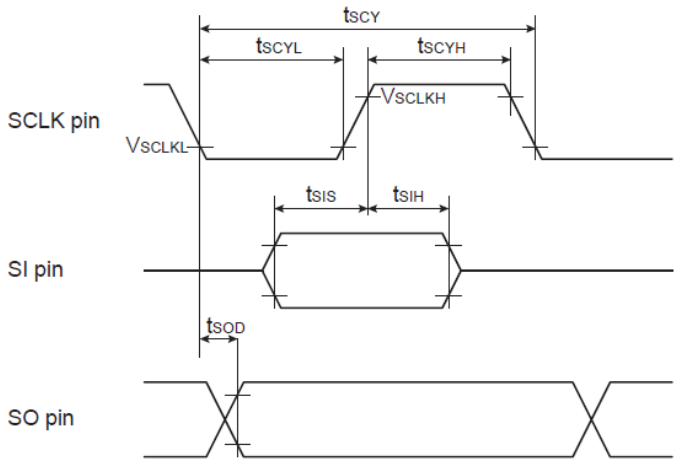



Figure 15.10 AC characteristics

($V_{SS}$ = 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<br>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<br>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 | - | $V_{DD}$ | |

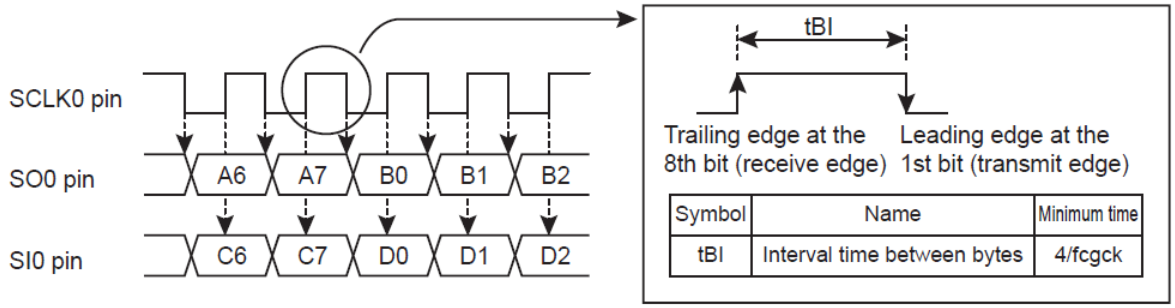



Figure 15.11 Interval time between bytes

# Appendix A. In-system Programming Function (ISP)

MQ6812/MQ6821 is equipped with ISP(In-System Programming). ISP function provide user to debug via MQ-Link and PC software operate . User can do software or system design adjustment by In-system programming function.

Folloing content will explain the control pin of ISP founction , and how to connect to target system.

## Control Pin

ISP function need two communication pins and three power and RESET pins. The pin description as table A.1; P40 and P41 are communication pins to ISP function.

| Pin description (OECD Mode) | Input/output | Function | Pin Description (MCU Mode) |
|---|---|---|---|
| ISPCLK | Input | Communication pin(clock ctrl) | P40/AIN0/KWI0 |
| ISPDIO | Input/Output | Communication pin (data ctrl) | P41/AIN1/KWI1 |
| RESETB | Input | Reset control pin | RESETB |
| VDD | Power | 5.0V | 2.0V to 5.5V |
| VSS | Power | 0V | |

Table A.1 Control pin of ISP function

## How to connect MQ-Link to target system

To use ISP function, user have to link specific pins of target system to MQ-Link. iMQ would provide interface control cable. Set interface contrl cabel to target system will make ISP function easier. User can refer to figure A.1. to link MQ-Link to target system.

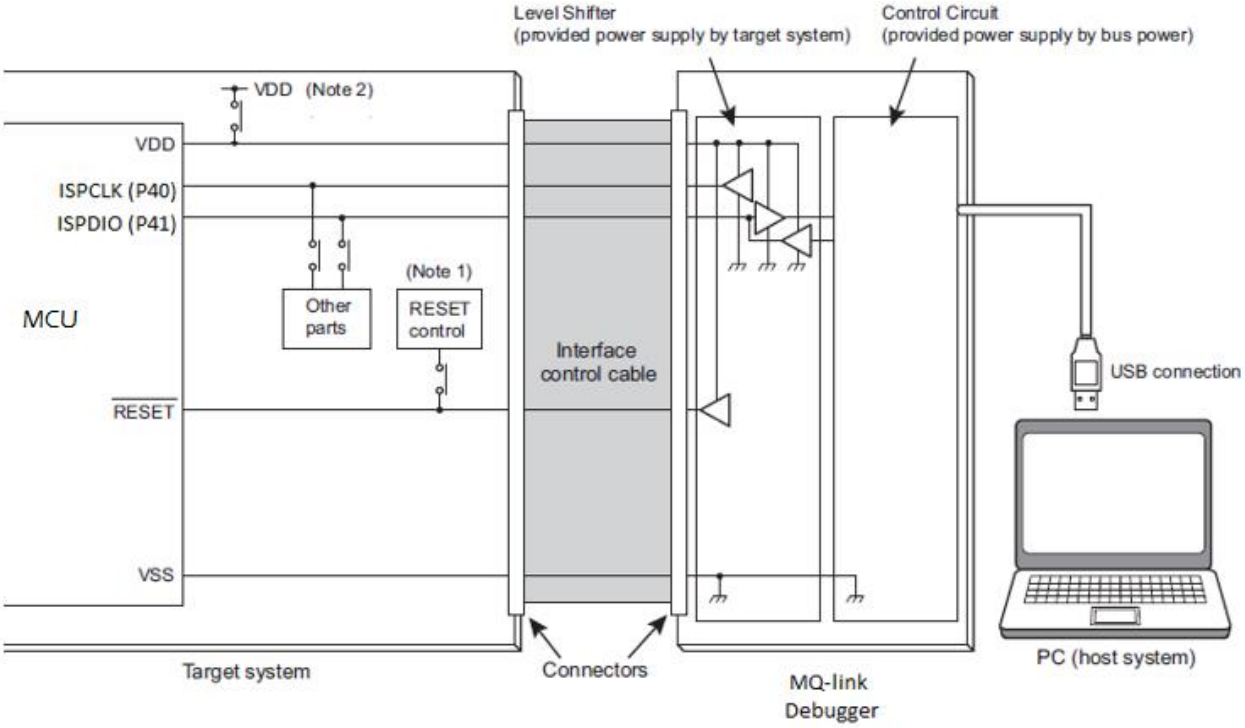



Figure A.1 Link MQ-Link to target system

*Note1：If the reset circuit of system board will effect ISP function, please use jumper or switch to disconnect.*

*Note2：When operate ISP function, the MQ-Link would supply VDD to MCU of target system. After ISP procedure finish, the target system would supply VDD to MCU.*

*Note3：Detail of MQ-Link , please see"iMQ i87 IDE user manual".*

# Appendix B. Product number information

Example： MQ 68 12 SP 028 H A E R

iMQ

Product Series

Sub Series

Package Type

| Code | Package type | Cod | 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

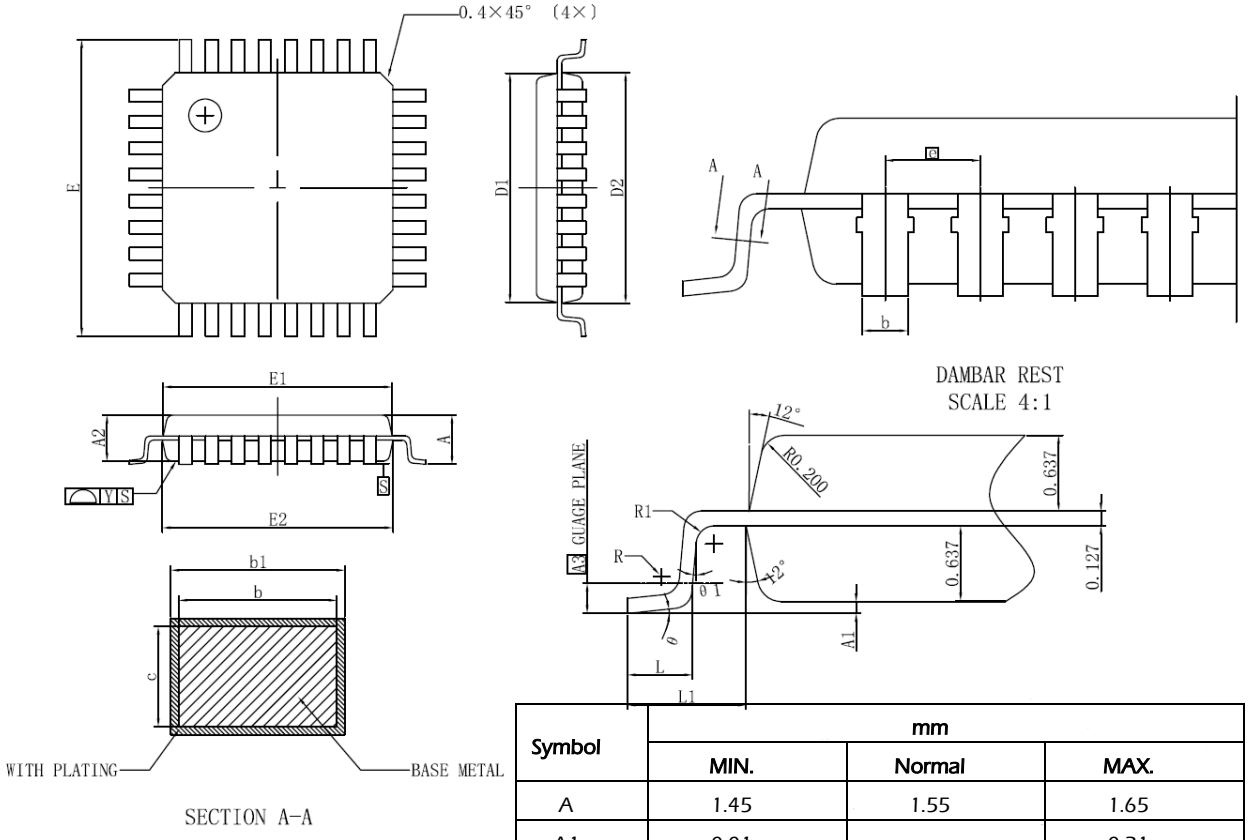
| Code | Program Flash/ Data flash/RAM Size | Code | Program Flash/ Data flash/RAM Size |
|---|---|---|---|
| A | 128 Bytes | K | 24K Bytes |
| B | 256 Bytes | M | 32K Bytes |
| E | 512 Bytes | N | 40K Bytes |
| J | 1K Bytes | P | 48K Bytes |
| L | 2K Bytes | S | 64K Bytes |
| T | 4K Bytes | U | 96K Bytes |
| G | 8K Bytes | W | 128K Bytes |
| C | 12K Bytes | V | 無 |
| H | 16K Bytes | | |

Operating Temp

| Code | Operating temp. |
|---|---|
| R | -40~85°C |
| X | -40~105°C |
| T | -40~125°C |

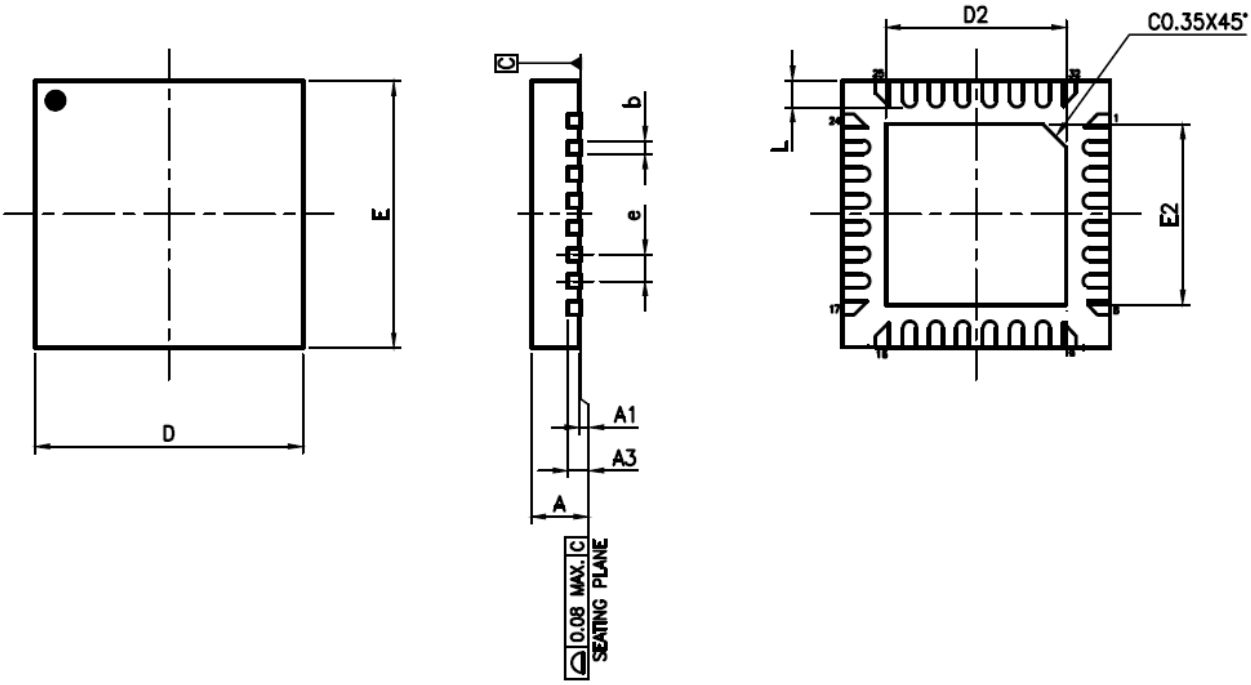
# Appendix C. Package Information

## LQFP 32 7x7 (Product No.: MQ6812LQ032HAER)



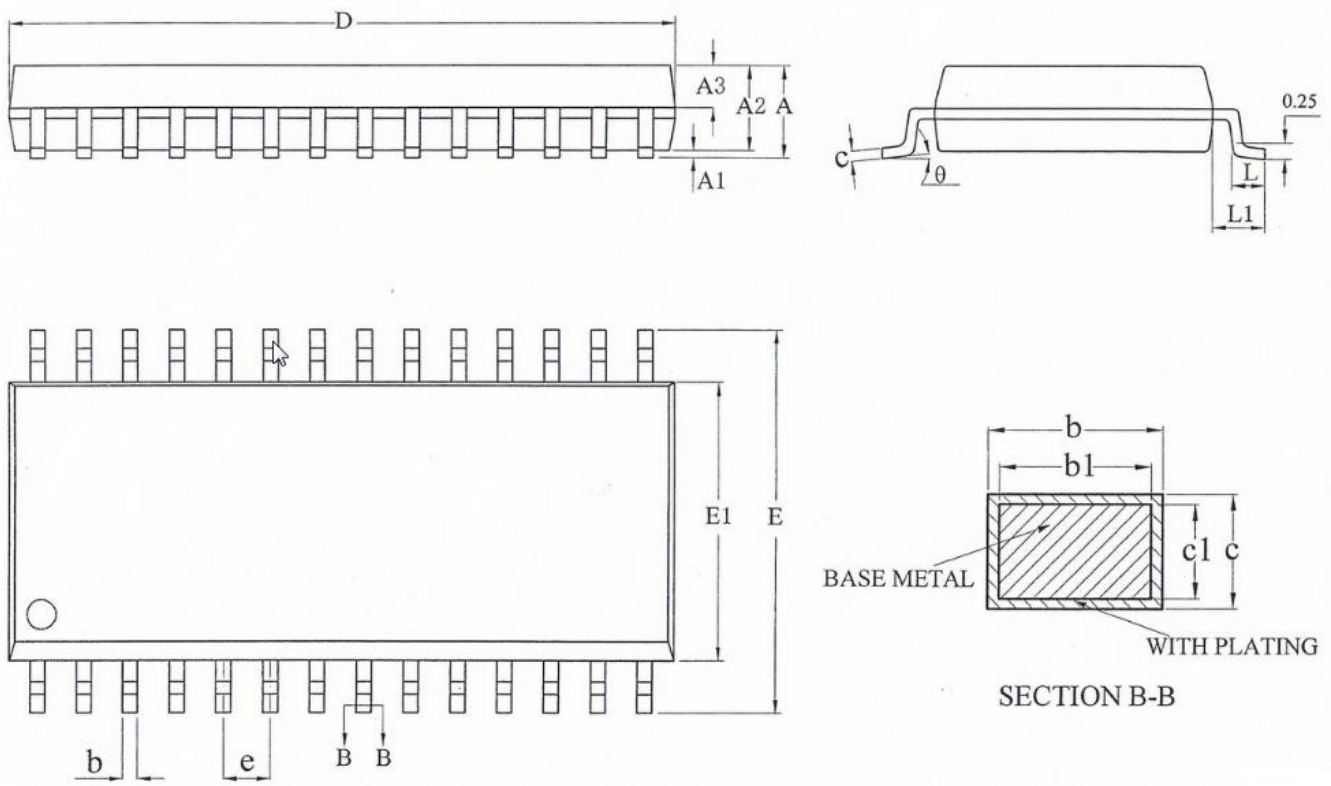

| Symbol | mm | | |
|---|---|---|---|
| | MIN. | Normal | MAX. |
| A | 1.45 | 1.55 | 1.65 |
| A1 | 0.01 | – | 0.21 |
| A2 | 1.3 | 1.4 | 1.5 |
| A3 | – | 0.254 | |
| b | 0.30 | 0.35 | 0.40 |
| b1 | 0.31 | 0.37 | 0.43 |
| c | – | 0.127 | – |
| D1 | 6.85 | 6.95 | 7.05 |
| D2 | 6.9 | 7.00 | 7.10 |
| E | 8.8 | 9.00 | 9.20 |
| E1 | 6.85 | 6.95 | 7.05 |
| E2 | 6.9 | 7.00 | 7.10 |
| e | – | 0.8 | – |
| L | 0.43 | – | 0.71 |
| L1 | 0.90 | 1.0 | 1.10 |
| R | 0.1 | – | 0.25 |
| R1 | 0.1 | – | – |
| θ | 0 | – | 10° |
| θ1 | 0 | – | – |
| y | – | – | 0.1 |
| Z | – | 0.70 | – |

# QFN 32 4x4 (Product No.: MQ6812N4032HAER)



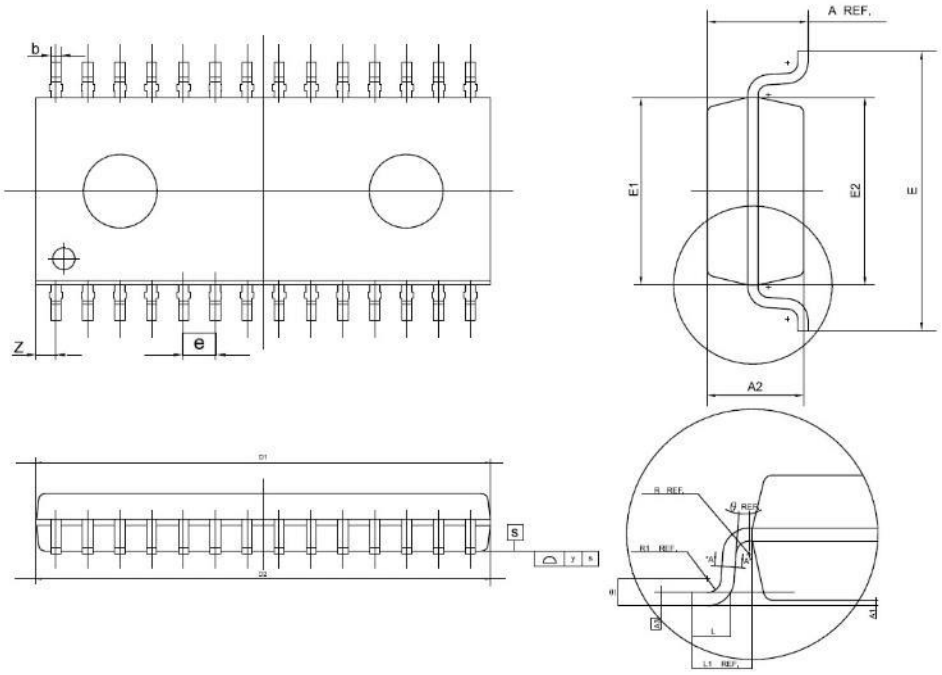

| Symbol | mm | | |
|---|---|---|---|
| | MIN. | NORMAL | MIN. |
| A | 0.70 | 0.75 | 0.80 |
| A1 | 0.00 | 0.02 | 0.05 |
| A3 | 0.203 REF | | |
| b | 0.15 | 0.20 | 0.25 |
| D | 4.00 BSC | | |
| E | 4.00 BSC | | |
| e | 0.40 BSC | | |
| D2 | 2.60 | 2.75 | 2.90 |
| E2 | 2.60 | 2.75 | 2.90 |
| L | 0.25 | 0.35 | 0.45 |

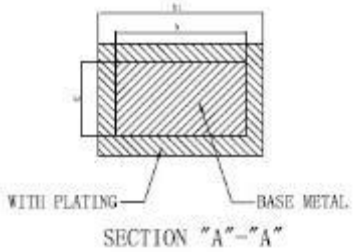
## SSOP 28 (Product No.: MQ6812SS028HAER)




| Symbol | mm | | |
|---|---|---|---|
| | MIN. | NORMAL | MIN. |
| A | – | – | 2.00 |
| A1 | 0.05 | – | 0.25 |
| A2 | 1.65 | 1.75 | 1.85 |
| A3 | 0.75 | 0.80 | 0.85 |
| b | 0.29 | – | 0.37 |
| b1 | 0.28 | 0.30 | 0.33 |
| c | 0.15 | – | 0.20 |
| c1 | 0.14 | 0.15 | 0.16 |
| D | 10.00 | 10.20 | 10.40 |
| E | 7.60 | 7.80 | 8.00 |
| E1 | 5.10 | 5.30 | 5.50 |
| e | 0.65BSC | | |
| L | 0.55 | 0.75 | 0.95 |
| L1 | 1.25BSC | | |
| θ | 0 | – | 10° |

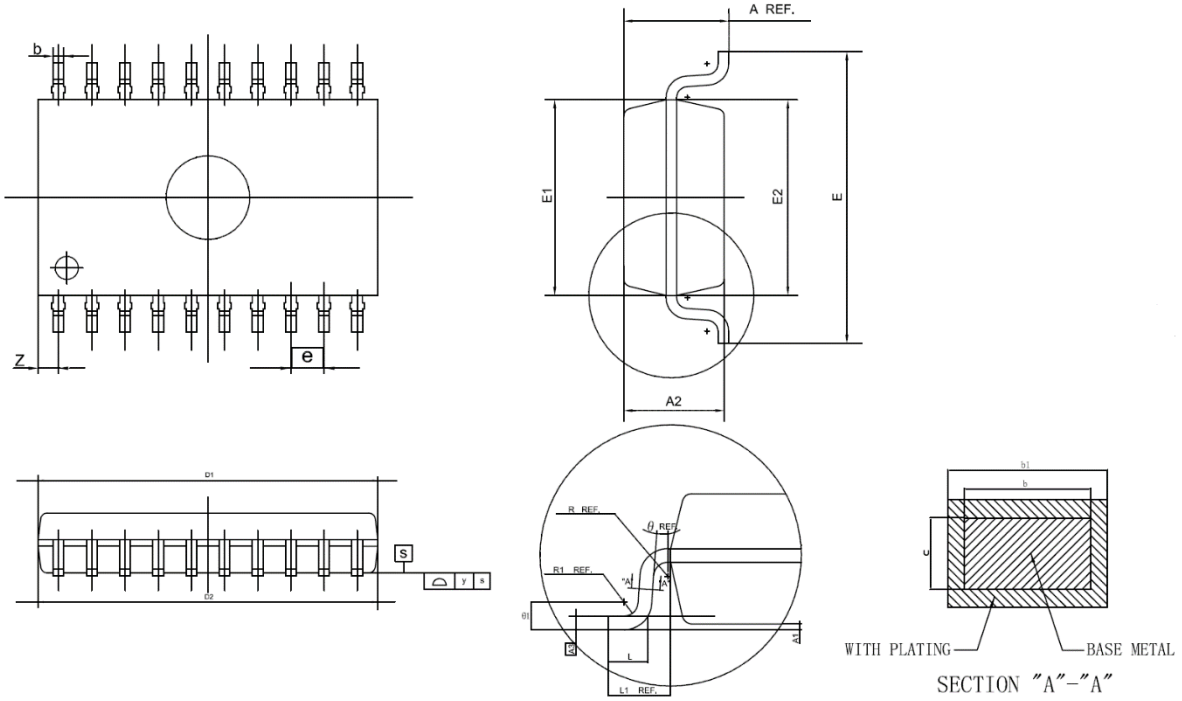
## SOP 28 (Product No.: MQ6812SP028HAER)



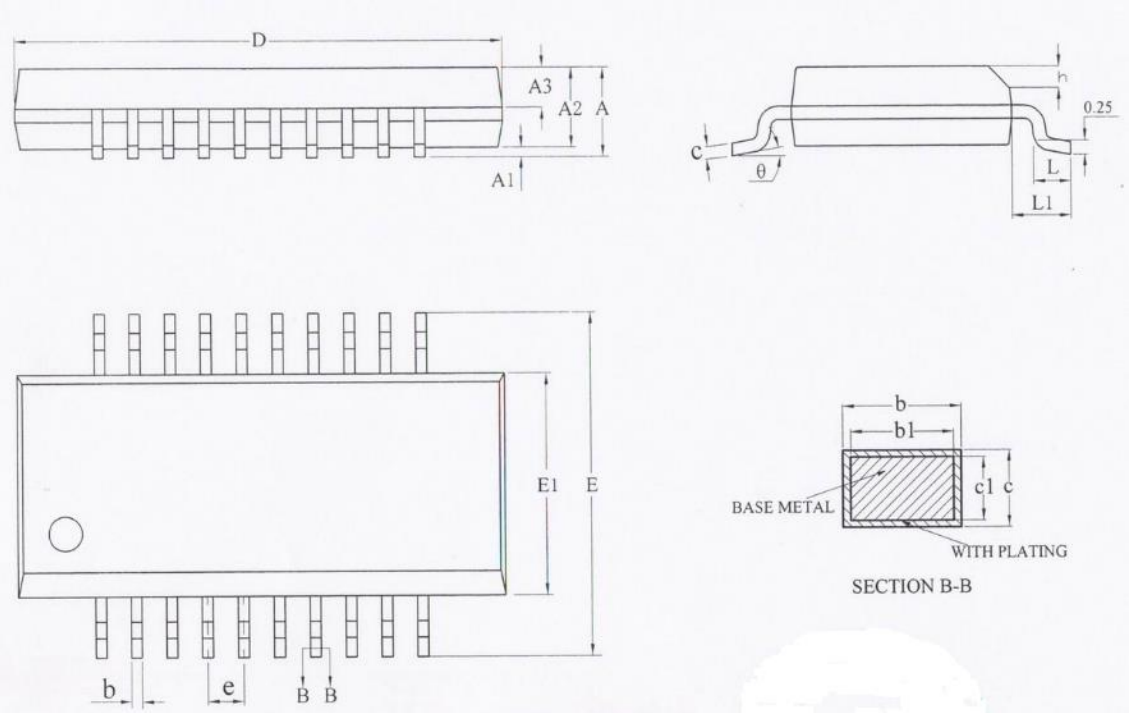

| Symbol | mm | | |
|---|---|---|---|
| | MIN. | NORMAL | MIN. |
| A | 2.465 | 2.515 | 2.565 |
| A1 | 0.10 | 0.15 | 0.20 |
| A2 | 2.1 | 2.3 | 2.5 |
| A3 | – | 0.274 | – |
| B | 0.356 | 0.406 | 0.456 |
| b1 | 0.366 | 0.426 | 0.486 |
| C | – | 0.254 | – |
| D1 | 17.750 | 17.950 | 18.150 |
| D2 | 17.8 | 18.0 | 18.2 |
| E | 10.1 | 10.3 | 10.500 |
| E1 | 7.374 | 7.450 | 7.574 |
| E2 | 7.424 | 7.500 | 7.624 |
| e | – | 1.270 | – |
| L | 0.764 | 0.864 | 0.964 |
| L1 | 1.303 | 1.403 | 1.503 |
| R | – | 0.2 | – |
| R1 | – | 0.3 | – |
| θ | 0 | – | – |
| θ1 | 0 | – | 10° |
| y | – | – | 0.1 |
| Z | – | 0.745 | – |

## SOP 20 (Product No.: MQ6821SP020HAER )



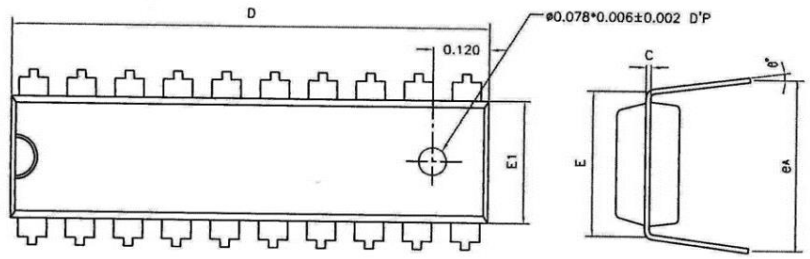

| Symbol | mm | | |
|---|---|---|---|
| | MIN. | NORMAL | MIN. |
| A | 2.465 | 2.515 | 2.565 |
| A1 | 0.100 | 0.150 | 0.200 |
| A2 | 2.100 | 2.300 | 2.500 |
| A3 | – | 0.274 | – |
| b | 0.356 | 0.406 | 0.456 |
| b1 | 0.366 | 0.426 | 0.486 |
| c | – | 0.254 | – |
| D1 | 12.500 | 12.700 | 12.900 |
| D2 | 12.550 | 12.750 | 12.950 |
| E | 10.206 | 10.306 | 10.406 |
| E1 | 7.400 | 7.450 | 7.500 |
| E2 | 7.450 | 7.500 | 7.550 |
| e | – | 1.270 | – |
| L | 0.800 | 0.864 | 0.900 |
| L1 | 1.303 | 1.403 | 1.503 |
| R | – | 0.200 | – |
| R1 | – | 0.300 | – |
| θ | 0 | – | – |
| θ1 | 0 | – | 10° |

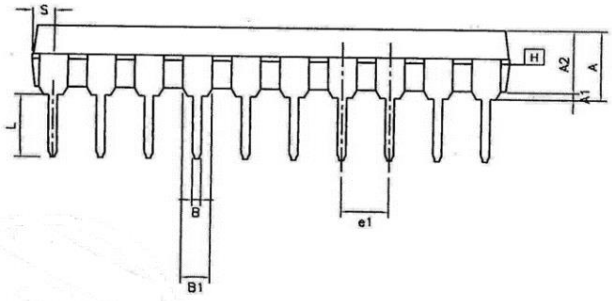
## SSOP 20 (Product No.: MQ6821SS020HAER )




| Symbol | mm | | |
|---|---|---|---|
| | MIN. | Normal | MIN. |
| A | – | – | 1.75 |
| A1 | 0.10 | – | 0.25 |
| A2 | 1.30 | 1.40 | 1.50 |
| A3 | 0.60 | 0.65 | 0.70 |
| b | 0.23 | – | 0.33 |
| ‘b1 | 0.22 | 0.25 | 0.28 |
| c | 0.21 | – | 0.26 |
| ‘c1 | 0.19 | 0.20 | 0.21 |
| D | 8.45 | 8.65 | 8.85 |
| E | 5.80 | 6.00 | 6.20 |
| E1 | 3.70 | 3.90 | 4.10 |
| e | 0.635BSC | | |
| h | 0.25 | – | 0.50 |
| L | 0.50 | – | 0.80 |
| L1 | 1.05BSC | | |
| θ | 0 | – | 8° |
| L/F size (mil) | 96*140 | | |

## DIP 20 (Product No.: MQ6821DP020HAER)



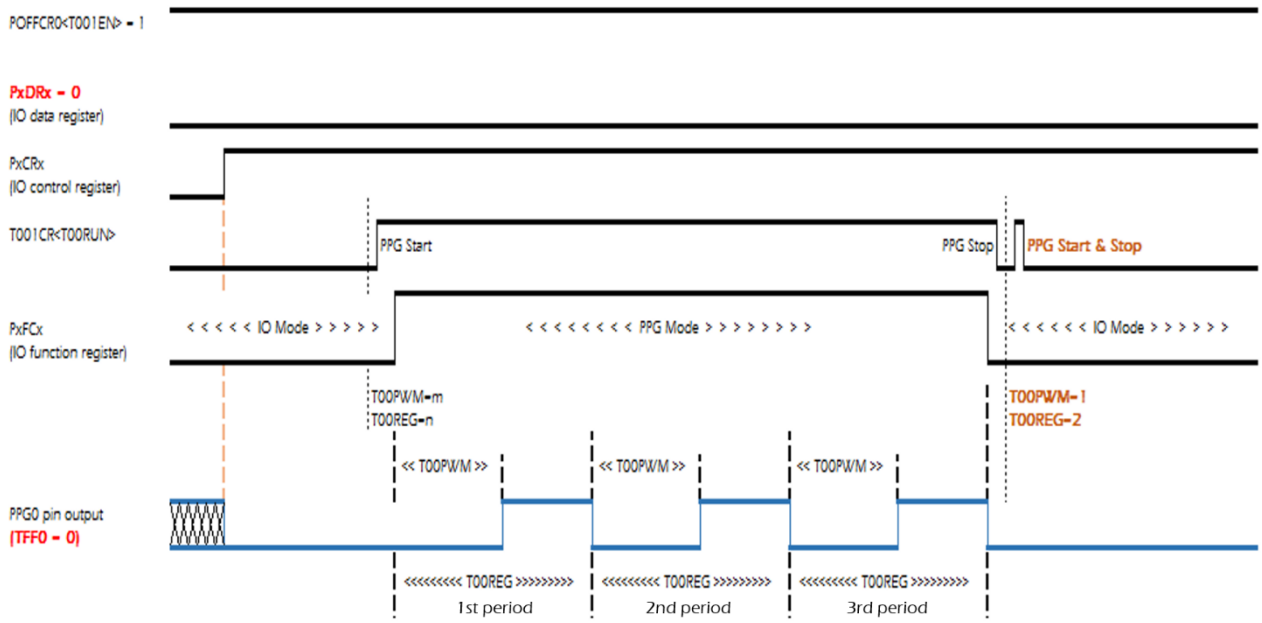




| Symbol | mm | | |
|---|---|---|---|
| | MIN. | Normal | MIN. |
| A | – | – | 4.445 |
| A1 | 0.381 | – | – |
| A2 | 3.175 | 3.302 | 3.429 |
| B | 0.406 | 0.457 | 0.508 |
| B1 | 1.473 | 1.524 | 1.626 |
| C | 0.203 | 0.254 | 0.279 |
| D | 25.705 | 26.060 | 26.416 |
| E | 7.366 | 7.620 | 7.874 |
| E1 | 6.223 | 6.350 | 6.477 |
| e1 | 2.286 | 2.540 | 2.794 |
| L | 3.048 | 3.302 | 3.556 |
| θ° | 0 | - | 15 |
| eA | 8.509 | 9.017 | 9.525 |
| S | – | – | 1.905 |

# Appendix D. Application Notice

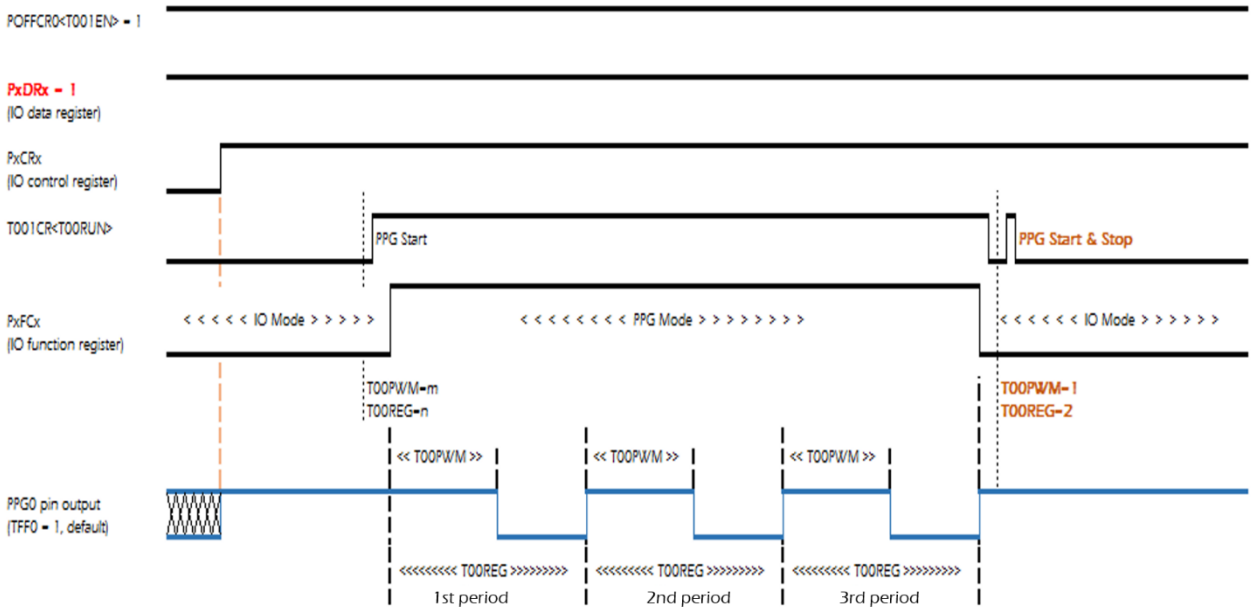
## (A) 10-bit PPG

PGG output level is effective from 2nd period, the level cannot be set by register. When PPG finished, set the pin to I/O function ,then to eliminate the un-known level by I/O output.




10-bit PPG TFFx=0 , I/O output to eliminate the un-known level

Figure D.1 PPG00B output (I/O output is low, PxDRx=0)




10-bit PPG TFFx=0 , I/O output to eliminate the un-known level

Figure D.2 PPG00B output (/O output is high, PxDRx=1)

## (B)10-bit timer/counter (TCQ0)

1. Only TCQ00,TCQ02,TCQ04 can use double buffer (detail refer to figure 10-11 and 10-15)
2. PPG pulse output do not support duty as 0% and 100%.

## (C) Pin related

1. Transfer P10 form RESTEB to I/O, or I/O to RESETB, when P10 is stable high leve. If transfer P10 when the status is low-level, it may cause reset.
2. The oscillator of PCBA should closed to MCU as possible.
3. When P2OUTCR4 or P2OUTCR3 is" 1" (P23,P24 set to open-drain pin), the IDD woud increase dozens uA in STOP mode.
4. The I/O pins which need to be pull-up or pull-low should be connected to VDD or GND through a resistor(resistance > 100 ohm). To connect I/O pins to VDD or GND directly should be avoided .

## (D) Operation mode and clock switch

1. In the time period from stop mode enabled to system wake up，the interrupts latches may set to "1" by external interrupt signal changed, and the system interrupt immediately when exit stop mode. Before stop mode enable, it is suggested to stop all interrupts. If it would enable any interrupts after exiting stop mode, it need to clear non-used interrupts latches.
2. Set SYSCR2<XEN> to"1" when P0FC0 is "0", the system clock would reset.
3. Transfer the clock, set the new clock first, then closed original clock. 4. Bit6~Bit3 is related to chip modifying when record the chip by writer. Please user use "bit operation" for programming. "SET (FSCTRL).0」" or "CLR (FSCTRL).0" for example. Do not use other bit to change the value of Bit 6~Bit3.5. In SLOW mode, do not change CGCR<FCGCKSEL>。
6. Set SYSCR2<XEN> to"1"，system clock would be reset when P0FC0 set to "0". If P00 or P01 is not used as port, set P0FC0 to "1".

## (E) ADC

1. In single mode, do not read ADCDRL or ADCDRH. If ADC finished between read ADCDRL and ADCDRH, then INTADC interrupt would be cancel, and lost the result of convert.
2. Read ADCDRL first then read ADCDRH. If read ADCDRH first then read ADCDRL or read ADCDRL only, it would not clear ADCCR2<EOCF>. The ADC interrupt would not occur.
3. MQ6812/ MQ6821 contains auto-calibration function. Detail please refer the application note on the website.

http://www.imqtech.com/zh-hant/tech/technicaldocuments/application_notes

## (F) Timer

1.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. Because the WDT/WDT2 of MQ6812/MQ6821 are fixed and cannot be stopped , please clear WDT/WDT2 per 1.5 second at lease.

## (G) LVD

1. The comparotor of LVD do not include Hysteretic architecture. When VDD close to LDxLVL, it may cause INTLVD requests occurred frequently. When VDD decrease to detected voltage or VDD increase to detected voltage, it would generate INTLVD request in both conditions.

## (H) Interrupt

1. Under interrupt service process, IMF would be "0" automatically. If use multi-interupts, control IL before setting IMF to "1".
2. If use multi-interupts, setting IMF to "1" after setting every EF.

## (I) I2C

1. Before enable I2C function ,set related register as below：
   -Low power consumption register1 (POFFCR1)
   Set POOFFCR1<SBI0EN> to"1" , before enable SBI/I2C function, unless the SBI/I2C setting is invalid.
   -Port P2 Output Control Registe (P2OUTCR)
   Set P2OUTCR3 and P2OUTCR4 to"1" , before enable SBI/I2C function
   - Port P2 Output Latch Register (P2DR)
   Set P2DR to"0" before enable SBI/I2C function, unless the SBI/I2C function would operate unexpectedly.

## (J) Instruction

Instruction "JP gg(note 1)" in extreme condition may cause next instruction code cannot be read correctly. Suggested to modified the code as below:

Note 1: gg is 16-bit general buffer WA, BC, DE, HL, IX, IY....

Suggest to add a "NOP instruction" between SWITCH and the 1st case.

**Original**

```
switch(variable){
 case 0:
    :
 break;
 case 1:
    :
 break;
    :
 default:
 break;
}
```

**modified**

```
switch(variable){
 __asm("NOP");//add this instruction
 case 0:
    :
 break;
 case 1:
    :
 break;
    :
 default:
 break;
}
```

The function these two program are the same. The modified code compilered by MQ-Link would appear "I87C1-Warning-538" (as below), this message can be ignored.