SQ7613
Datasheet V1.1

Content

1. Change History

Version	Approved Date	Description
V 1.1	2020/04/06	1.Update "9. Flash Memory Controller", remove 2-bytes description. 2.Update " 16.4.2 Serial Clock": slave mode maximum frequency is 4MHz, and table 16-3 . 3." 16.6 AC Characteristic", add the note for tsysclk.

2.Product Overview

2.1 Features

- **Basic Information**
  - Operating voltage: 2.0V ~ 5.5V
  - Operating temperature: -40°C ~ 85°C
  - Max system frequency 24 MHz
  - Instruction set is compatible with Toshiba TLCS-870/C1
- **Memory Configuration**
  - 64 KB Flash
  - 4 KB RAM
- **Operation modes**
  - Normal mode:150 uA/MHz @ 3.3V
  - Deep sleep mode: 1 uA @ 3.3V,CPU and RAM are retained
- **Clock Source**
  - 16 MHz external crystal(high frequency)
  - PLL
  - Internal crystal:
    - 32 kHz
    - 16 MHz
- **I/O**
  - Max 29 I/Os
  - 3 sets UART，2 sets I2C and 2 sets SIO
- **Multiplier and Dividor**
  - 16 x16-bit ,multiplication,and 32-bit addition
  - 32-bit divid 32-bit
- **Timer/Counter**
  - 8 16-bit timer/counter，Timer,External trigger, Event counter, Window, Pulse width measurement, PPG output modes
  - Watch-dog timer
- **External interrupt**
  - 8 external interrupt input (EINT0~EINT7)
- **12-bit ADC**
  - 7 CH ADC input
  - ADC VREF
- **Low voltage detection (LVD), total 8 levels.**
  - 2.0V/2.35V/2.65V/2.85V/3.15V/3.98V/4.2V/4.5V
- **Brown-out reset (BROR), total 4 levels**
  - 1.9V/2.25V/2.55V/2.75V
- **Code Protection**
- **Package type**
  - LQFP 7x7 32 pins
  - QFN 5x5 32 pins

2.2 Preface

SQ7613 is 870E core which is an energy efficient and low gate count core that implements the Toshiba TLCS-870/C1 CISC instruction set architecture. The variable length instruction set offers 38 core instructions, nine addressing modes and powerful memory manipulation operations. The smallest instructions have one-byte opcode and largest instruction five bytes. Instructions that are frequently used have two- to four-byte opcodes.

870E core is a three-stage execution pipeline design. The instruction queue and the core functional units are capable of executing frequently used instructions in a single cycle. The Harvard memory architecture allows simultaneous instruction fetch and data access. Dedicated hardware is designed to handle instruction and data alignment, eliminating software alignment overhead.

SQ7613 has 64K Bytes flash mamery, 4K Bytes RAM, various I/Os funcitons, timers/counters, and 12-bit ADC. There are variety of internal clock and external clock sources; different digital peripheral, and accurate analog features. User can optimize by different requirements.

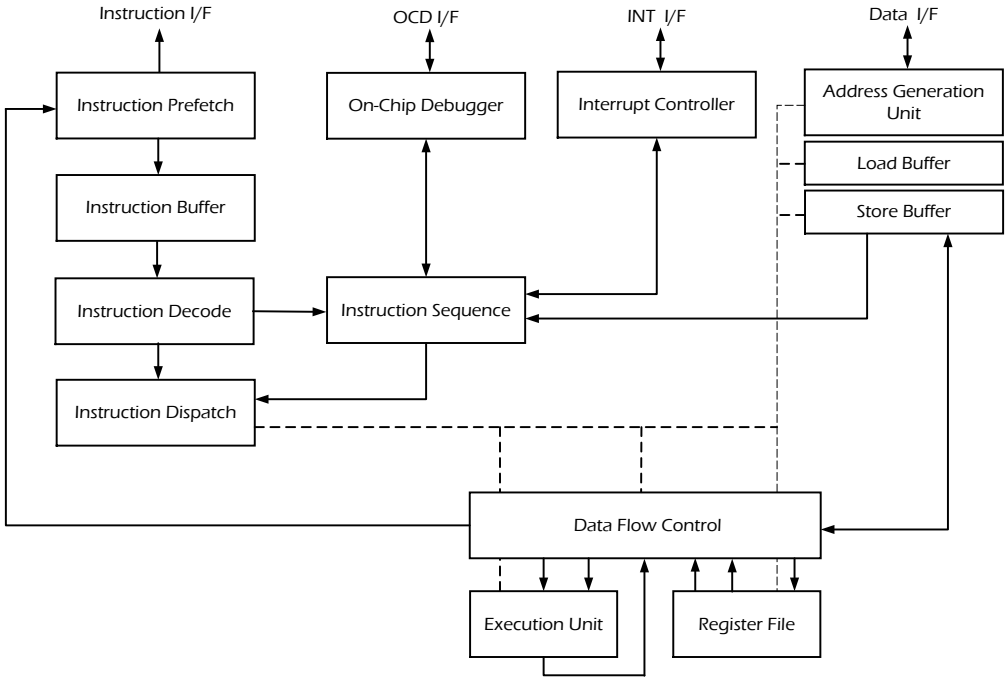


FIGURE 2-1 MQ870E Block Diagram

Product no.	SQ7613LQ032SETR	SQ7613N5032SETR
Pin count. (I/Os)	32 (29)	32 (29)
Operating voltage	2.0~5.5V	2.0~5.5V
Operating temp.	-40~85C	-40~85C
Flash size /(number of guaranteed writes to flash)	64K Bytes/100,000 times	64K Bytes/100,000 times
RAM	4K Bytes	4K Bytes
ADC	12-bit x 7-CH (VDD, Vref)	12-bit x 7-CH (VDD, Vref)
Key-on wake up	8	8
Interrupt	External: 8 Internal: 28	External: 8 Internal: 28
HIRC / Accuracy	16MHz +/- 1% @ 0~50C +/- 1.5% @ -20~70C +/- 3% @ -40~85C	16MHz +/- 1% @ 0~50C +/- 1.5% @ -20~70C +/- 3% @ -40~85C
External Oscillator	1~16MHz	1~16MHz
BROR	4 levels	4 levels
LVD	8 levels (+/- 3%)*2	8 levels (+/- 3%)*2
Timers/ Counters	16bit x 8	16bit x 8
	WDT,TBT	WDT,TBT
PWM/PPG	16bit x 8	16bit x 8
Communication	UART x 3, SIO x 2, I2C x 2	UART x 3, SIO x 2, I2C x 2
On-chip debug	Yes	Yes
Package type	LQFP32 (7x7)	QFN32 (5x5)

Note 1:"VDD" indicates that the ADC uses VDD as the internal reference voltage; "Vref" indicates that the ADC uses an external reference voltage.

Note 2: SQ products has 8 levels LVD; the LVD accuracy can be ±3%. The detail please refer chapter "3.6 LVD characters.".

2.3 Block Diagram

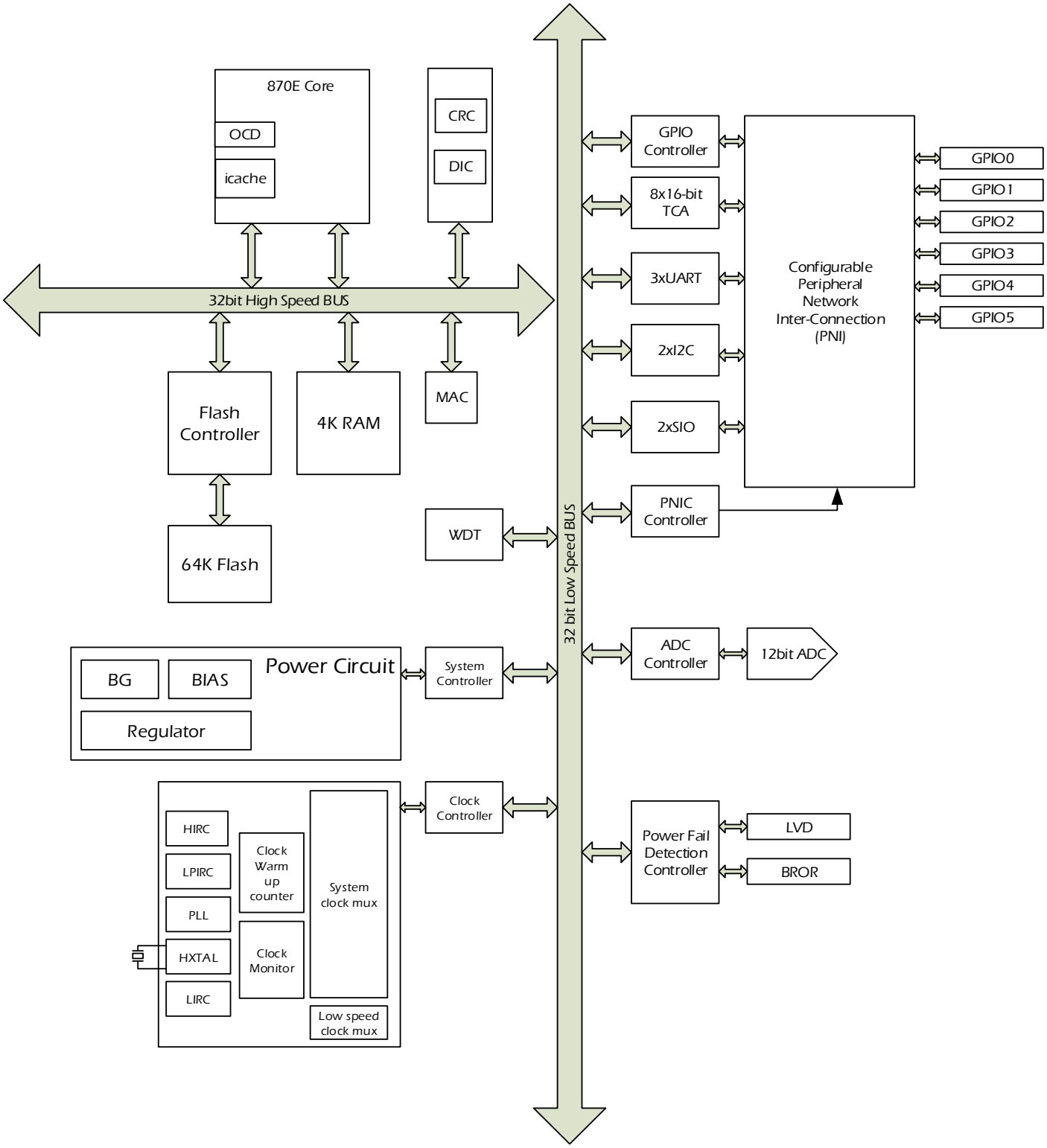


Figure 2- 2 SQ7613 Block Diagram

2.4 Pin Assignment/Description

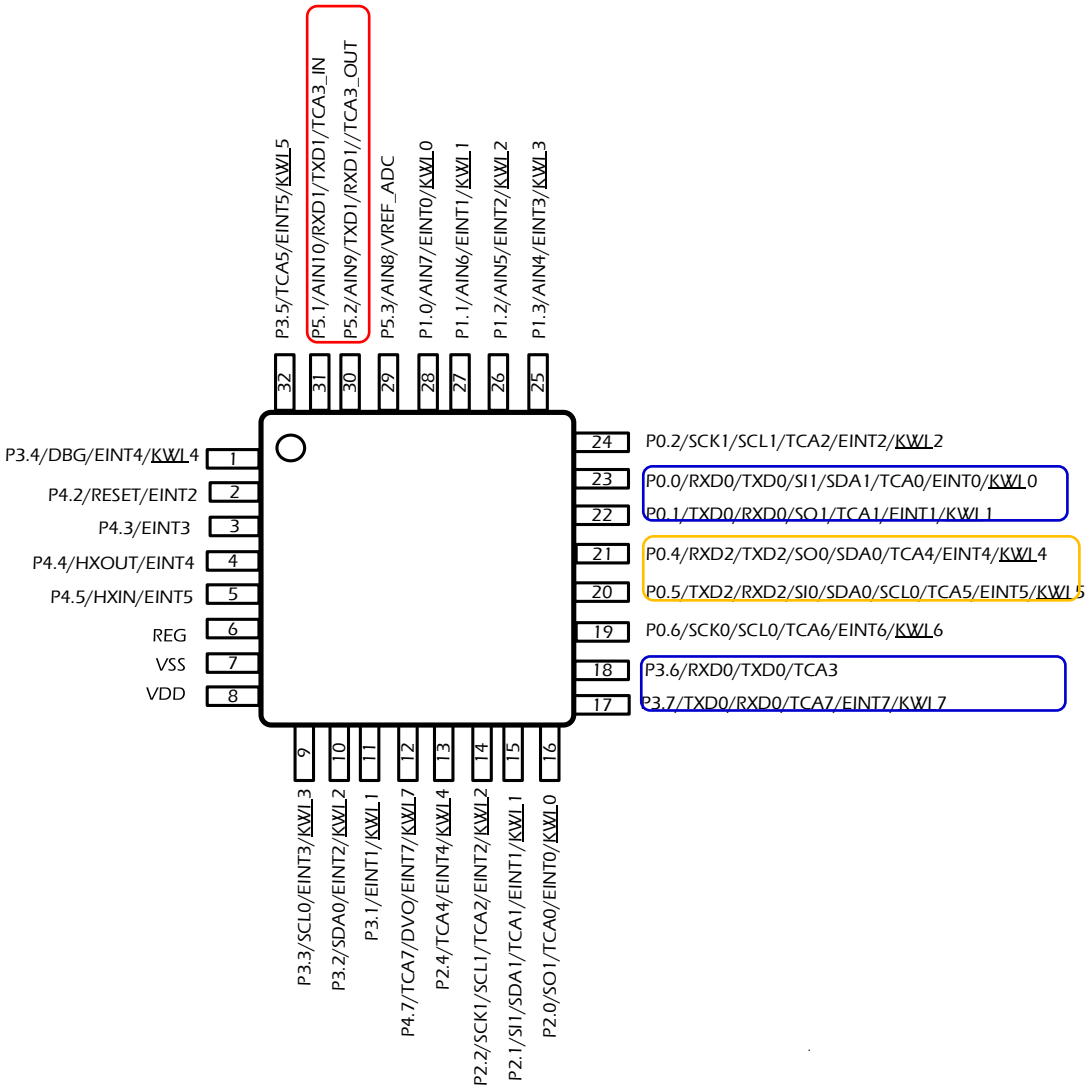


FIGURE 2- 2 PIN ASSIGNMENT OF SQ7613 LQFP-32

Note 1：SQ7613 support 4-wire emulation. User has to connect to P3.4/DBG, P4.2/RESET, VDD, GND under emulation.Suggest to reserve the emulation pins in the system board. Figure 2-4~ Figure 2-7 are reference circuits, other components added may affect emulation or function performance.

Note 2：SQ7613 can be programming (by writer) by below two type programming pins. Suggest to reserve the programming pins in the system board. Figure 2-4~ Figure 2-7 are reference circuits, other components added may affect progamming performance or functions.

(a) 4-wire OCDE Mode: the 4-wire OCDE pins are same as emulation pins (P3.4/DBG、P4.2/RESET、VDD、VSS). The OCDE programming time for 64Kbyte memory is around 16 seconds. iMQ MQ-Link and Flash Writer both suppot 4-wire OCDE progrmming .

(b) Bootloader : Bootloader pins are P0.0/ISPSI、P0.1/ISPSO、P0.2/ISPSCK、VDD、VSS. The Bootlader programming time for 64K byte memory is around 8 seconds. iMQ Flash Writer can suppot bootloader progrmming .

Note 3：TCAx support Input/output,only TCA3 is exception. P5.1/TCA3_IN supports input only, and P5.2/TCA3_OUT supports output only.

Note 4：UART / I2C/ SIO pins need to be paired as below. For example : if select P0.0 as RXD0, and P0.1 has to be TXD0.

UART0	TXD0/RXD0	P0.0/RXD0/TXD0 P0.1/TXD0/RXD0	P3.6/RXD0/TXD0 P3.7/TXD0/RXD0
UART1	TXD1/RXD1	P5.1/RXD1/TXD1 P5.2/TXD1/RXD1	
UART2	TXD2/RXD2	P0.4/RXD2/TXD2 P0.5/TXD2/RXD2	

I2C0	SCL0/SDA0	P0.6/SCL0 P0.5/SDA0	P0.5/SCL0 P0.4/SDA0	P3.3/SCL0 P3.2/SDA0
I2C1	SCL1/SDA1	P0.2/SCL1 P0.0/SDA1	P2.2/SCL1 P2.1/SDA1	

SIO0	SCK0/ SI0/ SO0	P0.6/SCK0 P0.5/SI0 P0.4/SO0	
SIO1	SCK1/SI1 / SO1	P0.2/SCLK1 P0.0/SI1 P0.1/SO1	P2.2/ SCLK1 P2.1/ SI1 P2.0/ SO1

Pin Name					Pin/Port function	
					Key-on Wakeup	External Interrupt
P0.0	P1.0	P2.0			KWI 0	EINT0
P0.1	P1.1	P2.1	P3.1		KWI 1	EINT1
P0.2	P1.2	P2.2	P3.2		KWI 2	EINT2
-	P1.3	-	P3.3 P3.6		KWI 3	EINT3
P0.4		P2.4	P3.4		KWI 4	EINT4
P0.5			P3.5		KWI 5	EINT5
P0.6			-	-	KWI 6	EINT6
-		-	P3.7	P4.7	KWI 7	EINT7
				P4.2		EINT2
				P4.3-		EINT3
				P4.4		EINT4
				P4.5		EINT5

TABLE 2-1 I/OS,KWI, EXTERNAL INTERRUPT TABLE

32-Pin No.	Pin Name/ Pin Option	I/O Type/	Function Description
1	P3.4/ DBG/EINT4/KWI 4	I/O (Type A)	P3.4 is bi-directional I/O pin, which are software configurable to be with pull-up or pull-down resistors. OCDE pin DBG, external interrupt EINT4, and wake up pin KWI 4 are pin-shared with P3.4.
2	P4.2/RESET/EINT2	I/O (Type A)	P10 is a bi-directional I/O pin, which is software configurable to be with pull-up resistors.. RESET and external interrupt EINT2 are pin-shared with P4.2. RESET is low-active.
3	P4.3/EINT3	I/O (Type A)	P4.3 is bi-directional I/O pin, which are software configurable to be with pull-up or pull-down resistors. External interrupt EINT3 is pin-shared with P4.3.
4	P4.4/HXOUT/EINT4	I/O (Type B)	P4.4 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors. HXOUT and external interrupt EINT4 are pin-shared with P4.4, and is connected to a high frequency external crystal for system clock.
5	P4.5/HXIN/EINT5	I/O (Type B)	P4.5 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors. HXIN and external interrupt EINT5 are pin-shared with P4.5, and is connected to a high frequency external crystal for system clock.
6	REG	(Type C)	Pin for connecting regulator output stabilization capacitance for internal operation. Connect the REG pin to VSS via a capacitor 1uF Note：REG pin cannot supply to external circuit.
7	VSS	GND	Ground
8	VDD	Power	Positive power supply
9	P3.3/SCL0/EINT3/ KWI3	I/O (TypeA)	P3.3 is bi-directional I/O pins, which are software configurable to be with pull-up or pull-down resistors. SCL0(I2C bus clock input/output 0), external interrupt EINT3, and wake up pin KWI 3 are pin-shared with P3.3.
10	P3.2/SDA0/EINT2/ KWI2	I/O (Type A)	P3.2 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. SDA0 (I2C bus data input/output 0),external interrupt EINT2, and wake up pin KWI 2 are pin-shared with P3.2
11	P3.1/EINT1/ KWI1	I/O (Type A)	P3.1 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. External interrupt EINT1, and wake up pin KWI1 are pin shared with P3.1. pin shared with P3.1.
12	P4.7/TCA7 / DVO/ EINT7/KWI7	I/O (Type A)	P4.7 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. 16-bit timer pin TCA7 ,DVO , external interrupt EINT7, and wake up pin KWI 7 are pin-shared with P4.7.

32-Pin No.	Pin Name/ Pin Option	I/O Type	Function Description
13	P2.4/TCA4/EINT4/ KWI4	I/O (Type A)	P2.4 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. 16-bit timer pin TCA4 , external interrupt EINT4, and wake up pin KWI 4 are pin-shared P2.4.
14	P2.2/SCK1/SCL1/TCA2/ EINT2/ KWI2	I/O (Type A)	P2.2 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. SCK1 (Serial clock input/output 1), SCL1(I2C bus clock input/output 1) ,16-bit timer pin TCA2 , external interrupt EINT2, and wake up pin and KWI2 are pin-shared P2.2.
15	P2.1/SI1/SDA1/TCA1/EINT1/ KWI1	I/O (Type A)	P2.1 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. SI1(serial data input 1), SDA1 (I2C bus data input/output 1) and 16-bit timer pin TCA1, external interrupt EINT1, and wake up pin KWI1 are pin-shared P2.1.
16	P2.0/SO1/TCA0/EINT0/ KWI0	I/O (Type A)	P2.0 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. SO1 (serial data output 1) , 16-bit timer pin TCA0 , external interrupt EINT0, and wake up pin KWI 0 are pin-shared with P2.0.
17	P3.7/TXD0/RXD0/TCA7/ EINT7/ KWI7	I/O (Type A)	P3.7 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. UART TXD0/RXD0 , 16-bit timer pin TCA7 , external interrupt EINT7, and wake up pin KWI7 are pin-shared with P3.7
18	P3.6/RXD0/TXD0/TCA3	I/O (Type A)	P3.6 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. UART RXD0/TXD0 , and 16-bit timer pin TCA3 are pin-shared with P3.6
19	P0.6/SCK0/SCL0/TCA6/ EINT6/ KWI6	I/O (Type A)	P0.6 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. SCK0 (Serial clock input/output 0), SCL0(I2C bus clock input/output 0) ,16-bit timer pin TCA6, external interrupt EINT6, and wake up pin KWI 6 are pin-shared P0.6
20	P0.5/TXD2/RXD2/SI0/SDA0/SCL0/TCA5/EINT5/ KWI5	I/O (Type A)	P0.5 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. UART TXD2/RXD2 , SI0(serial data input 0), SDA0 (I2C bus data input/output 0) , SCL0(I2C bus clock input/output 0),16-bit timer pin TCA5, external interrupt EINT5, and wake up pin KWI5 are pin-shared with P0.5
21	P0.4/RXD2/TXD2/SO0/SDA0/TCA4/EINT4/ KWI4	I/O (Type A)	P0.4 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. UART RXD2/TXD2 , SO0 (serial data output 0) SDA0 (I2C bus data input/output 0) , 16-bit timer pin TCA4 ,external interrupt EINT4, and wake up pin KWI4 are pin-shared with P0.4

32-Pin No.	Pin Name/ Pin Option	I/O Type	Function Description
22	P0.1/TXD0/RXD0/SO1/TCA1/ISPTxD/ISPSO/EINT1/ KWI1	I/O (Type A)	P0.1 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. UART TXD0/RXD0 , SO1 (serial data output 1), 16-bit timer pin TCA1 , external interrupt EINT1, wake up pin KWI 1, and ISPTxD/ ISPSO are pin-shared with P0.1
23	P0.0/RXD0/TXD0/SI1/SDA1/TCA0/ISPRxD/ISPSI/EINT0/ KWI0	I/O (Type A)	P0.0 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. UART RXD0/TXD0 , SI1 (serial data input 1), SDA1 (I2C bus data input/output 1) , 16-bit timer pin TCA0,external interrupt EINT0, wake up pin KWI0 and ISPRxD/ISPSI are pin-shared with P0.0.
24	P0.2/SCK1/SCL1/TCA2/ISPSCK/EINT2/KWI2	I/O (Type A)	P0.2 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. SCK1 (Serial clock input/output 1), SCL1(I2C bus clock input/output 1) ,16-bit timer pin TCA2 , external interrupt EINT2, wake up pin KWI2 and ISPSCK are pin-shared with P0.2.
25	P1.3/AIN4/EINT3/ KWI3	I/O (Type D)	P1.3 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN4 , external interrupt EINT3, and wake up pin KWI3 are pin-shared with P1.3.
26	P1.2/AIN5/EINT2/ KWI2	I/O (Type D)	P1.2 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN5 , external interrupt EINT2, and wake up pin KWI2 are pin-shared with P1.2.
27	P1.1/AIN6/EINT1/ KWI1	I/O (Type D)	P1.1 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN6 , external interrupt EINT1, and wake up pin KWI1 are pin-shared with P1.1.
28	P1.0/AIN7/EINT0/ KWI0	I/O (Type D)	P1.0 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN7, external interrupt EINT0, and wake up pin KWI0 are pin-shared with P1.0.
29	P5.3/AIN8/VREF_ ADC	I/O (Type D)	P5.3 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN8 and VREF_ADC is pin-shared with P5.3.
30	P5.2/AIN9/TXD1/RXD1/TCA3_OUT	I/O (Type D)	P5.2 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN9, UART TXD1/RXD1 ,and 16-bit timer pin TCA3_OUT are pin-shared with P5.2.
31	P5.1/AIN10/TXD1/RXD1/TCA3_IN	I/O (Type D)	P5.1 is a bi-directional I/O pin, which is software configurable to be with pull-up or pull-down resistors. ADC input AIN10, UART TXD1/RXD1 ,and 16-bit timer pin TCA3_IN are pin-shared with P5.1.
32	P3.5/ TCA5/EINT5/ KWI5	I/O (Type A)	P3.5 is bi-directional I/O pin, which are software configurable to be with pull-up or pull-down resistors. 16-bit timer pin TCA5, external interrupt EINT5, and wake up pin KWI5 are pin-shared with P3.5

Note 1 : User has to connect to P3.4/DBG, P4.2/RESET, VDD, GND under emulation.

Recommended external application circuits as below figures , please follow the recommendation to design.

1. ADC Input Filter：

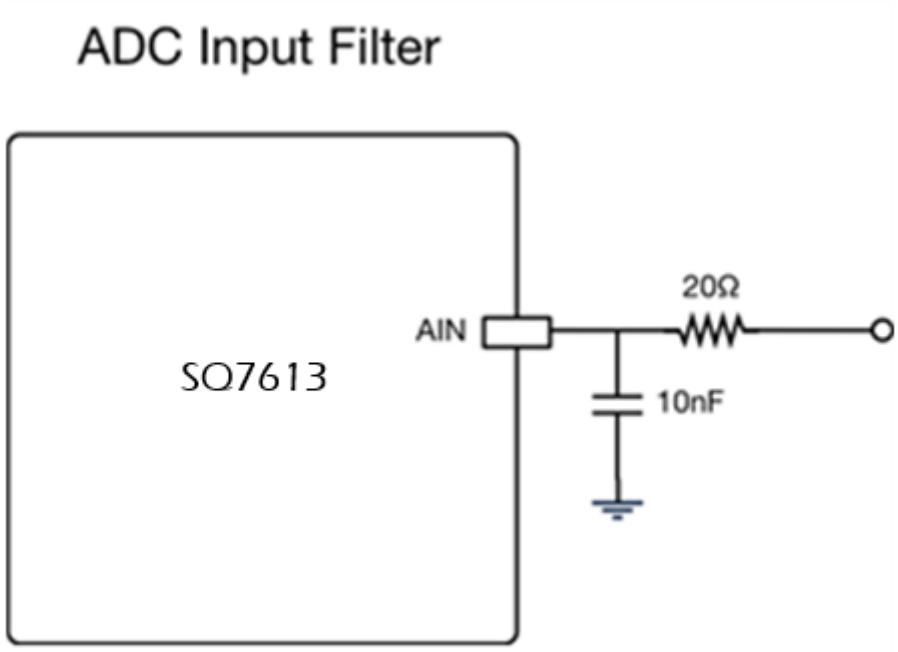


FIGURE 2- 4 SQ7613 RECOMMENDED EXTERNAL APPLICATION CIRCUITS (ADC INPUT FILTER)

2. External Crystal：

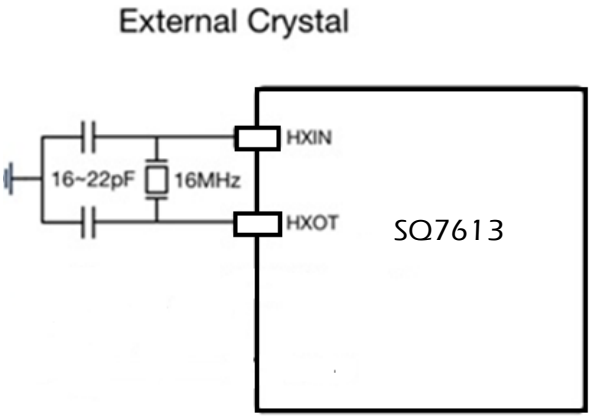


FIGURE 2- 5 SQ7613 RECOMMENDED EXTERNAL APPLICATION CIRCUITS (EXTERNAL CRYSTAL)

2. Power Decoupling：

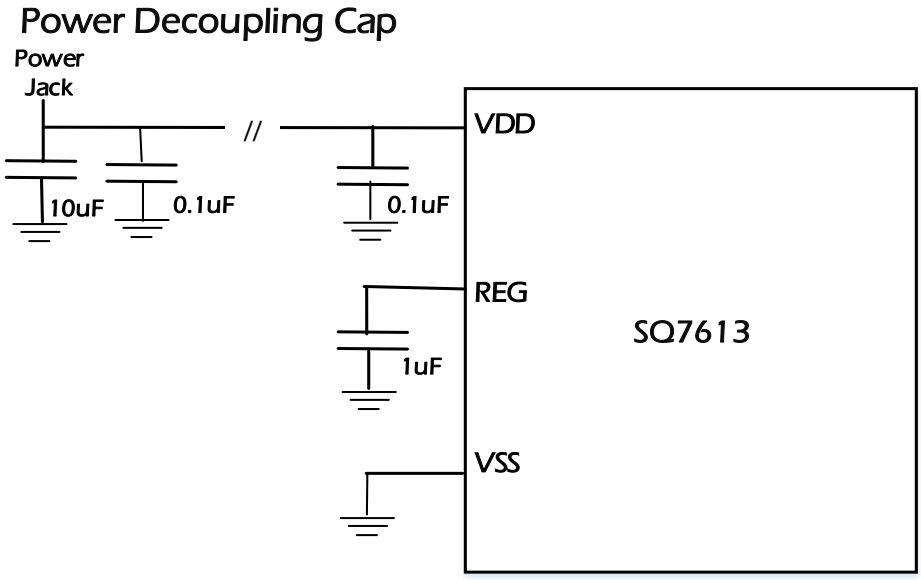


FIGURE 2- 6 SQ7613 RECOMMENDED EXTERNAL APPLICATION CIRCUITS (POWER DECOUPLING)

Note：The 0.1uF near the VDD should be as close as possible to the IC

4. RESET and DBG pin：

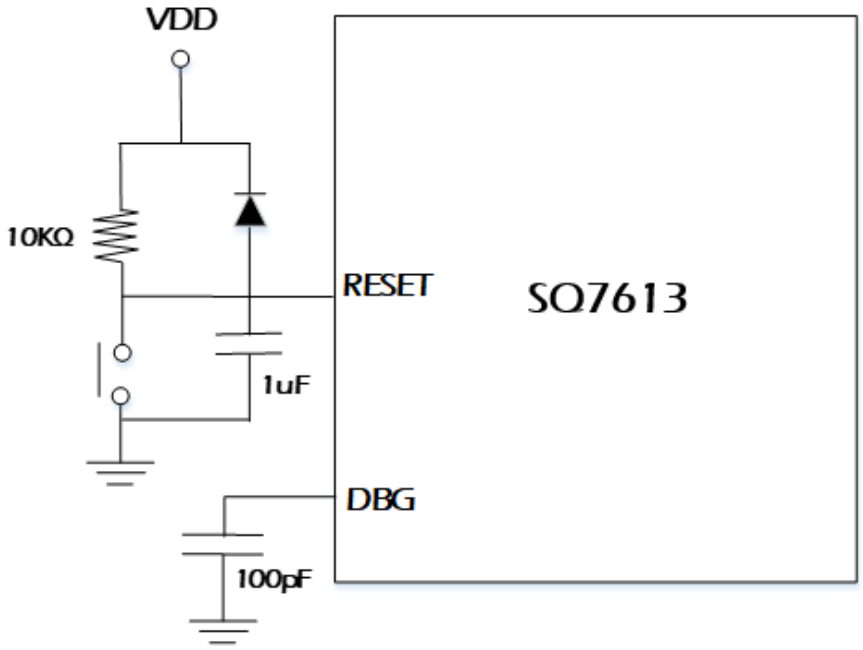
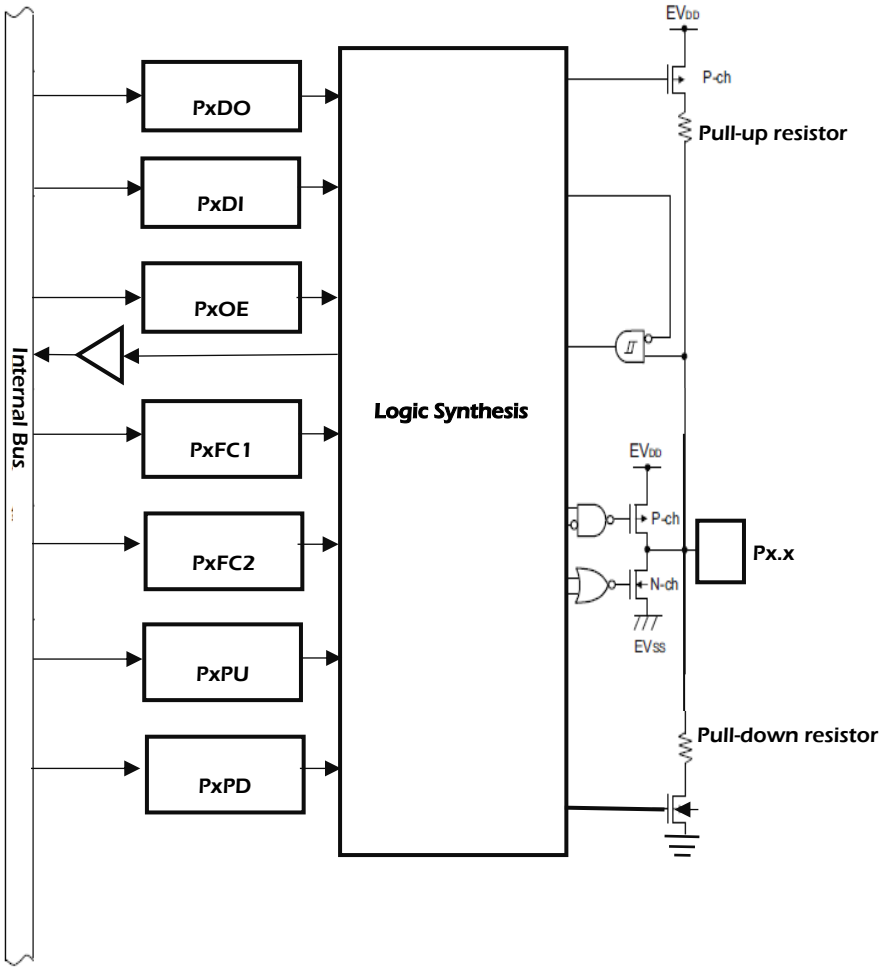
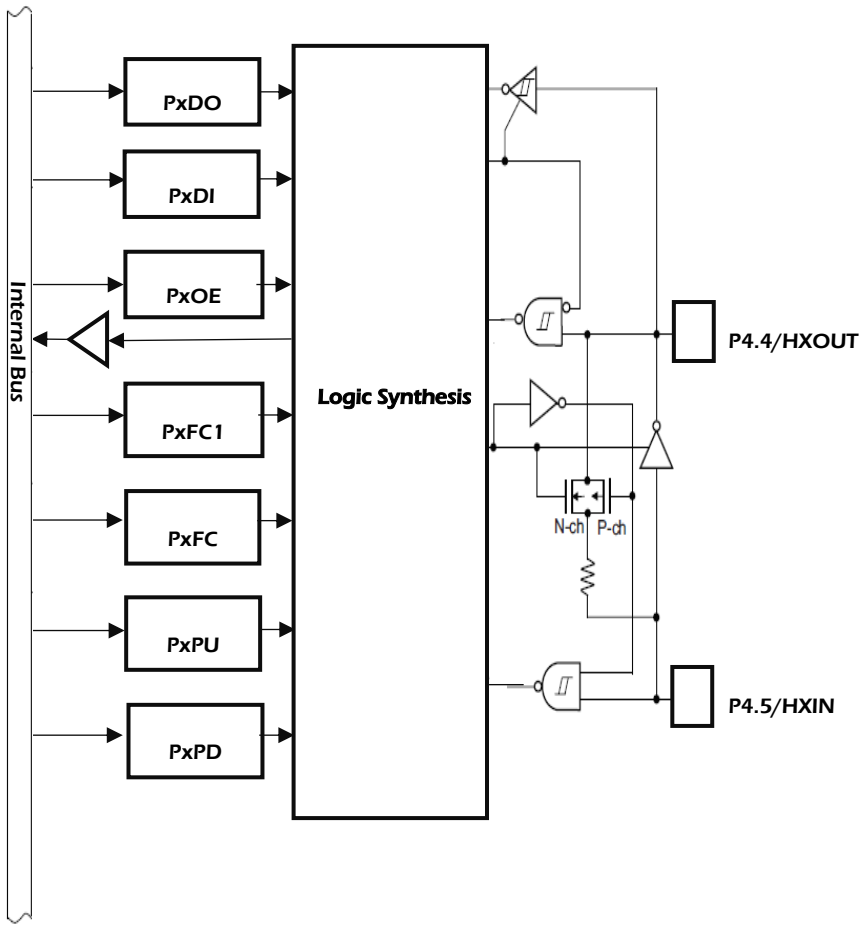
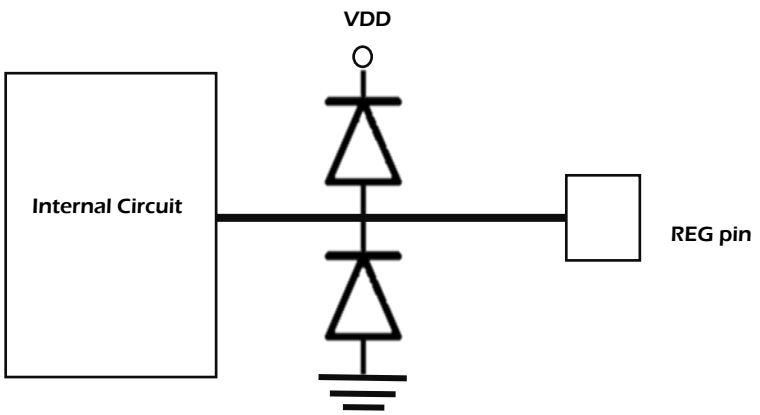
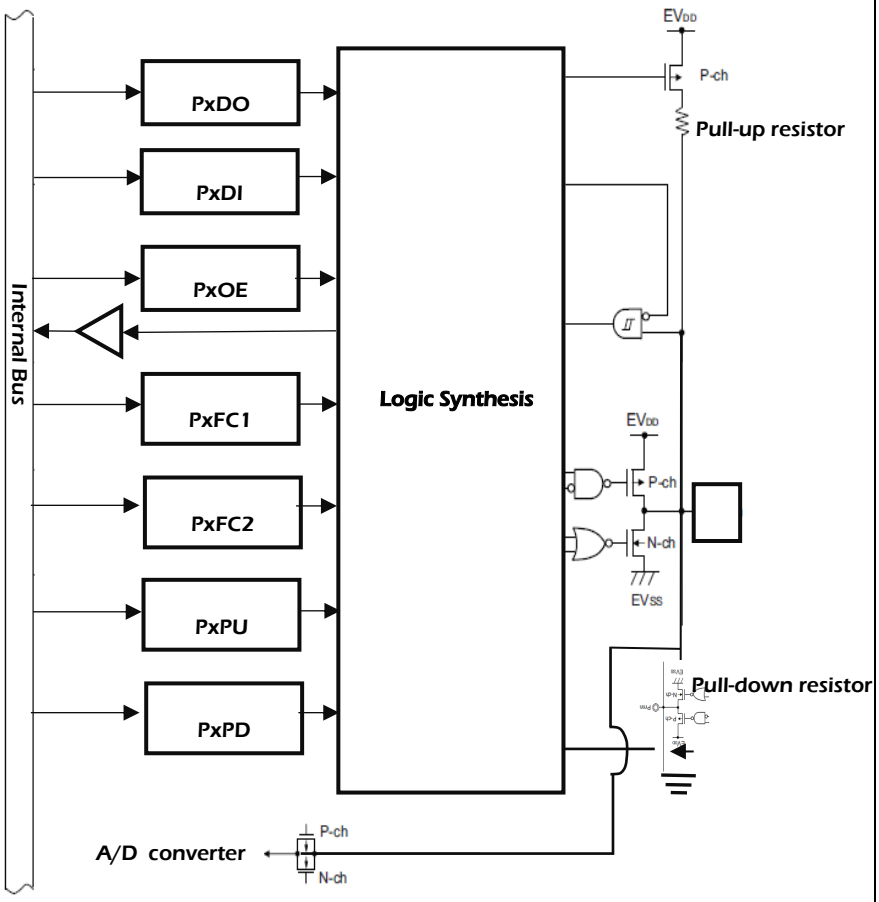


FIGURE 2- 7 SQ7613 RECOMMENDED EXTERNAL APPLICATION CIRCUITS (RESET AND DBG PIN)

2.5 I/O Circuit type

I/O Type	I/O Circuit	Description
Type A		GPIO(withoutADC input) DBG RESET

I/O Type	I/O Circuit	Description
Type B		Connect to external crystal

I/O Type	I/O Circuit	Description
Type C		REG
Type D		GPIO(with ADC input)

3. Electronic Characteristics

3.1 Absolute Maximum Ratings

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

(V_{SS} = 0V)

Parameter	Symbol	Pins	Ratings	Unit
Supply Voltage	V_{DD}		-0.3 to 6.0	V
Input Voltage	V_{IN}	All I/O pins	-0.3 to VDD+0.3	V
Output Current(total)		All I/O pins	100	mA
Storage Temperature	T_{STG}		-50 to 125	˚C

3.2 Operation Conditions

The following defines the electrical characteristics of the device when it is operated at voltage and temperature maximum/minimum values. Unless otherwise stated, the standard conditions were determined at "operating temperature 25 ° C and operating voltage VDD = 3.3 V".

3.2.1 Operation Conditions

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Supply Voltage	V_{DD}		2.0	3.3	5.5	V
Analog Supply Voltage	V_{DDA}		2.0	3.3	5.5	V
Reset Voltage (Note)	V_{RST}		1.89	1.95	2.01	V
Operating Temperature	Ta		-40	25	85	℃

Note：V_{RST} as the BROR 1st level

3.2.2 Clock Timing

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
External Clock Source						
High frequency external crystal (note 1)	f_{HXIN}		1		16	MHz
Internal Clock Source						
Low frequency internal reference clock	f_{LIRC}	T_A = 25℃	-25%	32	+ 25%	kHz
Low power internal reference clock	f_{LPIRC}	T_A = 25℃	- 1%	1	+ 1%	MHz
		T_A = 0~ 50℃ (note 2)	- 1%	1	+ 1%	
		T_A = -20~ 70℃ (note2)	- 1.5%	1	+ 1.5%	
		T_A = -40~ 85℃	- 3%	1	+ 3%	
High frequency internal reference clock	f_{HIRC}	T_A = 25℃	- 1%	16	+ 1%	MHz
		T_A = 0~ 50℃ (note 2)	- 1%	16	+ 1%	
		T_A = -20~ 70℃ (note 2)	- 1.5%	16	+ 1.5%	
		T_A = -40~ 85℃	- 3%	16	+ 3%	
Phase-locked loop	f_{PLL}	T_A = 25℃	(note 3)	24	(note 3)	MHz

Note1：The time froom high frequency external oscillation starts to fully oscillating is about 2.5ms (high frequency external oscillation is 16MHz, Topr= 25℃).

Note 2：The test condition is VDD= 5V ±10%

Note 3：The accuracy of fPLL is +/- 1%, which is the same as PLL reference clock source (16MHz f_{HXIN} or f_{LPIRC})

3.2.3 I/O Characteristics

VDD=3.3V ,Ta=-40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Input low voltage	V_{IL}		0		0.25 VDD	V
Input high voltage	V_{IH}		0.75 VDD		VDD	V
Output low voltage	V_{OL_050}	IOL=5 mA			0.4	V
Output high voltage	V_{OH_015}	IOH=1.5 mA	VDD-0.4			V
Output low current	I_{OL}	SDR=0,0.1xVDD	2.5	6.7		mA
		SDR=0,0.3xVDD	7	15		mA
Output high current	I_{OH}	SDR=0,0.9xVDD	1	2.4	-	mA
		SDR=0,0.7xVDD	3	5.8		mA
Pull-up Resistance	R_{PULLUP}		10	20	40	kΩ
Pull-low Resistance	R_{PULLDN}		10	20	40	kΩ

VDD=5V ,Ta=-40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Input low voltage	V_{IL}		0		0.25 VDD	V
Input high voltage	V_{IH}		0.75 VDD		VDD	V
Output low voltage	V_{OL_100}	IOL=10 mA			0.6	V
Output high voltage	V_{OH_035}	IOH= 3.5 mA	VDD-0.6			V
Output low current	I_{OL}	SDR=0,0.1xVDD	6	13.5		mA
		SDR=0,0.3xVDD	15	31		mA
Output high current	I_{OH}	SDR=0,0.9xVDD	2.5	4.8	-	mA
		SDR=0,0.7xVDD	6.5	12		mA
Pull-up Resistance	R_{PULLUP}		10	20	40	kΩ
Pull-low Resistance	R_{PULLDN}		10	20	40	kΩ

3.3 D.C. Characteristics

Operating @ 3.3V, Ta=-40~85℃						
Parameter	**Symbol**	**Condition**	**Min**	**Typ.**	**Max**	**Unit**
Normal Mode (LIRC on, code executing from flash)	I_{DD_N0}	LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=24MHz (PLL)	-	5.5	8.5	mA
	I_{DD_N1}	System clock is HIRC f_{HXIN}=0MHz,fsysclk=16 MHz (HIRC)	-	2.7	4.2	mA
	I_{DD_N2}	System clock is LIRC only LIRC enable, others are stopped. fsysclk=32KHz	-	0.7	1.1	mA
	I_{DD_N3}	System clock is HXTAL fsysclk=16MHz (HXTAL)	-	3.8	5.7	mA
Sleep Mode (LIRC on, CPU clock is stopped)	I_{DD_SL0}	LPIRC is PLL clock source f_{LPIRC}=1MHz ,fsysclk=24MHz (PLL)	-	2.7	4.1	mA
	I_{DD_SL1}	System clock is HIRC f_{HXIN}=0MHz,HIRC=16 MHz (HIRC)	-	1.3	2.1	mA
	I_{DD_SL2}	System clock is LIRC fsysclk=32KHz	-	0.7	1.1	mA
	I_{DD_SL3}	System clock is HXTAL fsysclk=16MHz (HXTAL)	-	2.0	3.1	mA
Deep Sleep Mode (LIRC on, CPU and RAM are retained.)	I_{DD_DS0}		-	1.0	-	uA

Operating @ 3.3V, Ta=25℃						
Parameter	**Symbol**	**Condition**	**Min**	**Typ.**	**Max**	**Unit**
Deep Sleep Mode (LIRC on, CPU and RAM are retained.)	I_{DD_DS0}		0.8	1.0	-	uA

Operating @ 5V,Ta=-40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Normal Mode (LIRC on, code executing from flash)	I_{DD_N0}	LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=24MHz (PLL)	-	5.5	8.5	mA
	I_{DD_N1}	System clock is HIRC f_{HXIN}=0MHz,fsysclk=16 MHz (HIRC)	-	2.7	4.2	mA
	I_{DD_N2}	System clock is LIRC only LIRC enable, others are stopped. fsysclk=32KHz	-	0.8	1.2	mA
	I_{DD_N3}	System clock is HXTAL fsysclk=16MHz (HXTAL)	-	3.8	5.7	mA
Sleep Mode (LIRC on, CPU clock is stopped)	I_{DD_SL0}	LPIRC is PLL clock source f_{LPIRC}=1MHz , fsysclk=24MHz (PLL)	-	2.7	4.1	mA
	I_{DD_SL1}	System clock is HIRC f_{HXIN}=0MHz,HIRC=16 MHz (HIRC)	-	1.3	2.1	mA
	I_{DD_SL2}	System clock is LIRC fsysclk=32KHz	-	0.8	1.2	mA
	I_{DD_SL3}	System clock is HXTAL fsysclk=16MHz (HXTAL)	-	2.0	3.1	mA
Deep Sleep Mode (LIRC on, CPU and RAM are retained.)	I_{DD_DS0}		-	1.1	-	uA

Operating @ 5V,Ta=25℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Deep Sleep Mode (LIRC on, CPU and RAM are retained.)	I_{DD_DS0}		0.9	1.1	-	uA

3.4 Power-on Reset Characteristics

Vss=0,Ta=-40~85℃

Symbol	Condition	Min	Typ.	Max	Unit
VPROFF	Power-on reset releasing voltage	1.89	1.95	2.01	V
VPRON	Power-on reset detecting voltage	1.84	1.90	1.96	V
tPROFF	Power-on reset releasing response time	-	0.01	0.1	ms
tPRON	Power-on reset detecting response time	-	0.01	0.1	ms
tPPW	Power-on reset minimum pulse width	1	-	-	ms
tPWUP	Warming-up time after a reset is clear and CPU ready(Note)	-	4	-	ms
tVDD	Power supply rise time	0.5	-	5	ms

Note: tPWUP dose not include BOOTROM code execute time. BOOTROM codeexecute time is around 50ms.

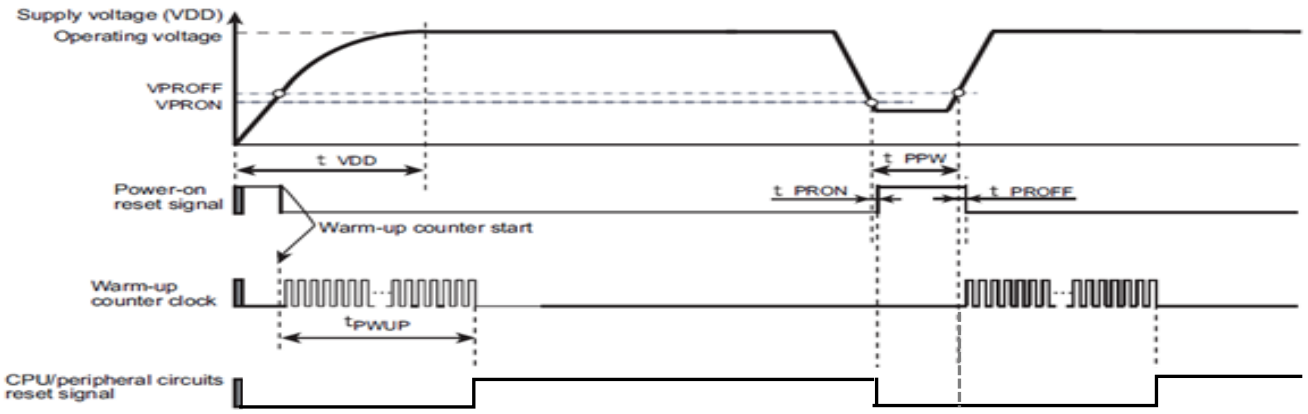


FIGURE 3- 1 OPERATION TIMING OF POWER ON RESET

3.5 BROR Characteristics

Ta=-40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
BROR	VBRORON1	1st level,BRORCFG=00	1.84	1.90	1.96	V
	VBROROFF1		1.89	1.95	2.01	V
	VBRORON2	2nd level,BRORCFG=01	2.18	2.25	2.32	V
	VBROROFF2		2.23	2.30	2.37	V
	VBRORON3	3rd level,BRORCFG=10	2.47	2.55	2.63	V
	VBROROFF3		2.52	2.60	2.68	V
	VBRORON4	4th level ,BRORCFG=11	2.67	2.75	2.83	V
	VBROROFF4		2.72	2.80	2.88	V

3.6 LVD Characteristics

Ta=-40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
LVD	VLVD1	Falling Mode,1st level, LVDCFG=000	1.94	2.00	2.06	V
	VLVD2	Falling Mode,2nd level, LVDCFG=001	2.28	2.35	2.42	V
	VLVD3	Falling Mode,3rd level, LVDCFG=010	2.57	2.65	2.73	V
	VLVD4	Falling Mode,4th level, LVDCFG=011	2.76	2.85	2.94	V
	VLVD5	Falling Mode,5th level, LVDCFG=100	3.06	3.15	3.24	V
	VLVD6	Falling Mode,6th level, LVDCFG=101	3.86	3.98	4.1	V
	VLVD7	Falling Mode,7th level, LVDCFG=110	4.07	4.20	4.33	V
	VLVD8	Falling Mode, 8th level, LVDCFG=111	4.37	4.50	4.64	V

3.7 ADC Characteristics

VREF_ADC=VDD
4.5V ≦ VDD ≦ 5.5V, T_A= -40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Resolution	RES_{ADC}			12		bits
Sampling Rate	f_{ADC}				470	KSPS
Differential Nonlinearity Error(DNL)	DNL_{ADC}				±2.5	LSB
Integral Nonlinearity Error(INL)	INL_{ADC}				±3.5	LSB
Gain error	E_{GAIN}				±5	LSB
Offset error	E_{OFFSET}				±4.5	LSB
Analog input voltage range	V_{ADC_RNG}				VDD	V
Analog Reference Voltage	V_{REF_ADC}		VDD(Note)			V
Note：VREF_ADC=VDD，Voltage range of VREF_ADC is 2~5.5V						

VREF_ADC=VDD
2V ≦ VDD ≦ 5.5V, T_A= -40~85℃

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Resolution	RES_{ADC}			12		bits
Sampling Rate	f_{ADC}				470	KSPS
Differential Nonlinearity Error(DNL)	DNL_{ADC}				±4	LSB
Integral Nonlinearity Error(INL)	INL_{ADC}				±5	LSB
Gain error	E_{GAIN}				±6	LSB
Offset error	E_{OFFSET}				±6	LSB
Analog input voltage range	V_{ADC_RNG}				VDD	V
Analog Reference Voltage	V_{REF_ADC}		VDDNote			V
Note：VREF_ADC=VDD，Voltage range of VREF_ADC is 2~5.5V						

3.8 Flash Characteristics

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

Parameter	Condition	Min	Typ.	Max	Unit
Number of guaranteed writes to flash memory		--	--	100,000	times
Flash memory write time	Write time (per byte)	--	--	7.5	μs
Flash memory erase time	chip erase	--	--	40	ms
	sector erase (1 sector = 512bytes)	--	--	5	

4. Central Processing Unit (CPU)

SQ7613 architecture is 870E core

- Rich instruction set optimized for compact C coding
  - Nine addressing modes
  - Multiply and divide instructions
  - Bit manipulation instructions
  - 16-bit ALU and load/store instructions
  - Jump and call instructions

- Register file supports fast context switches
  - Two banks of 8-bit and 16-bit general-purpose registers (GPRs)
  - Two sets of eight 8-bit GPRs
  - Two sets of two 16-bit GPRs
  - 16-bit Program Counter (PC)
  - 16-bit Stack Pointer (SP)
  - 7-bit Program Status Word (PSW)

- Memory
  - 64 KB Flash
  - 4 KB RAM

4.1 Symbols

Symbol	Description	Symbol	Description
A	A register	r,g	8-bit register
W	W register	rr, gg	16-bit register
B	B register	n	4-bit or 8-bit immediate data
C	C register	mn	16-bit immediate data
D	D register	d	Signed 5-bit or 8-bit displacement
E	E register	x,y	8-bit direct address
H	H register	vw, uz	16-bit direct address
L	L register	(XX)	Memory contents at the address specified by XX
WA	WA register	(xx+1, XX)	Two consecutive bytes from the memory location specified by XX
BC	BC register	b	Bit number (0 to 7)
DE	DE register	.b	Content of bit specified by b
HL	HL register	↔	Exchange
IX	IX register	+	Add
IY	IY register	-	Subtract
PC	Program Counter	x	Multiply
SP	Stack Pointer	÷	Division
PSW	Program Status Word	&	Bitwise AND
JF	Jump Status flag	|	Bitwise OR
CF	Carry flag	^	Bitwise exclusive OR
HF	Half carry flag	null	No operation
SF	Sign flag	$	Start address of instruction being executed
VF	Overflow flag	(src)	Source memory
/CF	Inverse of carry flag	(dst)	Destination memory
IMF	Interrupt Master Enable flag	(srcdst)	Source and destination memory
NxtOp	Address of next operation	RBS	Register Bank Selector

TABLE 4- 1 SYMBOLS USED IN THIS DOCUMENT

Symbol	Description	Symbol	Description
ADD	Add	OR	Logical OR
ADDC	Add with carry	POP	Pop up
AND	Logical AND	PUSH	Push down
CALL	Call	RET	Return from subroutine
CALLV	Vector call	RETI	Return from maskable interrupt service routine
CLR	Clear bit/byte	RETN	Return from non-maskable interrupt service routine
CMP	Compare	ROLC	Rotate left through carry
DAA	Decimal adjust for 8-bit addition	ROLD	Rotate left digit
DAS	Decimal adjust of 8-bit subtraction	RORC	Rotate right through carry
DEC	Decrement byte/word (Register)	RORD	Rotate right digit
DI*	Disable maskable interrupt	SET	Bit test and set
DIV	Divide byte quotient	SHLC	Logical shift left
EI*	Enable interrupt	SHLCA	Arithmetic shift left
INC	Increment byte/word (Register)	SHRC	Logical shift right
J*	Optimized jump	SHRCA	Arithmetic shift right
JP	Absolute jump	SUB	Subtract
JR	Relative jump	SUBB	Subtract with borrow
JRS	Short relative jump	SWAP	swap nibble
LD	Load bit/byte/word (Register)/effective address	SWI	Software interrupt
LDW	Load word (Memory)	TEST*	Bit test
MUL	Multiply	XCH	Exchange
NEG	Negate	XOR	Logical exclusive OR
NOP	No operation	OR	Logical OR

TABLE 4- 2 INSTRUCTION MNEMONICS

Note: Instructions marked with an asterisk (*) are extended assembler machine instructions

4.2 Core Register

The register banks and the core registers are depicted in the figure below.

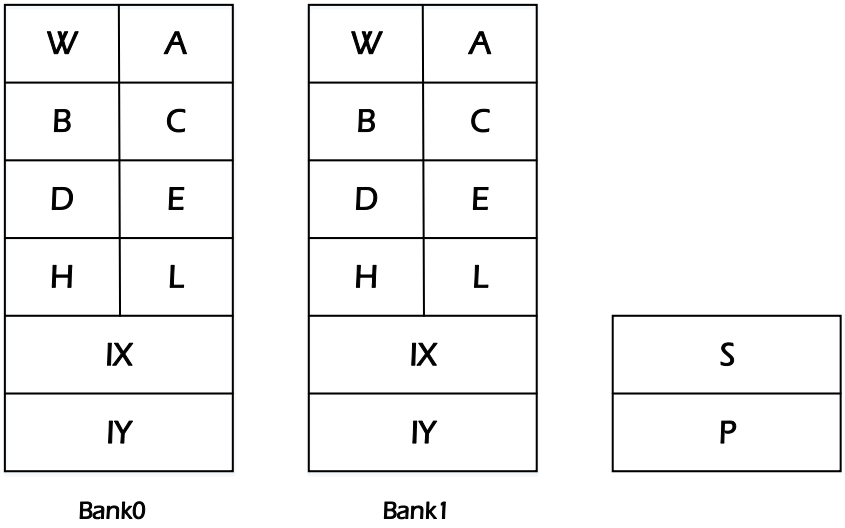


FIGURE 4- 1 CORE REGISTER

4.2.1 8-bit General Purpose Registers

There are two duplicate banks of eight 8-bit registers. They are W, A, B, C, D, E, H and L. These registers can be paired to be used as 16-bit registers. The 16-bit register pairs are WA, BC, DE and HL. These registers are reset to zeros by system reset.

The following sections describe special usages of these registers.

A Register

This 8-bit register is also used in bit manipulation instructions, and in instructions that support PC-Relative Register Indirect Addressing.

Example：

```
1.  SET     (0x56).A    ; The bit specified by A of the memory
                        location 0x0056 is set to 1.

2.  LD      A, (PC+A)   ; Load the content of the memory
                        address PC+A into A register
```

C Register

This 8-bit register is used to hold the divisor in divide instructions. It is also used as an offset register in Register Indexed Addressing.

Example：

1. DIV WA, C ; C is the divisor
2. LD A, (HL + C); C is an offset register

DE Register

This 16-bit register is used to hold the address of a memory location, in Register Indirect Addressing.

Example：

LD A, (DE) ; DE is the register that holds the address.

HL Register

This 16-bit register is used to hold the address of a memory location in Register Indirect Addressing. It is also used as an index register in Indexed Addressing.

Example：

LD A, (HL) ; HL HL is the register that holds the address
LD A, (HL+0x52) ; HL is an index register
LD A,(HL+C) ; HL is an index register

4.2.2 16-bit General-Purpose Register

There are two duplicate banks of two 16-bit registers called IX and IY. Besides general use, in Register Indirect Addressing, these registers hold the address of the memory location. In Indexed Addressing, they are used as an index register. These registers are reset to zeros by system reset.

Example：

LD A, (IX) ; IX is the register that holds the address
LD A(IY+0x52) ; IY is an index register
LD IX(0x3A) ; IX is a general-purpose register

The core register function can be used to store general purpose registers during non-multiple interrupt operations. At the beginning of the interrupt, set the operation instruction (such as the Example： LD RBS, 1), the core register function will be stored or converted. After the end of the interrupt, there is no need to re-

execute the operation instruction. The RETI instruction will automatically restore the core register to the register at the time of execution of the main task according to the contents of the PSW.

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

Example：The main task uses BANK0 and is converted to BANK1 by instructions.

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

4.2.3 Program Status Word (PSW)

The PSW register resides at address 0x003F. It consists of the following six flags：

Jump Status Flag, JF
Zero Flag, ZF
Carry Flag, CF
Half Carry Flag, HF
Sign Flag, SF
Overflow Flag, VF

Besides the general load instructions, dedicated instructions are available to access the PSW. The table below summarizes how the flags are used in conditional jump instructions, such as JJ cc, a and JRS cc,a instructions.

Conditional Code	Description	Flag Condition
T	1	JF = 1
F	0	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 ^ ZF) = 1
GT	Unsigned greater than	(CF ^ ZF) = 0
SLT	Signed less than	(SF ^ VF) = 1
SGE	Signed greater than or equal to	(SF ^ VF) = 0
SLE	Signed less than or equal to	ZF ^ (SF ^ VF) = 1
SGT	Signed greater than	ZF ^ (SF ^ VF) = 0

TABLE 4- 3 CONDITIONAL JUMP WITH PSW FLAGS

4.2.4 Stack Pointer (SP)

The SP is a 16-bit register that holds the address of the next available location on the stack. The SP is post-decremented in the following operations: Calls, push operations and interrupts. It is pre-incremented in the following operations: returns from subroutines and interrupts, and pop operations.

4.2.5 Program Counter (PC)

The PC is a 16-bit register that holds the address of the next instruction to be executed in the code area. Upon exiting reset, the CPU loads the reset vector stored in the vector table into the PC. The CPU then starts fetching and executing code from the address pointed to by the program counter.

4.3Addressing Mode

The SQ7613 has nine addressing modes. Some addressing modes have more than one type.

Mode	Number of types
Register Indirect	7
Direct	2
Register	1
Immediate	1
Relative	2
Absolute	1
Vector	1
Direct Bit	2
Register Indirect Bit	1
Total	18

TABLE4- 4 ADDRESSING MODE

4.3.1 Register Indirect Addressing

Register Indirect Addressing (HL), (DE), (IX), (IY)

The effective address is specified by the contents of a 16-bit register pair HL, DE, IX or IY.

Example：LD A,(HL)

Register Indirect with 8-bit Displacement Addressing (HL+d), (IX+d), (IY+d)

The effective address is formed by sign-extending the 8-bit displacement d in the instruction code and adding it to the contents of the 16-bit register HL, IX or IY.

Example： LD A, (HL + 0x12)

Register Indexed Addressing (HL + C)

The effective address is formed by sign-extending the contents of the C register and adding it to the contents of HL register.

Example： LD A, (HL + C)

Stack Pointer Indirect with Auto-Pre-Increment Addressing (+SP)

The contents of the SP is incremented to form an effective address. Incrementing the SP does not affect the flag bits. Note, this addressing mode can only be used to specify the source memory address.

Example： LD A, (+SP)

Stack Pointer Indirect with Auto-Decrement Addressing (SP-)

The SP holds the effective address. After the data manipulation, the contents of the SP is automatically decremented. This addressing mode can only be used to specify the destination memory address.

Example： LD (SP-),A

Stack Pointer Indirect with 8-bit Displacement Offset Addressing (SP+d)

The effective address is formed by sign-extending the 8-bit displacement d in the instruction code and adding it to the contents of the Stack Pointer SP.

Example：LD WA, (SP + 0xD6)

PC-Relative Register Indirect Addressing (PC+A)

The effective address is formed by sign-extending the contents of the A register and adding it to the contents of the Program Counter. This addressing mode can only be used to specify the source address.

Example：LD A, (PC + A)

4.3.2 Direct Addressing

8-Bit Direct Addressing (x)

The effective address is specified directly by the 8-bit value x in the instruction code. The address is in the range 0x0000 to 0x00FF.

16-Bit Direct Addressing (vw)

The effective address is specified directly by the 16-bit value vw in the instruction code. The address is in the range 0x0000 to 0xFFFF.

Example：LD A, (0x5678)

4.3.3 Register Addressing (r or rr)

The register specifier in the instruction opcode specifies which register is to be accessed.

Example：LD A, B

4.3.4 Immediate Addressing (n or mn)

The register specifier in the instruction opcode specifies which register is to be accessed.

Example：LD A, 0x53

4.3.5 Relative Addressing

PC-Relative with 8-Bit Displacement Addressing

The effective address is formed by sign-extending the 8-bit displacement d in the instruction opcode and adding it to the contents of the Program Counter. The JR instruction is the only instruction that has this addressing mode.

Example:JR $ + 2 + 0x35

PC-Relative with 5-Bit Displacement Addressing

The effective address is formed by sign-extending the 5-bit displacement d in the instruction opcode and adding it to the contents of the Program Counter. The JRS instruction is the only instruction that has this addressing mode.

Example:JRS $ + 2 + 0x14

4.3.6 Absolute Addressing

The effective address is specified by a 16-bit value in the instruction opcode.

Example：JR 0x0F1A3

4.3.7 Vector Addressing

The 4-bit operand is multiplied by 2 and added to the top address of the vector call table to form a pointer to a location where a 16-bit jump destination address (vector address) is located. Only CALLV instruction has the addressing mode.

4.3.8 Direct Bit Addressing

Register Bit Addressing

The register and bit specifiers in the instruction opcode specify a bit position in a register whose value should be tested or changed.

Example：SET A.3

Memory Bit Addressing

In Memory Bit addressing mode, the bit specifier in the instruction code specifies the bit in the memory location pointed to by (HL), (DE), (IX), (IY), (HL+d), (IX+d), (IY+d), (HL+C), (+SP), (SP+d), (PC+A), (x) or (vw). A bit manipulation is performed on the specified bit.

Example：SET (HL).1

4.3.9 Register Indirect Bit Addressing

In Memory Bit addressing mode, low-order 3 bits of the A register specify the bit in the memory location pointed to by (HL), (DE), (IX), (IY), (HL + d), (IX + d), (IY + d), (HL + C), (+SP), (SP + d), (PC + A), (x) or (vw). A bit manipulation is performed on the specified bit.

Example： SET (HL).A

4.4 Instruction Pipeline Stages

There are three stages in the core execution pipeline. Instructions that involve a memory read access have an additional memory access cycle. The instruction set architecture can be categorized as follows:

- Register-to-register operations
- Register-to-memory operations
- Memory-to-register operations
- Memory-to-memory operations
- Jump
- Subroutine Call and Return
- Software interrupt (SWI)

4.4.1 Register-to-Register Operations

This type of operations has three-stage pipeline.

Register-to-Register Operations	
Symbol	Stage Work
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where an intended operation is carried out in the execution unit. The result is written back to the register file at the end of the execution cycle.

4.4.2 Register-to-Memory Operations

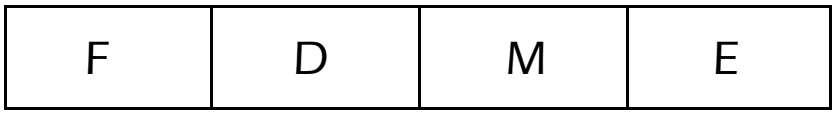
This type of operations has three-stage pipeline.

Register-to-Memory Operations	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where an intended operation is carried out in the execution unit. The result is written back to the store data buffer at the end of the execution cycle. The buffer store data is sent to the data bus interface in the next cycle.

4.4.3 Memory-to-Register Operations

This type of operations has a memory read. Therefore, there is an additional memory access cycle.

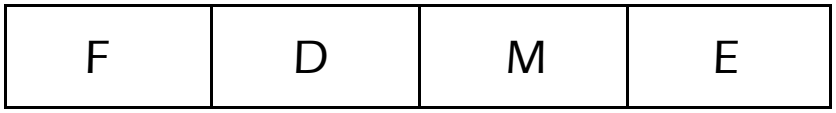
Memory-to-Register Operations	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
M	This is the memory access cycle where the address is generated and sent to the data bus interface.
E	This is the execution stage where the load data is returned and an intended operation is carried out in the execution unit. The result is written back to the register file at the end of the execution cycle.



4.4.4 Memory-Memory Operations

This type of operations has a memory read followed by a memory write cycle. Therefore, there is an additional memory access cycle.

Memory-Memory Operations	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
M	This is the memory access cycle where the address is generated and sent to the data bus interface.
E	This is the execution stage where the load data is returned and an intended operation is carried out in the execution unit. The result is written back to the store data buffer at the end of the execution cycle. The buffer store data is sent to the data bus interface in the next cycle.



4.4.5 Jump

There are two types of jumps:

Type 1		Type 2	
Addressing Mode	Opcode	Addressing Mode	Opcode
Register Addressing	JP gg	Register Indirect Addressing	JP (src*) *src: DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A
Immediate Addressing	JP mn	Direct Addressing	JP (src*) *src: x, vw
Relative Addressing	1) PC-Relative with 8-BIt Displacement Addressing JR T,$+2+d, etc. 2) PC-Relative with 5-Bit Displacement Addressing JRS T, $+2+d, etc.		
Absolute Addressing	JP 0x0F1A3		

◆ Type 1 pipeline：

This type of jump has three pipeline stages.

Type 1 pipeline	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where an intended operation is carried out in the execution unit. The result is written back to the register file at the end of the execution cycle.

◆ Type 2 pipeline：

This type of jump has five pipeline stages.

Type 2 pipeline	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where an indirect address in generated in the Data Unit and sent to the data bus interface.
E+1	The jump target address is returned and stored in the load data buffer.
E+2	Instruction fetch address is generated.

4.4.6 Subroutine Call and Return

There are two types of calls:

Type 1		Type 2	
Addressing Mode	Opcode	Addressing Mode	Opcode
Register Addressing	-	Register Addressing	-
Immediate Addressing	-	Direct addressing	-
Absolute Addressing	CALL 0x0F1A3		
Vector Addressing	CALLV 0x9		

◆ Type 1 pipeline: :

This type of jump has three pipeline stages.

Type 1 pipeline	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where an intended operation is carried out in the execution unit. The result is written back to the register file at the end of the execution cycle.

◆ Type 2 pipeline: :

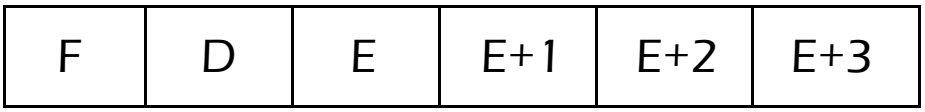
This type of jump has five pipeline stages.

Type 2 pipeline	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where an indirect address in generated in the Data Unit and sent to the data bus interface.
E+1	The jump target address is returned and stored in the load data buffer.
E+2	Instruction fetch address is generated.

4.4.7 SWI

The SWI instruction has six pipeline stages.

SWI	
Symbol	Stage description
F	This is the instruction fetch stage where instruction opcodes are returned from the code memory.
D	This is the instruction decode stage where an instruction is decoded and forwarded to functional units.
E	This is the execution stage where the SWI instruction is executed. The address of the interrupt vector is received in the Instruction Unit and the instruction fetch address is generated. The PSW is pushed onto stack in this cycle.
E+1	The interrupt vector is returned and entered the instruction buffer as a jump instruction. The address of the next opcode is pushed onto stack in this cycle.
E+2	The jump target address is decoded.
E+3	Instruction Unit vectors to SWI interrupt routine.



4.5 Instruction Set Summary

The instruction set is divided into six groups of instructions. Their instruction mnemonics and execution cycle are summarized in this section.

-Move/Load/Store and Exchange Instructions
-ALU Instructions
-Shift/Rotate and Nibble Manipulation Instructions
-Bit and Flag Manipulation Instructions
- Jump Instructions
- Call, Return, Software Interrupt and No Operation

4.5.1 Move/Load/Store and Exchange Instructions

Operation	Description	Assembler	Cycles
Move	8-bit register to register operation	ld r, g	1
	16-bit register to register operation	ld rr, gg	1
	8-bit immediate to register	ld r, n	1
	16-bit immediate to register	ld rr, mn	1
	16-bit SP register move operation	ld SP, SP+d	1
	16-bit SP register move operation	ld SP, SP-d	1
Load	8-bit memory to register operation	ld r, (src*)	1
	16-bit memory to register	ld rr, (src*)	1
Store	8-bit register to memory	ld (dst*), r	1
	16-bit register to memory	ld (dst*), rr	1
	8-bit immediate to memory	ld (dst*), n	1
	16-bit immediate to memory	ld (dst*), mn	1
Push	16-bit register to memory stack	push rr	1
	8-bit PSW register to memory stack	push PSW	1
Pop	16-bit register from memory stack	pop rr	1
	8-bit PSW register from memory stack	pop PSW	1
Exchange	8-bit register to register	xch r,g	1
	16-bit register to register	xch rr,gg	1
	8-bit register to memory	xch r,(src*)	1
	16-bit register to memory	xch rr,(src*)	1

Operation	Description	Assembler	Cycles
Note：src: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A dst: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, SP-			

TABLE 4- 5 MOVE/LOAD/STORE AND EXCHANGE INSTRUCTIONS

4.5.2 ALU Instructions

Operation	Description	Assembler	Cycles
Compare	8-bit register to an immediate value	cmp g,n	1
	16-bit register to an immediate value	cmp gg,mn	1
	8-bit register to another register	cmp r,g	1
	16-bit register to another register	cmp rr,gg	1
	8-bit register to memory content	cmp r,(src*)	1
	8-bit memory content to an immediate value	cmp (src*),n	1
	16-bit register to a memory content	cmp rr,(src*)	1
Add	8-bit register to an immediate value	add g,n	1
	16-bit register to an immediate value	add gg,mn	1
	8-bit register to another register	add r,g	1
	16-bit register to another register	add rr,gg	1
	8-bit register to memory content	add r,(src*)	1
	8-bit memory content to an immediate value	add (srcdst*),n	1
	16-bit register to a memory content	add rr,(src*)	1
Add with carry	8-bit register to an immediate value	addc g,n	1
	16-bit register to an immediate value	addc gg,mn	1
	8-bit register to another register	addc r,g	1
	16-bit register to another register	addc rr,gg	1
	8-bit register to memory content	addc r,(src*)	1
	8-bit memory content to an immediate value	addc (srcdst*),n	1
	16-bit register to a memory content	addc rr,(src*)	1
Substract	8-bit register to an immediate value	sub g,n	1
	16-bit register to an immediate value	sub gg,mn	1
	8-bit register to another register	sub r,g	1
	16-bit register to another register	sub rr,gg	1
	8-bit register to memory content	sub r,(src*)	1

Operation	Description	Assembler	Cycles
	8-bit memory content to an immediate value	sub (src*),n	1
	16-bit register to a memory content	sub rr,(src*)	1
Substract with borrow	8-bit register to an immediate value	subb g,n	1
	16-bit register to an immediate value	subb gg,mn	1
	8-bit register to another register	subb r,g	1
	16-bit register to another register	subb rr,gg	1
	8-bit register to memory content	subb r,(src*)	1
	8-bit memory content to an immediate value	subb (srcdst*),n	1
	16-bit register to a memory content	subb rr,(src*)	1
bitwise logical AND	8-bit register to an immediate value	and g,n	1
	16-bit register to an immediate value	and gg,mn	1
	8-bit register to another register	and r,g	1
	16-bit register to another register	and rr,gg	1
	8-bit register to memory content	and r,(src*)	1
	8-bit memory content to an immediate value	and (srcdst*),n	1
	16-bit register to a memory content	and rr,(src*)	1
bitwise logical OR	8-bit register to an immediate value	or g,n	1
	16-bit register to an immediate value	or gg,mn	1
	8-bit register to another register	or r,g	1
	16-bit register to another register	or rr,gg	1
	8-bit register to memory content	or r,(src*)	1
	8-bit memory content to an immediate value	or (srcdst*),n	1
	16-bit register to a memory content	or rr,(src*)	1
bitwise logical exclusive-OR	8-bit register to an immediate value	xor g,n	1
	16-bit register to an immediate value	xor gg,mn	1
	8-bit register to another register	xor r,g	1
	16-bit register to another register	xor rr,gg	1
	8-bit register to memory content	xor r,(src*)	1
	8-bit memory content to an immediate value	xor (srcdst*),n	1
	16-bit register to a memory content	xor rr,(src*)	1
Increment	8-bit register operation	dec r	1
	16-bit register operation	dec rr	1

Operation	Description	Assembler	Cycles
	8-bit memory operation	dec (srcdst*)	1
Decrement	8-bit register operation	dec r	1
	16-bit register operation	dec rr	1
	8-bit memory operation	dec (srcdst*)	1
Add with 8-bit packed BCD number	8-bit register operation	daa g	1
Subtract with 8-bit packed BCD number	8-bit register operation	das g	1
Multiply	8-bit register operation	mul mreg1*,mreg2*	1
Divide	8-bit register operation	div dreg1*, C	9
Negate	16-bit register operation	neg CS, gg	1
Note : src: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A srcdst: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A mreg1: W, B, D, H mreg2: A, C, E, L dreg1: WA, DE, HL			

TABLE 4- 6 ALU INSTRUCTIONS

4.5.3 Shift/Rotate and Nibble Manipulation Instructions

Operation	Description	Assembler	Cycles
Shift	8-bit register, logical shift left by one	shlc g	1
	8-bit register, logical shift right by one	shrc g	1
	16-bit register, arithmetic shift left by one	shlca gg	1
	16-bit register, arithmetic shift right by one	shrca gg	1
Rotate	8-bit register, rotate left with carry flag	rolc g	1
	8-bit register, rotate right with carry flag	rorc g	1
	8-bit memory-to-memory, rotate left and concatenate	rold A,(src*)	1
	8-bit memory-to-memory, rotate right and concatenate	rord A,(src*)	1
Swap	8-bit register, swap the high and low nibbles	swap g	1
Note： src: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A			

TABLE 4- 7 SHIFT/ROTATE AND NIBBLE MANIPULATION INSTRUCTIONS

4.5.4 Bit and Flag Manipulation Instructions

Operation	Description	Assembler	Operation
Bit set	Set a bit of an 8-bit register using a 3-bit b field	set g.b	1
	Set a bit of a memory content using a 3-bit b filed	set (src*).b	1
	set a bit of a memory content using the loworder 3 bits of A register	set (src*).A	1
Bit clear	clear a bit of an 8-bit register using a 3-bit b field	clr g.b	1
	clear a bit of a memory content using a 3-bit b filed	clr (src*).b	1
	clear a bit of a memory content using the loworder 3 bits of A register	clr (src*).A	1
Bit complement	complement a bit of an 8-bit register using a 3bit b field	cpl g.b	1
	complement a bit of a memory content using a 3-bit b filed	cpl (src*).b	1
	complement a bit of a memory content using the low-order 3 bits of A register	cpl (src*).A	1
Bit Test	Test a bit of an 8-bit register using a 3-bit b field	test g.b	1
	Set a bit of a memory content using a 3-bit b filed	test (src*).b	1
	set a bit of a memory content using the low order 3 bits of A register	test (src*).A	1
Load Carry flag	Load the value of bit b of an 8-bit register into the Carry flag	ld CF, g.b	1
	Load the value of bit b in a memory location into the Carry flag	ld CF, (src*).b	1
	Load the value of a memory bit specified by the low-order 3 bits of register A into the Carry flag	ld CF, (src*).b	1
Store Carry flag	Store CF flag into the value of bit b of an 8-bit register	ld g.b,CF	1

Operation	Description	Assembler	Operation
	Store CF flag into the value of bit b in a memory location	ld (src*).b,CF	1
	Store CF flag into the value of a memory bit specified by the low-order 3 bits of register A	ld (src*).b,A	1
Exclusive-OR Carry flag operation	Exclusive-OR the value of bit b of an 8-bit register with the Carry flag and place the result in the Carry flag	xor CF, g.b	1
	Exclusive-OR the value of bit b in a memory location with the Carry flag and place the result in the Carry flag	xor CF, (src*).b	1
	Exclusive-OR the value of a memory bit specified by the low-order 3 bits of register A with the Carry flag and place the result in the Carry flag	xor CF, (src*).b	1
Set Carry flag	Set the Carry flag	set cf	1
Clear Carry flag	Clear the Carry flag	clr cf	1
Complement Carry flag	Complement the Carry flag	cpl cf	1
Note： src: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A			

TABLE 4- 8 BIT AND FLAG MANIPULATION INSTRUCTIONS

4.5.5 Jump Instructions

Operation	Description	Assembler	Operation
Conditional jump	Short relative jump with true jump flag	jrs T,$+2+d	1
	Short relative jump with false jump flag	jrs F,$+2+d	1
	Relative jump with true jump flag	jr T,$+2+d	1
	Relative jump with false jump flag	jr F,$+2+d	1
	Relative jump with true Zero flag	jr EQ,$+2+d	1
	Relative jump with false Zero flag	jr NE,$+2+d	1
	Relative jump with true Carry flag	jr LT,$+2+d	1
	Relative jump with false Carry flag	jr GE,$+2+d	1
	Relative jump with true Carry and Zero flags	jr LE,$+2+d	1
	Relative jump with false Carry and Zero flags	jr GT,$+2+d	1
	Relative jump with true Sign flag	jr M,$+3+d	1
	Relative jump with false sign flag	jr P,$+3+d	1
	Relative jump with true result of an exclusive-OR operation of Sign and Overflow flags	jr SLT,$+3+d	1
	Relative jump with false result of an exclusive-OR operation of Sign and Overflow flags.	jr SGE,$+3+d	1
	Relative jump with true Zero flag and true result of an exclusive-OR operation of Sign and Overflow flags	jr SLE,$+3+d	1
	Relative jump with false Zero flag and false result of an exclusive-OR operation of Sign and Overflow flags	jr SGT,$+3+d	1
	Relative jump with true Overflow flag	jr VS,$+3+d	1
	Relative jump with false Overflow flag	jr VC,$+3+d	1
Unconditional Jump	Jump with immediate addressing	JP mn	1
	Jump with register addressing	JP gg	1
	Jump with direct addressing mode or register indirect addressing mode	JP (src*)	3
Note：src: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A			

TABLE 4- 9 JUMP INSTRUCTIONS

4.5.6 Call, Return, Software Interrupt and No Operation

Operation	Description	Assembler	Operation
Subroutine call	Vectored subroutine call	callv n	1
	Absolute subroutine call	call mn	1
	Subroutine call with register addressing mode	call gg	1
	Subroutine call with direct addressing mode or register indirect addressing mode	call (src*)	3
Return	Return from a subroutine	ret	3
	Return from a maskable interrupt service routine	reti	3
	Return from a non-maskable interrupt service routine	retn	3
Software interrupt	Software interrupt instruction	swi	4
NOP	No operation	nop	1
Note： src: x, vw, DE, HL, IX, IY, IX+d, IY+d, SP+d, HL+d, HL+C, +SP, PC+A			

TABLE 4- 10 CALL, RETURN, SOFTWARE INTERRUPT AND NO OPERATION

5. Addressing Space

The address space is divided into program and data spaces. The code and data accesses can be byte access or word access. The addressable memory space is 64kB of program and 64kB of data memory.

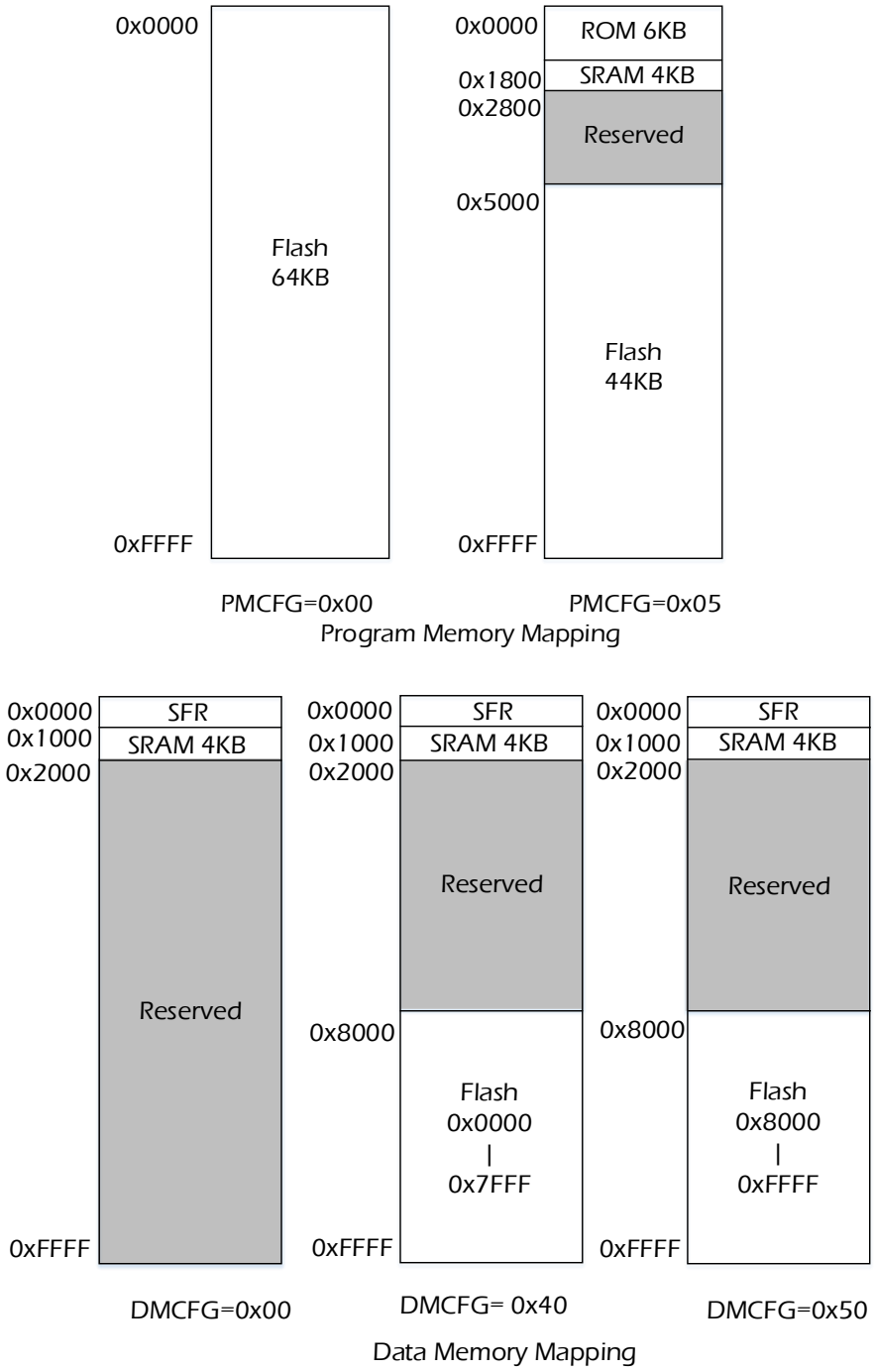


FIGURE 5-1 PROGRAM AND DATA MEMORY MAPPING

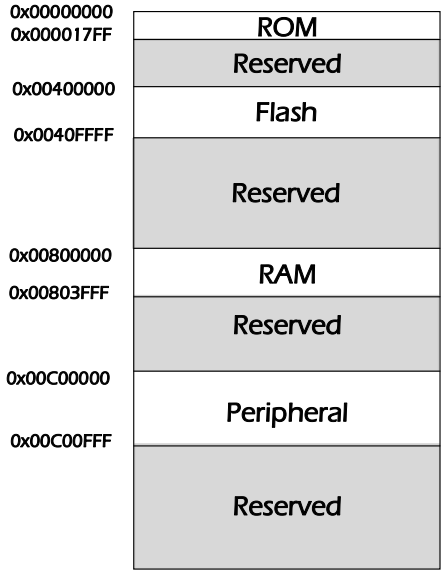


FIGURE 5-2 DATA SPACE

The data space is divided into platform, peripheral, and data memory areas. The following sections describe these areas in that order.

5.1 Plateform Memory Organization

This region has a total of 64 bytes of memory-mapped registers. The registers are divided into three functional groups within this region：

- System Configuration Registers
- System Control Registers
- Platform Peripherals and Control Registers

Function	Address	Register	Description
System Configuration Registers	0x0008	SYSCR0	System Control Register 0
	0x0009	Reserved	
	0x000A	PMR	Power Mode Register
	0x000B	RSTFLG	Reset Flag Register
	0x000C | 0x000F	Reserved	
System Control Registers	0x0010 | 0x001F	Reserved	
	0x0020	CLKCR0	Clock Control Register 0
	0x0021	CLKCR1	Clock Control Register 1
	0x0022	Reserved	
	0x0023	CLKCR3	Clock Control Register 3
	0x0024	PLLCR0	PLL Control Register 0
	0x0025	Reserved	
	0x0026		
	0x0027	FCKDIV	Flash Clock Divider Register
	0x0028	WDCTR	Watchdog Control Register
	0x0029	WDCDR	Watchdog Control Data Register
	0x002A	WDCNT	Watchdog Count Register
	0x002B	WDST	Watchdog Status Register
	0x002C	Reserved	
	0x002D		
	0x002E	TBTCR	Time-Based Timer Control Register
	0x002F	DVOCR	Divider Output Control Register
	0x0030	CMSR	Clock Monitor Status Register
	0x0031	LVDCR	Low Voltage Detection Control Register

Function	Address	Register	Description
	0x0032	Reserved	
	0x0033		
	0x0034	PONCR	Power ON Control Register
	0x0035	CMCR	Clock Monitor Control Register
	0x0036 | 0x0037	Reserved	
	0x0038	PMCFG	Program Memory Configuration Register
	0x0039	DMCFG	Data Memory Configuration Register
	0x003A	MIFR	Master Interrupt Enable Register
	0x003B | 0x003E	Reserved	
	0x003F	PSW	Program Status Word Register

TABLE 5- 1 PLATEFORM MEMORY ORGANIZATION

5.1.1 System Control Register

System control information is stored in this area.

Address	Register	Description
0x0008	SYSCR0	System Control Register 0
0x000A	PMR	Power Mode Register
0X000B	RSTFLG	Reset Flag Register

System Control Register 0(SYSCR0)

SYSCR0	7	6	5	4	3	2	1	0
Bit Symbol	reserved		reserved	reserved	XRSTDIS	OCDDIS	ROMST	reserved
Read/Write	-		-	-	R/W	R/W	R	-
After reset	0		0	0	0	0	0	0

Note 1：Bit 0 is reset by POR only.

Note 2：Bit 7:1 are reset by all hardware and software resets.

Note 3：Reserved bits must be written with zeros for future compatibility.

XRSTDIS	External Reset Disable	0：External reset pin is in use 1：External reset pin is repurposed to other functions
OCDDIS	OCD Disable	0：OCD pins are available 1：OCD pins are repurposed to other functions
ROMST	ROM status bit	0：ROM passes CRC check 1：ROM fails CRC check

Power Mode Register (PMR)

PMR	7	6	5	4	3	2	1	0
Bit Symbol	reserved		reserved	LDOON	DSM	PMODE[2:0]		
Read/Write	-		-	R/W	R/W	R/W	R/W	R/W
After reset	0		0	0	0	0	0	0

Note1 : Bits 2:0 are reset by all reset and valid wakeup sources.

LDOON	LDO ON	0：Turn off LDO in Deep Sleep Mod 1：Leave LDO ON in Deep Sleep Mode
DSM	Deep Sleep Mode bit	0：SLEEP mode using SLEEP instruction 1：Entering Deep Sleep mode with SLEEP instruction
PMODE [2:0]	Power Mode	000：Active Mode Others：System reserved

Reset Flag Register (RSTFLG)

RSTFLG	7	6	5	4	3	2	1	0
Bit Symbol	CLR	Reserved			BLMP	WDTF	Reserved	EXBRORF
Read/Write	W	R	R	R	R	R	R	R
After reset	0	0	0	0	1	0	0	1

CLR	RSTFLG Clear	0：no effect 1：Write 1 to clear this register. This bit will be automatically cleared to 0 when done.
BLMF	Bootloader Reset Flag	0：Not bootloader reset 1：Reset caused by bootloader
WDTF	Watch dog Reset Flag	0：Not watchdog reset 1：Reset caused by watchdog
EXBRORF	External Reset or BROR Reset Flag	0：Not External Reset or BROR Reset 1：Reset caused by External Reset or BROR

The table followed describes the wake-up sources for sleep mode and deep sleep mode：

Operation mode	Wake up source
Sleep Mode	All interrupts and all resets
Deep Sleep mode	KWI interrupt, LVD and OCD sleep release command. If use KWI, and LVD to exit deep sleep mode, before entering deep sleep mode, set CLKCR1<HIRCEN>=1.

5.1.2 Platform Peripherals and Control Registers

The platform peripherals are：

- Clock controller and Monitor
- Watchdog timer (WDT)
- Time-based timer (TBT)
- Clock divider output (DVO)

The platform registers are：

- Program Memory Configuration Register (PMCFG)
- Data Memory Configuration Register (DMCFG)
- Master Interrupt Enable Register (MIFR)
- Program Status Word Register (PSW)

Address	Register	Description
0x0038	PMCFG	Program Memory Configuration Register
0x0039	DMCFG	Data Memory Configuration Register
0x003A	MIFR	Master Interrupt Enable Register
0x003F	PSW	Program Status Word Register

Program Memory Configuration Register (PMCFG)

PMCFG	7	6	5	4	3	2	1	0
Bit Symbol	PMCFG[7:0]							
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1：This register is reset by all hardware and software resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

PMCFG	Program Memory Configuration	PMCFG=0x00: RAM,ROM not mapped into program space PMCFG=0x05: RAM,ROM mapped into program space, RAM start mapping at 0x1000, ROM start mapping 0x0000

Data Memory Configuration Register (DMCFG)

DMCFG	7	6	5	4	3	2	1	0
Bit Symbol	DMCFG[7:0]							
Read/Write	R	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1：This register is reset by all hardware and software resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

DMCFG	Data Memory Configuration	DMCFG=0x00: Flash,ROM not mapped into data area DMCFG=0x40: Flash 0x0000-0x7FFF mapping to 0x8000-0xFFFF , ROM not mapped into data area DMCFG=0x50: Flash 0x8000-0xFFFF mapping to 0x8000-0xFFFF , ROM not mapped into data area

Master Interrupt Enable Register (MIFR)

MIFR	7	6	5	4	3	2	1	0
Bit Symbol	reserved							IMF
Read/Write	R							R/W
After reset	0							0

Note 1：This register is reset by all hardware and software resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

IMF	Interrupt Master Enable Flag	This bit is set by the Enable Interrupt Instruction (EI) and cleared by the Disable Interrupt Instruction (DI).

Program Status Word Register (PSW)

PSW	7	6	5	4	3	2	1	0
Bit Symbol	JF	ZF	CF	HF	SF	VF	RBS	-
Read/Write	R	R	R	R	R	R	R	-
After reset	0	0	0	0	0	0	0	*

Note 1：This register is reset by all hardware and software resets.

Note 2：Bit 0 is reserved for the IMF flag. It is a read-only register bit. The physical register bit resides in 0x003A memory-mapped register.

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".The condition code is described below.

Condition Code	Meaning	Condition
T	1	JF = 1
F	0	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 less than or equal to	CF = 0
LE	Unsigned less than or equal to	(CF ^ ZF) = 1
GT	Unsigned greater than	(CF ^ ZF) = 0
SLT	Signed less than	(SF ^ VF) = 1
SGE	Signed greater than or equal to	(SF ^ VF) = 0
SLE	Signed less than or equal to	ZF ^ (SF ^ VF) = 1
SGT	Signed greater than	ZF ^ (SF ^ VF) = 0

TABLE 5- 2 CONDITIONA CODE (CC) TABLE

5.2 Peripheral Memory

This memory region starts at 0x0030 and ends at 0x0FFF.

5.2.1 Peripheral Area 1

This area has a total of 976 bytes of memory-mapped registers. All 870C1 compatible peripherals and peripheral control registers are located in this area. The 870C1 APB Compatible Peripherals include the following functions.

Flash Controller

- 16-bit Timer (TCA), up to eight instances
- UART,-three instances
- I2C, two instances
- SIO two instances.
- Key-on wakeup,eight keys.
- External interrupt controller, eight interrupts.
- Internal interrupt controller, 28 interrupts.
- GPIO controller,29 GPIO pins

Address	Byte3	Byte2	Byte1	Byte0	Register
0x0040	FADDR1	FADDR0	FCR1	FCR0	Flash Controller
0x0044	Reserved	Reserved	FDATA1	FDATA0	
0x0048 | 0x0067	Reserved				
0x0068	TA3CR	TA2CR	TA1CR	TA0CR	16-bit timer
0x006C	TA7CR	TA6CR	TA5CR	TA4CR	
0x0070	TA3MOD	TA2MOD	TA1MOD	TA0MOD	
0x0074	TA7MOD	TA6MOD	TA5MOD	TA4MOD	
0x0078	TA3SR	TA2SR	TA1SR	TA0SR	
0x007C	TA7SR	TA6SR	TA5SR	TA4SR	
0x0080	TA0DRBH	TA0DRBL	TA0DRAH	TA0DRAL	
0x0084	TA1DRBH	TA1DRBL	TA1DRAH	TA1DRAL	
0x0088	TA2DRBH	TA2DRBL	TA2DRAH	TA2DRAL	
0x008C	TA3DRBH	TA3DRBL	TA3DRAH	TA3DRAL	
0x0090	TA4DRBH	TA4DRBL	TA4DRAH	TA4DRAL	
0x0094	TA5DRBH	TA5DRBL	TA5DRAH	TA5DRAL	
0x0098	TA6DRBH	TA6DRBL	TA6DRAH	TA6DRAL	
0x009C	TA7DRBH	TA7DRBL	TA7DRAH	TA7DRAL	
0x00A0	UART0SR	UART0DR	UART0CR2	UART0CR1	UART
0x00A4	UART1CR2	UART1CR1	TD0BUF	RD0BUF	
0x00A8	TD1BUF	RD1BUF	UART1SR	UART1DR	
0x00AC	UART2SR	UART2DR	UART2CR2	UART2CR1	
0x00B0	Reserved		TD2BUF	RD2BUF	
0x00B4	Reserved				
0x00B8	I2C0AR	SBI0SR	SBI0CR2	SBI0CR1	I2C
0x00BC	SBI1SR	SBI1CR2	SBI1CR1	SBI0DBR	
0x00C0	Reserved		SBI1DBR	I2C1AR	
0x00C4	Reserved				
0x00C8					
0x00CC	Reserved				
0x00D0	SIO0BUF	SIO0SR	SIO0CR2	SIO0CR1	SIO
0x00D4	SIO1BUF	SIO1SR	SIO1CR2	SIO1CR1	
0x00D8	Reserved				
0x00DC					
0x00E0	P3DO	P2DO	P1DO	P0DO	GPIO DO
0x00E4	Reserved		P5DO	P4DO	
0x00E8	Reserved				
0x00EC					
0x00F0	P3DI	P2DI	P1DI	P0DI	GPIO DI
0x00F4	Reserved		P5DI	P4DI	
0x00F8	Reserved				
0x00FC					
0x0100	P3OE	P2OE	P1OE	P0OE	GPIO OE
0x0104	Reserved		P5OE	P4OE	
0x0108	Reserved				
0x010C					
0x0110	P3PU	P2PU	P1PU	P0PU	GPIO PU
0x0114	Reserved		P5PU	P4PU	
0x0118	Reserved				
0x011C					
0x0120	P3PD	P2PD	P1PD	P0PD	GPIO PD

Address	Byte3	Byte2	Byte1	Byte0	Register
0x0124	Reserved		P5PD	P4PD	
0x0128	Reserved				
0x012C					
0x0130	Reserved				
0x0134					
0x0138					
0x013C					
0x0140	P3FC1	P2FC1	P1FC1	P0FC1	GPIO FC1
0x0144	Reserved		P5FC1	P4FC1	
0x0148	Reserved				
0x014C					
0x0150	P3FC2	P2FC2	P1FC2	P0FC2	GPIO FC2
0x0154	Reserved		P5FC2	P4FC2	
0x0158	Reserved				
0x015C					
0x0160	FSELR3	FSELR2	FSELR1	FSELR0	Function Select
0x0164	Reserved		FSELR5	FSELR4	
0x0168	Reserved				
0x016C	Reserved	PCSELR2	PCSELR1	PCSELR0	Peripheral Channel Select
0x0170	Reserved			PCSELR4	
0x0174	Reserved				
0x0178	PCKEN3	PCKEN2	PCKEN1	PCKEN0	Peripheral Clock Enable
0x017C	PCKEN7	PCKEN6	PCKEN5	PCKEN4	
0x0180	Reserved				Reserved
0x0184	PRSTR7	Reserved			Peripheral Reset
0x0188	Reserved		KWUCR1	KWUCR0	Key-on wakeup
0x018C	Reserved		KWUSR1	KWUSR0	
0x0190	EINTCR3	EINTCR2	EINTCR1	EINTCR0	External Interrupt
0x0194	EINTCR7	EINTCR6	EINTCR5	EINTCR4	
0x0198 | 0x019F	Reserved				
0x01A0	IFR3	IFR2	IFR1	IFR0	Internal Interrupt
0x01A4	IFR7	IFR6	IFR5	IFR4	
0x01A8	Reserved	IFR10	IFR9	IFR8	
0x01AC	Reserved				
0x01B0	IER3	IER2	IER1	IER0	
0x01B4	IER7	IER6	IER5	IER4	
0x01B8	Reserved	IER10	IER9	IER8	
0x01BC	Reserved				
0x01C0	IPR3	IPR2	IPR1	IPR0	
0x01C4	IPR7	IPR6	IPR5	IPR4	
0x01C8	IPR11	IPR10	IPR9	IPR8	
0x01CC	IPR15	IPR14	IPR13	IPR12	
0x01D0	IPR19	IPR18	IPR17	IPR16	
0x01D4	Reserved	Reserved	Reserved	IPR20	
0x01D8	Reserved				
0x01DC	Reserved				

Address	Byte3	Byte2	Byte1	Byte0	Register
0x01E0 | 0x02FF	Reserved				
0x0300	Reserved	ADCCR2	ADCCR1	ADCCR0	ADC
0x0304	ADCSCAN0	ADCLV	Reserved	ADCCKDIV	
0x0308	ADCCHRDY	ADCSR	Reserved	ADCSCAN1	
0x030C	Reserved			ADCCHSEL	
0x0310	ADCLLVH	ADCLLVL	ADCDRH	ADCDRL	
0x0314	Reserved		ADCHLVH	ADCHLVL	
0x0318 | 0x086F	Reserved				
0x0870	Reserved		MACCR1	MACCR0	Multiplier
0x0874	MACA3	MACA2	MACA1	MACA0	
0x0878	MACB3	MACB2	MACB1	MACB0	
0x087C	MACC3	MACC2	MACC1	MACC0	
0x0880	Reserved			MACC4	
0x0900 | 0x0FFF	Reserved				

TABLE5- 3 PERIPHERAL MEMORY

6. System Operation

This section describe the following functionality:
- Operating Mode
- Reset
- Key-on Wakeup (KWU)
- Interrupt
- System Power Monitor

6.1 Operating Modes

SQ7613 has three operating modes:
- Normal mode
- Sleep mode
- Deep Sleep mode

The normal mode is the normal operating condition. In low-power mode, the CPU may enter either the Sleep, or Deep Sleep mode. These two power-saving modes progressively reduce the power dissipation from mA of current to uA.

Table below summarizes the functions that are enabled/disable in different mode.

Mode	Normal	Sleep	Deep Sleep
CPU Clock	ON	OFF	OFF
Periph Clock	ON*	ON*	OFF*
LDO	ON	ON	OFF
BROR	ON*	ON*	ON*
LVD	OFF*	OFF*	OFF*
PLL	OFF*	OFF*	OFF
HXTAL	OFF*	OFF*	OFF
HIRC	ON*	ON*	OFF
LIRC	ON	ON	ON
Flash	ON	ON	OFF
RAM	ON	ON	Retention
Note	*：user can enable or disable by software setting. Retention: data retention		

TABLE 6- 1 SYSTEM OPERATION MODES

6.1.1 Normal Mode

In normal mode, the CPU can execute instructions at the maximum clock speed to satisfy application's data throughput requirements. It's possible, however, to conserve power in this mode by reducing the system clock frequency or switching to low-frequency system clock when high throughput is no longer required. In addition, peripheral clocks to peripherals that are not in use can be switched off.

6.1.2 Sleep Mode

This low-power mode allows power savings, while offers very fast response to interrupts. In this mode, the CPU clock is turned off, and the PLL remains locked and stays in running state. Depending on the performance requirements, the PLL and high frequency internal reference clock, may be disabled. Peripherals that are not in use may also be turned off.

- Entering：

This mode can be entered by executing the SLEEP instruction.

- Exiting：

Any interrupt source or reset , excluding WDT INT/WDT RST will wake up the CPU from this mode.

6.1.3 Deep Sleep Mode

In Deep Sleep mode, the CPU and all peripheral clocks are switched off. The PLL and high-frequency internal reference clock are disabled. The ROM and the Flash are powered down. The core voltage regulator is switched to retention mode. If use KWI,and LVD to exit deep sleep mode, before entering deep sleep mode, set CLKCR1<HIRCEN>=1.

- Entering：

This mode can be entered by executing the SLEEP instruction.

- Exiting：

In this mode, Key-On Wakeup pins can wake up the CPU. If the System Power Monitor is enabled, the LVD events will wake up the CPU instantly.The CPU can response to a wakeup event within a few micro seconds.

6.1.4 Low Power Mode

There are two way to enter a low-power mode：Power Mode register , and SLEEP instruction

Low power mode entry

During a low-power mode, the CPU clock may be switched off. Products that support retention mode, may also have the CPU powered down during this time.

Low power mode exiting

The system receives an interrupt event and restarts the CPU clock. For productions that support retention mode, the CPU power is restored at this time. The CPU, then, continues executing code where it was stopped prior to entering the low-power mode.

6.2 Reset Function

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

6.2.1 Configuration

1. External reset input (RESET, external factor)
2. Power-on reset (POR, internal factor)
3. Brown-out reset (BROR, internal factor)
4. Watchdog timer reset (WDT, internal factor)

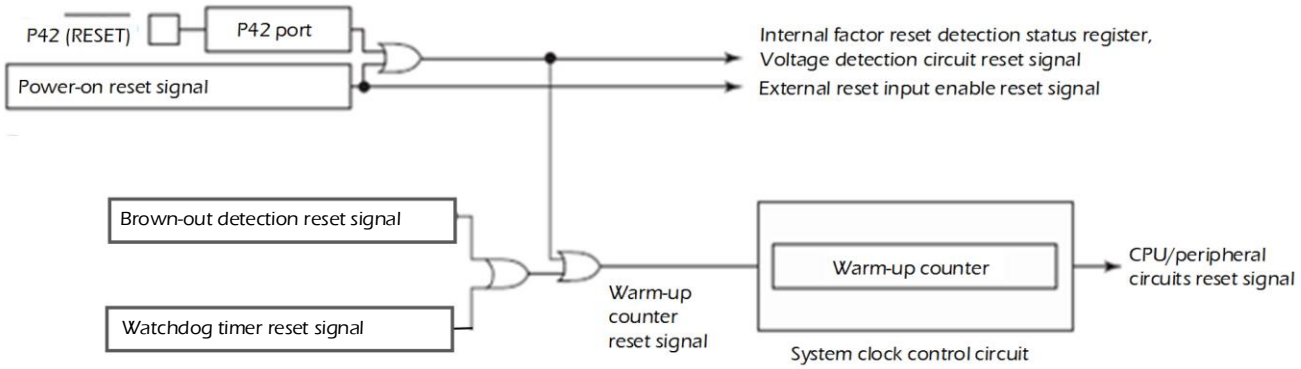


Figure 6- 1 Reset control circuit

6.2.2 Control

The reset control circuit is controlled by system control register0 (SYSCR0), and Reset Flag Register (RSTFLG).

Address	Register	Description
0x0008	SYSCR0	System Control Register 0
0X000B	RSTFLG	Reset Flag Register

System Control Register0 (SYSCR0)

SYSCR0	7	6	5	4	3	2	1	0
Bit Symbol	reserved		reserved	reserved	XRSTDIS	OCDDIS	ROMST	reserved
Read/Write	-		-	-	R/W	R/W	R	-
After reset	0		0	0	0	0	0	0

Note 1：Bit 0 is reset by POR only.

Note 2：Bit 7:1 are reset by all haredware and software resets .

Note 3：Reserved bits must be written with zeros for future compatibility.

XRSTDIS	External Reset Disable	0：External reset pin is in use 1： External reset pin is repurposed to other functions
OCDDIS	OCD Disable	0：OCD pins are available 1：OCD are repurposed to other functions
ROMST	ROM status bit	0：ROM passes CRC check 1：ROM fail CRC check

Reset Flag Register (RSTFLG)

RSTFLG	7	6	5	4	3	2	1	0
Bit Symbol	CLR	Reserved			BLMP	WDTF	Reserved	EXBRORF
Read/Write	W	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

CLR	Clear RSTFLG	0：no reset detect 1：detect reset (write 1 clear)
BLMF	Bootloader reset flag	0： bootloader reset 1： no bootloader reset
WDTF	Watch dog reset flag	0： watch dog reset 1： no watch dog reset
EXBRORF	RESET or BROR reset flag	0：RESET or BROR reset 1：No RESET or BROR reset

6.2.3 Function

During a system reset, all the core registers are reset to its reset value. The program counter (PC) is loaded with the reset interrupt vector. The CPU vectors to the reset handler based on the content of the interrupt vector.

System power monitor, RESET, WDT reset can cause reset. Exiting reset status, the device will initialize.

At power up, the power monitor generates a Power-On reset (POR) or Brown-Out Reset(BROR) to initialize the device. When a power failure is detected in one of the supply source, it generates a hardware reset to prevent improper chip operations.

The external reset input pin is a hardware reset. When the input pin is asserted, the device immediately goes through a reset cycle. RESET is low-active.

The watchdog timeout or other fault conditions detected will cause reset. The watchdog timeout is similar to the external reset. While, memory fault and the security fault reset is similar to external reset.

User can generate device reset throught below software setting :
PRSTR7 = 0x5A;
PRSTR7 = 0xA5;
PRSTR7 = 0xC3;
PRSTR7 = 0x3C;

The time is around 2us (@16 MHz) from the code progromming to CPU reset ; the time is 16us (@ 16MHz) from CPU reset to ready, excluding BOOTROM code execution.

6.2.4 Device Initialization

The followings happen during a reset:

-The core registers are reset to their reset value.
-The GPIO pins are reset to input high-Z state.

The table below summarizes the device initial conditions and system initialization among different resets.

Reset Source	CPU	GPIO	Peripheral SFR	Reset time(Note) (typ.,fsysclk=16MHz)
RESET(external reset input)	Yes	Yes	Yes	4 ms
BROR(Brown-out reset)	Yes	Yes	Yes	
Power-on reset	Yes	Yes	Yes	
WDT reset	Yes	Yes	Yes	145 us
Software reset	Yes	Yes	Yes	16 us

TABLE 6- 2 DEVICE INITIALIZATION

Note : The reset time does not include BOOTROM code execution; BOOTROM code execution time is around 50ms(typ.).

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)	0x1FFF	0x1FFF	0x1FFF
Program status word (PSW)	0x00	Indeterminate	Indeterminate
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
Interrupt enable Register(IER)	0	0	0
Interrupt flag Register (IFR)	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
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 SFR map	Refer to SFR map	Refer to SFR map

TABLE 6-3 INITIALIZATION OF BUILT-IN HARDWARE BY RESET OPERATION AND ITS STATUS AFTER RELEASE

Note 1: The voltage detection circuits are disabled by power-on reset only

Note 2："HiZ" indicates high-impedance

6.2.5 Reset Signal Generating Factors

Reset signals are generated by each factor as follows：

6.2.5.1 External Reset Input (RESET Pin Input)

This is an external reset that is generated by the RESET pin input. P42 is also used as the RESET pin, and it serves as the RESET pin after the power is turned on.

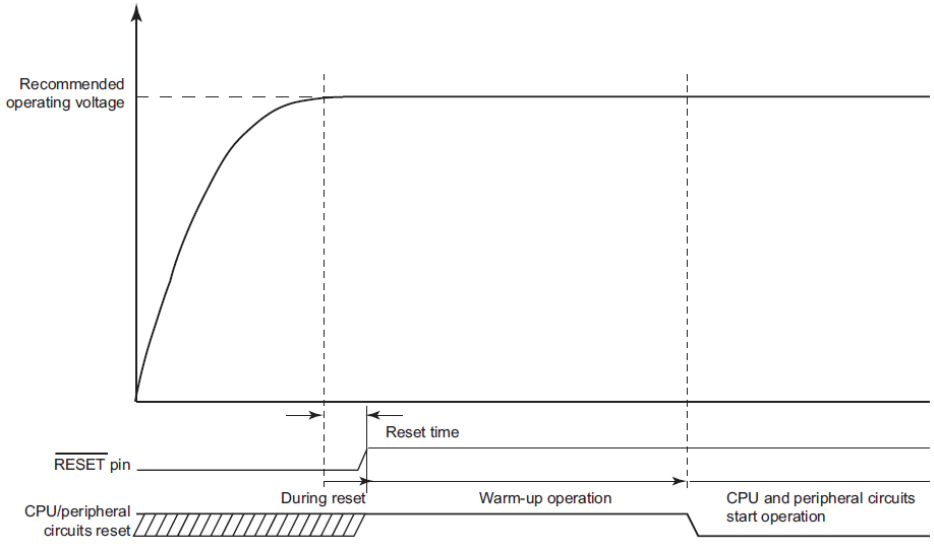


FIGURE6- 2 EXTERNAL RESET INPUT (DURING POWER-UP)

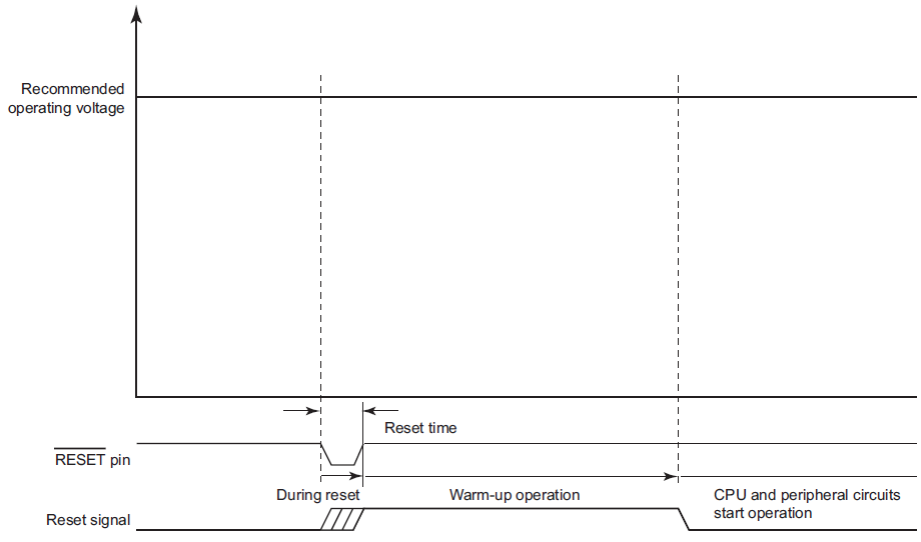


FIGURE 6- 3 EXTERNAL RESET INPUT (WHEN THE POWER IS STABILIZED)

During the power-on , and $\overline{\text{RESET}}$ pin set to "L" level, the system reset. The warm-up operation time is around 4ms.

When the supply voltage is within the recommended operating voltage range and the $\overline{\text{RESET}}$ pin is held "L" for 10us . In this case, the system will reset. The warm-up operation time is around 4ms.

In these two cases, after changing the RESET pin to "H" level, , the system will start the wake-up after releasing the reset.

- When the supply voltage rises rapidly：

During the power up,

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

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

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

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

Note ：When Supply voltage is equal to or lower than the detection voltage of the power-on reset level, even if the RESET pin is "H", the system would not exit the power-on reset.

6.2.5.2 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. Please refer to "6.3 Power-on Reset Circuit" for instructions.

6.2.5.3 Brown-out Reset

The Brown-out reset is an internal factor reset that occurs when detect the VDD level lower than the BROR trigger level (VBROR). Please refer to "6.4 Brown-out Reset " for instructions.

6.2.5.4 Watchdog Timer Reset

The watchdog timer reset is an internal factor reset that occurs when an overflow of the watchdog timer is detected. Please refer to "13.1 Watchdog Timer" for instructions.

6.2.5.5 How to use P42 as an External Reset

To use P42 as an external reset, keep P42 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 P4OE2 to "0", and connect a pull-up resistor to P42. Then clear SYSCR0<XRSTDIS> to "0". This enables the external reset function and makes P42 as a reset input pin.

To use P42 as an IO pin when it is used as a reset, set SYSCR0<XRSTDIS> to "1".

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 SYSCR0<XRSTDIS> 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.

6.3 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.

6.3.1 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.

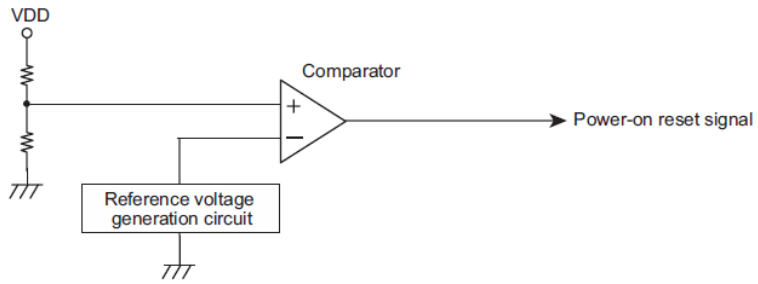


Figure 6- 4 Power-on Reset circuit

6.3.2 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.

Note：The detail of power-on characteristics please refer to "Chapter 3.4"

6.4 Brown-out Reset (BROR)

6.4.1 Configuration

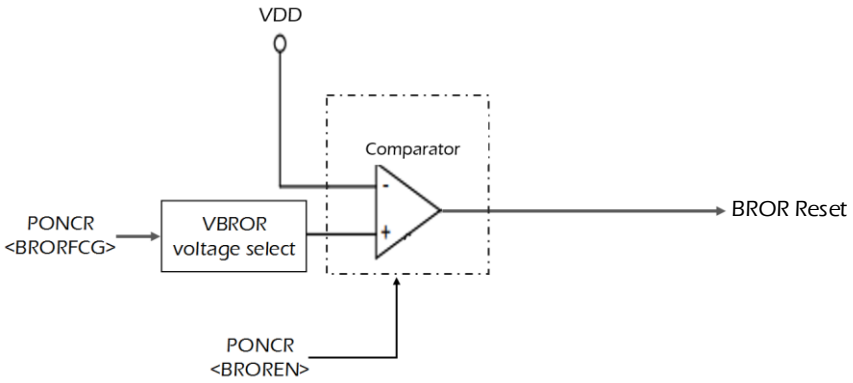


Figure 6-5 Brown-out reset circuit

6.4.2 Function

The Brown-out reset is used to monitor the VDD level during system operation. When VDD falls to the selected BROR detect level (VBROR) and PONCR<BROREN> is "1", the CPU will BROR-out reset. After a brownout reset, RSTFLG<EXBRORF> will clear to "0" automatically. Except RESET and BROR function, RSTFLG<EXBRORF> would not be"0". RSTLLG<EXBRORF> can be set or cleared by software.

6.4.3 Control

Power-on Control Register (PONCR)

PONCR	7	6	5	4	3	2	1	0
Bit Symbol	reserved (Note1)	reserved	reserved (Note1)	reserved (Note1)	reserved	BRORCFG[1:0]		BROREN
Read/Write	R/W	R	R/W	R/W	R	R/W		R/W
After reset	1	0	1	0	0	0		1

Note 1：Bit 7 must be written with 1, Bit 5 must be written with 1, Bit 4 must be written with 0

Note2：Bits are reset by POR reset

Note3：Reserved bits (Bit 6, and Bit 3) must be written with zeros for future compatibility.

BRORCFG [1:0]	Brown-out reset configuration	00：1.9V +/-57mV(default) 01：2.25V +/-67.5mV 10：2.55V +/-76.5mV 11：2.75V +/-82.5mV
BROREN	Brown-out reset Enable	0：Disable 1：Enable

6.5 Voltage Detection Circuit

The voltage detection circuit detects any decrease in the supply voltage and generates INTLVD interrupt request signals.

6.5.1 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 (VLVDx), and compared to the reference voltage in the comparator. When the supply voltage (VDD) becomes lower than the detection voltage (VLVDx), a voltage detection interrupt request signal is generated.

Whether to generate a voltage detection reset signal or an INTLVD interrupt request signal can be programmed by software.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 (VLVDx). 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.

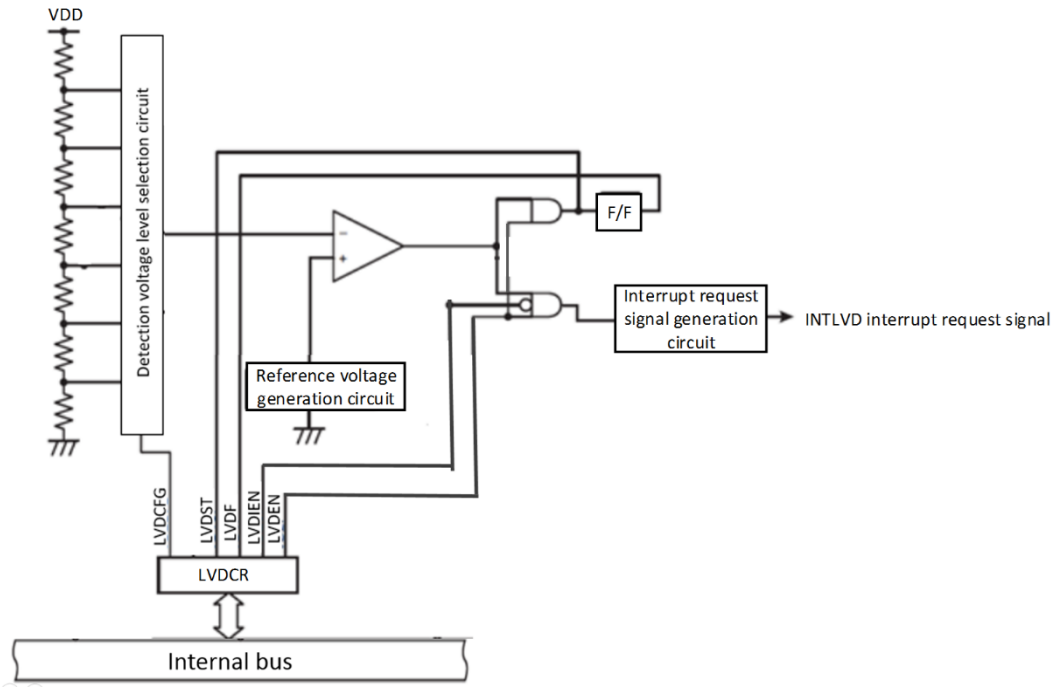


Figure 6-6 Voltage Detection Circuit

6.5.2 Control

The Voltage detection circuit is controlled by Low Voltage Control Register(LVDCR).

Address	Register	Description
0x0031	LVDCR	Low Voltage Control Register

Low Voltage Control Register (LVDCR)

LVDCR	7	6	5	4	3	2	1	0
Bit Symbol	-	LVDCFG [2:0]			LVDST	LVDF	LVDIEN	LVDEN
Read/Write	R/W	R/W			R	R/W1C	R/W	R/W
After reset	0	1	1	1	*	0	0	0

Note1： Bits are reset by all reset

Note 2：Bit7 must be written by 0.When LVDST is asserted, Bit 7 is cleared.

Note3：Reserved bits must be written with zeros for future compatibility.

LVDCFG [2:0]	LVD Configuration	000：Reserved 001：2.35V +/-70.5 mV 010：2.65V +/-79.5 mV 011：2.85V +/-85.5mV 100：3.15V +/-94.5 mV 101：3.98V +/-119.4 mV 110：4.2V +/-126 mV 111：4.5V +/-135 mV (Default)
LVDST	LVD Status , when interrupt is generated.	0：No LVD 1：LVD detected
LVDF	LVD Flag	0：No LVD 1：LVD detected
LVDIEN	LVD Interrupt Enable	0：Disable 1：Enable
LVDEN	LVD Enable	0：Disable 1：Enable

6.5.3 Function

6.5.3.1 Enabling / Disabling the Voltage Detection Operation

Setting LVDCR<LVDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

Note：When the supply voltage (VDD) is lower than the detection voltage(VLVDx), setting LVDCR <LVDEN> generates an INTLVD interrupt request signal at the time.

6.5.3.2 Selecting the Voltage Detection Operation Mode

When LVDCR<LVDIEN> is set to "1", the voltage detection operation mode is set to generate INTLVD interrupt request signals. When LVDCR< LVDIEN > is set to "0", the operation mode is not set to generate voltage interrupt request signals.

(a) When the operation mode is set to generate INTLVD interrupt signals (LVDCR<LVDIEN>= "1")

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

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

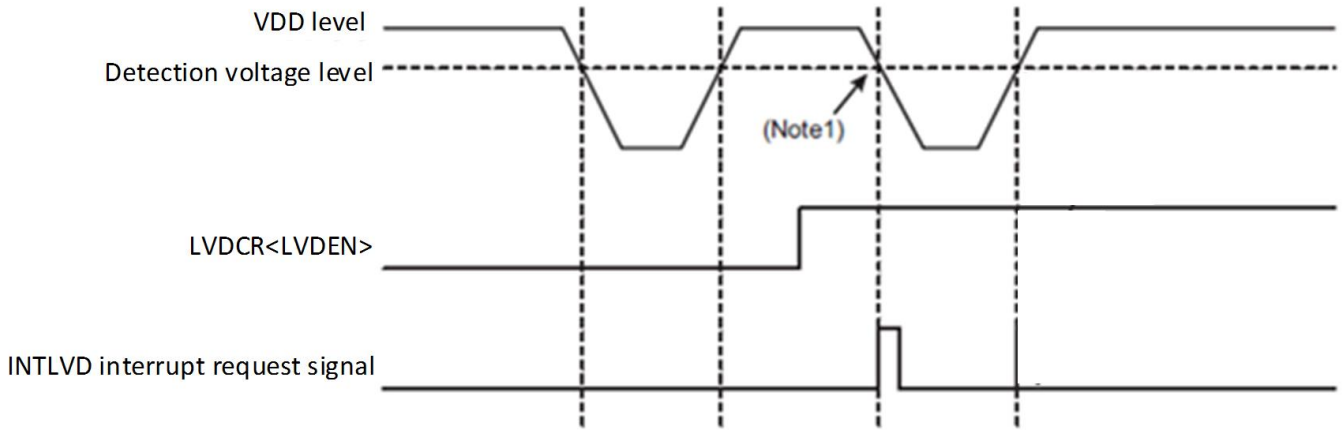


FIGURE 6-7 VOLTAGE DETECTION INTERRUPT REQUEST

6.5.3.3 Selecting the Detection Voltage Level

Select a detection voltage atLVDCR<LVDCFG>.

6.5.3.4 Voltage Detection Flag and Voltage Detection Status Flag

The magnitude relation between the supply voltage (VDD) and the detection voltage (VLVDx) can be checked by reading LVDCR<LVDCFG>.

If LVDCR<LVDEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VLVDx), LVDCR<LVDF> is set to "1" and is held in this state. LVDCR<LVDF> is not cleared to "0" when the supply voltage (VDD) becomes equal to or higher than the detection voltage (VLVDx).

When LVDCR<LVDF> is set at "1", after LVDCR<LVDEN> clear to "0", LVDCR<LVDF>the previous state is still held. To clear LVDCR<LVDF>, "0" must be written to it.

If LVDCR<LVDEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VLVDx), LVDCR<LVDST> is set to "1". When the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDxLVL), LVDCR<LVDST> is cleared to "0".

Unlike LVDCR<LVDF>, LVDCR<LVDF> does not hold the set state.

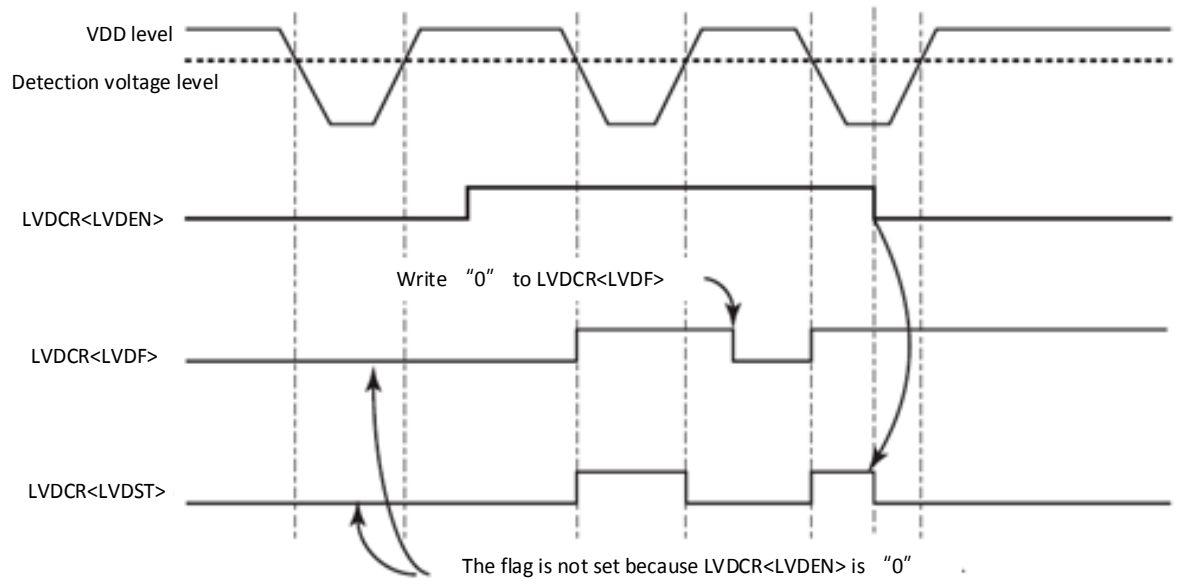


FIGURE 6-8 CHANGES IN THE VOLTAGE DETECTION FLAG AND THE VOLTAGE DETECTION STATUS FLAG

6.5.4 Register Setting

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

1. Clear the voltage detection circuit interrupt enable Register (IER) to "0". ◦
2. Set the detection voltage at LVDCR<LVDCFG>
3. Set LVDCR<LVDIEN> to"1", to set the operation mode to generate INTLVD interrupt request signals.
4. Set LVDCR<LVDEN> to "1" to enable the voltage detection operation.
5. Wait for 10μs or more until the voltage detection circuit becomes stable.
6. Make sure that LVDCR<LVDST> is "0".
7. Clear the voltage detection circuit interrupt flag register (IFR)to "0" and set the interrupt enable flag(IER) to "1" to enable interrupts.

Note：When the supply voltage (VDD) is close to the detection voltage (VLVDx), 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 flag register 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 Register (IER) to "0".
2. Clear LVDCR<LVDEN> to "0" to disable the voltage detection operation

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

6.6 Key-on Wakeup (KWU)

The key-on wakeup is a function for releasing the Deep sleep mode at pins KWI7 through KWI0.

6.6.1 Configuration

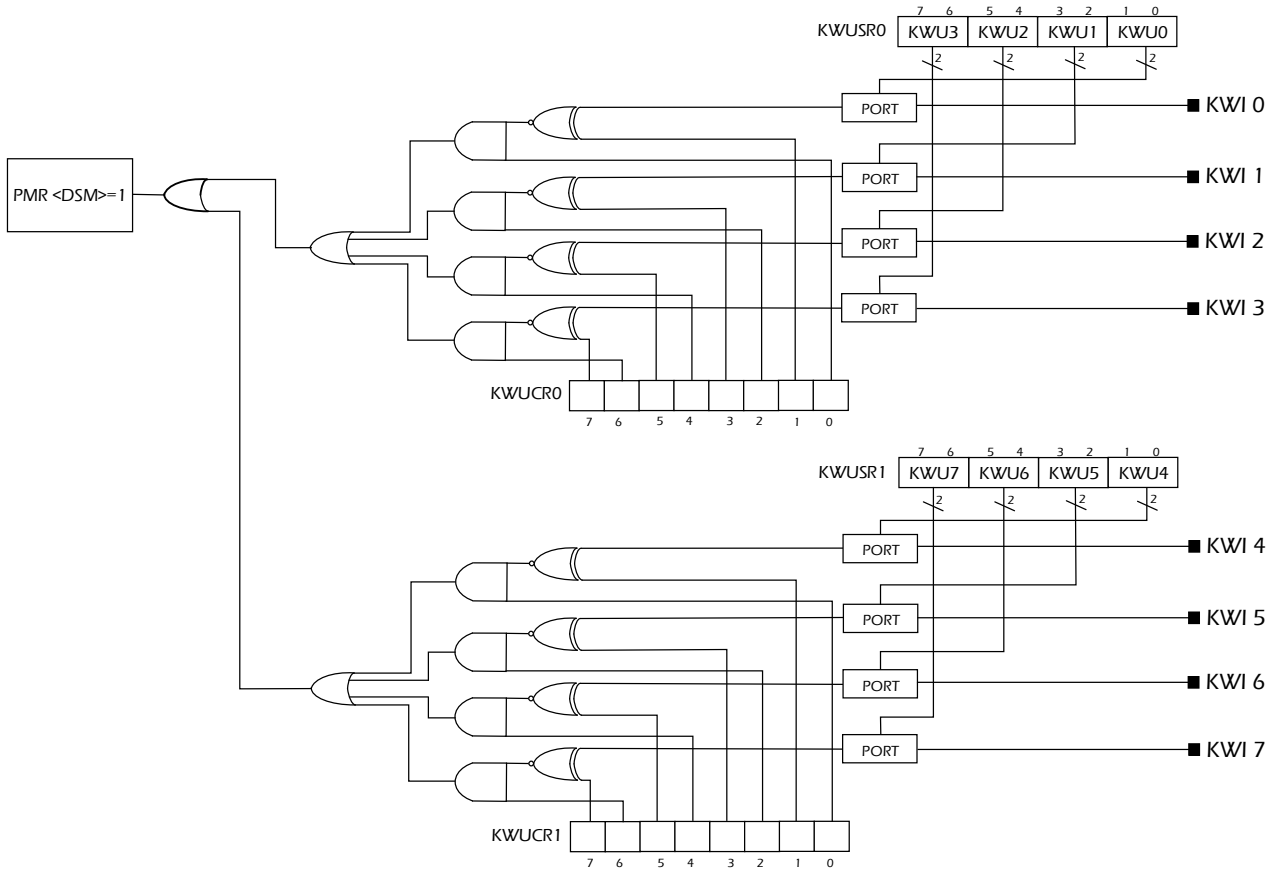


FIGURE 6- 9 KEY-ON WAKEUP CIRCUIT(EXAMPLE:KWI0~KWI7)

6.6.2 Control

Key-on wakeup control registers (KWUCR0 and KWUCR1) can be configured to designate the key-on wakeup pins (KWI7 through KWI0) as Deep Sleep mode release pins and to specify the Deep Sleep mode release levels of each of these designated pins.

Address	Register	Description
0x0188	KWUCR0	KWU Control Register0
0x0189	KWUCR1	KWU Control Register1
0x018C	KWUSR0	KWU Status Register 0
0x018D	KWUSR1	KWU Status Register 1

KWU Control Register 0(KWUCR0)

KWUCR0	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

Note：This register is reset by all hardware and software resets.

KW3LE	Deep sleep mode release level of KWI 3 pin	0: Low level 1: High level
KW3EN	Input enable / disable control of KWI 3 pin	0: Disable 1: Enable
KW2LE	Deep sleep mode release level of KWI 2 pin	0: Low level 1: High level
KW2EN	Input enable / disable control of KWI 2	0: Disable 1: Enable
KW1LE	Deep sleep mode release level of KWI 1 pin	0: Low level 1: High level
KW1EN	Input enable / disable control of KWI 1	0: Disable 1: Enable
KW0LE	Deep sleep mode release level of KWI 0 pin	0: Low level 1: High level
KW0EN	Input enable / disable control of KWI 0 pin	0: Disable 1: Enable

KWU Control Register 1(KWUCR1)

KWUCR1	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

Note：This register is reset by all hardware and software resets.

KW7LE	Deep sleep mode release level of KWI 7 pin	0: Low level 1: High level
KW7EN	Input enable / disable control of KWI 7 pin	0: Disable 1: Enable
KW6LE	Deep sleep mode release level of KWI 6 pin	0: Low level 1: High level
KW6EN	Input enable / disable control of KWI 6	0: Disable 1: Enable
KW5LE	Deep sleep mode release level of KWI 5 pin	0: Low level 1: High level
KW5EN	Input enable / disable control of KWI 5	0: Disable 1: Enable
KW4LE	Deep sleep mode release level of KWI 4 pin	0: Low level 1: High level
KW4EN	Input enable / disable control of KWI 4 pin	0: Disable 1: Enable

The port is multifunction. Key-on wakeup status registers (KWUSR0 and KWUSR1) can be configured to designate pin.

KWU Status Register 0(KWUSR0)

KWUSR0	7	6	5	4	3	2	1	0
Bit Symbol	KWU3		KWU2		KWU1		KWU0	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0		0		0		0	

Note：This register is reset by all hardware and software resets.

KWU KWUx	KWU0	KWU1	KWU2	KWU3
00	P0.0	P0.1	P0.2	P3.6
01	P1.0	P1.1	P1.2	P1.3
10	P2.0	P2.1	P2.2	-
11	-	P3.1	P3.2	P3.3

KWU Status Register 1(KWUSR1)

KWUSR0	7	6	5	4	3	2	1	0
Bit Symbol	KWU7		KWU6		KWU5		KWU4	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0		0		0		0	

Note：This register is reset by all hardware and software resets.

KWU KWUx	KWU4	KWU5	KWU6	KWU7
00	P0.4	P0.5	P0.6	P3.7
01	-	-	-	-
10	P2.4	-	-	P4.7
11	P3.4	P3.5	-	-

6.6.3 Function

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

6.6.3.1 Setting KWUSR

To designate a key-on wakeup pin (KWIm) as a deep sleep mode release pin, set KWUSRn. Port is multifunctional, user can configure to designate function. For example, user like to designate KWI0 to P1.0 as a deep sleep mode release pin. Then set KWUSR0 <KWU0> =01 to designate KWI0 to P1.0.

6.6.3.2 Setting KWUCR

To designate a key-on wakeup pin (KWIm) as a deep sleep mode release pin, set KWUCRn <KWmEN> to "1". After KWIm pin is set to "1" at KWUCRn <KWmEN>, a specific deep sleep mode release level can be specified for this pin at KWUCRn <KWmLE>. If KWUCRn <KWmLE> is set to "0", deep sleep mode is released when an input is at a low level.If it is set to "1", deep sleep mode is released when an input is at a high level. For example, if you want to release deep sleep mode by inputting a high-level signal into a KWI0 pin, set KWUCR0 <KW0EN> to "1", and KWUCR0 <KW0LE> to "1".

6.6.3.3 Starting Deep Sleep mode.

To start the deep sleep mode, set PWR<DSM>to "1", and use SLEEP instruction to starting deep sleep mode (more detail description please refer to chapater SYSCR1 <STOP> to "6.1.3 deep sleep mode".

6.6.3.4 Releasing Deep Sleep mode

To release deep sleep 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 deep sleep mode starts, the following instruction will be executed without starting the deep sleep 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.

6.7 Interrupt

SQ7613 supports up to 37 interrupts. There are two interrupt types, non-maskable and maskable interrupts. The non-maskable interrupts have higher priority than the maskable ones. In addition, all maskable interrupts can be nested with priorities.

The priority among each of the two types are discussed in the following sections.

6.7.1 Non-Maskable Interrupts

There are four non-maskable interrupts.

- Reset, 1st priority
- Software Interrupt, 2nd priority
- Undefined instruction interrupt, 2nd priority
- Watchdog interrupt, 3rd priority

The reset has the highest priority. The software interrupt and undefined instruction interrupt are mutually exclusive and have equal priority. The watchdog interrupt has the lowest priority.

6.7.2 Maskable Interrupts

SQ7613 supports maskable interrupts, interrupt 4 to interrupt 83.

The Natural Interrupt Priority (NIP) among them is in descending order, with interrupt 4 highest and interrupt 83 lowest.

6.7.3 Interrupt Table

Interrupt table is shown below:

Interrupt Source	Interrupt Name	Natural Interrupt Priority	Interrupt Vector	IER	IFR	IPR
Power-On Reset (Non-Maskable)	RESET_IRQ	1	0xFFFE	-	-	-
Software Interrupt (Non-Maskable)	SWI_IRQ	2	0xFFFC	-	-	-
Undefined Instruction (Non-Maskable)	UNDEF_IRQ	2	0xFFFC	-	-	-
Watchdog Timer (Non-Maskable)	WDT_IRQ	3	0xFFF8	-	IFR0.3	-
Low Voltage Detection	LVD_IRQ	4	0xFFF6	IER0.4	IFR0.4	IPR1[1:0]
Clock Fail Detection	CFD_IRQ	5	0xFFF4	IER0.5	IFR0.5	IPR1[3:2]
Reserved		6	0xFFF2	IER0.6	IFR0.6	IPR1[5:4]
Time-Based Timer	TBT_IRQ	7	0xFFF0	IER0.7	IFR0.7	IPR1[7:6]
Reserved		8	0xFFEE	IER1.0	IFR1.0	IPR2[1:0]
Reserved		9	0xFFEC	IER1.1	IFR1.1	IPR2[3:2]
Reserved		10	0xFFEA	IER1.2	IFR1.2	IPR2[5:4]
TCA0 16-bit Timer	TCA0_IRQ	11	0xFFE8	IER1.3	IFR1.3	IPR2[7:6]
TCA1 16-bit Timer	TCA1_IRQ	12	0xFFE6	IER1.4	IFR1.4	IPR3[1:0]
Reserved		13	0xFFE4	IER1.5	IFR1.5	IPR3[3:2]
Reserved		14	0xFFE2	IER1.6	IFR1.6	IPR3[5:4]
Reserved		15	0xFFE0	IER1.7	IFR1.7	IPR3[7:6]
UART0 RX	UART0_RX_IRQ	16	0xFFDE	IER2.0	IFR2.0	IPR4[1:0]
UART0 TX	UART0_TX_IRQ	17	0xFFDC	IER2.1	IFR2.1	IPR4[3:2]
I2C0	I2C0_IRQ	18	0xFFDA	IER2.2	IFR2.2	IPR4[5:4]
SIO0	SIO0_IRQ	19	0xFFD8	IER2.3	IFR2.3	IPR4[7:6]
External Interrupt 0	EXT0_IRQ	20	0xFFD6	IER2.4	IFR2.4	IPR5[1:0]
External Interrupt 1	EXT1_IRQ	21	0xFFD4	IER2.5	IFR2.5	IPR5[3:2]
External Interrupt 2	EXT2_IRQ	22	0xFFD2	IER2.6	IFR2.6	IPR5[5:4]
External Interrupt 3	EXT3_IRQ	23	0xFFD0	IER2.7	IFR2.7	IPR5[7:6]
ADC	ADC_IRQ	24	0xFFCE	IER3.0	IFR3.0	IPR6[1:0]
Reserved		25	0xFFCC	IER3.1	IFR3.1	IPR6[3:2]
Reserved		26	0xFFCA	IER3.2	IFR3.2	IPR6[5:4]

Interrupt Source	Interrupt Name	Natural Interrupt Priority	Interrupt Vector	IER	IFR	IPR
Reserved		27	0xFFC8	IER3.3	IFR3.3	IPR6[7:6]
Reserved		28	0xFFC6	IER3.4	IFR3.4	IPR7[1:0]
Flash Controller	FMC_IRQ	29	0xFFC4	IER3.5	IFR3.5	IPR7[3:2]
Reserved		30 | 38	0xFFC2 | 0xFFB2	IER3.6	IFR3.6	IPR7[5:4]
DIC	DIC_IRQ	39	0xFFB0	IER4.7	IFR4.7	IPR9[7:6]
Reserved		40	0xFFAE	IER5.0	IFR5.0	IPR10[1:0]
Reserved		41	0xFFAC	IER5.1	IFR5.1	IPR10[3:2]
Reserved		42	0xFFAA	IER5.2	IFR5.2	IPR10[5:4]
External Interrupt 4	EXT4_IRQ	43	0xFFA8	IER5.3	IFR5.3	IPR10[7:6]
External Interrupt 5	EXT5_IRQ	44	0xFFA6	IER5.4	IFR5.4	IPR11[1:0]
External Interrupt 6	EXT6_IRQ	45	0xFFA4	IER5.5	IFR5.5	IPR11[3:2]
External Interrupt 7	EXT7_IRQ	46	0xFFA2	IER5.6	IFR5.6	IPR11[5:4]
Multiplier	MAC_IRQ	47	0xFFA0	IER5.7	IFR5.7	IPR11[7:6]
Reserved		48	0xFF9E	IER6.0	IFR6.0	IPR12[1:0]
TCA2 Timer	TCA2_IRQ	49	0xFF9C	IER6.1	IFR6.1	IPR12[3:2]
TCA3 Timer	TCA3_IRQ	50	0xFF9A	IER6.2	IFR6.2	IPR12[5:4]
Reserved		51	0xFF98	IER6.3	IFR6.3	IPR12[7:6]
		52	0xFF96	IER6.4	IFR6.4	IPR13[1:0]
		53	0xFF94	IER6.5	IFR6.5	IPR13[3:2]
UART1 RX	UART1_RX1_IRQ	54	0xFF92	IER6.6	IFR6.6	IPR13[5:4]
UART1 TX	UART1_TX1_IRQ	55	0xFF90	IER6.7	IFR6.7	IPR13[7:6]
I2C1	I2C1_IRQ	56	0xFF8E	IER7.0	IFR7.0	IPR14[1:0]
SIO1	SIO1_IRQ	57	0xFF8C	IER7.1	IFR7.1	IPR14[3:2]
Reserved		58 | 66	0xFF8A | 0xFF7A	IER7.2	IFR7.2	IPR14[5:4]
TCA4 Timer	TCA4_IRQ	67	0xFF78	IER8.3	IFR8.3	IPR16[7:6]
TCA5 Timer	TCA5_IRQ	68	0xFF76	IER8.4	IFR8.4	IPR17[1:0]

Interrupt Source	Interrupt Name	Natural Interrupt Priority	Interrupt Vector	IER	IFR	IPR
Reserved		69	0xFF74	IER8.5	IFR8.5	IPR17[3:2]
Reserved		70	0xFF72	IER8.6	IFR8.6	IPR17[5:4]
Reserved		71	0xFF70	IER8.7	IFR8.7	IPR17[7:6]
UART2 RX	UART2_RX2_IRQ	72	0XFF6E	IER9.0	IFR9.0	IPR18[1:0]
UART2 TX	UART2_TX2_IRQ	73	0XFF6C	IER9.1	IFR9.1	IPR18[3:2]
Reserved		74 | 81	0XFF6A | 0XFF5C	IER9.2	IFR9.2	IPR18[5:4]
TCA6 Timer	TCA6_IRQ	82	0xFF5A	IER10.2	IFR10.2	IPR20[5:4]
TCA7 Timer	TCA7_IRQ	83	0xFF58	IER10.3	IFR10.3	IPR20[7:6]

TABLE 6- 3 INTERRUPT TABLE

6.7.4 Nested Vectore Interrupt Controller (INTC)

The interrupt controller supports up to 83 interrupts. The first four interrupt sources are non-maskable interrupts. They are reset, SWI, undefined instruction, and Watchdog interrupts. These interrupts have a fixed priority, as discussed in the System Interrupt section. Interrupts 4 to 83 are maskable interrupts.

The natural priority among the maskable interrupts is in descending order, with interrupt 4 highest and interrupt 83 lowest. In addition, each interrupt has a programmable priority register with four levels of priority. Level 0 has lowest priority and level 3 highest. Interrupt nesting with priorities is supported when the interrupt master enable (IMF) bit is set in the interrupt service routine.

Inside the controller, there are three sets of registers. The first set is the interrupt flags which holds the interrupt sources. The second set is the interrupt enables, which are used to enable the interrupts individually. The third set is the programmable interrupt priority registers.

When an interrupt occurs, its interrupt flag is set to a logic one. If the corresponding interrupt enable flag and the IMF flag is set, an interrupt request is generated and sent to the CPU for processing. If multiple maskable interrupts are generated simultaneously, the interrupts are serviced in the natural priority order. If the interrupt priority registers are programmed, the interrupt priority is determined based on the programmed interrupt level.

6.7.5 Interrupt Flag Register (IFRx , x=0~10)

An interrupt flag is provided for each maskable interrupt source. There are 11 registers.

When an interrupt generated by a peripheral, the flag is set to a logic one. The flag is cleared immediately after the interrupt is accepted by the CPU. All interrupt flags are initialized to zeros during a system reset. The flags can only be set by hardware. Writing a logic one has no effect. Writing a logic zero to a flag will clear it.

Register	Address
IFR0	0x01A0
IFR1	0x01A1
IFR2	0x01A2
IFR3	0x01A3
IFR4	0x01A4
IFR5	0x01A5
IFR6	0x01A6
IFR7	0x01A7
IFR8	0x01A8
IFR9	0x01A9
IFR10	0x01AA

Below is the description of IFR0 and IFR1, IFR2~ IFR10 please refert to Table 6- 4 Interrupt table.

IFR0	7	6	5	4	3	2	1	0
Bit Symbol	INT7	-	INT5	INT4	WDT	UNDEF	SWI	Reset
Interrupt Source	TBT	-	CFD	LVD	WDT	UNDEF	SWI	Reset
Read/Write	R/W	R/W	R/W	R/W	R/W	*	*	*
After reset	0	0	0	0	0	*	*	*

Note1 ： This register is reset by all hardware and software resets.

Note2：The register bits are set and cleared by hardware. Writing a logic one has no effect. Writing a logic zero clears a flag.

Note 3： * *These bits are not used*

IFR1	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT12	INT11	-	-	-
Interrupt Source	-	-	-	TCA1	TCA0	-	-	-
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

Note1 ：This register is reset by all hardware and software resets.

Note2：The register bits are set and cleared by hardware. Writing a logic one has no effect. Writing a logic zero clears a flag.

6.7.6 Interrupt Enable Register (IERx) ,x=0~10

The interrupt enable registers enable or disable individual maskable interrupt. The IMF flag is a master enable bit, which has a global effect on all maskable interrupts. Clearing the IMF flag disables all of them. Setting the IMF flag enables the interrupts that are specified by the individual interrupt enable flags in the IER registers. There are 11 IER registers, IER0 to IER10.

When an interrupt is serviced, the current IMF flag is pushed onto the stack along with the processor status flags. Upon entering the service routine, the IMF flag is cleared to zero to temporarily disable the subsequent maskable interrupts. After the interrupt service routine is executed, the current IMF flag is updated with the stacked IMF flag by the return interrupt instruction (RETI/RETN).

Note that, non-maskable interrupts are not affected by these registers.

Register	Address
IER0	0x01B0
IER1	0x01B1
IER2	0x01B2
IER3	0x01B3
IER4	0x01B4
IER5	0x01B5
IER6	0x01B6
IER7	0x01B7
IER8	0x01B8
IER9	0x01B9
IER10	0x01BA

Below is the description of IER0 and IER1, IER2~ IER 10 please refert to Table 6-5 Interrupt table.

IER0	7	6	5	4	3	2	1	0
Bit Symbol	IE7	IE6	IE5	IE4	*	*	*	*
Interrupt Source	TBT	-	CFD	LVD	*	*	*	*
Read/Write	R/W	R/W	R/W	R/W	*	*	*	*
After reset	0	0	0	0	*	*	*	*

Note1 ： This register is reset by all hardware and software resets.

Note 2： ＊ *These bits are not used*

IER1	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	IE12	IE11	-	-	-
Interrupt Source	-	-	-	TCA1	TCA0	-	-	-
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

Note1 ： This register is reset by all hardware and software resets.

6.7.7 Interrupt Processing

When an interrupt or multiple interrupts occur, the interrupt controller determines the interrupt to be serviced. An interrupt service request along with its interrupt vector is then sent to the CPU for processing. The interrupt latency, after a request is accepted, is 6 cycles.

E	E+1	E+2	E+3	F	D	E

interrupt latency	
Symbol	Stage description
E	This is the execution stage where the interrupt request is accepted. The interrupt vector is received in the Instruction Unit and the instruction fetch address is generated. The PSW is pushed onto stack in this cycle.
E+1	The content of the interrupt vector is returned and entered the instruction buffer as a jump instruction. The address of the next opcode is pushed onto stack in this cycle.
E+2	The address of the interrupt service routine is decoded.
E+3	Instruction Unit vectors to the interrupt service routine.
F	This is the fetch stage where the first opcodes of the interrupt service routine returned.
D	This is the decode stage where the instruction opcode is decoded.
E	This is the execution stage where the instruction is executed.

The RETI or RETN is the last instruction in an interrupt service routine. The address of the next opcode and the PSW are popped from the stack. The CPU then continues the code execution at the point it was interrupted.

Nested interrupts are supported when the interrupt master enable flag is set (IMF) in an interrupt service routine.

6.8 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.

6.8.1 Configuration

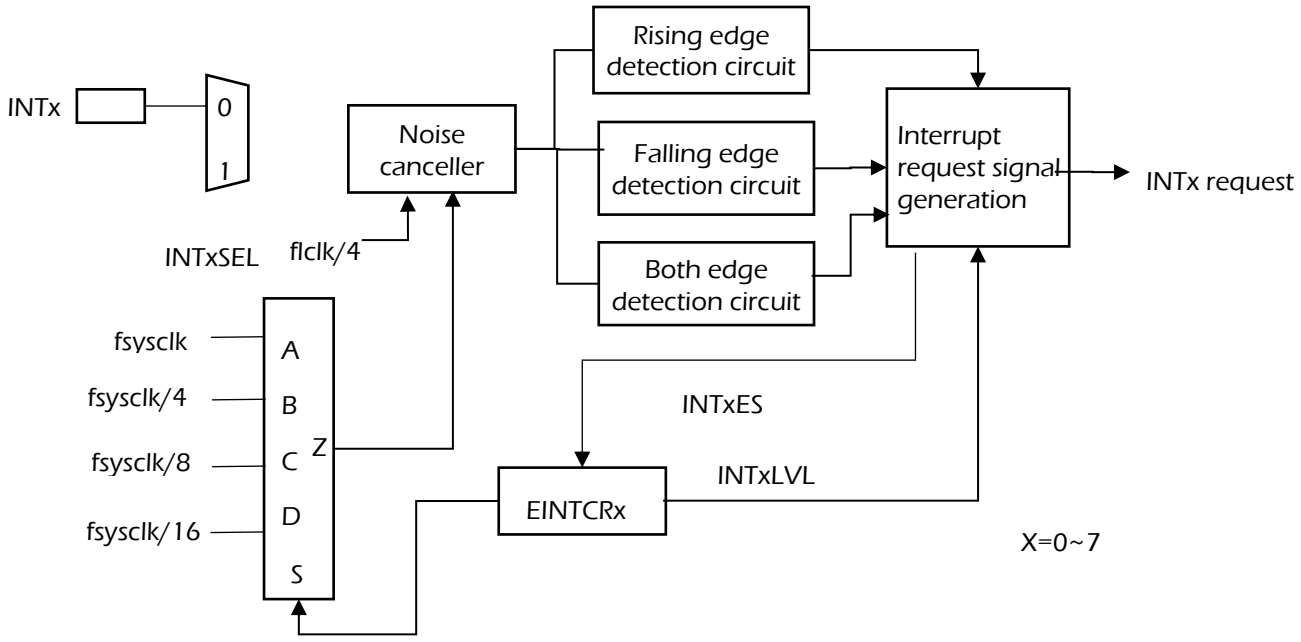


Figure 6-10 External Interrupts (INT0~INT7)

The external interrupt control circuit consists of a noise canceller, an edge 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.

6.8.2 Control

External interrupts are controlled by the following registers:

ADDRESS	REGISTER	DESCRIPTION
0x0190	EINTCR0	External interrupt control register 0
0x0191	EINTCR1	External interrupt control register 1
0x0192	EINTCR2	External interrupt control register 2
0x0193	EINTCR3	External interrupt control register 3
0x0194	EINTCR4	External interrupt control register 4
0x0195	EINTCR5	External interrupt control register 5
0x0196	EINTCR6	External interrupt control register 6
0x0197	EINTCR7	External interrupt control register 7

External Interrupt Control Register (EINTCRx), x=0 ~ 7

EINTCR	7	6	5	4	3	2	1	0
Bit Symbol	INTSEL[2:0]			INTLVL	INTES[1:0]		INTINC[1:0]	
Read/Write	R/W			R	R/W		R/W	
After reset	0			0	0		0	

Note：This register is reset by all hardware and software resets.

		EINTCRx [INTSEL]	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
			EINTCR7 [INTSEL]	EINTCR6 [INTSEL]	EINTCR5 [INTSEL]	EINTCR4 [INTSEL]	EINTCR3 [INTSEL]	EINTCR2 [INTSEL]	EINTCR1 [INTSEL]	EINTCR0 [INTSEL]
INTSEL[2:0]	Interrupt Pin selection	000	P3.7	P0.6	P0.5	P0.4	P3.6	P0.2	P0.1	P0.0
		001	-	-	-	-	P1.3	P1.2	P1.1	P1.0
		010	P4.7	-	-	P2.4	-	P2.2	P2.1	P2.0
		011	-	-	P3.5	P3.4	P3.3	P3.2	P3.1	-
		100	-	-	P4.5	P4.4	P4.3	P4.2	-	-

INTLVL	Signal level that passes noise canceller when the interrupt request signal is generated for external interrupt	0：Initial state or signal level "L" 1：Signal level "H"
INTES[1:0]	Select the interrupt request generating condition for external interrupt	00：An interrupt request is generated at the rising edge of the noise canceller pass signal 01：An interrupt request is generated at the falling edge of the noise canceller pass signal 10：An interrupt request is generated at both edges of the noise canceller pass signal 11：Reserved
INTINC[1:0]	Set the noise canceller sampling interval for external interrupt	00：fsysclk 01：fsysclk / 4 10：fsysclk / 8 11：fsysclk / 16

6.8.3 External Interrupt function

The condition for generating interrupt request signals and the noise cancel time are fixed for external interrupts 0 and 7.

Source	Pin	Enable Conditions	Interrupt request signal generated	External interrupt pin input signal width and noise removal	
				NORMAL/SLEEP MODE	NORMAL/SLEEP MODE (low-speed clock)

INT0	INT0	IMF=1 IER2.4=1	Falling edge Rising edge Both edges	Less than 2/fSIO：Noise More than 2/fspl and less than 3/fSIO+1/fsysclk：Indeterminate More than 3/fSIO+1/fsysclk:：Signal	Less than 4/flclk:：Noise More than 4/flclk and less than 8/flclk：Indeterminate More than 8/flclk：Signal
INT1	INT1	IMF=1 IER2.5=1			
INT2	INT2	IMF=1 IER2.6=1			
INT3	INT3	IMF=1 IER2.7=1			
INT4	INT4	IMF=1 IER5.3=1			
INT5	INT5	IMF=1 IER5.4=1			
INT6	INT6	IMF=1 IER5.5=1			
INT7	INT7	IMF=1 IER5.6=1			

Table 6-5 External Interrupts

Note：fsysclk: System clock ; flclk: Low frequency clock [Hz]; fspl: Sampling interval [Hz]

6.8.3.1 Peripheral circuit clock enable function

External interrupts have a function that saves power by using the low power consumption register PCKEN when they are not used. Setting PCKEN3<EINTx> to "0" stops (disables) the basic clock for external interrupts and helps save power. Note that this makes external interrupts unavailable. Setting PCKEN3<EINTx> to "1" supplies (enables) the basic clock for external interrupts and makes external interrupts available.

After reset, PCKEN3<EINTx> is "0" and external interrupts become unavailable. When using the external interrupt function for the first time, be sure to set PCKEN3 <EINTx> to "1" in the initial setting of software (before operating the external interrupt control registers).

Note: Interrupt request signal may be generated when EINTx is changed. Before changing EINTx, first clear the corresponding interrupt enable register (EIR) to "0" to disable the interrupt. When the operating mode is switched from the normal / sleep mode to the normal / sleep mode (low-speed clock), wait for 12 / flclk seconds after the mode transition, and then clear the interrupt latch. When the operating mode is switched from the normal / sleep mode (low-speed clock) to the normal / sleep mode, wait for 2 / fsysclk + 3 / fspl seconds after the mode transition, and then clear the interrupt latch.

6.8.3.2 External Interrupt 0 to 7

External interrupts 0 to 7 detect the falling edge, rising edge, both edges of the INT0 to INT7 pins, and then generate an interrupt request signal.

(a) Interrupt Request Signal Generating Condition Detection Function

Select an interrupt request signal generating condition at EINTCRx<INTxES> for external interrupt 0 to 7.

EINTCRx <INTxES>	Detectedat
00	Rising edge
01	Falling edge
10	Both edges
11	Resreved

Table 6-5 Selection of Interrupt Request Generation Edge

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

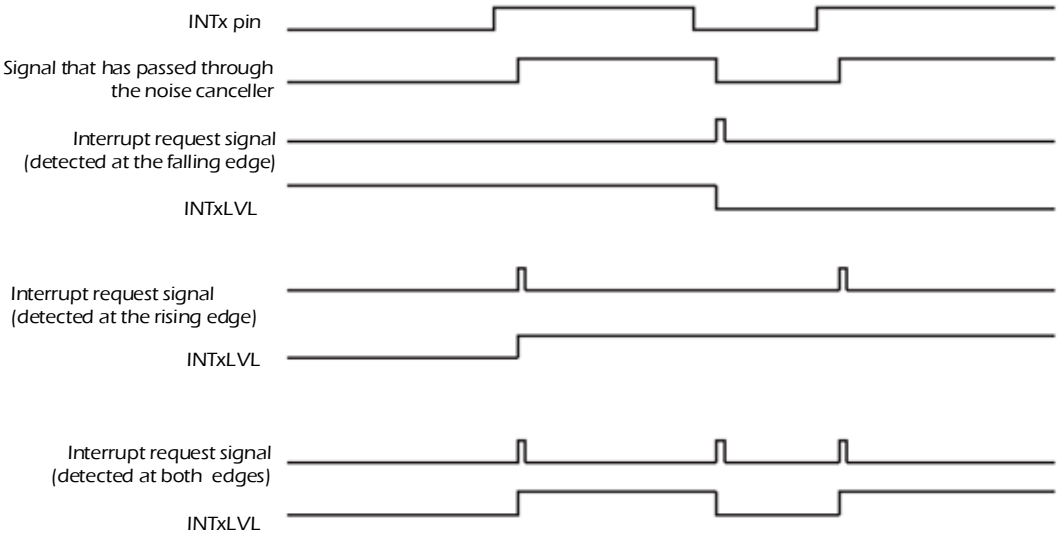


Figure 6-11 Interrupt Request Generation and EINTCRx<INTxLVL>(x = 0~7)

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 NORMAL or SLEEP mode, a signal that has been sampled by fsysclk is sampled at the sampling interval selected at EINTCRx. 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	fsysclk
01	fsysclk /2^2
10	fsysclk /2^3
11	fsysclk /2^4

Table 6-6 Noise Canceller Sampling Clock

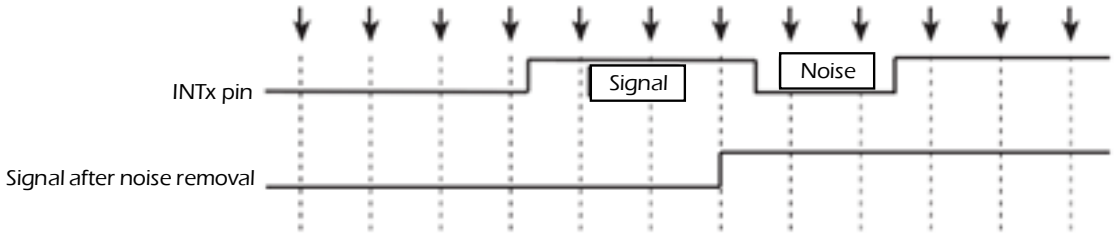


Figure 6-12 Noise Cancel Operation

In NORMAL or SLEEP(low speed clock) 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 DEEP SLEEP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL or SLEEP 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 NORMAL or SLEEP to NORMAL or SLEEP(low speed clock), wait 12/ficlk [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from NORMAL or SLEEP(low speed clock) to NORMAL or SLEEP mode wait 2/ fsysclk +3/fspl [s] after the operation mode is changed and clear the interrupt latch.

6.9 System Power Monitor

This block monitors the system power and generates Power-On Reset (POR) and Brown-Out Reset (BROR) during the power on condition. In addition, there are programmable low voltage and high voltage detection circuits that notify the system when the supply voltage is out of range.

6.9.1 System Power Monitor Control Registers

Address	Register	Description
0x0031	LVDCR	Low Voltage Control Register
0x0034	PONCR	Power-On Control Register

Low Voltage Control Register (LVDCR)

LVDCR	7	6	5	4	3	2	1	0
Bit Symbol	-	LVDCFG [2:0]			LVDST	LVDF	LVDIEN	LVDEN
Read/Write	R/W	R/W			R	R/W1C	R/W	R/W
After reset	0	1	1	1	*	0	0	0

Note1 ： This register is reset by all hardware and software resets.

Note 2： Bit7 must be written by 0。When LVDST is asserted, Bit 7 is cleared.

Note 3： Reserved bits must be written with zeros for future compatibility.

LVDCFG [2:0]	LVD Configuration	000：Reserved 001：2.35V +/-70.5 mV 010：2.65V +/-79.5 mV 011：2.85V +/-85.5mV 100：3.15V +/-94.5 mV 101：3.98V +/-119.4 mV 110：4.2V +/-126 mV 111：4.5V +/-135 mV (Default)
LVDST	LVD Status , when interrupt is generated.	0：No LVD 1：LVD detected
LVDF	LVD Flag	0：No LVD 1：LVD detected
LVDIEN	LVD Interrupt Enable	0：Disable 1：Enable
LVDEN	LVD Enable	0：Disable 1：Enable

Power-on Control Register (PONCR)

PONCR	7	6	5	4	3	2	1	0
Bit Symbol	reserved (Note1)	reserved	reserved (Note1)	reserved (Note1)	reserved	BRORCFG[1:0]		BROREN
Read/Write	R/W	R	R/W	R/W	R	R/W		R/W
After reset	1	0	1	0	0	0		1

Note 1：Bit 7 must be written with 1, Bit 5 must be written with 1, Bit 4 must be written with 0

Note2：Bits are reset by POR reset

Note 3：Reserved bits (Bit 6, and Bit 3) must be written with zeros for future compatibility.

BRORCFG [1:0]	Brown-out reset configuration	00：1.9V +/-57mV(default) 01：2.25V +/-67.5mV 10：2.55V +/-76.5mV 11：2.75V +/-82.5mV
BROREN	Brown-out reset Enable	0：Disable 1：Enable

7. System Clock Controller

This section describes the basic clock controller . Please to set clock input to input pins first before using any of the external clock source.

7.1 Clock Source

SQ7613 clock source as below table：

Clock source	Clock Frequency
High frequency internal reference clock (HIRC)	16 MHz
Low frequency internal reference clock (LIRC)	32 KHz
Phased Locked Loop (PLL)	24 MHz
Low power PLL internal reference clock (LPIRC)	1 MHz，can used to be PLL clock source
High frequency external oscillator (HXTAL)	16 MHz，can used to be PLL clock source

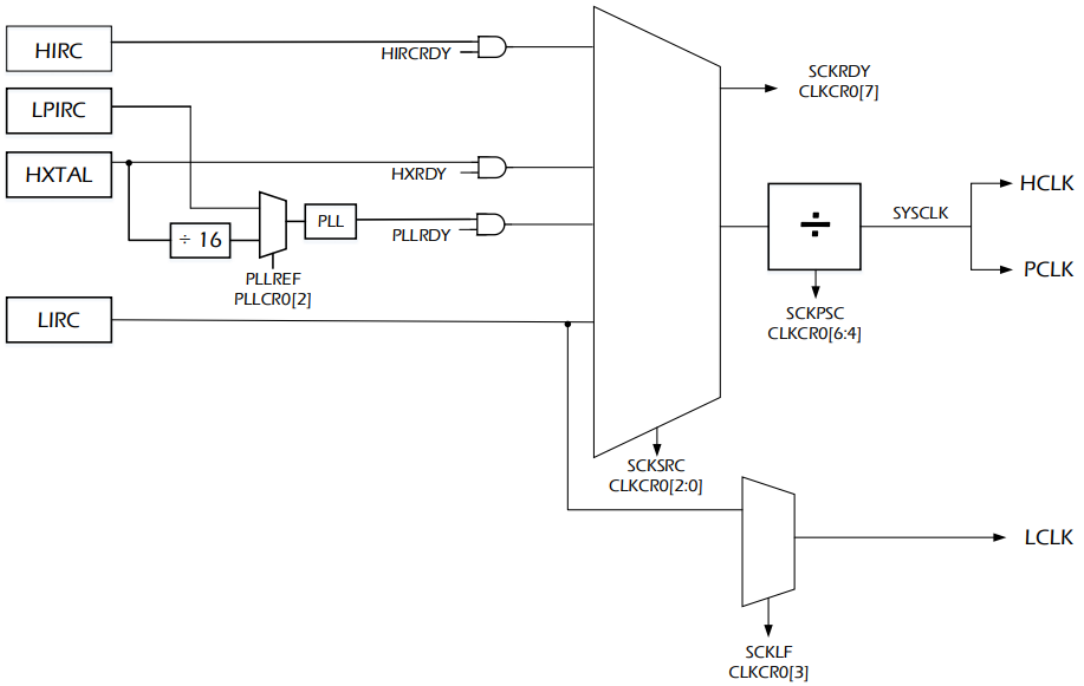


TABLE 7-1 SIMPLIFIED SYSTEM CLOCK DIAGRAM

Note：LIRC is always on.

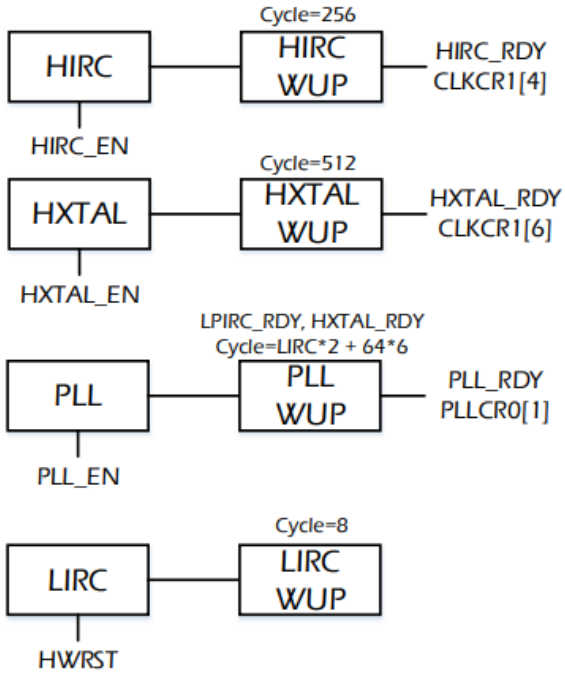


FIGURE7-1 EACH CLOCK START STROKE AND PERIOD

7.2 Clock Switching

Upon exiting a system reset, the HIRC clock is enabled and is selected as the default system clock. System integrity checks and initialization are safely performed using this clock source. When the process is complete, the control is returned to the application. Clock source selection from here on is entirely under software control.

There are two scenarios the clock controller will take control and select a safe clock source to protect the system. In the first scenario, all the clocks, except the LIRC, are disabled by software. This happens when the clock enable bits in the CLKCR1 register are cleared to zeros. The clock controller will switch to the LIRC by setting CLKCR0<SCKSRC> to 0b010. CLKCR0<SCKPSC> is also reset to zeros.

In the second scenario, the clock frequency monitor detects a potential clock issue and notifies the clock controller. The controller then reenables the HIRC, if this clock is not already ON, and switches to this clock source immediately. The CLKCR0<SCKPSC> register is also reset in this case.

7.3 Clock Monitor

The operation of a device relies on the availability and health of its supplied clock. Irregularity in the clock source may ultimately led to erroneous operation. Therefore, it is important that health of the clock source, especially external clock inputs, be monitored. The Clock Monitor Enable (CLKCR3<CMEN>) acts as the master clock monitor enable. Together with individual enable in Clock Monitor Control Register (CLKCR3<CMCR>), different aspect of the external clock inputs can be monitored. When the individual enable is set to 1 (enabled), the specific condition will be monitored. When out of range condition is detected, the status will be reflected in the Clock Monitor Status Register (CMSR). This can generate an interrupt if the Clock Monitor Interrupt Enable is enabled (CLKCR3<CMIE>=1).

To monitor external high frequency clock, the external high frequency clock must be enabled CLKCR1<HXEN> to 1. For low frequency clock glitch detection, HIRC must be enabled CLKCR1<HIRCEN> to 1.

Condition	Description
HX FAIL	External high frequency clock is running out of range **(50 %)**
HX FAST	External high frequency clock is running too fast **(>105 %)**
HX SLOW	External high frequency clock is running too slow **(<95 %)**

Note : When an external clock frequency problem is detected, the clock control automatically enables the HIRC clock and clears the *<SCKSRC>[2:0]* to zeros. In this mode, the default clock is the HIRC clock.

7.4 Clock Control Registers

ADDRESS	REGISTER	DESCRIPTION
0x0020	CLKCR0	Clock Control Register 0
0x0021	CLKCR1	Clock Control Register 1
0x0023	CLKCR3	Clock Control Register 3
0x0024	PLLCR0	PLL Control Register 0
0x0030	CMSR	Clock Monitor Status Register
0x0035	CMCR	Clock Frequency Monitor Register

Clock Control Register 0 (CLKCR0)

CLKCR0	7	6	5	4	3	2	1	0
Bit Symbol	SCKRDY	SCKPSC[2:0]			SCKLF	SCKSRC[2:0]		
Read/Write	R	R/W			R/W	R/W		
After reset	0	0	0	1	0	0		

Note 1：This register is reset by all hardware and software resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

SCKRDY	System Clock Ready	0：Not Ready 1：Ready
SCKPSC [2:0]	System Clock Prescaler	000：/1 (PLL is not supported) 001：/2 010：/4 011：/8 100：/16 101：/32 110：/64 111：/128
SCKLF	System Clock Low Frequency Clock Select	0：LIRC (Default) 1：Reserved
SCKSRC [2:0]	System Clock Source	000：Internal High Frequency Reference Clock (HIRC) 001：Phase Lock Loop (PLL) 010：Internal Low Frequency Reference Clock (LIRC) 011：High Frequency External Crystal (HXTAL) Others：Reserved

Clock Control Register1 (CLKCR1)

CLKCR1	7	6	5	4	3	2	1	0
Bit Symbol	reserved	HXRDY	LPIRCRDY	HIRCRDY	reserved	HXEN	LPIRCEN	HIRCEN
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	1	1	0	0	1	1

Note: This register is reset by all hardware and software resets.

HXRDY	HXTAL Ready	0：Not Ready. 1：Ready.
LPIRCRDY	LPIRC Ready	0：Not Ready. 1：Ready.
HIRCRDY	HIRCReady	0：Not Ready. 1：Ready.
HXEN	HXTAL Enable	0：Disabled. 1：Enabled.
LPIRCEN	LPIRC Enable	0：Disabled. 1：Enabled.
HIRCEN	HIRC Enable	0：Disabled. 1：Enabled. *Note：This bit is automatically enabled when a clock frequency failure is detected.* *Refer to 7.3 clock monitor.*

Clock Control Register 3 (CLKCR3)

CLKCR3	7	6	5	4	3	2	1	0
Bit Symbol	CMIE	reserved	HXCMEN	reserved				
Read/Write	R/W	R	R/W	R				
After reset	0	0	0	0				

Note 1：This register is reset by all resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

CMIE	Clock Monitor Interrupt Enable	0：Disabled. 1：Enabled.
HXCMEN	High frequency crystal Monitor Enable	0：Disabled. High Frequency related enable in CMCR has no effect .1：Enabled.

PLL Control Register0 (PLLCR0)

PLLCR0	7	6	5	4	3	2	1	0
Bit Symbol	reserved					PLLREF	PLLRDY	PLLEN
Read/Write	R					R/W	R	R/W
After reset	0					0	0	0

Note 1：This register is reset by all hardware and software resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

PLLREF	PLL Reference Clock Select	0：LPIRC 1：HXTAL
PLLRDY	PLL Ready	0：Not Ready. 1：Ready.
PLLEN	PLL Enable	0：Disabled. 1：Enabled.

Clock Frequency Monitor Register (CMCR)

CMCR	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	reserved	HXSLWEN	HXFSTEN	HXFAILEN
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1：This function is used to detect potential problems with external clocks. The detection status is displayed in the clock monitoring status register CMSR.

Note 2：This register is reset by all hardware and software resets.

HXSLWEN	HXTAL Slow Detect Enable	0：Disabled. 1：Enabled.
HXFSTEN	HXTAL Fast Detect Enable	0：Disabled. 1：Enabled.
HXFAILEN	HXTAL Fail Detect Enable	0：Disabled. 1：Enabled.

Clock Monitor Status Register (CMSR)

CMSR	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	reserved	HXSLW	HXFST	HXFAIL
Read/Write	R	R	R	R	R	R/W1C	R/W1C	R/W1C
After reset	0	0	0	0	0	0	0	0

Note ：This register is reset by all hardware and software resets.

HXSLW	HXTAL Slow Detect	0：Not detected. 1：Detected.
HXFST	HXTAL Fast Detect	0：Not detected. 1：Detected.
HXFAIL	HXTAL Fail Detect	0：Not detected. 1：Detected.

7.5 System and Peripheral clocks

7.5.1 Functional Clock Gating

All functions are individually controlled using the peripheral clock enable register, PCKENx.

ADDRESS	BIT SYMBOL	DESCRIPTION
0x0178	PCKEN0	Peripheral Clock Enable Register 0
0x0179	PCKEN1	Peripheral Clock Enable Register 1
0x017A	PCKEN2	Peripheral Clock Enable Register 2
0x017B	PCKEN3	Peripheral Clock Enable Register 3
0x017C	PCKEN4	Peripheral Clock Enable Register 4
0x017D	PCKEN5	Peripheral Clock Enable Register 5
0x017E	PCKEN6	Peripheral Clock Enable Register 6
0x017F	PCKEN7	Peripheral Clock Enable Register 7

Peripheral Clock Enable Register0(PCKEN0)

PCKEN0	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN0[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN0[0]	reserved
PCKEN0[1]	reserved
PCKEN0[2]	reserved
PCKEN0[3]	reserved
PCKEN0[4]	TCA0
PCKEN0[5]	TCA1
PCKEN0[6]	TCA2
PCKEN0[7]	TCA3

Peripheral Clock Enable Register1 (PCKEN1)

PCKEN1	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN1[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN1[0]	TCA4
PCKEN1[1]	TCA5
PCKEN1[2]	TCA6
PCKEN1[3]	TCA7
PCKEN1[4]	UART0
PCKEN1[5]	UART1
PCKEN1[6]	UART2
PCKEN1[7]	reserved

Peripheral Clock Enable Register2(PCKEN2)

PCKEN2	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN2[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN2[0]	I2C0
PCKEN2[1]	I2C1
PCKEN2[2]	reserved
PCKEN2[3]	reserved
PCKEN2[4]	SIO0
PCKEN2[5]	SIO1
PCKEN2[6]	reserved
PCKEN2[7]	reserved

Peripheral Clock Enable Register3 (PCKEN3)

PCKEN3	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN3[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN3[0]	EINT0
PCKEN3[1]	EINT1
PCKEN3[2]	EINT2
PCKEN3[3]	EINT3
PCKEN3[4]	EINT4
PCKEN3[5]	EINT5
PCKEN3[6]	EINT6
PCKEN3[7]	EINT7

Peripheral Clock Enable Register4 (PCKEN4)

PCKEN4	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN4[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN4[0]	reserved
PCKEN4[1]	reserved
PCKEN4[2]	reserved
PCKEN4[3]	reserved
PCKEN4[4]	reserved
PCKEN4[5]	reserved
PCKEN4[6]	reserved
PCKEN4[7]	reserved

Peripheral Clock Enable Register5(PCKEN5)

PCKEN5	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN5[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN5[0]	reserved
PCKEN5[1]	MAC
PCKEN5[2]	reserved
PCKEN5[3]	reserved
PCKEN5[4]	reserved
PCKEN5[5]	reserved
PCKEN5[6]	reserved
PCKEN5[7]	reserved

Peripheral Clock Enable Register6(PCKEN6)

PCKEN6	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN6[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN6[0]	reserved
PCKEN6[1]	reserved
PCKEN6[2]	ADC
PCKEN6[3]	reserved
PCKEN6[4]	reserved
PCKEN6[5]	reserved
PCKEN6[6]	reserved
PCKEN6[7]	reserved

Peripheral Clock Enable Register7(PCKEN7)

PCKEN7	7	6	5	4	3	2	1	0
Bit Symbol	PCKEN7[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

Each bit in this register enable the specified peripheral clock supply. In order for the specified peripheral to operate, its clock must be enabled. To enable clock, set the corresponding bit to 1.

Bit Symbol	Peripheral Clock
PCKEN7[0]	reserved
PCKEN7[1]	CRC
PCKEN7[2]	reserved
PCKEN7[3]	reserved
PCKEN7[4]	reserved
PCKEN7[5]	reserved
PCKEN7[6]	reserved
PCKEN7[7]	reserved

8. 12-bit ADC

SQ7613 has a real 12-bit AD converter (ADC), which is a successive approximation type ADC.The ADC channels of SQ7613 is 7 channels. (AIN4-AIN10).

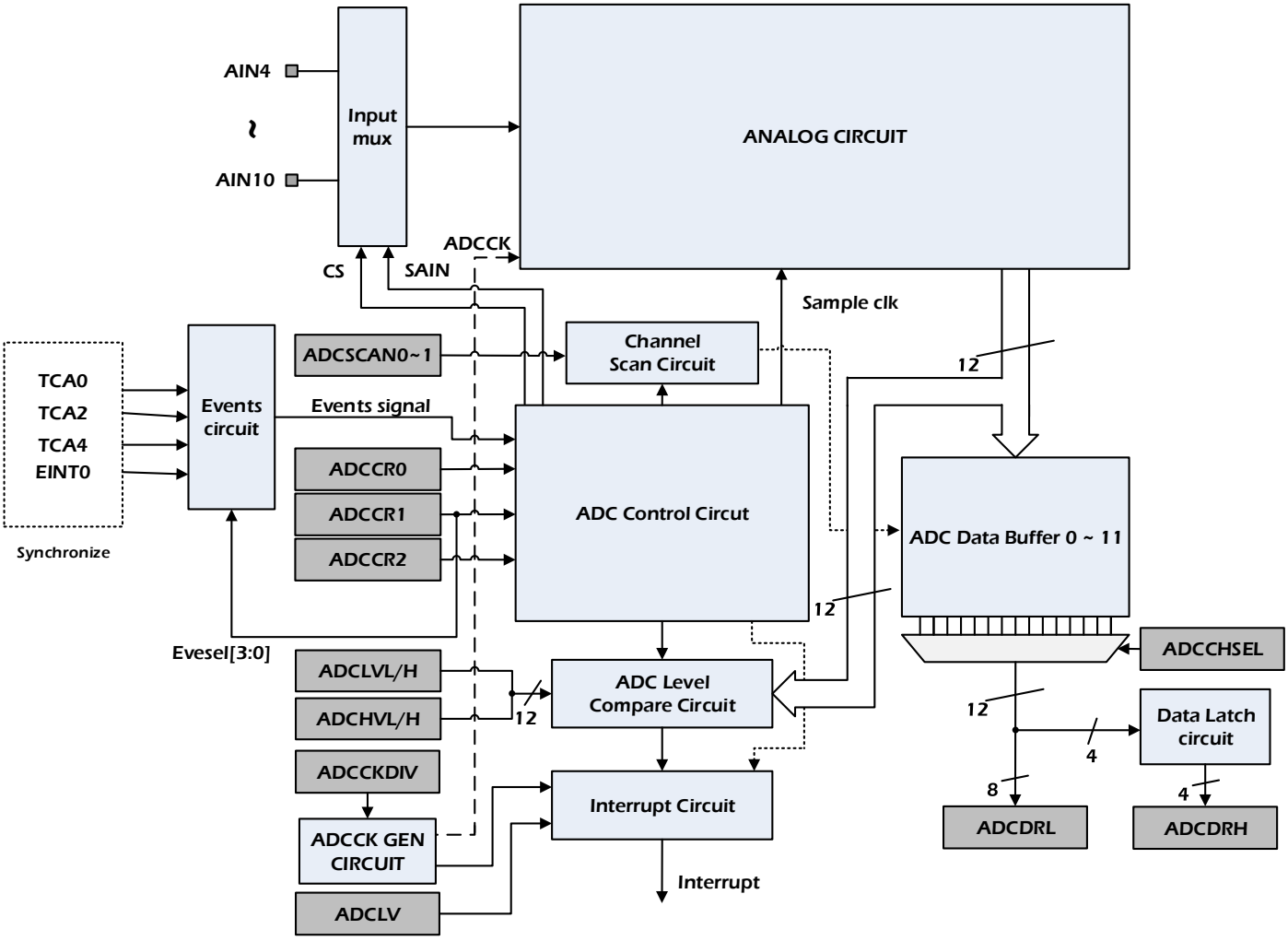


FIGURE 8-1 ADC BLOCK DIAGRAM

Note：Before using the AD converter, you need to set the port function control register (PxFC1, PxFC2) to 1.

8.1 Function

The circuit configuration of the AD converter is shown in figure 8-1. It consists of four control registers (ADCCR0 – 3), converted data registers ADCDRL and ADCDRH, a sample-hold circuit, a comparator, a successive comparison circuit, etc.

8.1.1 ADC Contro Register

1. ADC Control Register 0 (ADCCR0)
This register selects an ADC operating mode, auto power down setting, interrupt enable,interrupt interval, internal reference enable, and start of the AD converter.

2. ADC Control Register 1 (ADCCR1)
This register selects the trigger event.

3. ADC Control Register 2 (ADCCR2)
This register enable the level interrupt.

4.ADC Clock Divide Register (ADCCKDIV)
This register can set ADC clock, and define the ADC clock divider in relationship to the system clock.

5.ADC Level Registers (ADCLV)
This register can enable level compare and setting the interval and channel.

6.ADC Scan Register (ADCSCANx, x=0~1)
This register can control ADC channel scan.

7.ADC Status Register (ADCSR)
This register collect ADC status.

8.ADC Channel Ready Register (ADCCHRDY)
This register is updated with the channel number of the least ADC conversion channel.

9.ADC Channel Select Register (ADCCHSEL)
This register is defined the channel from which ADC data is read.

10. ADC Data Register (ADCDRH and ADCDRL)
ADC data register (ADCDRH and ADCDRL), contains the ADC converted value.

11. ADC high Level Register (ADCHLVH and ADCHLVL)
This register contains the ADC high level comparison value.

12. ADC Low Level Register (ADCLLVH and ADCLLVL)
This register contains the ADC low level comparison value.

8.1.2 Data Buffer

Each ADC channel has its own data buffer. After each conversion, ADC result will be written to the data buffer correspond to the associated channel. In addition, the ADC Data Ready Register will also be updated with the ADC channel number.

To access a specific channel data, software will write the channel number to ADC Channel Select Register (ADCCHSEL), and read from ADC Data Register (ADCDRL/ADCDRH). User should read ADCDRL first, then ADCDRH.

Once the ADC conversion result is written to its data buffer, the data buffer is locked until one of the following condition occurs:

1. ADCDR has been read. For a particular channel, if ADCDRL is read, the data buffer for this channel will be locked until ADCDRH is also read to preserve integrity.

2. Data buffer unlock (ADCSR<UNLCK> = 1). In this mode, the current content of the ADC buffer is unlocked – treated as if user software has read all of the content so new conversion result can be written to the data buffer.

3. Data buffer override is enabled (ADCSR<BUFOVR>= 1). In this mode, ADC data will be continuously updated when available

8.1.3 Multiple channel scanning

ADC Scan Registers ADCSCAN0 and ADCSCAN1 support Multiple channel scanning. Each register bit represents an ADC channel. A value of "1" indicates that the corresponding channel will be included in the channel scanning. A value of "0" indicates that the corresponding channel will be excluded from the channel scanning. ADC scanning order can only be changed when ADC is idle (ADCSR<ADBF>=0). If ADC conversion is in progress, ADC conversion must first be stopped (ADCCR0<AMD>=00) before changes can be made to ADC Channel Scan Registers.

8.1.4 ADC Clock Selection

ADC sampling clock is derived from the system clock. The divide ratio can be set via the ADC Clock dovode register ADCCKDIV.

8.1.5 ADC Reference

By default, the ADC use an external reference supplied by VREF pin. An internal reference is available when external reference is not available. The internal reference is enable by setting ADCCR0<IRFEN> to 1.

8.1.6 ADC Event Source

AD conversion can be started by setting ADCCR1<EVSEL> to 00, and ADCSR<ADRS> to 1 when ADCCR1<EVSEL> is set to 0.When ADCCR1<EVSEL> is set to other values, AD conversion is triggered by the source selected and ADRS input is ignored.

8.1.7 ADC Level Comparison

In some application, user is only interested when ADC value is within a certain range. To minimize unnecessary interrupt to the CPU, the ADC Level Compare Enable ADCLV<LVCMP> and ADC High Level (ADCHLVL, ADCHLVH) and ADC Low Level (ADCLLVL, ADCLLVH) Registers can be used to filter out ADC value.

LVCMP	ADC conversion finished interrupt (INTADC) is generate when	Interrupt condition
00	At the end of every conversion.	Generate interrupt at the end of every conversion.
01	ADC data < ADCLLV	ADCLLV = 0x0060，ADCHLV=0x3FFF， generate interrupt when ADCDR ≦ 0x005F
10	ADC data > ADCHLV	ADCLLV = 0x0000，ADCHLV=0x005F， generate interrupt when ADCDR ≧ 0x0060
11	ADCLLV < ADC data and ADC data > ADCHLV	ADCLLV = 0x0200，ADCHLV=0x00FF， generate interrupt when ADCDR is from 0x100 to 0x1FF ADCLLV = 0x0100，ADCHLV=0x01FF， generate interrupt when ADCDR≦ 0x00FF or ADCDR ≧ 0x0200

Level Compare Interval determines whether level comparison is applied to all the samples or specific sample. When all samples are chosen (LVINTVL=0), the Level comparison will be performed on each of the scanned channels. When only specific sample is required (LVINTVL=1), the specified channel number is written to Level Compare Channel Select (LVSEL) and only samples from this channel will be compared against. When the compare condition (LVCMP) is met, the Level Compare Detect (LVDET) will be set to 1.

8.1.8 Interrupt Generation

There are two ways to generate an interrupt, End of Conversion Flag (EOCF) or Level Detect (LVDET).

1. EOCF. If interrupt interval (INTVL) is cleared to 0 (INTVL=0), EOCF is set to 1 at the end of each sample conversion. If INTVL=1, EOCF will be set to 1 at the end of the scan sequence. EOCF will remain set until cleared by software. To clear EOCF to 0, write '1' to EOCF. Writing 0 to EOCF has no effect. If ADC interrupt is enabled (INTEN=1), and EOCF is set to 1, an interrupt will be generated.

2. LVDET. When the comparison condition met, the Level Compare Detect (LVDET) will be set to 1. LVDET will remain set unless cleared by software. To clear LVDET to 0, write '1' to LVDET. Writing 0 to LVDET has no effect. If ADC Leve Interrupt is enabled (LVINTEN=1), setting LVDET to 1 will generate an interrupt.

Regardless of interrupt setting, the ADC busy flag(ADBF) will remain set until all ADC conversion are completed.

8.1.9 ADC operating mode

The 12-bit AD converter operates in either one-shote mode in which AD conversion is performed only one-shot mode or repeat mode in which AD conversion is performed repeatedly.

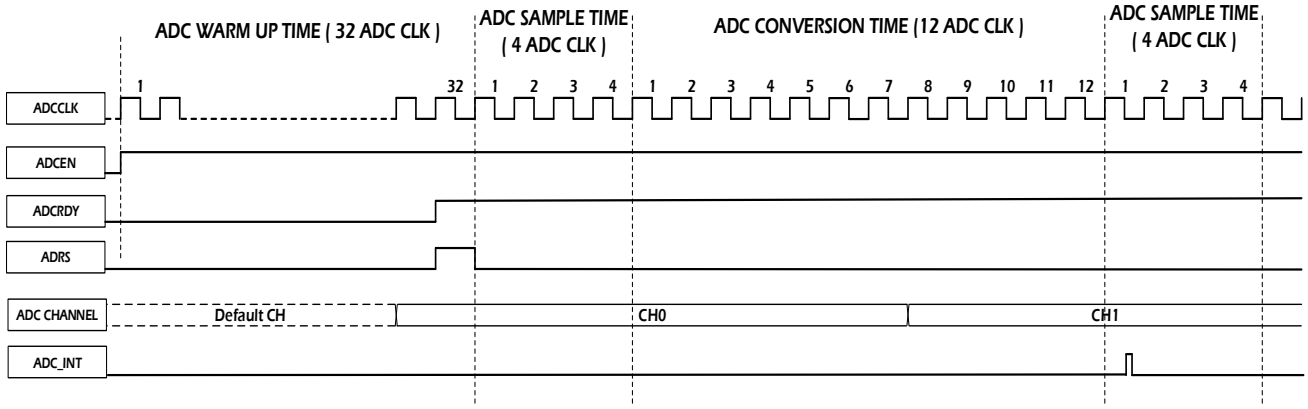


FIGURE 8- 2 ADC POWER-ON

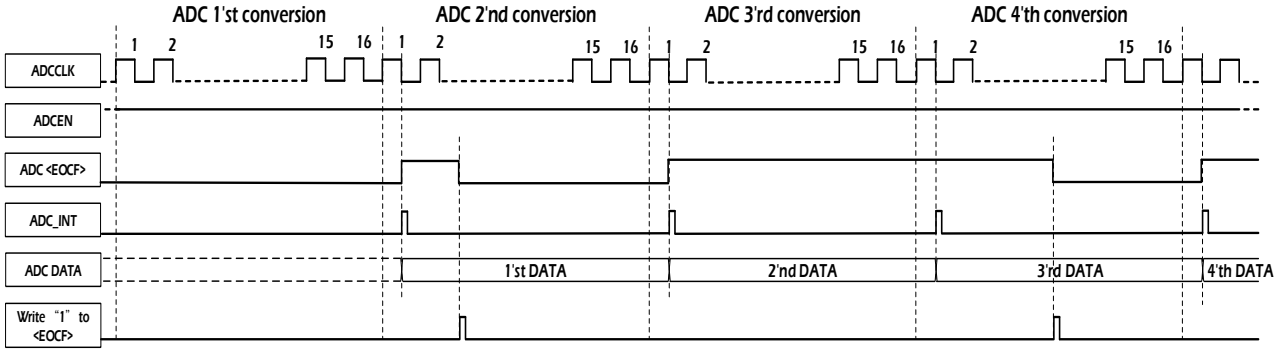


FIGURE 8- 3 ADC<EOCF> AND DATA BUS

8.1.9.1 One-Shot Mode

In one-shot mode, the voltage at the selected analog input (AIN) is measured only once.

After AD conversion starts, ADCSR<ADBF> is set to "1" . ADCSR<ADBF> is clear to "0", when AD conversion is finished or AD conversion is forced to stop.

After AD conversion is finished, the conversion result is stored in the AD data registers (ADCDRL and ADCDRH), ADCSR<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. The ADC data registers (ADCDRL and ADCDRH) can be read in the INTADC interrupt processing routine.

After AD conversion starts (ADCSR<ADRS>=1), and ADC one-shot mode complete, the value is save to AD data registers (ADCDRL and ADCDRH), you can read from ADCDRL/ADCDRH. When next converting start, <EOCF> would not clear to 0 automatically, clear <EOCF> to 0 by software setting.

To conserve power, the ADC can be configured to automatically power down after each conversion. When ADCCR0<AUTOPD> is set to 1, ADC will be powered down after each conversion. When AUTOPD is cleared to 0, ADC will remains powered on. Note that if ADC is automatically powered down after each conversion, there will be a delay when the next conversion is initiated. The delay is the same as if the ADC is initially enabled.

After ADCCR0<ADEN> is "1", you need to wait ADRDY to "1", then start ADC convertion (ADCSR<ADRS> is 1). After finish AD convertion, ADCCR0<ADEN> clear to "0" automatically. To start next AD convertion, please repeat it again.

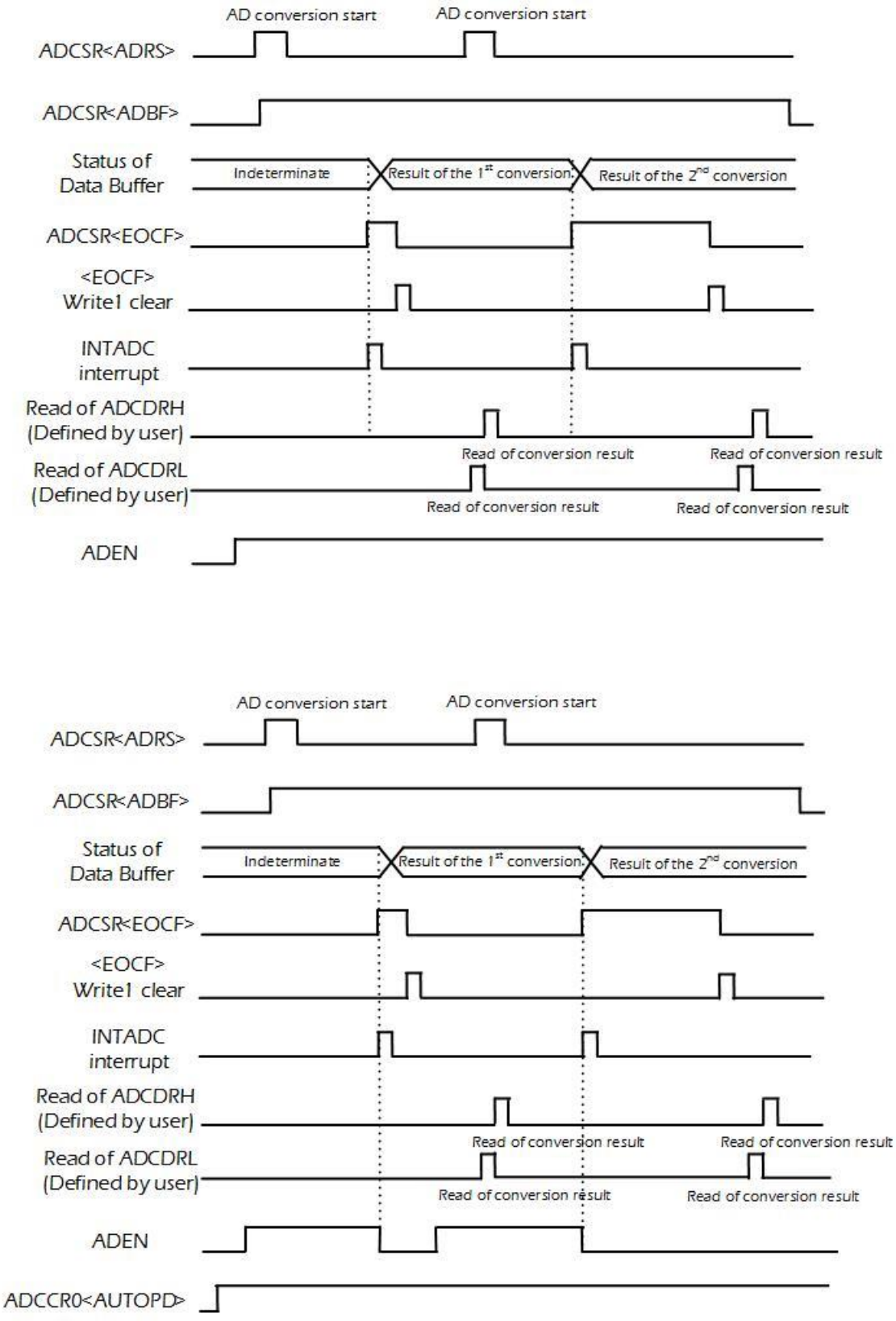


FIGURE 8- 4 ADC ONE-SHOT MODE

8..1.9.2 Repeat Mode

In repeat mode, the voltage at the selected analog input <ADCSCAN> is measured repeatedly.

To use repeat mode, set ADCCR0<AMD> to "11". Setting ADCSR<ADRS> to "1" starts AD conversion.

After AD conversion starts, ADCSR<ADRS> is automatically cleared. After the first AD conversion is finished, the conversion result is stored in the ADC data registers (ADCDRL and ADCDRH), ADCSR<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.

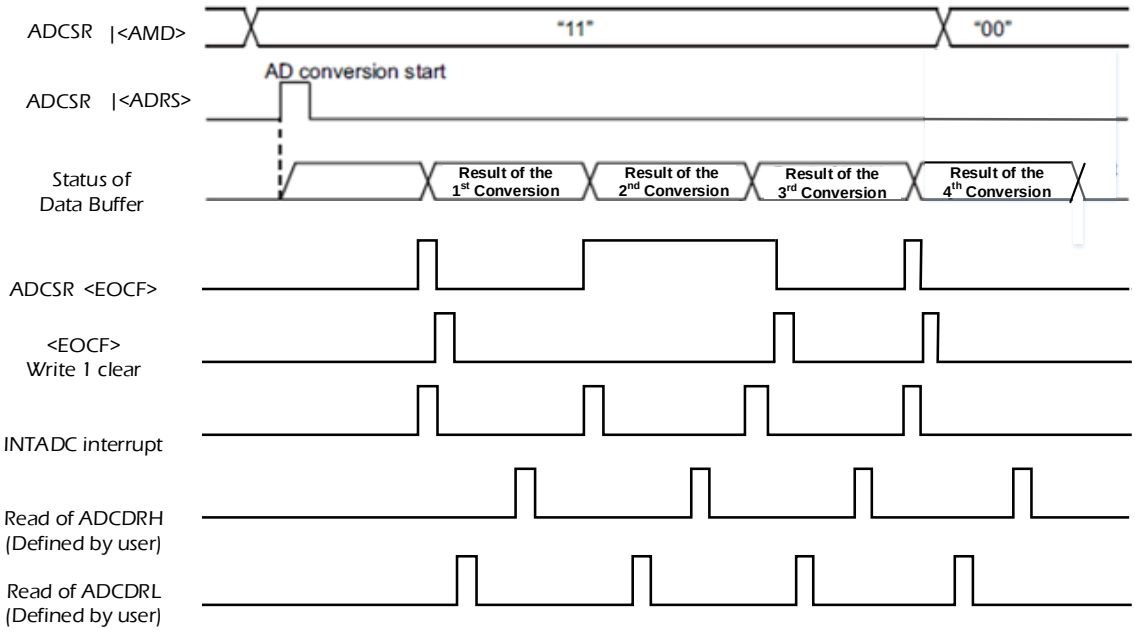


FIGURE 8- 5 ADC REPEAT MODE

8.1.10 ADC Operation Disable

Regardless of operation mode, AD converter can be forced to stop by setting ADCCR0<AMD> to "00".

When ADCCR0<AMD> is set to "00":
- AD conversion stops immediately
- Converted value is not stored in the AD data register.
- ADCSR<EOCF>, ADCSR<ADBF> are initialized to "0".

After power-down, <ADEN>=0 , and ADCDRL/ADCDRH are initialize to "0".

8.1.11 ADC Register Setting

1. Set ADC Clock Frequency with ADCCKDIV<ADCCKDIV>

2. Enable ADC by setting ADCCR1<ADEN> to 1, then Select ADC Reference ADCCR1<IRFEN> and Level Comparator ADCCR1<LVCMP>.

3. Select ADC trigger event source ADCCR1<EVSEL>.

4. Select ADC operation mode ADCCR0<AMD>

5. Ensure ADCSR <ADRDY> is 1

6. Start ADC conversion by setting ADCCR1<ADRS> to 1.

7. When ADC conversion is finished, the AD conversion end flag ADCSR<EOCF> is set to 1. The AD conversion result is stored in the ADC data registers (ADCDRH and ADCDRL), and the INTADC interrupt request is generated.

8. After the conversion result is read from the AD data register (ADCDRH), EOCF is cleared to 0 (write 1 clear). EOCF will also be cleared to 0 if AD conversion is performed once again before reading the AD data register (ADCDRH). In this case, the previous conversion result is retained until AD conversion is finished.

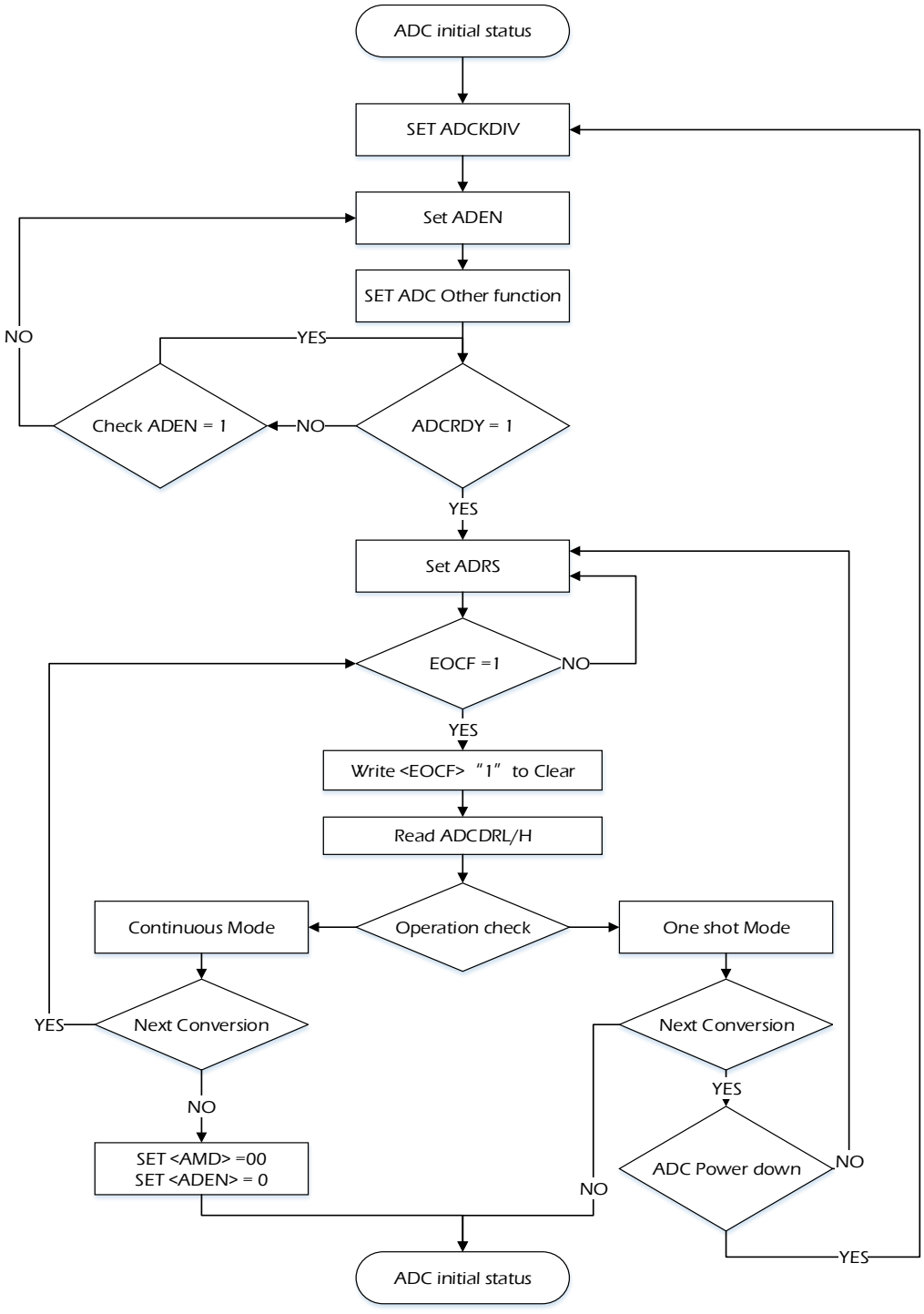


FIGURE 8- 6 ADC FLOW CHART

8.2 Precautions about the AD converter

8.2.1 Analog input pin voltage range

Analog input pins (AIN) should be used at voltages from VREF 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.

8.2.2 Analog input pins used as input/output ports

Analog input pins 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.

8.2.3 Noice Countermeasure

The internal equivalent circuit of the analog input pins is shown below Figure 8-7. 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 kΩor less. It is recommended that a capacitor be attached externally.

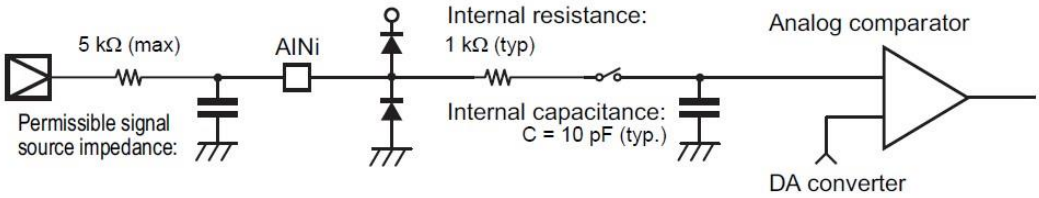


Figure 8- 7 Analog Input Equivalent Circuit and Example of Input Pin Processing

Note：i=4~10

8.3 Control

Address	Register	Description
0x0300	ADCCR0	ADC Control Register 0
0x0301	ADCCR1	ADC Control Register 1
0x0302	ADCCR2	ADC Control Register2
0x0304	ADCCKDIV	ADC Clock Divide Register
0x0306	ADCLV	ADC Level Registers
0x0307	ADCSCAN0	ADC Scan Register 0
0x0308	ADCSCAN1	ADC Scan Register 1
0x030A	ADCSR	ADC Status Register
0x030B	ADCCHRDY	ADC Channel Ready Register
0x030C	ADCCHSEL	ADC Channel Select Register
0x0310	ADCDRL	ADC Data Register (Low byte)
0x0311	ADCDRH	ADC Data Register (High byte)
0x0312	ADCLLVL	ADC Low Level Register (Low byte)
0x0313	ADCLLVH	ADC Low Level Register (High byte)
0x0314	ADCHLVL	ADC High Level Register (Low byte)
0x0315	ADCHLVH	ADCHigh Level Register (High byte)

ADC Control Register 0 (ADCCR0)

ADCCR0 (0x0300)	7	6	5	4	3	2	1	0
Bit Symbol	AMD[1:0]		AUTOPD	INTLV	INTEN	IRFEN[1:0]		ADEN
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
After reset	0	1	0	0	0	0	1	0

Note1：This register is reset by all hardware and software resets.

AMD [1:0]	AD Operating Mode	00：ADC operation disable, forcibly stop ADC operation 01：One-Shot 10：Reserved 11：Repeat
AUTOPD	Auto Power Down	0：Do not power down ADC in between conversion 1：Automatically power down ADC after every conversion in single mode
INTLV	Interrupt Interval	0：Every sample 1：End of scan
INTEN	Interrupt Enable	0： Disable 1： Enable
IRFEN [1:0]	Internal Reference Enable	01：VDDA_ADC 10：External Reference Others：Reserved
ADEN	AD Enable	0：ADC Disable 1：ADC Enable

ADC Control Register 1 (ADCCR1)

ADCCR1 (0x0301)	7	6	5	4	3	2	1	0
Bit Symbol	EVSEL[3:0]				reserved			
Read/Write	R/W				R			
After reset	0				0			

Note1：This register is reset by all hardware and software resets.

EVSEL0 [3:0]	Event Select	0000：ADRS 1000：TCA0 1001：TCA2 1010：TCA4 1100：EINT0 Others：Reserved

ADC Control Register 2 (ADCCR2)

ADCCR2 (0x0302)	7	6	5	4	3	2	1	0
Bit Symbol	reserved							LVINTEN
Read/Write	R							R/W
After reset	0							0

Note1：This register is reset by all hardware and software resets.

LVINTE	Level Interrupt Enable	0：Disable 1：Enable

ADC Clock Divide Register (ADCCKDIV)

ADCCKDIV (0x0304)	7	6	5	4	3	2	1	0
Bit Symbol	Reserved				ADCKDIV[3:0]			
Read/Write	R				R/W			
After reset	0				0			

Note1：This register is reset by all hardware and software resets.

Note2：This register is not writable when ADC is busy.

ADCKDIV[3:0]	ADC Clock Divider： These bits define the ADC clock divider in relationship to the system clock.	0000 : fsysclk 0001 : fsysclk/2 0010 : fsysclk/2^2 0011 : fsysclk/2^3 0100 : fsysclk/2^4 0101 : fsysclk/2^5 0110 : fsysclk/2^6 0111 : fsysclk/2^7 1000 : fsysclk/2^8 1001 : fsysclk/2^9

ADC Level Registers (ADCLV)

ADCLV (0x0306)	7	6	5	4	3	2	1	0
Bit Symbol	LVCMP[1:0]		LVINTVL	LVSEL[4:0]				
Read/Write	R/W		R/W	R/W				
After reset	0		0	0				

Note：This register is reset by all hardware and software resets.

LVCMP[1:0]	Level Compare Enable	00：Level compare disable 01：Low level compare (ADCLLV < ADC data) 10：High level compare (ADC data > ADCHLV) 11：Both high and low level compare (ADCLLV < ADC data and ADC data > ADCHLV)
LVINTVL	Level Compare Interval	0：Applies to all sample 1：Applies to channel specified in LVSEL
LVSEL [4:0]	Level Compare Channel Select	These bits select the channel to compare when LVINTVL=1

ADC Scan Register 0 (ADCSCAN0)

ADCSCAN0 (0x0307)	7	6	5	4	3	2	1	0
Bit Symbol	ADCSCAN[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

ADCSCAN [7:0]	ADC Channel Scan Bits [7:0]	This register contains bit [7:0] of the ADC channel scan control.

ADC Scan Register 1 (ADCSCAN1)

ADCSCAN1 (0x0308)	7	6	5	4	3	2	1	0
Bit Symbol	Reserved				ADCSCAN[11:8]			
Read/Write	R/W				R/W			
After reset	0				0			

Note：This register is reset by all hardware and software resets.

ADCSCAN [11:8]	ADC Channel Scan Bits [11:8]	This register contains bit [11:8] of the ADC channel scan control.

ADC Status Register (ADCSR)

ADCSR (0x030A)	7	6	5	4	3	2	1	0
Bit Symbol	EOCF	ADBF	ADRDY	reserved	BUFOVR	UNLCK	LVDET	ADRS
Read/Write	R/W1C	R	R/W	R	R/W	R/W	R/W1C	R/W
After reset	0	0	0	0	0	0	0	0

Note：This register is reset by all hardware and software resets.

EOCF	AD conversion end flag	0：Before conversion or conversion is in progress. 1：Conversion is completed.
ADBF	AD conversion BUSY flag	0：AD conversion halted 1：AD conversion in progress
ADCRDY	ADC Ready Flag	0：ADC not ready 1：ADC ready
BUFOVR	Data Buffer Override	0：Disabled. 1：Enabled.
UNLCK	Data Buffer Unlock	0：Data buffer locked 1：Data buffer unlocked
LVDET	Level Compare Detect	0：No level compare detect 1：Level compare detect (write 1 clear)
ADRS	ADC start	0：- 1：ADC start

ADC Channel Ready Register (ADCCHRDY)

ADCCHRDY (0x030B)	7	6	5	4	3	2	1	0
Bit Symbol	reserved			CHRDY[4:0]				
Read/Write	R			R				
After reset	0			0				

Note：This register is reset by all hardware and software resets.

CHRDY [4:0]	Channel Reay	These bits are updated with the channel number of the least ADC conversion channel.

ADC Channel Select Register (ADCCHSEL)

ADCCHSEL (0x030C)	7	6	5	4	3	2	1	0
Bit Symbol	reserved			CHSEL[4:0]				
Read/Write	R			R/W				
After reset	0			0				

Note：This register is reset by all hardware and software resets.

CHSEL [4:0]	Channel Select	These bits defines the channel from which ADC Data is read.

ADC Data Register (Low byte) (ADCDRL)

ADCDRL (0x0310)	7	6	5	4	3	2	1	0
Bit Symbol	ADCDRL[7:0]							
Read/Write	R							
After reset	0							

Note：This register is reset by all hardware and software resets.

ADCDRL[7:0]	ADC Data Register Low Byte.	This register contains bit [7:0] of lower bits of the ADC converted value.

ADC Data Register (High byte) (ADCDRH)

ADCDRH (0x0311)	7	6	5	4	3	2	1	0
Bit Symbol	reserved				ADCDRH[3:0]			
Read/Write	R				R			
After reset	0				0			

Note：This register is reset by all hardware and software resets.

ADCDRH[3:0]	ADC Data Register High Byte	This register contains upper bits of the ADC converted value.

ADC Low Level Register (Low byte) (ADCLLVL)

ADCLLVL (0x0312)	7	6	5	4	3	2	1	0
Bit Symbol	ADCLLVL[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

ADCLLVL[7:0]	ADC Low Level Register Low Byte.	This register contains bit [7:0] of the ADC low level comparison value.

ADC Low Level Register (High byte) (ADCLLVH)

ADCLLVH (0x0313)	7	6	5	4	3	2	1	0
Bit Symbol	reserved				ADCLLVH[3:0]			
Read/Write	R				R/W			
After reset	0				0			

Note：This register is reset by all hardware and software resets.

ADCLLVH[3:0]	ADC Low Level Register High Byte	This register contains upper bits of the ADC low level comparison value.

ADC High Level Register(Low byte) (ADCHLVL)

ADCHLVL (0x0314)	7	6	5	4	3	2	1	0
Bit Symbol	ADCHLVL[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all hardware and software resets.

ADCHLVL[7:0]	ADC High Level Register Low Byte.	This register contains bit [7:0] of the ADC high level comparison value.

ADC High Level Register(High byte) (ADCHLVH)

ADCHLVH (0x0315)	7	6	5	4	3	2	1	0
Bit Symbol	reserved				ADCHLVH[3:0]			
Read/Write	R				R/W			
After reset	0				0			

Note：This register is reset by all hardware and software resets.

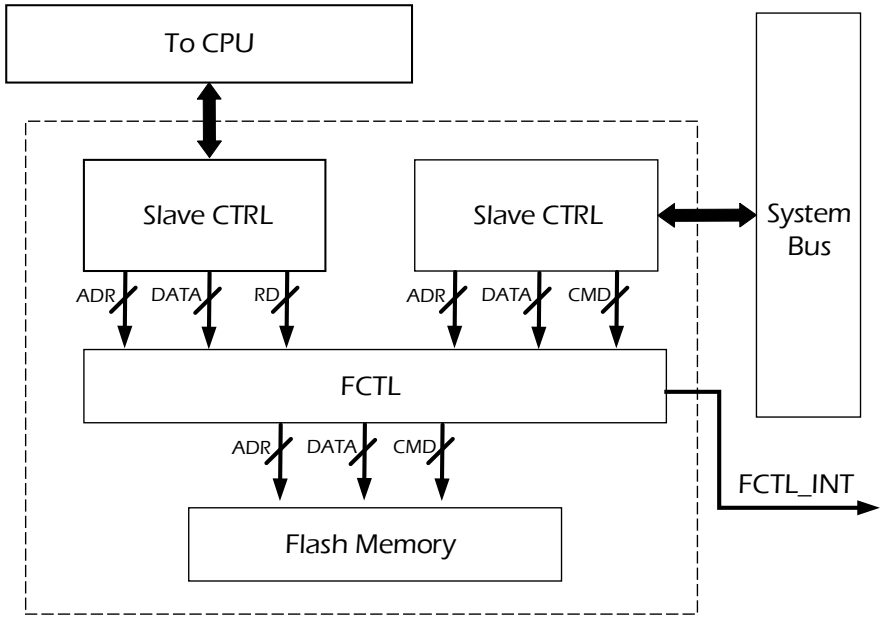
ADCHLVH[3:0]	ADC High Level Register High Byte.	This register contains upper bits of the ADC high level comparison value.

9. Flash Memory Controller (FMC)

Flashmemory controller (Flash Memory Contorller, FMC) can control write and erase to be performed on flash memory in the MCU mode. FMC can support below job types:

- Byte read-access
- Byte programing
- Sector erase and mass erase

After progrmming or erase complete, FMC generate interrupt request. Flash memory controller block shows as below figure:



FIGUTE 9- 1 FLASH MEMORY CONTROLLER BLOCK

9.1 Function

In read mode, the slave interface accepts read request from the system bus. If the controller is not performing a programming or an erase operation, the request is sent to the Flash memory in the same cycle. For zero-wait state access, the read data is returned to the system bus the next cycle. For non-zero-wait state, the read data is returned when the wait states are over. In case, a programming or an erase operation is currently in progress, the controller immediately holds the bus ready low to delay the read access. When the operation is complete, the BUSY indication is removed and the read access can proceed. The ready signal returns to logic one when the read data is ready. Flash frequency must be 1MHz, when flash program/read/write/ erase. Flash frequency can be set by FCKDIV. The flash frequency is Fsysclk/(FCKDIV+1). In reset default condition，flash frequency is 1MHz, user can perform flash program/read/write/erase without setting FCKDIV.

In programming and erase mode, the interface is used to set up the FMC registers. There are two address registers(FADDR0,FADDR1), two data registers (FDATA0,FDATA1), and two control registers(FCR0,FCR1). The address registers accommodate 64kB address space. The data registers can hold up to 16-bit data. The control registers are used for operation configuration. In any operation, a command must be written last into the control register 0 (FCR0) to start the operation.

Brief eaxample:
Setting DATASZ as"00" (byte data) , an address is first entered into the address registers(FADDR0,FADDR1). The write data is then written into the data registers (FDATA0, FDATA1). The command to perform the operation must be entered last into the control register. The operation is started as soon as the last write is complete. Additional register writes to any registers are ignored until the operation is finished and the BUSY bit is cleared. For page erase, only the address is required. For mass erase, only the command is needed.

Peform falsh memory control by the steps as below :

1. Ensure flash page datais 0xFF (flash is programmed from 1 to 0 only). If flash page data is not 0xFF then "PROGA_ERR"
2. Input data to flash.
3. Read data from flash then compare with input data. If the data are not match, then "PROGD_ERR".

9.2 Flash Memory Control Register

Address	Register	Description
0x0027	FCKDIV	Flash Clock Divider Register
0x0040	FCR0	Flash Control Register 0
0x0041	FCR1	Flash Control Register 1
0x0042	FADDR0	Flash Address Register 0
0x0043	FADDR1	Flash Address Register 1
0x0044	FDATA0	Flash Data Register 0
0x0045	FDATA1	Flash Data Register 1

Table 9.1 FMC register table

Flash Clock Divider Register (FCKDIV)

FCKDIV	7	6	5	4	3	2	1	0
Bit Symbol	FCKDIV[7:0]							
Read/Write	R/W							
After reset	0x0F							

Note 1：Flash clock= fsysclk/ (FCKDIV +1)

Note 2：This register is reset by all resets.

Flash Control Register 0 (FCR0)

FCR0	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	DATASZ0[1:0]		FCMD[3:0]			
Read/Write	R	R	R/W		R/W			
After reset	0	0	0		0			

Note 1： Bits [6:0] is reset by all resets.

Note 2：Bits [7] is reset by POR

Note 3：Reserved bits must be written with zeros for future compatibility.

DATASZ [1:0]	Flash data size	00：Byte Others：Reserved
FCMD [3:0]	Flash command register	FCMD[1:0] 00：Read command 01：Write command 10：Page erase command 11：Mass erase command FCMD[3:2] must be written by 00 Note：These register bits are cleared by hardware after the operation is complete.

Flash Control Register 1 (FCR1)

FCR1	7	6	5	4	3	2	1	0
Bit Symbol	BUSY	PROGA_ ERR	PROGD_ ERR	reserved	reserved			
Read/Write	R	R	R	R/W	R			
After reset	0	0	0	0	0			

Note 1：This register is reset by all resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

BUSY	Flash BUSY indication	0：Idle 1：Busy, a flash operation is in progress Note：A transition from 1 to 0 will generate an interrupt.
PROGA_ERR	Flash address error	0：address correct 1：address incorrect
PROGD_ERR	Flash data error	0：data correct 1：data incorrect

Flash Address Register 0 (FADDR0)

FADDR0	7	6	5	4	3	2	1	0
Bit Symbol	FADDR[7:0]							
Read/Write	R/W							
After reset	0							

Note：This register is reset by all resets.

FADDR [7:0]	Flash Address bits [7:0]

Flash Address Register1 (FADDR1)

FADDR1	7	6	5	4	3	2	1	0
Bit Symbol	FADDR[15:8]							
Read/Write	R/W							
After reset	0							

Note : This register is reset by all resets

FADDR [15:8]	Flash address bits [15:8]

Flash Data Register 0 (FDATA0)

FDATA0	7	6	5	4	3	2	1	0
Bit Symbol	FDATA[7:0]							
Read/Write	R/W							
After reset	0							

Note : This register is reset by all resets

FDATA [7:0]	Flash data bits [7:0]

Flash Data Register 1 (FDATA1)

FDATA1	7	6	5	4	3	2	1	0
Bit Symbol	FDATA[15:8]							
Read/Write	R/W							
After reset	0							

Note : This register is reset by all resets

FDATA [15:8]	Flash data bits [15:8]

10. IO Ports

SQ7613 has 6 parallel input / output ports:

Port Name	Pin Name	No. of Pins	Input/output	Secondary Functions	
PortP0	P0.0 P0.1 P0.2 P0.4 P0.5 P0.6	6	Input / Output	P0.0	UART,SIO,I2C,TCA,ISP_Rx function.
				P0.1	UART,SIO,TCA,ISP_Tx function.
				P0.2 P0.6	SIO,I2C,TCA function.
				P0.4 P0.5	UART,SIO,I2C,TCA function.
PortP1	P1.0 to P1.3	4	Input / Output	P1.0 to P1.3	ADC input
PortP2	P2.0 P2.1 P2.2 P2.4	4	Input / Output	P2.0	SIO and the external interrupt input
				P2.1	SIO,I2C,and the external interrupt function
				P2.2	SIO,I2C,and the external interrupt input
				P2.4	the key-on wakeup input.
PortP3	P3.1 to P3.7	7	Input / Output	P3.1	-
				P3.2 toP3.3	I2C function
				P3.4	DBG input
				P3.5	TCA and the DBG input
				P3.6	UART,TCA and the external interrupt input
				P3.7	UART and TCA function
PortP4	P4. 2 to P4.5,P4.7	5	Input / Output	P4.2	-
				P4.3	External interrupt function
				P4.4 P4.5	External high speed reference clock connection.
				P4.7	TCA , Divider output and the external interrupt input
PortP5	P5.1 P5.2 P5.3	3	Input / Output	P5.1	ADC input ,UART and TCA input.
				P5.2	ADC input ,UART and TCA output.
				P5.3	ADC input and ADC reference voltage input

TABLE 10- 1 LIST OF I/O PORTS

Pin Name					Pin Options	
					Key-on Wakeup	External Interrupt
P0.0	P1.0	P2.0			KWI 0	EINT0
P0.1	P1.1	P2.1	P3.1		KWI 1	EINT1
P0.2	P1.2	P2.2	P3.2		KWI 2	EINT2
-	P1.3	-	P3.3 P3.6		KWI 3	EINT3
P0.4		P2.4	P3.4		KWI 4	EINT4
P0.5			P3.5		KWI 5	EINT5
P0.6			-	-	KWI 6	EINT6
-		-	P3.7	P4.7	KWI 7	EINT7
				P4.2		EINT2
				P4.3		EINT3
				P4.4		EINT4
				P4.5		EINT5

TABLE 10-2 I/O AND KEY-ON WAKE UP AND EXTERNAL INTERUPT TABLE

10.1 IO Port Control Register

The following control registers are used for input and output I/O. x indicates the port number. The register can be set or not related to the port. Refer to the instructions for each item.

PxDO Register

This is the register for setting output data. When a port is set to the "output mode", the value specified at PxDO is output from the port.

PxDI 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 PxDI.

PxOE Register

This register switches a port between input and output. A port can be switched between the "input mode" and the "output mode".

PxFC1、PxFC2 Register

This register enables the function output of each port. The 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-low resistor is connected when a port is used in the input mode.

Note：If the system needs to read the GPIO status immediately after entering the external interrupt, please add NOP in the program to avoid reading error. Please refer to Appendix D.

10.2 IO Port Register

Function	Address	Byte 3	Byte 2	Byte 1	Byte 0
GPIO DO	0x00E0	P3DO	P2DO	P1DO	P0DO
	0x00E4	Reserved		P5DO	P4DO
GPIO DI	0x00F0	P3DI	P2DI	P1DI	P0DI
	0x00F4	Reserved		P5DI	P4DI
GPIO OE	0x0100	P3OE	P2OE	P1OE	P0OE
	0x0104	Reserved		P5OE	P4OE
GPIO PU	0x0110	P3PU	P2PU	P1PU	P0PU
	0x0114	Reserved		P5PU	P4PU
GPIO PD	0x0120	P3PD	P2PD	P1PD	P0PD
	0x0124	Reserved		P5PD	P4PD
GPIO FC1	0x0140	P3FC1	P2FC1	P1FC1	P0FC1
	0x0144	Reserved		P5FC1	P4FC1
GPIO FC2	0x0150	P3FC2	P2FC2	P1FC2	P0FC2
	0x0154	Reserved		P5FC2	P4FC2

10.2.1 Port P0 Register

Port P0 Output Latch Register (P0DO)

P0DO		7	6	5	4	3	2	1	0
Bit Symbol		reserved	P6	P5	P4	reserved	P2	P1	P0
Read/Write		R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	Outputs L level when the output mode is selected							
	1	Outputs H level when the output mode is selected							

Port P0 Input Data Register (P0DI)

P0DI	7	6	5	4	3	2	1	0
Bit Symbol	reserved	P6	P5	P4	reserved	P2	P1	P0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	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.							

Port P0 Input / Output Control Register (P0OE)

P0OE		7	6	5	4	3	2	1	0
Bit Symbol		reserved	P6	P5	P4	reserved	P2	P1	P0
Read/Write		R	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	Input mode (port input)							
	1	Output mode (port output)							

Port P0 Built-in Pull-up Resistor Control Register (P0PU)

P0PU		7	6	5	4	3	2	1	0
Bit Symbol		reserved	P6	P5	P4	reserved	P2	P1	P0
Read/Write		R	R/W	R/W	R/W	R	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 built-in resistor is not connected.. This resistor is connected in the input mode only. Under any other conditions,setting to "1" does not make resistor connected.							

Port P0 Built-in Pull-low Resistor Control Resistor (P0PD)

P0PD		7	6	5	4	3	2	1	0
Bit Symbol		reversed	P6	P5	P4	reversed	P2	P1	P0
Read/Write		R	R/W	R/W	R/W	R	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	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Note: If PxPUx and PxPDx are both set to "1", the port will only be connected to the pull-up resistor. (x = 0, 1)

Port P0 Function Control Register 1、2 (P0FC1、P0FC2)

P0FC1 P0FC2	7	6	5	4	3	2	1	0
Bit Symbol	reserved	P6	P5	P4	reserved	P2	P1	P0
Read/Write	R	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
FUNCTION P0FC1=0,P0FC2=0	-	Port function	Port function	Port function	-	Port function	Port function	Port function
P0FC1=0,P0FC2=1	-	-	-	-	-	-	-	-
P0FC1=1,P0FC2=0	-	TCA6/ SCK0/SCL0	TXD2/RXD2/ SI0/SDA0/ SCL0/TCA5	RXD2/TXD2/ SO0/TCA4/ SDA0	-	TCA2/SCK1/S DA1/TCA2	TXD0/ RXD0/SO1/ TCA1	RXD0/TXD0/ SI1/SCL1/ TCA0
P0FC1=1,P0FC2=1	-	-	-	-	-	-	-	-

10.2.2 Port P1 Register

Port P1 Output Latch Register (P1DO)

P1DO		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	P0
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 P1 Input Data Register (P1DI)

P1DI	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	P3	P2	P1	P0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	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.							

Port P1 Input / Output Control Register (P1OE)

P1OE		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	P0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	Input mode (port input)							
	1	Output mode (port output)							

Port P1 Built-in Pull-up Resistor Control Resistor (P1PU)

P1PU		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	P0
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	The built-in resistor is not connected.							
	1	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Port P1 Built-in Pull-low Resistor Control Resistor (P1PD)

P1PD		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	P0
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	The built-in resistor is not connected.							
	1	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Note: If PxPUx and PxPDx are both set to "1", the port will only be connected to the pull-up resistor. (x = 0, 1)

Port P1 Function Control Register 1、2 (P1FC1、P1FC2)

P1FC1 P1FC2	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	P3	P2	P1	P0
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 P1FC1=0,P1FC2=0	-	-	-	-	Port function	Port function	Port function	Port function
P1FC1=0,P1FC2=1	-	-	-	-	-	-	-	-
P1FC1=1,P1FC2=0	-	-	-	-	-	-	-	-
P1FC1=1,P1FC2=1	-	-	-	-	AIN4	AIN5	AIN6	AIN7

10.2.3 Port P2 Register

Port P2 Output Latch Register (P2DO)

P2DO		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	P4	reserved	P2	P1	P0
Read/Write		R	R	R	R/W	R	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 P2 Input Data Register (P2DI)

P2DI	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	P4	reserved	P2	P1	P0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	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.							

Port P2 Input / Output Control Register (P2OE)

P2OE		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	P4	reserved	P2	P1	P0
Read/Write		R/W	R	R	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	Input mode (port input)							
	1	Output mode (port output)							

Port P2 Built-in Pull-up Resistor Control Resistor (P2PU)

P2PU		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	P4	reserved	P2	P1	P0
Read/Write		R	R	R	R/W	R	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	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Port P2 Built-in Pull-low Resistor Control Resistor (P2PD)

P2PD		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	P4	reserved	P2	P1	P0
Read/Write		R	R	R	R/W	R	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	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Note: If PxPUx and PxPDx are both set to "1", the port will only be connected to the pull-up resistor. (x = 0, 1)

Port P2 Function Control Register 1、2 (P2FC1、P2FC2)

P2FC1 P2FC2	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	P4	reserved	P2	P1	P0
Read/Write	R	R	R	R/W	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
FUNCTION P2FC1=0,P2FC2=0	-	-	-	Port function	-	Port function	Port function	Port function
P2FC1=0,P2FC2=1	-	-	-	-	-	-	-	-
P2FC1=1,P2FC2=0	-	-	-	TCA4	-	SCL1/SCK1/ TCA2	SDA1/SI1/ TCA1	TCA0/SO1
P2FC1=1,P2FC2=1	-	-	-	-	-	-	-	-

10.2.4 Port P3 Register

Port P3 Output Latch Register (P3DO)

P3DO		7	6	5	4	3	2	1	0
Bit Symbol		P7	P6	P5	P4	P3	P2	P1	reserved
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	0	Outputs L level when the output mode is selected							
	1	Outputs H level when the output mode is selected							

Port P3 Input Data Registe (P3DI)

P3DI	7	6	5	4	3	2	1	0
Bit Symbol	P7	P6	P5	P4	P3	P2	P1	reserved
Read/Write	R	R	R	R	R	R	R	R
After reset	0	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.							

Port P3 Input / Output Control Register (P3OE)

P3OE		7	6	5	4	3	2	1	0
Bit Symbol		P7	P6	P5	P4	P3	P2	P1	reserved
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	0	Input mode (port input)							
	1	Output mode (port output)							

Port P3 Built-in Pull-up Resistor Control Resistor (P3PU)

P3PU		7	6	5	4	3	2	1	0
Bit Symbol		P7	P6	P5	P4	P3	P2	P1	reserved
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	0	The built-in resistor is not connected.							
	1	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Port P3 Built-in Pull-low Resistor Control Resistor (P3PD)

P3PD		7	6	5	4	3	2	1	0
Bit Symbol		P7	P6	P5	P4	P3	P2	P1	reserved
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	0	The built-in resistor is not connected.							
	1	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Note: If PxPUx and PxPDx are both set to "1", the port will only be connected to the pull-up resistor. (x = 0, 1)

Port P3 Function Control Register 1、2 (P3FC1、P3FC2)

P3FC1 P3FC2	7	6	5	4	3	2	1	0
Bit Symbol	P7	P6	P5	P4	P3	P2	P1	reserved
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 P3FC1=0,P3FC2=0	Port function	Port function	Port function	Port function	Port function	Port function	Port function	-
P3FC1=0,P3FC2=1	-	-	-	-	-	-	-	-
P3FC1=1,P3FC2=0	TXD0/RXD0 /TCA7	RXD0/TXD0 /TCA3	TCA5	-	SCL0	SDA0	-	-
P3FC1=1,P3FC2=1	-	-	-	-	-	-	-	-

10.2.5 Port P4 Register

Port P4 Output Latch Register (P4DO)

P4DO		7	6	5	4	3	2	1	0
Bit Symbol		P7	reserved	P5	P4	P3	P2	reserved	reserved
Read/Write		R/W	R	R/W	R/W	R/W	R/W	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 P4 Input Data Register (P4DI)

P4DI	7	6	5	4	3	2	1	0
Bit Symbol	P7	reserved	P5	P4	P3	P2	reserved	reserved
Read/Write	R	R	R	R	R	R	R	R
After reset	0	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.							

Port P4 Input / Output Control Register (P4OE)

P4OE		7	6	5	4	3	2	1	0
Bit Symbol		P7	reserved	P5	P4	P3	P2	reserved	reserved
Read/Write		R/W	R	R/W	R/W	R/W	R/W	R	R
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	Input mode (port input)							
	1	Output mode (port output)							

Port P4 Built-in Pull-low Resistor Control Resistor (P4PU)

P4PU		7	6	5	4	3	2	1	0
Bit Symbol		P7	reserved	P5	P4	P3	P2	reserved	reserved
Read/Write		R/W	R	R/W	R/W	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	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Port P4 Built-in Pull-low Resistor Control Resistor (P4PD)

P4PD		7	6	5	4	3	2	1	0
Bit Symbol		P7	reserved	P5	P4	P3	P2	reserved	reserved
Read/Write		R/W	R	R/W	R/W	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	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Note: If PxPUx and PxPDx are both set to "1", the port will only be connected to the pull-up resistor. (x = 0,1)

Port P4 Function Control Register 1、2 (P4FC1、P4FC2)

P4FC1 P4FC2	7	6	5	4	3	2	1	0
Bit Symbol	P7	reserved	P5	P4	P3	P2	reserved	reserved
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R	R
After reset	0	0	0	0	0	0	0	0
FUNCTION P4FC1=0,P4FC2=0	Port function	-	Port function	Port function	Port function	Port function	-	-
P4FC1=0,P4FC2=1	DVO	-	-	-	-	-	-	-
P4FC1=1,P4FC2=0	TCA7	-	-	-	-	-	-	-
P4FC1=1,P4FC2=1	-	-	XOUT	XIN	-	-	-	-

10.2.6 Port P5 Register

Port P5 Output Latch Register (P5DO)

P5DO		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	reserved
Read/Write		R	R	R	R	R/W	R/W	R/W	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 P5 Input Data Register (P5DI)

P5DI	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	P3	P2	P1	reserved
Read/Write	R	R	R	R	R	R	R	R
After reset	0	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.							

Port P2 Input / Output Control Register (P5OE)

P5OE		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	reserved
Read/Write		R	R	R	R	R/W	R/W	R/W	R
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	Input mode (port input)							
	1	Output mode (port output)							

Port P5 Built-in Pull-up Resistor Control Resistor (P5PU)

P5PU		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	reserved
Read/Write		R	R	R	R	R/W	R/W	R/W	R
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	The built-in resistor is not connected.							
	1	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Port P5 Built-in Pull-low Resistor Control Resistor (P5PD)

P5PD		7	6	5	4	3	2	1	0
Bit Symbol		reserved	reserved	reserved	reserved	P3	P2	P1	reserved
Read/Write		R/W	R/W	R/W	R	R/W	R/W	R/W	R
After reset		0	0	0	0	0	0	0	0
FUNCTION	0	The built-in resistor is not connected.							
	1	Connect the built-in resistor. This resistor is only connected in input mode. Setting to "1" under other conditions will not connect this built-in resistor.							

Note: If PxPUx and PxPDx are both set to "1", the port will only be connected to the pull-up resistor. (x = 0, 1)

Port P5 Function Control Register 1、2 (P5FC1、P5FC2)

P5FC1 P5FC2	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	P3	P2	P1	reserved
Read/Write	R	R	R	R	R/W	R/W	R/W	R
After reset	0	0	0	0	0	0	0	0
FUNCTION P5FC1=0,P5FC2=0	-	-	-	-	Port function	Port function	Port function	-
P5FC1=0,P5FC2=1	-	-	-	-	-	-	-	-
P5FC1=1,P5FC2=0	-	-	-	-	-	TXD1/RXD1	RXD1/TXD1	-
P5FC1=1,P5FC2=1	-	-	-	-	AIN8 VREF	AIN9	AIN10	-

11. Multiplier (MAC)

The multiplier supports the following functions:

- 32-bit addition
- 32-bit subtraction
- 16x16-bit multiplication
- 32-bit unsigned division
- 40-bit accumulator
- Multiply and add
- Multiply and subtract
- Quick operand zeroization

11.1 Operation

The multiplier operation is determined its mode setting (MACCR0<MODE>). The multiplier supports the following operation：

MODE	OPERATION
0000	C = A + B
0001	C = A - B
0010	C = A x B
0100	C = C + A + B
0101	C = C - (A + B)
0110	C = C + A x B
0111	C = C – A x B
1000	C=A/B

TABLE 11- 1 MULTIPLIER OPERATION

The multiplier consists of two operands (A, B) and one accumulator (C). MAC A Register (MACA) and MAC B Register (MACB) are 32-bit registers. During multiplication (AxB) when used as multiplier, MACA and MACB are limited to 16-bits and only the lower 16-bits are valid (MACA0, MACA1, MACB0, MACB1). If used as a parameter for addition or subtraction operation, MACA and MACB are used as 32-bit registers. MAC C Register (MACC) is a 40-bit register.

Each of the registers can be cleared to 0 by setting the corresponding clear bit to 1 (MACCR1<CLEARA>, MACCR1<CLEARB>, MACCR1<CLEARC>). These clear bits are automatically cleared to 0 when done or when calculation starts.

The multiplier can carry out both signed and unsigned operation. The MACCR0<SIGN > selects the operation. When SIGN=0, the operand is unsigned. When SIGN = 1, the operand is signed or 2's complement.

MACC	SIGN=0	SIGN=1
Maximum Value	0xFF_FFFF_FFFF	0x7F_FFFF_FFFF
Minimum Value	0x00_0000_0000	0x80_0000_0000

TABLE 11- 2 MULTIPLIER ACCUMULATOR MAXIMUM AND MINIMUM VALUES

The multiplier is started by setting MACCR1<START> to 1. This bit is self-cleared to 0 when the operation is completed. If interrupt is enabled (MACCR0<INTEN>=1), this will set the interrupt flag (MACCR1<INTF>) to 1 and generate an interrupt.

For unsigned operation (SIGN=0), the carry flag (MACCR1<CF>) is set to 1 when the accumulator results in a carry out of or borrow form the most significant bit, such as when (0xFF_FFFF_FFFF +1) or (0-1). User application can ignore the overflow flag in unsigned operation.

For signed operation (SIGN=1), the overflow flag (MACCR1<OF>) is set to 1 when

1. Addition of two positive number yields a negative result
(0x7F_FFFF_FFFF + 0x7F_FFFF_FFFF) or
2. Addition of two negative number yields a positive result
(0x80_0000_0000 + 0x80_0000_0000)

User application can ignore the carry flag in signed operation.

The overflow and carry flag are updated upon multiplier operation completion.

The saturation mode bit controls how the accumulator MACC will react in overflow or underflow situation. When saturation mode is disabled (MACCR0<SAT>=0), MACC will rollover when overflow or carry occurs. When saturation mode is enabled (MACCR0<SAT>=1), MACC will be capped at its maximum value or minimum value depending on the SIGN operation.

Regardless of saturation mode setting, the overflow and carry flag will still be set if the conditions described above are met.

Example 1: C – A – B
00_1234_5678 – FFFF_FFFF – FFFF_FFFC

SIGN	SAT	RESULT	CF	OF
0	0	FE_1234_567D	1	N/A
0	1	00_0000_0000	1	N/A
1	0	00_1234_567D	N/A	0
1	1	00_1234_567D	N/A	0

Example 2: C + A + B
7F_FFEE_DDCC + 1234_5678 +4433_2211

SIGN	SAT	RESULT	CF	OF
0	0	80_5656_5655	0	N/A
0	1	80_5656_5655	0	N/A
1	0	80_5656_5655	N/A	1
1	1	7F_FFFF_FFFF	N/A	1

n division mode,

Quotient = Dividend / Divisor

Where:

Dividend = MACA

Divisor = MACB

Quotient = MACC

Remainder = MACA

Sign (MACCR0.SIGN), Saturation (MACCR0.SAT), Carry Flag (MACCR1.CF) and Overflow Flag (MACCR1.OF) are not applicable in division and therefore are unaffected by the operation.

Example 3：C=A / B

MACA (Dividend)	MACB (Divisor)	MACC (Quotient)	MACA (Remainder)	DIVERR
44B1_7E22	0000_0045	00_00FE_DCBA	0000_0000	0
0000_1234	0000_5678	00_0000_0000	0000_1234	0
FEDC_BA98	0123_4567	00_0000_00E0	0000_0078	0
0000_0000	FFFF_1234	00_0000_0000	0000_0000	0
44B1_7E22	0000_0000	00_0000_0000	44B1_7E22	1

11.2 Multiplier Registers

ADDRESS	REGISTER	DESCRIPTION
0x0870	MACCR0	MAC Control Register0
0x0871	MACCR1	MAC Control Register1
0x0872 | 0x0873	Reserved	
0x0874	MACA0	MAC ARegister0 [7:0]
0x0875	MACA1	MAC ARegister1 [15:8]
0x0876	MACA2	MAC ARegister2 [23:16]
0x0877	MACA3	MAC ARegister3 [31:24]
0x0878	MACB0	MAC BRegister0 [7:0]
0x0879	MACB1	MAC BRegister1 [15:8]
0x087A	MACB2	MAC BRegister2 [23:16]
0x087B	MACB3	MAC BRegister3 [31:24]
0x087C	MACC0	MAC CRegister0 [7:0]
0x087D	MACC1	MAC CRegister1 [15:8]
0x087E	MACC2	MAC CRegister2 [23:16]
0x087F	MACC3	MAC CRegister3 [31:24]
0x0880	MACC4	MAC CRegister4 [39:32]

TABLE 11- 3 MULTIPLIER REGISTERS

MAC Control Register 0 (MACCR0)

MACCR0	7	6	5	4	3	2	1	0
Bit Symbol	MODE[3:0]				reserved	SIGN	SAT	INTEN
Read/Write	R/W				R	R/W	R/W	R/W
After reset	0				0	0	0	0

Note：Reserved bits must be written with zeros for future compatibility.

MODE[3:0]	Mode selection	0000 : C = A + B 0001 : C = A - B 0010 : C = A x B 0011 : Reserved 0100 : C = C + A + B 0101 : C = C - (A + B) 0110 : C = C + A x B 0111 : C = C – A x B 1000 : C =A / B, A = A % B Others：Reserved
SIGN	Sign Operation	0：unsigned 1：signed
SAT	Saturation mode	0：disable 1：enable
INTEN	Interrupt enable	0：disable 1：enable

MAC Control Register 1(MACCR1)

MACCR1	7	6	5	4	3	2	1	0
Bit Symbol	DIVERR	INTF	OF	CF	CLEARC	CLEARB	CLEARA	START
Read/Write	R	R/W	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note：Reserved bits must be written with zeros for future compatibility.

DIVERR	Division Error	0：No interrupt 1：Division Error This bit is cleared to 0 by writing "1" to it. Writing 0 to this bit is ignored.
INTF	Interrupt Flag	0 : No interrupt 1 : Interrupt pending This bit is cleared to 0 by writing "1" to it. Writing 0 to this bit is ignored.
OF	Overflow Flag	0：No overflow 1：Overflow
CF	Carry Flag	0：No carry 1：Carry
CLEARC	Clear MACC Register	0: No action. 1: Clear the register. This bit is automatically clear to 0 by hardware when operation completes or when calculation start.
CLEARB	Clear MACB Register	0: No action. 1: Clear the register. This bit is automatically clear to 0 by hardware when operation completes or when calculation start.
CLEARA	Clear MACA Register	0: No action. 1: Clear the register. This bit is automatically clear to 0 by hardware when operation completes or when calculation start.
START	Start calculation	0: Idle 1: Start MAC operation This bit is automatically clear to 0 by hardware when operation completes.

MAC A Register 0 (MACA0)

MACA0	7	6	5	4	3	2	1	0
Bit Symbol	MACA[7:0]							
Read/Write	R/W							
After reset	0							

MACA[7:0]	MAC A Register[7:0]

Register	Description
MACA0	MAC ARegister0 [7:0]
MACA1	MAC ARegister1 [15:8]
MACA2	MAC ARegister2 [23:16]
MACA3	MAC ARegister3 [31:24]

The MAC A register can be up to 32 bits, and its corresponding register is listed above. The address can be referred to the table "11.3 MAC Register List".

MAC B Register0 (MACB0)

MACB0	7	6	5	4	3	2	1	0
Bit Symbol	MACB[7:0]							
Read/Write	R/W							
After reset	0							

MACB[7:0]	MAC B Register[7:0]

Register	Description
MACB0	MAC B Register0 [7:0]
MACB1	MAC B Register1 [15:8]
MACB2	MAC B Register2 [23:16]
MACB3	MAC B Register3 [31:24]

The MAC B register can be up to 32 bits, and its corresponding register is listed above.The address can be referred to the table "11.3 MAC Register List".

MAC C Register0 (MACC0)

MACC0	7	6	5	4	3	2	1	0
Bit Symbol	MACC[7:0]							
Read/Write	R/W							
After reset	0							

MACC[7:0]	MAC C Register[7:0]

Register	Description
MACC0	MAC CRegister0 [7:0]
MACC1	MAC CRegister1 [15:8]
MACC2	MAC CRegister2 [23:16]
MACC3	MAC CRegister3 [31:24]
MACC4	MAC CRegister4 [39:32]

The MAC C register can be up to 40 bits, and its corresponding register is listed above. The address can be referred to the table "11.3 MAC Register List".

12 Pheripheral Network Inter-Connect (PNIC)

12.1 Function

Peripheral Network Inter-Connect (PNIC) is a configurable connection matrix. The function of each IO pin can be set by the control function selection register and the peripheral channel selection register.

Each function and each channel has a dedicated programmable register group, which can specify the functions required for IO execution. The PNIC architecture enables the functions within each cluster to operate in parallel, increasing the flexibility of the system and supporting a wider range of applications.

12.2 Operation flowchart

The figure below briefly explains the operation flow of PNIC, and points out the method of function selection for reference.

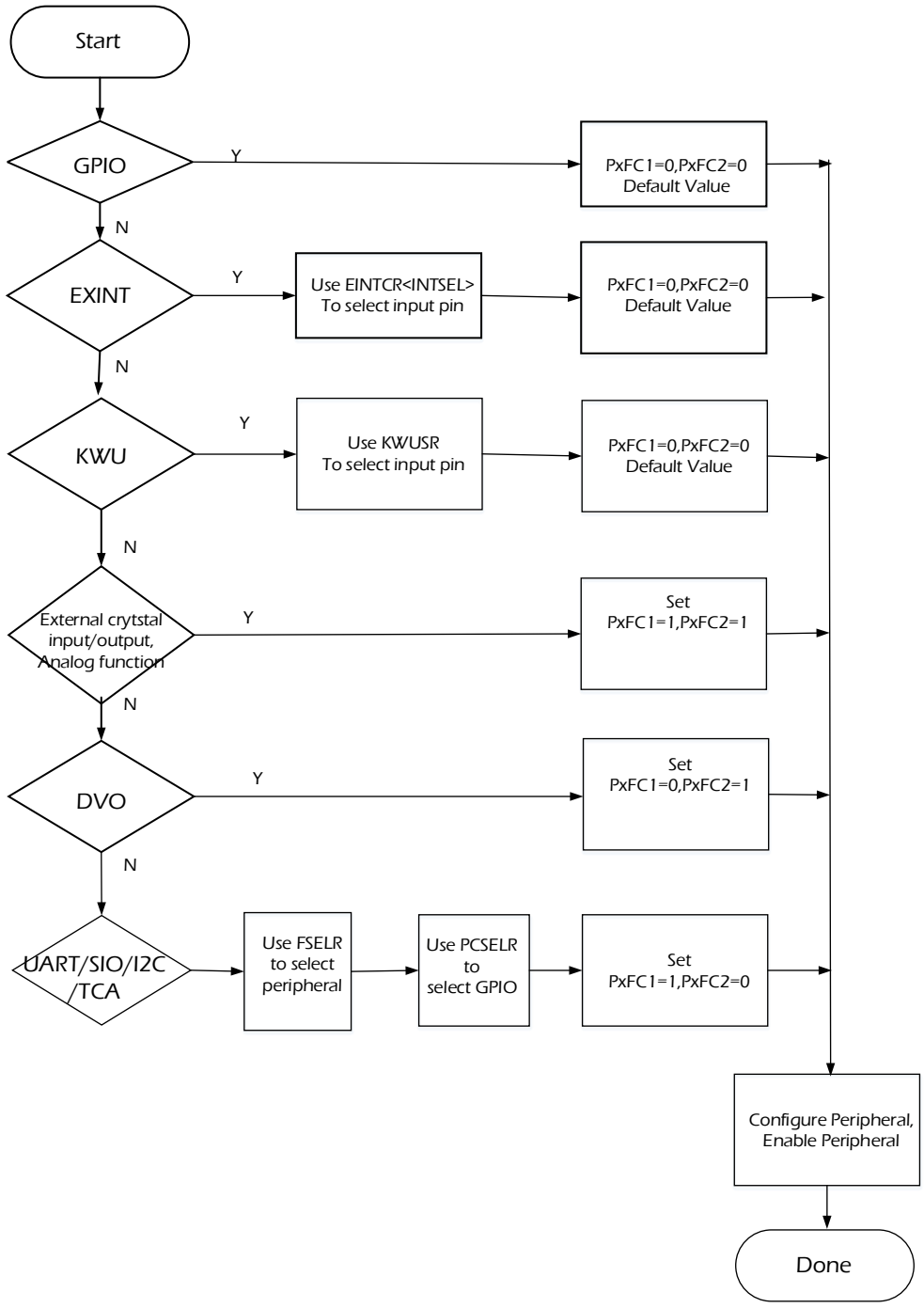


FIGURE 12- 1 PNIC OPERATION FLOWCHART

Note 1：When setting the peripheral functions, set in the following order: FSELR, then PCSELR, and then PxFC1 and PxFC2.

Note 2：For the contents of PxFC1 and PxFC2 registers, please refer to "10.2 I/ O Port Registers".

12.3 Control

There are six function select registers ,FSELR0 to FSELR 5. Each register is used to configure a paris of peripheral channels . In addition, there are six channel select registers, PCSELR0 to PCSELR5. These registers are used to select an active channel that is currently communicating with the corresponding peripheral.The FSELR and PCSELR registers are detailed in the following tables.

FSELR Register	FSELRx[6:4]	FSELRx[2:0]
FSELR0	P0[7:4]	P0[3:0]
FSELR1	P1[7:4]	P1[3:0]
FSELR2	P2[7:4]	P2[3:0]
FSELR3	P3[7:4]	P3[3:0]
FSELR4	P4[7:4]	P4[3:0]
FSELR5	P5[7:4]	P5[3:0]

TABLE12- 1 FSELR REGISTER AND CORRESPONDING PERIPHERAL CHANNEL

FSELRx[2:0] or FSELRx[6:4]	PxL (Px[0]、Px[1]、Px[2]、Px[3]) or PxH (Px[4]、Px[5]、Px[6]、Px[7])
0b000	UART, I2C, TCA
0b001	UART, I2C, TCA
0b010	I2C, SIO
0b011	UART, I2C, TCA
0b100	I2C, TCA
0b101	TCA

TABLE 12- 2 FSELR FUNCTION SELECT

The PCSELR register uses two bits as a unit to select the peripheral channel. For example, the peripheral function UART0 is connected to four channels P0.0, P3.6, P0.1 and P3.7. If FSELR [2: 0] = 000b, the system defaults PCSELR0 [1: 0] to 2'b00. Indicates that UART0 uses P0.0 for data transmission. PCSELR0 [1: 0] can also be programmed as 2'b01, which means using P3.6 channel transmission. The following table is a complete description of each PCSELR description.

Register	Bits 7:6	Bits 5:4	Bits 3:2	Bits1:0
PCSELR 0	reserved	UART2	UART1	UART0
PCSELR 1	reserved	reserved	I2C1	I2C0
PCSELR 2	reserved	reserved	SIO1	SIO0
PCSELR 3	reserved	reserved	reserved	reserved
PCSELR 4	TCA3	TCA2	TCA1	TCA0
PCSELR 5	TCA7	TCA6	TCA5	TCA4

TABLE 12- 3 PCSELR BYTE AND CORRESPONDING PERIPHERAL FUNCTION

There are 4 types of port pins for setting, as the following table. When the pin has no setting function (PxFC1 = 0, PxFC2 = 0), as general-purpose IO, the input and output of the port are set through the corresponding PxOE (input and output control register); when the pin has a setting function (not PxFC1 = 0 , PxFC2 = 0), the input / output of the port is completely controlled by the function set by PxFC, regardless of the setting of PxOE.

	Pin function
PxFC1=0, PxFC2=0	GPIO function
PxFC1=0, PxFC2=1	Peripheral function： DVO
PxFC1=1, PxFC2=0	Peripheral function：UART, SIO, I2C, TCA
PxFC1=1, PxFC2=1	External clock input / output, Analog function

Note 1: PxFC1.PxFC2 (X=0~5)
Note 2 : the pin is general-purpose IO (PxFC1 = 0, PxFC2 = 0), the input and output of the port is set through PxOE (input and output control register).
Note 3: For the contents of PxFC1, PxFC2, please refer to "10 General I / O Chapter"

Table 12- 4 PxFC register and function list

Address	Register	Description
0x016C	PCSELR0	Peripheral Channel Select Register 0
0x016D	PCSELR1	Peripheral Channel Select Register 1
0x016E	PCSELR2	Peripheral Channel Select Register 2
0x0170	PCSELR4	Peripheral Channel Select Register 4
0x0171	PCSELR5	Peripheral Channel Select Register 5
0x0190	EINTCR0	External Interrupt Control Register 0
0x0191	EINTCR1	External Interrupt Control Register 1
0x0192	EINTCR2	External Interrupt Control Register 2
0x0193	EINTCR3	External Interrupt Control Register 3
0x0194	EINTCR4	External Interrupt Control Register 4
0x0195	EINTCR5	External Interrupt Control Register 5
0x0196	EINTCR6	External Interrupt Control Register 6
0x0197	EINTCR7	External Interrupt Control Register 7
0x018C	KWUSR0	Key-on Wakeup Status Register 0
0x018D	KWUSR1	Key-on Wakeup Status Register 1

TABLE12- 5 PNICREGISTER TABLE

The list of PNIC control registers is shown in the table above.

Peripheral Channel Select Register 0(PCSELR0)

PCSELR0	7	6	5	4	3	2	1	0
Bit Symbol	reserved		UART2[1:0]		UART1[1:0]		UART0[1:0]	
Read/Write	R		R/W		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

UART2	FSELR[6:4] / [2:0]		UART1	FSELR[6:4] / [2:0]		UART0	FSELR[6:4] / [2:0]	
PCSELR0 [5:4]	0x000	0x001	PCSELR0 [3:2]	0x000	0x001	PCSELR0 [1:0]	0x000	0x001
00	P0.4/RXD2 P0.5/TXD2	P0.5/RXD2 P0.4/TXD2	00	-	-	00	P0.0/RXD0 P0.1/TXD0	P0.1/RXD0 P0.0/TXD0
01	-	-	01	P5.1/RXD1 P5.2/TXD1	P5.2/RXD1 P5.1/TXD1	01	P3.6/RXD0 P3.7/TXD0	P3.7/RXD0 P3.6/TXD0
10	-	-	10	-	-	10	-	-
11	-	-	11	-	-	11	-	-

Peripheral Channel Select Register 1(PCSELR1)

PCSELR1	7	6	5	4	3	2	1	0
Bit Symbol	reserved		reserved		I2C1[1:0]		I2C0[1:0]	
Read/Write	R		R		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

I2C1 PCSELR1 [3:2]	SCL1	SDA1	I2C0 PCSELR1 [1:0]	SCL0	SDA0
00	P0.2	P0.0	00	P0.6	P0.5
01	P2.2	P2.1	01	P0.5	P0.4
10	-	-	10	-	-
11	-	-	11	P3.3	P3.2

Peripheral Channel Select Register 2(PCSELR2)

PCSELR2	7	6	5	4	3	2	1	0
Bit Symbol	reserved		reserved		SIO1[1:0]		SIO0[1:0]	
Read/Write	R		R		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

SIO1 PCSELR2 [3:2]	SIO0_CLK	SIO0_SI	SIO0_SO	SIO0 PCSELR2 [1:0]	SIO0_CLK	SIO0_SI	SIO0_SO
00	P0.2	P0.0	P0.1	00	P0.6	P0.5	P0.4
01	P2.2	P2.1	P2.0	01	-	-	-
10	-	-		10	-	-	-
11	-	-		11	-	-	-

Peripheral Channel Select Register 4(PCSELR4)

PCSELR4	7	6	5	4	3	2	1	0
Bit Symbol	TCA3[1:0]		TCA2[1:0]		TCA1[1:0]		TCA0[1:0]	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

TCA	TCA3	TCA2	TCA1	TCA0
	PCSELR4[7:6]	PCSELR4[5:4]	PCSELR4[3:2]	PCSELR4[1:0]
00	P3.6	P0.2	P0.1	P0.0
01	-	P2.2	P2.1	P2.0
10	P5.1/TCA3_IN P5.2/TCA3_OUT	-	-	-
11	-	-	-	-

Peripheral Channel Select Register 5(PCSELR5)

PCSELR5	7	6	5	4	3	2	1	0
Bit Symbol	TCA7[1:0]		TCA6[1:0]		TCA5[1:0]		TCA4[1:0]	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

TCA	TCA7	TCA6	TCA5	TCA4
	PCSELR5[7:6]	PCSELR5[5:4]	PCSELR5[3:2]	PCSELR5[1:0]
00	P3.7	P0.6	P0.5	P0.4
01	P4.7	-	-	P2.4
10	-	-	P3.5	-
11	-	-	-	-

External Interrupt Control Register (EINTCRx, x=0 ~ 7)

EINTCR	7	6	5	4	3	2	1	0
Bit Symbol	INTSEL[2:0]			INTLVL	INTES[1:0]		INTINC[1:0]	
Read/Write	R/W			R	R/W		R/W	
After reset	0			0	0		0	

Note 1：This register is reset by all resets.

		EINTCRx [INTSEL]	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	EINT1	EINT0
			EINTCR7 [INTSEL]	EINTCR6 [INTSEL]	EINTCR5 [INTSEL]	EINTCR4 [INTSEL]	EINTCR3 [INTSEL]	EINTCR2 [INTSEL]	EINTCR1 [INTSEL]	EINTCR0 [INTSEL]
INTSEL[2:0]	External Interrupt pin select	000	P3.7	P0.6	P0.5	P0.4	P3.6	P0.2	P0.1	P0.0
		001	-	-	-	-	P1.3	P1.2	P1.1	P1.0
		010	P4.7	-	-	P2.4	-	P2.2	P2.1	P2.0
		011	-	-	P3.5	P3.4	P3.3	P3.2	P3.1	-
		100	-	-	P4.5	-	-	-	-	-

INTLVL	Noise canceller pass signal level when the interrupt request signal is generated	0：Initial state or signal level "L" 1：Signal level "H"
INTES[1:0]	Selects external interrupt request generating condition	00：At the rising edge of the noise canceller pass signal 01：At the falling edge of the noise canceller pass signal 10：At both edge of the noise canceller pass signal 11：Reserved
INTINC[1:0]	Sets external interrupt noise canceller sampling interval	00：fsysclk 01：fsysclk / 4 10：fsysclk / 8 11：fsysclk / 16

Key-on Wakeup Status Register 0(KWUSR0)

KWUSR0	7	6	5	4	3	2	1	0
Bit Symbol	KWU3		KWU2		KWU1		KWU0	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

KWU	KWU3	KWU2	KWU1	KWU0
	KWUSR0[7:6]	KWUSR0[5:4]	KWUSR0[3:2]	KWUSR0[1:0]
00	P3.6	P0.2	P0.1	P0.0
01	P1.3	P1.2	P1.1	P1.0
10	-	P2.2	P2.1	P2.0
11	P3.3	P3.2	P3.1	-

Key-on Wakeup Status Register 1(KWUSR1)

KWUSR1	7	6	5	4	3	2	1	0
Bit Symbol	KWU7		KWU6		KWU5		KWU4	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0		0		0		0	

Note 1：This register is reset by all resets.

KWU	KWU7	KWU6	KWU5	KWU4
	KWUSR1[7:6]	KWUSR1[5:4]	KWUSR1[3:2]	KWUSR1[1:0]
00	P3.7	P0.6	P0.5	P0.4
01	-	-	-	-
10	P4.7	-	-	P2.4
11	-	-	P3.5	P3.4

12.4 PNIC Diagram

The following figures show the PNIC operation diagram. One is the architecture of the peripheral transmission to the I/O, and the other one is the I/O transmission th the peripheral.

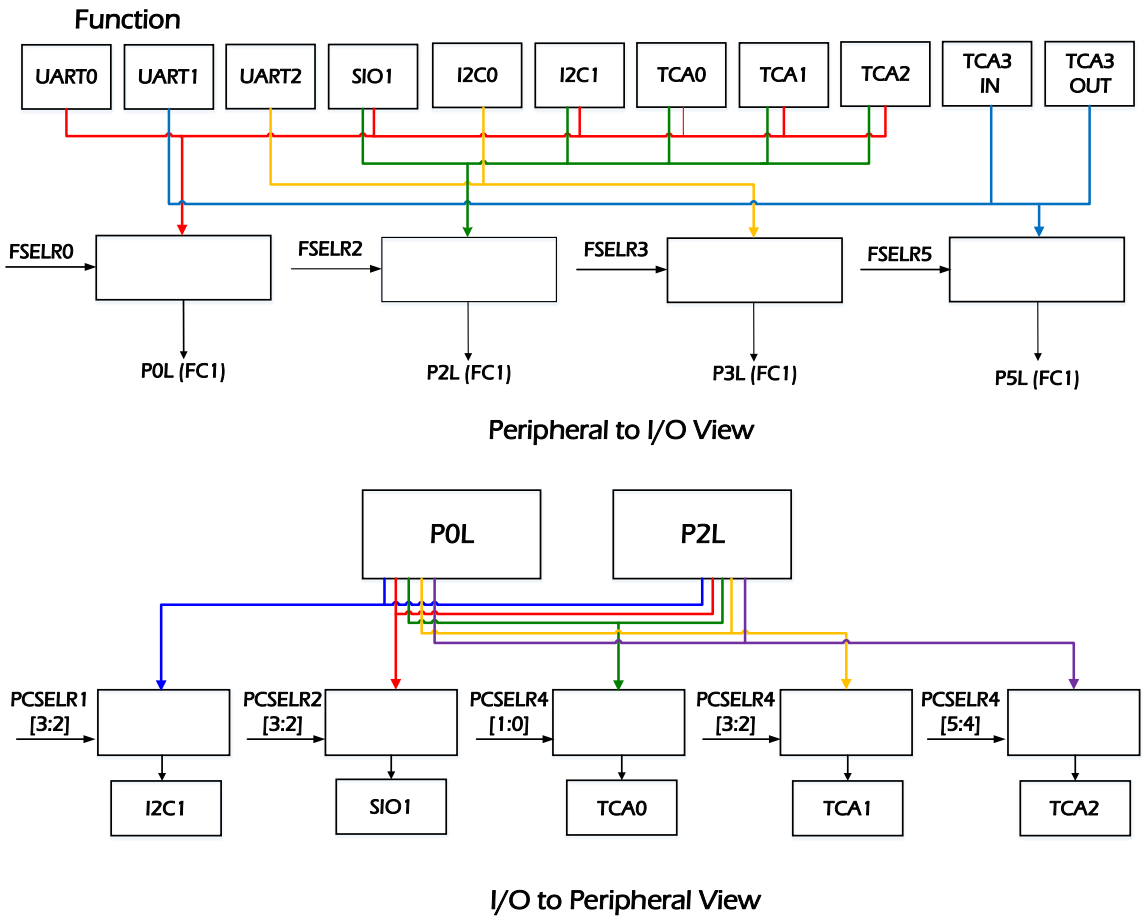


FIGURE 12- 2 PNIC STRUCTURE

GPIOs Low Bytes Px[3:0] and High Bytes Px[7:4] are present as PxL and PxH, which are grouped together to form a peripheral matrix. The Function Select Register (FSELR) determine how each group is configured. FSELR[3:0] determines how Px[3:0] is configured whereas FSELR[4:0] determine how Px[7:4] is configured. Valid combinations are UART + I2C, UART + Timer, etc. If a function in a combination is not required, the pins can still function as GPIO. For example, when FSELR0[2:0]=0000b, select P0L as UART0 and TCA2 combination. If TCA2 is not needed, P0.2 can be GPIO by setting PxFC1=0 and PxFC2=0.

Because the Peripheral Network Inter-Connect (PNIC) has high flexibility, the same peripheral can be supported by different I/Os. For example, P0L and P2L both can control SIO1. In this case, user sets PCSELRx to select P0L or P2L to select the SIO channel I/O.

13. Watchdog Timer (WDT)

13.1 Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to detect rapidly the CPU malfunctions such as endless loops due to spurious noises or the deadlock conditions, and return the CPU to a system recovery routine.

The watchdog timer signals used for detecting malfunctions can be programmed as watchdog interrupt request signals or watchdog timer reset signals.

Note 1：Care must be taken in system designing since the watchdog timer may not fulfill its functions due to disturbing noise and other effects.

Note 2：In Sleep mode, please use WDT INT/WDT RST other interrupt sources or reset wake-up.

13.1.1 Watchdog Timer Configuration

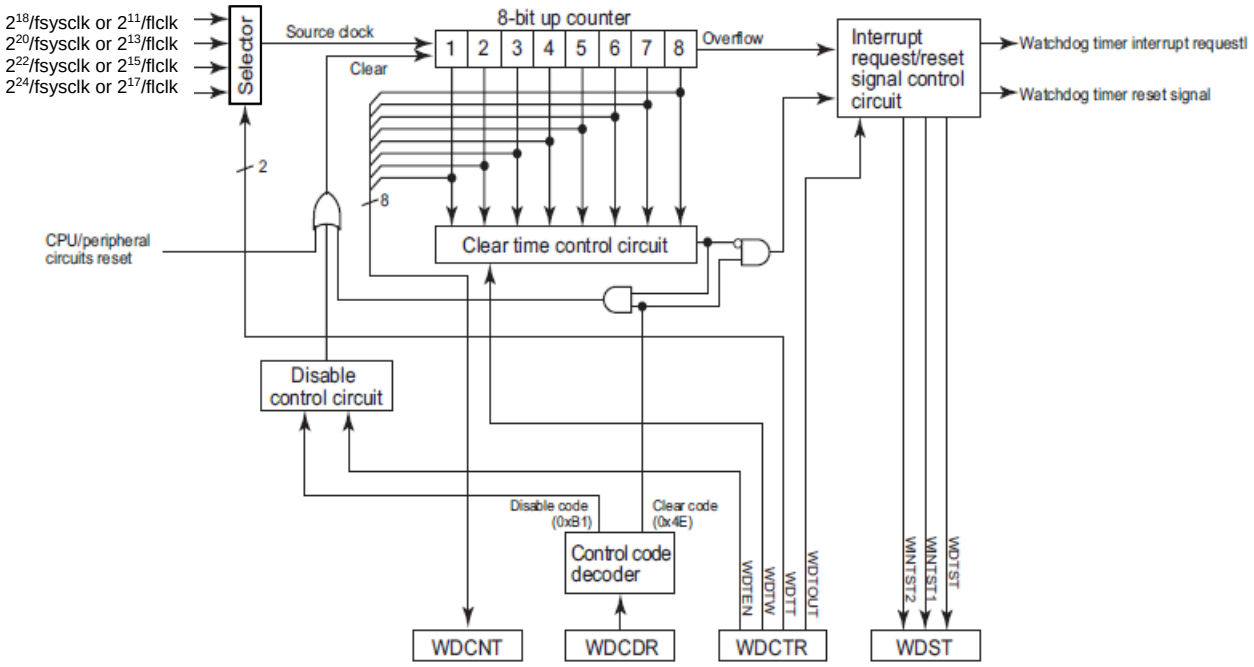


FIGURE 13- 1 WATCHDOG TIMER CONFIGURATION

13.1.2 Watchdog TimerC ontrol

The watchdog timer is controlled by the watchdog timer control register (WDCTR), the watchdog timer control code register (WDCDR), the watchdog timer counter monitor (WDCNT) and the watchdog timer status (WDST).

The watchdog timer is enabled automatically just after the warm-up operation that follows reset is finished.

Address	Register	Description
0x0028	WDCTR	Watchdog Timer Control Register
0x0029	WDCDR	Watchdog Timer Control Register
0x002A	WDCNT	8-bit Up Counter Monitor
0x002B	WDST	Watchdog Timer Status

Watchdog Timer Control Register (WDCTR)

WDCTR	7	6	5	4	3	2	1	0
Bit Symbol	-	-	WDTEN	WDTW[1:0]		WDTT[1:0]		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 watchdog timer	0: Disable 1: Enable
WDTW[1:0]	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. 01: A watchdog timer interrupt request is generated by writing the clear code at a point within the first quarter of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first quarter of the overflow time has elapsed. 10: A watchdog timer interrupt request is generated by writing the clear code at a point within the first half of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first half of the overflow time has elapsed. 11: A watchdog timer interrupt request is generated by writing the clear code at a point within the first three quarters of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first three quarters of the overflow time have elapsed.
WDTT[1:0]	Set the overflow time of the 8-bit up counter.	

	NORMAL mode (fsysclk=HIRC/PLL/HXTAL)		SLOW mode (fsysclk=LIRC)
	TBTCR<DV9CK>=0	TBTCR<DV9CK>=1	
00:	2^{18}/fsysclk	2^{11}/flclk	2^{11}/flclk
01:	2^{20}/fsysclk	2^{13}/flclk	2^{13}/flclk
10:	2^{22}/fsysclk	2^{15}/flclk	2^{15}/flclk
11:	2^{24}/fsysclk	2^{17}/flclk	2^{17}/flclk

WDTOUT	Select an overflow detection signal of the 8-bit up counter.	0 : Watchdog timer interrupt request signal 1 : Watchdog timer reset request signal

Note 1： fsysclk, 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)

WDCDR	7	6	5	4	3	2	1	0
Bit Symbol	WDTCR[7:0]							
Read/Write	W							
After reset	0							

WDTCR[7:0]	Write watchdog timer control codes.	0x4E: Clear the watchdog timer. (clear code) 0xB1: Disable the watchdog timer operation and clear the 8-bit up counter when WDCTR <WDTEN> is "0". (disable code) Others: Invalid

8-bit Up Counter Monitor (WDCNT)

WDCNT	7	6	5	4	3	2	1	0
Bit Symbol	WDCNT[7:0]							
Read/Write	R							
After reset	0							

WDCNT[7:0]	Monitor the count value of the 8-bit up counter.	The count value of the 8-bit up counter is read.

Watchdog Timer Status (WDST)

WDST	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	WINTST2	WINTST1	WDTST
Read/Write	R	R	R	R	R	R	R	R
After reset	0	1	0	1	1	0	0	1

WINTST2	Watchdog timer interrupt request signal factor status 2	0: No watchdog timer interrupt request signal has occurred. 1: A watchdog timer interrupt request signal has occurred due to the overflow of the 8-bit up counter.
WINTST1	Watchdog timer interrupt request signal factor status 1	0: No watchdog timer interrupt request signal has occurred. 1: A watchdog timer interrupt request signal has occurred due to releasing of the 8-bit up counter outside the clear time.
WDTST	Watchdog timer operating state status	0: Operation disabled 1: Operation enabled

Note 1：WDST <WINTST2> and WDST <WINTST1> are cleared to "0" by reading WDST.

Note 2：Values after reset are read from bits 7 to 3 of WDST.

13.1.2.1 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.

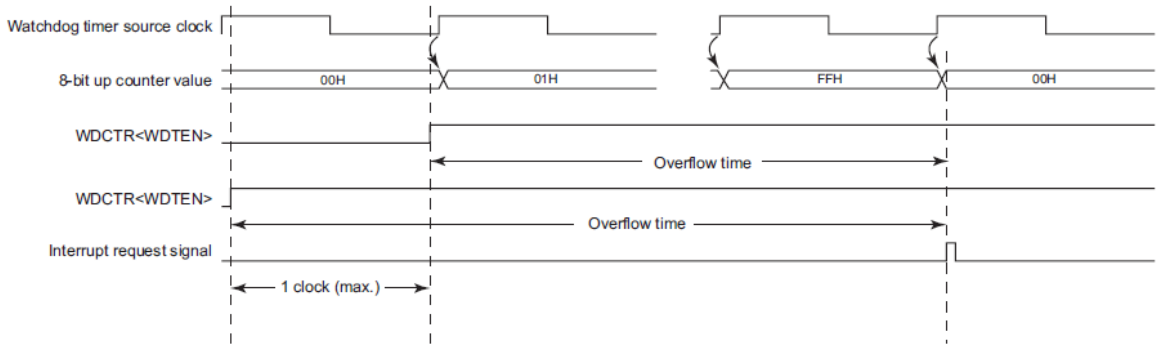


FIGURE 13- 2 WDCTR<WDTEN>SET TIMING AND OVERFLOW TIME

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.

13.1.2.2 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.

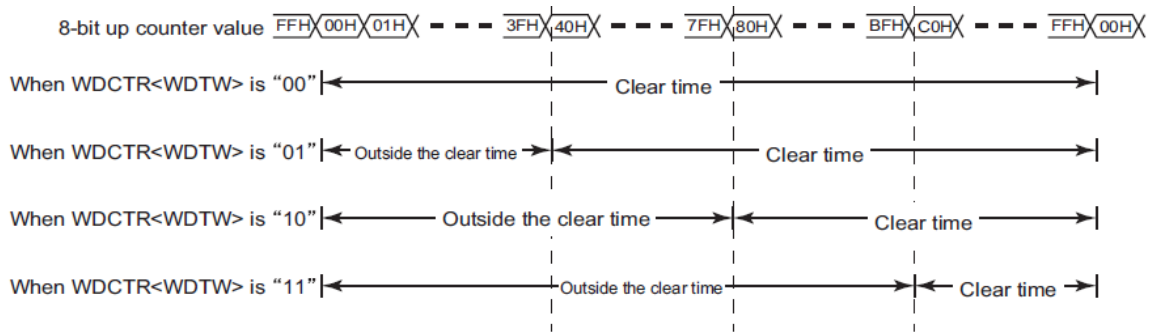


FIGURE 13- 3 WDCTR <WDTW> AND THE 8-BIT UP COUNTER CLEAR TIME

13.1.2.3 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 overflow time		
	Normal mode (fsysclk=HIRC/PLL/HXTAL)		Normal mode (fsysclk=LIRC)
	TBTCR<DV9CK> = 0	TBTCR<DV9CK> = 1	
00	10.92ms	62.5ms	62.5ms
01	43.70 ms	250ms	250ms
10	174.76ms	1s	1s
11	699.06ms	4s	4s

TABLE 13- 1 WATCHDOG TIMER OVERFLOW TIME (FSYSCLK=8.0 MHZ; FLCLK=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.

13.1.2.4 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.

13.1.2.5 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.

13.1.2.6 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.

13.1.2.7 Reading the Watchdog Timer Status

The watchdog timer status can be read at WDST.

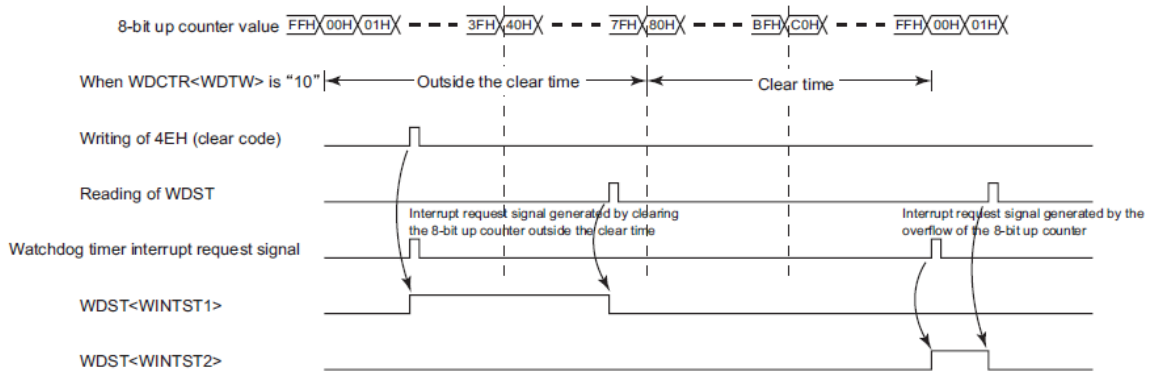
WDST <WDTST> is set to "1" when the watchdog timer operation is enabled, and it is cleared to "0" when the watchdog timer operation is disabled.

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.



FIURE 13- 4 CHANGES IN THE WATCHDOG TIMER STATUS

13.2 Divider Output (DVOB)

13.2.1 Configuration

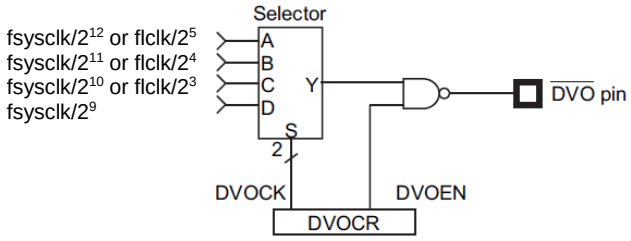


FIGURE 13- 5 DIVIDER OUTPUT

13.2.2 Control

The divider output is controlled by the divider output control register (DVOCR).

Divider Output Control Register (DVOCR, 0x002F)

DVOCR	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	DVOEN	DVOCK[1:0]	
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 1: Enable			
DVOCK[1:0]	Select the divider output frequency Unit: [Hz]		(fsysclk=HIRC/PLL/HXTAL)		(fsysclk=LIRC)
			TBTCR<DV9CK>=0	TBTCR<DV9CK>=1	
		00:	fsysclk /2^{12}	flclk/2^5	flclk/2^5
		01:	fsysclk /2^{11}	flclk/2^4	flclk/2^4
		10:	fsysclk /2^{10}	flclk/2^3	flclk/2^3
		11:	fsysclk /2^9	Reserved	Reserved

Note 1： fsysclk: system clock [Hz], flclk: 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 fsysclk=LIRC, the DVO frequency is subject to some fluctuations to synchronize fs and fSYSCLK.
Note 4：Bits 7 to 3 of DVOCR are read as "0".

13.2.3 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>.

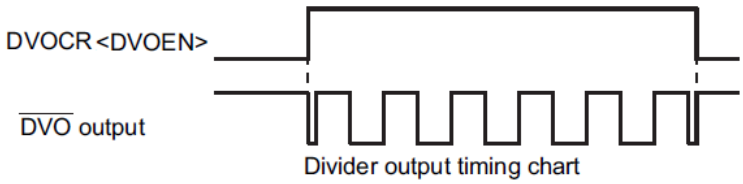


FIGURE 13- 6 DIVIDER OUTPUT TIMING

When the operation is changed from Normal mode(system clock: HIRC/ PLL/HXTAL) to Normal mode(system clock: LIRC) or from Normal mode(system clock: LIRC) to Normal mode(system clock: HIRC/ PLL/HXTAL), the divider output frequency does not reach the expected value due to synchronization of the gear clock (fsysclk) and the low-frequency clock (flclk).

DVOCK	Divider Output Frequency [Hz]		
	Normal/ Sleep mode (fsysclk=HIRC/PLL/HXTAL)		Normal mode Sleep mode (fsysclk=LIRC)
	TBTCR<DV9CK> = 0	TBTCR<DV9CK> = 1	
00	5.86k	1.024k	1.024k
01	11.72k	2.048k	2.048k
10	23.44k	4.096k	4.096k
11	46.875k	Reserved	Reserved

TABLE 13- 2 DIVIDER OUTPUT FREQUENCY

(Example: fsysclk =24MHz, flclk=32.768kHz)

13.3 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.

13.3.1 Configuration

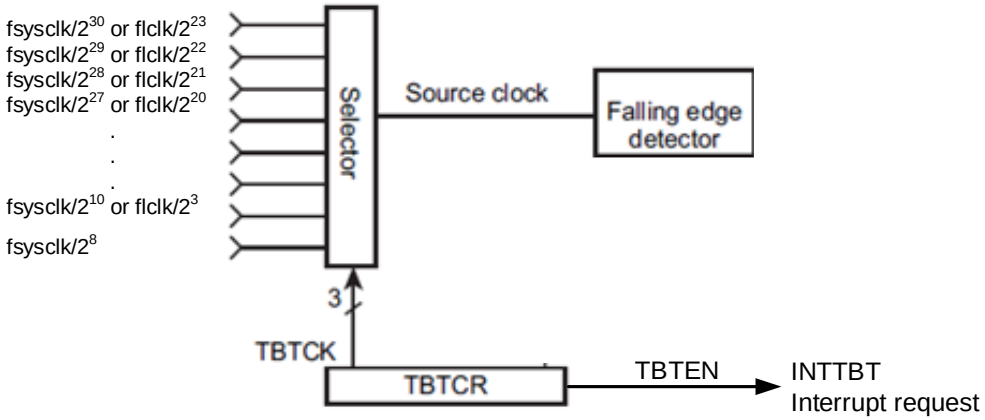


FIGURE 13- 7 TIME BASE TIMER CONFIGURATION

13.3.2 Control

The time base timer is controlled by the time base timer control register (TBTCR).

Time Base Timer Control Register (TBTCR, 0x002E)

TBTCR	7	6	5	4	3	2	1	0
Bit Symbol	-	-	DV9CK	TBTEN	TBTCK[3:0]			
Read/Write	R	R	R/W	R/W	R/W			
After reset	0	0	0	0	0	0	0	0

DV9CK	Select the input clock of the 9-stage divider	0: fsysclk/2^9 1: flclk/4			
TBTEN	Enable / disable the time base timer interrupt requests.	0: Disable 1: Enable			
TBTCK[3:0]	Select the time base timer interrupt frequency Unit: [Hz]	TBTCK	Normal/Sleep mode (fsysclk=HIRC/PLL/HXTAL)		Normal/Sleep mode (fsysclk=LIRC) (**SCKSRC=0x02**)
			TBTCR<DV9CK>=0	TBTCR<DV9CK>=1	
		0000:	fsysclk/2^{30}	flclk/2^{23}	flclk/2^{23}
		0001:	fsysclk/2^{29}	flclk/2^{22}	flclk/2^{22}
		0010:	fsysclk/2^{28}	flclk/2^{21}	flclk/2^{21}
		0011:	fsysclk/2^{27}	flclk/2^{20}	flclk/2^{20}
		0100:	fsysclk/2^{26}	flclk/2^{19}	flclk/2^{19}
		0101:	fsysclk/2^{25}	flclk/2^{18}	flclk/2^{18}
		0110:	fsysclk/2^{24}	flclk/2^{17}	flclk/2^{17}
		0111:	fsysclk/2^{23}	flclk/2^{16}	flclk/2^{16}
		1000:	fsysclk/2^{22}	flclk/2^{15}	flclk/2^{15}
		1001:	fsysclk /2^{20}	flclk/2^{13}	flclk/2^{13}
		1010:	fsysclk /2^{15}	flclk/2^8	Reserved
		1011:	fsysclk /2^{13}	flclk/2^6	Reserved
		1100:	fsysclk /2^{12}	flclk/2^5	Reserved
		1101:	fsysclk /2^{11}	flclk/2^4	Reserved
		1110:	fsysclk /2^{10}	flclk/2^3	Reserved
		1111:	fsysclk /2^8	Reserved	Reserved

Note 1: fsysclk : system clock, flclk : Low-frequency clock [Hz]

Note 2: When the operation is changed to the deep sleep 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 fsysclk=LIRC, the interrupt request is subject to some fluctuations to synchronize flclk and fsysclk.

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

13.3.3 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>.

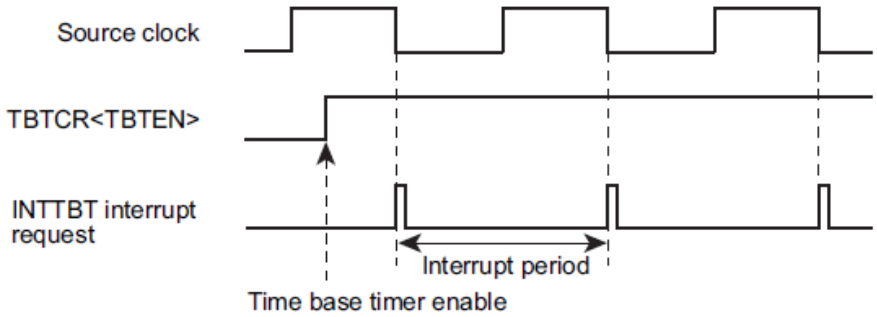


FIGURE 13- 8 TIME BASE TIMER INTERRUPT

When the operation is changed from Normal mode(system clock: HIRC/ PLL/HXTAL) to Normal mode(system clock: LIRC) or from Normal mode(system clock: LIRC) to Normal mode(system clock: HIRC/ PLL/HXTAL), the interrupt request will not occur at the expected timing due to synchronization of the gear clock (fsysclk) and the low-frequency clock (flclk). It is recommended that the operation mode is changed when TBTCR <TBTEN> is "0".

TBTCK	Time Base Timer Interrupt Frequency [Hz]		
	fsysclk=HIRC/PLL/HXTAL		fsysclk=LIRC (SCKSRC=0x02)
	TBTCR<DV9CK> = 0	TBTCR<DV9CK> = 1	
0000	0.0224	0.0039	0.0039
0001	0.0447	0.0078	0.0078
0010	0.0894	0.0156	0.0156
0011	0.1788	0.0313	0.0313
0100	0.3576	0.0625	0.0625
0101	0.7153	0.125	0.125
0110	1.431	0.25	0.25
0111	2.861	0.5	0.5
1000	5.722	1	1
1001	22.89	4	4
1010	732.42	128	reserved
1011	2930	512	reserved
1100	5859	1024	reserved
1101	11719	2048	reserved
1110	23438	4096	reserved
1111	93750	reserved	reserved

table 13- 3 Time Base Timer Interrupt Frequency (Example: fsysclk=24.0MHz，flclk=32.768kHz)

13.4 16-bit Timer Counter(TCA)

SQ7613 contains 8 channels of high-performance 16-bit timer counters.(TCA0~TCA7)

	Timer Input Pin	Timer Output Pin
Timer TCA0	TCA0	TCA0
Timer TCA1	TCA1	TCA1
Timer TCA2	TCA2	TCA2
Timer TCA3	TCA3	TCA3
Timer TCA4	TCA4	TCA4
Timer TCA5	TCA5	TCA5
Timer TCA6	TCA6	TCA6
Timer TCA7	TCA7	TCA7

TABLE 13- 4 16-BIT TIMER COUNTER PIN NAME

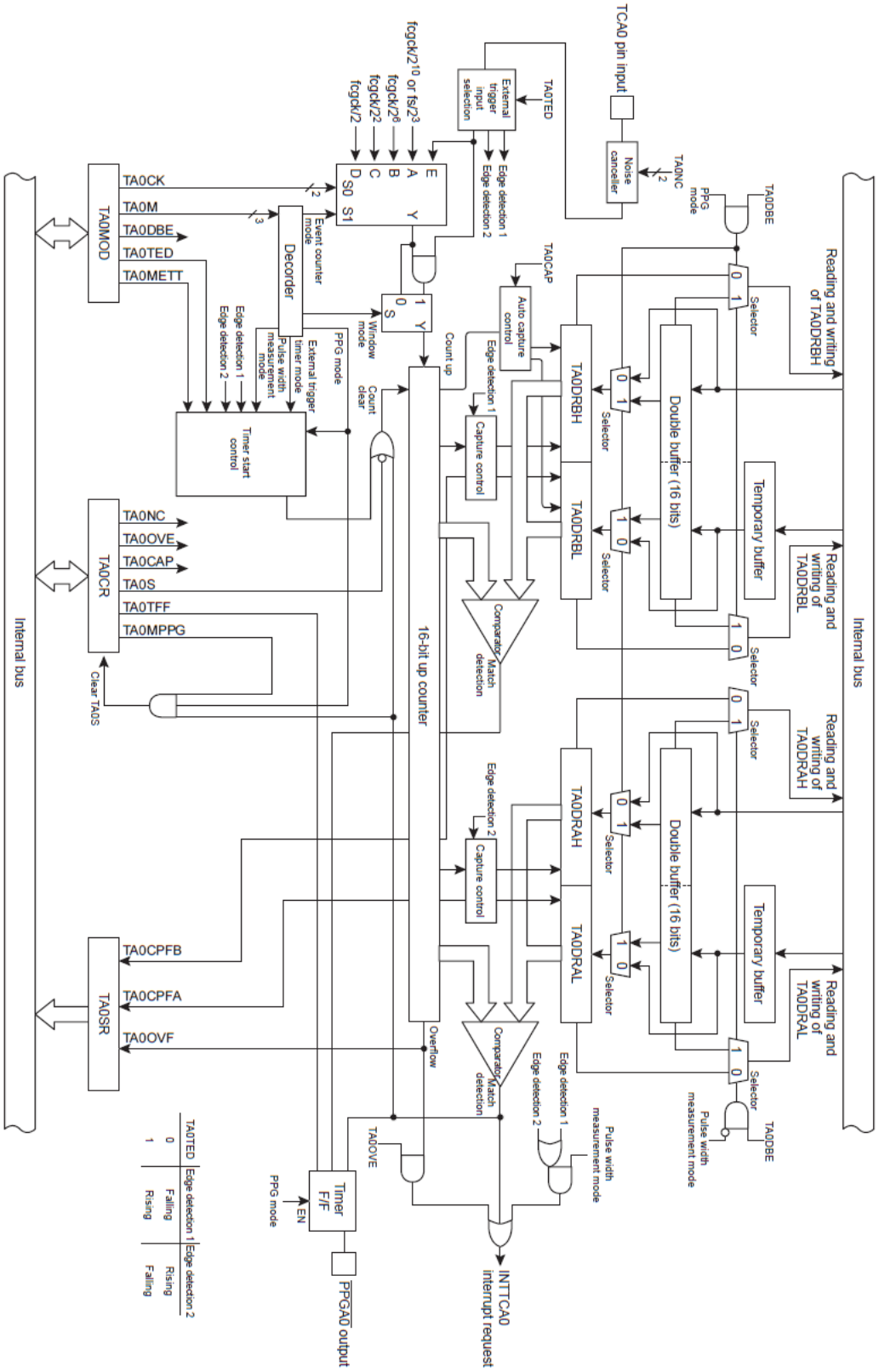


FIGURE 13- 9 16-BIT TIMER COUNTER

13.4.1 Control

Timer counter TCA is controlled by the peripheral circuit clock enable register PCKEN0/PCKEN1, the timer TCAx mode register TAxMOD (x=0~7), the timer TCAx control register TAxCR (x=0~7), the 16-bit timer TCAx register TAxDRA and TAxDRB (x=0~7).

ADDRESS	REGISTER	DESCRITION
0x0178	PCKEN0	Peripheral circuit clock enable register 0
0x0179	PCKEN1	Peripheral circuit clock enable register 1
0x0070	TA0MOD	Timer TCA0 mode register
0x0071	TA1MOD	Timer TCA1 mode register
0x0072	TA2MOD	Timer TCA2 mode register
0x0073	TA3MOD	Timer TCA3 mode register
0x0074	TA4MOD	Timer TCA4 mode register
0x0075	TA5MOD	Timer TCA5 mode register
0x0076	TA6MOD	Timer TCA6 mode register
0x0077	TA7MOD	Timer TCA7 mode register
0x0068	TA0CR	Timer TCA0 Control Register
0x0069	TA1CR	Timer TCA1 Control Register
0x006A	TA2CR	Timer TCA2 Control Register
0x006B	TA3CR	Timer TCA3 Control Register
0x006C	TA4CR	Timer TCA4 Control Register
0x006D	TA5CR	Timer TCA5 Control Register
0x006E	TA6CR	Timer TCA6 Control Register
0x006F	TA7CR	Timer TCA7 Control Register
0x0078	TA0SR	Timer TCA0 status register A
0x0079	TA1SR	Timer TCA1 status register A
0x007A	TA2SR	Timer TCA2 status register A
0x007B	TA3SR	Timer TCA3 status register A
0x007C	TA4SR	Timer TCA4 status register A
0x007D	TA5SR	Timer TCA5 status register A
0x007E	TA6SR	Timer TCA6 status register A
0x007F	TA7SR	Timer TCA7 status register A
0x0081	TA0DRAH	Timer TCA0 high register A
0x0085	TA1DRAH	Timer TCA1 high register A
0x0089	TA2DRAH	Timer TCA2 high register A
0x008D	TA3DRAH	Timer TCA3 high register A
0x0091	TA4DRAH	Timer TCA4 high register A
0x0095	TA5DRAH	Timer TCA5 high register A
0x0099	TA6DRAH	Timer TCA6 high register A
0x009D	TA7DRAH	Timer TCA7 high register A
0x0080	TA0DRAL	Timer TCA0 low register A
0x0084	TA1DRAL	Timer TCA1 low register A
0x0088	TA2DRAL	Timer TCA2 low register A
0x008C	TA3DRAL	Timer TCA3 low register A
0x0090	TA4DRAL	Timer TCA4 low register A
0x0094	TA5DRAL	Timer TCA5 low register A
0x0098	TA6DRAL	Timer TCA6 low register A
0x009C	TA7DRAL	Timer TCA7 low register A
0x0083	TA0DRBH	Timer TCA0 high register B
0x0087	TA1DRBH	Timer TCA1 high register B

ADDRESS	REGISTER	DESCRITION
0x008B	TA2DRBH	Timer TCA2 high register B
0x008F	TA3DRBH	Timer TCA3 high register B
0x0093	TA4DRBH	Timer TCA4 high register B
0x0097	TA5DRBH	Timer TCA5 high register B
0x009B	TA6DRBH	Timer TCA6 high register B
0x009F	TA7DRBH	Timer TCA7 high register B
0x0082	TA0DRBL	Timer TCA0 low register B
0x0086	TA1DRBL	Timer TCA1 low register B
0x008A	TA2DRBL	Timer TCA2 low register B
0x008E	TA3DRBL	Timer TCA3 low register B
0x0092	TA4DRBL	Timer TCA4 low register B
0x0096	TA5DRBL	Timer TCA5 low register B
0x009A	TA6DRBL	Timer TCA6 low register B
0x009E	TA7DRBL	Timer TCA7 low register B

This table shows the TCA register address and the description of each register. TCA setting are similar, so the registers are described by the general symbol x (x=0~7).

Peripheral Circuit Clock Enable Register 0 (PCKEN0)

PCKEN0	7	6	5	4	3	2	1	0
Bit Symbol	TCA3	TCA2	TCA1	TCA0	reserved	reserved	reserved	reserved
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

TCA3	TCA3 enable control	0: Disable 1: Enable
TCA2	TCA2 enable control	0: Disable 1: Enable
TCA1	TCA1 enable control	0: Disable 1: Enable
TCA0	TCA0 enable control	0: Disable 1: Enable

Peripheral Circuit Clock Enable Register1(PCKEN1)

PCKEN1	7	6	5	4	3	2	1	0
Bit Symbol	reserved	UART2	UART1	UART0	TCA7	TCA6	TCA5	TCA4
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

TCA7	TCA7 enable control	0: Disable 1: Enable
TCA6	TCA6 enable control	0: Disable 1: Enable
TCA5	TCA5 enable control	0: Disable 1: Enable
TCA4	TCA4 enable control	0: Disable 1: Enable

Timer TCAx Mode Register (TAxMOD), x=0~7

TAxMOD	7	6	5	4	3	2	1	0
Bit Symbol	TAxDBE	TAxTED	TAxCAP TAxMETT	TAxCK[1:0]		TAxM[2:0]		
Read/Write	R/W	R/W	R/W	R/W		R/W		
After reset	1	0	0	0	0	0	0	0

TAxDBE	Double buffer control	0: Disable the double buffer 1: Enable the double buffer			
TAxTED	External trigger input selection	0: Rising edge / H Level 1: Falling edge / L Level			
TAxCAP	Pulse width measurement mode control	0: Double edge capture 1: Single edge capture			
TAxMETT	External trigger timer mode control	0: Trigger start 1: Trigger start and stop			
TAxCK [1:0]	Timer TCAx operation mode selection		Normal/SLEEP mode (fsysclk=HIRC/PLL/HXTAL)		NORMAL/ SLEEP mode (fsysclk=LIRC)
			TBTCR<DV9CK>=0	TBTCR<DV9CK>=1	
		00:	fsysclk/2^{10}	fIclk/2^3	fIclk/2^3
		01:	fsysclk /2^6	fsysclk/2^6	-
		10:	fsysclk /2^2	fsysclk/2^2	-
		11:	fsysclk /2	fsysclk/2	-
TAxM[2:0]	Timer TCAx working 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： fsysclk: Gear clock [Hz], fIclk: Low-frequency clock [Hz]
Note 2：Set TAxMOD in the stopped state (TAxCR <TAxS>="0"). Writing to TAxMOD is invalid during the operation (TAxCR <TAxS>="1").

Timer TCAx Control Register (TAxCR), x=0~7

TAxCR	7	6	5	4	3	2	1	0
Bit Symbol	TAxOVE	TAxTFF	TAxNC [1:0]		-	-	TAxCAP TAxMPPG	TAxS
Read/Write	R/W	R/W	R/W		R	R	R/W	R/W
After reset	0	1	0	0	0	0	0	0

TAxOVE	Overflow interrupt control	0: No INTTCAx interrupt request when the counter overflow occurs. 1: INTTCAx interrupt request when the counter overflow occurs.		
TAxTFF	Timer F/F control	0: Clear 1: Set		
TAxNC[1:0]	Noise canceller sampling interval setting		Normal/SLEEP mode	NORMAL/ SLEEP mode(Slow Clock)
		00:	No noise canceller	No noise canceller
		01:	fsysclk /2	-
		10:	fsysclk /2^2	-
		11:	fsysclk /2^8	flclk/2
TAxCAP	Auto capture function	0: Disable Auto capture 1: Enable Auto capture		
TAxMPPG	PPG output control	0: Continuous 1:One-shot		
TAxS	TCA start control	0: Stop and counter clear 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 TAxTFF, TAxOVE and TAxNC in the stopped state (TAxS="0"). Writing is invalid during the operation (TAxS="1").
Note 3: When the DEEP SLEEP mode is started, the start control (TAxS) is automatically cleared to "0" and the timer stops. Set TAxS again to use the timer counter after the release of the STOP mode.
Note 4: When a read instruction is executed on TAxCR, bits 3 and 2 are read as "0".
Note 5: Do not set TAxNC to "01" or "10" when the normal mode(slow clock) or SLEEP mode(slow clock) is used. Setting TAxNC to "01" or "10" stops the noise canceller and no signal is input to the timer.

Timer TCAx Status Register (TAxSR), x=0~7

TAxSR	7	6	5	4	3	2	1	0
Bit Symbol	TAxOVF	-	-	-	-	-	TAxCPFA	TAxCPFB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

TAxOVF	Overflow flag	0: No overflow has occurred. 1: At least an overflow has occurred.
TAxCPFA	Capture completion flag A	0: No capture operation has been executed. 1: At least a pulse width capture has been executed in the double-edge capture
TAxCPFB	Capture completion flag B	0: No capture operation has been executed. 1: At least a capture operation has been executed in the single-edge capture. At least a pulse duty width capture has been executed in the double-edge capture.

Note 1：TAxOVF, TAxCPFA and TAxCPFB are cleared to "0" automatically after TAxSR is read. Writing to TCAxSR is invalid.
Note 2：When a read instruction is executed on TAxSR, bits 6 to 2 are read as "0".

Timer TCAx High RegisterA (TAxDRAH), x=0~7

TAxDRAH	15	14	13	12	11	10	9	8
Bit Symbol	TADRAH[15:8]							
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 TCAx Low RegisterA (TAxDRAL) , x=0~7

TAxDRAL	7	6	5	4	3	2	1	0
Bit Symbol	TAxDRAL[7:0]							
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 TCAx High Register B (TAxDRBH) , x=0~7

TAxDRBH	15	14	13	12	11	10	9	8
Bit Symbol	TAxDRBH[15:8]							
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 TCAx Low Register B (TAxDRBL) , x=0~7

TAxDRBL	7	6	5	4	3	2	1	0
Bit Symbol	TAxDRBL[7:0]							
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 register cannot be written in the pulse width measurement mode.

13.4.2 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.

13.4.3 Timer Function

Timer counter TCAx has six types of operation modes; timer, external trigger timer, event counter, window, pulse width measurement and programmable pulse generate (PPG) output modes.

13.4.3.1 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		
	NORMAL mode or SLEEP mode		NORMAL mode or SLEEP mode (Slow Clock)	fsysclk =24MHz		fIclk = 32.769 Hz
	TBTCR <DV9CK> =0	TBTCR <DV9CK> =1		TBTCR <DV9CK> =0	TBTCR <DV9CK> =1	
00	fsysclk/2^{10}	fIclk/2^3	fIclk/2^3	42.67us	244.14us	244.14us
01	fsysclk/2^6	fsysclk/2^6	-	2.67 us	2.67 us	-
10	fsysclk/2^2	fsysclk/2^2	-	166.67ns	166.67ns	-
11	fsysclk/2	fsysclk/2	-	83.34ns	83.34ns	-

TABLE 13- 5 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 (TCA0DRB) by setting TA0CR <TA0CAP> to "1" (auto capture function). When TA0CR<TA0CAP> 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 <TA0CAP> written to "1", the auto capture is enabled immediately after the timer is started.

Note：The value set to TA0CR <TA0CAP> 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

The MCU 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 TAxDRAH in this order.

2. Double Buffer

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.

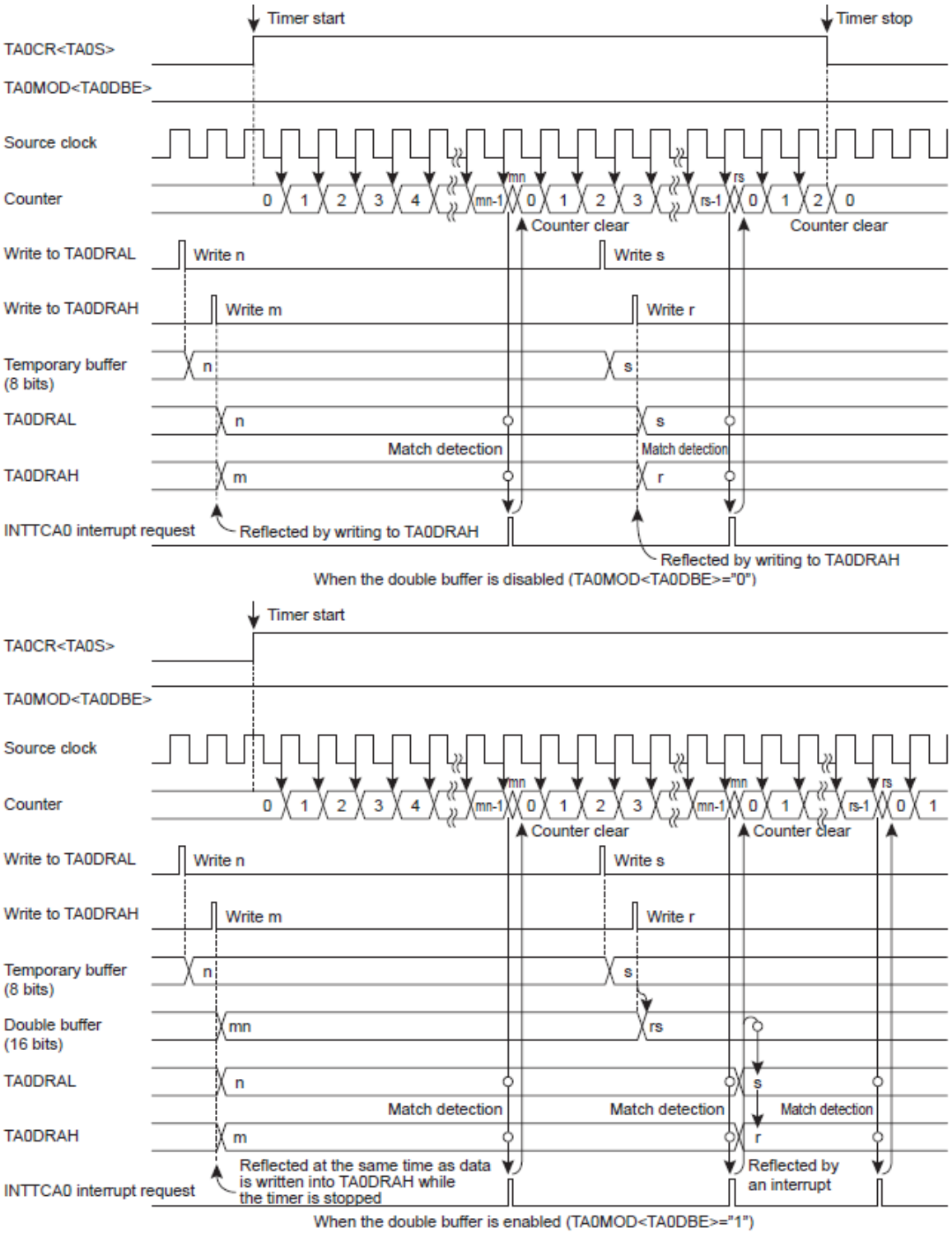


FIGURE 13- 10 TIMER MODE TIMING CHART

- 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.

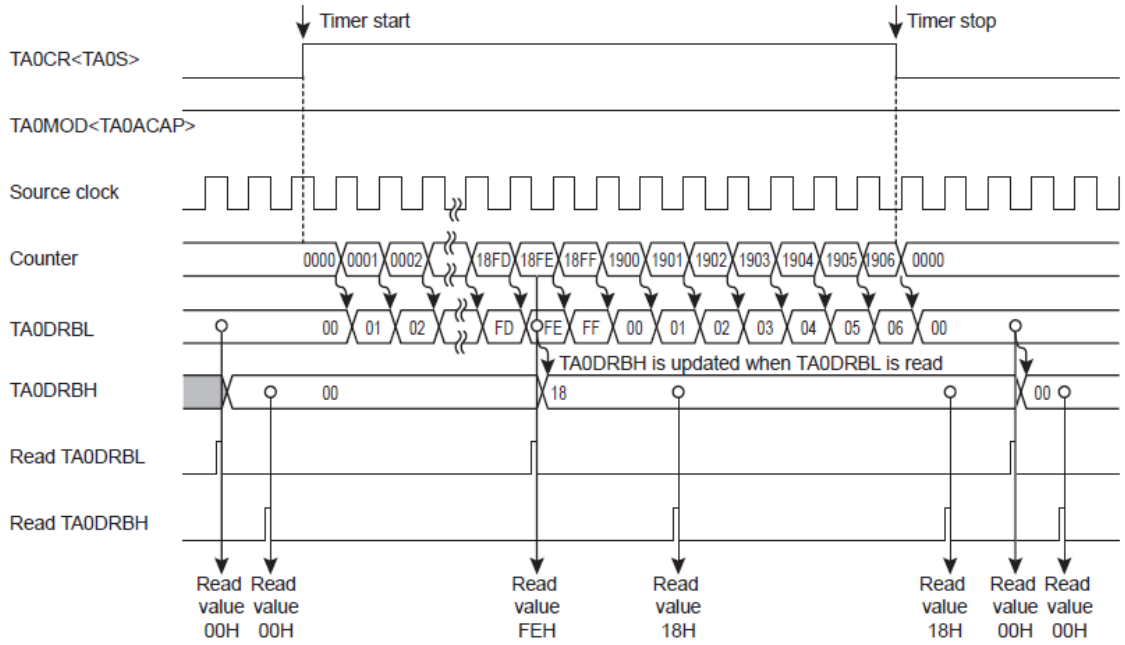


FIGURE 13- 11 TCA0 TIMER MODE TIMING CHART (AUTO CAPTURE)

13.4.3.2 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.

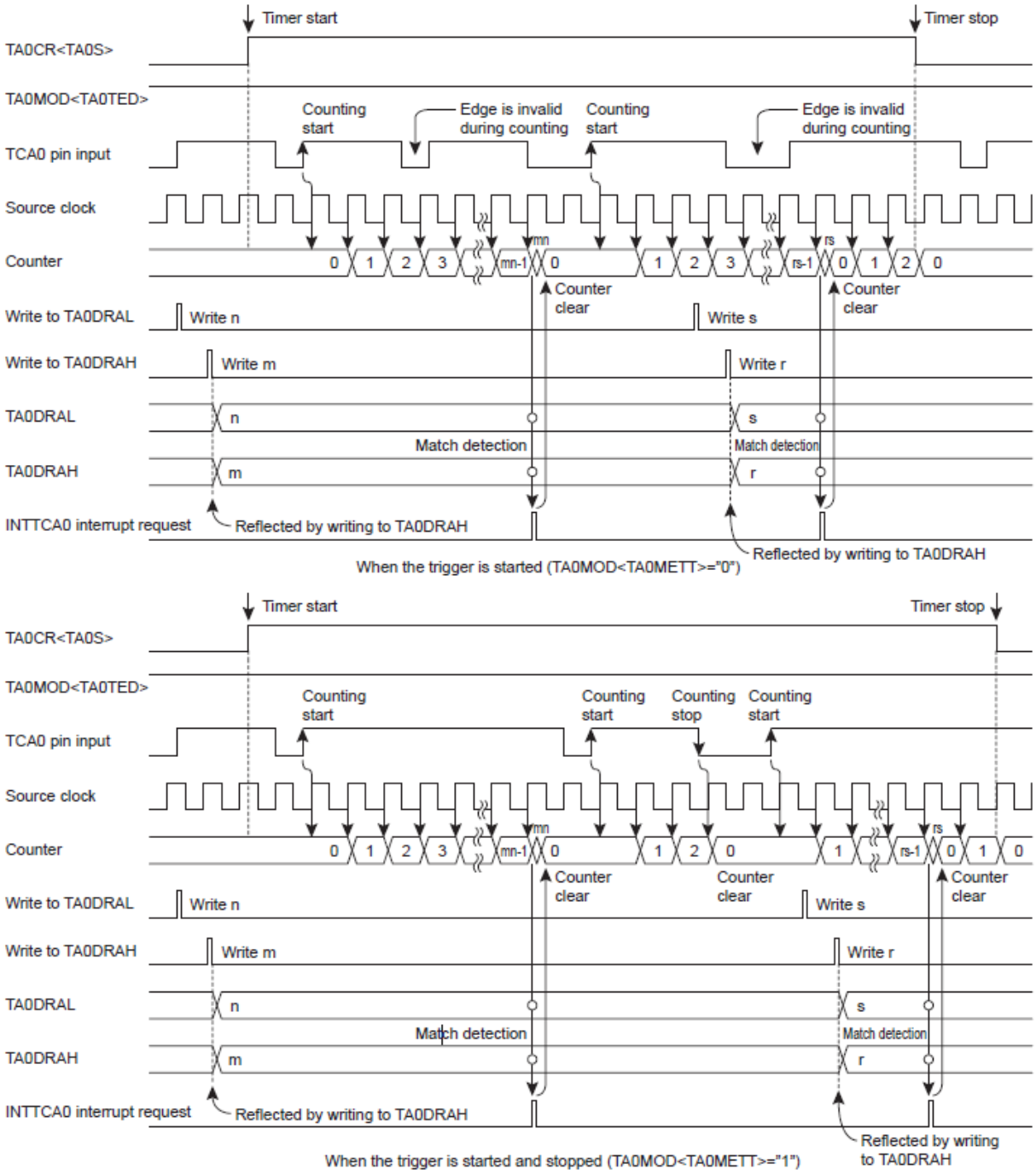


FIGURE 13- 11 EXTERNAL TRIGGER TIMER MODE TIMING CHART

(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 TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the timer is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments according to the selected source clock. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting.
When TA0MOD <TA0METT> is "1" and the edge opposite to the selected trigger edge is detected, the up counter stops counting and is cleared to "0x0000". Subsequently, when the selected trigger edge is detected, the up counter restarts counting. In this mode, an interrupt request can be generated by detecting that the input pulse exceeds a certain pulse width. If TA0MOD <TA0METT> is "0", the detection of the selected edge and the opposite edge is ignored during the period from the detection of the specified trigger edge and the start of counting through until the match detection.

Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

(c) Auto Capture

Refer to "13.4.3.1 - (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "13.4.3.1 - (d) Register Buffer Configuration ".

13.4.3.3 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 TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

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".

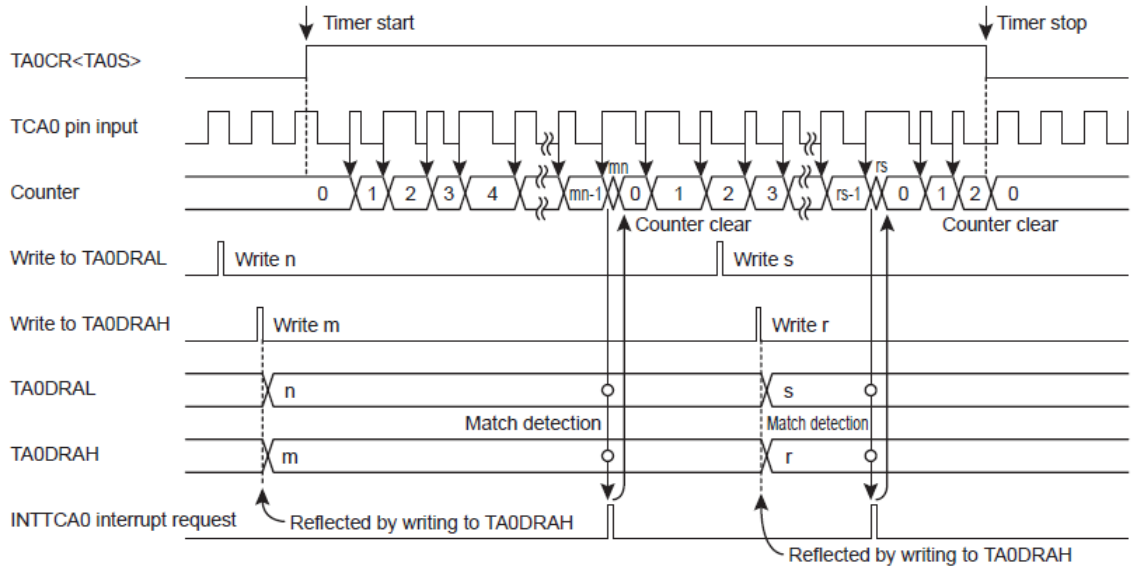
The maximum frequency to be supplied is $fsysclk/2^2$ [Hz] (NORMAL mode or SLEEP mode) or $flclk/2^4$ [Hz] (NORMAL mode(Slow Clock) or SLEEP mode(Slow Clock)), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

(c) Auto Capture

Refer to "13.4.3.1- (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "13.4.3.1- (d) Register Buffer Configuration ".



When the rising edge is selected (TA0MOD<TA0TED>="0")

figure 13- 12 Event Counter Mode Timing Chart

13.4.3.4 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.

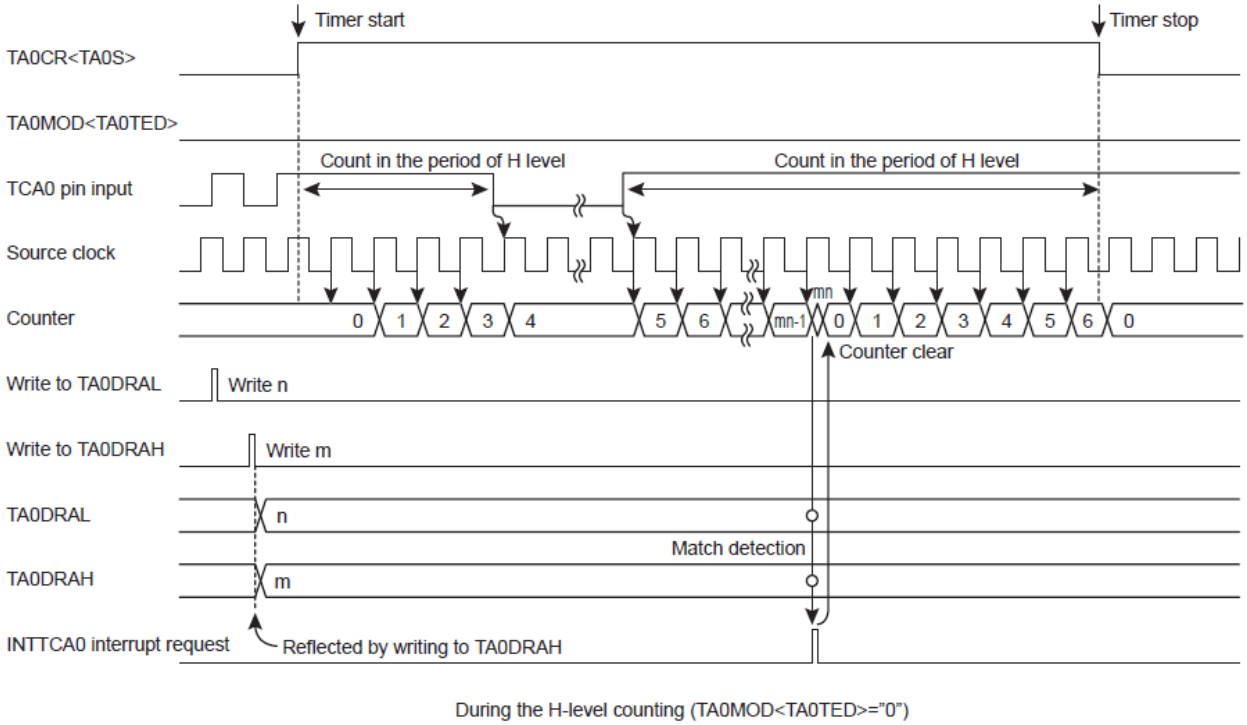


FIGURE 13- 13 WINDOW MODE TIMING CHART

(a) Setting

Setting the operation mode selection TA0MOD <TA0M> to "101" activates the window mode. Select the source clock at TA0MOD <TA0CK>.

Select the window pulse level at the trigger edge input selection TA0MOD <TA0TED>. Setting TA0MOD <TA0TED> to "0" enables counting up as long as the window pulse is at the "H" level. Setting TA0MOD <TA0TED> to "1" enables counting up as long as the window pulse is at the "L" level.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the operation is started, when the level selected at TA0MOD <TA0TED> is input to the TCA0 pin, the up counter increments according to the source clock selected at TA0MOD <TA0CK>. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting.

The maximum frequency to be supplied must be slow enough for the program to analyze the count value. Define a frequency pulse that is sufficiently lower than the programmed internal source clock.

Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

(c) Auto Capture

Refer to "13.4.3.1 - (c) Auto Capture".

(d) Register Buffer Configuration

Refer to "13.4.3.1 - (d) Register Buffer Configuration ".

13.4.3.5 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 TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR <TA0S> to "1". In this time, TA0DRA and TA0DRB register are initialized to "0x0000". After the timer is started, writing to TA0MOD and TA0CR <TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

(b) Operation

After the timer is started, when the selected trigger edge (start edge) is input to the TCA0 pin, INTTCA0 interrupt request is generated, and then the up counter increments according to the selected source clock. Subsequently, when the edge opposite to the selected edge is detected, the up counter value is captured into TA0DRB, an INTTCA0 interrupt request is generated, and TA0SR <TA0CPFB> is set to "1". Depending on the TA0MOD <TA0MCAP> setting, the operation differs as follows:

1. Double-edge capture (When TA0MOD <TA0MCAP> is "0")

The up counter continues counting up after the edge opposite to the selected edge is detected. Subsequently, when the selected trigger edge is input, the up counter value is captured into TA0DRA, an INTTCA0 interrupt request is generated, and TA0SR <TA0CPFA> is set to "1". At this time, the up counter is cleared to "0x0000".

2. Single-edge capture (When TA0MOD <TA0MCAP> is "1")

The up counter stops counting up and is cleared to "0x0000" when the edge opposite to the selected edge is detected. Subsequently, when the start edge is input, INTTCA0 interrupt request is generated, and then the up counter restarts increment.

When the up counter overflows during capturing, the overflow flag TA0SR <TA0OVF> is set to "1". At this time, an INTTCA0 interrupt request occurs if the overflow interrupt control TA0CR <TA0OVE> is set to "1".

The capture completion flags (TA0SR <TA0CPFA, TA0CPFB> and the overflow flag (TA0SR <TA0OVF>) are cleared to "0" automatically when TA0SR is read.

The captured value must be read from TA0DRB (and also from TA0DRA for the double-edge capture) before the next trigger edge is detected. If the captured value is not read, it becomes undefined. TA0DRA and TA0DRB must be read by using a 16-bit access instruction.

Setting TA0CR <TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

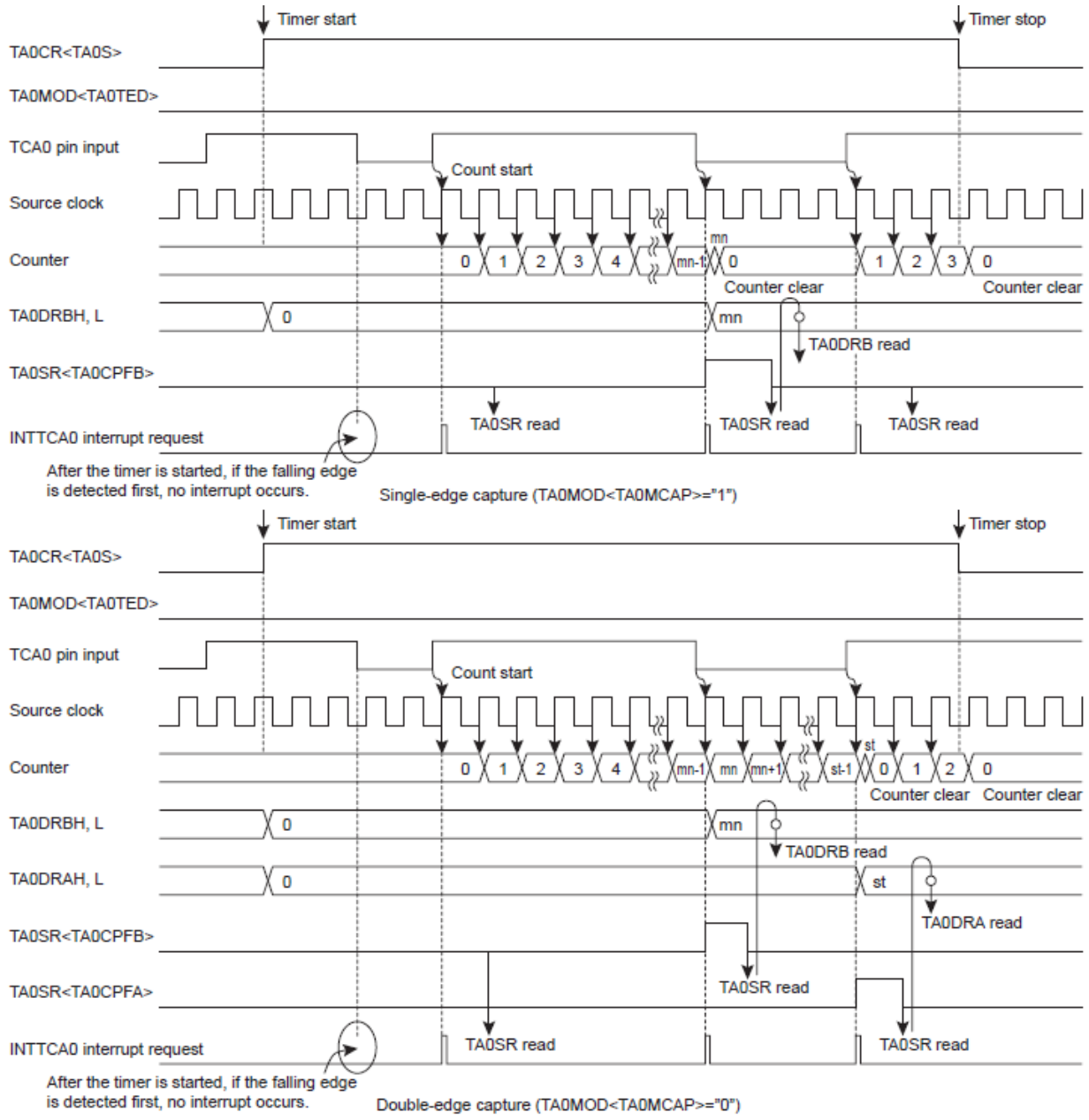


FIGURE 13- 14 PULSE WIDTH MEASUREMENT MODE TIMING CHART

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

This figure 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).

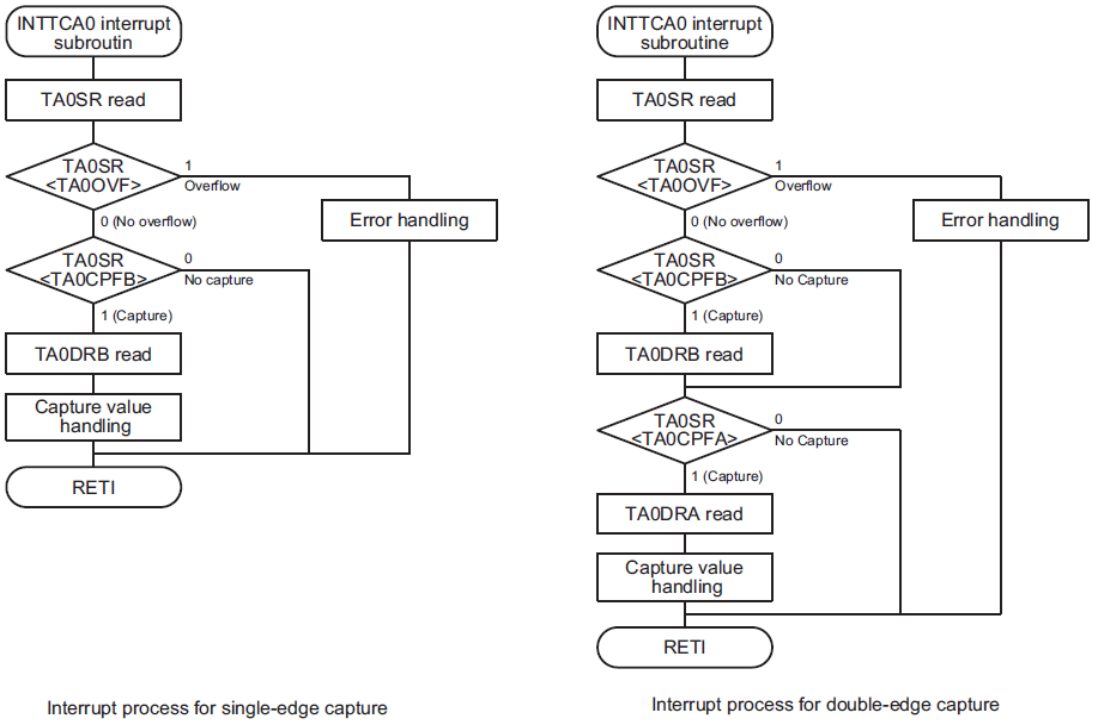


FIGURE 13- 16 EXAMPLE OF CAPTURE PROCESS

13.4.3.6 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.

(b) Register Buffer Configuration

1. Temporary Buffer

The MCU 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

The double buffer can be used by setting TA0CR <TA0DBF>. Setting TA0CR <TA0DBF> to "0" disables the double buffer. Setting TA0CR <TA0DBF> to "1" enables the double buffer.

- When the double buffer is enabled

When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L (TA0DRBH/L) compare the last set values to the counter value.

If a match is detected, an INTTCA0 interrupt request is generated and the double buffer set value is stored into TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

When a read instruction is executed on TA0DRAH/L (TA0DRBH/L), the double buffer value (the last set value) is read, not the TA0DRAH/L (TA0DRBH/L) values (the current effective values).

When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L (TA0DRBH/L).

- When the double buffer is disabled

When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is immediately stored in TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

If the values set to TA0DRAH/L (TA0DRBH/L) are smaller than the up counter value, the up counter overflows and the match detection is executed using a new set value. As a result, the output pulse width may be longer than the set time. If that is a problem, enable the double buffer.

When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into TA0DRAH/L (TA0DRBH/L).

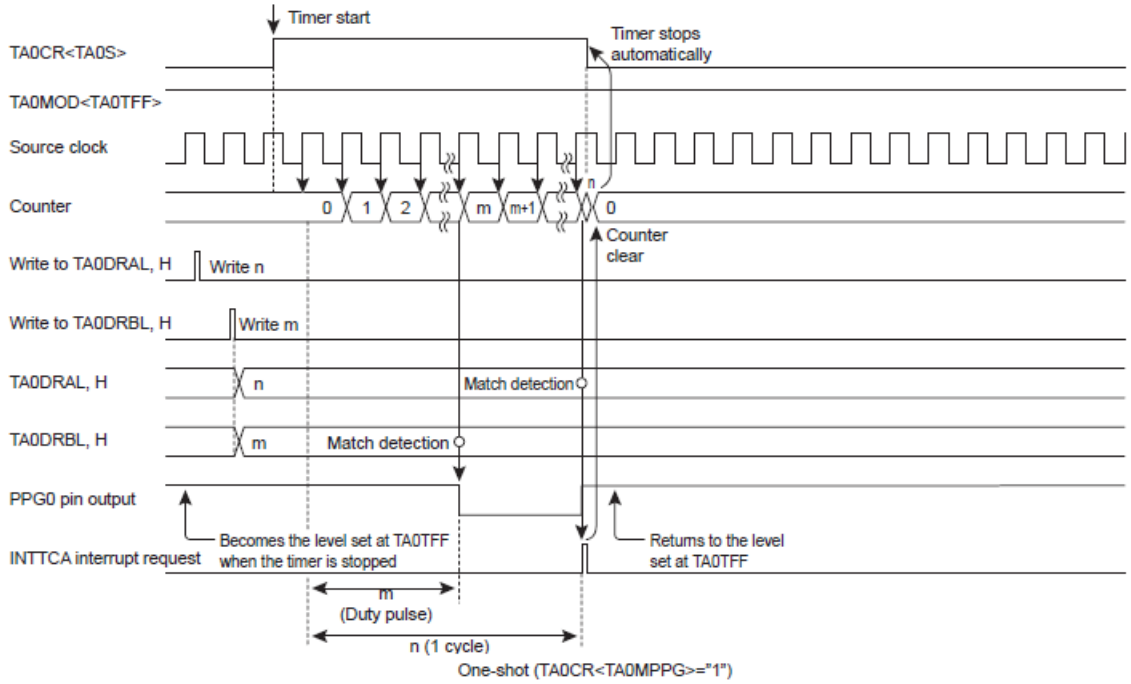


FIGURE 13- 17 PPG PPG MODE TIMING CHART - ONE SHOT

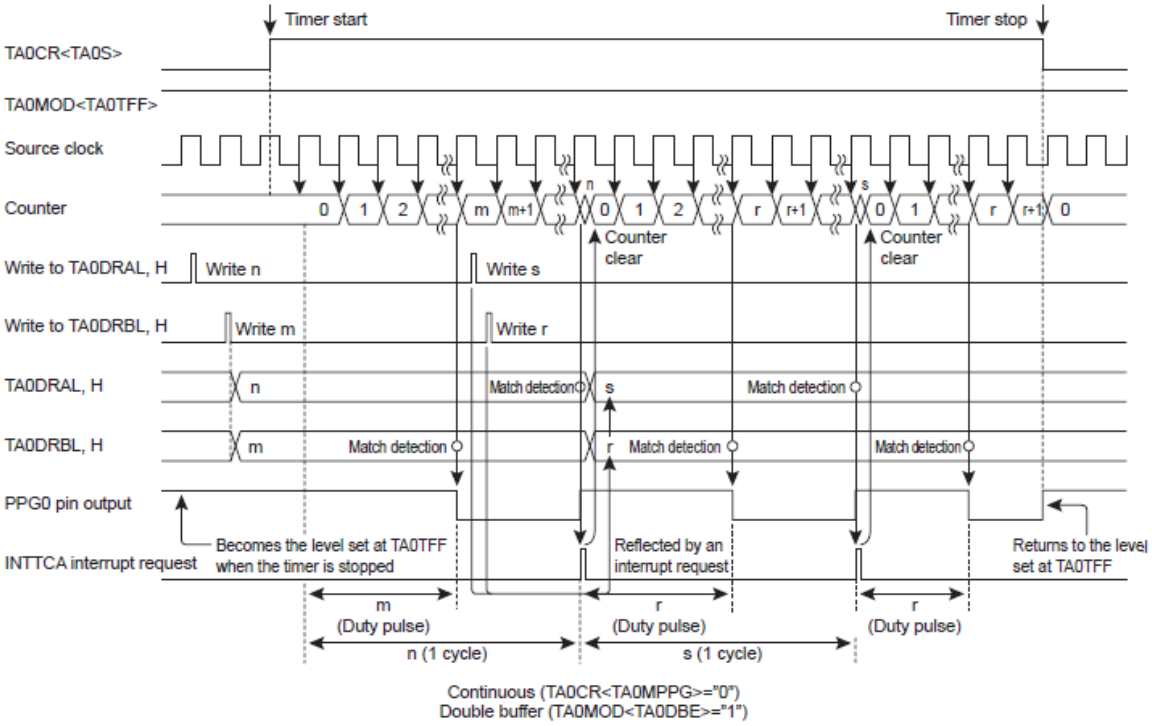


Figure 13.18 PPG Mode Timing Chart - Continuous

13.4.4 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.

14. Asynchronous Serial Interface (UART)

SQ7613 contains 3 channels of asynchronous serial interfaces (UART).This chapter describes asynchronous serial interface 0 (UART0). For UART1 and UART2, replace the SFR addresses and pin namesas shown in Table 14.1 and Table 14.2.

	UARTxCR1 (Address)	UARTxCR2 (Address)	UARTxDR (Address)	UARTxSR (Address)	RDxBUF (Address)	TDxBUF (Address)
UART0	UART0CR1 (0x00A0)	UART0CR2 (0x00A1)	UART0DR (0x00A2)	UART0SR (0x00A3)	RD0BUF (0x00A4)	TD0BUF (0x00A5)
UART1	UART1CR1 (0x00A6)	UART1CR2 (0x00A7)	UART1DR (0x00A8)	UART1SR (0x00A9)	RD1BUF (0x00AA)	TD1BUF (0x00AB)
UART2	UART2CR1 (0x00AC)	UART2CR2 (0x00AD)	UART2DR (0x00AE)	UART2SR (0x00AF)	RD2BUF (0x00B0)	TD2BUF (0x00B1)

TABLE 14- 1 SFR ADDRESS ASSIGNMENT

	Serial Data Input Pin	Serial Data Output Pin
UART0	RXD0	TXD0
UART1	RXD1	TXD1
UART2	RXD2	TXD2

TABLE 14- 2 PIN NAMES

14.1 UART Configuration

UARTx (x=0~2) is controlled by peripheral circuit clock enable registers PCKEN1, UARTx (x=0~2), control registers UARTxCR1 ,UARTxCR2, and UARTx baud rate (baud) register UARTxDR (x=0~2). The operating status can be monitored by the UART status control register UARTxSR (x=0~2).

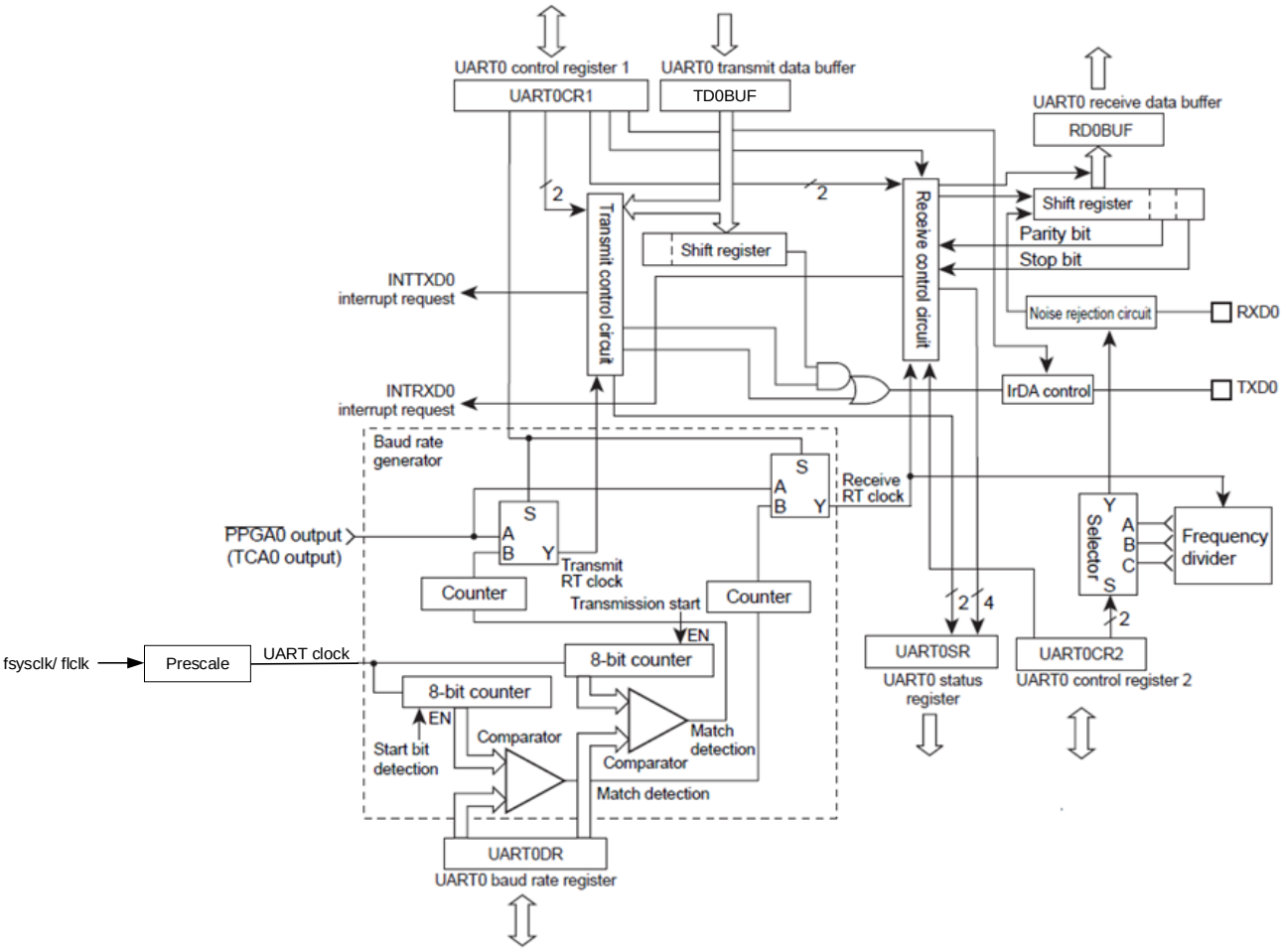


FIGURE 14- 1 ASYNCHRONOUS SERIAL INTERFACE (UART)

Peripheral Clock Enable Register 1 (PCKEN1, 0x0179)

PCKEN1	7	6	5	4	3	2	1	0
Bit Symbol	reserved	UART2	UART1	UART0	TCA7	TCA6	TCA5	TCA4
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

UART2	UART2 enable control	0: Disable 1: Enable
UART1	UART1 enable control	0: Disable 1: Enable
UART0	UART0 enable control	0: Disable 1: Enable

UARTx (x=0~2) control registers (UARTxCR1 and UARTxCR2), baud rate register UARTxDR, UART status control register UARTxSR, receive data register RDxBUF and transmit data register rDxBUF are all the same, so the lower register is a common format. (fill in the UART number to be operated, x=0~2). The address can be compared with Table 13.1 SFR address.

UARTx Control Register1 (UARTxCR1)

UARTxCR1	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 Note 2 1: Enable		
RXE	Receive operation	0: Disable Note 2 1: Enable		
STOPBT	Transmit stop bit length	0: 1 bit 1: 2 bits		
EVEN	Parity selection	0: Odd-numbered parity 1: Even number parity		
PE	Parity addition	0: No parity 1: Parity added		
IRDASEL	TXD pin output selectin	0: UART output 1: IrDA output		
BRG	Transfer base clock selection		Normal mode (fsysclk=HIRC/PLL/HXTAL)	Normal mode (fsysclk=LIRC)
		0	Fsysclk NOTE 1	FIclk NOTE 1
		1	TCAx output NOTE 5	

Note 1： *fsysclk，System clock[Hz]，flclk，Low-frequency [Hz]。*

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 "18.3 Protection to Prevent UART1CR1 and UART1CR2 Registers from Being Changed ".

UARTx Control Register2 (UARTxCR2)

UARTxCR2	7	6	5	4	3	2	1	0
Bit Symbol	DV[1:0]		RTSEL[2:0]			RXDNC[1:0]		STOPBR
Read/Write	R/W		R/W			R/W		R/W
After reset	0		0			0		0

DV [1:0]	Clock divider	00：fsysclk/1 01：fsysclk/2 10：fsysclk/4 11：fsysclk/8		
RTSEL[2:0]	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[1:0]	Selects the RXD input noise rejection time (Time of pulses to be removed as noise)	00: No noise rejection 01: 1 x (UARTxDR + 1) / (Transfer base clock frequency) [s] 10: 2 x (UARTxDR + 1) / (Transfer base clock frequency) [s] 11: 4 x (UARTxDR + 1) / (Transfer base clock frequency) [s]		
STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits		

Note 1：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 "13.7.1 Transfer baud rate calculation method".

Note 2：For details of the RXDNC noise rejection time, refer to "13.9 Received Data Noise Rejection".

Note 3：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 4：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 "18.3 Protection to Prevent UART1CR1 and UART1CR2 Registers from Being Changed ".

UARTx Baud Rate Register (UARTxDR)

UARTxDR	7	6	5	4	3	2	1	0
Bit Symbol	UARTxDR7	UARTxDR6	UARTxDR5	UARTxDR4	UARTxDR3	UARTxDR2	UARTxDR1	UARTxDR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1：Set UART0CR1<RXE> and UART0CR1<TXE> to "0" before changing UART0DR. For the set values, refer to "13.7.1 Transfer Baud Rate".

Note 2：When UART0CR1<BRG> is set to the TCA0 output, the value set to UART0DR has no meaning.

UARTx Status Register (UARTxSR)

UARTxSR	7	6	5	4	3	2	1	0
Bit Symbol	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

PERR	Parity error flag	0: No parity error 1: Parity error
RFERR	Framing error flag	0: No framing error 1: Framing error
OERR	Overrun error flag	0: No overrun error 1: Overrun error
RBSY	Receive busy flag	0: Before receiving or end of receiving 1: On receiving
RBFL	Receive buffer full flag	0: Receive buffer empty 1: Receive buffer full
TBSY	Transmit busy flag	0: Before transmission or end of transmission 1: On transmission
TBFL	Transmit buffer full flag	0: Transmit buffer empty 1: Transmit buffer full

Note 1：TBFL is cleared to "0" automatically after an INTTXD1 interrupt request is generated, and is set to "1" when data is set to TD1BUF.
Note 2：When a read instruction is executed on UART1SR, bit 4 is read as "0"

UARTx Receive Data Register (RDxBUF)

RDxBUF	7	6	5	4	3	2	1	0
Bit Symbol	RDxDR7	RDxDR6	RDxDR5	RDxDR4	RDxDR3	RDxDR2	RDxDR1	RDxDR0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

UARTx Transmit Data Register(TDxBUF)

TD0BUF	7	6	5	4	3	2	1	0
Bit Symbol	TDxDR7	TDxDR6	TDxDR5	TDxDR4	TDxDR3	TDxDR2	TDxDR1	TDxDR0
Read/Write	W	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0

14.2 UART Control

The UART contains 1 peripheral circuit clock control register, PCKEN1, which saves system power when no UART function is required.

Setting PCKEN 1<UARTxEN> to "0" stops the basic clock supply of the UART to save system power; the UART cannot be used at this time. Setting PCKEN 1<UARTxEN> to "1" starts the basic clock supply of the UART and starts the UART operation.

After reset, PCKEN 1<UARTxEN> will be restored to the initial setting of "0" and the UART operation will stop. Before using the UART for the first time, you must set PCKEN 1<UARTxEN> to "1" (before the UART control register is active) in the initial program settings.

Do not change PCKEN 1<UARTxEN> to "0" while the UART is operating, otherwise the UART may experience unexpected operation.

14.3 Protection of UART0CR1 and UART0CR2 Registers from Being Changed

The MCU has a function that protects the registers from being changed so that the UART communication settings (for example, stop bit and parity) are not changed accidentally during the UART operation.

Specific bits of UART0CR1 and UART0CR2 can be changed only under the conditions shown in Table 14.3. 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.

<table>
<tr><th rowspan="2">Bit to be changed</th><th rowspan="2">Function</th><th colspan="4">Conditions that allow the bit ti be changed</th></tr>
<tr><th>UARTxCR1
<TXE></th><th>UARTxSR
<TBSY></th><th>UARTxCR1
<RXE></th><th>UARTxSR
<TXE></th></tr>
<tr><td><UARTxCR1<STOPBT></td><td>Transmit stop bit length</td><td colspan="2">Both of these bits are "0"</td><td>-</td><td>-</td></tr>
<tr><td><UARTxCR1<EVEN></td><td>Parity selection</td><td colspan="4" rowspan="2">All of these bits are "0"</td></tr>
<tr><td><UARTxCR1<PE></td><td>Parity addition</td></tr>
<tr><td><UARTxCR1<IRDASEL></td><td>TXD pin output selection</td><td colspan="2">Both of these bits are "0"</td><td>-</td><td>-</td></tr>
<tr><td><UARTxCR1<BRG></td><td>Transfer base clock selection</td><td colspan="4" rowspan="2">All of these bits are "0"</td></tr>
<tr><td><UARTxCR2<RTSEL></td><td>Selection of number of RT clocks</td></tr>
<tr><td><UARTxCR2<RXDNC></td><td>Selection of RXD pin input noise rejection time</td><td rowspan="2">-</td><td rowspan="2">-</td><td colspan="2" rowspan="2">Both of these bits are "0"</td></tr>
<tr><td><UARTxCR2<STOPBR></td><td>Receive stop bit length</td></tr>
</table>

TABLE 14- 3 CHANGING OF UARTXCR1 AND UARTXCR2

14.4 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 UARTxCR1 <PE> that selects the presence or absence of parity and UARTxCR1 <EVEN> that selects even- or odd-numbered parity. The bit length of the stop bit can be selected at UARTxCR1 <STBT>.

Figure 14.2 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)

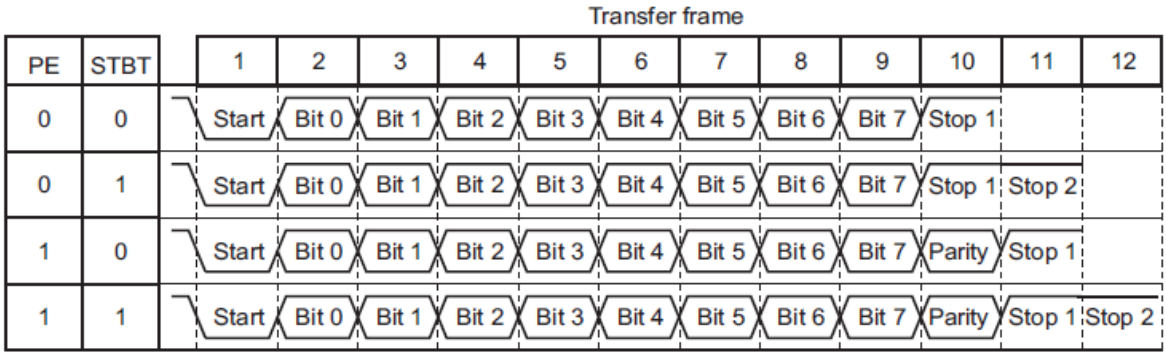


FIGURE 14- 2 TRANSFER DATA FORMAT

14.5 Infrared Data Format Transfer Mode

The TXD pin can output data in the infrared data format (IrDA) by the setting of the IrDA output control register. Setting UARTxCR1 <IRDASEL> to "1" allows the TXD pin to output data in the infrared data format

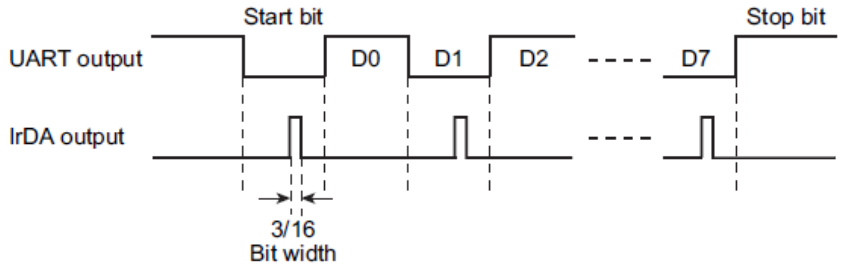


FIGURE 14- 3 Example of Infrared Data Format (Comparison between Normal Output and Ir-DA Output)

14.6 Transfer Baud Rate

he 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 "14.6.1 Transfer baud rate calculation method".

Basic baud rate[baud]	Register	Operating frequency						
		24MHz	16MHz	12MHz	8MHz	4MHz	2MHz	1MHz
128000	UARTxDR[7:0]	0x0A	0x07	0x05	0x03	0x01	0x00	-
	RTSEL[2:0]	0y100	0y011	0y011	0y011	0y011	0y011	-
	Error	(+0.27%)	(+0.81%)	(+0.81%)	(+0.81%)	(+0.81%)	(+0.81%)	-
115200	UARTxDR[7:0]	0x0C	0x08	0x06	-	-	-	-
	RTSEL[2:0]	0y000	0y011	0y010	-	-	-	-
	Error	(+0.16%)	(-0.44%)	(-0.79%)	-	-	-	-
76800	UARTxDR[7:0]	0x12	0x0C	0x09	0x06	-	-	-
	RTSEL[2:0]	0y001	0y000	0y011	0y010	-	-	-
	Error	(-0.32%)	(+0.16%)	(+0.81%)	(-0.79%)	-	-	-
62500	UARTxDR[7:0]	0x17	0x0F	0x0B	0x07	0x03	0x01	0x00
	RTSEL[2:0]	0y000	0y000	0y000	0y000	0y000	0y000	0y000
	Error	0%	0%	0%	0%	0%	0%	0%
57600	UARTxDR[7:0]	0x19	0x11	0x0C	0x08	-	-	-
	RTSEL[2:0]	0y000	0y011	0y000	0y011	-	-	-
	Error	(+0.16%)	(-0.44%)	(+0.16%)	(-0.44%)	-	-	-
38400	UARTxDR[7:0]	0x26	0x19	0x12	0x0C	0x06	-	-
	RTSEL[2:0]	0y000	0y000	0y001	0y000	0y010	-	-
	Error	(+0.16%)	(+0.16%)	(-0.32%)	(+0.16%)	(-0.79%)	-	-
19200	UARTxDR[7:0]	0x4D	0x30	0x26	0x19	0x0C	0x06	-
	RTSEL[2:0]	0y000	0y100	0y000	0y000	0y000	0y010	-
	Error	(+0.16%)	(+0.04%)	(+0.16%)	(+0.16%)	(+0.16%)	(-0.79%)	-
9600	UARTxDR[7:0]	0x92	0x64	0x4D	0x30	0x19	0x0C	0x06
	RTSEL[2:0]	0y100	0y001	0y000	0y100	0y000	0y000	0y010
	Error	(+0.04%)	(+0.01%)	(+0.16%)	(+0.04%)	(+0.16%)	(+0.16%)	(-0.79%)
4800	UARTxDR[7:0]	-	0xC9	0x92	0x64	0x30	0x19	0x0C
	RTSEL[2:0]	-	0y001	0y100	0y001	0y100	0y000	0y000
	Error	-	(+0.01%)	(+0.04%)	(+0.01%)	(+0.04%)	(+0.16%)	(+0.16%)

Basic baud rate[baud]	Register	Operating frequency						
		24MHz	16MHz	12MHz	8MHz	4MHz	2MHz	1MHz
2400	UARTxDR[7:0]	-	-	-	0xCF	0x67	0x33	0x19
	RTSEL[2:0]	-	-	-	0y000	0y000	0y000	0y000
	Error	-	-	-	(+0.16%)	(+0.16%)	(+0.16%)	(+0.16%)
1200	UARTxDR[7:0]	-	-	-	-	0xCF	0x67	0x33
	RTSEL[2:0]	-	-	-	-	0y000	0y000	0y000
	Error	-	-	-	-	(+0.16%)	(+0.16%)	(+0.16%)

Basic baud rate[baud]	Register	Operating frequency
		32.768kHz
300	UARTxDR[7:0]	0x06
	RTSEL[2:0]	0y011
	Error	(+0.67%)
150	UARTxDR[7:0]	0x0D
	RTSEL[2:0]	0y011
	Error	(+0.67%)
134	UARTxDR[7:0]	0x0E
	RTSEL[2:0]	0y001
	Error	(-1.20%)
110	UARTxDR[7:0]	0x11
	RTSEL[2:0]	0y001
	Error	(+0.30%)
75	UARTxDR[7:0]	0x1C
	RTSEL[2:0]	0y010
	Error	(+0.44%)

14.6.1 Transfer Baud Rate Calculation Method

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												Gene fsysclk ud rate
000	16	16	16	16	16	16	16	16	16	16	16	16	$\frac{fcgck}{16\times(U fsysclk\ )}$ [baud]
001	16	17	16	17	16	17	16	17	16	17	16	17	$\frac{fcgck}{16.5\times(U fsysclk\ 1)}$ [baud]
010	15	15	15	15	15	15	15	15	15	15	15	15	$\frac{fcgck}{15\times(UA fsysclk\ )}$ [baud]
011	15	16	15	16	15	16	15	16	15	16	15	16	$\frac{fc fsysclk}{15.5\times(UA 1)}$ [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 14- 4 FINE ADJUSTMENT OF BAUD RATE CLOCK USING UART11R2 <RTSEL>

The bit width of transmitted/received data can be finely adjusted by changing UART1CR2 <RTSEL>. The number of RT clocks per bit can be changed in a range of 15 to 17 clocks by changing UART1CR2<RTSEL>. The RT clock is the transfer base clock, which is the pulses obtained by counting the clock selected at UART1CR1<BRG> the number of times of (UART1DR set value) + 1. Especially, when UART1CR2 <RTSEL> is set to "0y001" or "0y011", two types of RT clocks alternate at each bit, so that the pseudo baud rates of RT × 15.5 clocks and RT × 16.5 clocks can be generated. The number of RT clocks per bit of transfer frame is shown in Figure 14-4.

For example, when fsysclk is 16 [MHz], UART0CR2<RTSEL> is set to "000" and UART0DR is set to "0xoC", the baud rate calculated using the formula in Figure 14-4 is expressed as: fsysclk / (16 × (UART0DR + 1) = 76923 [baud]

These settings generate a baud rate close to 76800 [baud] (+0.16%).

Calculation of Set Values of UARTxCR2 <RTSEL> and UARTxDR

The set value of UARTxDR for an operating frequency and baud rate can be calculated using the calculation formula shown in Figure 14.5. For example, to generate a basic baud rate of 38400 [baud] with fsysclk=16[MHz], calculate the set value of UARTxDR for each setting of UARTxCR2 <RTSEL> and compensate the calculated value to a positive number to obtain the generated baud rate as shown in Figure 14-6. Basically, select the set value of UARTxCR2 <RTSEL> that has the smallest baud rate error from among the generated baud rates. In Figure 14-5, the setting of UARTxCR2 <RTSEL>="000" has the smallest error among the

calculated baud rates, and thus the generated baud rate is 38462 [baud] (+0.16%) against the basic baud rate of 38400 [baud].

RTSEL	UARTxDR
000	$UARTxDR = \frac{fsysclk\ (Hz)}{16 \times A\ (baud)} - 1$
001	$UARTxDR = \frac{fsysclk\ (Hz)}{16.5 \times A\ (baud)} - 1$
010	$UARTxDR = \frac{fsysclk\ (Hz)}{15 \times A\ (baud)} - 1$
011	$UARTxDR = \frac{fsysclk\ (Hz)}{15.5 \times A\ (baud)} - 1$
100	$UARTxDR = \frac{fsysclk\ (Hz)}{17 \times A\ (baud)} - 1$

table 14- 4 UART0DR Calculation Method (When BRG Is Set to fsysclk)

RTSEL	UARTxDR	Baud rate
000	$UARTxDR = \frac{16000000\ (Hz)}{16 \times 38400(baud)} - 1 \approx 25$	$\frac{16000000\ (Hz)}{16 \times (25+1)}$ = 38462 baud (+0.16%)
001	$UARTxDR = \frac{16000000\ (Hz)}{16.5 \times 38400(baud)} - 1 \approx 24$	$\frac{16000000\ (Hz)}{16.5 \times (24+1)}$ = 38788 baud (+1.01%)
010	$UARTxDR = \frac{16000000\ (Hz)}{15x\ 38400(baud)} - 1 \approx 26$	$\frac{16000000\ (Hz)}{15 \times (26+1)}$ = 39506 baud (+2.88%)
011	$UARTxDR = \frac{16000000\ (Hz)}{15.5 \times 38400(baud)} - 1 \approx 25$	$\frac{16000000\ (Hz)}{15.5 \times (25+1)}$ = 39702 baud (+3.39%)
100	$UARTxDR = \frac{16000000\ (Hz)}{17 \times 38400(baud)} - 1 \approx 24$	$\frac{16000000\ (Hz)}{17 \times (24+1)}$ = 37647 baud (-1.96%)

TABLE 14- 5 EXAMPLE OF UART0DR CALCULATION

Note: The error from the basic baud rate should be within the frequency difference of the clock source. For the frequency difference specifications of each clock source, please refer to "3.2.2 Clock Source". Even if the error is within the frequency difference of the clock source, the UART communication may fail due to the frequency error of the external control device (such as a personal computer) and the oscillation crystal and load capacitance of the communication pin.

14.7 Data Sampling Method

The sampling methods of each UART channel are the same. The following uses UART0 as an example to explain.

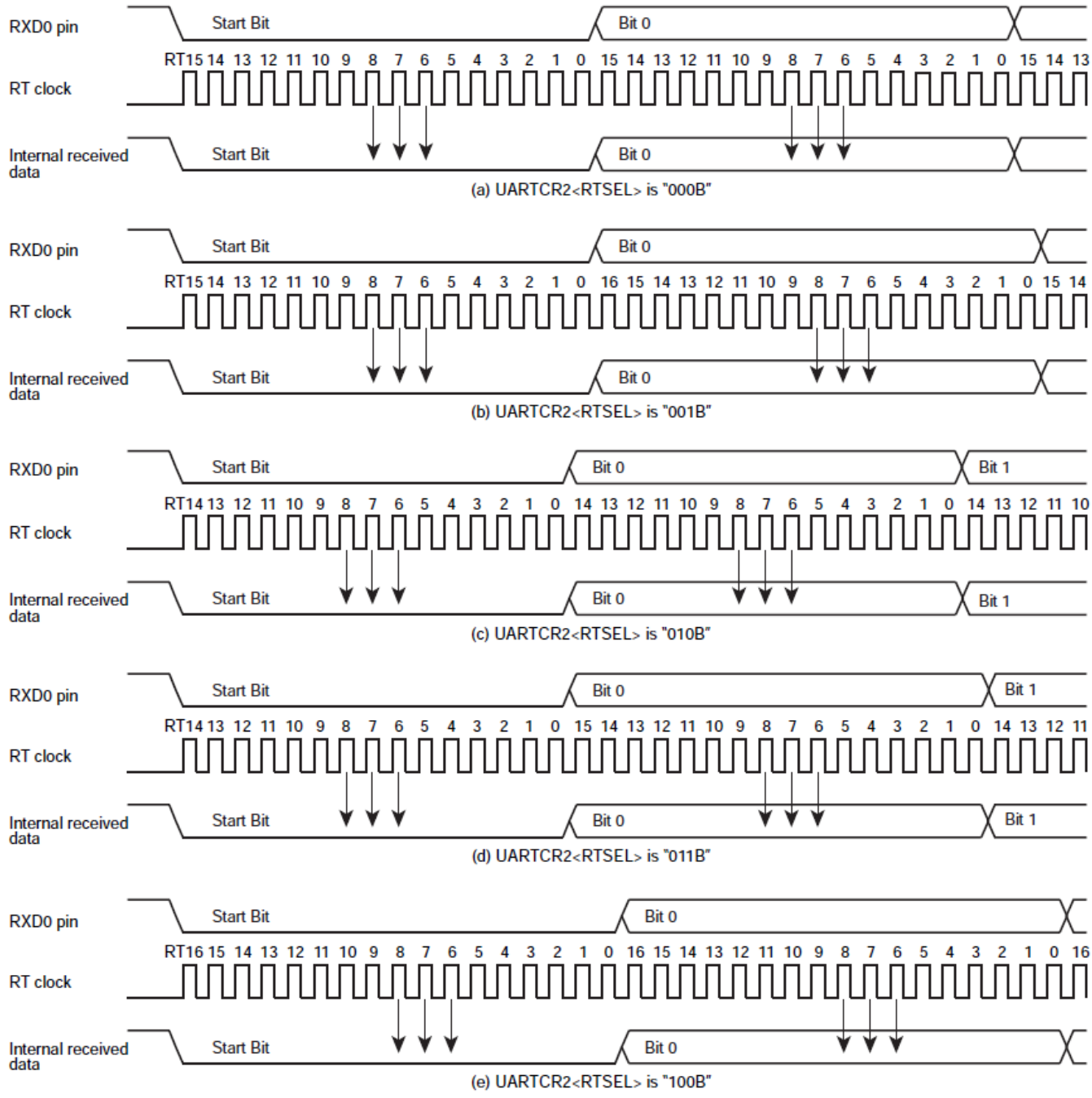


FIGURE 14- 5 DATA SAMPLING IN EACH CASE OF UARTCR2 <RTSEL>

The UART receive control circuit starts RT clock counting when it detects a falling edge of the input pulses to the RXD0 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 14-5).

If '1' is detected in the sampling of the start bit due to noise and other factors, the count of the RT clock will stop and the data reception will be terminated. Then, when RXD0 detects the falling edge of the input pulse again, The count of the RT clock will restart and the data reception will resume from the start bit.

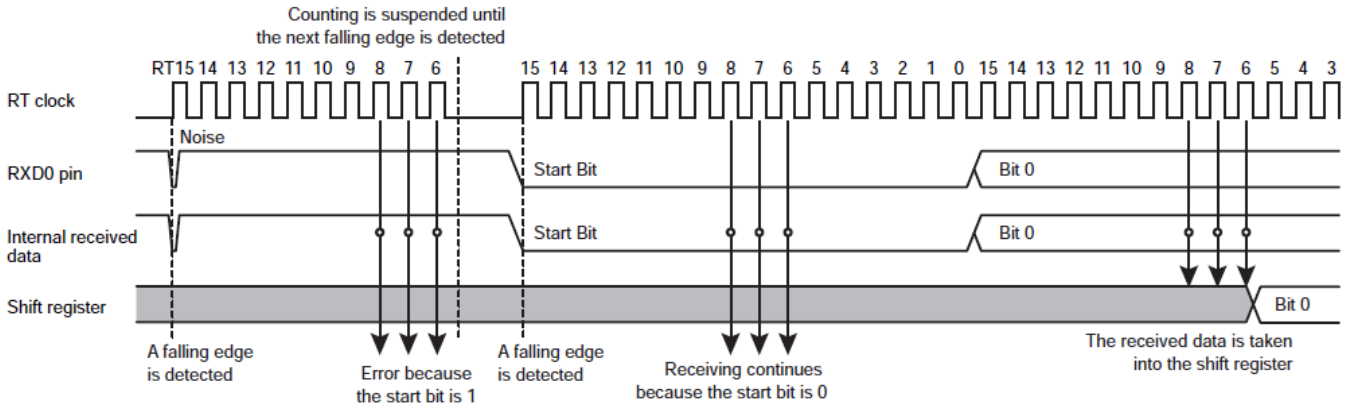


FIGURE 14- 6 START BIT SAMPLING

14.8 Received Data Noise Rejection

When noise rejection is enabled at UART0CR2 <RXDNC>, the time of pulses to be regarded as signals is as shown in Table as below.

RXDNC	Noise rejection time [s]	Time of pulses to be regarded as signals
00	No noise rejection	-
01	(UART0DR+1)/(Transfer base clock frequency)	2 × (UART0DR+1)/(Transfer base clock frequency)
10	2 × (UART0DR+1)/(Transfer base clock frequency)	4 × (UART0DR+1)/(Transfer base clock frequency)
11	4 × (UART0DR+1)/(Transfer base clock frequency)	8 × (UART0DR+1)/(Transfer base clock frequency)

TABLE 14- 6 RECEIVED DATA NOISE REJECTION TIME

NOTE： The transfer base clock frequency is the clock frequency selected at UARTCR1<BRG>.

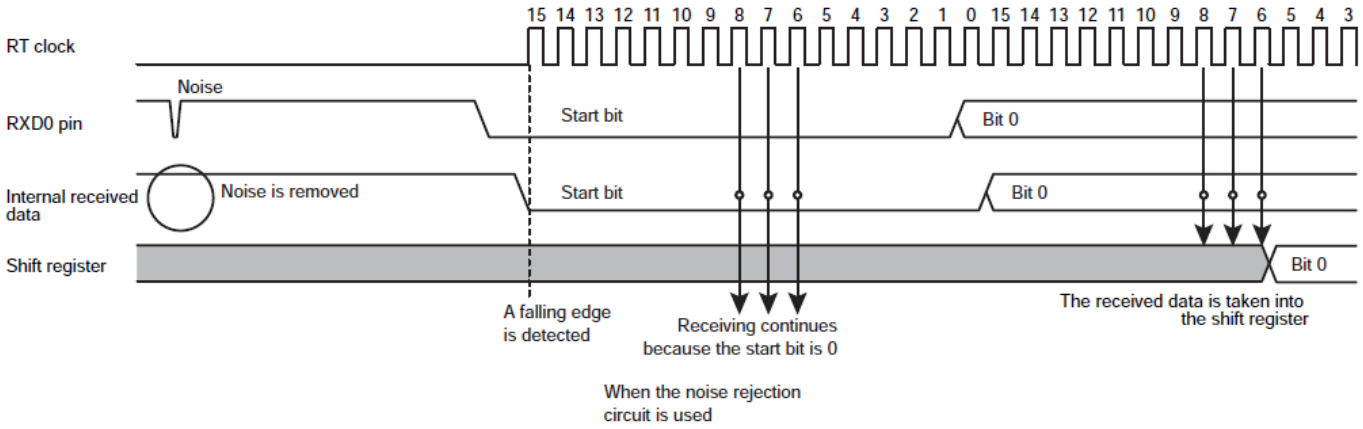


FIGURE 14- 7 RECEIVED DATA NOISE REJECTION

14.9 Transmit/Receive Operation

14.9.1 Data Transmit Operation

Set UART0CR1 <TXE> to "1". Check UART0SR <TBFL> = "0", and then write data into TD0BUF (transmit data buffer). Writing data into TD0BUF sets UART0SR<TBFL> to "1", transfers the data to the transmit shift register, and outputs the data sequentially from the TXD0 pin. The data output includes a start bit, stop bits whose number is specified in UART0CR1 <STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART0CR1 <BRG>, UART0CR2 <RTSEL> and UART0DR. When data transmission starts, the transmit buffer full flag UART0SR <TBFL> is cleared to "0" and an INTTXD0 interrupt request is generated.

Note 1： After data is written into TD0BUF, if new data is written into TD0BUF before the previous data is transferred to the shift register, the new data is written over the previous data and is transferred to the shift register.
Note 2：Under the conditions shown in Table 14.5, the TXD0 pin output is fixed at the L or H level according to the setting of UART0CR1 <IRDASEL>.

Condition	TDX	
	IRDASEL = "0"	IRDASEL = "1"
When UART0CR1<TXE> = "0"	H level	L level
From when "1" is written to UART0CR1<TXE> to when the transmitted data is written to TD0BUF		

TABLE14- 7 TXD0 PIN OUTPUT

14.9.2 Data Receive Operation

Set UART0CR1 <RXE> to "1". When data is received via the RXD0 pin, the received data is transferred to RD0BUF (receive data buffer). At this time, the transmitted data includes a start bit, stop bit(s) and a parity bit if parity addition is specified. When the stop bit(s) are received, data only is extracted and transferred to RD0BUF (receive data buffer). Then the receive buffer full flag UART0SR <RBFL> is set and an INTRXD0 interrupt request is generated. Set the data transfer baud rate using UART0CR1 <BRG>, UART0CR2 <RTSEL> and UART0DR.

If an overrun error occurs when data is received, the data is not transferred to RD0BUF (receive data buffer) but discarded; data in the RD0BUF is not affected.

14.10 Status Flag

14.10.1 Parity Error

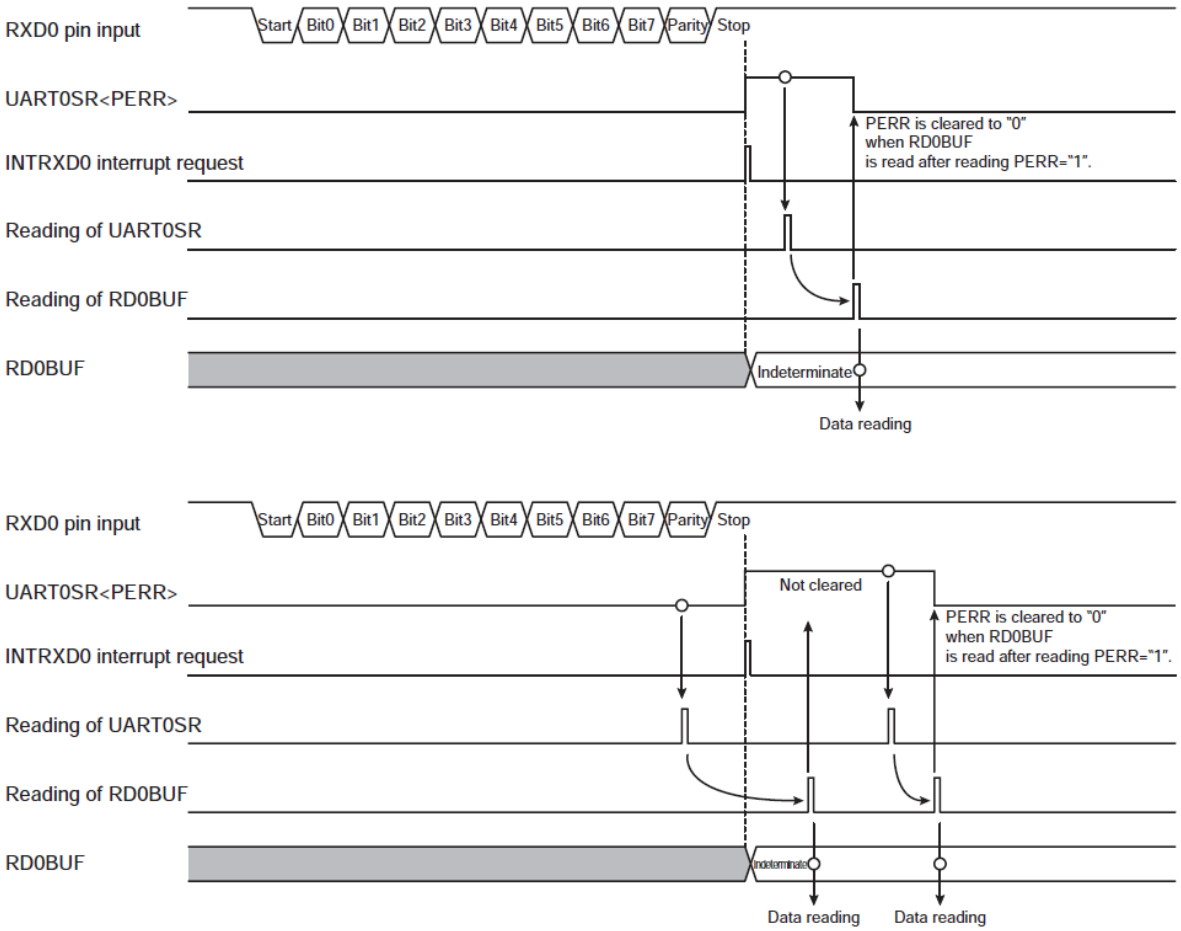


figure 14- 8 Occurrence of Parity Error

When the parity determined using the receive data bits differs from the received parity bit, the parity error flag UART0SR <PERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR <PERR> is "1" when UART0SR is read, UART0SR <PERR> will be cleared to "0" when RD0BUF is read subsequently. (The RD0BUF read value becomes undefined.)

If UART0SR <PERR> is set to "1" after UART0SR is read, UART0SR <PERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <PERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

14.10.2 Framing Error

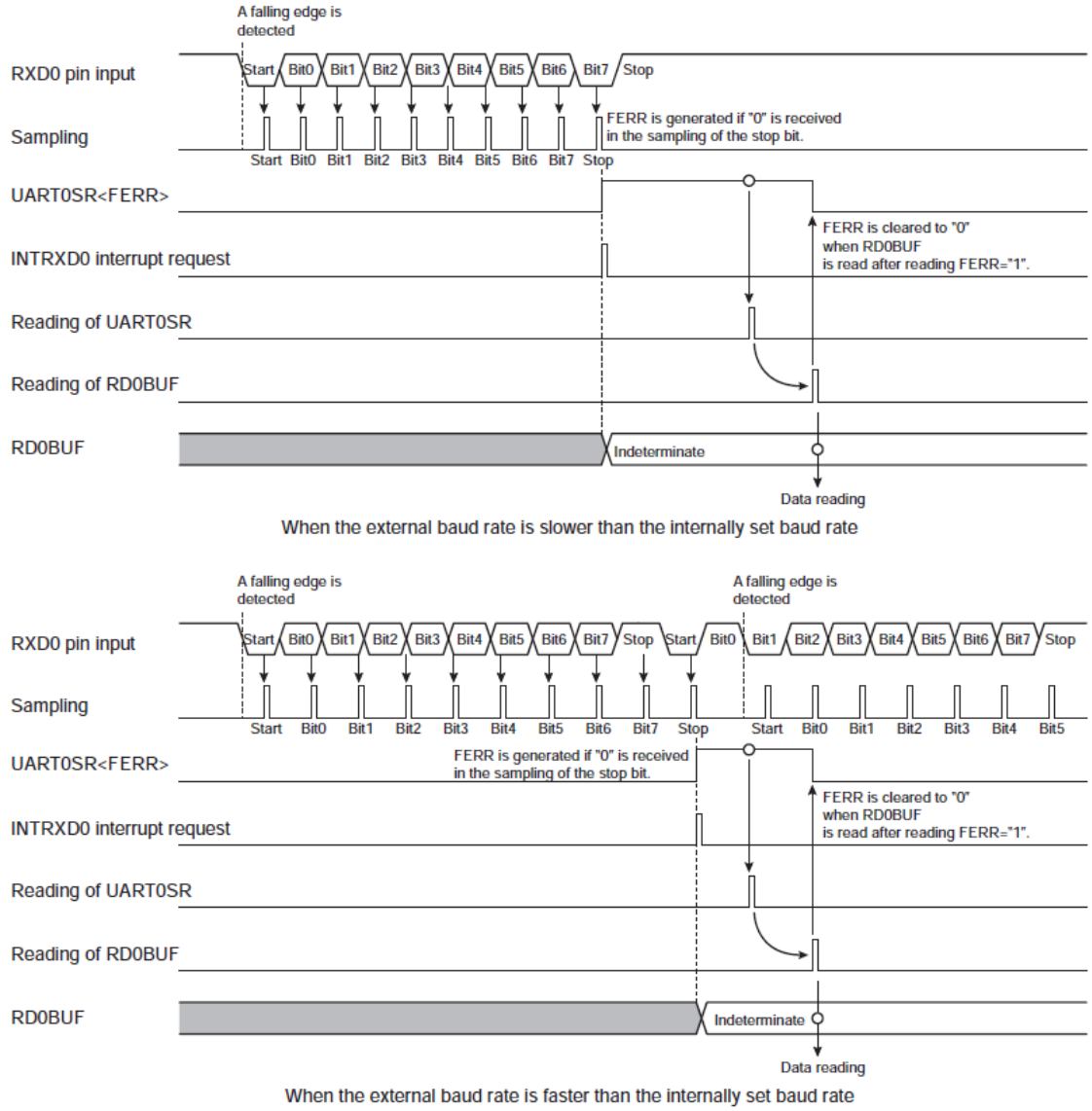


FIGURE 14- 9 OCCURRENCE OF FRAMING ERROR

If the internal and external baud rates differ or "0" is sampled as the stop bit of received data due to the influence of noise on the RXD0 pin, the framing error flag UART0SR <FERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR <FERR> is "1" when UART0SR is read, UART0SR <FERR> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR <FERR> is set to "1" after UART0SR is read, UART0SR <FERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <FERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

14.10.3 Overrun Error Flag

If receiving of all data bits is completed before the previous received data is read from RD0BUF, the overrun error flag UART0SR <OERR> is set to "1" and an INTRXD0 interrupt request is generated. The data received at the occurrence of the overrun error is discarded and the previous received data is maintained. Subsequently, if data is received while UART0SR <OERR> is still "1", no INTRXD0 interrupt request is generated, and the received data is discarded. FIGURE 14- 10

Note that parity or framing errors in the discarded received data cannot be detected. (These error flags are not set.) That is to say, if these errors are detected together with an overrun error during the reading of UART0SR, they have occurred in the previous received data (the data stored in RD0BUF). FIGURE 14- 11

If UART0SR <OERR> is "1" when UART0SR is read, UART0SR <OERR> will be cleared to "0" whenRD1BUF is read subsequently. FIGURE 14- 12

If UART0SR <OERR> is set to "1" after UART0SR is read, UART0SR <OERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <OERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read. figure 14- 12

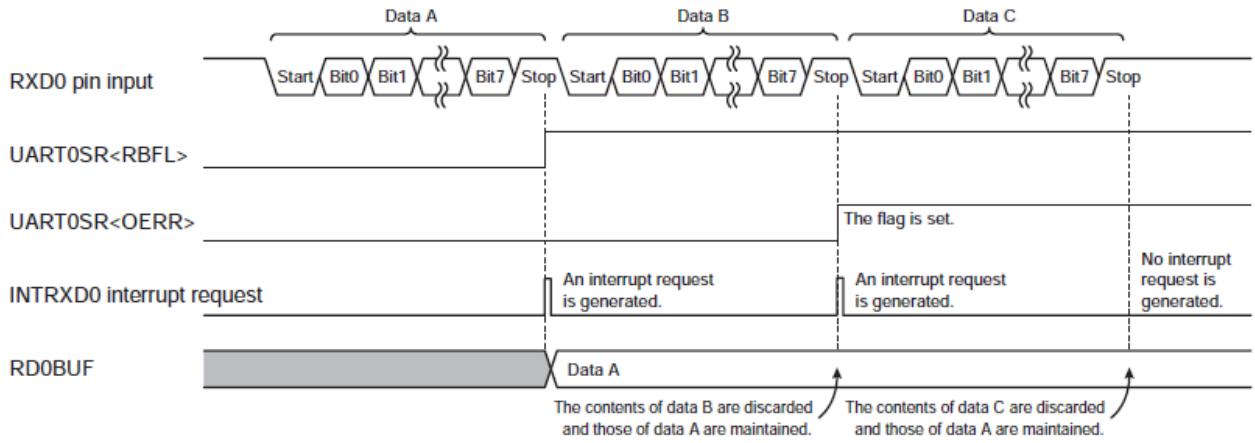
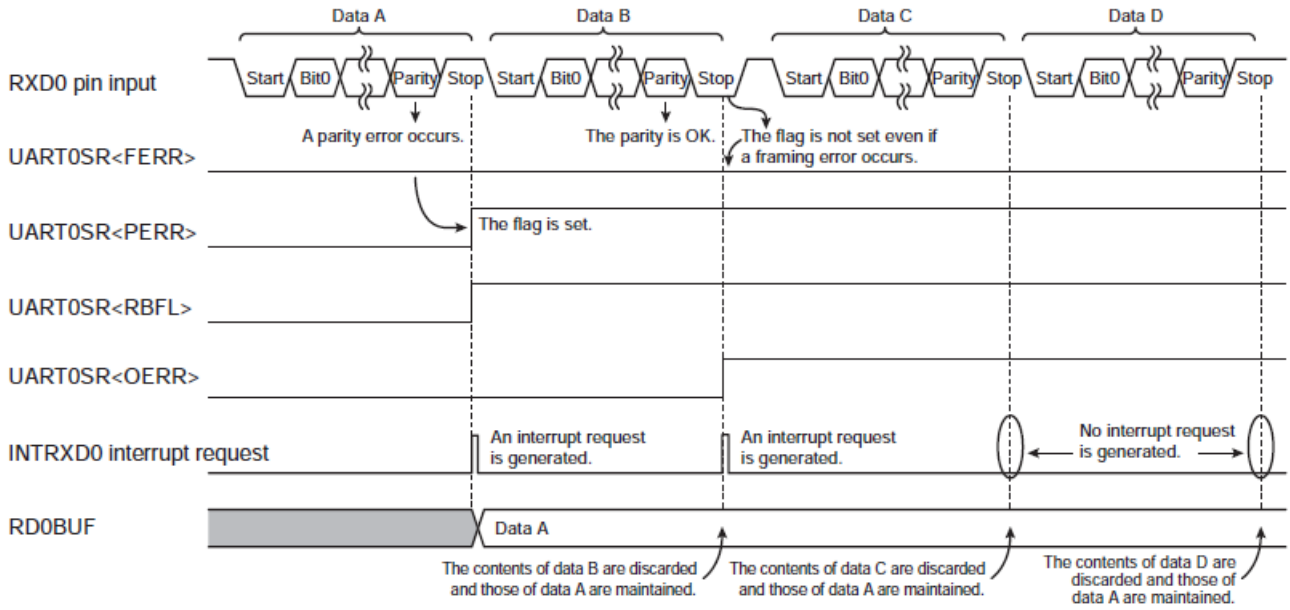
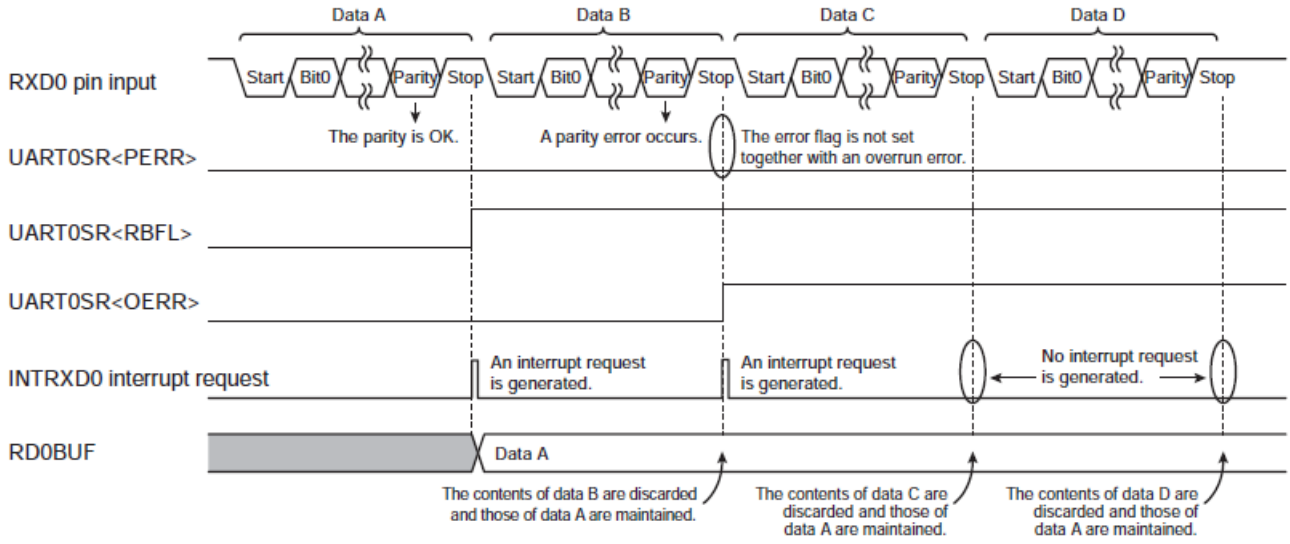


FIGURE 14- 10 GENERATION OF INTRXD0 INTERRUPT REQUEST



When a parity error occurs in the first received data and a framing error occurs in the second data



When a parity error occurs in the second received data

FIGURE 14- 11 FRAMING/PARITY ERROR FLAGS WHEN AN OVERRUN ERROR OCCURS

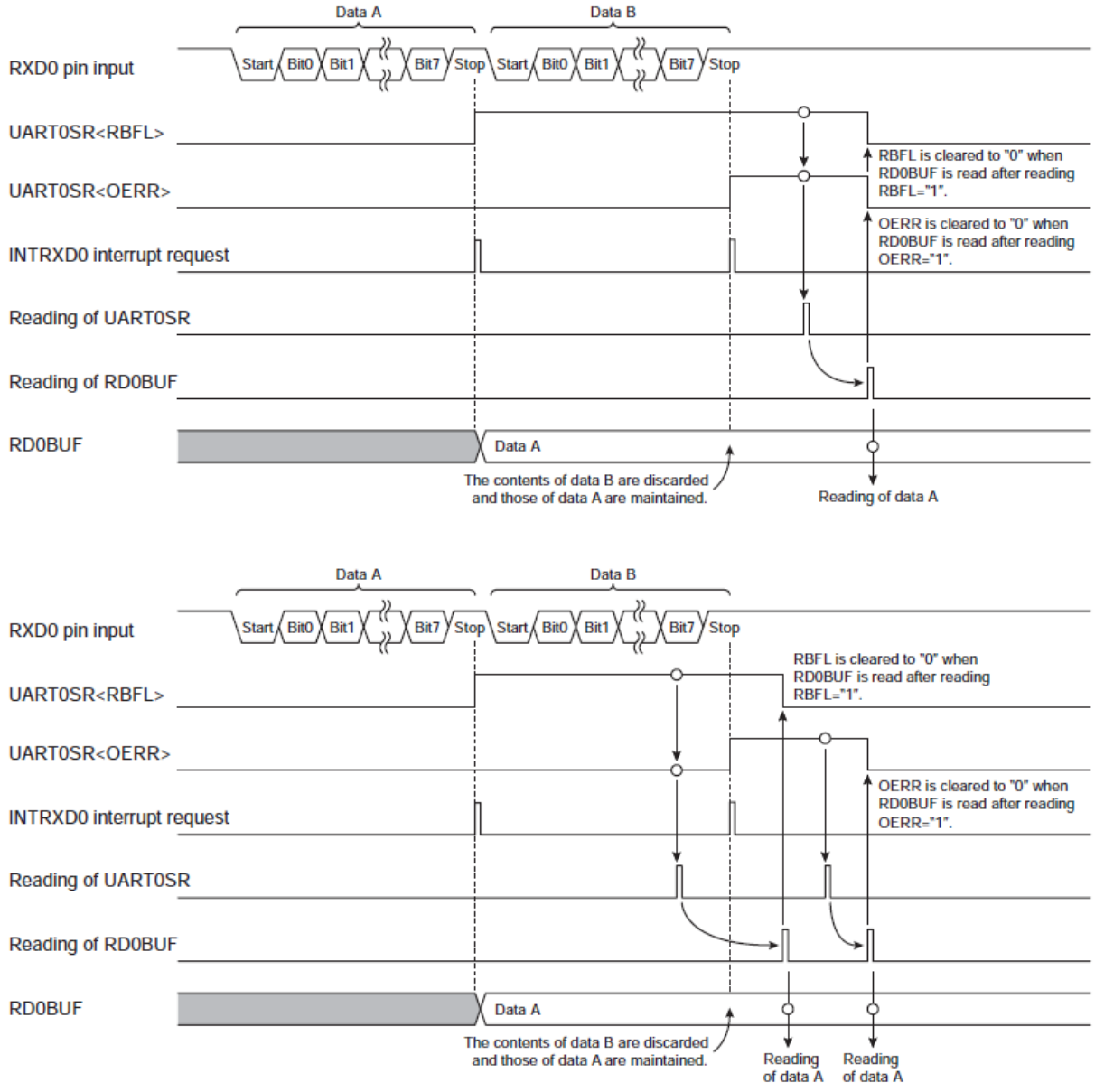


FIGURE 14- 12 CLEARANCE OF OVERRUN ERROR FLAG

14.10.4 Receive Data Buffer Full

Loading the received data in RD0BUF sets UART0SR <RBFL> to "1".

If UART0SR <RBFL> is "1" when UART0SR is read, UART0SR <RBFL> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR <RBFL> is set to "1" after UART0SR is read, UART0SR <RBFL> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR <RBFL> will be cleared to "0" when UART0SR is read again and RD0BUF is read

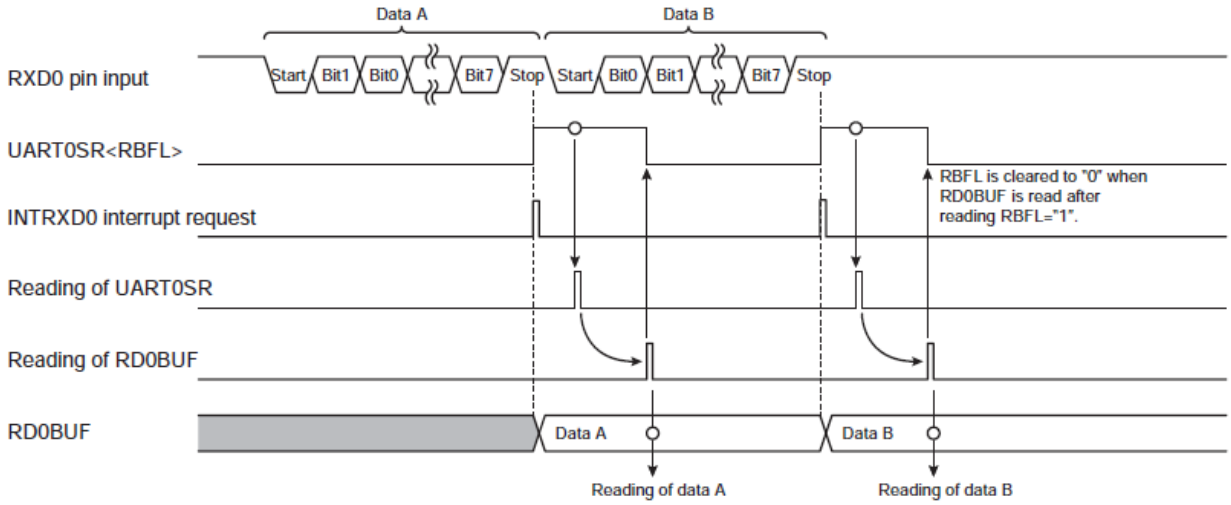


FIGURE 14- 13 OCCURRENCE OF RECEIVE DATA BUFFER FULL

14.10.5 Transmit Busy Flag

If transmission is completed with no waiting data in TD0BUF (when UART0SR <TBFL>="0"), UART0SR <TBSY> is cleared to "0". When transmission is restarted after data is written into TD0BUF, UART0SR <TBSY> is set to "1". At this time, an INTTXD0 interrupt request is generated.

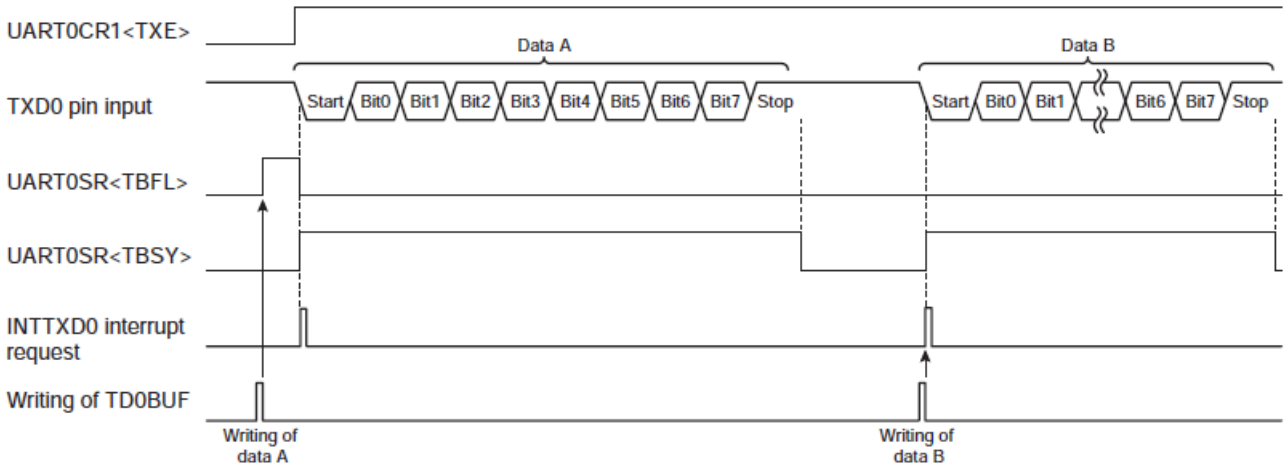


figure 14- 14 Transmit Busy Flag and Occurrence of Transmit Buffer Full

14.10.6 Transmit Buffer Full

When TD0BUF has no data, or when data in TD0BUF is transferred to the transmit shift register and transmission is started, UART0SR <TBFL> is cleared to "0". At this time, an INTTXD0 interrupt request is generated.

Writing data into TD0BUF sets UART0SR <TBFL> to "1".

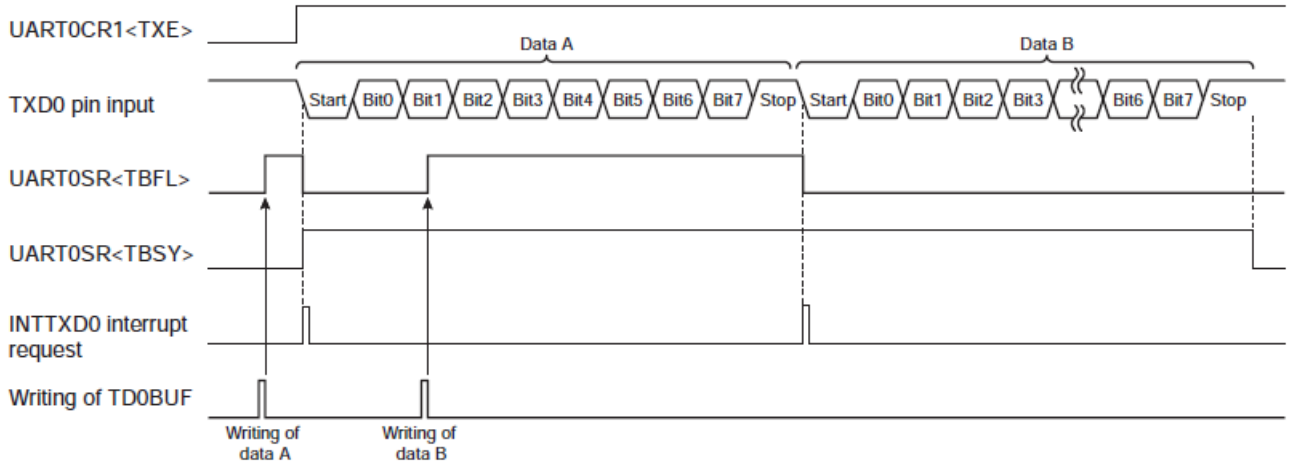


FIGURE 14- 15 OCCURRENCE OF TRANSMIT BUFFER FULL

14.11 Receiving Process

The figure 14-16 shows an example of the receiving process. Details of flag judgments in the processing are shown in Table 14-8 and Table 14-9 .

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 RD0BUF and receiving the data again.

If any overrun error is detected, the receiving of one or more pieces of data is unfinished. It is impossible to determine the number of pieces of data that could not be received. Execute the error handling, for example, by receiving data again from the beginning of the transfer. Basically, an overrun error occurs when the internal software processing cannot follow the data transfer speed. It is recommended to slow the transfer baud rate or modify the software to execute flow control.

Note): If multiple interrupts are used in the INTRXD0 interrupt subroutine, the interrupt should be enabled after reading UART1SR and RD1BUF.

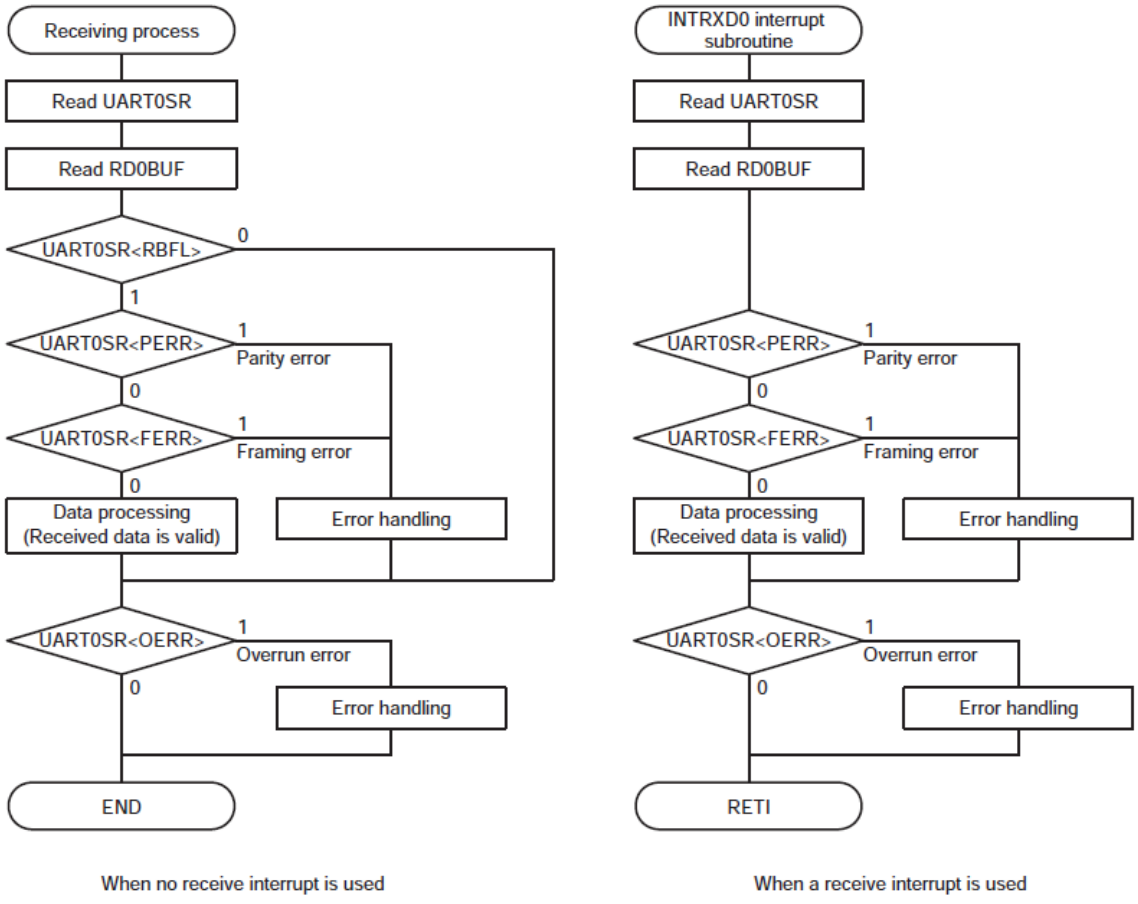


FIGURE 14- 16 EXAMPLE OF RECEIVING PROCESS

RBFL	FERR/PERR	OERR	State
0	-	0	Data has not been received yet.
0	-	1	Some pieces of data could not be received during the previous data receiving process (Receiving of next data is completed in the period from when UART0SR is read to when RD0BUF is read in the previous data receiving process.)
1	0	0	Receiving has been completed properly.
1	0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	1	0	Received data has erroneous value(s).
1	1	1	Received data has erroneous value(s) and some pieces of data could not be received.

TABLE 14- 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 14- 9 FLAG JUDGMENTS WHEN A RECEIVE INTERRUPT IS USED

15. Serial Bus Interface(SBI)/I2C

SQ7613 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.

15.1 Communication Format

15.1.1 I2C bus

The I2C bus is connected to devices via the SDA and SCL pins and can communicate with multiple devices.

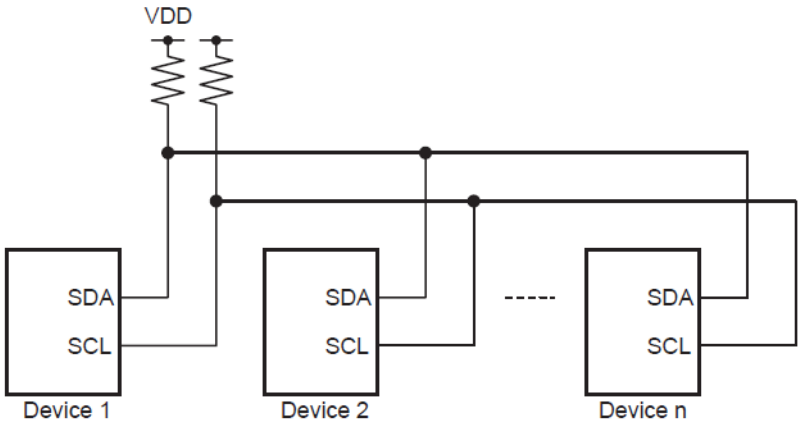


FIGURE 15- 1 DEVICE CONNECTIONS

Communications are implemented between a master and slave.

The master transmits the start condition, the slave addresses, the direction bit and the stop condition to the slave(s) connected to the bus, and transmits and receives data. The slave detects these conditions transmitted from the master by the hardware, and transmits and receives data. The data format of the I2C bus that can communicate via the serial bus interface is shown in the figure 15-2 as below.

The serial bus interface does not support the following functions among those specified by the I2C bus standards:

1. Start byte
2. 10-bit addressing
3. SDA and SCL pins falling edge slope control

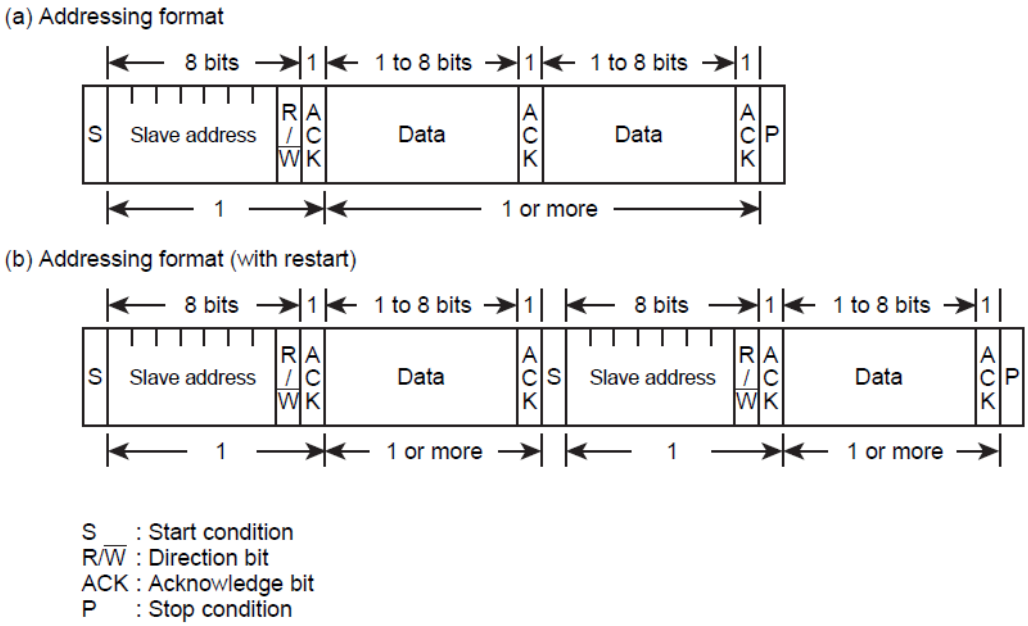


FIGURE 15- 2 DATA FORMAT IF I2C BUS

15.1.2 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.

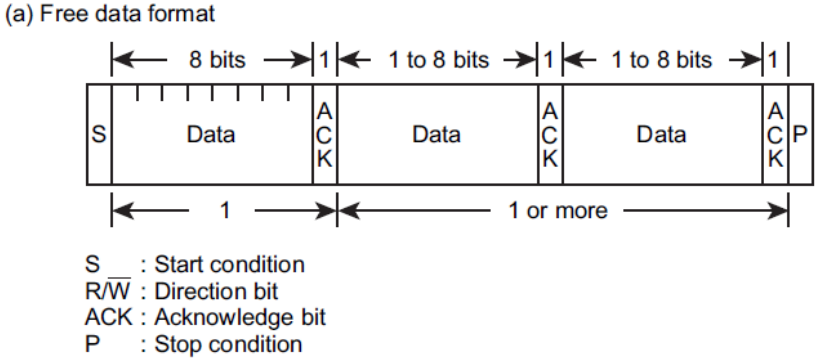


FIGURE 15- 3 FREE DATA FORMAT

15.2 Configuration

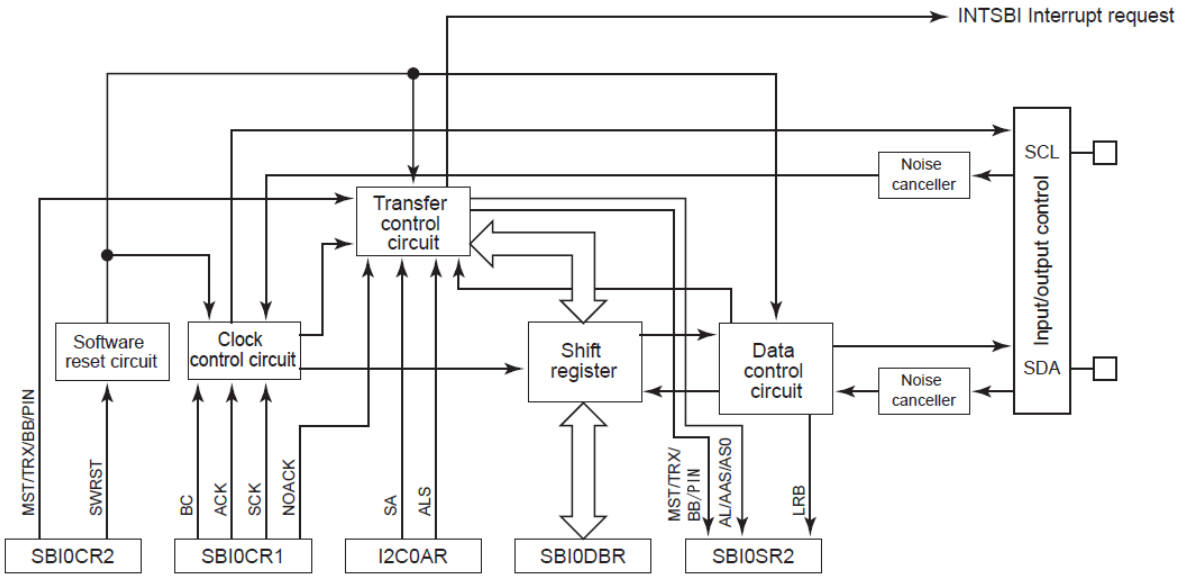


FIGURE 15- 4 SERIAL BUS INTERFACE0 (SBI0)

15.3 Control

The following registers are used to control the serial bus interface and monitor the operation status.

- Serial bus interface control register 1 ((SBIxCR1, x=0~1)
- Serial bus interface control register 2 (SBIxCR2, x=0~1)
- Serial bus interface status register (SBIxSR2, x=0~1)
- Serial bus interface data buffer register (SBIxBR, x=0~1)
- I2C bus address register (I2CxAR, , x=0~1)

In addition, the serial bus interface has Peripheral Clock Enable Register2 (PCKEN2) that save power when the serial bus interface is not being used.

地址	Register	Description
0x017A	PCKEN2	Peripheral Clock Enable Register 2
0x00B8	SBI0CR1	Serial bus interface control register 1
0x00B9	SBI0CR2	Serial bus interface control register 2
0x00BA	SBI0SR	Serial bus interface status register
0x00BB	I2C0AR	I2C bus address register
0x00BC	SBI0DBR	Serial bus interface data buffer register
0x00BD	SBI1CR1	Serial bus interface 1 control register 1
0x00BE	SBI1CR2	Serial bus interface 1 control register 2
0x00BF	SBI1SR	Serial bus interface 1 status register
0x00C0	I2C1AR	I2C bus 1 address register
0x00C1	SBI1DBR	Serial bus interface 1 data buffer register

The above table is the register address, and the description of each register. The setting of SBI/I2C channel 0~channel 1 is the same, so the registers are all described by the common symbol x (x=0, 1).

Peripheral Circuit clock Enable Register 2(PCKEN2)

PCKEN2	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	SIO1	SIO0	reserved	reserved	I2C1	I2C0
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

SIO1	SIO1 enable control	0: Disable 1: Enable
SIO0	SIO0 enable control	0: Disable 1: Enable
I2C1	I2C1 enable control	0: Disable 1: Enable
I2C0	I2C0 enable control	0: Disable 1: Enable

Note：When I2CxEN is cleared to "0", the clock providing the serial bus interface will stop. At the same time, the data written to the serial bus interface control register will be invalid. When the serial bus interface is used, set I2CxEN to "1". Then the data is written to the serial bus interface control register.

Serial bus interface control register 1 (SBIxCR1) , x=0,1

SBIxCR1	7	6	5	4	3	2	1	0
Bit Symbol	BC[2:0]			ACK	NOACK	SCK[2:0]		
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[2:0]	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	Numberof 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 receive is finished (non-acknowledgement mode)		Generate an interrupt request when the data receive is finished (non-acknowledgement mode)	
		1:	Generate the clocks for an acknowledge signal and an interrupt request when the data receive is finished (acknowledgement mode)		Count the clocks for an acknowledge signal and generate an interrupt request when the data receive 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 slae address match detection and the GENERAL CALL detection	
SCK[2:0]	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/fsysclk)	t_{LOW}(n/fsysclk)	fscl@fsysclk=24MHz	
			m	n		
		000	*9*	*12*	*1143KHz*	
		001	*11*	*14*	*960KHz*	
		010	*15*	*18*	*727KHz*	
		011	*23*	*26*	*490KHz*	
		100	*39*	*42*	*296KHz*	
		101	*71*	*74*	*166KHz*	
		110	*135*	*138*	*88KHz*	
		111	*263*	*266*	*45KHz*	

Note 1: fsysclk = Gear clock [Hz], ficlk= 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 DEEP SLEEP, SLEEP or NORMAL mode(slow clock), the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Note 5: When fsysclk 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 Register 2 (SBIxCR2) , x=0,1

SBIxCR2	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	SBIM	-	SWRST[1:0]	
Read/Write	W	W	W	W	W	R	W	W
After reset	0	0	0	1	0	0	0	0

MST	Master/slave selection	0:Slave 1:Master
TRX	Transmitter/receiver selection	0:Receiver 1:Transmitter
BB	Start/stop generation	0:Generate the stop condition(when MST、TRX and PIN are" 1") 1: Generate the start condition (when MST、TRX and PIN are" 1")
PIN	Cancel interrupt service request	0:- (cannot clear this bit by software) 1: Cancel interrupt service request
SBIM	Serial bus interface operation mode register	0: Port mode 1:Serial bus interface mode
SWRST[1:0]	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 DEEP SLEEP, SLEEP 0 or NORMAL mode(slow clock), the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Note 7: SBIxCR2 [2] reset value must be 0.

Serial Bus Interface Status Register (SBIxSR) , x=0,1

SBIxSR	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	1	0	0	0	*

MST	Master/slave selection status monitor	0:Slave 1:Master
TRX	Transmitter/receiver selection status monitor	0:Receiver 1:Transmitter
BB	Bus status monitor	0:Bus free 1: Bus busy
PIN	Interrupt service requests status monitor	0:Requesting interrupt service 1:Releasing interrupt service
AL	Arbitration lost detection monitor	0: - 1:Aritration lost detected
AAS	Slave address match detection monitor	0: - 1:Detect slave address match or "GENERAL CALL"
AD0	"GENERAL CALL" detection monitor	0: - 1: Detect "GENERAL CALL"
LRB	Last received bit monitor	0: Last received bit is"0" 1: Last received bit is"1"

Note 1 : When SBIxCR2<SBIM> becomes "0", SBI0SR is initialized.

Note 2 : After a software reset is generated, all the bits of the SBIxCR2 register except SBIxCR2<SBIM> and the SBIxCR1, I2CxAR and SBIxSR2 registers are initialized.

Note 3 : When the operation is switched to DEEP SLEEP, SLEEP or NORMAL mode(slow clock), the SBIxCR2 register, except SBIxCR2<SBIM>, and the SBIxCR1, I2CxAR and SBIxDBR registers are initialized.

I²C Bus Address Register(I2CxAR) , x=0,1

I2CxAR	7	6	5	4	3	2	1	0
Bit Symbol	SA[6:0]							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[6:0]	Slave address setting	Slave address in the slave mode
ALS	Communication format selection	0: I²C bus mode 1: Free data format

Note 1: Don't set I2CxAR<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 SBIxCR2 register except SBIxCR2<SBIM> and the SBIxCR1, I2CxAR and SBIxSR2 registers are initialized.

Note 4: When the operation is switched to fsysclk=LIRC, *the SBI0CR2 register, except SBIxCR2<SBIM>, and the SBIxCR1, I2CxAR and SBIxDBR registers are initialized.*

Serial bus interface data buffer register (SBIxDBR) , x=0,1

SBIxDBR	7	6	5	4	3	2	1	0
Bit Symbol	SBIxDBR[7:0]							
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：Don't set I2CxAR<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 SBIxCR2 register except SBIxCR2<SBIM> and the SBIxCR1, I2CxAR and SBIxSR2 registers are initialized.

Note 4:：When the operation is switched to DEEP SLEEP, SLEEP or NORMAL mode(slow clock), the SBIxCR2 register, except SBIxCR2<SBIM>, and the SBIxCR1, I2CxAR and SBIxDBR registers are initialized.

15.4 Functions

15.4.1 Low power consumption function

The serial bus interface has Peripheral Clock Enable Register2 (PCKEN2) that saves power when the serial bus interface is not being used.

Setting PCKEN2< I2CxEN > 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 PCKEN2< I2CxEN > to "1" enables the basic clock supply to the serial bus interface and makes external interrupts usable.

After reset, PCKEN2< I2CxEN > 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 PCKEN2< I2CxEN > to "1" in the initial setting of the program (before the serial bus interface control registers are operated).

Do not change PCKEN2< I2CxEN > to "0" during the serial bus interface operation, otherwise serial bus interface may operate unexpectedly.

15.4.2 Selecting the slave address match detection and the GENERAL CALL detection

SBIxCR1<NOACK> enables and disables the slave address match detection and the GENERAL CALL detection in the slave mode.

learing SBIxCR1<NOACK> to "0" enables the slave address match detection and the GENERAL CALL detection.

Setting SBI0xCR1<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, SBIxCR1<NOACK> is ignored and has no influence on the operation.

Note：If SBIxCR1<NOACK> is cleared to "0" during data transfer in the slave mode, it remains at "1" and returns an acknowledge signal of data transfer.

15.4.3 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.

15.4.3.1 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.

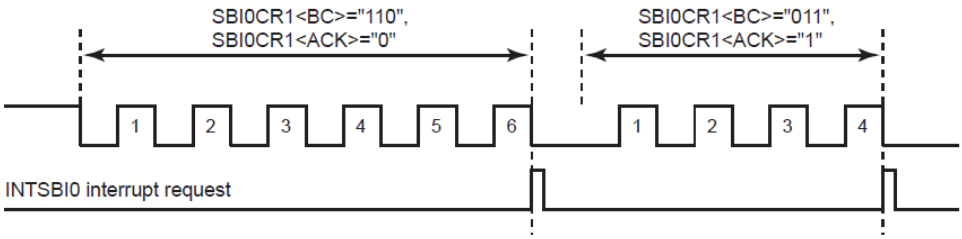


figure 15- 5 Number of clocks for Data transfer and SBI0CR1<BC> and SBI0CR1<ACK>

The relationship between the number of clocks for data transfer and SBI0CR1<BC> and SBI0CR1<ACK> is shown in Table 15.1

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 15- 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.

15.4.3.2 Output of an acknowledge signal

In the acknowledgment mode, the SDAx(x=0~3) pin changes as follows during the period of the clocks for an acknowledge signal.

(a) In the master mode

In the transmitter mode, the SDAx 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 SDAx 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 I2CxAR<SA> is detected or when a GENERAL CALL is received, the SDAx 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 SDAx 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 SDAx pin is pulled down to the low level and an acknowledge signal is generated. Table 15-2 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 15- 2 STATES OF THE SCL0 AND SDA0 PINS IN THE ACKNOWLEDGMENT MODE

15.4.4 Serial clock

15.4.4.1 Clock source

SBIxCR1<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/fsysclk)	t_{LOW}(n/fsysclk)
	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

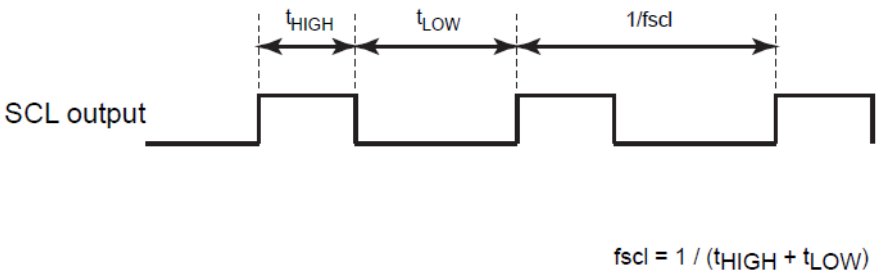


FIGURE 15- 6 SCL OUTPUT

Note: There are cases where the HIGH period differs from tHIGH selected at SBIxCR1<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 tHIGH [s].

When SBIxCR2<PIN> is set to "1" in the slave mode, the time that elapses before the release of the SCL pin is tLOW [s].

In both the master and slave modes, the high level period must be 3/ fsysclk[s] or longer and the low level period must be 5/ fsysclk[s] or longer for the externally input clock, regardless of the SBIxCR1<SCK> setting.

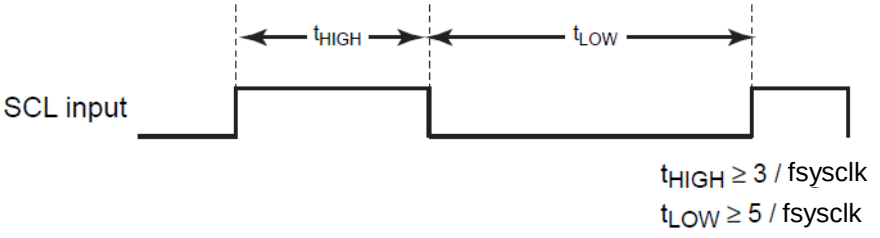


图 15- 7　SCL 输入

15.4.4.2 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.

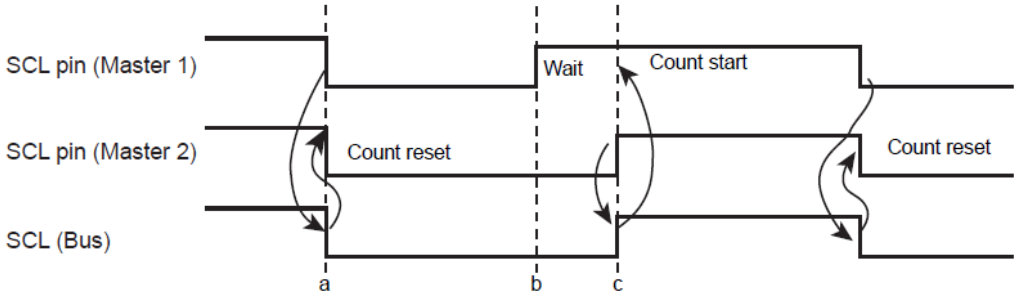


FIGURE 15- 8 EXAMPLE OF CLOCK SYNCHRONIZATION

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.

15.4.5 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.

15.4.6 Transmitter/receiver selection

To set the device as a transmitter, SBI0CR2<TRX> should be set to "1". To set the device as a receiver, SBI0CR2<TRX> should be cleared to "0".

For the I2C bus data transfer in the slave mode, SBI0CR2<TRX> is set to "1" by the hardware if the direction bit (R/W) sent from the master device is "1", and is cleared to "0" if the bit is "0".

In the master mode, after an acknowledge signal is returned from the slave device, SBI0CR2<TRX> is cleared to "0" by hardware if a transmitted direction bit is "1", and is set to "1" by hardware if it is "0".

When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<TRX> is cleared to "0" by the hardware. The table shows SBI0CR2<TRX> changing conditions in each mode and SBI0CR2<TRX> value after changing.

Note：When SBI0CR1<NOACK> is "1", the slave address match detection and the GENERAL CALL detection are disabled, and thus SBI0CR2<TRX> remains unchanged.

Mode	Direction bit	Changing condition	TRX after changing
Slave mode	"0"	A received slave address is the same as the value set to I2CxAR<SA>	"0"
	"1"		"1"
Master mode	"0"	ACK signal is returned	"1"
	"1"		"0"

TABLE 15- 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.

15.4.7 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.

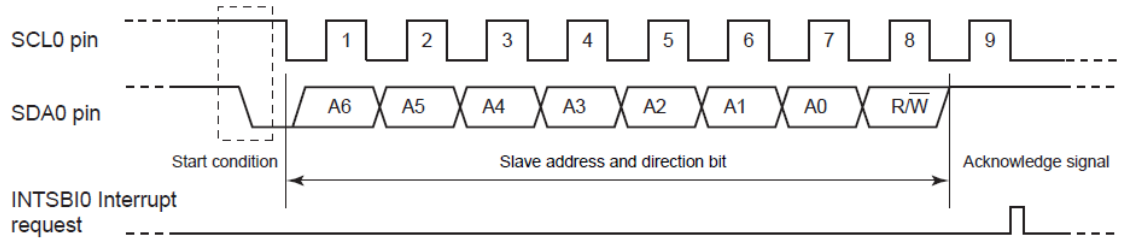


figure15- 9 Generating the start condition and a slave address

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.

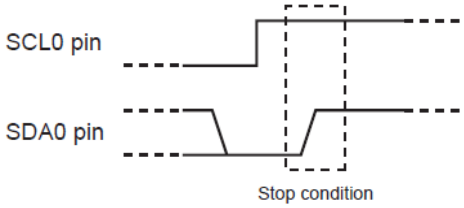


FIGURE 15- 10 STOP CONDITION GENERATION

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).

15.4.8 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 tLOW. Although SBI0CR2<PIN> can be set to "1" by the software, SBI0CR2<PIN> can not be cleared to "0" by the software.

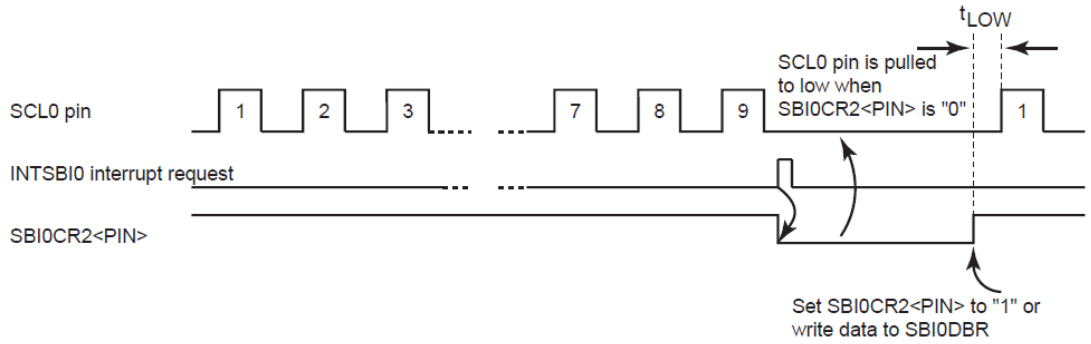


FIGURE 15- 11 S SBI0CR2<PIN> AND SCL0 PIN

15.4.9 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.

15.4.10 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.

15.4.11 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.

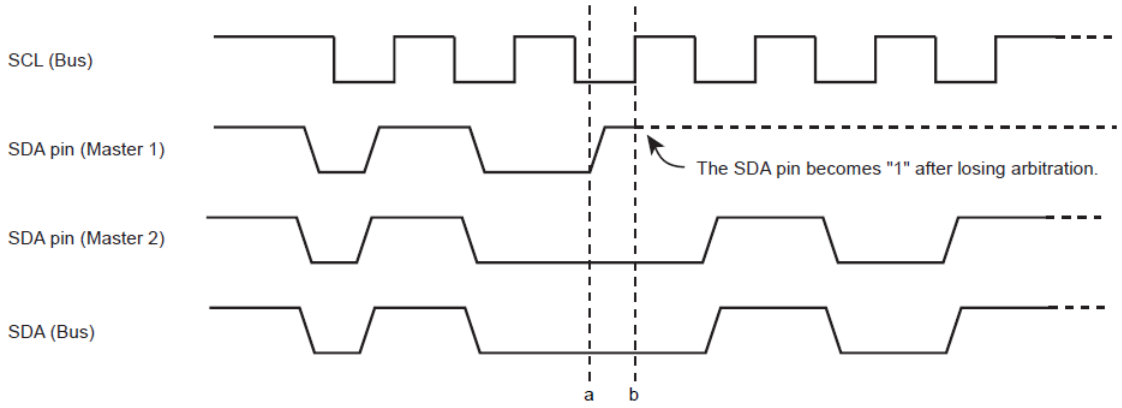


FIGURE 15- 12 ARBITRATION LOST

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.

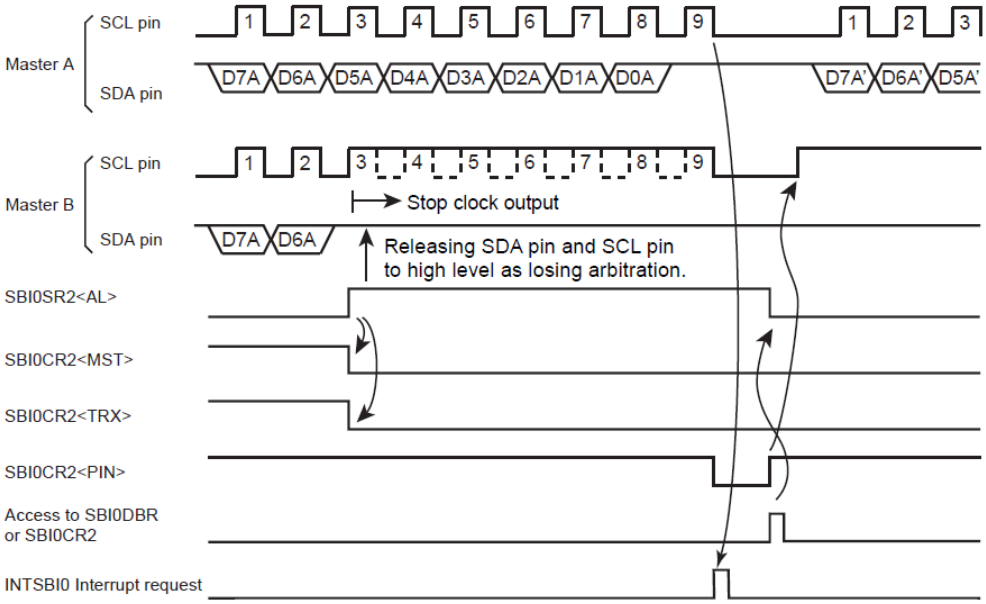


FIGURE 15- 13 EXAMPLE WHEN MASTER B IS A SERIAL BUS INTERFACE CIRCUIT

15.4.12 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.

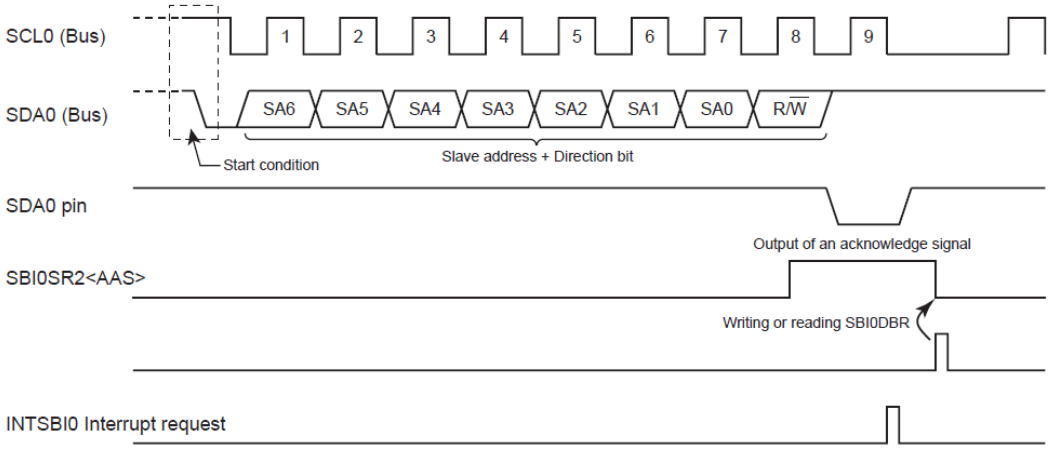


FIGURE 15- 14 CHANGE IN THE SLAVE ADDRESS MATCH DETECTION MONITOR

15.4.13 GENERAL CALL detection monitor

SBI0SR2<AD0> is set to "1" when SBI0CR1<NOACK> is "0" and GENERAL CALL (all 8-bit received data is "0" immediately after a start condition) in a slave mode.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AD0> remains at "0" even if a "GENERAL CALL" is received.

SBI0SR2<AD0> is cleared to "0" when a start or stop condition is detected on a bus.

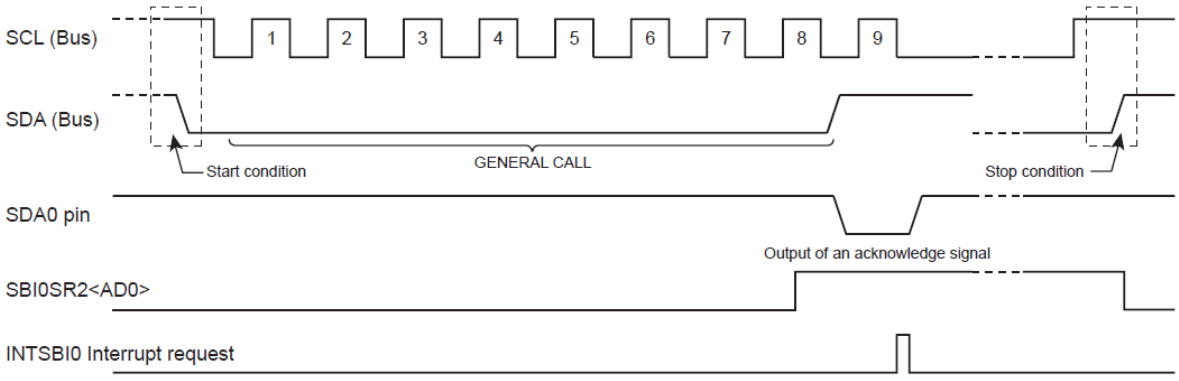


FIGURE 15- 15 CHANGES IN THE GENERAL CALL DETECTION MONITOR

15.4.14 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>.

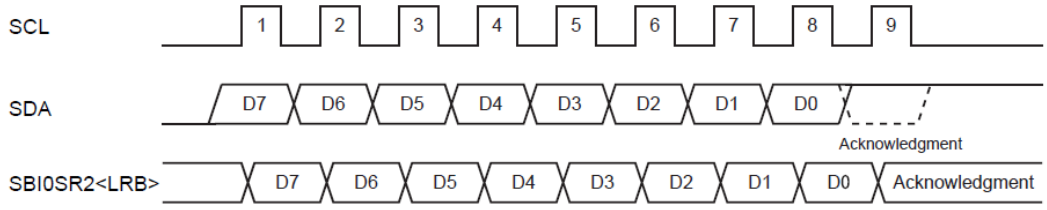


FIGURE 15- 16 CHANGES IN THE LAST RECEIVED BIT MONITOR

15.4.15 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.

15.5 I2C Data transfer of I2C Bus

15.5.1 Device initialization

Set PCKEN2<I2CxEN> 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 tHIGH and tLOW at SBI0CR1<SCK>.

Set a slave address at I2C0AR<SA> and set I2C0AR<ALS> to "0" to select the I2C bus mode. Finally, set SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0", SBI0CR2<PIN> to "1" and SBI0CR2<SWRST> to "00" for specifying the default setting to a slave receiver mode.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data cannot be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

15.5.2 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 tHIGH.

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.

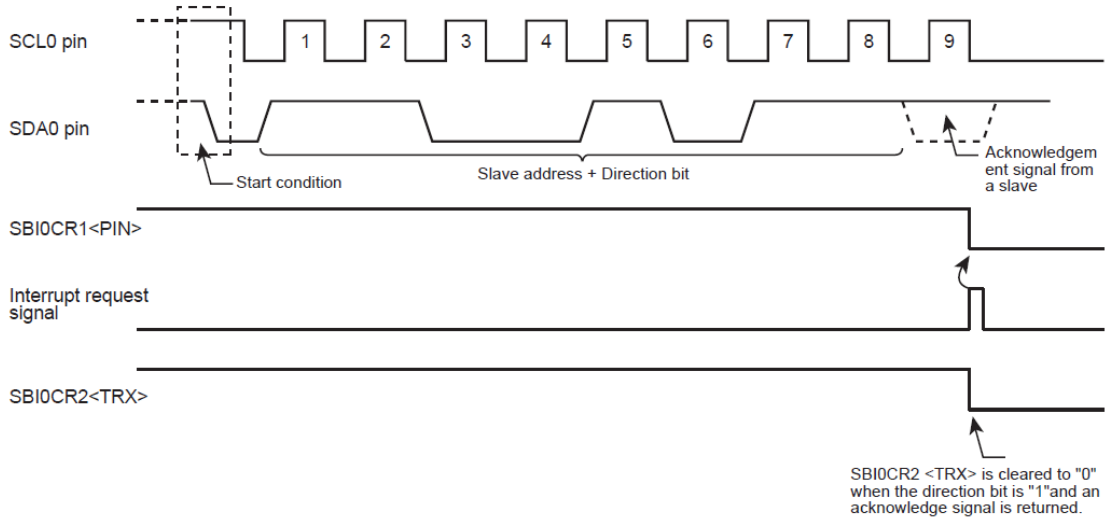


figure 15- 17 Generating the start condition and the slave address

15.5.3 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.

15.5.3.1 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.

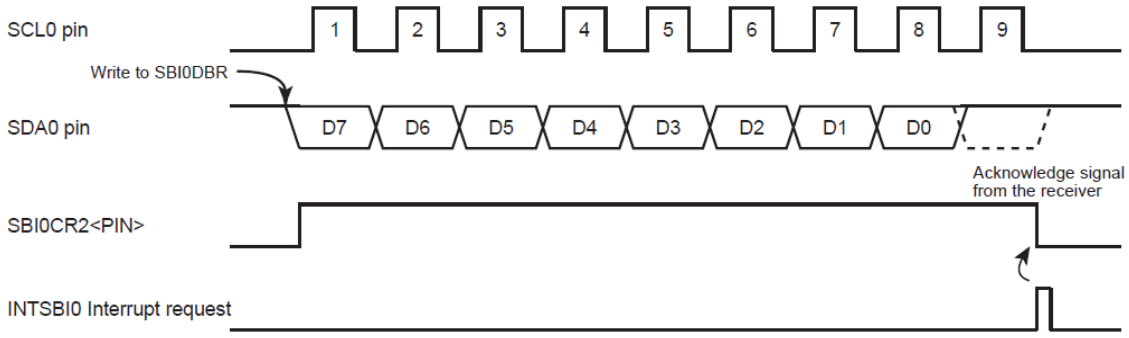


FIGURE 15- 18 EXAMPLE WHEN SBI0CR1<BC>="000" AND SBI0CR1<ACK>="1"

(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.

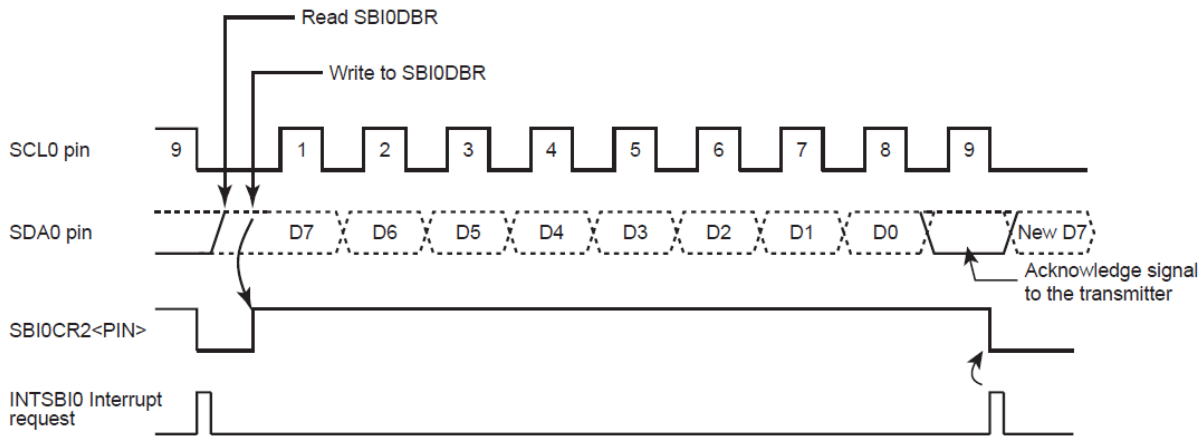


FIGURE 15- 19 EXAMPLE WHEN SBI0CR1<BC>="000" AND SBI0CR1<ACK>="1

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.

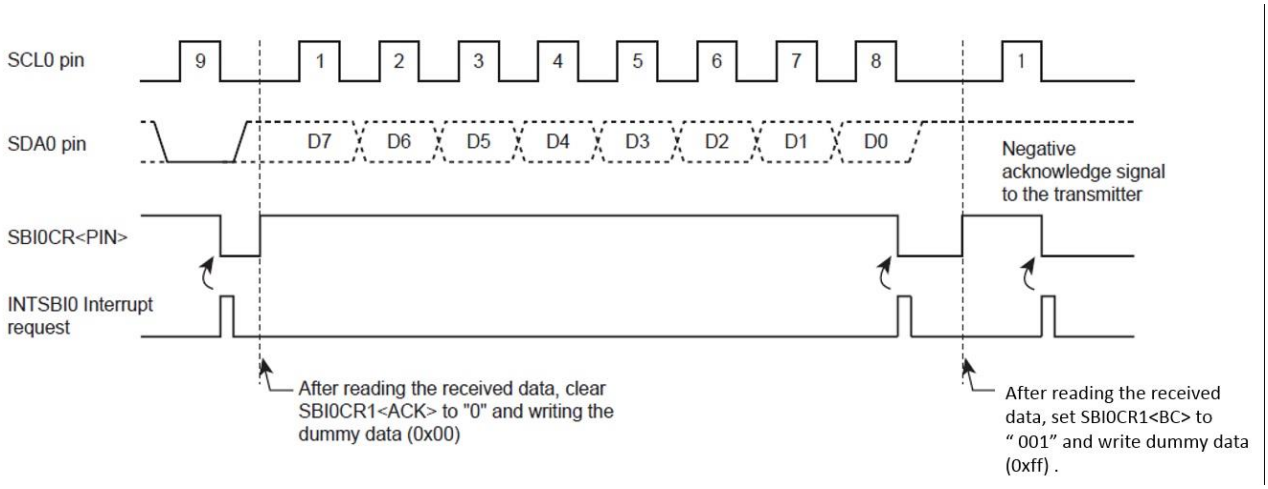


FIGURE 15- 20 TERMINATION OF DATA TRANSFER IN THE MASTER RECEIVER MODE

15.5.3.2 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 Table15.4.

	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".	

figure 15- 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 tLOW.

Check SBI0SR2<AL>, SBI0SR2<TRX>, SBI0SR2<AAS> and SBI0SR2<AD0> and implement processes according to conditions listed in table 15.5.

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 15- 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".

15.5.4 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}.

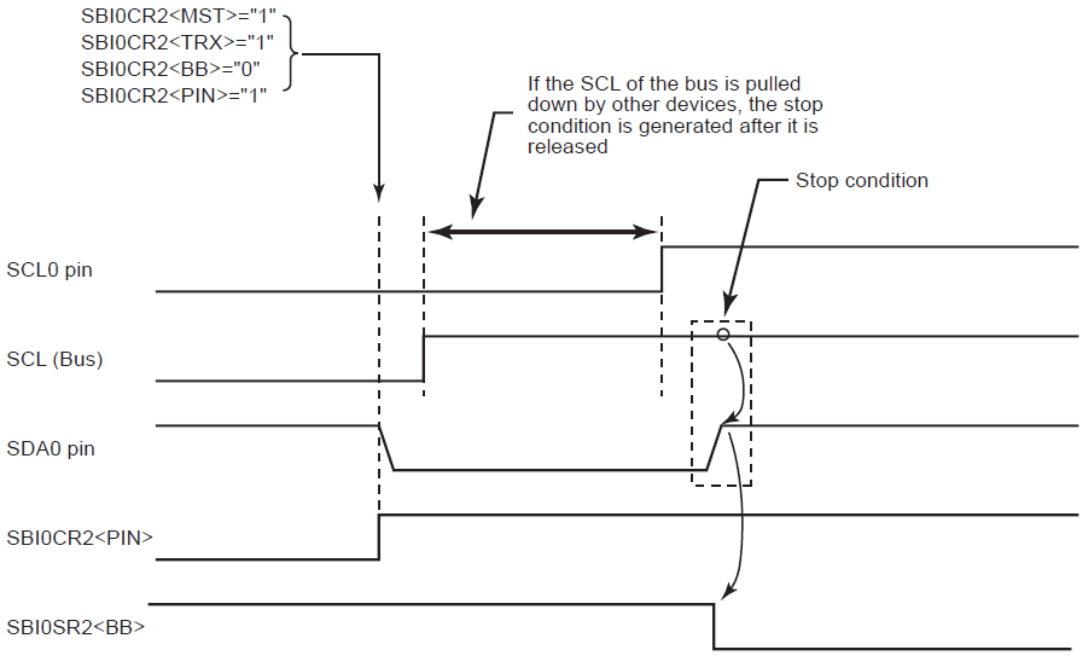


FIGURE15- 21 STOP CONDITION GENERATION

15.5.5 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.

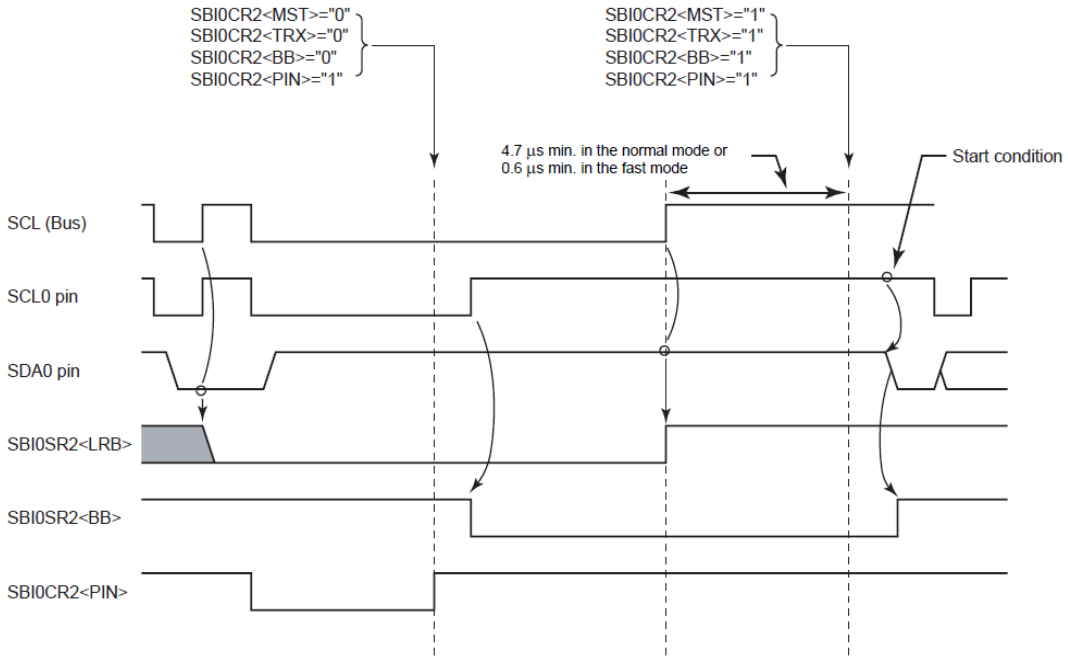


FIGURE 15- 22 TIMING DIAGRAM WHEN RESTARTING

15.6 AC Specifications

The operating mode (fast or standard) mode should be selected suitable for frequency of fsysclk. For these operating mode, refer to the following table.

Parameter	Symbol	Standard mode		Fast mode		Unit
		Min.	Max.	Min.	Max.	kHz
SCL clock frequency	f_{SCL}	0	fsysclk / (m+n)	0	fsysclk / (m+n)	us
Hold time (re)start condition.This period is followed by generation of the first clock pulse.	$t_{HD;STA}$	m / fsysclk	-	m / fsysclk	-	us
Low-level period of SCL clock(output)	t_{LOW}	n / fsysclk	-	n / fsysclk	-	us
High-level period of SCL clock(output)	t_{HIGH}	m / fsysclk	-	m / fsysclk	-	us
Low-level period of SCL clock(input)	t_{LOW}	5 / fsysclk	-	5 / fsysclk	-	us
High-level period of SCL clock(input)	t_{HIGH}	3 / fsysclk	-	3 / fsysclk	-	us
Restart condition setup time	$t_{SU;STA}$	Depends on the software	-	Depends on the software	-	us
Data hold time	$t_{HD;DAT}$	0	5 / fsysclk	0	5 / fsysclk	us
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 / fsysclk	-	m / fsysclk	-	us
Bus free time between the stop condition and start condition	t_{BUF}	Depends on the software	-	Depends on the software	-	us
Time before rising of SCL after SBICR2<PIN> is changed from "0" to "1"	$t_{SU;SCL}$	n / fsysclk	-	n / fsysclk	-	us

TABLE 15- 6 AC SPECIFICATIONS (CIRCUIT OUTPUT TIMING)

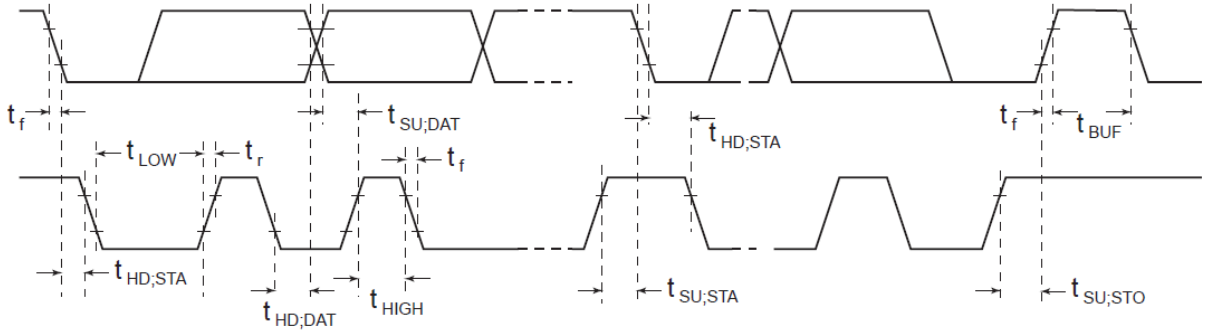


FIGURE 15- 23 DEFINITION OF TIMING (NO.1)

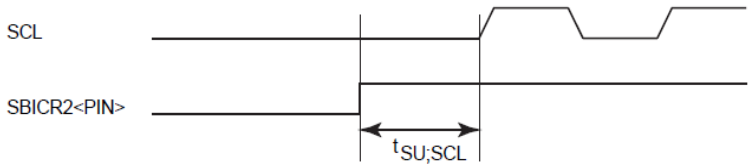


FIGURE 15- 24 DEFINITION OF TIMING (NO.2)

16 Synchronous Serial Interface (SIO)

SQ7613 contains 2 channel of 8-bit serial interface of the clock synchronization type.

	SIOxCR1 (Address)	SIOxCR2 (Address)	SIOxSR (Address)	SIOxBUF (Address)
SIO0	SIO0CR1 (0x00D0)	SIO0CR2 (0x00D1)	SIO0SR (0x00D2)	SIO0BUF (0x00D3)
SIO1	SIO1CR1 (0x00D4)	SIO1CR2 (0x00D5)	SIO1SR (0x00D6)	SIO1BUF (0x00D7)

table16- 1 SIO Address Assignment

16.1 Configuration

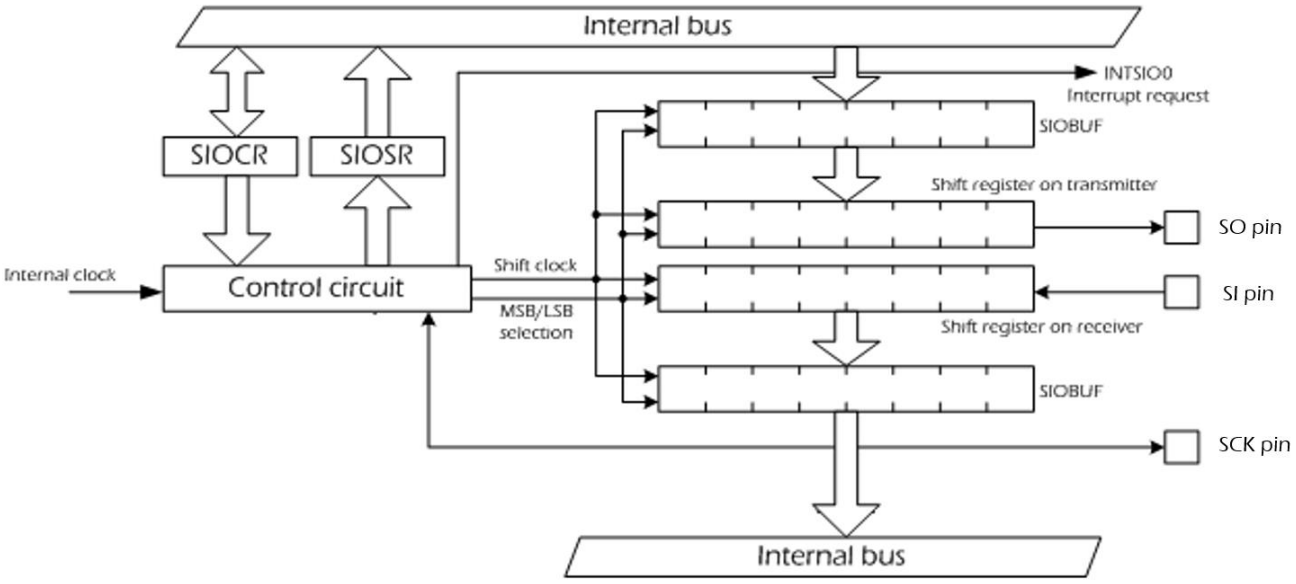


FIGURE 16- 1 SERIAL INTERFACE

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.For details,refter to 10. I/O ports.

16.2 Control

The synchronous serial interface SIO is controlled by peripheral circuit clock enable register 2 (PCKEN2), serial interface control register (SIOxCR), serial interface status register (SIOxSR) and serial interface buffer register (SIOxBUF).

ADDRESS	REGISTER	DESCRIPTION
0x017A	PCKEN2	Peripheral circuit clock enable register 2
0x00D0	SIO0CR1	Serial interface 0 control register 1
0x00D1	SIO0CR2	Serial interface 0 control register 2
0x00D2	SIO0SR	Serial interface 0 status register
0x00D3	SIO0BUF	Serial interface 0 buffer register
0x00D4	SIO1CR1	Serial interface 1 control register 1
0x00D5	SIO1CR2	Serial interface 1 control register 2
0x00D6	SIO1SR	Serial interface 1 status register
0x00D7	SIO1BUF	Serial interface 1 buffer register

Since the settings of the two sets of channels of the serial interface SIO are the same, the lower registers are described by the general symbol x (x = 0, 1), and the register positions can be searched by the table 15.1.

Peripheral circuit clock enable register 2(PCKEN2)

PCKEN2	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	SIO1	SIO0	reserved	reserved	I2C1	I2C0
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

SIO1	SIO1 enable control	0: Disable 1: Enable
SIO0	SIO0 enable control	0: Disable 1: Enable
I2C1	I2C1 enable control	0: Disable 1: Enable
I2C0	I2C0 enable control	0: Disable 1: Enable

Serial interface x control register 1(SIOxCR1), x=0,1

SIOxCR1	7	6	5	4	3	2。	1	0
Bit Symbol	SIOEDG	SIOCKS[2:0]			SIODIR	SIOS	SIOM[1:0]	
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 1: Transmit data at a rising edge and receive data at a falling edge.	
SIOCKS[2:0]	Serial clock selection [Hz]	Normal/Sleep mode fsysclk=HIRC/PLL/HXTAL	Normal/Sleep mode fsysclk =LIRC
		000：fsysclk/2^9 001：fsysclk /2^6 010：fsysclk /2^5 011：fsysclk /2^4 100：fsysclk /2^3 101：fsysclk /2^2 110：fsysclk /2	fsysclk/2
		111：Ext. clock input	
SIODIR	Transfer format (MSB/LSB)	0: LSB first(transfer from 0 bit) 1: MSB first(transfer from 7 bit)	
SIOS	Transfer operation start/stop instruction	0:Operation stop(reserved stop) 1:Operation start	
SIOM[1:0]	Transfer mode selection and operation	00：Operation stop(forced stop) 01：8-bit transmit mode 10：8-bit receive mode 11：8-bit transmit and receive mode	

Note1: *fsysclk* is System 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 NORMAL mode (slow clock) or SLEEP mode(slow clock), be sure to set SIOCKS to "110". If SIOCKs is set to any other value SIO will not operate. When SIO is used in NORMAL mode (slow clock) or SLEEP mode(slow clock), execute communications with SIOCKS="110" in advance or change SIOCKs after SIO is stopped.

Note6: When STOP, SLEEP mode or SLEEP (slow clock) 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 control register 2(SIOxCR2), x=0,1

SIOxCR2	7	6	5	4	3	2	1	0
Bit Symbol	reserved						reserved	SIOEN
Read/Write	R						R/W	R/W
After reset	0						0	0

SIOEN	SIO enable	0：Disable 1：Enable

Serial interface status register (SIOxSR), x=0,1

SIOxSR	7	6	5	4	3	2	1	0
Bit Symbol	SIOF	SEF	OERR	RENDB	UERR	TBFL	reserved	reserved
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Note1: The OERR and UERR flags are cleared by reading SIO0SR.

Note2: The REND flag is cleared by reading SIO0BUF.

Note 3:Bit 1 to 0 of SIO0SR are read"0".

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

Serial interface buffer register (SIOxBUF), x=0,1

SIOxBUF	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF[7:0]							
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Serial interface buffer register (SIOxBUF), x=0,1

SIOxBUF	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF[7:0]							
Read/Write	W	W	W	W	W	W	W	W
After reset	1	1	1	1	1	1	1	1

Note: SIO0BUF is a data buffer for serial transmission/reception of data. The data read from SIOxBUF each time is the latest data received by the serial interface. If SIOxBUF does not receive any data, the value read from SIOxBUF will be 0. When writing data to SIOxBUF, the data written is the data to be sent by the serial interface.

16.3 Low power consumption function

The serial bus interface has a Peripheral Clock Enable Register2 (PCKEN2) that saves power when the serial bus interface is not being used.

Setting PCKEN2<SIOx> to "0" disables the basic clock supply to serial interface 0 to save power. Note that this renders the serial interface unusable. Setting PCKEN2<SIOx> to "1" enables the basic clock supply to serial interface 0 and allows the serial interface to operate.

After reset, PCKEN2<SIOx> are initialized to "0", and this renders the serial interface unusable. When using the serial interface for the first time, be sure to set PCKEN2<SIOx> 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 PCKEN2<SIOx> to "0". Otherwise serial interface 0 may operate unexpectedly.

16.4 Functions

16.4.1 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.

16.4.2 Serial clock

The serial clock can be selected by using SIOxCR1<SIOCK>. In master mode, the maximum frequency is fsysclk/2 ; in slave mode, the maximum frequency is 4MHz.

Setting SIOxCR1<SIOCK> to "000"to110" selects the internal clock as the serial clock. In this case, the serial clock is output from the SCLK pin. The serial data is transferred in synchronization with the edge of the SCLK pin output.

Setting SIOxCR1<SIOCK> to"111" selects an external clock as the serial clock. In this case, an external serial clock must be input to the SCLK 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 "16.4.3 Transfer edge selection".

SIOxCR <SIOCKS>	Serial clock [Hz]		fsysclk=8MHz		fsysclk=16MHz		flclk=32.768kHz	
	fsysclk: HIRC/PLL/HXTAL	fsysclk: LIRC	1-bit time (us)	Baud rate (bps)	1-bit time (us)	Baud rate (bps)	1-bit time (us)	Baud rate (bps)
000	fsysclk/2^9	-	64	15.625k	32	31.3k	-	-
001	fsysclk/2^6	-	8	125k	4	250k	-	-
010	fsysclk/2^5	-	4	250k	2	500k	-	-
011	fsysclk/2^4	-	2	500k	1	1M	-	-
100	fsysclk/2^3	-	1	1M	0.5	2M	-	-
101	fsysclk/2^2	-	0.5	2M	0.25	4M	-	-
110	fsysclk/2	fsysclk/2	0.25	4M	0.13	8M	61	16.4k

TABLE 16- 2 TRANSFER BAUD RATE

16.4.3 Transfer edge selection

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

SIOxCR1 <SIOEDG>	Data transmission	Data reception
0	Falling edge	Rising edge
1	Rising edge	Falling edge

TABLE 16- 3 TRANSFER EDGE SELECTION

When SIOxCR1<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 SIOxCR1<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.

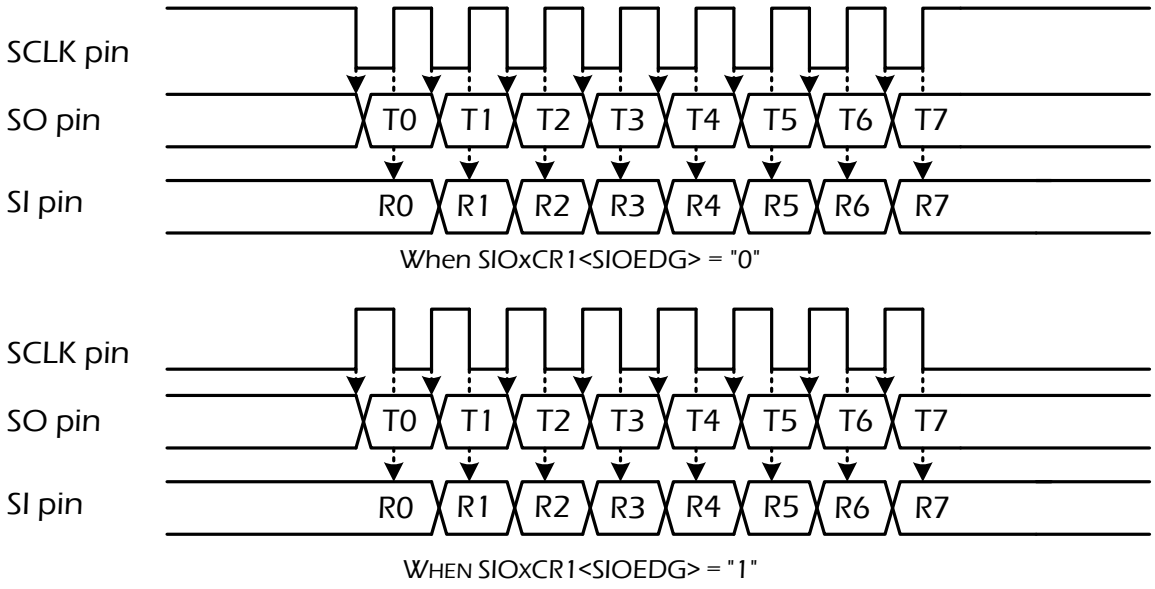


FIGURE 16- 2 TRANSFER EDGE

Note: When an external clock input is used, 4/fsysclk longer is needed between the receive edge at the 8th bit and the transfer edge at the first bit of the next transfer.

16.5 Transfer Modes

16.5.1 8-bit transmit mode

Setting SIO0CR<SIOM> to"01",to select 8-bit transmit mode.

16.5.1.1 Setting

Before starting the transmit operation, select the transfer edges at SIOxCR1<SIOEDG>, a transfer format at SIOxCR1<SIODIR>and a serial clock at SIOxCR1<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIOxCR1<SIOCKS>. To use an external clock as the serial clock, set SIOxCR1<SIOCKS> to"111".

Setting SIOxCR1<SIOM> to" 01"， to select the 8-bit transmit mode.

The transmit operation is started by writing the first byte of transmit data to SIOxBUF and then setting SIOxCR1<SIOS> to "1".

Writing data to SIOxCR1<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIOxSR <SIOF> is1. Make these setting while the serial communication is stopped.

While the serial communication is in progress (SIOxSR<SIOF>=" 1"), only writing "00" to SIOxCR1<SIOM> or writing "0" to SIOxCR1<SIOS> is valid.

16.5.1.2 Starting the transmit operation

The transmit operation is started by writing data to SIOxBUF and then setting SIOxCR1<SIOS> to "1". The transmit data is transferred from SIOxBUF to the shift register, and then transmitted as the serial data from the SO pin according to the settings of SIOxCR1<SIOEDG, SIOCKs an SIODIR>. The serial data becomes undefined if the transmit operation is started without writing any transmit data to SIOxBUF.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK pin. In the external clock operation, an external clock must be supplied to the SCLK pin.

By setting SIOxCR1<SIOS> to "1", SIOxSR<SIOF and SEF> are automatically set to "1" and an INTSIOx interrupt request is generated. SIOxSR <SEF> is cleared to "0" when the 8th bit of the serial data is output.

16.5.1.3 Transmit buffer and shift operation

If data is written to SIOxBUF 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, SIOxSR<TBFL> remains at "0".

If data is written to SIOxBUF when some data remains in the shift register, SIOxSR<TBFL> is set to "1". If new data is written to SIOxBUF in this state, the contents of SIOxBUF are overwritten by the new value. Make sure that SIOxSR<TBFL> is "0" before writing data to SIOxBUF.

16.5.1.4 Operation on completion of transmission

The operation on completion of the data transmission varies depending on the operating clock and the state of SIOxSR<TBFL>.

(a) When the internal clock is used and SIO0SR<TBFL> is "0"

When the data transmission is completed, the SCLK pin becomes the initial state and the SO pin becomes the "H" level. SIOxSR<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 SIOxBUF (automatic wait).

When the subsequent data is written into SIOxBUF, SIOxSR<SEF> is set to "1", the SCLK pin outputs the serial clock, and the transmit operation is restarted. An INTSIOx 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 SCLK pin after completion of the data transmission, an

undefined value is transmitted and the transmit underrun error flag SIOxSR<UERR> is set to "1".

If a transmit underrun error occurs, data must not be written to SIOxBUF during the transmission of an undefined value. It is recommended to finish the transmit operation by setting SIOxCR<SIOS> to "0" or force the transmit operation to stop by setting SIOxCR<SIOM> to "00".

The transmit underrun error flag SIOxSR<UERR> is cleared by reading SIOxSR.

(c) When an internal or external clock is used an SIO0SR<TBFL> is "1"

When the data transmission is completed, SIOxSR<TBFL> is cleared to "0". The data in SIOxBUF Is transferred to the shift register and the transmission of subsequent data is started. At this time, SIOxSR<SEF> is set to "1" and an INTSIOx interrupt request is generated.

16.5.1.5 Stopping the transmit operation

Set SIOxCR<SIOS> to "0" to stop the transmit operation. When SIOxSR<SEF> is "0", or when the shift operation is not in progress, the transmit operation is stopped immediately and an INTSIOx interrupt request is generated. When SIOxSR<SEF> is "1", the transmit operation is stopped after all the data in the shift register is transmitted (reserved stop). At this time, an INTSIOx interrupt request is generated again.

When the transmit operation is completed, SIOxSR<SIOF, SEF and TBFL> are cleared to "0". Other SIOxSR registers keep their values.

The transmit operation can be forced to stop by setting SIOxCR<SIOM> to "00" during the operation. By setting SIOxCR<SIOM> to "00", SIOxCR<SIOS> and SIOxSR are cleared to "0" and the SIO stops the operation, regardless of the SIOxSR <SEF> value. The SO0pin becomes the "H" level. If the internal clock is selected, the SCLK pin returns to the initial level.

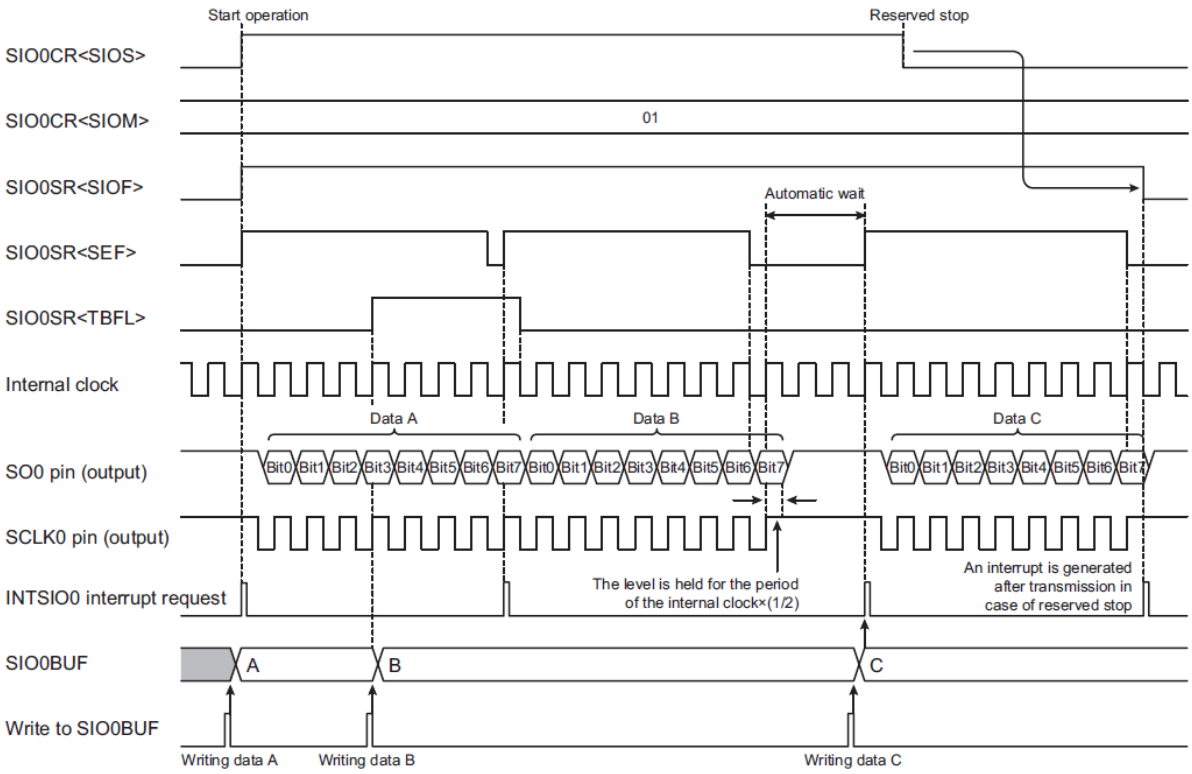


figure 16- 3 8-bit Transmit Mode (Internal Clock and Reserved Stop)

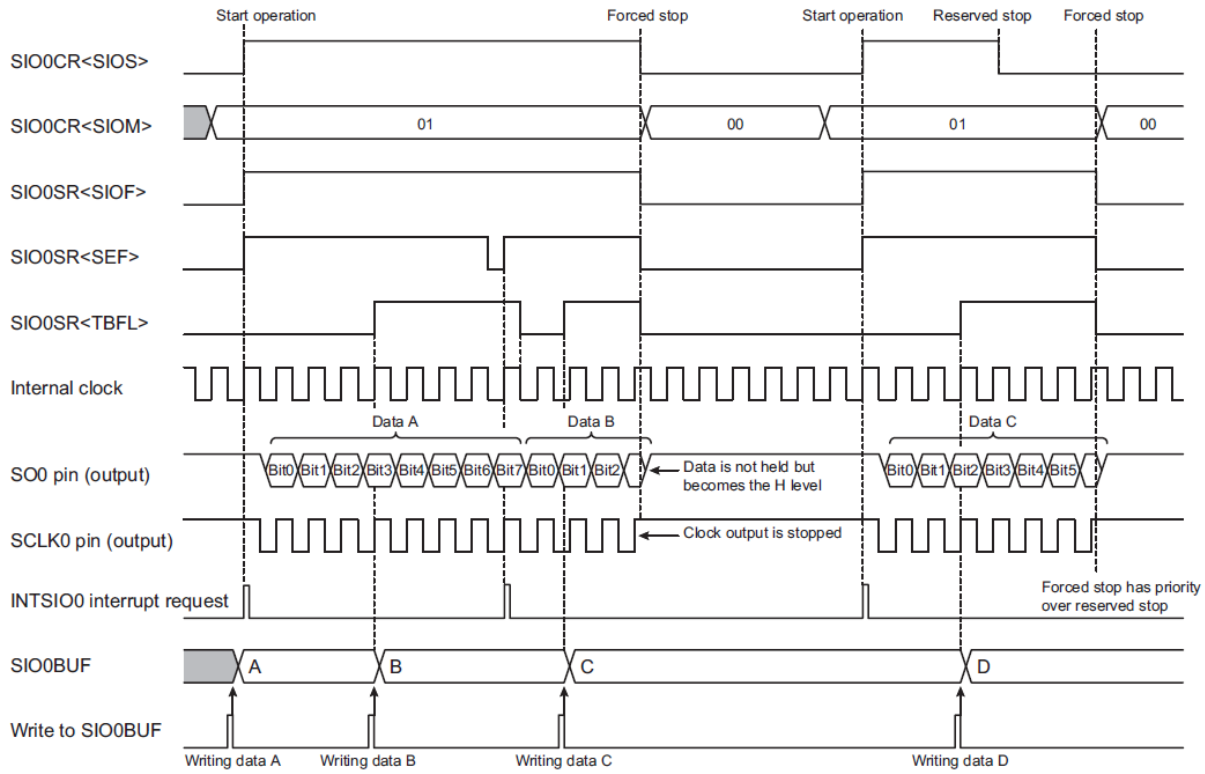


FIGURE 16- 4 8-BIT TRANSMIT MODE(INTERNAL CLOCK AND FORCED STOP)

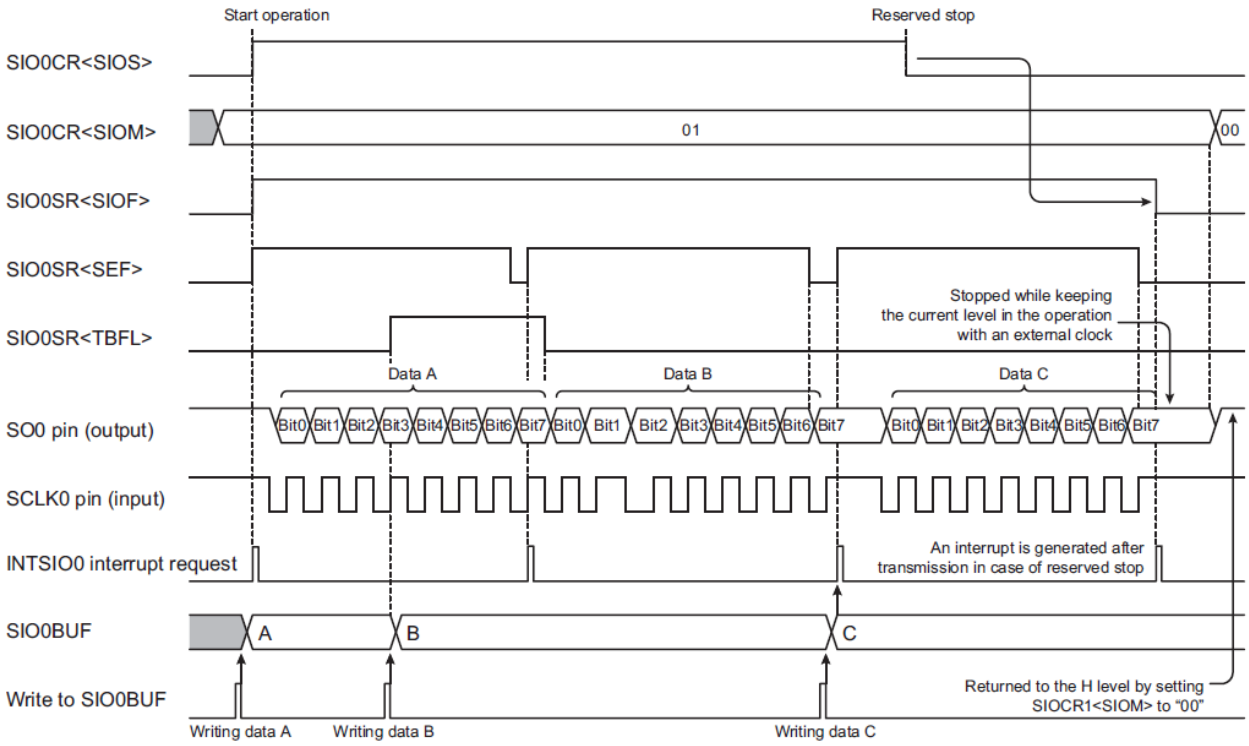


FIGURE 16- 5 8-BIT TRANSMIT MODE (EXTERNAL CLOCK AND RESERVED STOP)

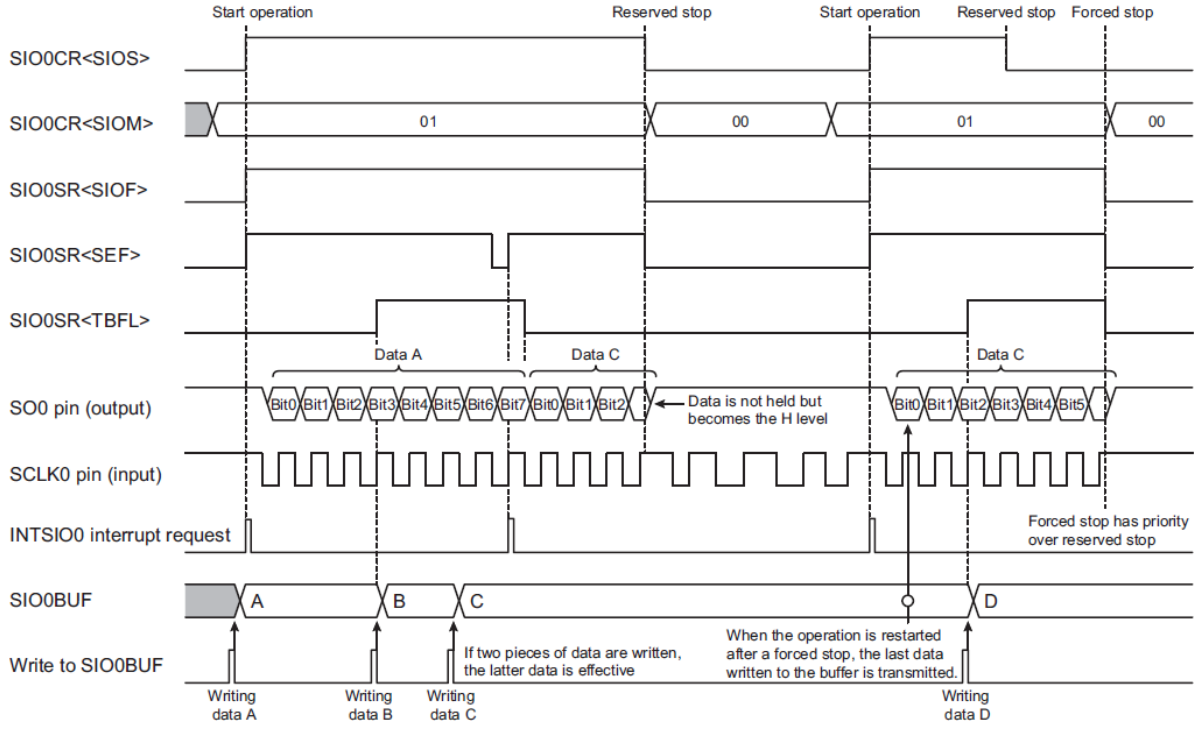


FIGURE 16- 6 8-BIT TRANSMIT MODE (EXTERNAL CLOCK AND FORCED STOP)

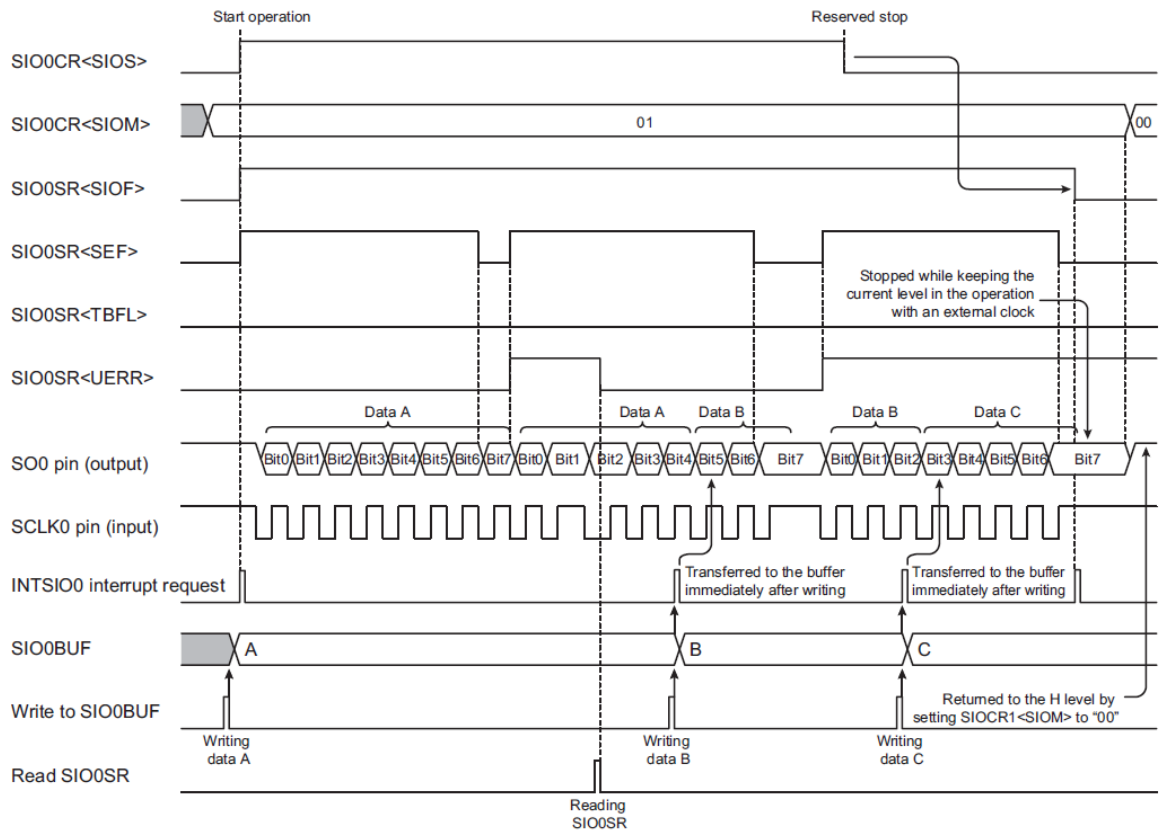


FIGURE 16- 7 8-BIT TRANSMIT MODE (EXTERNAL CLOCK AND OCCURRENCE OF TRANSMIT UNDERRUN ERROR)

16.5.2 8-bit Receive Mode

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to"10".

16.5.2.1 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.

16.5.2.2 Starting the receive operation

Reception is started by setting SIO0CR<SIOS> to "1". External serial data is taken into the shift register from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKs and SIODIR>.

Internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.
I
By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1".

16.5.2.3 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.

16.5.2.4 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 INTSIO0 interrupt request is generated in this state.

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.

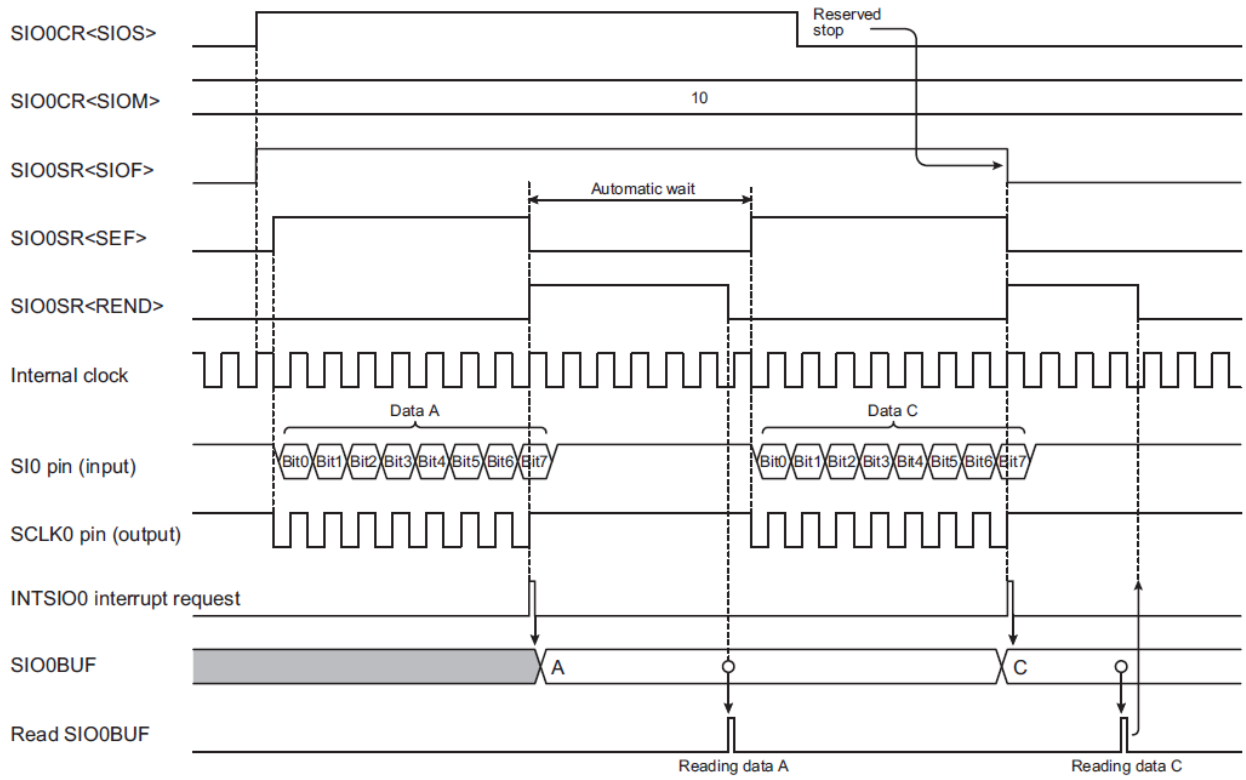


FIGURE 16- 8 8-BIT RECEIVE MODE (INTERNAL CLOCK AND RESERVED STOP)

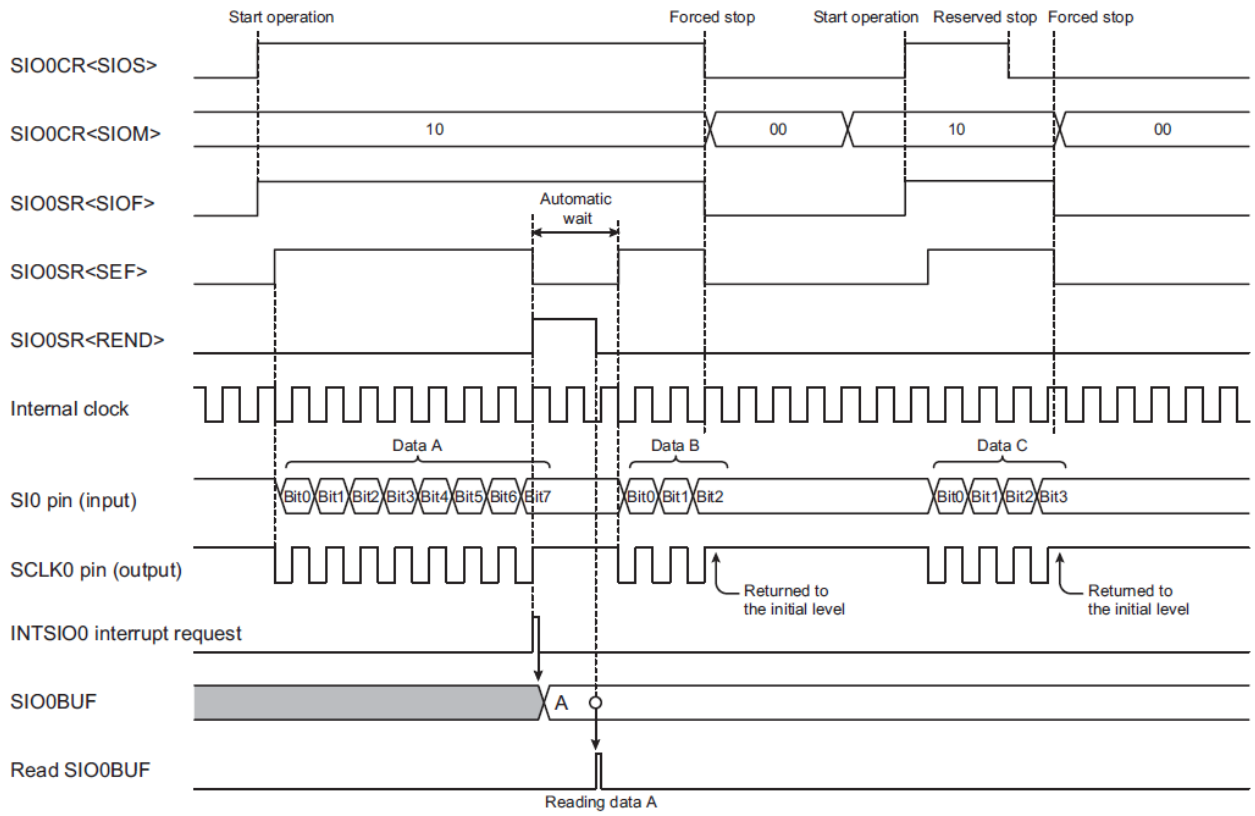


FIGURE 16- 9 8-BIT RECEIVE MODE (INTERNAL CLOCK AND FORCED STOP)

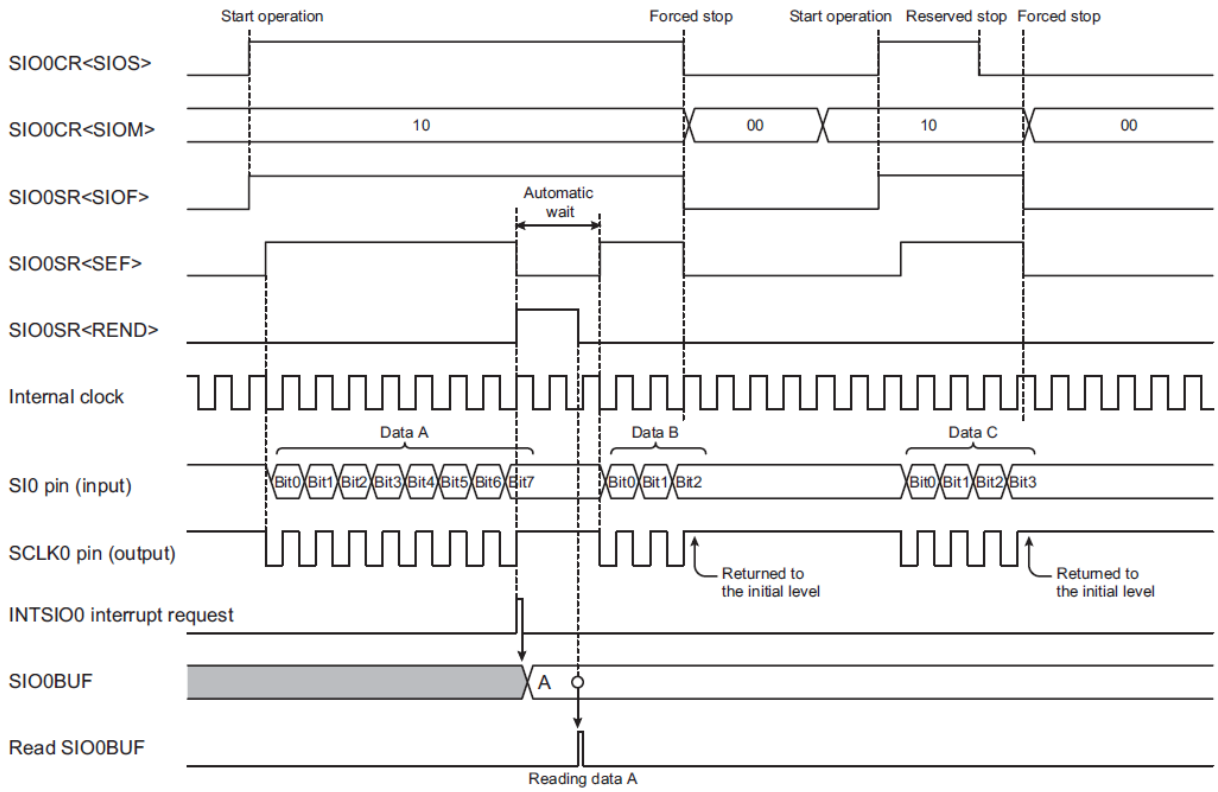


FIGURE 16- 10 8-BIT RECEIVE MODE (EXTERNAL CLOCK AND RESERVED STOP)

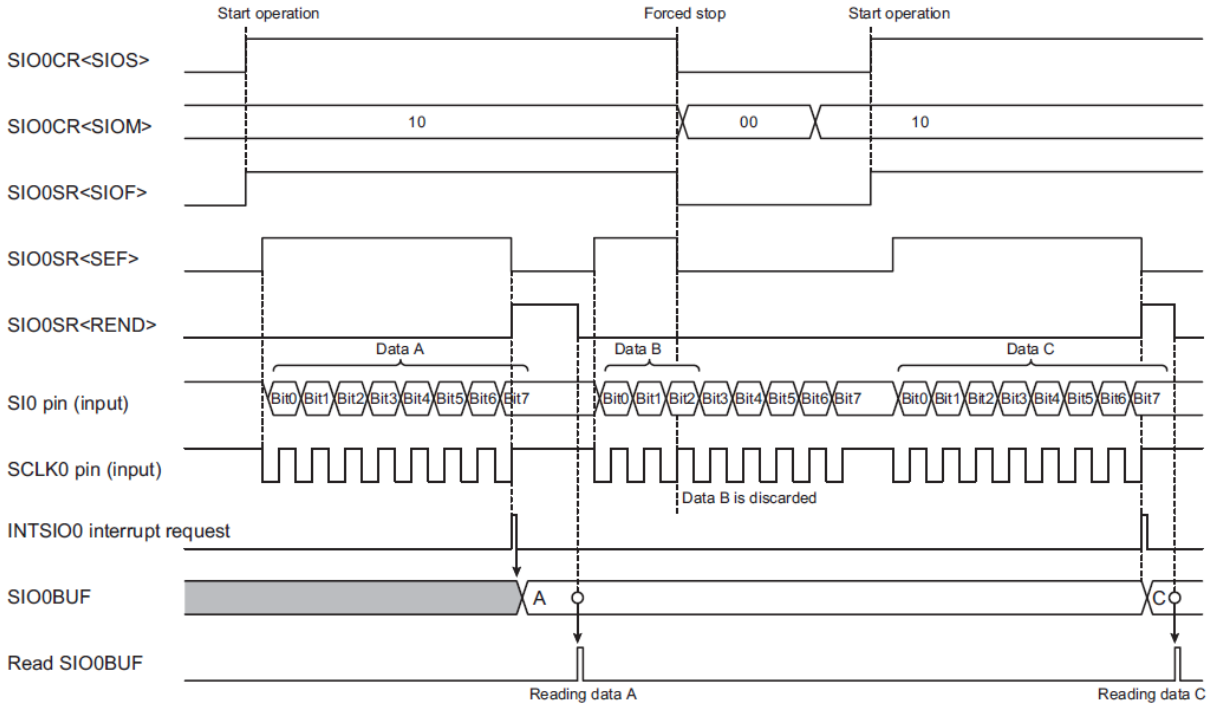


FIGURE 16- 11 8-BIT RECEIVE MODE(EXTERNAL CLOCK AND FORCED STOP)

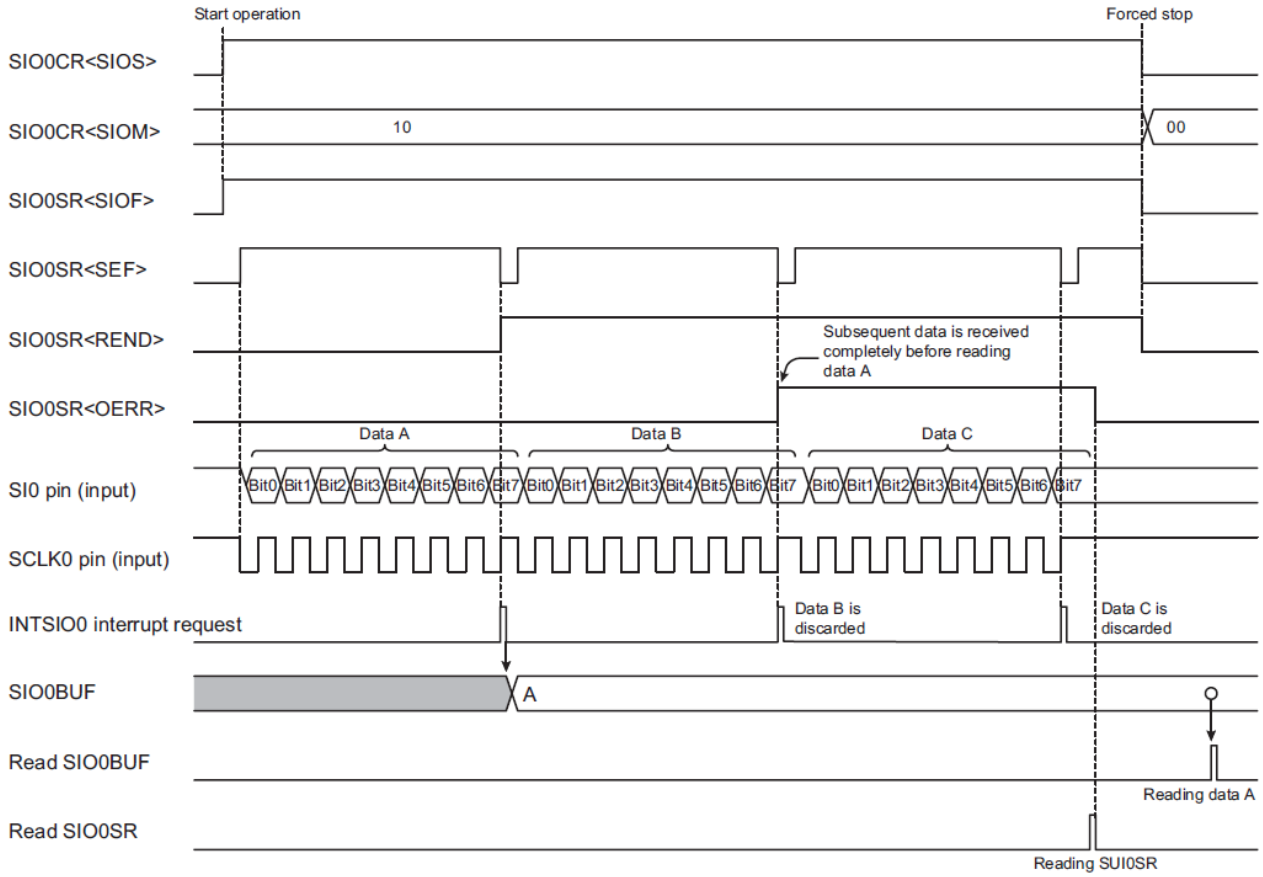


FIGURE 16- 12　8-BIT RECEIVE MODE(EXTERNAL CLOCK AND OCCURRENCE OF OVERRUN ERROR)

16.5.3 8-bit Transmit/receive mode

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

16.5.3.1 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.

16.5.3.2 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.

16.5.3.3 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

16.5.3.4 Operation on completion of transmission/ reception

When the data transmission/reception is completed, SIO0SR<REND> is set to "0" 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".

16.5.3.5 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.

After the operation has stopped completely, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

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.

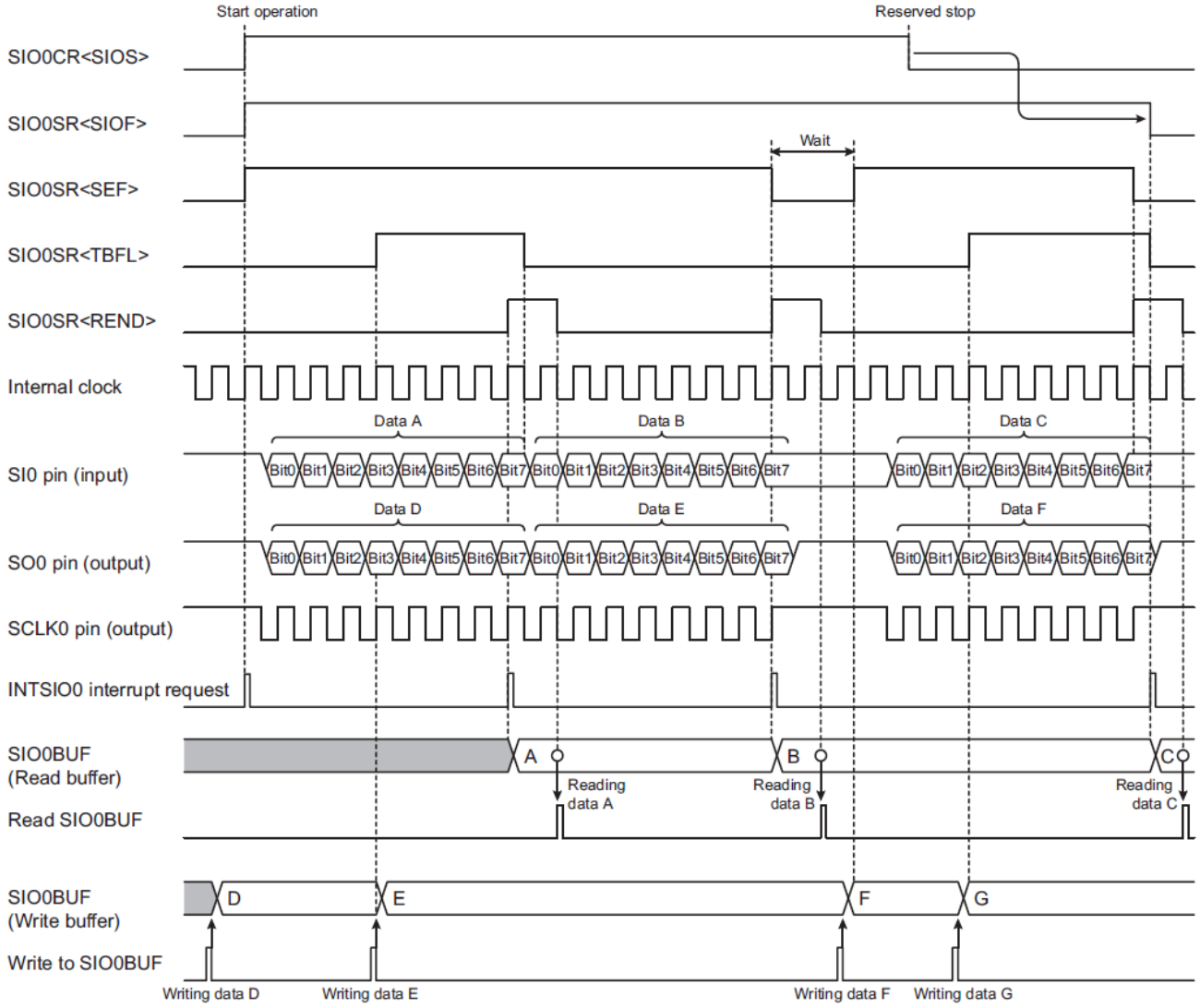


FIGURE 16- 13 8-BIT TRANSMIT RECEIVE MODE(INTERNAL CLOCK AND RESERVED STOP)

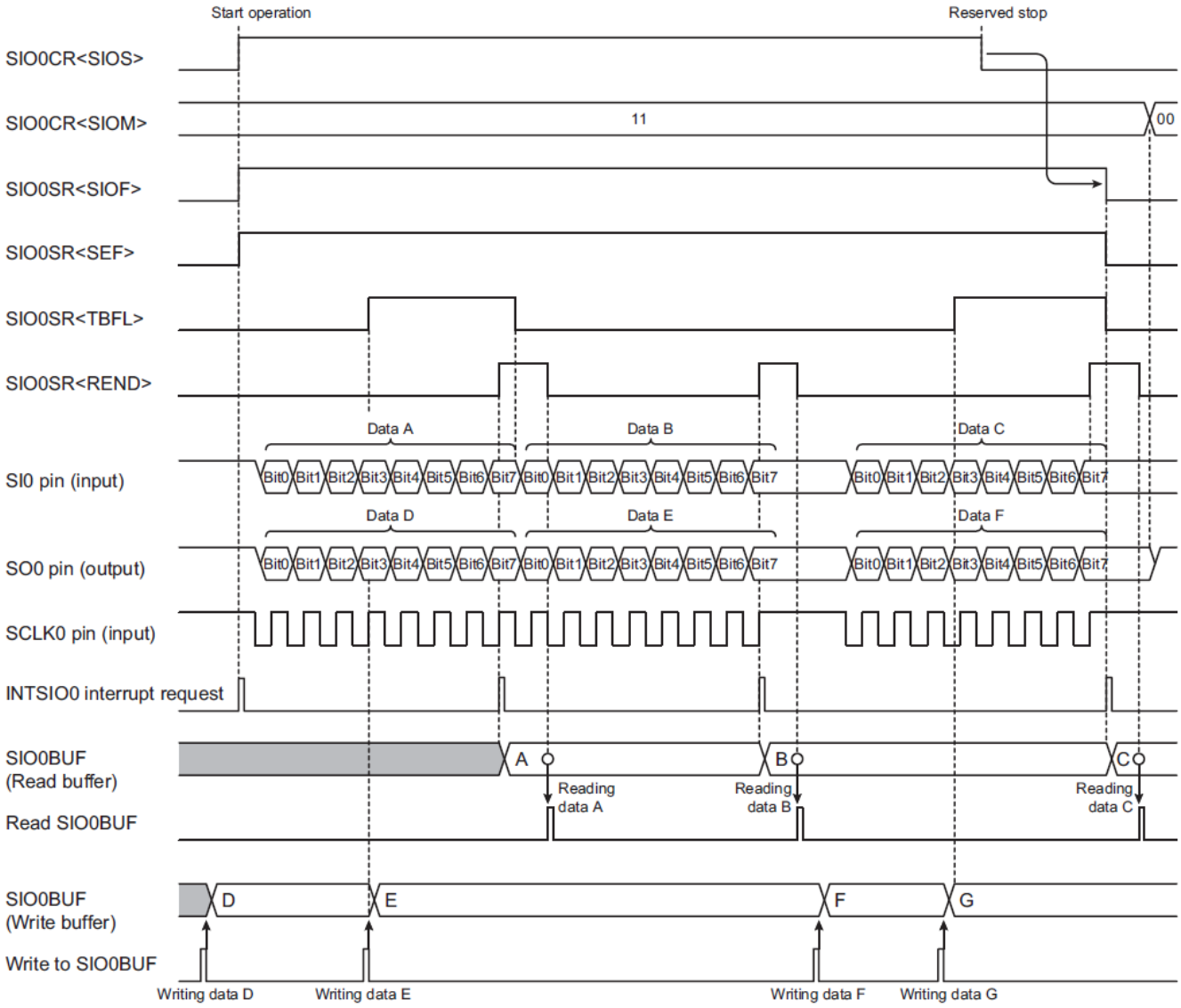


FIGURE 16- 14 8-BIT TRANSMIT/RECEIVE MODE(EXTERNAL CLOCK AND RESERVED STOP)

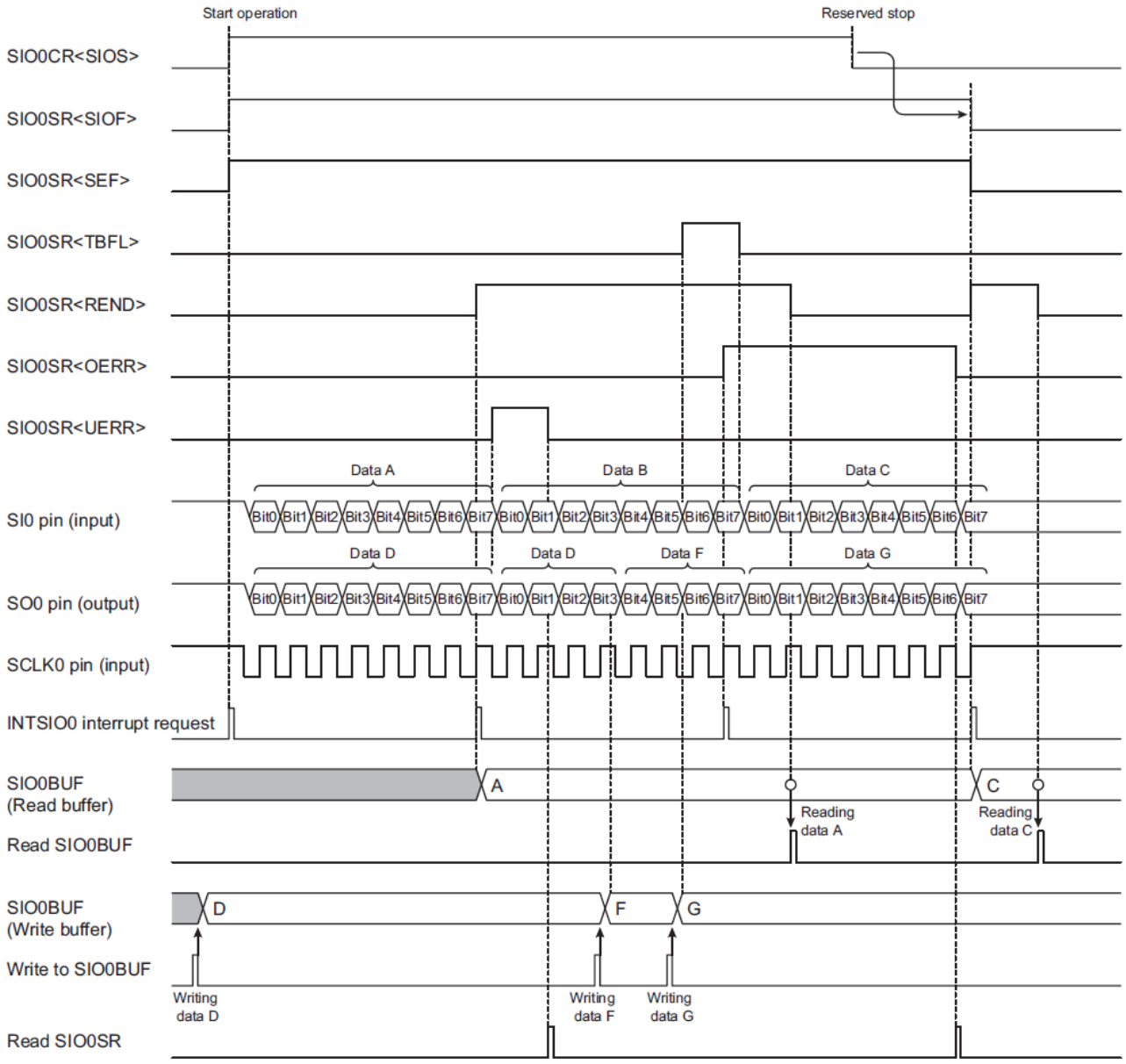


FIGURE 16- 15 8-BIT TRANSMIT/RECEIVE MODE(EXTERNAL CLOCK, OCCURRENCE OF TRANSMIT UNDERRUN ERROR AND OCCURRENCE OF OVERRUN ERROR)

16.6 AC Characteristic

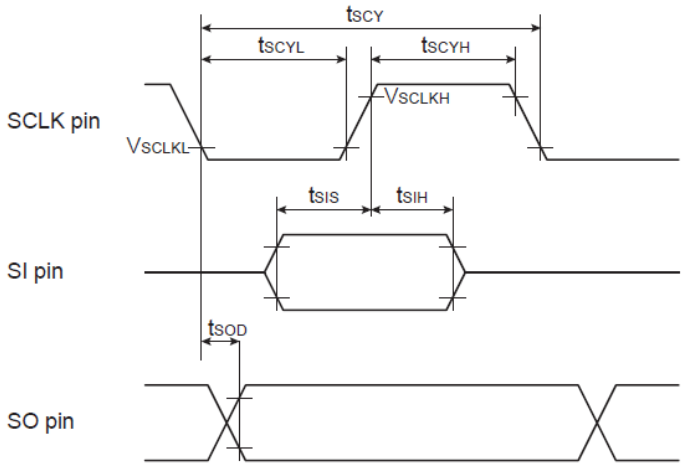


FIGURE 16- 16 AC CHARACTERISTICS

Vss=0, VDD=4.5V-5.5V, Topr=-40~85℃

Parameter	Symbol	Condition	Min	Typ	Max	Unit
SCLK cycle time	t_{SCY}	Internal clock operation SO pin and SCLK pin load capacity= 100pF	2 tsysclk	-		ns
SCLK"L" pulse width	t_{SCYL}		tsysclk -25	-		
SCLK"H" pulse width	t_{SCYH}		tsysclk-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-2}	External clock operation SO pin and SCLK pin load capacity= 100pF	2 tsysclk	-		
SCLK"L" pulse width	t_{SCYL-2}		tsysclk	-		
SCLK"H" pulse width	t_{SCYH-2}		tsysclk	-		
SI input setup time	t_{SIS-2}		50	-		
SI input hold time	t_{SIH-2}		50	-		
SO output delay time	t_{SOD-2}		0	-	60	
SCLK low-level input voltage	t_{SCLKL}		0	-	V_{DD} x 0.30	V
SCLK high-level input voltage	t_{SCLKH}		V_{DD} x 0.70	-	V_{DD}	

Note1: tsysclk=1/fsysclk.

Note2: In slave mode, minimum cycle time = 250ns (slave mode maximum frequency is 4MHz).

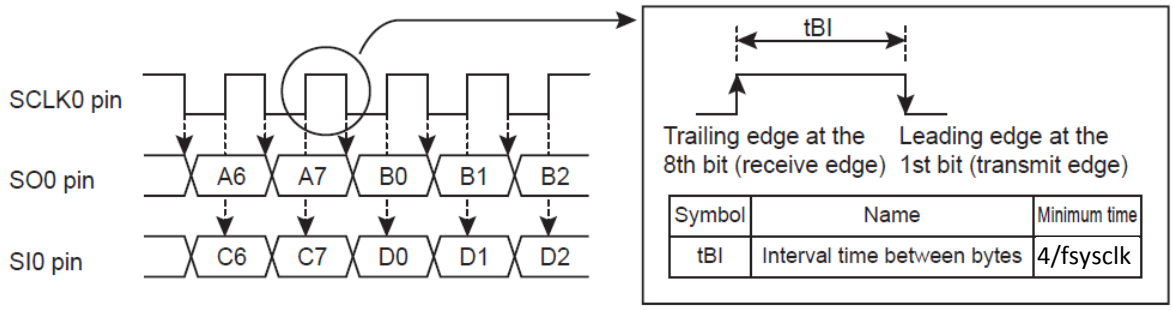


FIGURE 16- 17 INTERVAL TIME BETWEEN BYTES

17 Security

SQ7613 has Cyclic Redundancy Check (CRC) and Data Integrity Check (DIC) functions.

17. 1 Cyclic Redundancy Check (CRC)

17.1.1 Function

Cyclic Redundancy Check (CRC) correcting errors by adding a derivative bit of a block or a bit in a block symbol bit. Large blocks may compare CRCs probabilistically, therefore the CRC of each block is pre-computed and then compared. IF the CRC comparison result is different, the block is different, but the CRC comparison result is consistent, there is still a small chance that the block is inconsistent. The error probability can be reduced by increasing the CRC bit. When CRC is operating, CRCCR1 has to set to "0x02".

CRC generator polynomial used complies with "$X^{16}+X^{12}+X^5+1$" of CRC-16-CCITT.

17. 1.2 Control

Address	Register	Description
0x0850	CRCCR0	CRC Control Register 0
0x0851	CRCCR1	CRC Control Register 1
0x0858	CRCDI	CRC Data Input Register
0x085C	CRCDO0	CRC Data Output Register 0
0x085D	CRCDO1	CRC Data Output Register 1

CRC Control Register 0 (CRCCR0)

CRCCR0 (0x0850)	7	6	5	4	3	2	1	0
Bit Symbol	DATARDY	reserved						CRCEN
Read/Write	R	R/W						R/W
After reset	0	0	0	0		0	0	0

Note 1：This register is reset by all resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

Note 3：Bit[6:1] is reserved

DATARDY	CRC Data Ready	0：CRC is programming 1：CRC complete. When write into CRCDI, CRC enale, this bit will be auto-clear by hardware.
CRCEN	CRC Enable	0：CRC disable 1：CRC enable

CRC Control Register 1 (CRCCR1)

CRCCR1 (0x0851)	7	6	5	4	3	2	1	0
Bit Symbol	reserved	reserved	reserved	reserved	reserved	reserved	reserved	reserved
Read/Write	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	-	0	0	0	0	0	0	0

Note 1：This register is reset by all resets.

Note 2：CRCCR1 must be set by 0x02, when programming.

CRC Data Input Register (CRCDI)

CRCDI (0x0858)	7	6	5	4	3	2	1	0
Bit Symbol	CRCDI[7:0]							
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：This register is reset by all resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

CRCDI[7:0]	CRC data input

CRC Data output Register 0(CRCDO0)

CRCDO0 (0x085C)	7	6	5	4	3	2	1	0
Bit Symbol	CRCDO0[7:0]							
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：This register is reset by all resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

Note 3：Read CRCDO0 first, then read CRCDO1.

CRCD0[7:0]	CRC data output

CRC Data output Register 1(CRCDO1)

CRCDO1 (0x085D)	7	6	5	4	3	2	1	0
Bit Symbol	CRCDO1[7:0]							
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：This register is reset by all resets.

Note 2：Reserved bits must be written with zeros for future compatibility.

Note 3：Read CRCDO0 first, then read CRCDO1.

CRCDO1[7:0]	CRC data output

If the CRC value is "0x9015" , the crc_out would be {0x15,0x90}.

16-bit Integer

0x15	0x90
CRCDO0	CRCDO1

17. 2 Data Integrity Check (DIC)

17.2.1 DIC Function

Data Integrity Check (DIC) use th CRC block for caculations, using either Flash or SRAM. Because DIC use CRC to calculation, so enable CRC function before starting DIC operating.

DIC does not have a corresponding PCKEN, and writes the count value to the DIC count register (DICCNT). If DICCNT = 0, it is 65536 bytes; if DICCNT = 1, it is 1 bytes. The start address is then written to the DIC Address Register (DICADR). To ensure DIC operate normally, the DIC can only be performed in one memory space (Flash or SRAM) at a time, and cannot be performed in both Flash and SRAM. The user needs to ensure that both the start bit and the counted position are within the specified memory space.

To start DIC, set DICEN "1". When DIC is performed, the DIC completion bit (DONE) is automatically cleared to 0. When DIC is completed, the DIC start bit will be cleared to 0 (DICEN = 0) and the DIC completion bit (DONE) will be set to 1. You can poll the value of DONE to determine if the DIC operation is complete. If the DIC and the program start from the same position, the CPU will stall until the DIC operation is completed.

Brief sample code of DIC as below :

```
PCKEN7_CRC = 1;       // PCKEN7 enable CRC
CRCCR0 = 0x00;        // Initilized CRC
CRCCR1 = 0x02;        // Initilized CRC
DICCR = 0x00;         // Initilized DIC
CRCCR0 = 0x01;        // Enable CRC
DICCNT0 = 0x00;       // DIC length set to "0x1000"
DICCNT1 = 0x10;       // DIC length set to 0x1000

DICADR0 = 0x00;       // Start FLASH Memory Map : 0x00400000
DICADR1 = 0x00;       // Start SRAM Memory Map : 0x00800000
DICADR2 = 0x40;
DICADR3 = 0x00;

DICCR = 0x01;         // DIC enable

while(DICCR_DONE == 0) // Wait for DIC done
{
      __ASM("NOP");
}

dic_data[0] = CRCDO0;  // CRC output
dic_data[1] = CRCDO1;  // CRC output
```

17.2.2 DIC Control

Address	Register	Description
0x0860	DICCR	DIC Control Register 0
0x0862	DICCNT0	DIC Count Register 0
0x0863	DICCNT1	DIC Count Register 1
0x0864	DICADR0	DIC Address Register 0
0x0865	DICADR1	DIC Address Register 1
0x0866	DICADR2	DIC Address Register 2
0x0867	DICADR3	DIC Address Register 3

DIC Control Register 0(DICCR)

DICCR (0x0860)	7	6	5	4	3	2	1	0
Bit Symbol	DONE	reserved	reserved	DICIE	reserved	reserved		DICEN
Read/Write	R	-	-	R/W	-	R/W		R/W
After reset	1	-	-	0	-	1		0

Note 1：This register is reset by all resets.

Note 2：When DIC is performing, DICCR- Bit [2:1] must set to "0"

DONE	DIC done flag	0: DIC is programming 1:DICcomplete / Idle When DICEN set to "1", DIC start automatically, this bit is cleared by auto.
DICIE	DIC Interrupt Enable	0: DIC interrupt Disable 1: DIC interrupt Enable
DICEN	DIC Enable	0:DIC Disable 1:DIC Enable When DIC complete, this bit would be cleared automatically.

DIC Count Register 0 (DICCNT0)

DICCNT0 (0x0862)	7	6	5	4	3	2	1	0
Bit Symbol	DICCNT0							
Read/Write	R/W							
After reset	0							

Note 1：This register is reset by all resets.

DICCNT	DIC data length count

DIC Count Register 1(DICCNT1)

DICCNT1 (0x0863)	7	6	5	4	3	2	1	0
Bit Symbol	DICCNT1							
Read/Write	R/W							
After reset	0							

Note 1：This register is reset by all resets.

DICCNT1	DIC data length count

DIC Address Register 0(DICADR0)

DICADR0 (0x0864)	7	6	5	4	3	2	1	0
Bit Symbol	DICADR0[7:0]							
Read/Write	R/W							
After reset	0							

Note 1：This register is reset by all resets.

DICADR1[7:0]	DIC 32-bit address register, include data zone starting address.

DIC Address Register 1 (DICADR1)

DICADR1 (0x0865)	7	6	5	4	3	2	1	0
Bit Symbol	DICADR1[7:0]							
Read/Write	R/W							
After reset	0							

Note 1：This register is reset by all resets.

DICADR1[7:0]	DIC 32-bit address register, include data zone starting address.

DIC Address Register 2(DICADR2)

DICADR2 (0x0866)	7	6	5	4	3	2	1	0
Bit Symbol	DICADR2[7:0]							
Read/Write	R/W							
After reset	0							

Note 1：This register is reset by all resets.

DICADR2[7:0]	DIC 32-bit address register, include data zone starting address.

DIC Address Register 3(DICADR3)

DICADR3 (0x0867)	7	6	5	4	3	2	1	0
Bit Symbol	DICADR3[7:0]							
Read/Write	R/W							
After reset	0							

Note 1：This register is reset by all resets.

DICADR3[7:0]	DIC 32-bit address register, include data zone starting address.

Appendix A. On-chip Debug

SQ7613 has an in-system programming (ISP) function. Using a combination of this function and iMQ on-chip debug emulator MQ-Link, the user is able to perform software debugging in the on-board environment. This emulator can be operated from a debugger installed on a PC so that the emulation and debugging functions of an application program can be used to modify a program or for other purposes.

This chapter describes the control pins needed to use the ISP function and how a target system is connected.

Control Pins

The pins used for the on-chip debug function are shown in Table A-1.

On-chip debug data				
Pin Name (during ISP function)	**Input/Output**	**Corresponding pin of MQ-LINK**	**Function**	**Pin Name** (in MCU mode)
DBG	I/O	OCDIO	Communication control pin	P3.4/ KWI 14/EINT4
RESET	Input	RESET	Reset control pin	RESET
VDD	Power Supply	VCC	5.0V (2.0V to 5.5V)	
VSS	GND	GND	0V	

TABLE A- 1 PINS USED FOR ISP FUNCTION

FIGURE A- 1 MQ-LINK TOP VIEW

How to Connect MQ-Link Debugger to a Target System

To use the ISP function, the specific pins on a target system must be connected to the MQ-Link debugging system. MQ-Link can be connected to a target system via an interface control cable. iMQ provides a connector for this interface control cable as an accessory tool. Mounting this connector on a target system will make it easier to use the ISP function.

The connection between the MQ-Link and a target system is shown in Figure A-2.

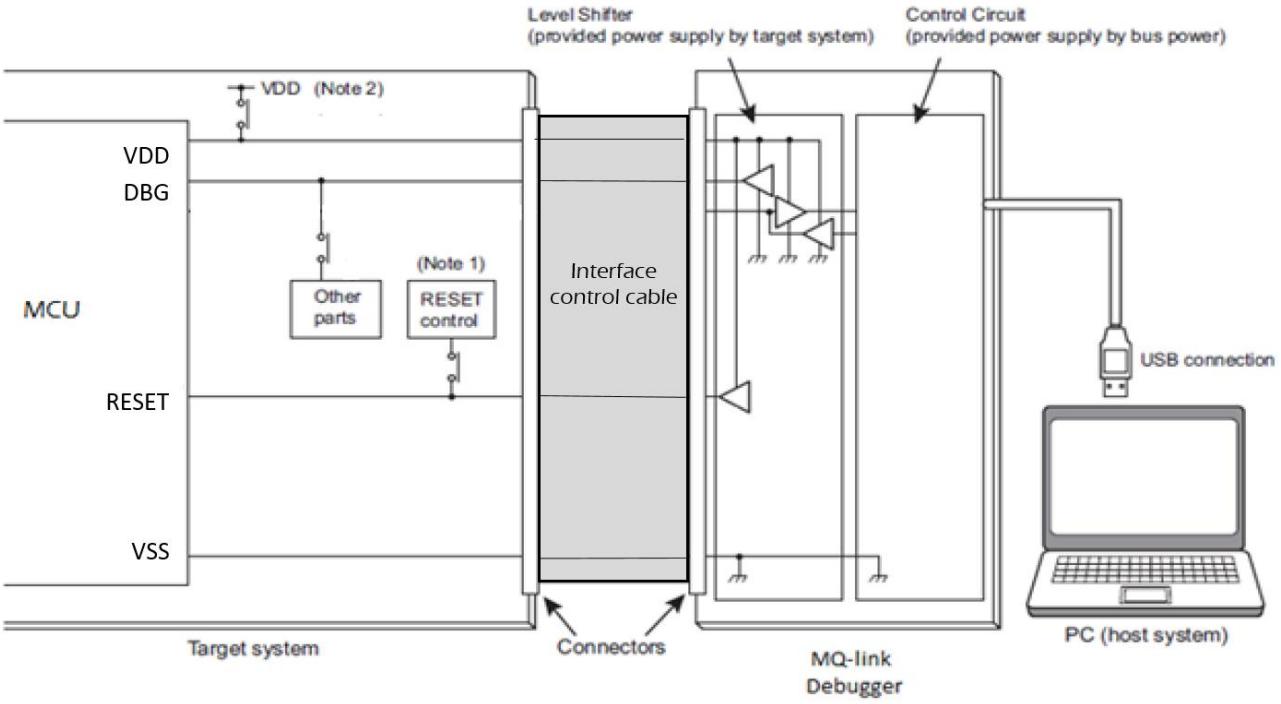


FIGURE A- 2 HOW TO CONNECT MQ-LINK DEBUGGER TO A TARGET SYSTEM

Note 1: If the reset control circuit on an application board affects the control of the ISP function, it must be disconnected using a jumper, switch, etc.

Note 2: During the ISP function, the power supply of MCU on target system is provided by MQ-Link debugger directly. After finishing ISP function, MCU can use the original power supply on target system.

Note 3: For details of MQ-Link, please refer to "iMQ i87-IDE User Manual".

Appendix B. Product Number Information

Example： SQ 76 13 LQ 032 S E T R

iMQ SQ product

Product Series

Sub Series

Pakage Type

Code	Pakagr Type	Code	Pakagr 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	Pakagr Type	Code	Pakagr Type
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

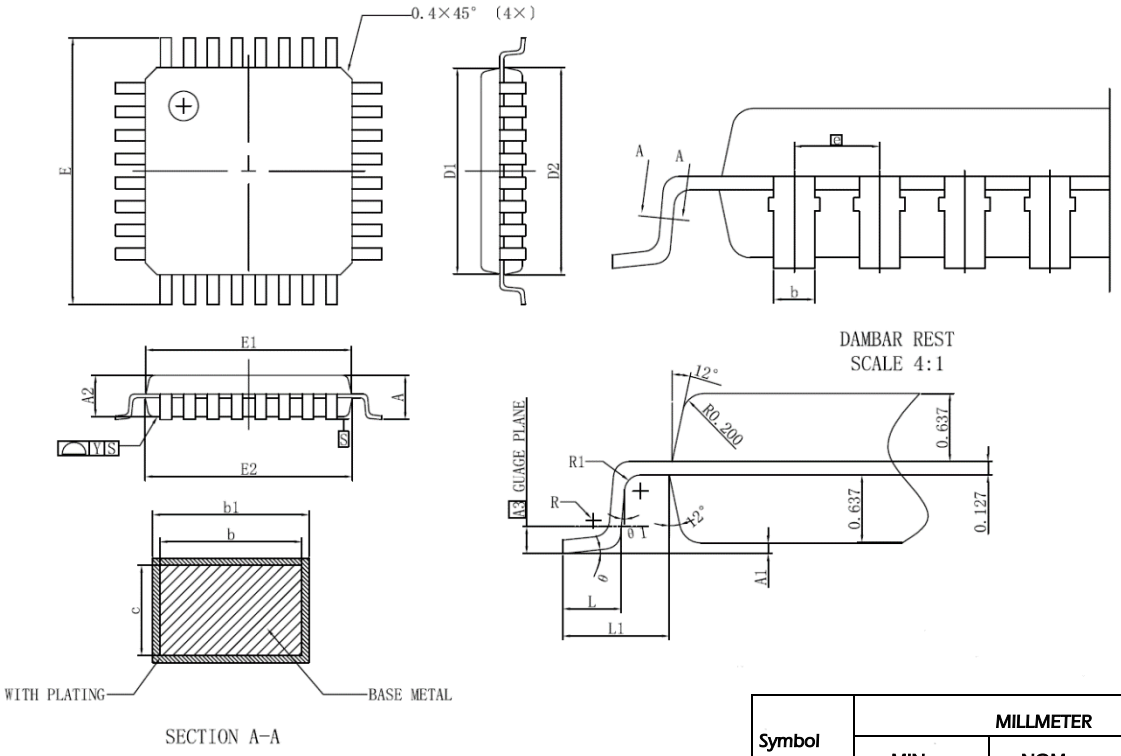
Code	Program Flash/ Data Flash/ RAM Size	Code	Program Flash/ Data Flash/ RAM Size
A	128 Bytes	K	24K Bytes
B	256 Bytes	M	32K Bytes
E	512 Bytes	N	40K Bytes
J	1K Bytes	P	48K Bytes
L	2K Bytes	S	64K Bytes
T	4K Bytes	U	96K Bytes
G	8K Bytes	W	128K Bytes
C	12K Bytes	V	无
H	16K Bytes		

Operating Temp

Code	Operating Temp.
R	-40~85°C

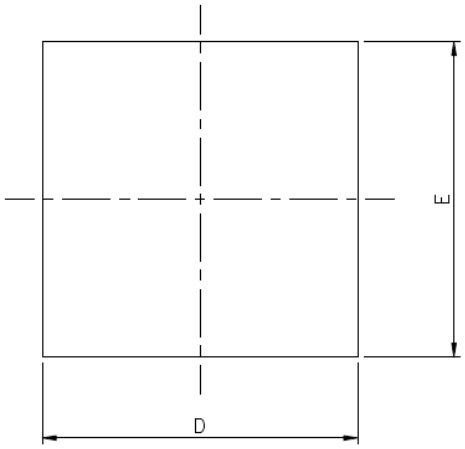
Appendic C. Package Dimensions

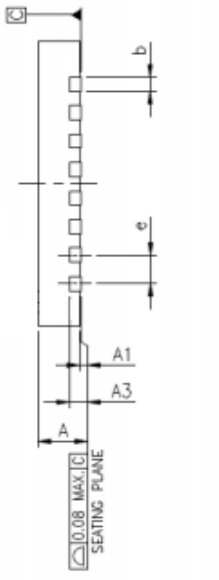
LQFP 32 7x7

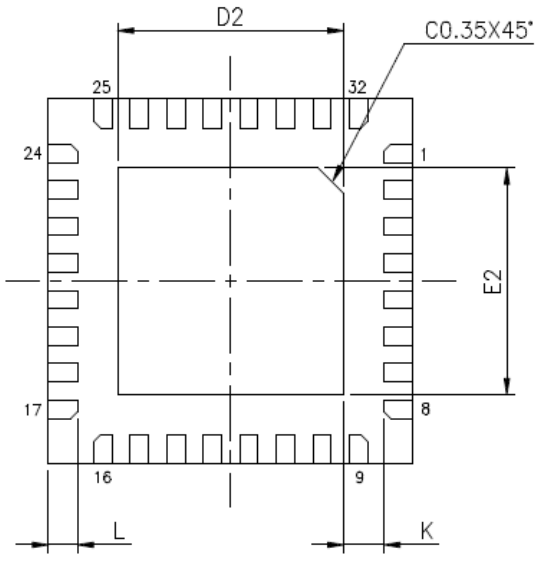


Symbol	MILLMETER		
	MIN	NOM	MIN
A	1.45	1.55	1.65
A1	0.01	–	0.21
A2	1.3	1.4	1.5
A3	–	0.254	
b	0.30	0.35	0.40
b1	0.31	0.37	0.43
c	–	0.127	–
D1	6.85	6.95	7.05
D2	6.9	7.00	7.10
E	8.8	9.00	9.20
E1	6.85	6.95	7.05
E2	6.9	7.00	7.10
e	–	0.8	–
L	0.43	–	0.71
L1	0.90	1.0	1.10
R	0.1	–	0.25
R1	0.1	–	–
θ	0	–	10°
θ1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

QFN 32 (5x5)







Symbol	MILLMETER		
	MIN	NOM	MIN
A	0.7	0.75	0.80
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.18	0.25	0.30
D	5.00 BSC		
E	5.00 BSC		
e	0.50 BSC		
L	0.35	0.40	0.45
K	0.20	–	–
D2	3.40	3.50	3.60
E2	3.40	3.50	3.60

Appendix D. Application Notice

(A) Emulation related

1. Do not support emulation at low frequency. When clock source is LIRC, it can not support emulation. Suggest to emulation under other clock source.
2. Under emulation, and CPU operate as single stepping. When CPU interrupt , TCAx (x=0~7) will not interrupt and continue operating.
3. Under emulation, ADC will continue operating when CPU interrupt.

(B) Clock source related

1. High frequenct crystal oscillator has to wait 2,5ms (16NHz, 25℃) to stable status.

(C) Operating mode related

1. If use KWI, and LVD to exit deep sleep mode, before entering deep sleep mode, set CLKCR1<HIRCEN>=1.

(D) I/O and power related

1. Do not input signal or power to I/Os when IC is power-off. To avoid causing current injection and IC operate incorrectly.

(E) GPIO related

1. If you need to read the GPIO status immediately., when system enter interrupt . Add "NOP" to the program to avoid reading incorrectly. Sample code as below.

```
//*****************************************************************************************

// Name   : __interrupt IntEX0()
// Function: IntEX0 Interrupt subroutine
// Note:
//*****************************************************************************************
void __interrupt IntEX0(void){
__asm("NOP");                                          // Add "NOP" instruction after entering
                                                        external interrupt
}
//*****************************************************************************************
// Name   : __interrupt IntEX1()
//Function: IntEX1 Interrupt subroutine
// Note:
//*****************************************************************************************
void __interrupt IntEX1(void){
__asm("NOP");                                          // Add "NOP" instruction after entering
                                                       external interrupt
}
//*****************************************************************************************
// Name   : __interrupt IntEX2()
```

```
// Function: IntEX2 Interrupt subroutine
// Note:
//*************************************************************************************************
void __interrupt IntEX2(void){
__asm("NOP");                                   // Add "NOP" instruction after entering external interrupt

}
```

(F) DC character related

IDD reverence valus at different frequencies as follows:

Operating @ 5V,Ta=-25℃				
Parameter	**Symbol**	**Condition**	**Typ.**	**Unit**
Normal Mode (LIRCon · code executing from flash)	I_{DD_N0}	LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=24MHz (PLL)	5.5	mA
		LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=12MHz (fsysclk= PLL 24MHz divided by 2) ,	3.3	
		LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=6MHz (PLL 24MHz divided by 4) ,	2.2	
		LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=3MHz (PLL 24MHz divided by 8) ,	1.6	
	I_{DD_N1}	System clock is HIRC f_{HXIN}=0MHz,fsysclk=16 MHz (HIRC 16MHz)	2.7	mA
		System clock is HIRC f_{HXIN}=0MHz,fsysclk=8 MHz (HIRC 16MHz divided by 2)	2.7	
		System clock is HIRC f_{HXIN}=0MHz,fsysclk=4 MHz (HIRC 16MHz divided by 4)	1.4	
		System clock is HIRC f_{HXIN}=0MHz,fsysclk=2 MHz (HIRC 16MHz divided by 8)	1.4	
	I_{DD_N3}	System clock is HXTAL fsysclk=16MHz (HXTAL 16MHz)	3.8	mA
		System clock is HXTAL fsysclk=8MHz (HXTAL 16MHz divided by 2)	2.3	
		System clock is HXTAL fsysclk=4MHz (HXTAL 16MHz divided by 4)	1.5	
		System clock is HXTAL fsysclk=2MHz (HXTAL 16MHz divided by 8)	1.2	

Operating @ 5V,Ta=-25℃				
Parameter	Symbol	Condition	Typ.	Unit
Sleep mode (LIRC on，CPU clock is stopped)	I_{DD_SL0}	LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=24MHz (PLL 24MHz)	2.7	mA
		LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=12MHz (fsysclk= PLL 24MHz divided by 2) ,	1.8	
		LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=6MHz (PLL 24MHz divided by 4) ,	1.4	
		LPIRC is PLL clock source f_{LPIRC}=1MHz,fsysclk=3MHz (PLL 24MHz divided by 8) ,	1	
	I_{DD_SL1}	System clock is HIRC f_{HXIN}=0MHz,HIRC 16 MHz (HIRC 16MHz)	1.3	mA
		System clock is HIRC f_{HXIN}=0MHz,fsysclk=8 MHz (HIRC 16MHz divided by 2)	1.3	
		System clock is HIRC , f_{HXIN}=0MHz,fsysclk=4 MHz (HIRC 16MHz divided by 4)	0.7	
		System clock is HIRC , f_{HXIN}=0MHz,fsysclk=2 MHz (HIRC 16MHz divided by 8)	0.7	
	I_{DD_SL3}	System clock is HXTAL fsysclk=16MHz (HXTAL 16MHz)	2.0	mA
		System clock is HXTAL fsysclk=8MHz (HXTAL 16MHz divided by 2)	1.4	
		System clock is HXTAL fsysclk=4MHz (HXTAL 16MHz divided by 4)	1	
		System clock is HXTAL fsysclk=2MHz (HXTAL 16MHz divided by 8)	0.8	

(G) The RAM area allocate for bootrom.

After power-on or after reset , 0x1000 to 0x1075 are allocated for bootrom usage. The data of this section will be changed. Recommand not use this section for data saving. .

(H) PNIC setting related

When setting peripheral functions with PNIC, please set in the following order: FSELR, PCSELR, and PxFC1, PxFC2.