ABOV SEMICONDUCTOR Co., Ltd.
8-BIT MICROCONTROLLERS

MC97F2664

User's Manual (Ver. 1.7)

REVISION HISTORY

VERSION 0.0 (June 22, 2012)

VERSION 1.0 (July 24, 2012)

Change '±6LSB' to "ILE" in A/D CONVERTER characteristics.
Change '3.0V' to "Vdd @ 0.4-16MHz" in RECOMMENDED OPERATING condition.
Change '4.5/9.0mA (Typ/Max)" to "IDD1 @16MHz Crystal" in DC electrical characteristics.
Change '4.0/8.0mA (Typ/Max)" to "IDD1 @16MHz IRC" in DC electrical characteristics.
Change '2.0/4.0mA (Typ/Max)" to "IDD2 @16MHz Crystal" in DC electrical characteristics.
Change '1.0/2.0mA (Typ/Max)" to "IDD2 @12MHz Crystal" in DC electrical characteristics.
Change '1.2/2.4mA (Typ/Max)" to "IDD2 @16MHz IRC" in DC electrical characteristics.
Change '1.60/1.79V (Typ/Max)" to "V_{LVR}, V_{LVI}" in LVR and LVI characteristics.
Change Figure 10.6 Effective Timing of Interrupt
Change Figure 14.1 Block Diagram of On-Chip Debug System
Change Figure 1.1 Connection of Transmission
Remove 42 PIN DIP package
Remove INTERRUPT GROUP PRIORITY LEVEL at INTERRUPT CONTROLLER
Remove Appendix B
Add 64 PIN LQFP-1414 package
Add RUNFLAG Pull-Down Resistor(R_{PD}) in DC electrical characteristics.

VERSION 1.1 (October 4, 2012)

Modify Figure 7.17 Recommended Circuit and Layout
Add Figure 7.18 Recommended Circuit and Layout with SMPS Power

VERSION 1.2 (January 11, 2013)

Add 0.1uF Bypass capacitor in Internal RC Oscillator characteristics
Change '14/24mA, 10/18mA' to LVR/LVI current in LVR/LVI electrical characteristics.
Change '±5LSB' to "ZOE" in A/D CONVERTER characteristics
Change '±5LSB' to "FSE" in A/D CONVERTER characteristics
Change '10,000/100,000 times' to Endurance of Write/Erase in internal flash rom characteristics.

VERSION 1.3 (March 18, 2014)

Add 64 PIN QFN package, MC97F2664UB
AVREF range changed from 1.8V~VDD to 2.7V~VDD
Figure 10.3 modified
Add contents, "Writing "1" has no effect" in all interrupt flag bits.
Appendix 1 "DJNZ Rn,rel" instruction 3bytes → 2bytes

VERSION 1.4 (March 18, 2014)

AVREF range changed from 2.7V~VDD to 1.8V~VDD

VERSION 1.5 (July 7, 2014)

Removed "PGM Plus USB" in Development Tools.
Modify note of LVRCR in RESET, "The LVRST, LVRVS[3:0] bits are cleared by a power-on reset but are retained by other reset signals".
Change "Read Protection" to "Code Read Protection" in Configure Option.
Change "Hard-Lock" to "Code Write Protection" in Configure Option.
Change "Vector Area Protection" to "Vector Area Write Protection" in Configure Option.
Change "RESETB select" to "Select RESETB pin" in Configure Option.
Change "Protection Area" to "Specific Area Write Protection" in Configure Option.
Add note of "Specific Area Write Protection" in Configure Option, "When PAEN = '1', it is applied."
Fix the typo.

VERSION 1.6 (Sep 24, 2014)

Change MAX value of VDD rising time "30V/ms" to "8V/ms" in Power-on Reset Characteristics
Modify contents of BUZZER
Modify Figure 10.3 Interrupt Sequence Flow
Fix the typo.

VERSION 1.7 (April 13, 2015) This Book

Change MAX value of VDD rising time "8V/ms" to "5V/ms" in Power-on Reset Characteristics
Change a Figure 11.53 A/D Analog Input pin with Capacitor in12-Bit A/D Converter.
Add contents of Flash, "Protection for Invalid Erase/Write".

Version 1.7
Published by FAE Team

Additional information of this manual may be served by ABOV Semiconductor offices in Korea or Distributors.

ABOV Semiconductor reserves the right to make changes to any information here in at any time without notice.

The information, diagrams and other data in this manual are correct and reliable; however, ABOV Semiconductor is in no way responsible for any violations of patents or other rights of the third party generated by the use of this manual.

Table of Contents

List of Figures

List of Tables

MC97F2664

CMOS SINGLE-CHIP 8-BIT MICROCONTROLLER WITH 12-BIT A/D CONVERTER

1. Overview

1.1 Description

The MC97F2664 is advanced CMOS 8-bit microcontroller with 64k bytes of FLASH. This is powerful microcontroller which provides a highly flexible and cost effective solution to many embedded control applications. This provides the following features : 64k bytes of FLASH, 256 bytes of IRAM, 4,096 bytes of XRAM , general purpose I/O, basic interval timer, watchdog timer, 8/16-bit timer/counter, 16-bit PPG output, 8-bit PWM output, watch timer, buzzer driving port, SPI, UART, I2C, 12-bit A/D converter, on-chip POR, LVR, LVI, on-chip oscillator and clock circuitry. The MC97F2664 also supports power saving modes to reduce power consumption.

Device Name	FLASH	XRAM	IRAM	ADC	I/O PORT	Package
MC97F2664L	64k bytes	4,096 bytes	256 bytes	15 channel	61	64 LQFP-1010
MC97F2664L14				15 channel	61	64 LQFP-1414
MC97F2664UB				15 channel	61	64 QFN
MC97F2464Q				10 channel	41	44 MQFP-1010

1.2 Features

- **CPU**
  - 8 Bit CISC Core (8051 Compatible)
- **ROM (FLASH) Capacity**
  - 64k Bytes
  - Flash with self read/write capability
  - On chip debug and In-system programming (ISP)
  - Endurance : 10,000 times (Sector 0~1019)
    100,000 times (Sector 1020~1023)
- **256 Bytes IRAM**
- **4,096 Bytes XRAM**
- **General Purpose I/O (GPIO)**
  - Normal I/O : 61 Ports
    (P0, P1,P2,P3,P4,P5,P6,P7[4:0])
- **Basic Interval Timer (BIT)**
  - 8Bit × 1ch
  - **Watch Dog Timer (WDT)**
  - 8Bit × 1ch
  - 5kHz internal RC oscillator
- **Timer/ Counter**
  - 8Bit × 4ch (T0/T1/T2/T3)
  - 16Bit × 6ch (T4/T5/T6/T7/T8/T9)
- **Programmable Pulse Generation**
  - 8Bit PWM (by T0/T1/T2/T3)
  - Pulse generation (by T4/T5/T6/T7/T8/T9)
- **Watch Timer (WT)**
  - 3.91ms/0.25s/0.5s/1s/1m interval at 32.768kHz
- **Buzzer**
  - 8Bit × 1ch
- **SPI**
  - 8Bit × 2ch
- **UART**
  - 8Bit × 3ch
- **USI (UART + SPI + I2C)**
  - 8Bit UART × 2ch, 8Bit SPI × 2ch, and I2C × 2ch
- **12 Bit A/D Converter**
  - 15 Input channels
- **Power On Reset**
  - Reset release level (1.4V)
- **Low Voltage Reset**
  - 14 level detect (1.60V/ 2.00V/ 2.10V/ 2.20V/ 2.32V/ 2.44V/ 2.59V/ 2.75V/ 2.93V/ 3.14V/ 3.38V/ 3.67V/ 4.00V/ 4.40V)
- **Low Voltage Indicator**
  - 13 level detect (2.00V/ 2.10V/ 2.20V/ 2.32V/ 2.44V/ 2.59V/ 2.75V/ 2.93V/ 3.14V/ 3.38V/ 3.67V/ 4.00V/ 4.40V)
- **Interrupt Sources**
  - External Interrupts
    (EINT0~A, EINT10~19) (21)
  - Timer(0/1/2/3/4/5/6/7/8/9) (14)
  - WDT (1)
  - BIT (1)
  - WT (1)
  - SPI 2/3 (2)
  - UART 2/3/4 (6)
  - USI0/1 (4)
  - ADC (1)
  - Stack OVF(1)
- **Internal RC Oscillator**
  - Internal RC frequency:
    16MHz ±1.5% (T_A= 0 ~ +50°C)
- **Power Down Mode**
  - STOP, IDLE mode
- **Operating Voltage and Frequency**
  - 1.8V ~ 5.5V (@32 ~ 38kHz with X-tal)
  - 1.8V ~ 5.5V (@0.4 ~ 4.2MHz with X-tal)
  - 2.7V ~ 5.5V (@0.4 ~ 12.0MHz with X-tal)
  - 3.0V ~ 5.5V (@0.4 ~ 16.0MHz with X-tal)
  - 1.8V ~ 5.5V (@0.5 ~ 8.0MHz with Internal RC)
  - 2.0V ~ 5.5V (@0.5 ~ 16.0MHz with Internal RC)
  - Voltage dropout converter included for core
- **Minimum Instruction Execution Time**
  - 125nS (@ 16MHz main clock)
  - 61us (@t 32.768kHz sub clock)
- **Operating Temperature:** – 40 ~ + 85℃
- **Oscillator Type**
  - 0.4-16MHz Crystal or Ceramic for main clock
  - 32.768kHz Crystal for sub clock
- **Package Type**
  - 64 LQFP-1010
  - 64 LQFP-1414
  - 64 QFN
  - 44 MQFP-1010
  - Pb-free package

1.3 Ordering Information

Table 1-1 Ordering Information of MC97F2664

Device name	ROM size	IRAM size	XRAM size	Package
MC97F2664L	64k bytes FLASH	256 bytes	4,096 bytes	64 LQFP-1010
MC97F2664L14				64 LQFP-1414
MC97F2664UB				64 QFN
MC97F2464Q				44 MQFP-1010

1.4 Development Tools

1.4.1 Compiler

We do not provide the compiler. Please contact the third parties.

The core of MC97F2664 is Mentor 8051. And, device ROM size is smaller than 64k bytes. Developer can use all kinds of third party's standard 8051 compiler.

1.4.2 OCD2 emulator and debugger

The OCD2 (On Chip Debug 2) emulator supports ABOV Semiconductor's 8051 series MCU emulation.

The OCD2 interface uses two-wire interfacing between PC and MCU which is attached to user's system. The OCD2 can read or change the value of MCU internal memory and I/O peripherals. And the OCD2 also controls MCU internal debugging logic, it means OCD2 controls emulation, step run, monitoring, etc.

The OCD2 Debugger program works on Microsoft-Windows NT, 2000, XP, Vista (32bit), 7 operating system.

If you want to see more details, please refer to OCD2 debugger manual. You can download debugger S/W and manual from our web-site.

Connection:

- DSCL (MC97F2664 P62 port)
- DSDA (MC97F2664 P63 port)
- RTIME (MC97F2664 RUNFLAG port, Option)

  NOTE) MC97F2664 Use Only OCD2.

OCD2 connector diagram: Connect OCD2 with user system

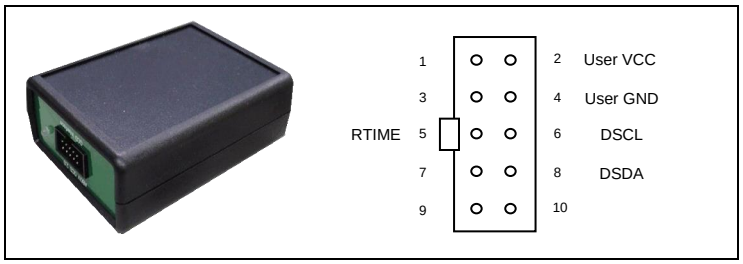


Figure 1.1 OCD2 Debugger and Pin Description

1.4.3 Programmer

Single programmer:

StandAlone PGMplus: It programs MCU device directly.

Figure 1.2 StandAlone PGMplus (Single Writer)

OCD emulator: It can write code in MCU device too, because OCD debugging supports ISP (In System Programming).

It does not require additional H/W, except developer's target system.

Gang programmer:

It programs 8 MCU devices at once.

So, it is mainly used in mass production line.

Gang programmer is standalone type, it means it does not require host PC, after a program is downloaded from host PC to Gang programmer.

Figure 1.3 StandAlone Gang8 (for Mass Production)

2. Block Diagram

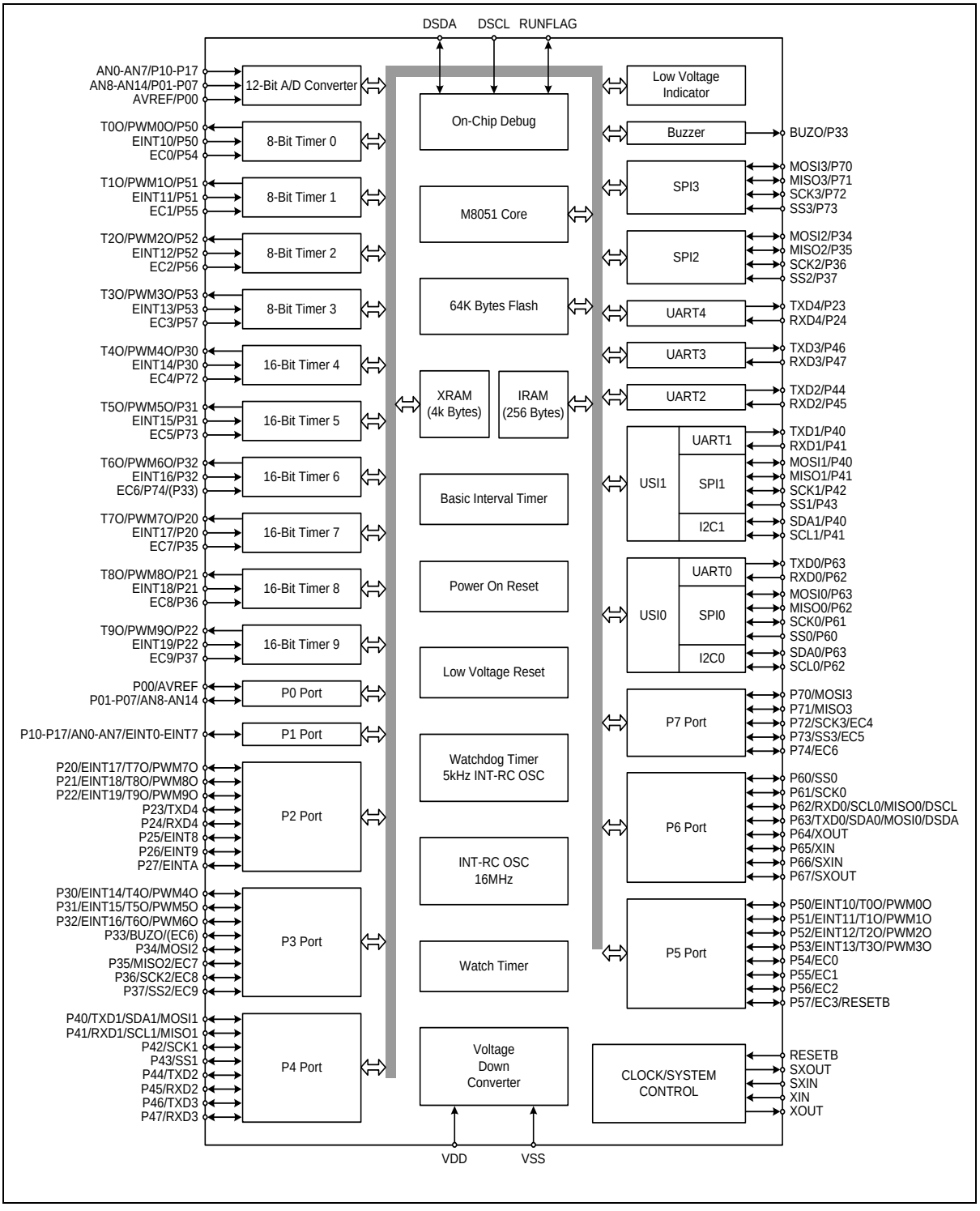


Figure 2.1 Block Diagram

NOTE) The P03-P07, P23-P27, P52-P56, and P7 are not in the 44-pin package.

3. Pin Assignment

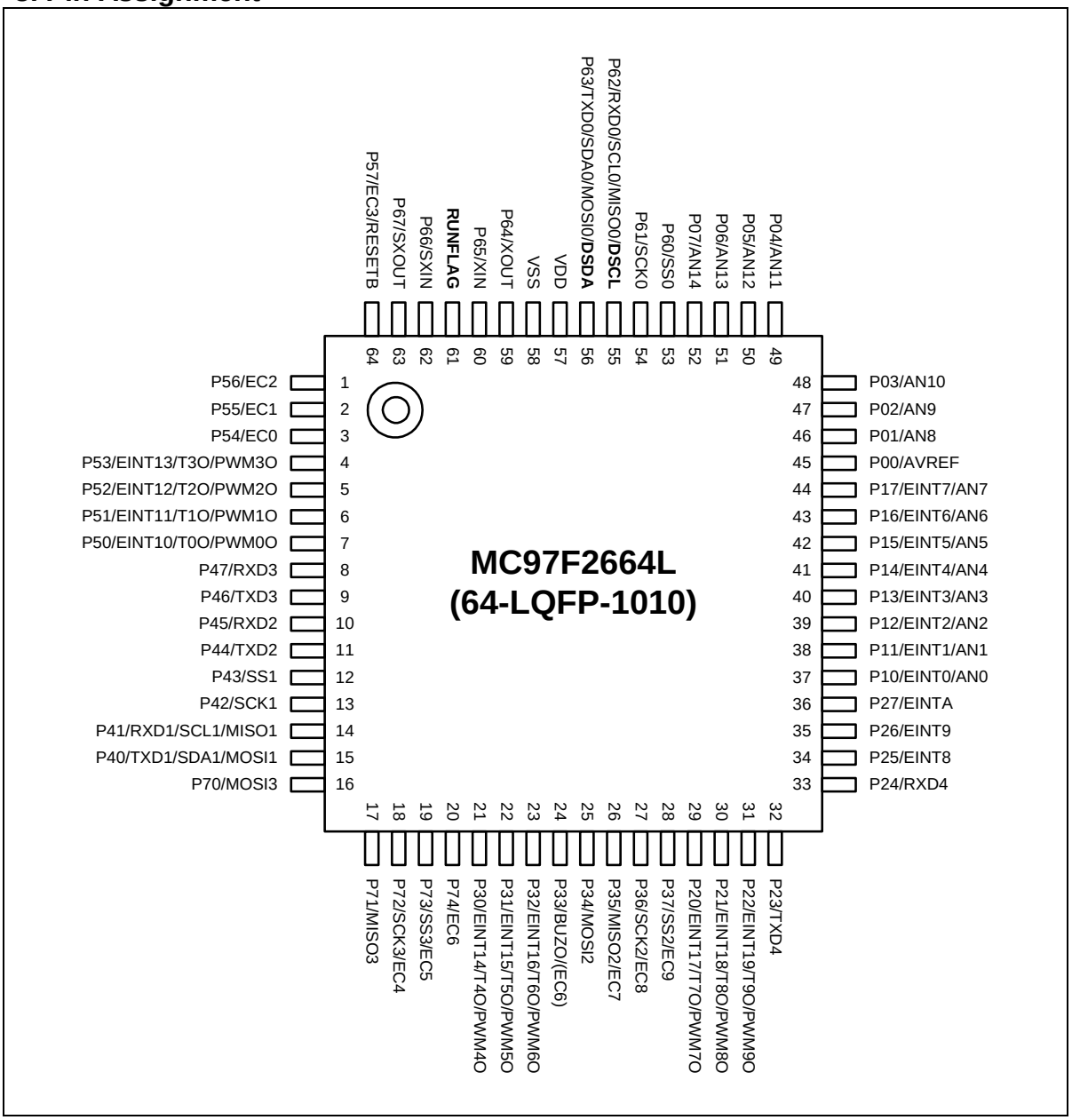


Figure 3.1 MC97F2664L 64LQFP-1010 Pin Assignment

NOTES) 1. On On-Chip Debugging, ISP uses P6[3:2] pin as DSDA, DSCL.
2. The pin in parentheses can be configured by software control.

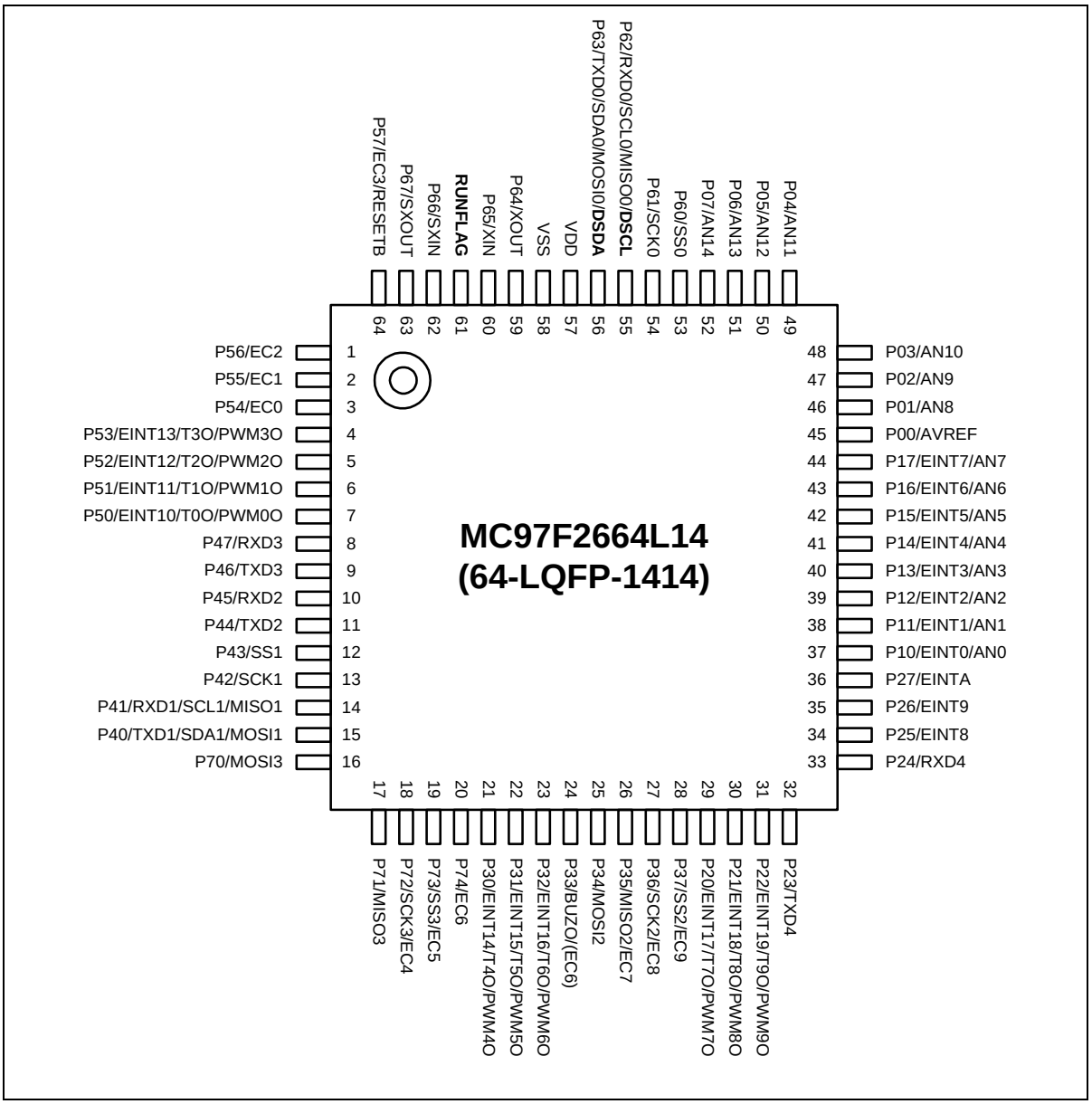


Figure 3.2 MC97F2664L14 64LQFP-1414 Pin Assignment

NOTES) 1. On On-Chip Debugging, ISP uses P6[3:2] pin as DSDA, DSCL.
2. The pin in parentheses can be configured by software control.

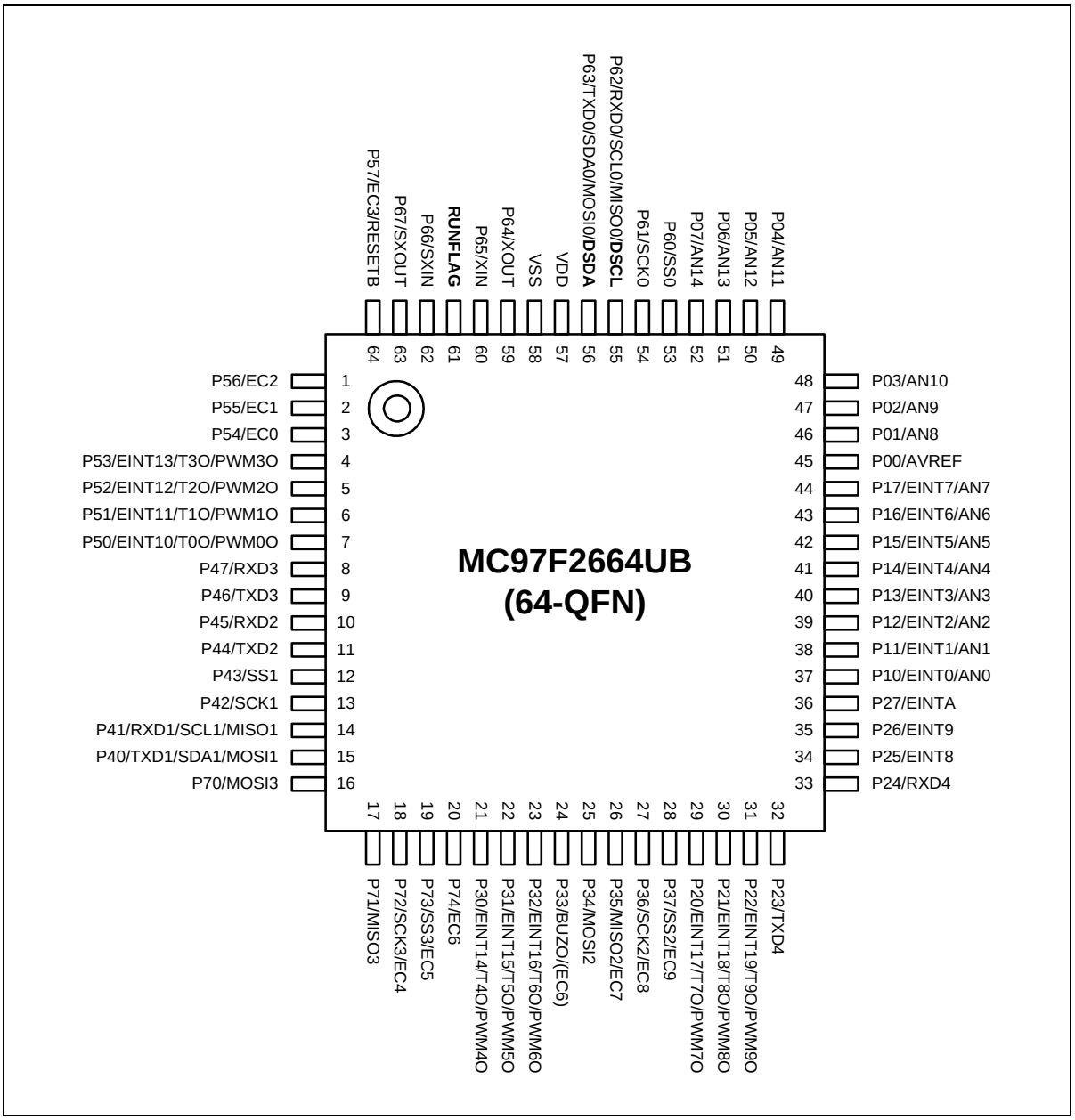


Figure 3.3 MC97F2664UB 64QFN Pin Assignment

NOTES) 1. On On-Chip Debugging, ISP uses P6[3:2] pin as DSDA, DSCL.
2. The pin in parentheses can be configured by software control.

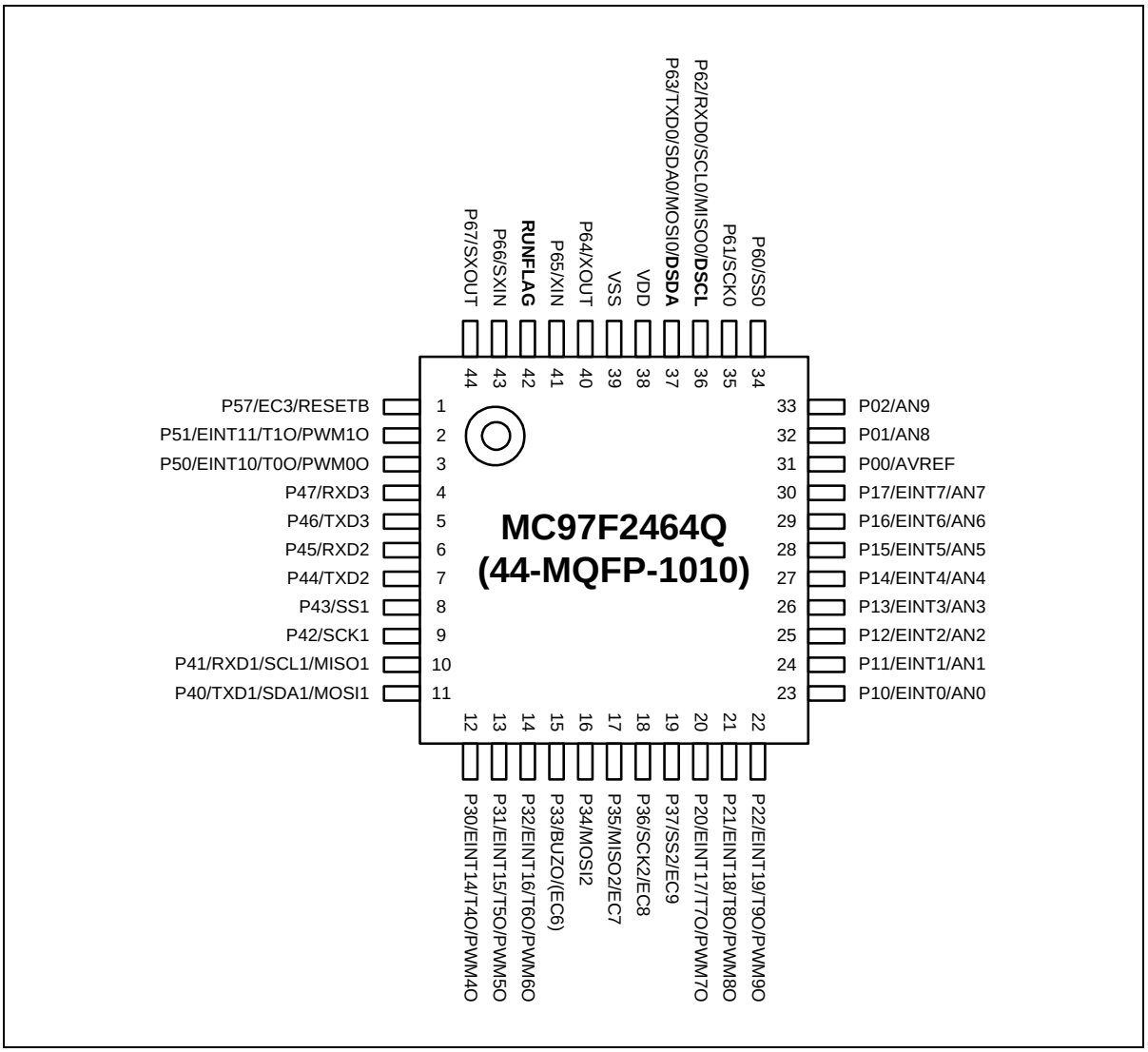
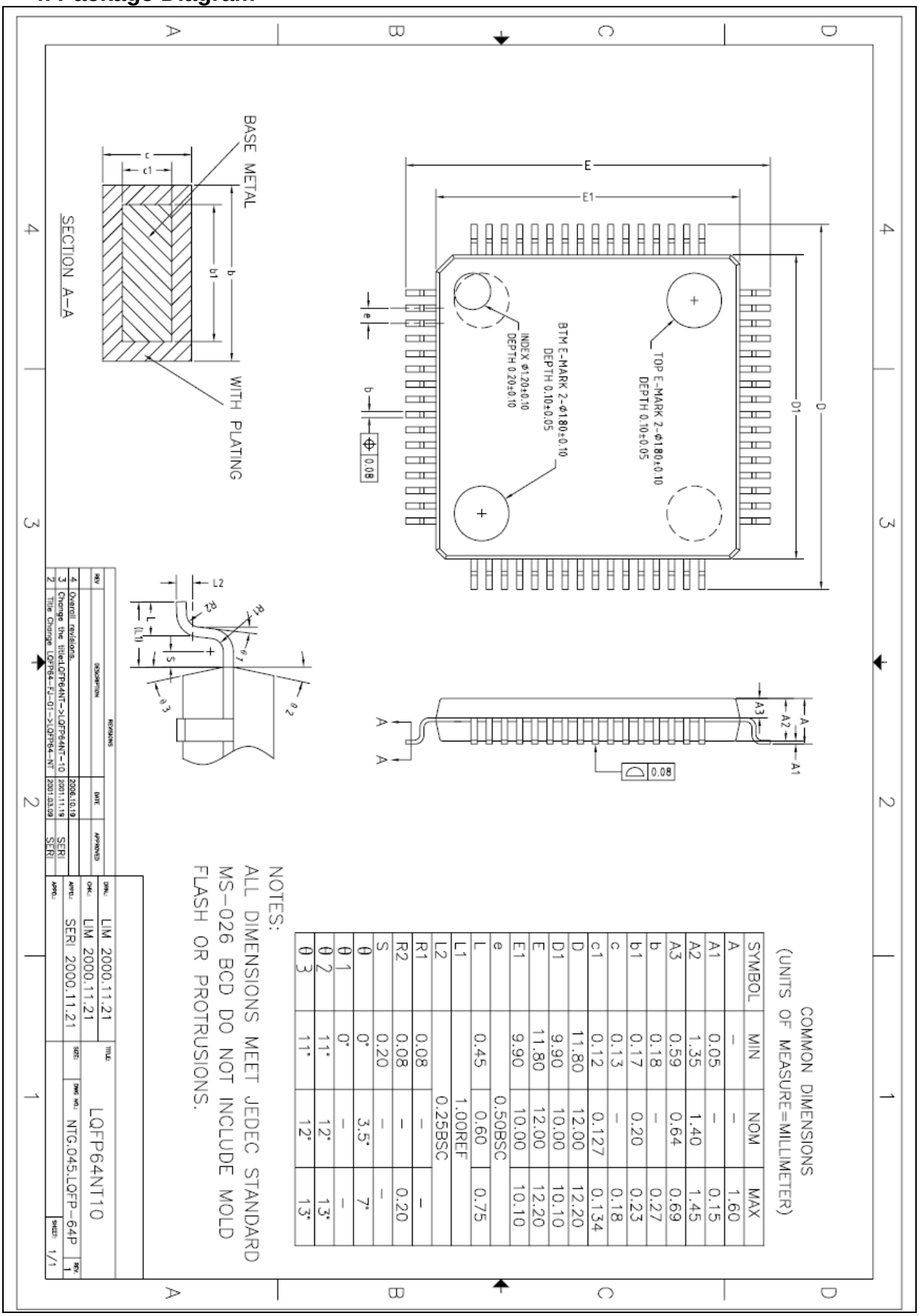


Figure 3.4 MC97F2464 44MQFP-1010 Pin Assignment

NOTES) 1. On On-Chip Debugging, ISP uses P6[3:2] pin as DSDA, DSCL.
2. The P03-P07, P23-P27, P52-P56, and P7 pins should be selected as a push-pull output or an input with pull-up resistor by software control when the 44-pin package is used.

4. Package Diagram



COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.18	–	0.27
b1	0.17	0.20	0.23
c	0.13	–	0.18
c1	0.12	0.127	0.134
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e		0.50BSC	
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R1	0.08	–	–
R2	0.08	–	0.20
S	0.20	–	–
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°

Figure 4.1 64-Pin LQFP-1010 Package

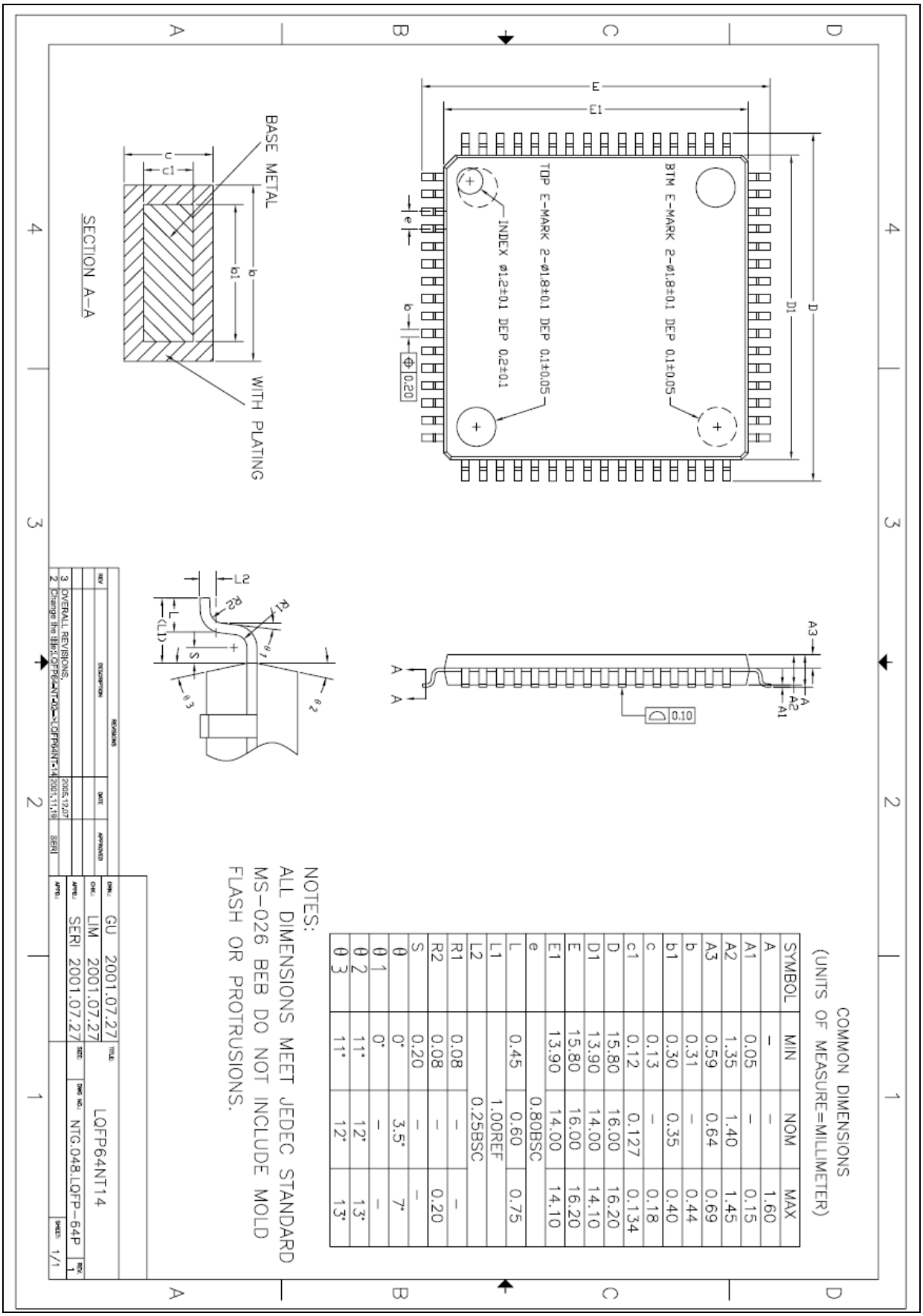
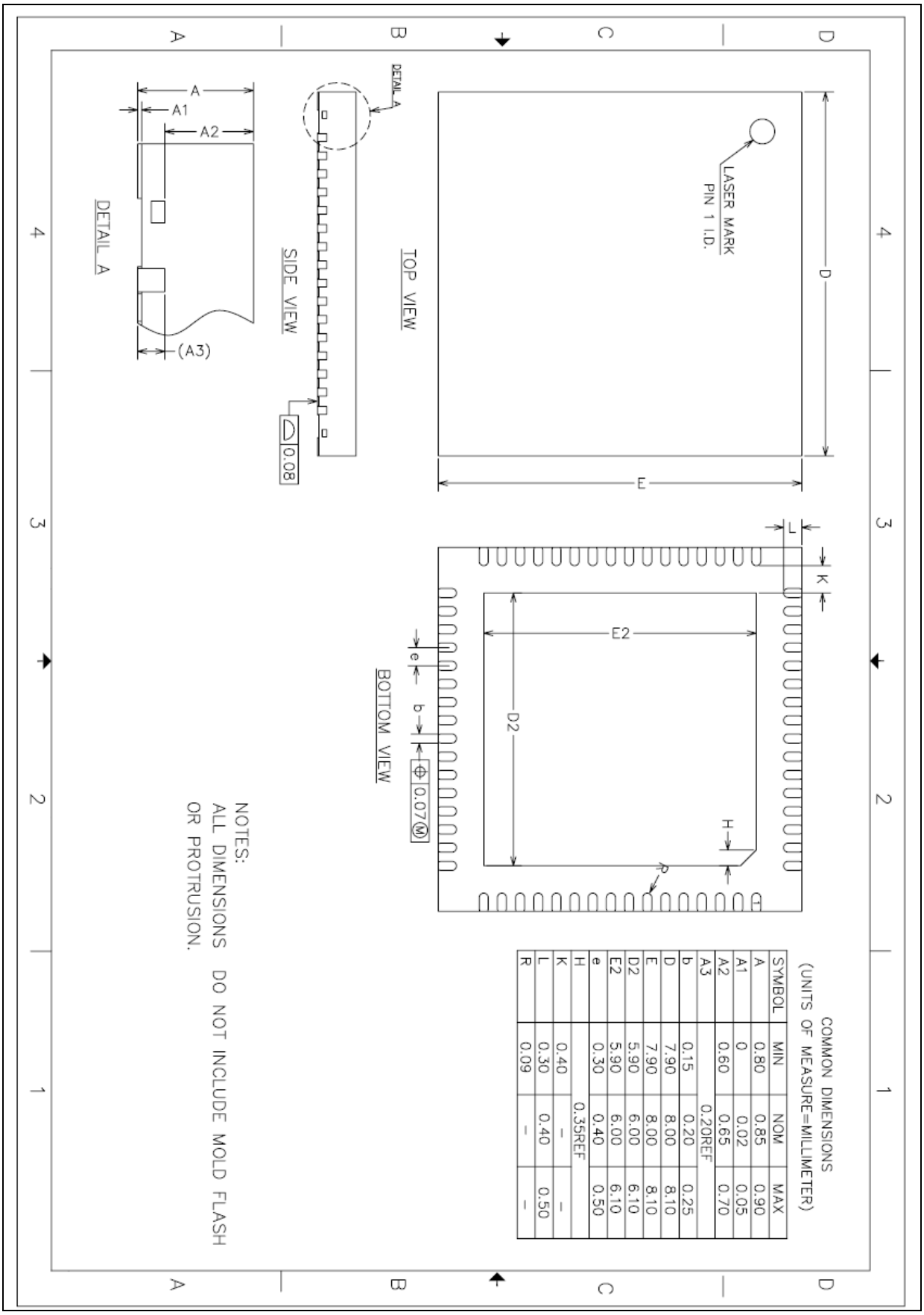
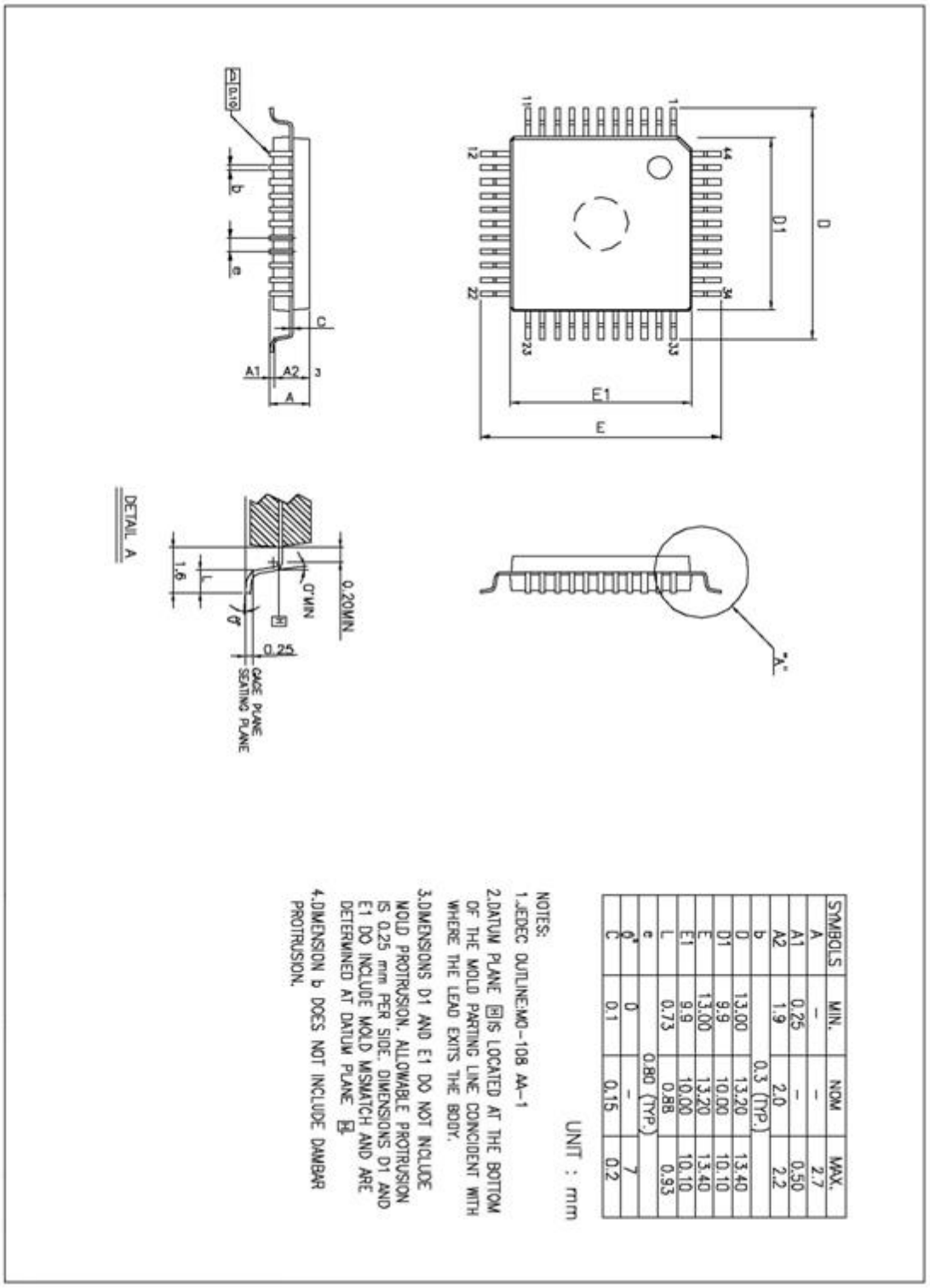


Figure 4.2 64-Pin LQFP-1414 Package



SYMBOL	MIN	NOM	MAX
A	0.80	0.85	0.90
A1	0	0.02	0.05
A2	0.60	0.65	0.70
A3		0.20REF	
b	0.15	0.20	0.25
D	7.90	8.00	8.10
E	7.90	8.00	8.10
D2	5.90	6.00	6.10
E2	5.90	6.00	6.10
e	0.30	0.40	0.50
H		0.35REF	
K	0.40	–	–
L	0.30	0.40	0.50
R	0.09	–	–

Figure 4.3 64-Pin QFN Package



SYMBOLS	MIN.	NOM.	MAX.
A	–	–	2.7
A1	0.25	–	0.50
A2	1.9	2.0	2.2
b		0.3 (TYP.)	
D	13.00	13.20	13.40
D1	9.9	10.00	10.10
E	13.00	13.20	13.40
E1	9.9	10.00	10.10
L	0.73	0.88	0.93
e		0.80 (TYP.)	
θ°	0	–	7
c	0.1	0.15	0.2

UNIT : mm

NOTES:

1.JEDEC OUTLINE:MO-108 AA-1

2.DATUM PLANE [H] IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.

3.DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE [H].

4.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

Figure 4.4 44-Pin MQFP-1010 Package

5. Pin Description

Table 5-1 Normal Pin Description

PIN Name	I/O	Function	@RESET	Shared with
P00	I/O	Port 0 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit. The P03-P07 are not in the 44-Pin package.	Input	AVERF
P01				AN8
P02				AN9
P03				AN10
P04				AN11
P05				AN12
P06				AN13
P07				AN14
P10	I/O	Port 1 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	AN0/EINT0
P11				AN1/EINT1
P12				AN2/EINT2
P13				AN3/EINT3
P14				AN4/EINT4
P15				AN5/EINT5
P16				AN6/EINT6
P17				AN7/EINT7
P20	I/O	Port 2 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit. The P23-P27 are not in the 44-Pin package.	Input	EINT17/T7O/PWM7O
P21				EINT18/T8O/PWM8O
P22				EINT19/T9O/PWM9O
P23				TXD4
P24				RXD4
P25				EINT8
P26				EINT9
P27				EINTA
P30	I/O	Port 3 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	EINT14/T4O/PWM4O
P31				EINT15/T5O/PWM5O
P32				EINT16/T6O/PWM6O
P33				BUZO/(EC6)
P34				MOSI2
P35				MISO2/EC7
P36				SCK2/EC8
P37				SS2/EC9
P40	I/O	Port 4 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	TXD1/SDA1/MOSI1
P41				RXD1/SCL1/MISO1
P42				SCK1
P43				SS1
P44				TXD2
P45				RXD2
P46				TXD3
P47				RXD3

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
P50	I/O	Port 5 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit. The P52-P56 are not in the 44-Pin package.	Input	EINT10/T0O/PWM0O
P51				EINT11/T1O/PWM1O
P52				EINT12/T2O/PWM2O
P53				EINT13/T3O/PWM3O
P54				EC0
P55				EC1
P56				EC2
P57				EC3/RESETB
P60	I/O	Port 6 is a bit-programmable I/O port which can be configured as a schmitt-trigger input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	SS0
P61				SCK0
P62				RXD0/SCL0/MISO0/DSCL
P63				TXD0/SDA0/MOSI0/DSDA
P64				XOUT
P65				XIN
P66				SXIN
P67				SXOUT
P70	I/O	Port 7 is a bit-programmable I/O port which can be configured as an input, a push-pull output, or an open-drain output. A pull-up resistor can be specified in 1-bit unit.	Input	MOSI3
P71				MISO3
P72				SCK3/EC4
P73				SS3/EC5
P74				EC6
EINT0	I/O	External interrupt inputs	Input	P10/AN0
EINT1				P11/AN1
EINT2				P12/AN2
EINT3				P13/AN3
EINT4				P14/AN4
EINT5				P15/AN5
EINT6				P16/AN6
EINT7				P17/AN7
EINT8				P25
EINT9				P26
EINTA				P27
EINT10	I/O	External interrupt and Timer 0 capture input	Input	P50/T0O/PWM0O
EINT11	I/O	External interrupt and Timer 1 capture input	Input	P51/T1O/PWM1O
EINT12	I/O	External interrupt and Timer 2 capture input	Input	P52/T2O/PWM2O

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
EINT13	I/O	External interrupt and Timer 3 capture input	Input	P53/T3O/PWM3O
EINT14	I/O	External interrupt and Timer 4 capture input	Input	P30/T4O/PWM4O
EINT15	I/O	External interrupt and Timer 5 capture input	Input	P31/T5O/PWM5O
EINT16	I/O	External interrupt and Timer 6 capture input	Input	P32/T6O/PWM6O
EINT17	I/O	External interrupt and Timer 7 capture input	Input	P20/T7O/PWM7O
EINT18	I/O	External interrupt and Timer 8 capture input	Input	P21/T8O/PWM8O
EINT19	I/O	External interrupt and Timer 9 capture input	Input	P22/T9O/PWM9O
T0O	I/O	Timer 0 interval output	Input	P50/EINT0/PWM0O
T1O	I/O	Timer 1 interval output	Input	P51/EINT1/PWM1O
T2O	I/O	Timer 2 interval output	Input	P52/EINT2/PWM2O
T3O	I/O	Timer 3 interval output	Input	P53/EINT3/PWM3O
T4O	I/O	Timer 4 interval output	Input	P30/EINT4/PWM4O
T5O	I/O	Timer 5 interval output	Input	P31/EINT5/PWM5O
T6O	I/O	Timer 6 interval output	Input	P32/EINT6/PWM6O
T7O	I/O	Timer 7 interval output	Input	P20/EINT7/PWM7O
T8O	I/O	Timer 8 interval output	Input	P21/EINT8/PWM8O
T9O	I/O	Timer 9 interval output	Input	P22/EINT9/PWM9O
PWM0O	I/O	Timer 0 pulse output	Input	P50/EINT0/T0O
PWM1O	I/O	Timer 1 pulse output	Input	P51/EINT1/T1O
PWM2O	I/O	Timer 2 pulse output	Input	P52/EINT2/T2O
PWM3O	I/O	Timer 3 pulse output	Input	P53/EINT3/T3O
PWM4O	I/O	Timer 4 pulse output	Input	P30/EINT4/T4O
PWM5O	I/O	Timer 5 pulse output	Input	P31/EINT5/T5O
PWM6O	I/O	Timer 6 pulse output	Input	P32/EINT6/T6O
PWM7O	I/O	Timer 7 pulse output	Input	P20/EINT7/T7O
PWM8O	I/O	Timer 8 pulse output	Input	P21/EINT8/T8O
PWM9O	I/O	Timer 9 pulse output	Input	P22/EINT9/T9O
EC0	I/O	Timer 0 event count input	Input	P54
EC1	I/O	Timer 1 event count input	Input	P55
EC2	I/O	Timer 2 event count input	Input	P56
EC3	I/O	Timer 3 event count input	Input	P57/RESETB
EC4	I/O	Timer 4 event count input	Input	P72/SCK3
EC5	I/O	Timer 5 event count input	Input	P73/SS3
EC6	I/O	Timer 6 event count input	Input	P74
EC7	I/O	Timer 7 event count input	Input	P35/MISO2
EC8	I/O	Timer 8 event count input	Input	P36/SCK2
EC9	I/O	Timer 9 event count input	Input	P37/SS2
BUZO	I/O	Buzzer signal output	Input	P33/(EC6)
SCK0	I/O	Serial 0 clock input/output	Input	P61
SCK1	I/O	Serial 1 clock input/output	Input	P42
SCK2	I/O	Serial 2 clock input/output	Input	P36/EC8

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
SCK3	I/O	Serial 3 clock input/output	Input	P72/EC4
MOSI0	I/O	SPI 0 master output, slave input	Input	P63/TXD0/SDA0/DSDA
MOSI1	I/O	SPI 1 master output, slave input	Input	P40/TXD1/SDA1
MOSI2	I/O	SPI 2 master output, slave input	Input	P34
MOSI3	I/O	SPI 3 master output, slave input	Input	P70
MISO0	I/O	SPI 0 master input, slave output	Input	P62/RXD0/SCL0/DSCL
MISO1	I/O	SPI 1 master input, slave output	Input	P41/RXD1/SCL1
MISO2	I/O	SPI 2 master input, slave output	Input	P35/EC7
MISO3	I/O	SPI 3 master input, slave output	Input	P71
SS0	I/O	SPI 0 slave select input	Input	P60
SS1	I/O	SPI 1 slave select input	Input	P43
SS2	I/O	SPI 2 slave select input	Input	P37/EC9
SS3	I/O	SPI 3 slave select input	Input	P73/EC5
TXD0	I/O	UART 0 data output	Input	P63/SDA0/MOSI0/DSDA
TXD1	I/O	UART 1 data output	Input	P40/SDA1/MOSI1
TXD2	I/O	UART 2 data output	Input	P44
TXD3	I/O	UART 3 data output	Input	P46
TXD4	I/O	UART 4 data output	Input	P23
RXD0	I/O	UART 0 data input	Input	P62/SCL0/MISO0/DSCL
RXD1	I/O	UART 1 data input	Input	P41/SCL1/MISO1
RXD2	I/O	UART 2 data input	Input	P45
RXD3	I/O	UART 3 data input	Input	P47
RXD4	I/O	UART 4 data input	Input	P24
SCL0	I/O	I2C 0 clock input/output	Input	P62/RXD0/MISO0/DSCL
SCL1	I/O	I2C 1 clock input/output	Input	P41/RXD1/MISO1
SDA0	I/O	I2C 0 data input/output	Input	P63/TXD0/MOSI0/DSDA
SDA1	I/O	I2C 1 data input/output	Input	P40/TXD1/MOSI1
AVREF	I/O	A/D converter reference voltage	Input	P00
AN0	I/O	A/D converter analog input channels	Input	P10/EINT0
AN1				P11/EINT1
AN2				P12/EINT2
AN3				P13/EINT3
AN4				P14/EINT4
AN5				P15/EINT5
AN6				P16/EINT6
AN7				P17/EINT7
AN8				P01
AN9				P02
AN10				P03
AN11				P04
AN12				P05

Table 5-1 Normal Pin Description (Continued)

PIN Name	I/O	Function	@RESET	Shared with
A13	I/O	A/D converter analog input channels	Input	P06
A14				P07
RESETB	I/O	System reset pin with a pull-up resistor when it is selected as the RESETB by CONFIGURE OPTION	Input	P57/EC3
DSDA	I/O	On chip debugger data input/output (NOTE3)	Input	P63/TXD0/SDA0/MOSI0
DSCL	I/O	On chip debugger clock input (NOTE3)	Input	P62/RXD0/SCL0/MISO0
RUNFLAG	I/O	On chip debugger run flag(NOTE3) with a pull-down resistor	Input	–
XIN	I/O	Main oscillator pins	Input	P65
XOUT				P64
SXIN	I/O	Sub oscillator pins	Input	P66
SXOUT				P67
VDD, VSS	–	Power input pins	–	–

NOTES) 1. The P03-P07, P22-P26, P53-P57, and P7are not in the 44-Pin package.
2. The P57/EC3/RESETB pin is configured as one of the P57/EC3 and the RESETB pin by the “CONFIGURE OPTION”.
3. If the P63/TXD0/SDA0/MOSI0/DSDA, P62/RXD0/SCL0/MISO0/DSCL, and RUNFLAG pins are connected to an emulator, the pins are automatically configured as the debugger pins.
4. The P64/XOUT, P65/XIN, P66/SXIN, and P67/SXOUT pins are configured as a function pin by a software control.

6. Port Structures

6.1 General Purpose I/O Port

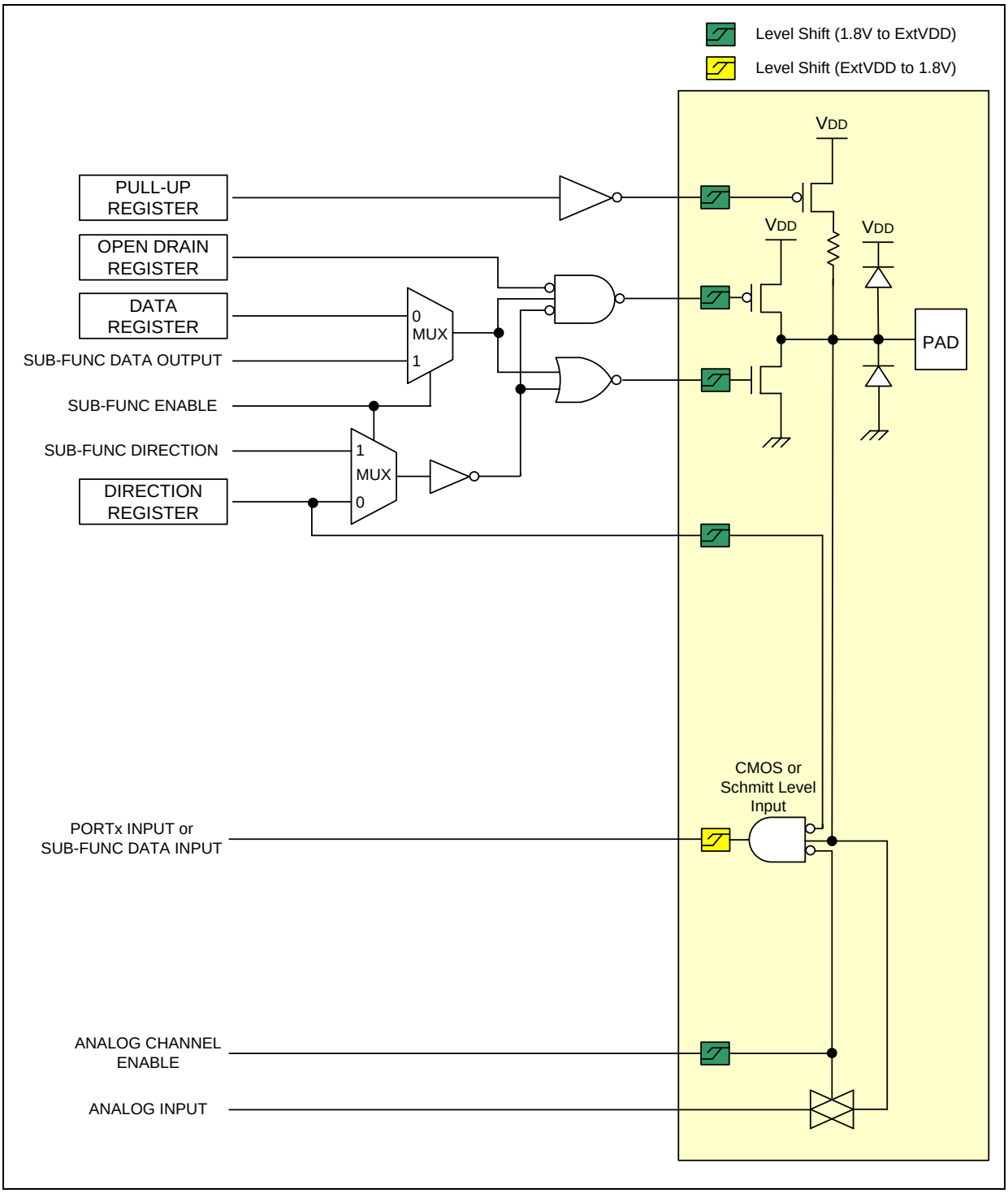


Figure 6.1 General Purpose I/O Port

6.2 External Interrupt I/O Port

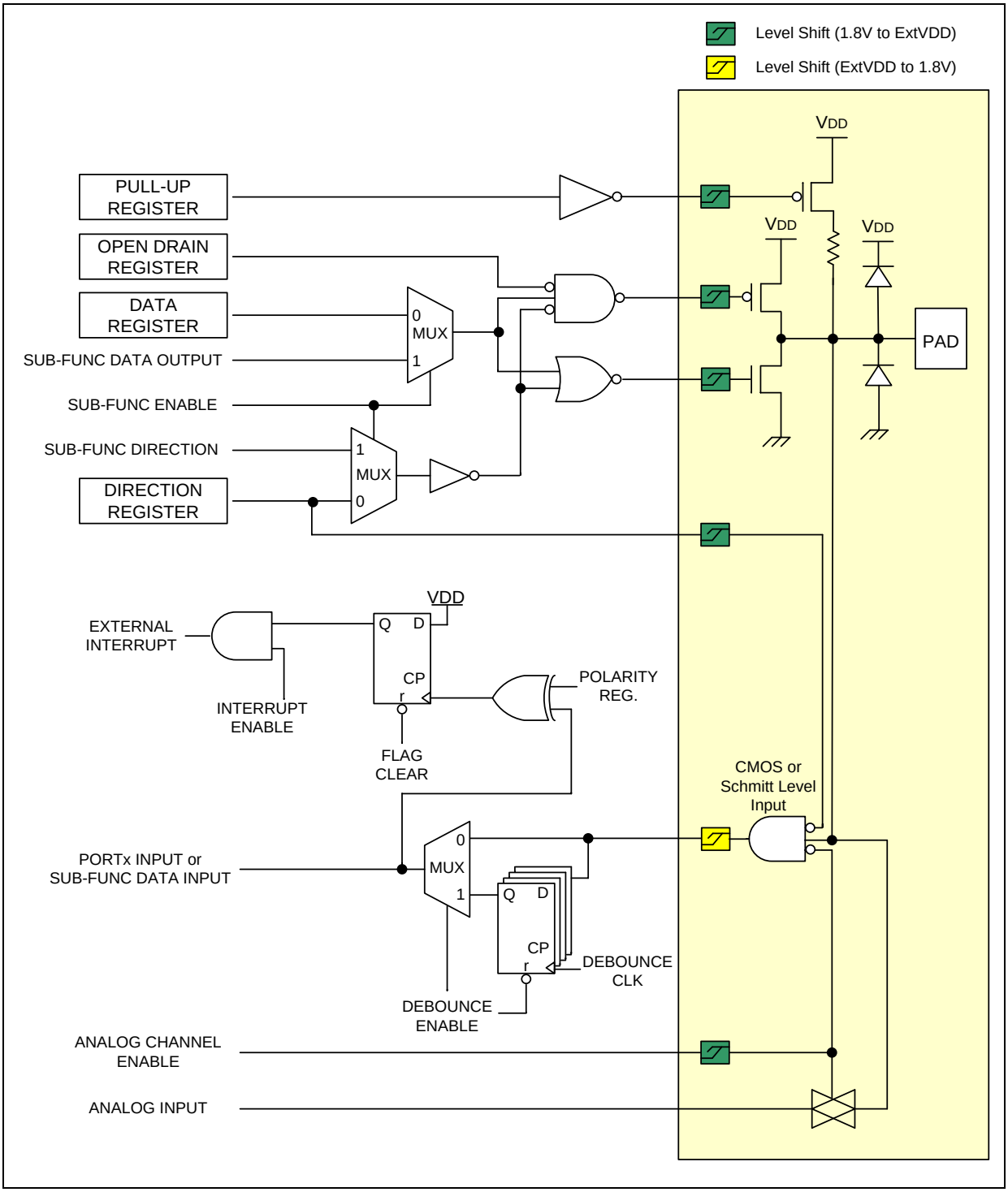


Figure 6.2 External Interrupt I/O Port

7. Electrical Characteristics

7.1 Absolute Maximum Ratings

Table 7-1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	Note
Supply Voltage	VDD	-0.3 ~ +6.5	V	–
Normal Voltage Pin	V_I	-0.3 ~ VDD+0.3	V	Voltage on any pin with respect to VSS
	V_O	-0.3 ~ VDD+0.3	V	
	I_{OH}	-10	mA	Maximum current output sourced by (I_{OH} per I/O pin)
	$\sum I_{OH}$	-80	mA	Maximum current ($\sum I_{OH}$)
	I_{OL}	60	mA	Maximum current sunk by (I_{OL} per I/O pin)
	$\sum I_{OL}$	120	mA	Maximum current ($\sum I_{OL}$)
Total Power Dissipation	P_T	600	mW	–
Storage Temperature	T_{STG}	-65 ~ +150	°C	–

NOTE) Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

7.2 Recommended Operating Conditions

Table 7-2 Recommended Operating Conditions

(T_A= -40°C ~ +85°C)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Operating Voltage	VDD	f_X= 32 ~ 38kHz	SX-tal	1.8	–	5.5	V
		f_X= 0.4 ~ 4.2MHz	X-tal	1.8	–	5.5	
		f_X= 0.4 ~ 12.0MHz		2.7	–	5.5	
		fx= 0.4 ~ 16MHz		3.0	–	5.5	
		f_X= 0.5 ~ 8.0MHz	Internal RC	1.8	–	5.5	
		f_X= 0.5 ~ 16.0MHz		2.0	–	5.5	
Operating Temperature	T_{OPR}	VDD= 1.8 ~ 5.5V		-40	–	85	°C

7.3 A/D Converter Characteristics

Table 7-3 A/D Converter Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Resolution	–	–		–-	12	–	bit
Integral Linear Error	ILE	AVREF= 2.7V – 5.5V fx= 8MHz		–	–	±6	LSB
Differential Linearity Error	DLE			–	–	±1	
Zero Offset Error	ZOE			–	–	±5	
Full Scale Error	FSE			–	–	±5	
Conversion Time	t_{CON}	12bit resolution, 8MHz		20	–	–	us
Analog Input Voltage	V_{AN}	–		VSS	–	AVREF	V
Analog Reference Voltage	AVREF	*Note 3		1.8	–	VDD	
VDD18	–	–		–	1.8	–	V
Analog Input Leakage Current	I_{AN}	AVREF= 5.12V		–	–	2	uA
ADC Operating Current	I_{ADC}	Enable	VDD= 5.12V	–	1	2	mA
		Disable		–	–	0.1	uA

NOTES) 1. Zero offset error is the difference between 000000000000 and the converted output for zero input voltage (VSS).

2. Full scale error is the difference between 111111111111 and the converted output for full-scale input voltage (AVREF).

3. When AVREF is lower than 2.7V, the ADC resolution is worse.

7.4 Power-On Reset Characteristics

Table 7-4 Power-on Reset Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESET Release Level	V_{POR}	–	–	1.4	–	V
VDD Voltage Rising Time	t_R	–	0.05	–	5	V/ms
POR Current	I_{POR}	–	–	0.2	–	uA

7.5 Low Voltage Reset and Low Voltage Indicator Characteristics

Table 7-5 LVR and LVI Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Detection Level	V_{LVR} V_{LVI}	The LVR can select all levels but LVI can select other levels except 1.60V.		–	1.60	1.79	V
				1.85	2.00	2.15	
				1.95	2.10	2.25	
				2.05	2.20	2.35	
				2.17	2.32	2.47	
				2.29	2.44	2.59	
				2.39	2.59	2.79	
				2.55	2.75	2.95	
				2.73	2.93	3.13	
				2.94	3.14	3.34	
				3.18	3.38	3.58	
				3.37	3.67	3.97	
				3.70	4.00	4.30	
				4.10	4.40	4.70	
Hysteresis	△V	–		–	50	150	mV
Minimum Pulse Width	t_{LW}	–		100	–	–	us
LVR and LVI Current	I_{BL}	Enable (Both)	VDD= 3V, RUN Mode	–	14.0	24.0	uA
		Enable (One of two)		–	10.0	18.0	
		Disable (Both)	VDD= 3V	–	–	0.1	

7.6 Internal RC Oscillator Characteristics

Table 7-6 High Internal RC Oscillator Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Frequency	f_{IRC}	V_{DD} = 2.0 – 5.5 V		–	16	–	MHz
Tolerance	–	T_A = 0°C to +50 °C	With 0.1uF Bypass capacitor	–	–	±1.5	%
		T_A = -20°C to +85 °C				±2.5	
		T_A = -40°C to +85 °C				±3.5	
Clock Duty Ratio	TOD	–		40	50	60	%
Stabilization Time	T_{HFS}	–		–	–	100	us
IRC Current	I_{IRC}	Enable		–	0.2	–	mA
		Disable		–	–	0.1	uA

NOTE) A 0.1uF bypass capacitor should be connected to VDD and VSS. Refer to the “7.21 Recommended Circuit and Layout".

7.7 Internal Watch-Dog Timer RC Oscillator Characteristics

Table 7-7 Internal WDTRC Oscillator Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Frequency	f_{WDTRC}	–	2	5	10	kHz
Stabilization Time	t_{WDTS}	–	–	–	1	ms
WDTRC Current	I_{WDTRC}	Enable	–	1	–	uA
		Disable	–	–	0.1	

7.8 DC Characteristics

Table 7-8 DC Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V, f_{XIN}= 16MHz)

Parameter	Symbol	Conditions		MIN	TYP	MAX	Unit
Input High Voltage	V_{IH1}	P1, P2, P3, P5, P6, RESETB		0.8VDD	–	VDD	V
	V_{IH2}	All input pins except VIH1		0.7VDD	–	VDD	V
Input Low Voltage	V_{IL1}	P1, P2, P3, P5, P6, RESETB		–	–	0.2VDD	V
	V_{IL2}	All input pins except VIL1		–	–	0.3VDD	V
Output High Voltage	V_{OH}	VDD= 4.5V, I_{OH}= -2mA, All output ports;		VDD-1.0	–	–	V
Output Low Voltage	V_{OL1}	VDD=4.5V, I_{OL}= 5mA; All output ports except V_{OL2}		–	–	1.0	V
	V_{OL2}	VDD= 4.5V, I_{OL}= 15mA ; P2,P3,P50-53		–	–	1.0	V
Input High Leakage Current	I_{IH}	All input ports		–	–	1	uA
Input Low Leakage Current	I_{IL}	All input ports		-1	–	–	uA
Pull-Up Resistor	R_{PU1}	VI=0V, T_A= 25°C All Input ports	VDD=5.0V	25	50	100	kΩ
			VDD=3.0V	50	100	200	
	R_{PU2}	VI=0V, T_A= 25°C RESETB	VDD=5.0V	150	250	400	kΩ
			VDD=3.0V	300	500	700	
Pull-Down Resistor	R_{PD}	VDD= 5.0V, T_A= 25°C RUNFLAG		10	20	30	kΩ
OSC feedback resistor	R_{X1}	XIN= VDD, XOUT= VSS T_A= 25°C, VDD= 5V		600	1200	2000	kΩ
	R_{X2}	SXIN=VDD, SXOUT=VSS T_A= 25 °C ,VDD=5V		2500	5000	10000	

Table 7-9 DC Characteristics (Continued)

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V, f_{XIN}= 16MHz)

Parameter	Symbol	Condition		MIN	TYP	MAX	Unit
Supply Current	I_{DD1} (RUN)	f_{XIN}= 16MHz, VDD= 5V±10%		–	4.5	9.0	mA
		f_{XIN}= 12MHz, VDD= 3V±10%		–	3.0	6.0	
		f_{IRC}= 16MHz, VDD= 5V±10%		–	4.0	8.0	
	I_{DD2} (IDLE)	f_{XIN}= 16MHz, VDD= 5V±10%		–	2.0	4.0	mA
		f_{XIN}= 12MHz, VDD= 3V±10%		–	1.0	2.0	
		f_{IRC}= 16MHz, VDD= 5V±10%		–	1.2	2.4	
	I_{DD3}	f_{XIN}= 32.768kHz VDD= 3V±10% T_A= 25°C	Sub RUN	–	60.0	90.0	uA
	I_{DD4}		Sub IDLE	–	8.0	16.0	uA
	I_{DD5}	STOP, VDD= 5V±10%, T_A= 25°C		–	0.5	3.0	uA

NOTES) 1. Where the f_{XIN} is an external main oscillator, the f_{SUB} is an external sub oscillator, the f_{IRC} is an internal RC oscillator, and the fx is the selected system clock.

2. All supply current items don't include the current of an internal Watch-dog timer RC (WDTRC) oscillator and a peripheral block.

3. All supply current items include the current of the power-on reset (POR) block.

7.9 AC Characteristics

Table 7-9 AC Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
RESETB input low width	t_{RST}	Input, VDD= 5V	10	–	–	us
Interrupt input high, low width	t_{IWH}, t_{IWL}	All interrupt, VDD= 5V	200	–	–	ns
External Counter Input High, Low Pulse Width	t_{ECWH}, t_{ECWL}	ECn, VDD = 5 V (n= 0 - 9)	200	–	–	
External Counter Transition Time	t_{REC}, t_{FEC}	ECn, VDD = 5 V (n= 0 - 9)	20	–	–	

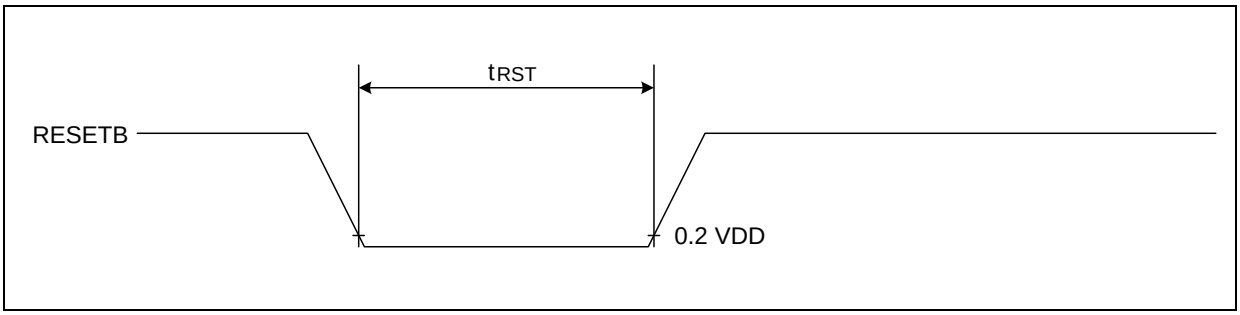


Figure 7.1 Input Timing for RESETB

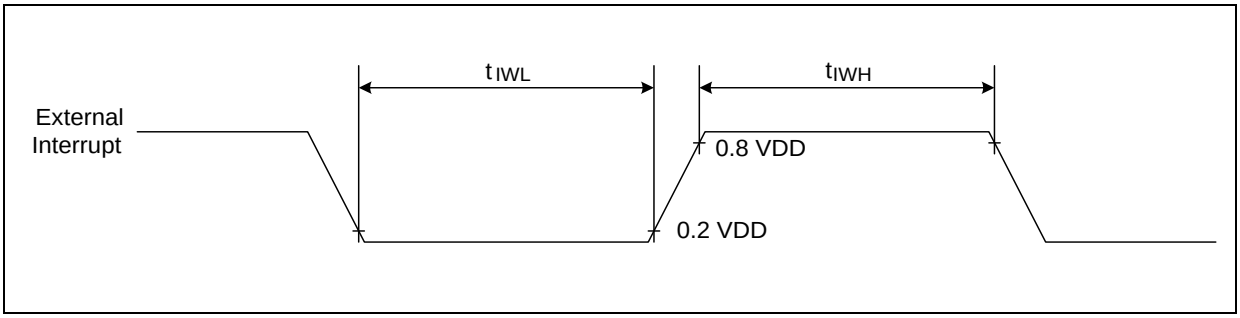


Figure 7.2 Input Timing for External Interrupts

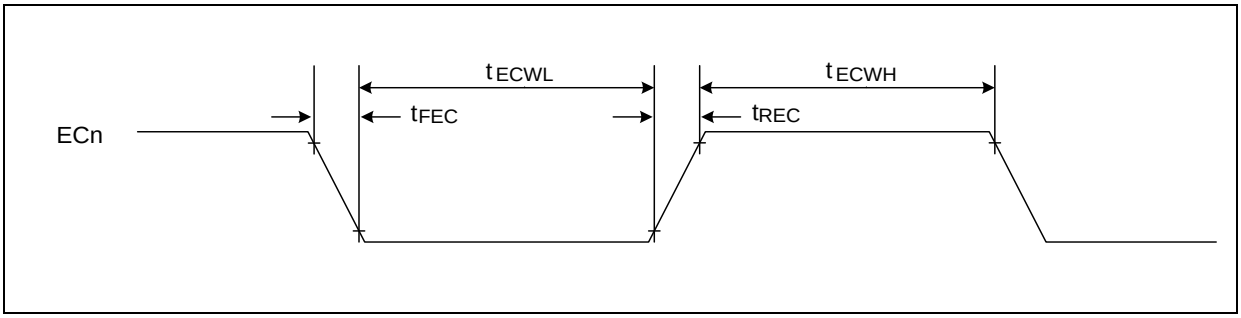


Figure 7.3 Input Timing for EC0 – EC9

7.10 SPI Characteristics

Table 7-10 SPI Characteristics

(T_A= -40°C – +85°C, VDD= 1.8V – 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Output Clock Pulse Period	t_{SCK}	Internal SCK source	200	–	–	ns
Input Clock Pulse Period		External SCK source	200	–	–	
Output Clock High, Low Pulse Width	t_{SCKH}, t_{SCKL}	Internal SCK source	70	–	–	
Input Clock High, Low Pulse Width		External SCK source	70	–	–	
First Output Clock Delay Time	t_{FOD}	Internal/External SCK source	100	–	–	
Output Clock Delay Time	t_{DS}	–	–	–	50	
Input Setup Time	t_{DIS}	–	100	–	–	
Input Hold Time	t_{DIH}	–	150	–	–	

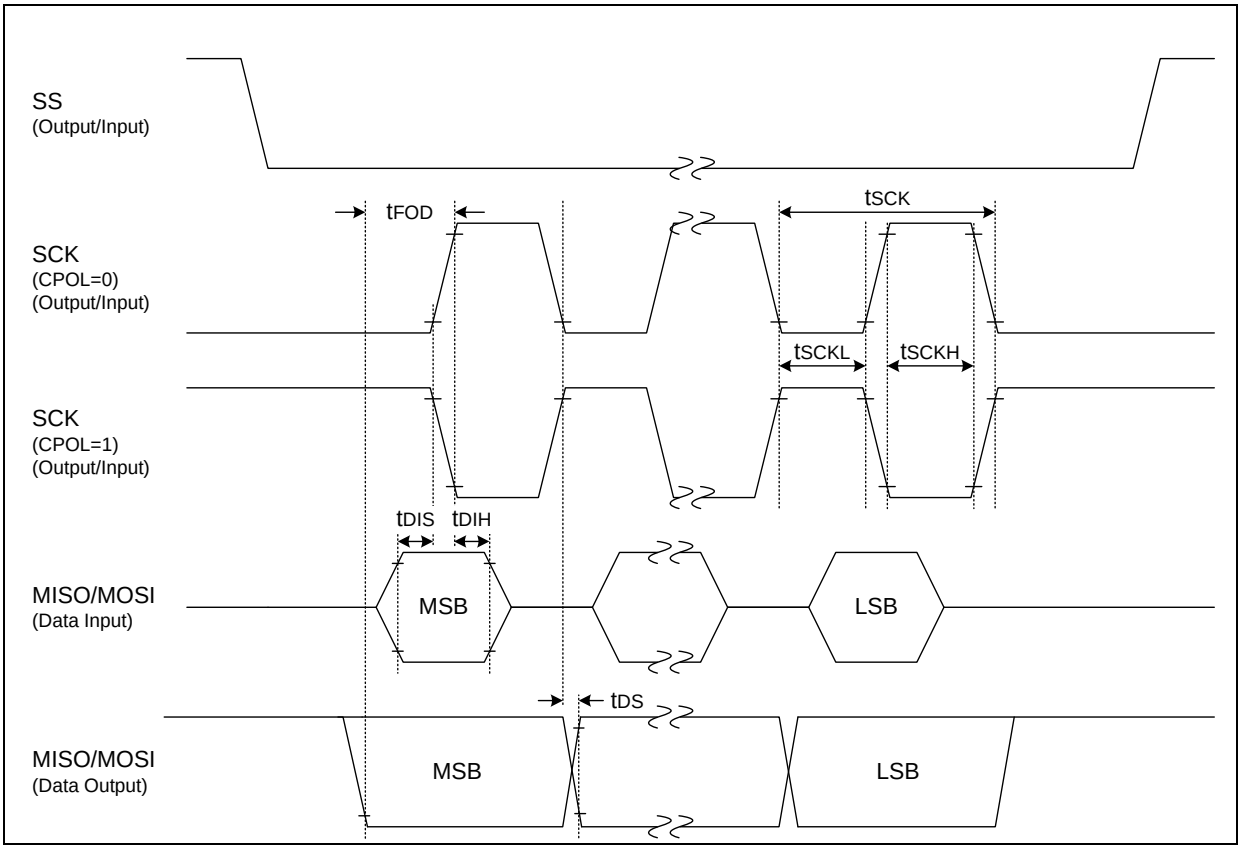


Figure 7.4 SPI Timing

7.11 UART Characteristics

Table 7-11 UART Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, f_{XIN}=11.1MHz)

Parameter	Symbol	MIN	TYP	MAX	Unit
Serial port clock cycle time	t_{SCK}	1250	t_{CPU} x 16	1650	ns
Output data setup to clock rising edge	t_{S1}	590	t_{CPU} x 13	–	ns
Clock rising edge to input data valid	t_{S2}	–	–	590	ns
Output data hold after clock rising edge	t_{H1}	t_{CPU} - 50	t_{CPU}	–	ns
Input data hold after clock rising edge	t_{H2}	0	–	–	ns
Serial port clock High, Low level width	t_{HIGH}, t_{LOW}	470	t_{CPU} x 8	970	ns

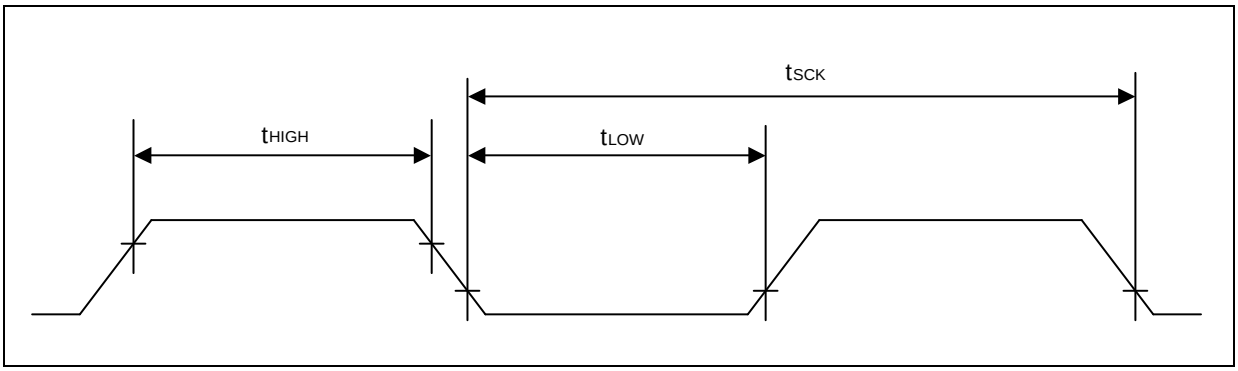


Figure 7.5 Waveform for UART Timing Characteristics

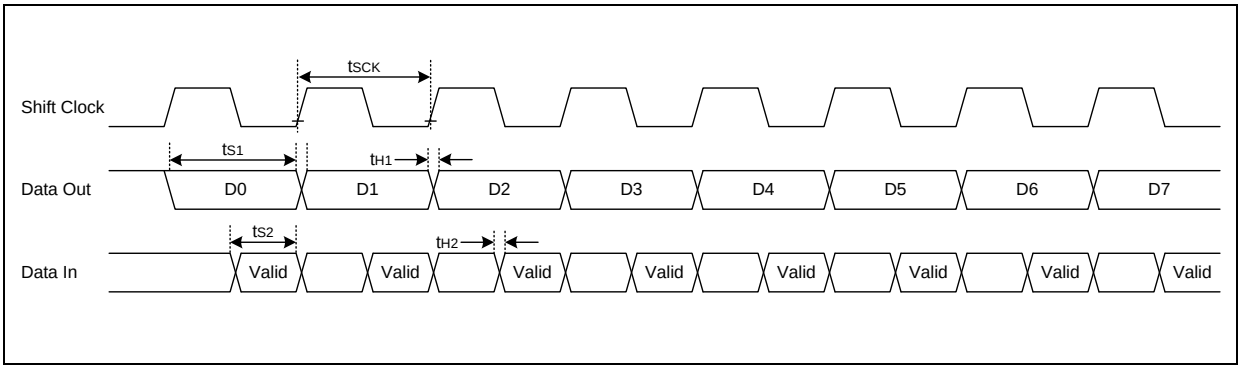


Figure 7.6 Timing Waveform for the UART Module

7.12 I2C Characteristics

Table 7-12 I2C Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Parameter	Symbol	Standard Mode		High-Speed Mode		Unit
		MIN	MAX	MIN	MAX	
Clock frequency	t_{SCL}	0	100	0	400	kHz
Clock High Pulse Width	t_{SCLH}	4.0	–	0.6	–	us
Clock Low Pulse Width	t_{SCLL}	4.7	–	1.3	–	
Bus Free Time	t_{BF}	4.7	–	1.3	–	
Start Condition Setup Time	t_{STSU}	4.7	–	0.6	–	
Start Condition Hold Time	t_{STHD}	4.0	–	0.6	–	
Stop Condition Setup Time	t_{SPSU}	4.0	–	0.6	–	
Stop Condition Hold Time	t_{SPHD}	4.0	–	0.6	–	
Output Valid from Clock	t_{VD}	0	–	0	–	
Data Input Hold Time	t_{DIH}	0	–	0	1.0	
Data Input Setup Time	t_{DIS}	250	–	100	–	ns

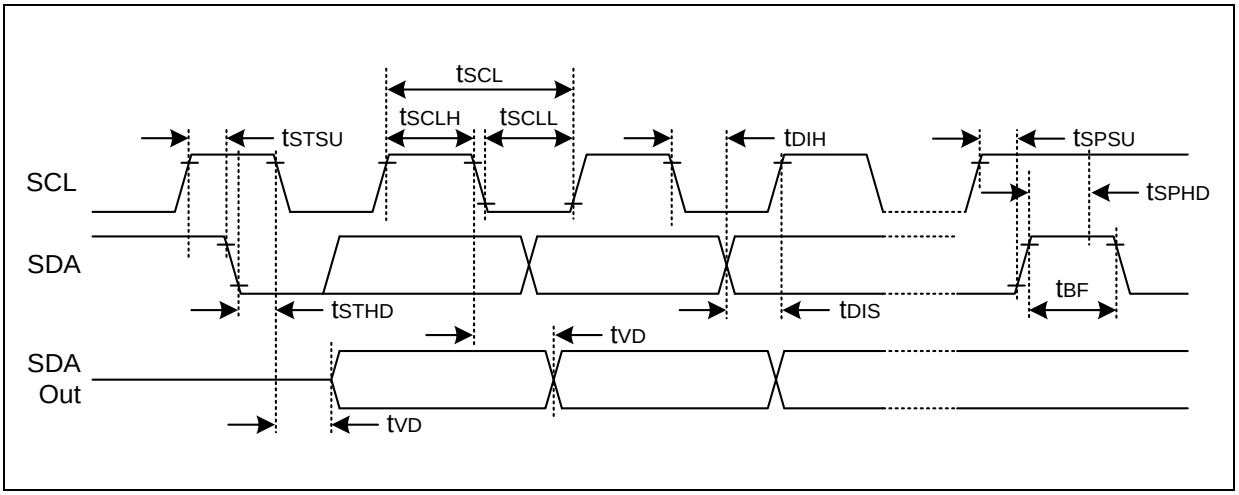


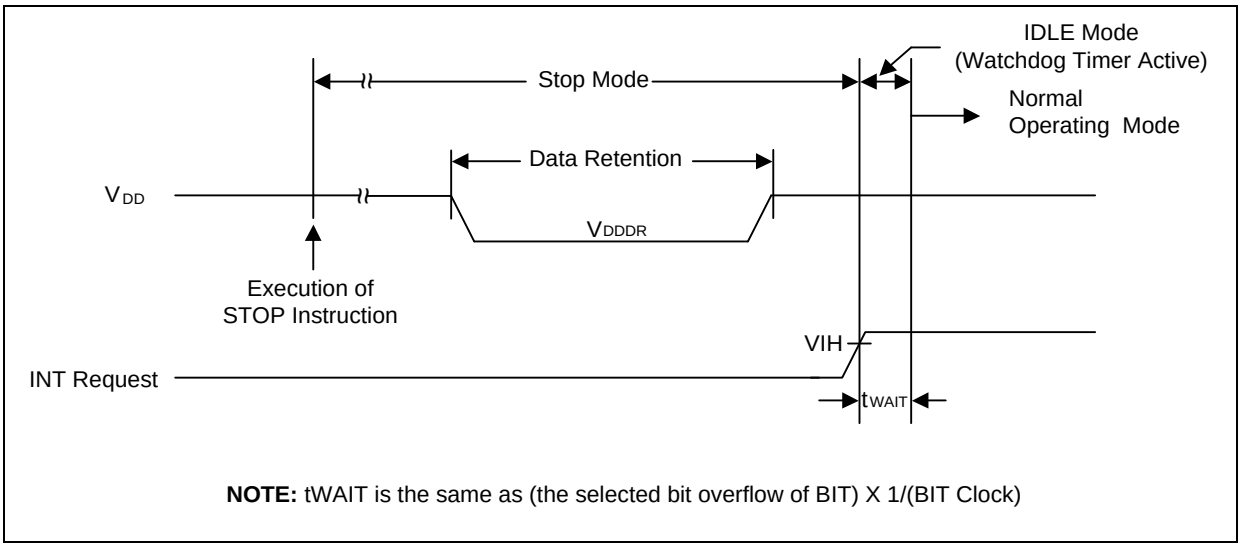
Figure 7.7 I2C Timing

7.13 Data Retention Voltage in Stop Mode

Table 7-13 Data Retention Voltage in Stop Mode

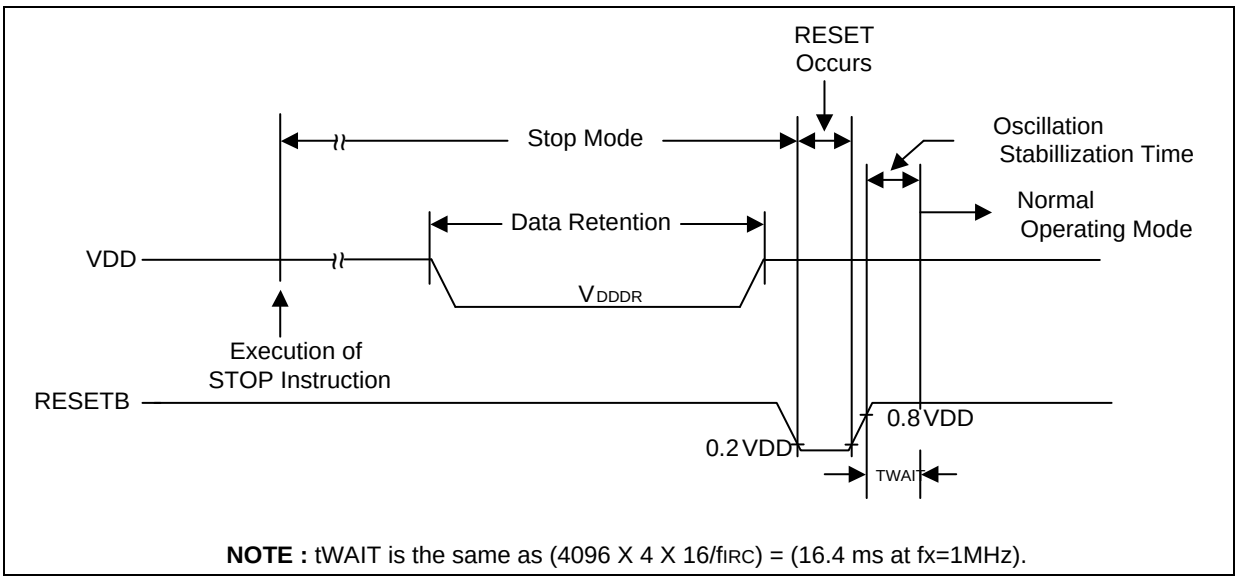
(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Parameter	Symbol	Conditions	MIN	TYP	MAX	Unit
Data retention supply voltage	V_{DDDR}	–	1.8	–	5.5	V
Data retention supply current	I_{DDDR}	VDDR= 1.8V, (T_A= 25°C), Stop mode	–	–	1	uA



NOTE: tWAIT is the same as (the selected bit overflow of BIT) X 1/(BIT Clock)

Figure 7.8 Stop Mode Release Timing when Initiated by an Interrupt



NOTE : tWAIT is the same as (4096 X 4 X 16/f_{IRC}) = (16.4 ms at fx=1MHz).

Figure 7.9 Stop Mode Release Timing when Initiated by RESETB

7.14 Internal Flash Rom Characteristics

Table 7-14 Internal Flash Rom Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V, VSS= 0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Sector Write Time	t_{FSW}	–	–	2.5	2.7	ms
Sector Erase Time	t_{FSE}	–	–	2.5	2.7	
Code Write Protection Time	t_{FHL}	–	–	2.5	2.7	
Page Buffer Reset Time	t_{FBR}	–	–	–	5	us
Flash Programming Frequency	f_{PGM}	–	0.4	–	–	MHz
Endurance of Write/Erase (Sector 0~1019)	NF_{WE}	–	–	–	10,000	Times
Endurance of Write/Erase (Sector 1020~1023)					100,000	

NOTE) During a flash operation, SCLK[1:0] of SCCR must be set to “00” or “01” (INT-RC OSC or Main X-TAL for system clock).

7.15 Input/Output Capacitance

Table 7-15 Input/Output Capacitance

(T_A= -40°C ~ +85°C, VDD= 0V)

Parameter	Symbol	Condition	MIN	TYP	MAX	Unit
Input Capacitance	C_{IN}	fx= 1MHz Unmeasured pins are connected to VSS	–	–	10	pF
Output Capacitance	C_{OUT}					
I/O Capacitance	C_{IO}					

7.16 Main Clock Oscillator Characteristics

Table 7-16 Main Clock Oscillator Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
Crystal	Main oscillation frequency	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	12.0	
		3.0V – 5.5V	0.4	–	16.0	
Ceramic Oscillator	Main oscillation frequency	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	12.0	
		3.0V – 5.5V	0.4	–	16.0	
External Clock	XIN input frequency	1.8V – 5.5V	0.4	–	4.2	MHz
		2.7V – 5.5V	0.4	–	12.0	
		3.0 V –5.5V	0.4	–	16.0	

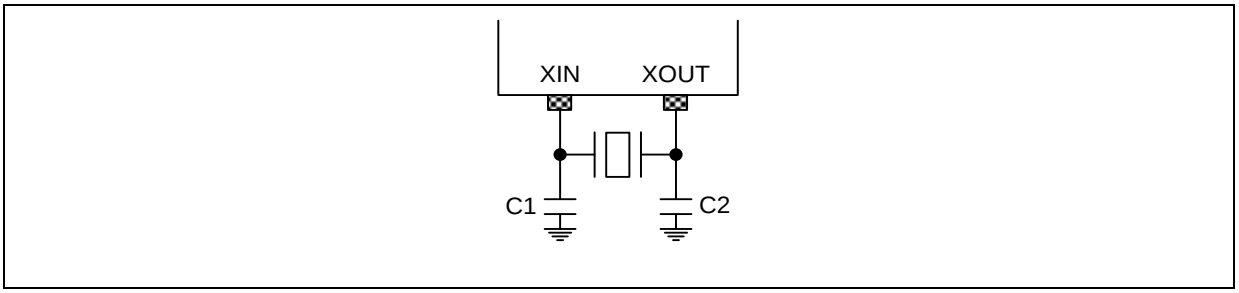


Figure 7.10 Crystal/Ceramic Oscillator

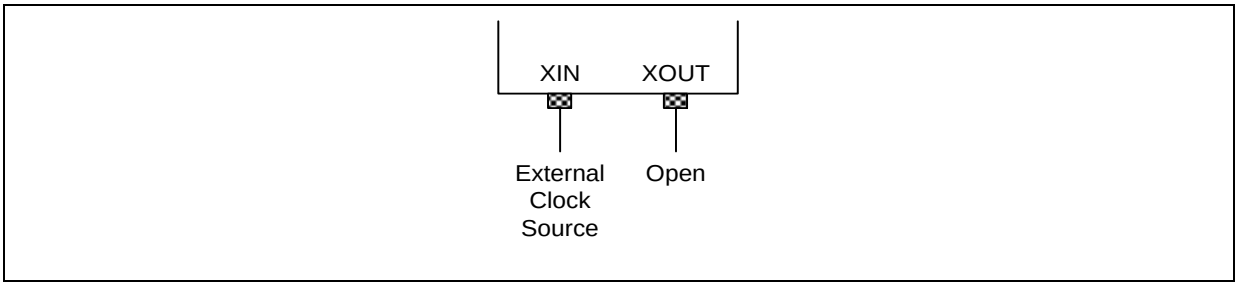


Figure 7.11 External Clock

7.17 Sub Clock Oscillator Characteristics

Table 7-17 Sub Clock Oscillator Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Oscillator	Parameter	Condition	MIN	TYP	MAX	Unit
Crystal	Sub oscillation frequency	1.8V – 5.5V	32	32.768	38	kHz
External Clock	SXIN input frequency		32	–	100	kHz

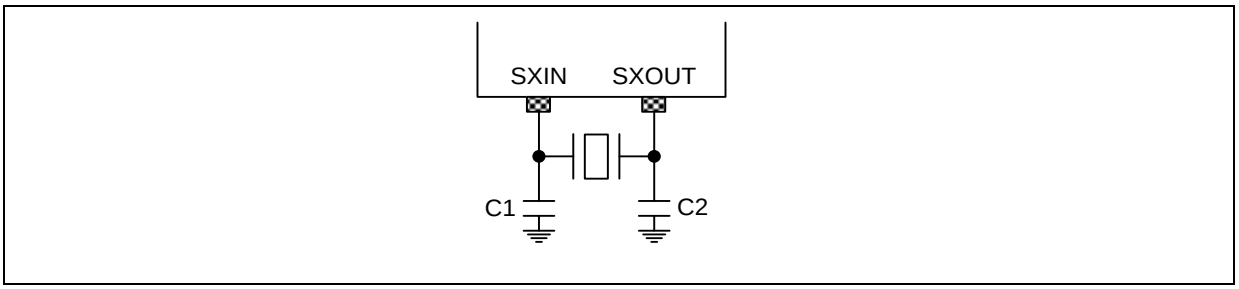


Figure 7.12 Crystal Oscillator

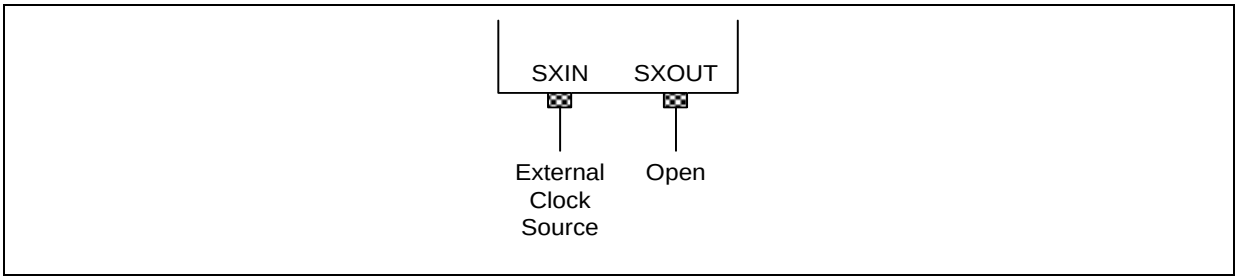


Figure 7.13 External Clock

7.18 Main Oscillation Stabilization Characteristics

Table 7-18 Main Oscillation Stabilization Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	fx > 1MHz Oscillation stabilization occurs when VDD is equal to the minimum oscillator voltage range.	–	–	60	ms
Ceramic		–	–	10	ms
External Clock	f_{XIN} = 0.4 to 16MHz XIN input high and low width (t_{XH}, t_{XL})	31	–	1250	ns

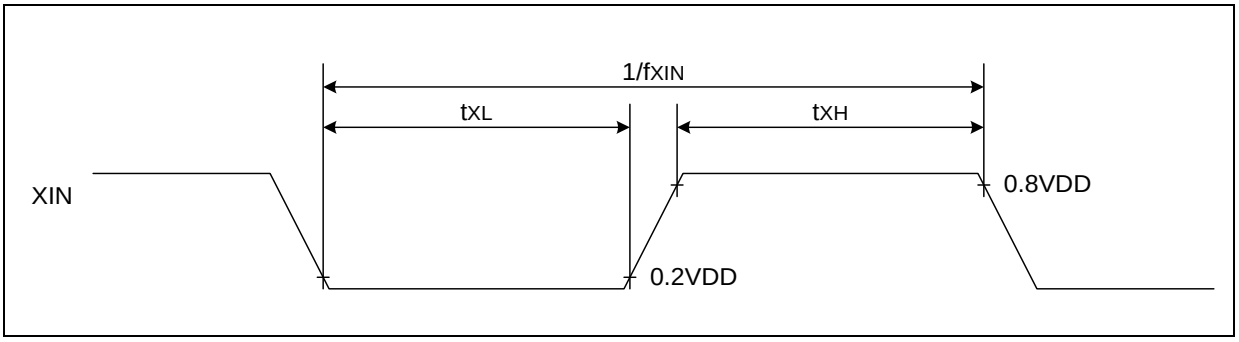


Figure 7.14 Clock Timing Measurement at XIN

7.19 Sub Oscillation Characteristics

Table 7-19 Sub Oscillation Stabilization Characteristics

(T_A= -40°C ~ +85°C, VDD= 1.8V ~ 5.5V)

Oscillator	Parameter	MIN	TYP	MAX	Unit
Crystal	–	–	–	10	s
External Clock	SXIN input high and low width (t_{XH}, t_{XL})	5	–	15	us

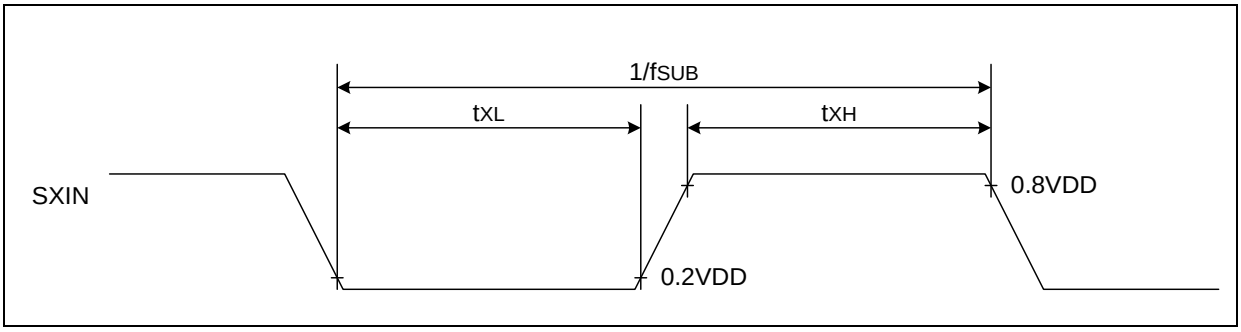


Figure 7.15 Clock Timing Measurement at SXIN

7.20 Operating Voltage Range

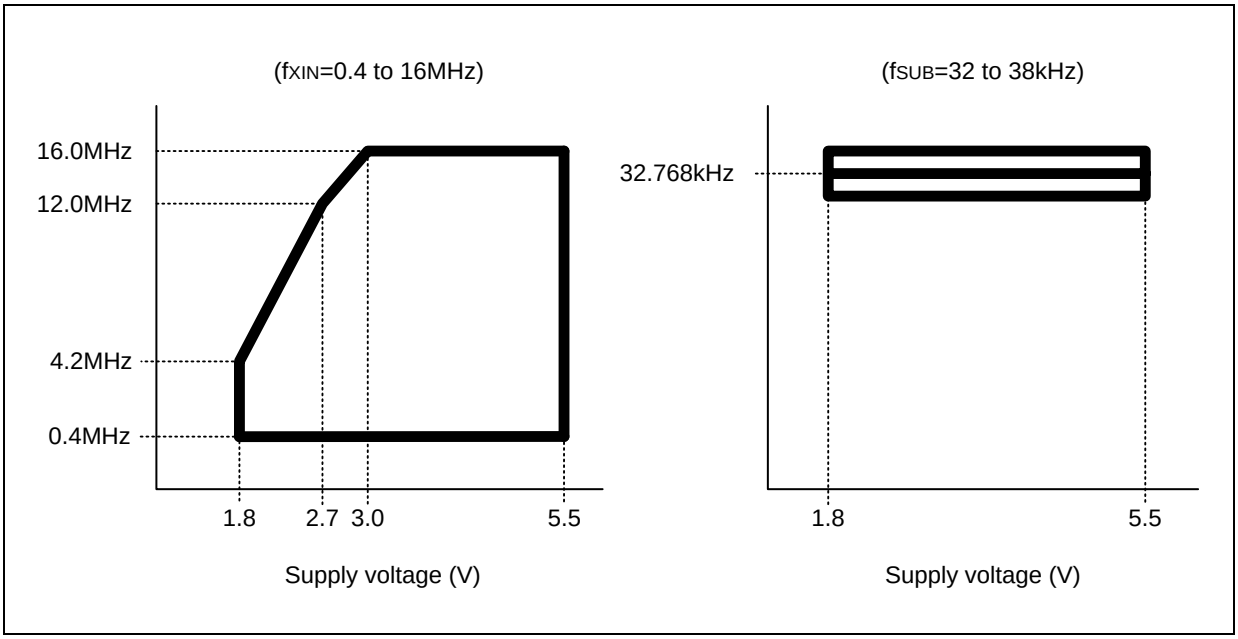


Figure 7.16 Operating Voltage Range

7.21 Recommended Circuit and Layout

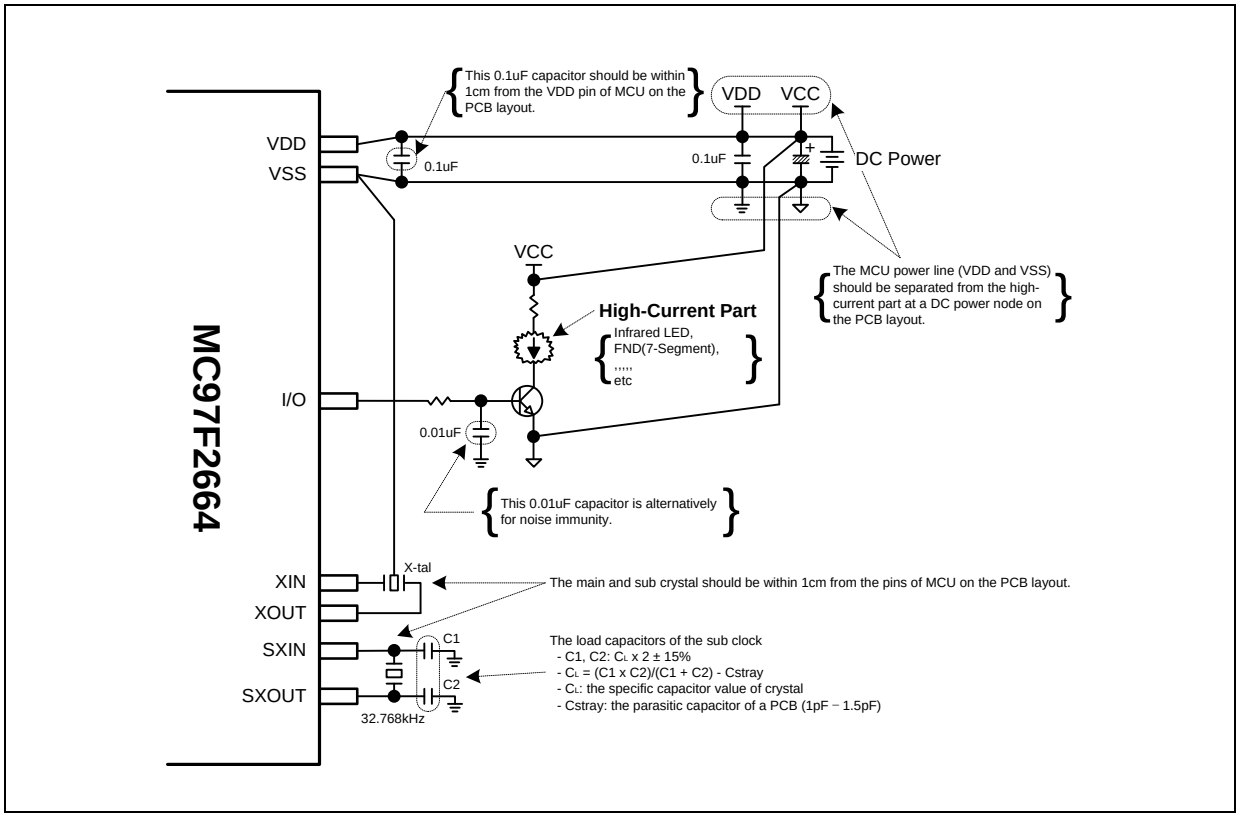
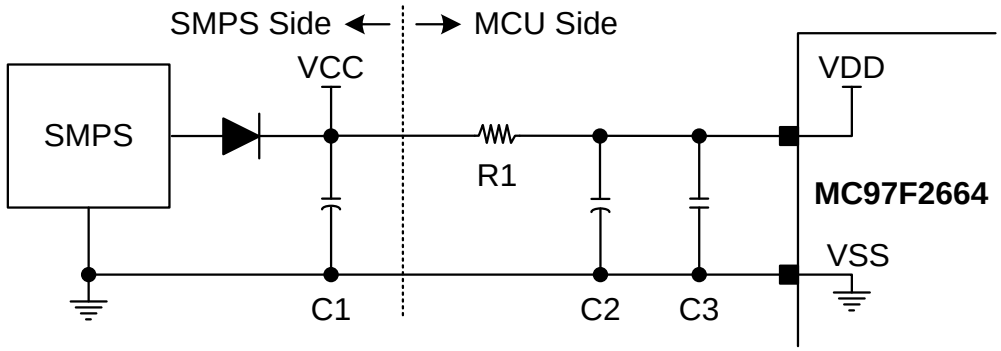


Figure 7.17 Recommended Circuit and Layout

7.22 Recommended Circuit and Layout with SMPS Power



1. The C1 capacitor is to flatten out the voltage of the SMPS power, VCC.
   √ Recommended C1: 470uF/25V more.
2. The R1 and C2 are the RC filter for VDD and suppress the ripple of VCC.
   √ Recommended R1: 10Ω - 20Ω
   √ Recommended C2: 47uF/25V more
   √ The R1 and C2 should be as close by the C3 as possible.
3. The C3 capacitor is used for temperature compensation because an electrolytic capacitor becomes worse characteristics at low temperature.
   √ Recommended C3: ceramic capacitor 2.2uF more
   √ The C3 should be within 1cm from VDD pin of MCU on the PCB layout.
4. The above circuit is recommended to improve noise immunity (EFT, Surge, ESD, etc) when the SMPS supplies the VDD of MCU.

Figure 7.18 Recommended Circuit and Layout with SMPS Power

7.23 Typical Characteristics

These graphs and tables provided in this section are only for design guidance and are not tested or guaranteed. In graphs or tables some data are out of specified operating range (e.g. out of specified VDD range). This is only for information and devices are guaranteed to operate properly only within the specified range.

The data presented in this section is a statistical summary of data collected on units from different lots over a period of time. “Typical” represents the mean of the distribution while “max” or “min” represents (mean + 3σ) and (mean - 3σ) respectively where σ is standard deviation.

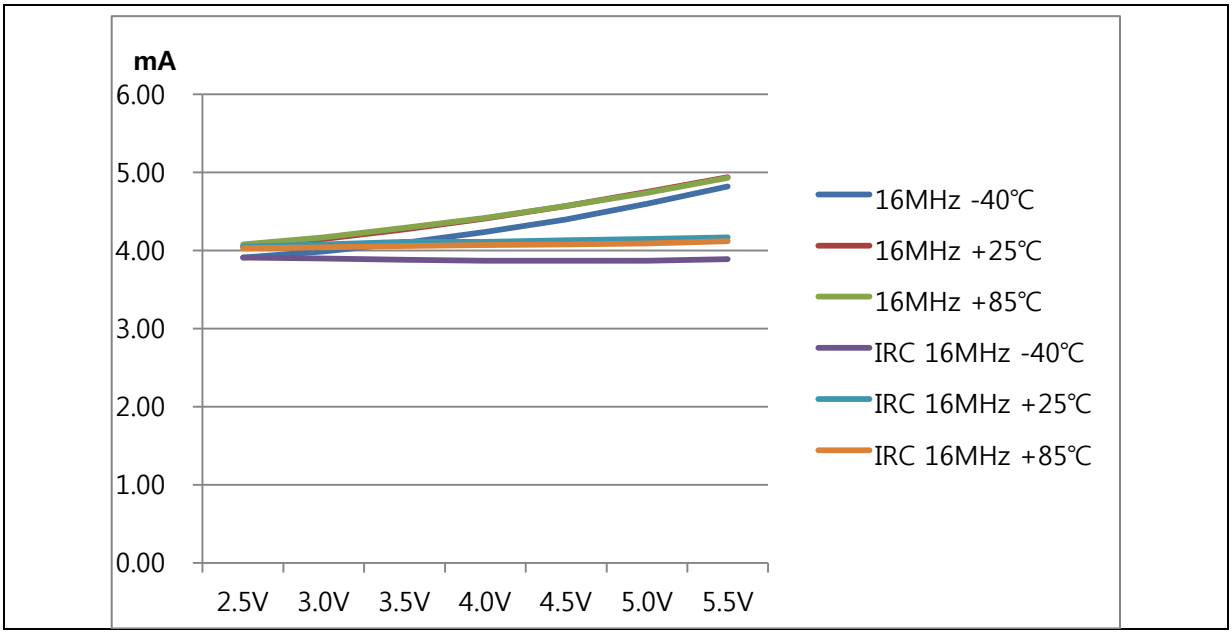


Figure 7.19 RUN (IDD1) Current

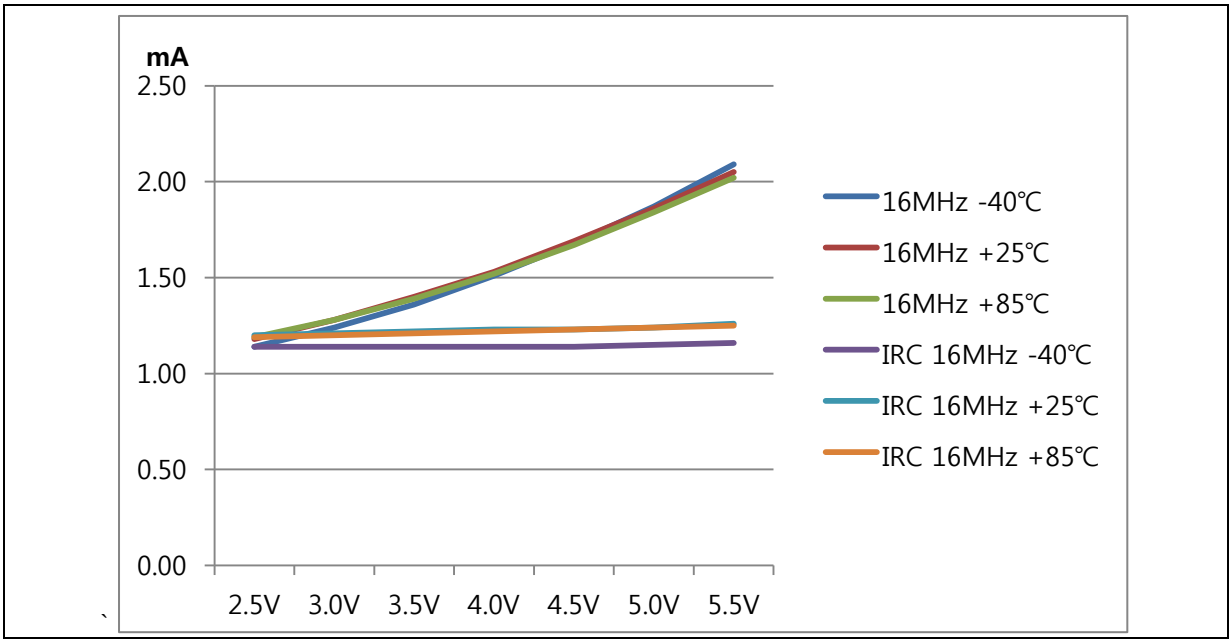


Figure 7.20 IDLE (IDD2) Current

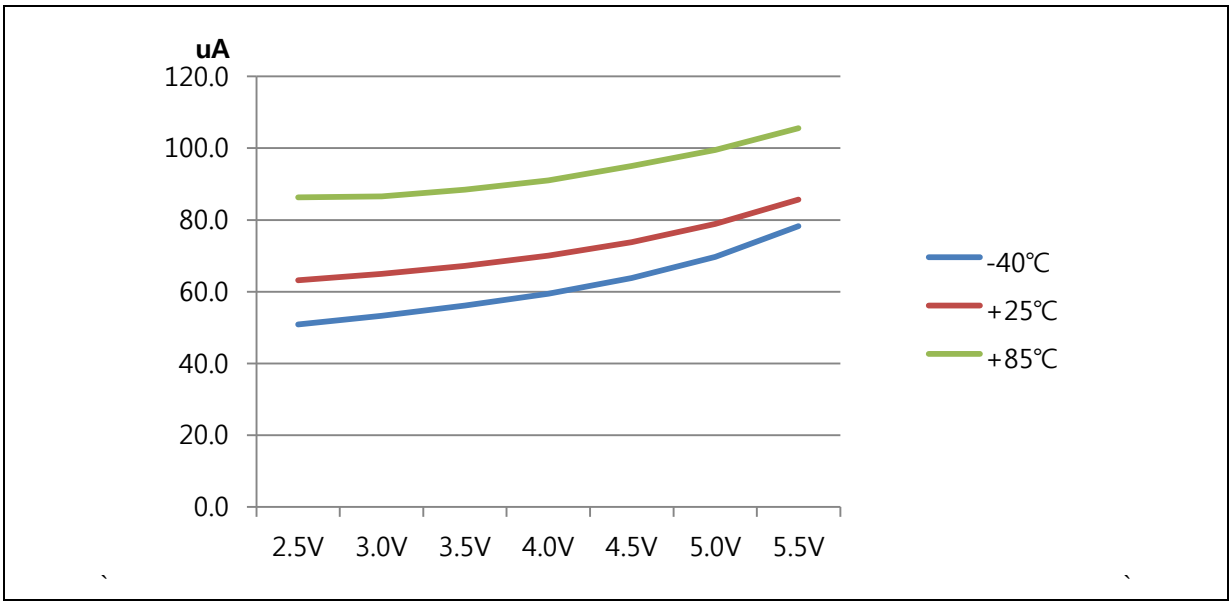


Figure 7.21 SUB RUN (IDD3) Current

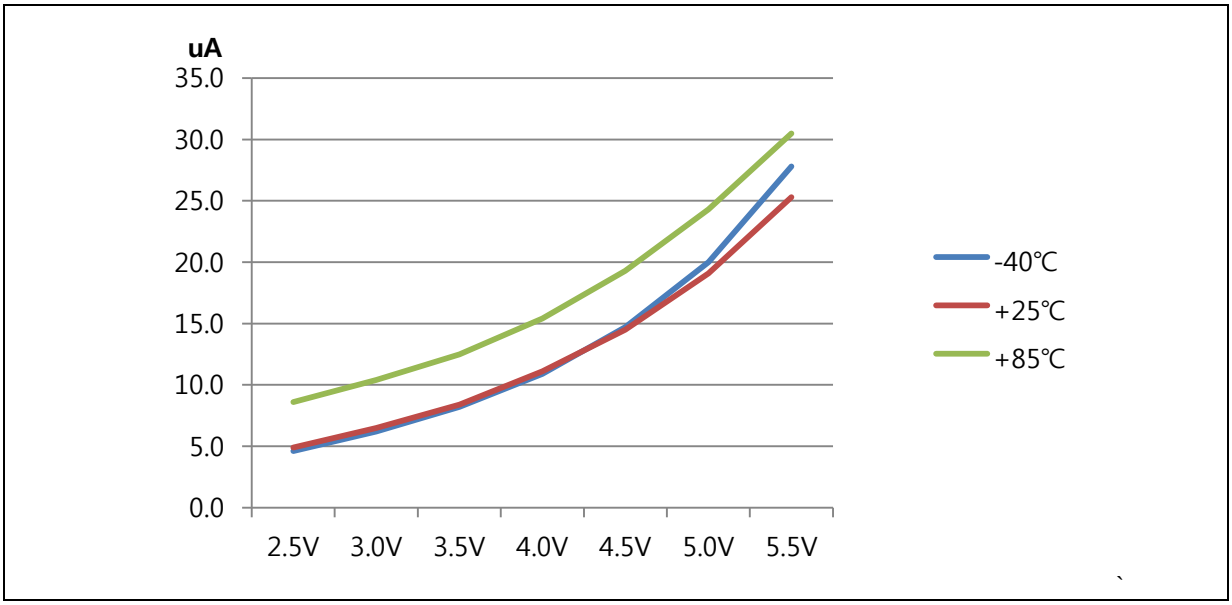


Figure 7.22 SUB IDLE (IDD4) Current

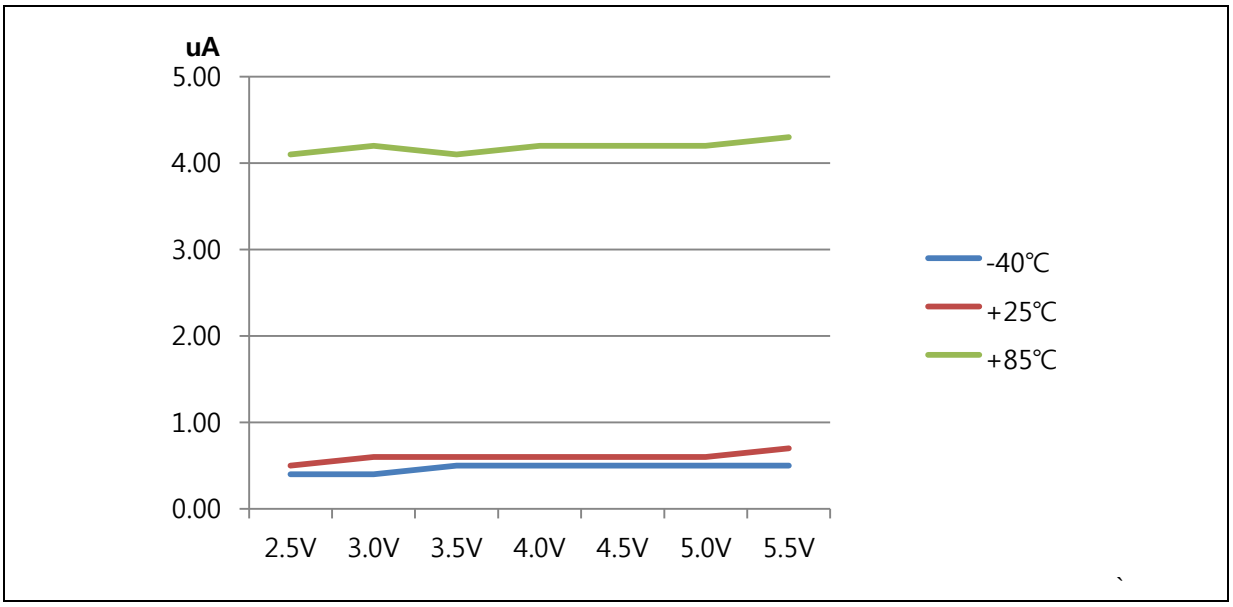


Figure 7.23 STOP (IDD5) Current

8. Memory

The MC97F2664 addresses two separate address memory stores: Program memory and Data memory. The logical separation of Program and Data memory allows Data memory to be accessed by 8-bit addresses, which makes the 8-bit CPU access the data memory more rapidly. Nevertheless, 16-bit Data memory addresses can also be generated through the DPTR register.

MC97F2664 provides on-chip 64k bytes of the ISP type flash program memory, which can be read and written to. Internal data memory (IRAM) is 256 bytes and it includes the stack area. External data memory (XRAM) is 4,096 bytes and it can be used as the extended stack area

8.1 Program Memory

A 16-bit program counter is capable of addressing up to 64k bytes, this device has just 64k bytes program memory space.

Figure 8-1 shows the map of the lower part of the program memory. After reset, the CPU begins execution from location 0000H. Each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External interrupt 0-7, for example, is assigned to location 002BH. If external interrupt 0-7 is going to be used, its service routine must begin at location 002BH. If the interrupt is not going to be used, its service location is available as general purpose program memory. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8 byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

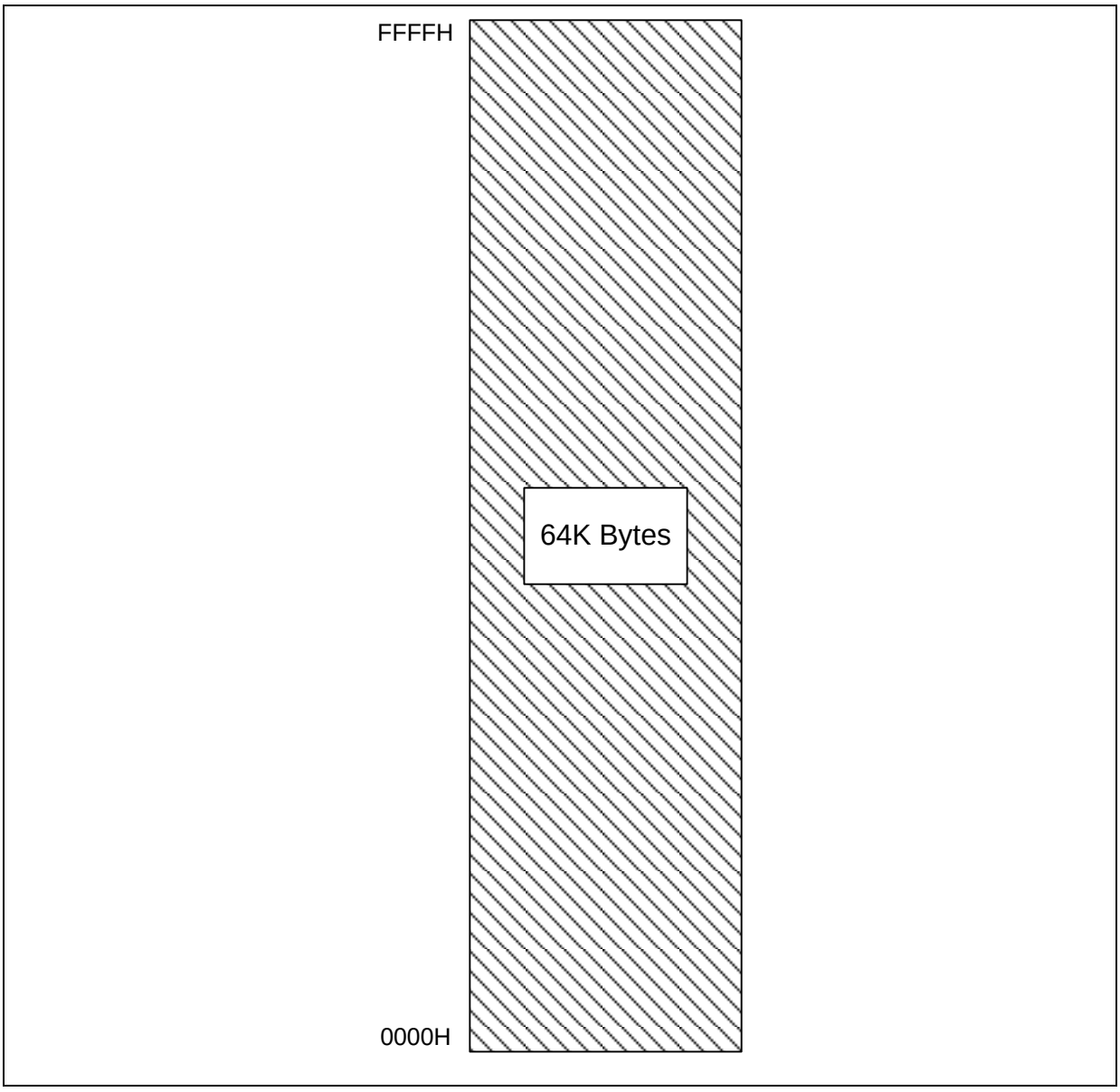


Figure 8.1 Program Memory

- 64k Bytes Including Interrupt Vector Region

8.2 Data Memory

Figure 8-2 shows the internal data memory space available.

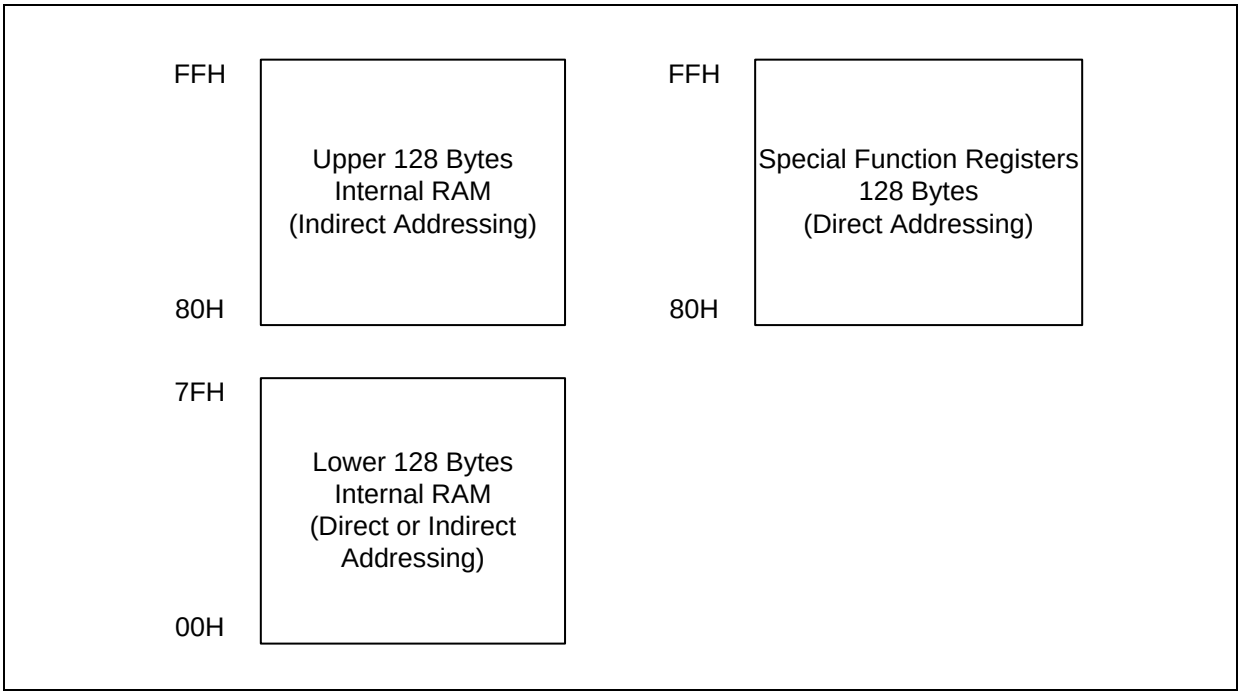


Figure 8.2 Data Memory Map

The internal data memory space is divided into three blocks, which are generally referred to as the lower 128 bytes, upper 128 bytes, and SFR space.

Internal data memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, in fact the addressing modes for internal RAM can accommodate up to 384 bytes by using a simple trick. Direct addresses higher than 7FH access one memory space and indirect addresses higher than 7FH access a different memory space. Thus Figure 8-2 shows the upper 128 bytes and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

The lower 128 bytes of RAM are present in all 8051 devices as mapped in Figure 8-3. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The 8051 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the lower 128 bytes can be accessed by either direct or indirect addressing. The upper 128 bytes RAM can only be accessed by indirect addressing. These spaces are used for data RAM and stack.

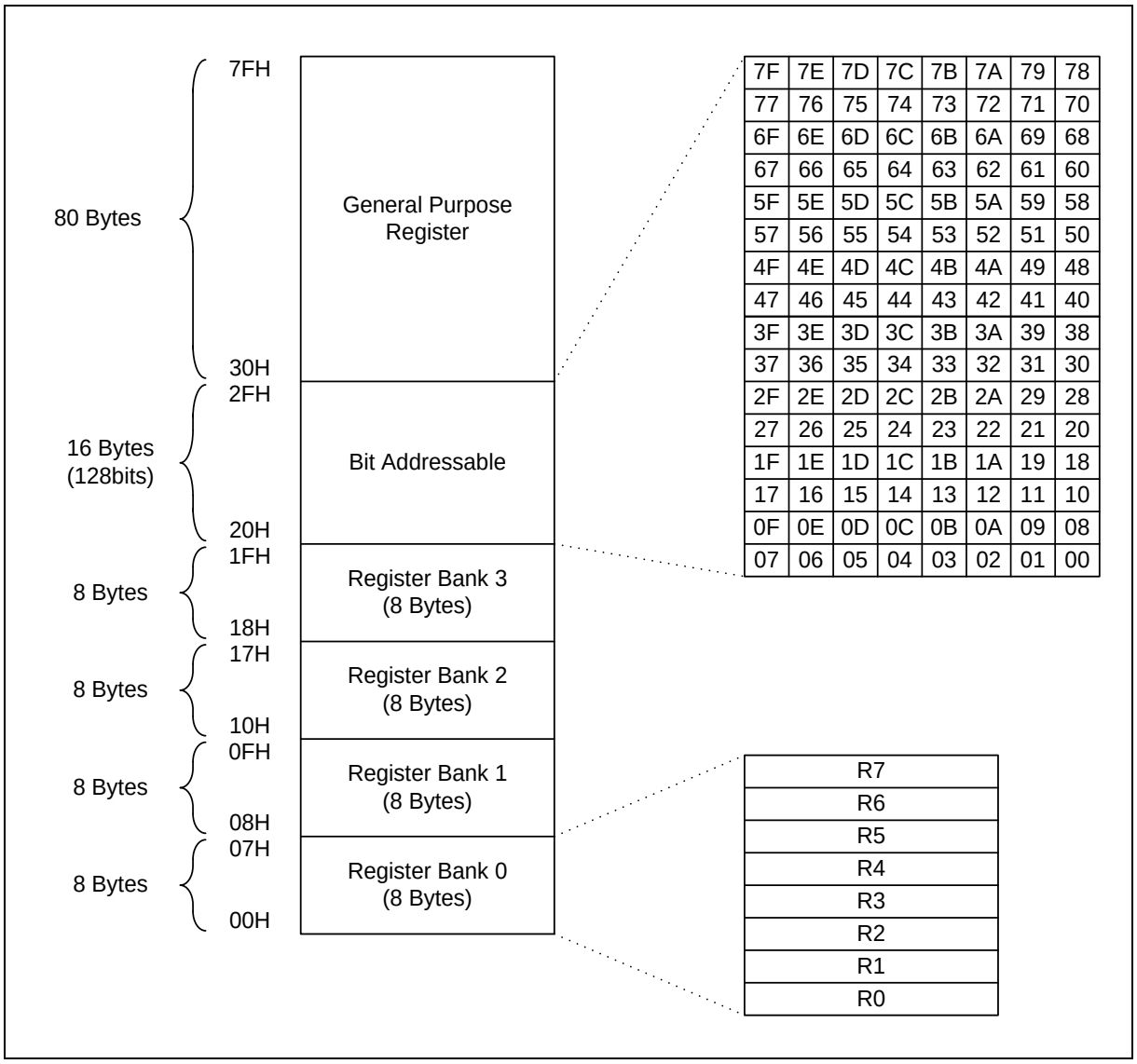


Figure 8.3 Lower 128 Bytes RAM

8.3 XRAM Memory

MC97F2664 has 4,096 bytes XRAM. This area has no relation with RAM/FLASH. It can be read and written to through SFR with 8-bit unit.

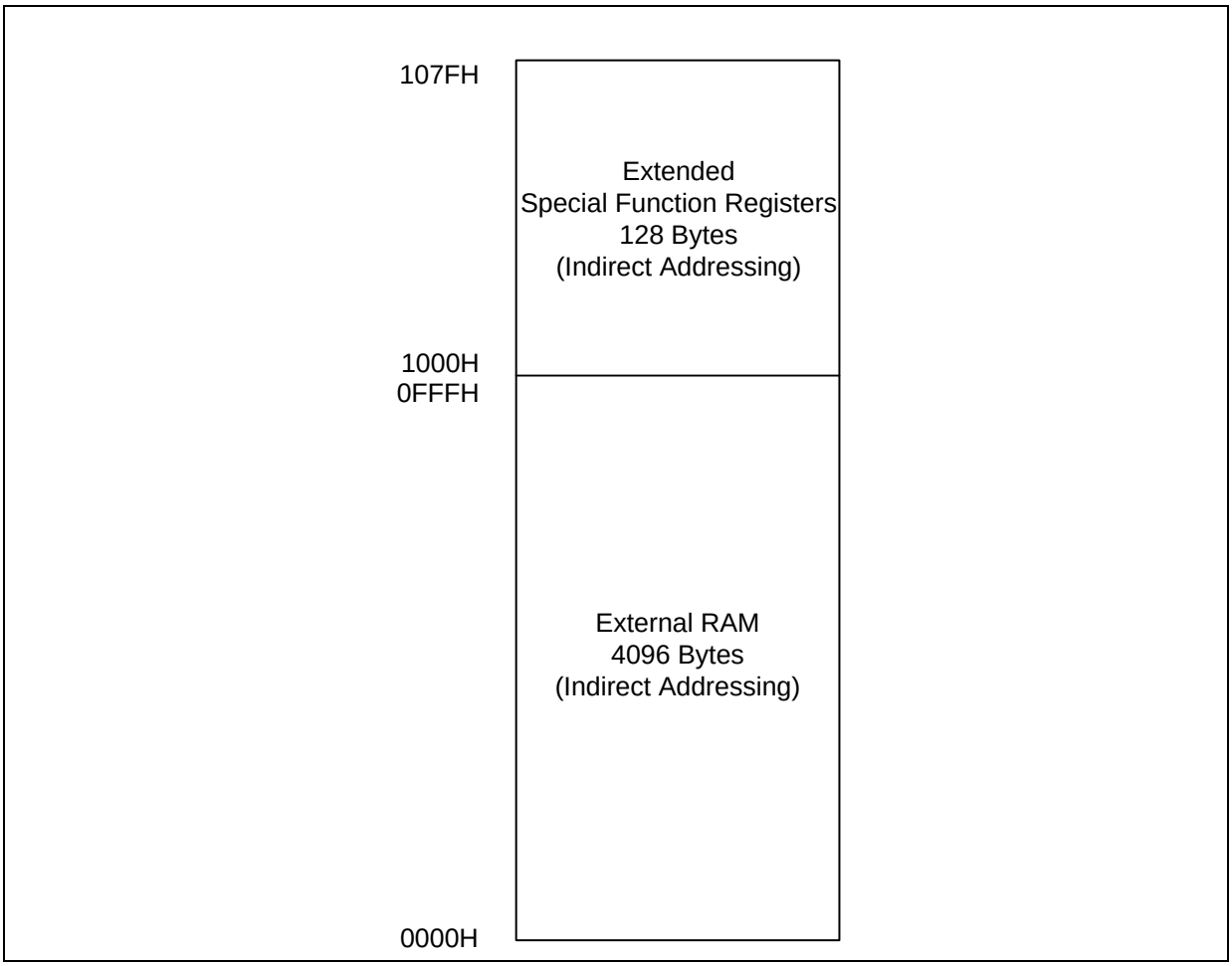


Figure 8.4 XDATA Memory Area

8.4 SFR Map

8.4.1 SFR Map Summary

Table 8-1 SFR Map Summary

– Reserved

M8051 compatible

	00H/8H(1)	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
0F8H	XBANK	-	FSADRH	FSADRM	FSADRL	FIDR	FMCR	MODINR
0F0H	B	P35DB	BUZCR	BUZDR	SPWRL	SPWRH	XSPCR	SINTCR
0E8H	RSTFR	P2DB	T8CRL	T8CRH	T8ADRL	T8ADRH	T8BDRL	T8BDRH
0E0H	ACC	P1DB	T7CRL	T7CRH	T7ADRL	T7ADRH	T7BDRL	T7BDRH
0D8H	LVRCR	P7IO	T6CRL	T6CRH	T6ADRL	T6ADRH	T6BDRL	T6BDRH
0D0H	PSW	P6IO	T5CRL	T5CRH	T5ADRL	T5ADRH	T5BDRL	T5BDRH
0C8H	OSCCR	P5IO	T4CRL	T4CRH	T4ADRL	T4ADRH	T4BDRL	T4BDRH
0C0H	P7	P4IO	SPI2CR	SPI2DR	SPI2SR	SPI3CR	SPI3DR	SPI3SR
0B8H	P6	P3IO	T2CR	T2CNT	T2DR/ T2CDR	T3CR	T3CNT	T3DR/ T3CDR
0B0H	P5	P2IO	T0CR	T0CNT	T0DR/ T0CDR	T1CR	T1CNT	T1DR/ T1CDR
0A8H	IE	IE1	IE2	IE3	EIPOL1L	EIPOL1H	EIPOL2L	EIPOL2H
0A0H	P4	P1IO	EO	EIFLAG0	EIFLAG1	EIFLAG2	EIPOL0L	EIPOL0H
98H	P3	P0IO	IP1L	IP1H	IP2L	IP2H	IP3L	IP3H
90H	P2	XSP	IP0L	IP0H		TINTCR	TIFLAG0	TIFLAG1
88H	P1	WTDR/ WTCNT	SCCR	BITCR	BITCNT	WDTCR	WDTDR/ WDTCNT	WTCR
80H	P0	SP	DPL	DPH	DPL1	DPH1	LVICR	PCON

NOTE) 1.These registers are bit-addressable

Table 8-2 Extended SFR Map Summary

-	Reserved

	00H/8H[1]	01H/9H	02H/0AH	03H/0BH	04H/0CH	05H/0DH	06H/0EH	07H/0FH
1078H								
1070H								
1068H								
1060H								
1058H	T9CRL	T9CRH	T9ADRL	T9ADRH	T9BDRL	T9BDRH		
1050H	ADCCRL	ADCCRH	ADCDRL	ADCDRH				
1048H	UART4CR1	UART4CR2	UART4CR3	UART4ST	UART4BD	UART4DR		
1040H	UART3CR1	UART3CR2	UART3CR3	UART3ST	UART3BD	UART3DR		
1038H	UART2CR1	UART2CR2	UART2CR3	UART2ST	UART2BD	UART2DR		
1030H	USI1ST1	USI1ST2	USI1BD	USI1SDHR	USI1DR	USI1SCLR	USI1SCHR	USI1SAR
1028H	USI1CR1	USI1CR2	USI1CR3	USI1CR4				
1020H	USI0ST1	USI0ST2	USI0BD	USI0SDHR	USI0DR	USI0SCLR	USI0SCHR	USI0SAR
1018H	USI0CR1	USI0CR2	USI0CR3	USI0CR4				
1010H	P0FSR	P1FSR	P2FSR	P3FSR	P4FSR	P5FSR	P6FSR	P7FSR
1008H	P0PU	P1PU	P2PU	P3PU	P4PU	P5PU	P6PU	P7PU
1000H	P0OD	P1OD	P2OD	P3OD	P4OD	P5OD	P6OD	P7OD

8.4.2 SFR Map

Table 8-3 SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
80H	P0 Data Register	P0	R/W	0	0	0	0	0	0	0	0
81H	Stack Pointer	SP	R/W	0	0	0	0	0	1	1	1
82H	Data Pointer Register Low	DPL	R/W	0	0	0	0	0	0	0	0
83H	Data Pointer Register High	DPH	R/W	0	0	0	0	0	0	0	0
84H	Data Pointer Register Low 1	DPL1	R/W	0	0	0	0	0	0	0	0
85H	Data Pointer Register High 1	DPH1	R/W	0	0	0	0	0	0	0	0
86H	Low Voltage Indicator Control Register	LVICR	R/W	–	–	0	0	0	0	0	0
87H	Power Control Register	PCON	R/W	0	–	–	–	0	0	0	0
88H	P1 Data Register	P1	R/W	0	0	0	0	0	0	0	0
89H	Watch Timer Data Register	WTDR	W	0	1	1	1	1	1	1	1
	Watch Timer Counter Register	WTCNT	R	–	0	0	0	0	0	0	0
8AH	System and Clock Control Register	SCCR	R/W	–	–	–	–	–	–	0	0
8BH	BIT Control Register	BITCR	R/W	0	0	0	–	0	0	0	1
8CH	Basic Interval Timer Counter Register	BITCNT	R	0	0	0	0	0	0	0	0
8DH	Watch Dog Timer Control Register	WDTCR	R/W	0	0	0	–	–	–	0	0
8EH	Watch Dog Timer Data Register	WDTDR	W	1	1	1	1	1	1	1	1
	Watch Dog Timer Counter Register	WDTCNT	R	0	0	0	0	0	0	0	0
8FH	Watch Timer Control Register	WTCR	R/W	0	–	0	0	0	0	0	0
90H	P2 Data Register	P2	R/W	0	0	0	0	0	0	0	0
91H	Extended Stack Pointer	XSP	R/W	0	0	0	0	0	0	0	0
92H	Interrupt Priority Register 0 Low Register	IP0L	R/W	–	–	0	0	0	0	0	0
93H	Interrupt Priority Register 0 High Register	IP0H	R/W	–	–	0	0	0	0	0	0
94H	Reserved	–	–	–	–	–	–	–	–	–	–
95H	Timer Interrupt Control Register	TINTCR	R/W	0	0	0	0	0	0	0	0
96H	Timer Interrupt Flag 0 Register	TIFLAG0	R/W	0	0	0	0	0	0	0	0
97H	Timer Interrupt Flag 1 Register	TIFLAG1	R/W	–	–	–	–	0	0	0	0
98H	P3 Data Register	P3	R/W	0	0	0	0	0	0	0	0
99H	P0 Direction Register	P0IO	R/W	0	0	0	0	0	0	0	0
9AH	Interrupt Priority Register 1 Low Register	IP1L	R/W	–	–	0	0	0	0	0	0
9BH	Interrupt Priority Register 1 High Register	IP1H	R/W	–	–	0	0	0	0	0	0
9CH	Interrupt Priority Register 2 Low Register	IP2L	R/W	–	–	0	0	0	0	0	0
9DH	Interrupt Priority Register 2 High Register	IP2H	R/W	–	–	0	0	0	0	0	0
9EH	Interrupt Priority Register 3 Low Register	IP3L	R/W	–	–	0	0	0	0	0	0
9FH	Interrupt Priority Register 3 High Register	IP3H	R/W	–	–	0	0	0	0	0	0

Table 8-3 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
A0H	P4 Data Register	P4	R/W	0	0	0	0	0	0	0	0
A1H	P1 Direction Register	P1IO	R/W	0	0	0	0	0	0	0	0
A2H	Extended Operation Register	EO	R/W	–	–	–	0	–	0	0	0
A3H	External Interrupt Flag 0 Register	EIFLAG0	R/W	0	0	0	0	0	0	0	0
A4H	External Interrupt Flag 1 Register	EIFLAG1	R/W	0	0	0	0	0	0	0	0
A5H	External Interrupt Flag 2 Register	EIFLAG2	R/W	–	–	–	0	0	0	0	0
A6H	External Interrupt Polarity 0 Low Register	EIPOL0L	R/W	0	0	0	0	0	0	0	0
A7H	External Interrupt Polarity 0 High Register	EIPOL0H	R/W	0	0	0	0	0	0	0	0
A8H	Interrupt Enable Register	IE	R/W	0	–	0	0	0	0	0	0
A9H	Interrupt Enable Register 1	IE1	R/W	–	–	0	0	0	0	–	0
AAH	Interrupt Enable Register 2	IE2	R/W	–	–	0	0	0	0	0	0
ABH	Interrupt Enable Register 3	IE3	R/W	–	–	0	0	0	0	0	0
ACH	External Interrupt Polarity 1 Low Register	EIPOL1L	R/W	0	0	0	0	0	0	0	0
ADH	External Interrupt Polarity 1 High Register	EIPOL1H	R/W	0	0	0	0	0	0	0	0
AEH	External Interrupt Polarity 2 Low Register	EIPOL2L	R/W	–	–	0	0	0	0	0	0
AFH	External Interrupt Polarity 2 High Register	EIPOL2H	R/W	–	–	–	–	0	0	0	0
B0H	P5 Data Register	P5	R/W	0	0	0	0	0	0	0	0
B1H	P2 Direction Register	P2IO	R/W	0	0	0	0	0	0	0	0
B2H	Timer 0 Control Register	T0CR	R/W	0	–	0	0	0	0	0	0
B3H	Timer 0 Counter Register	T0CNT	R	0	0	0	0	0	0	0	0
B4H	Timer 0 Data Register	T0DR	R/W	1	1	1	1	1	1	1	1
	Timer 0 Capture Data Register	T0CDR	R	0	0	0	0	0	0	0	0
B5H	Timer 1 Control Register	T1CR	R/W	0	–	0	0	0	0	0	0
B6H	Timer 1 Counter Register	T1CNT	R	0	0	0	0	0	0	0	0
B7H	Timer 1 Data Register	T1DR	R/W	1	1	1	1	1	1	1	1
	Timer 1 Capture Data Register	T1CDR	R	0	0	0	0	0	0	0	0
B8H	P6 Data Register	P6	R/W	0	0	0	0	0	0	0	0
B9H	P3 Direction Register	P3IO	R/W	0	0	0	0	0	0	0	0
BAH	Timer 2 Control Register	T2CR	R/W	0	–	0	0	0	0	0	0
BBH	Timer 2 Counter Register	T2CNT	R	0	0	0	0	0	0	0	0
BCH	Timer 2 Data Register	T2DR	R/W	1	1	1	1	1	1	1	1
	Timer 2 Capture Data Register	T2CDR	R	0	0	0	0	0	0	0	0

Table 8-3 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
BDH	Timer 3 Control Register	T3CR	R/W	0	–	0	0	0	0	0	0
BEH	Timer 3 Counter Register	T3CNT	R	0	0	0	0	0	0	0	0
BFH	Timer 3 Data Register	T3DR	R/W	1	1	1	1	1	1	1	1
	Timer 3 Capture Data Register	T3CDR	R	0	0	0	0	0	0	0	0
C0H	P7 Data Register	P7	R/W	–	–	–	0	0	0	0	0
C1H	P4 Direction Register	P4IO	R/W	0	0	0	0	0	0	0	0
C2H	SPI 2 Control Register	SPI2CR	R/W	0	0	0	0	0	0	0	0
C3H	SPI 2 Data Register	SPI2DR	R/W	0	0	0	0	0	0	0	0
C4H	SPI 2 Status Register	SPI2SR	R/W	0	0	0	–	0	0	–	–
C5H	SPI 3 Control Register	SPI3CR	R/W	0	0	0	0	0	0	0	0
C6H	SPI 3 Data Register	SPI3DR	R/W	0	0	0	0	0	0	0	0
C7H	SPI 3 Status Register	SPI3SR	R/W	0	0	0	–	0	0	–	–
C8H	Oscillator Control Register	OSCCR	R/W	–	–	0	0	1	0	0	0
C9H	P5 Direction Register	P5IO	R/W	0	0	0	0	0	0	0	0
CAH	Timer 4 Control Low Register	T4CRL	R/W	0	0	0	0	–	0	0	0
CBH	Timer 4 Control High Register	T4CRH	R/W	0	–	0	0	–	–	–	0
CCH	Timer 4 A Data Low Register	T4ADRL	R/W	1	1	1	1	1	1	1	1
CDH	Timer 4 A Data High Register	T4ADRH	R/W	1	1	1	1	1	1	1	1
CEH	Timer 4 B Data Low Register	T4BDRL	R/W	1	1	1	1	1	1	1	1
CFH	Timer 4 B Data High Register	T4BDRH	R/W	1	1	1	1	1	1	1	1
D0H	Program Status Word Register	PSW	R/W	0	0	0	0	0	0	0	0
D1H	P6 Direction Register	P6IO	R/W	0	0	0	0	0	0	0	0
D2H	Timer 5 Control Low Register	T5CRL	R/W	0	0	0	0	–	0	0	0
D3H	Timer 5 Control High Register	T5CRH	R/W	0	–	0	0	–	–	–	0
D4H	Timer 5 A Data Low Register	T5ADRL	R/W	1	1	1	1	1	1	1	1
D5H	Timer 5 A Data High Register	T5ADRH	R/W	1	1	1	1	1	1	1	1
D6H	Timer 5 B Data Low Register	T5BDRL	R/W	1	1	1	1	1	1	1	1
D7H	Timer 5 B Data High Register	T5BDRH	R/W	1	1	1	1	1	1	1	1
D8H	Low Voltage Reset Control Register	LVRCR	R/W	0	–	–	0	0	0	0	0
D9H	P7 Direction Register	P7IO	R/W	–	–	–	0	0	0	0	0
DAH	Timer 6 Control Low Register	T6CRL	R/W	0	0	0	–	–	0	0	0
DBH	Timer 6 Control High Register	T6CRH	R/W	0	0	0	0	–	–	–	0
DCH	Timer 6 A Data Low Register	T6ADRL	R/W	1	1	1	1	1	1	1	1
DDH	Timer 6 A Data High Register	T6ADRH	R/W	1	1	1	1	1	1	1	1
DEH	Timer 6 B Data Low Register	T6BDRL	R/W	1	1	1	1	1	1	1	1
DFH	Timer 6 B Data High Register	T6BDRH	R/W	1	1	1	1	1	1	1	1

Table 8-3 SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
E0H	Accumulator Register	ACC	R/W	0	0	0	0	0	0	0	0
E1H	P1 Debounce Enable Register	P1DB	R/W	0	0	0	0	0	0	0	0
E2H	Timer 7 Control Low Register	T7CRL	R/W	0	0	0	–	–	0	0	0
E3H	Timer 7 Control High Register	T7CRH	R/W	0	0	0	0	–	–	–	0
E4H	Timer 7 A Data Low Register	T7ADRL	R/W	1	1	1	1	1	1	1	1
E5H	Timer 7 A Data High Register	T7ADRH	R/W	1	1	1	1	1	1	1	1
E6H	Timer 7 B Data Low Register	T7BDRL	R/W	1	1	1	1	1	1	1	1
E7H	Timer 7 B Data High Register	T7BDRH	R/W	1	1	1	1	1	1	1	1
E8H	Reset Flag Register	RSTFR	R/W	1	x	0	0	x	–	–	–
E9H	P2 Debounce Enable Register	P2DB	R/W	0	0	0	0	0	0	0	0
EAH	Timer 8 Control Low Register	T8CRL	R/W	0	0	0	–	–	0	0	0
EBH	Timer 8 Control High Register	T8CRH	R/W	0	0	0	0	–	–	–	0
ECH	Timer 8 A Data Low Register	T8ADRL	R/W	1	1	1	1	1	1	1	1
EDH	Timer 8 A Data High Register	T8ADRH	R/W	1	1	1	1	1	1	1	1
EEH	Timer 8 B Data Low Register	T8BDRL	R/W	1	1	1	1	1	1	1	1
EFH	Timer 8 B Data High Register	T8BDRH	R/W	1	1	1	1	1	1	1	1
F0H	B Register	B	R/W	0	0	0	0	0	0	0	0
F1H	P35 Debounce Enable Register	P35DB	R/W	–	0	0	0	0	0	0	0
F2H	BUZZER Control Register	BUZCR	R/W	–	–	–	–	0	0	0	0
F3H	BUZZER Data Register	BUZDR	R/W	1	1	1	1	1	1	1	1
F4H	Stack Pointer Watch Low Register	SPWRL	R/W	1	1	1	1	1	1	1	1
F5H	Stack Pointer Watch High Register	SPWRH	R/W	0	0	0	0	1	1	1	1
F6H	Extended Stack Pointer Control Register	XSPCR	R/W	–	–	–	–	–	–	–	0
F7H	System Interrupt Control Register	SINTCR	R/W	–	–	–	0	–	–	–	0
F8H	External RAM Bank Selection Register	XBANK	R/W	0	0	0	0	0	0	0	0
F9H	Reserved	–	–	–	–	–	–	–	–	–	–
FAH	Flash Sector Address High Register	FSADRH	R/W	–	–	–	–	0	0	0	0
FBH	Flash Sector Address Middle Register	FSADRM	R/W	0	0	0	0	0	0	0	0
FCH	Flash Sector Address Low Register	FSADRL	R/W	0	0	0	0	0	0	0	0
FDH	Flash Identification Register	FIDR	R/W	0	0	0	0	0	0	0	0
FEH	Flash Mode Control Register	FMCR	R/W	0	–	–	–	–	0	0	0
FFH	Mode Entry Register	MODINR	R	–	–	–	–	–	–	0	0

Table 8-4 Extended SFR Map

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1000H	P0 Open-drain Selection Register	P0OD	R/W	0	0	0	0	0	0	0	0
1001H	P1 Open-drain Selection Register	P1OD	R/W	0	0	0	0	0	0	0	0
1002H	P2 Open-drain Selection Register	P2OD	R/W	0	0	0	0	0	0	0	0
1003H	P3 Open-drain Selection Register	P3OD	R/W	0	0	0	0	0	0	0	0
1004H	P4 Open-drain Selection Register	P4OD	R/W	0	0	0	0	0	0	0	0
1005H	P5 Open-drain Selection Register	P5OD	R/W	0	0	0	0	0	0	0	0
1006H	P6 Open-drain Selection Register	P6OD	R/W	0	0	0	0	0	0	0	0
1007H	P7 Open-drain Selection Register	P7OD	R/W	–	–	–	0	0	0	0	0
1008H	P0 Pull-up Resistor Selection Register	P0PU	R/W	0	0	0	0	0	0	0	0
1009H	P1 Pull-up Resistor Selection Register	P1PU	R/W	0	0	0	0	0	0	0	0
100AH	P2 Pull-up Resistor Selection Register	P2PU	R/W	0	0	0	0	0	0	0	0
100BH	P3 Pull-up Resistor Selection Register	P3PU	R/W	0	0	0	0	0	0	0	0
100CH	P4 Pull-up Resistor Selection Register	P4PU	R/W	0	0	0	0	0	0	0	0
100DH	P5 Pull-up Resistor Selection Register	P5PU	R/W	0	0	0	0	0	0	0	0
100EH	P6 Pull-up Resistor Selection Register	P6PU	R/W	0	0	0	0	0	0	0	0
100FH	P7 Pull-up Resistor Selection Register	P7PU	R/W	–	–	–	0	0	0	0	0
1010H	P0 Function Selection Register	P0FSR	R/W	0	0	0	0	0	0	0	0
1011H	P1 Function Selection Register	P1FSR	R/W	0	0	0	0	0	0	0	0
1012H	P2 Function Selection Register	P2FSR	R/W	–	–	–	–	0	0	0	0
1013H	P3 Function Selection Register	P3FSR	R/W	0	0	0	0	0	0	0	0
1014H	P4 Function Selection Register	P4FSR	R/W	–	–	–	0	0	0	0	0
1015H	P5 Function Selection Register	P5FSR	R/W	–	–	–	–	0	0	0	0
1016H	P6 Function Selection Register	P6FSR	R/W	–	0	0	0	0	0	0	0
1017H	P7 Function Selection Register	P7FSR	R/W	–	–	–	–	0	0	0	0
1018H	USI0 Control Register 1	USI0CR1	R/W	0	0	0	0	0	0	0	0
1019H	USI0 Control Register 2	USI0CR2	R/W	0	0	0	0	0	0	0	0
101AH	USI0 Control Register 3	USI0CR3	R/W	0	0	0	0	0	0	0	0
101BH	USI0 Control Register 4	USI0CR4	R/W	0	–	0	0	0	0	0	0
101CH	Reserved	–	–	–							
101DH	Reserved	–	–	–							
101EH	Reserved	–	–	–							
101FH	Reserved	–	–	–							

Table 8-4 Extended SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1020H	USI0 Status Register 1	USI0ST1	R/W	1	0	0	0	0	0	0	0
1021H	USI0 Status Register 2	USI0ST2	R/W	0	0	0	0	0	0	0	0
1022H	USI0 Baud Rate Generation Register	USI0BD	R/W	1	1	1	1	1	1	1	1
1023H	USI0 SDA Hold Time Register	USI0SDHR	R/W	0	0	0	0	0	0	0	1
1024H	USI0 Data Register	USI0DR	R/W	0	0	0	0	0	0	0	0
1025H	USI0 SCL Low Period Register	USI0SCLR	R/W	0	0	1	1	1	1	1	1
1026H	USI0 SCL High Period Register	USI0SCHR	R/W	0	0	1	1	1	1	1	1
1027H	USI0 Slave Address Register	USI0SAR	R/W	0	0	0	0	0	0	0	0
1028H	USI1 Control Register 1	USI1CR1	R/W	0	0	0	0	0	0	0	0
1029H	USI1 Control Register 2	USI1CR2	R/W	0	0	0	0	0	0	0	0
102AH	USI1 Control Register 3	USI1CR3	R/W	0	0	0	0	0	0	0	0
102BH	USI1 Control Register 4	USI1CR4	R/W	0	–	0	0	0	0	0	0
102CH	Reserved	–	–	–							
102DH	Reserved	–	–	–							
102EH	Reserved	–	–	–							
102FH	Reserved	–	–	–							
1030H	USI1 Status Register 1	USI1ST1	R/W	1	0	0	0	0	0	0	0
1031H	USI1 Status Register 2	USI1ST2	R/W	0	0	0	0	0	0	0	0
1032H	USI1 Baud Rate Generation Register	USI1BD	R/W	1	1	1	1	1	1	1	1
1033H	USI1 SDA Hold Time Register	USI1SDHR	R/W	0	0	0	0	0	0	0	1
1034H	USI1 Data Register	USI1DR	R/W	0	0	0	0	0	0	0	0
1035H	USI1 SCL Low Period Register	USI1SCLR	R/W	0	0	1	1	1	1	1	1
1036H	USI1 SCL High Period Register	USI1SCHR	R/W	0	0	1	1	1	1	1	1
1037H	USI1 Slave Address Register	USI1SAR	R/W	0	0	0	0	0	0	0	0
1038H	UART2 Control Register 1	UART2CR1	R/W	–	–	0	0	0	0	0	–
1039H	UART2 Control Register 2	UART2CR2	R/W	0	0	0	0	0	0	0	0
103AH	UART2 Control Register 3	UART2CR3	R/W	–	0	–	–	–	0	0	0
103BH	UART2 Status Register	UART2ST	R/W	1	0	0	0	0	0	0	0
103CH	UART2 Baud Rate Generation Register	UART2BD	R/W	1	1	1	1	1	1	1	1
103DH	UART2 Data Register	UART2DR	R/W	0	0	0	0	0	0	0	0
103EH	Reserved	–	–	–							
103FH	Reserved	–	–	–							

Table 8-4 Extended SFR Map (Continued)

Address	Function	Symbol	R/W	@Reset							
				7	6	5	4	3	2	1	0
1040H	UART3 Control Register 1	UART3CR1	R/W	–	–	0	0	0	0	0	–
1041H	UART3 Control Register 2	UART3CR2	R/W	0	0	0	0	0	0	0	0
1042H	UART3 Control Register 3	UART3CR3	R/W	–	0	–	–	–	0	0	0
1043H	UART3 Status Register	UART3ST	R/W	1	0	0	0	0	0	0	0
1044H	UART3 Baud Rate Generation Register	UART3BD	R/W	1	1	1	1	1	1	1	1
1045H	UART3 Data Register	UART3DR	R/W	0	0	0	0	0	0	0	0
1046H	Reserved	–	–	–							
1047H	Reserved	–	–	–							
1048H	UART4 Control Register 1	UART4CR1	R/W	–	–	0	0	0	0	0	–
1049H	UART4 Control Register 2	UART4CR2	R/W	0	0	0	0	0	0	0	0
104AH	UART4 Control Register 3	UART4CR3	R/W	–	0	–	–	–	0	0	0
104BH	UART4 Status Register	UART4ST	R/W	1	0	0	0	0	0	0	0
104CH	UART4 Baud Rate Generation Register	UART4BD	R/W	1	1	1	1	1	1	1	1
104DH	UART4 Data Register	UART4DR	R/W	0	0	0	0	0	0	0	0
104EH	Reserved	–	–	–							
104FH	Reserved	–	–	–							
1050H	A/D Converter Control Low Register	ADCCRL	R/W	0	0	0	0	0	0	0	0
1051H	A/D Converter Control High Register	ADCCRH	R/W	0	–	0	0	0	0	0	0
1052H	A/D Converter Data Low Register	ADCDRL	R	x	x	x	x	x	x	x	x
1053H	A/D Converter Data High Register	ADCDRH	R	x	x	x	x	x	x	x	x
1054H	Reserved	–	–	–							
1055H	Reserved	–	–	–							
1056H	Reserved	–	–	–							
1057H	Reserved	–	–	–							
1058H	Timer 9 Control Low Register	T9CRL	R/W	0	0	0	–	–	0	0	0
1059H	Timer 9 Control High Register	T9CRH	R/W	0	0	0	0	–	–	–	0
105AH	Timer 9 A Data Low Register	T9ADRL	R/W	1	1	1	1	1	1	1	1
105BH	Timer 9 A Data High Register	T9ADRH	R/W	1	1	1	1	1	1	1	1
105CH	Timer 9 B Data Low Register	T9BDRL	R/W	1	1	1	1	1	1	1	1
105DH	Timer 9 B Data High Register	T9BDRH	R/W	1	1	1	1	1	1	1	1
- - - - - - - - - - - - - - - - - - - -											
Reserved	–	107FH	–	–							

8.4.3 Compiler Compatible SFR

ACC (Accumulator Register) : E0H

7	6	5	4	3	2	1	0
ACC							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ACC Accumulator

B (B Register) : F0H

7	6	5	4	3	2	1	0
B							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

B B Register

SP (Stack Pointer) : 81H

7	6	5	4	3	2	1	0
SP							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 07H

SP Stack Pointer

XSP (Extend Stack Pointer) : 91H

7	6	5	4	3	2	1	0
XSP							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

XSP Extended Stack Pointer
The XSP is a high of stack pointer when XSPEN bit of the XSPCR register is set. In this case Stack is located in XDATA memory and the maximal size of stack is equal to 64k-bytes. If the XSPEN bit is set to ‘0’, the XSP register is ignored.

XSPCR (Extended Stack Pointer Control Register) : F6H

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	XSPEN
–	–	–	–	–	–	–	R/W

Initial value : 00H

SP Extended Stack Pointer Enable/Disable

0 Disable

1 Enable

DPL (Data Pointer Register Low) : 82H

7	6	5	4	3	2	1	0
DPL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPL	Data Pointer Low

DPH (Data Pointer Register High) : 83H

7	6	5	4	3	2	1	0
DPH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPH	Data Pointer High

DPL1 (Data Pointer Register Low 1) : 84H

7	6	5	4	3	2	1	0
DPL1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPL1	Data Pointer Low 1

DPH1 (Data Pointer Register High 1) : 85H

7	6	5	4	3	2	1	0
DPH1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DPH1	Data Pointer High 1

PSW (Program Status Word Register) : D0H

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

CY	Carry Flag
AC	Auxiliary Carry Flag
F0	General Purpose User-Definable Flag
RS1	Register Bank Select bit 1
RS0	Register Bank Select bit 0
OV	Overflow Flag
F1	User-Definable Flag
P	Parity Flag. Set/Cleared by hardware each instruction cycle to indicate an odd/even number of ‘1’ bits in the accumulator

EO (Extended Operation Register) : A2H

7	6	5	4	3	2	1	0
–	–	–	TRAP_EN	–	DPSEL2	DPSEL1	DPSEL0
–	–	–	R/W	–	R/W	R/W	R/W

Initial value : 00H

TRAP_EN Select the Instruction **(Keep always '0')**.

0 Select MOVC @(DPTR++), A

1 Select Software TRAP Instruction

DPSEL[2:0] Select Banked Data Pointer Register

DPSEL2	DPSEL1	SPSEL0	Description
0	0	0	DPTR0
0	0	1	DPTR1

Reserved

SPWRL (Stack Pointer Watch Register Low Byte) : F4H

7	6	5	4	3	2	1	0
SPWRL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

SPWRL Stack Pointer Watch Low

SPWRH (Stack Pointer Watch Register High Byte) : F5H

7	6	5	4	3	2	1	0
SPWRH							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 0FH

SPWRH Stack Pointer Watch High

MODINR (Mode Entry Register) : FFH

7	6	5	4	3	2	1	0
–	–	–	–	–	–	MODE1	MODE0
–	–	–	–	–	–	R	R

Initial value : 00H

MODE[1:0] MCU Mode Indicator

MODE1	MODE0	description
0	0	User mode
0	1	OCD mode
1	0	Runflag High Indicator, If the RUNFLAG pin is high level during a power-on reset, the MODE[1:0] is set to "10b"
1	1	Not available

SINTCR(System Interrupt Control Register) : F7H

7	6	5	4	3	2	1	0
–	–	–	SPOVIE	–	–	–	SPOVIFR
–	–	–	R/W	–	–	–	R/W

Initial value : 00H

SPOVIE Enable or Disable Stack Pointer Overflow Interrupt

0 Disable

1 Enable

SPOVIFR When SPOVF Interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 Stack pointer overflow Interrupt no generation

1 Stack pointer overflow Interrupt generation

Notes) 1. When the XSPEN bit of the XSPCR register is "0b"
- The stack pointer (SP) register is compared to the stack pointer watch low register (SWARL).
- If the values are same (SP[7:0] == SWARL[7:0]), the SPOVIFR bit is set to "1b".
- At this time, the XSP and SWARH registers are don't care.

2. When the XSPEN bit of the XSPCR register is "1b"
- The extended stack pointer and stack pointer (XSP:SP) registers are compared to the stack pointer watch high and low register (SWARH:SWARL).
- If the values are same (XSP:SP[15:0] == SWARH:SWARL[15:0]), the SPOVIFR bit is set to "1b".

XBANK (XRAM Bank Pointer) : F8H

7	6	5	4	3	2	1	0
XBANK							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

XBANK XRAM Bank Pointer

Notes) 1. This XBANK register holds the [15:8] part of memory address during access to data.
2. Address[15:0]: "XBANK:Ri" (Ri: R0 or R1)
3. Ex) MOVX A,@Ri ; Move external data (XBANK:Ri[15:0]) to A
MOVX @Ri,A ; Move A to external data (XBANK:Ri[15:0])

9. I/O Ports

9.1 I/O Ports

The MC97F2664 has eight groups of I/O ports (P0 ~ P7). Each port can be easily configured by software as I/O pin, internal pull up and open-drain pin to meet various system configurations and design requirements. Also P1, P2, P3, and P5 include function that can generate interrupt according to change of state of the pin.

9.2 Port Register

9.2.1 Data Register (Px)

Data Register is a bidirectional I/O port. If ports are configured as output ports, data can be written to the corresponding bit of the Px. If ports are configured as input ports, the data can be read from the corresponding bit of the Px.

9.2.2 Direction Register (PxIO)

Each I/O pin can be independently used as an input or an output through the PxIO register. Bits cleared in this register will make the corresponding pin of Px to input mode. Set bits of this register will make the pin to output mode. Almost bits are cleared by a system reset, but some bits are set by a system reset.

9.2.3 Pull-up Resistor Selection Register (PxPU)

The on-chip pull-up resistor can be connected to I/O ports individually with a pull-up resistor selection register (PxPU). The pull-up register selection controls the pull-up resister enable/disable of each port. When the corresponding bit is 1, the pull-up resister of the pin is enabled. When 0, the pull-up resister is disabled. All bits are cleared by a system reset.

9.2.4 Open-drain Selection Register (PxOD)

There are internally open-drain selection registers (PxOD) for P0 ~ P7. The open-drain selection register controls the open-drain enable/disable of each port. Almost ports become push-pull by a system reset, but some ports become open-drain by a system reset.

9.2.5 Debounce Enable Register (PxDB)

P1[7:0], P2[2:0], P2[7:5], P3[2:0] and P5[3:0] support debounce function. Debounce clocks of each ports are fx/1, fx/4, and fx/4096.

9.2.6 Port Function Selection Register (PxFSR)

These registers define alternative functions of ports. Please remember that these registers should be set properly for alternative port function. A reset clears the PxFSR register to ‘00H’, which makes all pins to normal I/O ports.

9.2.7 Register Map

Table 9-1 Port Register Map

Name	Address	Dir	Default	Description
P0	80H	R/W	00H	P0 Data Register
P0IO	99H	R/W	00H	P0 Direction Register
P0OD	1000H (XSFR)	R/W	00H	P0 Open-drain Selection Register
P0PU	1008H (XSFR)	R/W	00H	P0 Pull-up Resistor Selection Register
P0FSR	1010H (XSFR)	R/W	00H	P0 Function Selection Register
P1	88H	R/W	00H	P1 Data Register
P1IO	A1H	R/W	00H	P1 Direction Register
P1OD	1001H (XSFR)	R/W	00H	P1 Open-drain Selection Register
P1PU	1009H (XSFR)	R/W	00H	P1 Pull-up Resistor Selection Register
P1DB	E1H	R/W	00H	P1 Debounce Enable Register
P1FSR	1011H (XSFR)	R/W	00H	P1 Function Selection Register
P2	90H	R/W	00H	P2 Data Register
P2IO	B1H	R/W	00H	P2 Direction Register
P2OD	1002H (XSFR)	R/W	00H	P2 Open-drain Selection Register
P2PU	100AH (XSFR)	R/W	00H	P2 Pull-up Resistor Selection Register
P2DB	E9H	R/W	00H	P2 Debounce Enable Register
P2FSR	1012H (XSFR)	R/W	00H	P2 Function Selection Register
P3	98H	R/W	00H	P3 Data Register
P3IO	B9H	R/W	00H	P3 Direction Register
P3PU	100BH (XSFR)	R/W	00H	P3 Pull-up Resistor Selection Register
P3OD	1003H (XSFR)	R/W	00H	P3 Open-drain Selection Register
P35DB	F1H	R/W	00H	P3/P5 Debounce Enable Register
P3FSR	1013H (XSFR)	R/W	00H	P3 Function Selection Register
P4	A0H	R/W	00H	P4 Data Register
P4IO	C1H	R/W	00H	P4 Direction Register
P4OD	1004H (XSFR)	R/W	00H	P4 Open-drain Selection Register
P4PU	100CH (XSFR)	R/W	00H	P4 Pull-up Resistor Selection Register
P4FSR	1014H (XSFR)	R/W	00H	P4 Function Selection Register
P5	B0H	R/W	00H	P5 Data Register
P5IO	C9H	R/W	00H	P5 Direction Register
P5OD	1005H (XSFR)	R/W	00H	P5 Open-drain Selection Register
P5PU	100DH (XSFR)	R/W	00H	P5 Pull-up Resistor Selection Register
P5FSR	1015H (XSFR)	R/W	00H	P5 Function Selection Register

Table 9-1 Port Register Map (CONTINUED)

Name	Address	Dir	Default	Description
P6	B8H	R/W	00H	P6 Data Register
P6IO	D1H	R/W	00H	P6 Direction Register
P6OD	1006H (XSFR)	R/W	00H	P6 Open-drain Selection Register
P6PU	100EH (XSFR)	R/W	00H	P6 Pull-up Resistor Selection Register
P6FSR	1016H (XSFR)	R/W	00H	P6 Function Selection Register
P7	C0H	R/W	00H	P7 Data Register
P7IO	D9H	R/W	00H	P7 Direction Register
P7OD	1007H (XSFR)	R/W	00H	P7 Open-drain Selection Register
P7PU	100FH (XSFR)	R/W	00H	P7 Pull-up Resistor Selection Register
P7FSR	1017H (XSFR)	R/W	00H	P7 Function Selection Register

9.3 P0 Port

9.3.1 P0 Port Description

P0 is 8-bit I/O port. P0 control registers consist of P0 data register (P0), P0 direction register (P0IO), P0 pull-up resistor selection register (P0PU), and P0 open-drain selection register (P0OD). Refer to the port function selection registers for the P0 function selection.

9.3.2 Register description for P0

P0 (P0 Data Register) : 80H

7	6	5	4	3	2	1	0
P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0[7:0] I/O Data

P0IO (P0 Direction Register) : 99H

7	6	5	4	3	2	1	0
P07IO	P06IO	P05IO	P04IO	P03IO	P02IO	P01IO	P00IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0IO[7:0] P0 Data I/O Direction.

0 Input

1 Output

P0PU (P0 Pull-up Resistor Selection Register) : 1008H (XFSR)

7	6	5	4	3	2	1	0
P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0PU[7:0] Configure Pull-up Resistor of P0 Port

0 Disable

1 Enable

P0OD (P0 Open-drain Selection Register) : 1000H (XFSR)

7	6	5	4	3	2	1	0
P07OD	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0OD[7:0] Configure Open-drain of P0 Port

0 Push-pull output

1 Open-drain output

P0FSR (Port 0 Function Selection Register) : 1010H (XSFR)

7	6	5	4	3	2	1	0
P0FSR7	P0FSR6	P0FSR5	P0FSR4	P0FSR3	P0FSR2	P0FSR1	P0FSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P0FSR7 P07 Function select
- 0 I/O Port
- 1 AN14 Function

P0FSR6 P06 Function select
- 0 I/O Port
- 1 AN13 Function

P0FSR5 P05 Function select
- 0 I/O Port
- 1 AN12 Function

P0FSR4 P04 Function select
- 0 I/O Port
- 1 AN11 Function

P0FSR3 P03 Function Select
- 0 I/O Port
- 1 AN10 Function

P0FSR2 P02 Function select
- 0 I/O Port
- 1 AN9 Function

P0FSR1 P01 Function select
- 0 I/O Port
- 1 AN8 Function

P0FSR0 P00 Function select
- 0 I/O Port
- 1 AVREF Function

9.4 P1 Port

9.4.1 P1 Port Description

P1 is 8-bit I/O port. P1 control registers consist of P1 data register (P1), P1 direction register (P1IO), debounce enable register (P1DB), P1 pull-up resistor selection register (P1PU), and P1 open-drain selection register (P1OD) . Refer to the port function selection registers for the P1 function selection.

9.4.2 Register description for P1

P1 (P1 Data Register) : 88H

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1[7:0] I/O Data

P1IO (P1 Direction Register) : A1H

7	6	5	4	3	2	1	0
P17IO	P16IO	P15IO	P14IO	P13IO	P12IO	P11IO	P10IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1IO[7:0] P1 Data I/O Direction

0 Input

1 Output

NOTE) EINT0-EINT7 function possible when input

P1PU (P1 Pull-up Resistor Selection Register) : 1009H (XSFR)

7	6	5	4	3	2	1	0
P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1PU[7:0] Configure Pull-up Resistor of P1 Port

0 Disable

1 Enable

P1OD (P1 Open-drain Selection Register) : 1001H (XSFR)

7	6	5	4	3	2	1	0
P17OD	P16OD	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1OD[7:0] Configure Open-drain of P1 Port

0 Push-pull output

1 Open-drain output

P1DB (P1 Debounce Enable Register) : E1H

7	6	5	4	3	2	1	0
P17DB	P16DB	P15DB	P14DB	P13DB	P12DB	P11DB	P10DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P17DB Configure Debounce of P17 Port
- 0 Disable
- 1 Enable

P16DB Configure Debounce of P16 Port
- 0 Disable
- 1 Enable

P15DB Configure Debounce of P15 Port
- 0 Disable
- 1 Enable

P14DB Configure Debounce of P14 Port
- 0 Disable
- 1 Enable

P13DB Configure Debounce of P13 Port
- 0 Disable
- 1 Enable

P12DB Configure Debounce of P12 Port
- 0 Disable
- 1 Enable

P11DB Configure Debounce of P11 Port
- 0 Disable
- 1 Enable

P10DB Configure Debounce of P10 Port
- 0 Disable
- 1 Enable

NOTES) 1. If a level is not detected on an enabled pin three or more times in a row at the sampling clock, the signal is eliminated as noise.

2. A pulse level should be input for the duration of 3 clocks or more to be actually detected as a valid edge.
3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.
4. Refer to the port 2 debounce enable register (P2DB) for the debounce clock of port 1.

P1FSR (Port 1 Function Selection Register) : 1011H (XSFR)

7	6	5	4	3	2	1	0
P1FSR7	P1FSR6	P1FSR5	P1FSR4	P1FSR3	P1FSR2	P1FSR1	P1FSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P1FSR7 P17 Function select
- 0 I/O Port (EINT7 function possible when input)
- 1 AN7 Function

P1FSR6 P16 Function select
- 0 I/O Port (EINT6 function possible when input)
- 1 AN6 Function

P1FSR5 P15 Function select
- 0 I/O Port (EINT5 function possible when input)
- 1 AN5 Function

P1FSR4 P14 Function select
- 0 I/O Port (EINT4 function possible when input)
- 1 AN4 Function

P1FSR3 P13 Function Select
- 0 I/O Port (EINT3 function possible when input)
- 1 AN3 Function

P1FSR2 P12 Function select
- 0 I/O Port (EINT2 function possible when input)
- 1 AN2 Function

P1FSR1 P11 Function select
- 0 I/O Port (EINT1 function possible when input)
- 1 AN1 Function

P1FSR0 P10 Function select
- 0 I/O Port (EINT0 function possible when input)
- 1 AN0 Function

9.5 P2 Port

9.5.1 P2 Port Description

P2 is 8-bit I/O port. P2 control registers consist of P2 data register (P2), P2 direction register (P2IO), debounce enable register (P2DB), P2 pull-up resistor selection register (P2PU) and P2 open-drain selection register (P2OD). Refer to the port function selection registers for the P2 function selection.

9.5.2 Register description for P2

P2 (P2 Data Register) : 90H

7	6	5	4	3	2	1	0
P27	P26	P25	P24	P23	P22	P21	P20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P2[7:0] I/O Data

P2IO (P2 Direction Register) : B1H

7	6	5	4	3	2	1	0
P27IO	P26IO	P25IO	P24IO	P23IO	P22IO	P21IO	P20IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P2IO[7:0] P2 Data I/O Direction

0 Input

1 Output

NOTE) EINT8/EINT9/EINTA/EINT17/EINT18/EINT19/RXD4 function possible when input

P2PU (P2 Pull-up Resistor Selection Register) : 100AH (XSFR)

7	6	5	4	3	2	1	0
P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P2PU[7:0] Configure Pull-up Resistor of P2 Port

0 Disable

1 Enable

P2OD (P2 Open-drain Selection Register) : 1002H (XSFR)

7	6	5	4	3	2	1	0
P27OD	P26OD	P25OD	P24OD	P23OD	P22OD	P21OD	P20OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P2OD[7:0] Configure Open-drain of P2 Port

0 Push-pull output

1 Open-drain output

P2DB (P2 Debounce Enable Register) : E9H

7	6	5	4	3	2	1	0
DBCLK1	DBCLK0	P27DB	P26DB	P25DB	P22DB	P21DB	P20DB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DBCLK[1:0] Configure Debounce Clock of Port

DBCLK1	DBCLK0	description
0	0	fx/1
0	1	fx/4
1	0	fx/4096
1	1	Reserved

P27DB Configure Debounce of P27 Port

0 Disable

1 Enable

P26DB Configure Debounce of P26 Port

0 Disable

1 Enable

P25DB Configure Debounce of P25 Port

0 Disable

1 Enable

P22DB Configure Debounce of P22 Port

0 Disable

1 Enable

P21DB Configure Debounce of P21 Port

0 Disable

1 Enable

P20DB Configure Debounce of P20 Port

0 Disable

1 Enable

NOTES) 1. If a level is not detected on an enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.

2. A pulse level should be input for the duration of 3 clocks or more to be actually detected as a valid edge.

3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.

P2FSR (Port 2 Function Selection Register) : 1012H (XSFR)

7	6	5	4	3	2	1	0
–	–	–	–	P2FSR3	P2FSR2	P2FSR1	P2FSR0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P2FSR3 P23 Function select
- 0 I/O Port
- 1 TXD4 Function

P2FSR2 P22 Function select
- 0 I/O Port (EINT19 function possible when input)
- 1 T9O/PWM9O Function

P2FSR1 P21 Function select
- 0 I/O Port (EINT18 function possible when input)
- 1 T8O/PWM8O Function

P2FSR0 P20 Function select
- 0 I/O Port (EINT17 function possible when input)
- 1 T7O/PWM7O Function

9.6 P3 Port

9.6.1 P3 Port Description

P3 is 8-bit I/O port. P3 control registers consist of P3 data register (P3), P3 direction register (P3IO), debounce enable register (P35DB),P3 pull-up resistor selection register (P3PU) and P3 open-drain selection register (P3OD). Refer to the port function selection registers for the P3 function selection.

9.6.2 Register description for P3

P3 (P3 Data Register) : 98H

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P3[7:0] I/O Data

P3IO (P3 Direction Register) : B9H

7	6	5	4	3	2	1	0
P37IO	P36IO	P35IO	P34IO	P33IO	P32IO	P31IO	P30IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P3IO[7:0] P3 Data I/O Direction

0 Input

1 Output

NOTE) EINT14/EINT15/EINT16/SS2/EC7/EC8/EC9 function possible when input

P3PU (P3 Pull-up Resistor Selection Register) : 100BH (XSFR)

7	6	5	4	3	2	1	0
P37PU	P36PU	P35PU	P34PU	P33PU	P32PU	P31PU	P30PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P3PU[7:0] Configure Pull-up Resistor of P3 Port

0 Disable

1 Enable

P3OD (P3 Open-drain Selection Register) : 1003H (XSFR)

7	6	5	4	3	2	1	0
P37OD	P36OD	P35OD	P34OD	P33OD	P32OD	P31OD	P30OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P3OD[7:0] Configure Open-drain of P3 Port

0 Push-pull output

1 Open-drain output

P35DB (P35 Debounce Enable Register) : F1H

7	6	5	4	3	2	1	0
–	P53DB	P52DB	P51DB	P50DB	P32DB	P31DB	P30DB
–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P53DB Configure Debounce of P53 Port
- 0 Disable
- 1 Enable

P52DB Configure Debounce of P52 Port
- 0 Disable
- 1 Enable

P51DB Configure Debounce of P51 Port
- 0 Disable
- 1 Enable

P50DB Configure Debounce of P50 Port
- 0 Disable
- 1 Enable

P32DB Configure Debounce of P32 Port
- 0 Disable
- 1 Enable

P31DB Configure Debounce of P31 Port
- 0 Disable
- 1 Enable

P30DB Configure Debounce of P30 Port
- 0 Disable
- 1 Enable

NOTES) 1. If a level is not detected on an enabled pin three or four times in a row at the sampling clock, the signal is eliminated as noise.

2. A pulse level should be input for the duration of 3 clocks or more to be actually detected as a valid edge.

3. The port debounce is automatically disabled at stop mode and recovered after stop mode release.

4. Refer to the port 2 debounce enable register (P2DB) for the debounce clock of port 3 and port 5.

P3FSR (Port 3 Function Selection Register) : 1013H (XSFR)

7	6	5	4	3	2	1	0
P3FSR7	P3FSR6	P3FSR5	P3FSR4	P3FSR3	P3FSR2	P3FSR1	P3FSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P3FSR7 P36 Function select
- 0 I/O Port (EC8 function possible when input)
- 1 SCK2 Function

P3FSR6 P35 Function select
- 0 I/O Port (EC7 function possible when input)
- 1 MISO2 Function

P3FSR5 P34 Function select
- 0 I/O Port
- 1 MOSI2 Function

P3FSR[4:3] P33 Function select

P3FSR4	P3FSR3	description
0	0	I/O Port
0	1	BUZO
1	0	EC6
1	1	Not used

P3FSR2 P32 Function select
- 0 I/O Port (EINT16 function possible when input)
- 1 T6O/PWM6O Function

P3FSR1 P31 Function select
- 0 I/O Port (EINT15 function possible when input)
- 1 T5O/PWM5O Function

P3FSR0 P30 Function select
- 0 I/O Port (EINT14 function possible when input)
- 1 T4O/PWM4O Function

9.7 P4 Port

9.7.1 P4 Port Description

P4 is 8-bit I/O port. P4 control registers consist of P4 data register (P4), P4 direction register (P4IO), P4 pull-up resistor selection register (P4PU) and P4 open-drain selection register (P4OD). Refer to the port function selection registers for the P4 function selection.

9.7.2 Register description for P4

P4 (P4 Data Register) : A0H

7	6	5	4	3	2	1	0
P47	P46	P45	P44	P43	P42	P41	P40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P4[7:0] I/O Data

P4IO (P4 Direction Register) : C1H

7	6	5	4	3	2	1	0
P47IO	P46IO	P45IO	P44IO	P43IO	P42IO	P41IO	P40IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P4IO[7:0] P4 Data I/O Direction

0 Input

1 Output

NOTE) SS1/RXD2/RXD3 function possible when input

P4PU (P4 Pull-up Resistor Selection Register) : 100CH (XSFR)

7	6	5	4	3	2	1	0
P47PU	P46PU	P45PU	P44PU	P43PU	P42PU	P41PU	P40PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P4PU[7:0] Configure Pull-up Resistor of P4 Port

0 Disable

1 Enable

P4OD (P4 Open-drain Selection Register) : 1004H (XSFR)

7	6	5	4	3	2	1	0
P47OD	P46OD	P45OD	P44OD	P43OD	P42OD	P41OD	P40OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P4OD[7:0] Configure Open-drain of P4 Port

0 Push-pull output

P4FSR (Port 4 Function Selection Register) : 1014H (XSFR)

7	6	5	4	3	2	1	0
–	–	–	P4FSR4	P4FSR3	P4FSR2	P4FSR1	P4FSR0
–	–	–	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P4FSR4 P46 Function select
- 0 I/O Port
- 1 TXD3 Function

P4FSR3 P44 Function Select
- 0 I/O Port
- 1 TXD2 Function

P4FSR2 P42 Function select
- 0 I/O Port
- 1 SCK1 Function

P4FSR1 P41 Function Select
- 0 I/O Port
- 1 RXD1/SCL1/MISO1 Function

P4FSR0 P40 Function select
- 0 I/O Port
- 1 TXD1/SDA1/MOSI1 Function

9.8 P5 Port

9.8.1 P5 Port Description

P5 is 8-bit I/O port. P5 control registers consist of P5 data register (P5), P5 direction register (P5IO), P5 pull-up resistor selection register (P5PU) and P5 open-drain selection register (P5OD). Refer to the port function selection registers for the P5 function selection.

9.8.2 Register description for P5

P5 (P5 Data Register) : B0H

7	6	5	4	3	2	1	0
P57	P56	P55	P54	P53	P52	P51	P50
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P5[7:0] I/O Data

P5IO (P5 Direction Register) : C9H

7	6	5	4	3	2	1	0
P57IO	P56IO	P55IO	P54IO	P53IO	P52IO	P51IO	P50IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P5IO[7:0] P5 Data I/O Direction

0 Input

1 Output

NOTE) EINT10-EINT13/EC0-EC3 function possible when input

P5PU (P5 Pull-up Resistor Selection Register) : 100DH (XSFR)

7	6	5	4	3	2	1	0
P57PU	P56PU	P55PU	P54PU	P53PU	P52PU	P51PU	P50PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P5PU[7:0] Configure Pull-up Resistor of P5 Port

0 Disable

1 Enable

P5OD (P5 Open-drain Selection Register) : 1005H (XSFR)

7	6	5	4	3	2	1	0
P57OD	P56OD	P55OD	P54OD	P53OD	P52OD	P51OD	P50OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P5OD[7:0] Configure Open-drain of P5 Port

0 Push-pull output

1 Open-drain output

P5FSR (Port 5 Function Selection Register) : 1015H (XSFR)

7	6	5	4	3	2	1	0
–	–	–	–	P5FSR3	P5FSR2	P5FSR1	P5FSR0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P5FSR3 P53 Function Select
- 0 I/O Port (EINT13 function possible when input)
- 1 T3O/PWM3O Function

P5FSR2 P52 Function select
- 0 I/O Port (EINT12 function possible when input)
- 1 T2O/PWM2O Function

P5FSR1 P51 Function Select
- 0 I/O Port (EINT11 function possible when input)
- 1 T1O/PWM1O Function

P5FSR0 P50 Function select
- 0 I/O Port (EINT10 function possible when input)
- 1 T0O/PWM0O Function

NOTE) Refer to the configure option for the P57/EC3/RESETB.

9.9 P6 Port

9.9.1 P6 Port Description

P6 is 8-bit I/O port. P6 control registers consist of P6 data register (P6), P6 direction register (P6IO), P6 pull-up resistor selection register (P6PU) and P6 open-drain selection register (P6OD). Refer to the port function selection registers for the P6 function selection.

9.9.2 Register description for P6

P6 (P6 Data Register) : B8H

7	6	5	4	3	2	1	0
P67	P66	P65	P64	P63	P62	P61	P60
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P6[7:0] I/O Data

P6IO (P6 Direction Register) : D1H

7	6	5	4	3	2	1	0
P67IO	P66IO	P65IO	P64IO	P63IO	P62IO	P61IO	P60IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P6IO[7:0] P6 Data I/O Direction

0 Input

1 Output

NOTE) SS0 function possible when input

P6PU (P6 Pull-up Resistor Selection Register) : 100EH (XSFR)

7	6	5	4	3	2	1	0
P67PU	P66PU	P65PU	P64PU	P63PU	P62PU	P61PU	P60PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P6PU[7:0] Configure Pull-up Resistor of P6 Port

0 Disable

1 Enable

P6OD (P6 Open-drain Selection Register) : 1006H (XSFR)

7	6	5	4	3	2	1	0
P67OD	P66OD	P65OD	P64OD	P63OD	P62OD	P61OD	P60OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P6OD[7:0] Configure Open-drain of P6 Port

0 Push-pull output

1 Open-drain output

P6FSR (Port 6 Function Selection Register) : 1016H (XSFR)

7	6	5	4	3	2	1	0
–	P6FSR6	P6FSR5	P6FSR4	P6FSR3	P6FSR2	P6FSR1	P6FSR0
–	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P6FSR6 P67 Function select
- 0 I/O Port
- 1 SXOUT Function

P6FSR5 P66 Function select
- 0 I/O Port
- 1 SXIN Function

P6FSR4 P65 Function select
- 0 I/O Port
- 1 XIN Function

P6FSR3 P64 Function Select
- 0 I/O Port
- 1 XOUT Function

P6FSR2 P63 Function select
- 0 I/O Port
- 1 TXD0/SDA0/MOSI0 Function

P6FSR1 P62 Function select
- 0 I/O Port
- 1 RXD0/SCL0/MISO0 Function

P6FSR0 P61 Function select
- 0 I/O Port
- 1 SCK0 Function

9.10 P7 Port

9.10.1 P7 Port Description

P7 is 5-bit I/O port. P7 control registers consist of P7 data register (P7), P7 direction register (P7IO), P7 pull-up resistor selection register (P7PU) and P7 open-drain selection register (P7OD). Refer to the port function selection registers for the P7 function selection.

9.10.2 Register description for P7

P7 (P7 Data Register) : C0H

7	6	5	4	3	2	1	0
P77	P76	P75	P74	P73	P72	P71	P70
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P7[7:0] I/O Data

P7IO (P7 Direction Register) : D9H

7	6	5	4	3	2	1	0
P77IO	P76IO	P75IO	P74IO	P73IO	P72IO	P71IO	P70IO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P7IO[7:0] P7 Data I/O Direction

0 Input

1 Output

NOTE) EC4/EC5/SS3 function possible when input

P7PU (P7 Pull-up Resistor Selection Register) : 100FH (XSFR)

7	6	5	4	3	2	1	0
P77PU	P76PU	P75PU	P74PU	P73PU	P72PU	P71PU	P70PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P7PU[7:0] Configure Pull-up Resistor of P7 Port

0 Disable

1 Enable

P7OD (P7 Open-drain Selection Register) : 1007H (XSFR)

7	6	5	4	3	2	1	0
P77OD	P76OD	P75OD	P74OD	P73OD	P72OD	P71OD	P70OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

P7OD[7:0] Configure Open-drain of P7 Port

0 Push-pull output

1 Open-drain output

P7FSR (Port 7 Function Selection Register) : 1017H (XSFR)

7	6	5	4	3	2	1	0
–	–	–	–	P7FSR3	P7FSR2	P7FSR1	P7FSR0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

P7FSR3 P73 Function Select
- 0 I/O Port
- 1 EC6 Function

P7FSR2 P72 Function select
- 0 I/O Port (EC4 function possible when input)
- 1 SCK3 Function

P7FSR1 P71 Function Select
- 0 I/O Port
- 1 MISO3 Function

P7FSR0 P70 Function select
- 0 I/O Port
- 1 MOSI3 Function

10. Interrupt Controller

10.1 Overview

The MC97F2664 supports up to 24 interrupt sources. The interrupts have separate enable register bits associated with them, allowing software control. They can also have four levels of priority assigned to them. The non-maskable interrupt source is always enabled with a higher priority than any other interrupt source, and is not controllable by software. The interrupt controller has following features:

- Receive the request from 24 interrupt source
- Individual priority
- 4 priority levels
- Multi Interrupt possibility
- If the requests of different priority levels are received simultaneously, the request of higher priority level is served first.
- Each interrupt source can be controlled by EA bit and each IEx bit
- Interrupt latency: 3~9 machine cycles in single interrupt system

The non-maskable interrupt is always enabled. The maskable interrupts are enabled through four pair of interrupt enable registers (IE, IE1, IE2, IE3). Each bit of IE, IE1, IE2, IE3 register individually enables/disables the corresponding interrupt source. Overall control is provided by bit 7 of IE (EA). When EA is set to ‘0’, all interrupts are disabled: when EA is set to ‘1’, interrupts are individually enabled or disabled through the other bits of the interrupt enable registers. The EA bit is always cleared to ‘0’ jumping to an interrupt service vector and set to ‘1’ executing the [RETI] instruction. The MC97F2664 supports a four-level priority scheme. Each maskable interrupt is individually assigned to one of four priority levels according to IP0H/L, IP1H/L, IP2H/L, and IP3H/L.

Default interrupt mode is level-trigger mode basically, but if needed, it is possible to change to edge-trigger mode. Priority of a group is set by two bits of interrupt priority registers (one bit from IP, another one from IP1). Interrupt service routine serves higher priority interrupt first. If two requests of different priority levels are received simultaneously, the request of higher priority level is served prior to the lower one.

10.2 External Interrupt

The external interrupt on EINT0~A and EINT10~19 pins receive various interrupt request depending on the external interrupt polarity 0 high/low register (EIPOL0H/L), external interrupt polarity 1 high/low register (EIPOL1H/L), and external interrupt polarity 2 high/low register (EIPOL2H/L), as shown in Figure 10.1. Also each external interrupt source has enable/disable bits. The external interrupt flag 0 register (EIFLAG0), external interrupt flag 1 register (EIFLAG1) and external interrupt flag 2 register (EIFLAG2) provides the status of external interrupts.

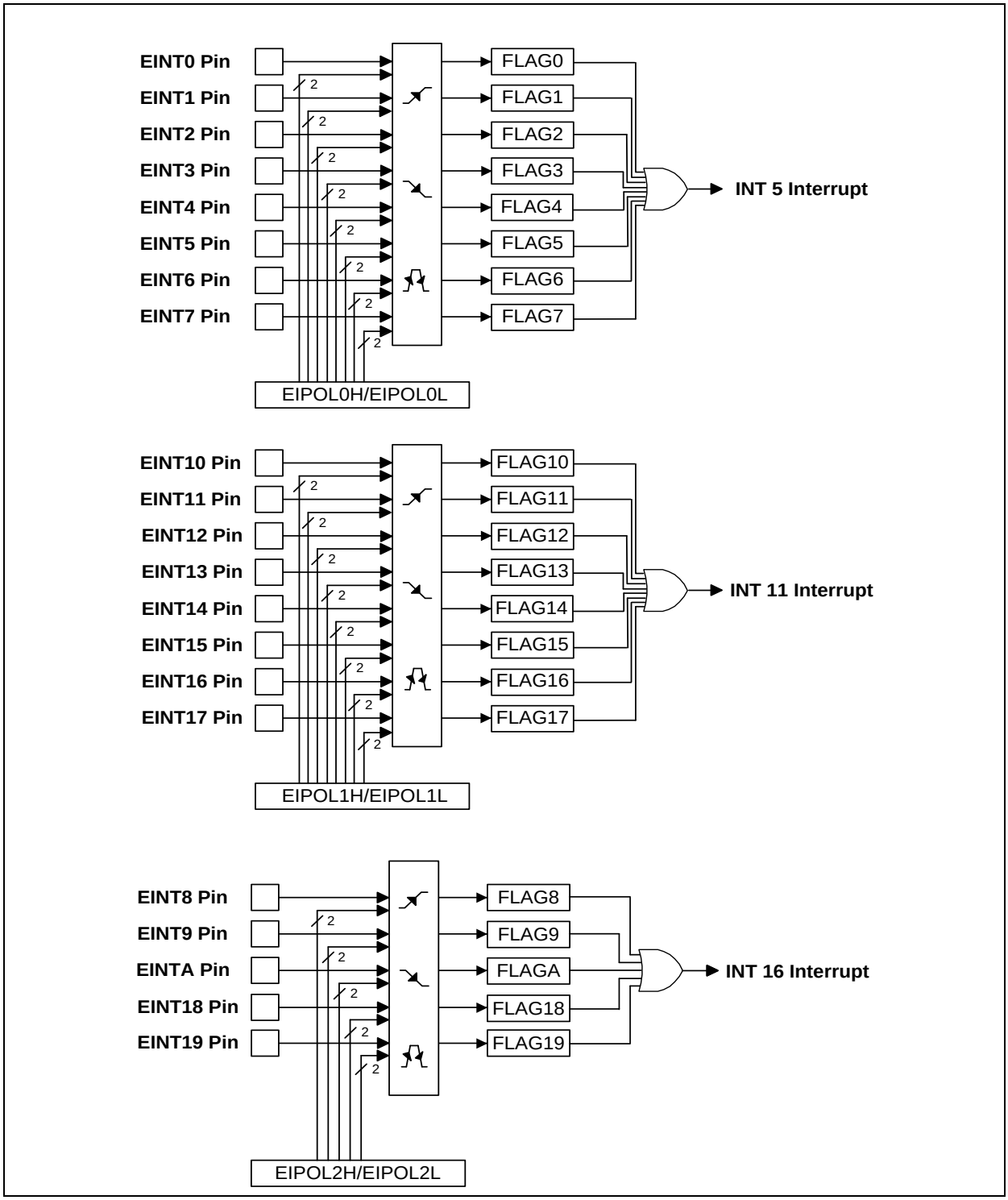


Figure 10.1 External Interrupt Description

10.3 Block Diagram

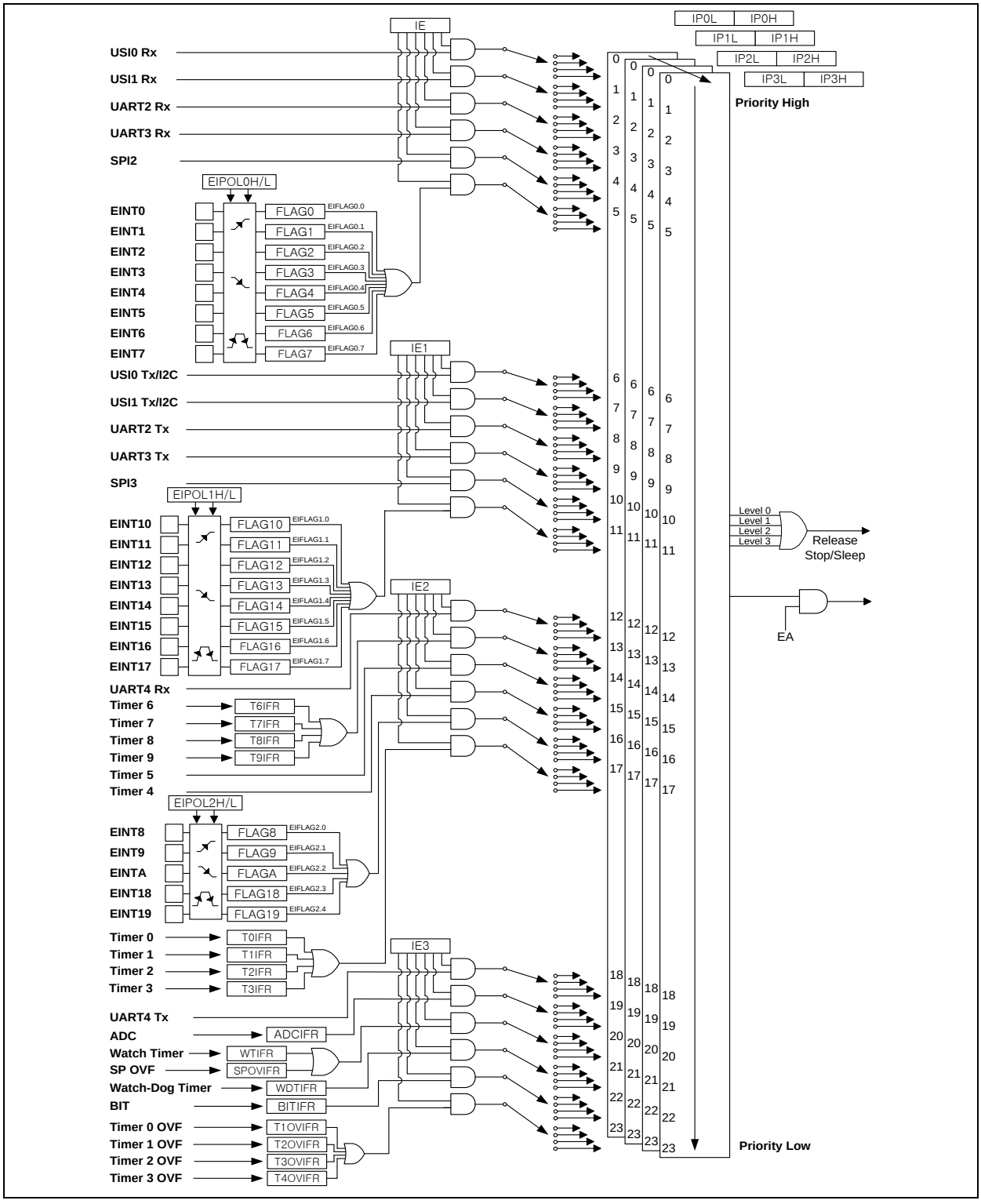


Figure 10.2 Block Diagram of Interrupt

NOTES) 1. The release signal may be generated by all interrupt sources which are enabled without reference to a priority level.

2. An interrupt request is delayed during data are written to IE, IE1, IE2, IE3, IP0L/H, IP1L/H, IP2L/H, IP3L/H and PCON register.

10.4 Interrupt Vector Table

The interrupt controller supports 24 interrupt sources as shown in the Table 10-2. When interrupt is served, long call instruction (LCALL) is executed and program counter jumps to the vector address. All interrupt requests have their own priority order.

Table 10-1 Interrupt Vector Address Table

Interrupt Source	Symbol	Interrupt Enable Bit	Polarity	Mask	Vector Address
Hardware RESET	RESETB	0 0	0	Non-Maskable	0000H
USI0 Rx Interrupt	INT0	IE.0	1	Maskable	0003H
USI1 Rx Interrupt	INT1	IE.1	2	Maskable	000BH
UART2 Rx Interrupt	INT2	IE.2	3	Maskable	0013H
UART3 Rx Interrupt	INT3	IE.3	4	Maskable	001BH
SPI2 Interrupt	INT4	IE.4	5	Maskable	0023H
External Interrupt 0 – 7	INT5	IE.5	6	Maskable	002BH
USI0 Tx/I2C Interrupt	INT6	IE1.0	7	Maskable	0033H
USI1 Tx/I2C Interrupt	INT7	IE1.1	8	Maskable	003BH
UART2 Tx Interrupt	INT8	IE1.2	9	Maskable	0043H
UART3 Tx Interrupt	INT9	IE1.3	10	Maskable	004BH
SPI3 Interrupt	INT10	IE1.4	11	Maskable	0053H
External Interrupt 10 – 17	INT11	IE1.5	12	Maskable	005BH
UART4 Rx Interrupt	INT12	IE2.0	13	Maskable	0063H
T6/7/8/9 Match Interrupt	INT13	IE2.1	14	Maskable	006BH
T5 Match Interrupt	INT14	IE2.2	15	Maskable	0073H
T4 Match Interrupt	INT15	IE2.3	16	Maskable	007BH
External Interrupt 8 – A External Interrupt 18-19	INT16	IE2.4	17	Maskable	0083H
T0/1/2/3 Match Interrupt	INT17	IE2.5	18	Maskable	008BH
UART4 Tx Interrupt	INT18	IE3.0	19	Maskable	0093H
ADC Interrupt	INT19	IE3.1	20	Maskable	009BH
WT Interrupt SP Overflow Interrupt	INT20	IE3.2	21	Maskable	00A3H
WDT Interrupt	INT21	IE3.3	22	Maskable	00ABH
BIT Interrupt	INT22	IE3.4	23	Maskable	00B3H
T0/1/2/3 Overflow Interrupt	INT23	IE3.5	24	Maskable	00BBH

For maskable interrupt execution, EA bit must set ‘1’ and specific interrupt must be enabled by writing ‘1’ to associated bit in the IEx. If an interrupt request is received, the specific interrupt request flag is set to ‘1’. And it remains ‘1’ until CPU accepts interrupt. If the interrupt is served, the interrupt request flag will be cleared automatically.

10.5 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to '0' by a reset or an instruction. Interrupt acceptance always generates at last cycle of the instruction. So instead of fetching the current instruction, CPU executes internally LCALL instruction and saves the PC at stack. For the interrupt service routine, the interrupt controller gives the address of LJMP instruction to CPU. Since the end of the execution of current instruction, it needs 3~9 machine cycles to go to the interrupt service routine. The interrupt service task is terminated by the interrupt return instruction [RETI]. Once an interrupt request is generated, the following process is performed.

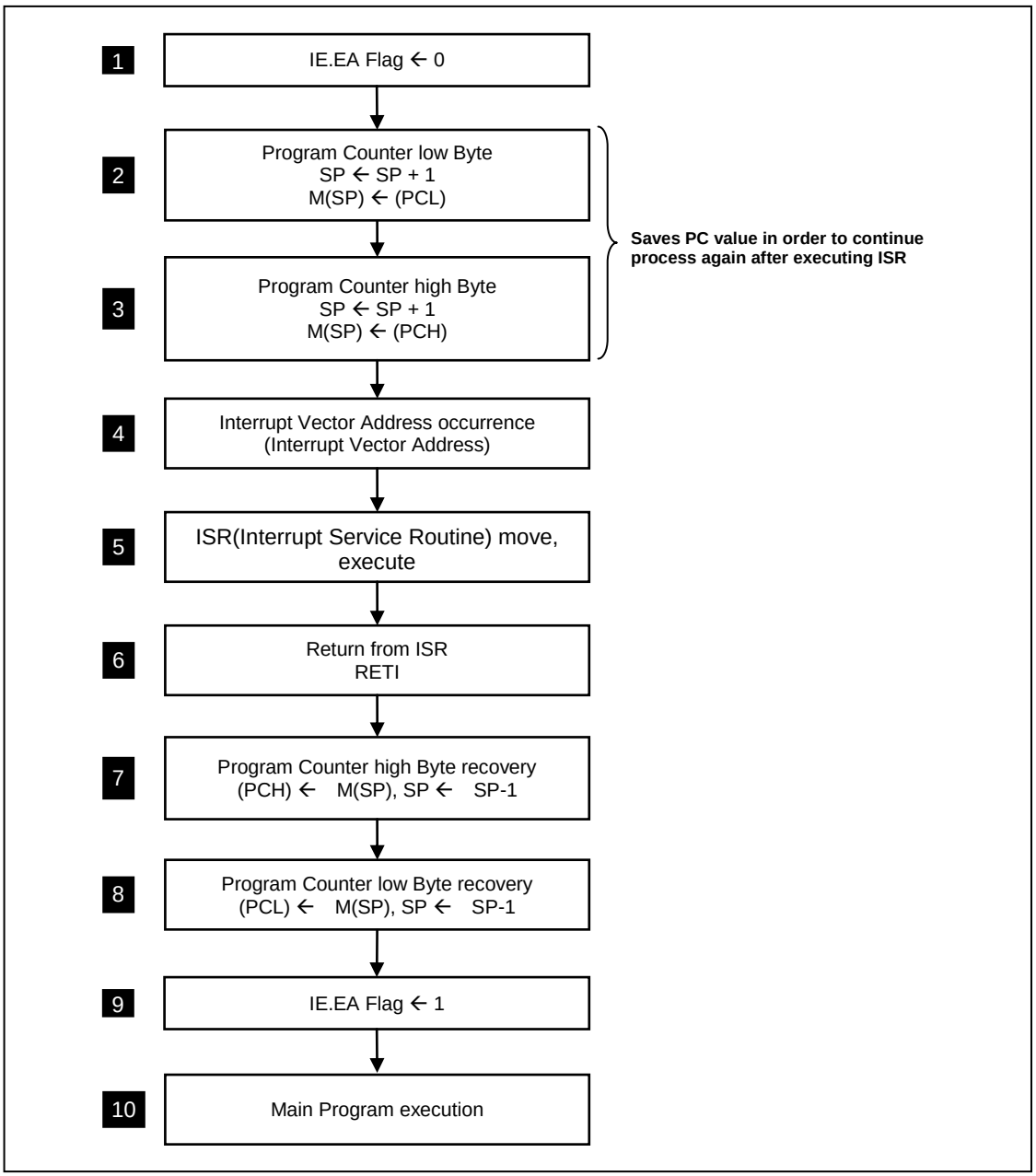


Figure 10.3 Interrupt Sequence Flow

10.6 Effective Timing after Controlling Interrupt Bit

Case a) Control Interrupt Enable Register (IE, IE1, IE2, IE3)

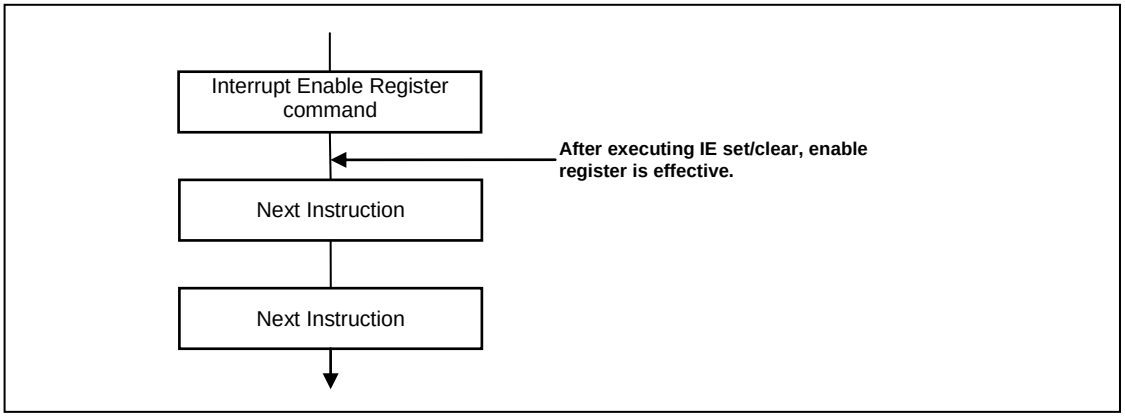


Figure 10.4 Effective Timing of Interrupt Enable Register

Case b) Interrupt flag Register

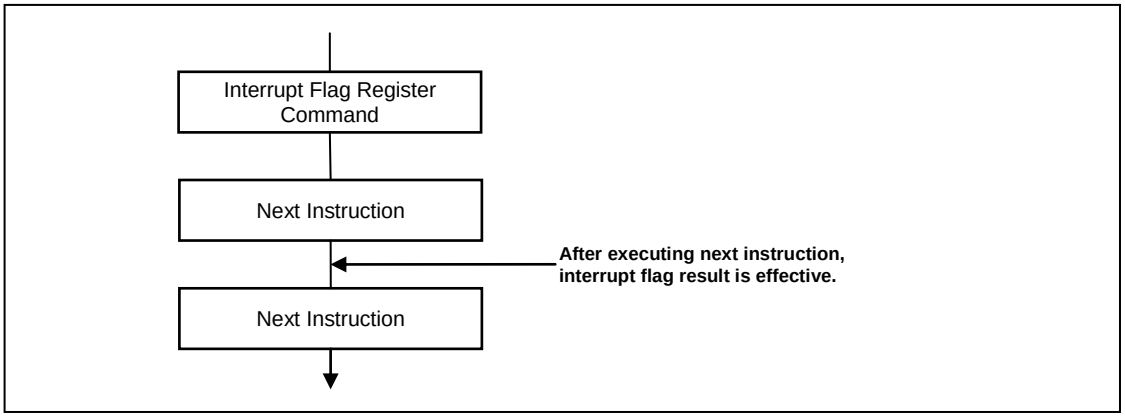


Figure 10.5 Effective Timing of Interrupt Flag Register

10.7 Multi Interrupt

If two requests of different priority levels are received simultaneously, the request of higher priority level is served first. If more than one interrupt request are received, the interrupt polling sequence determines which request is served first by hardware. However, for special features, multi-interrupt processing can be executed by software.

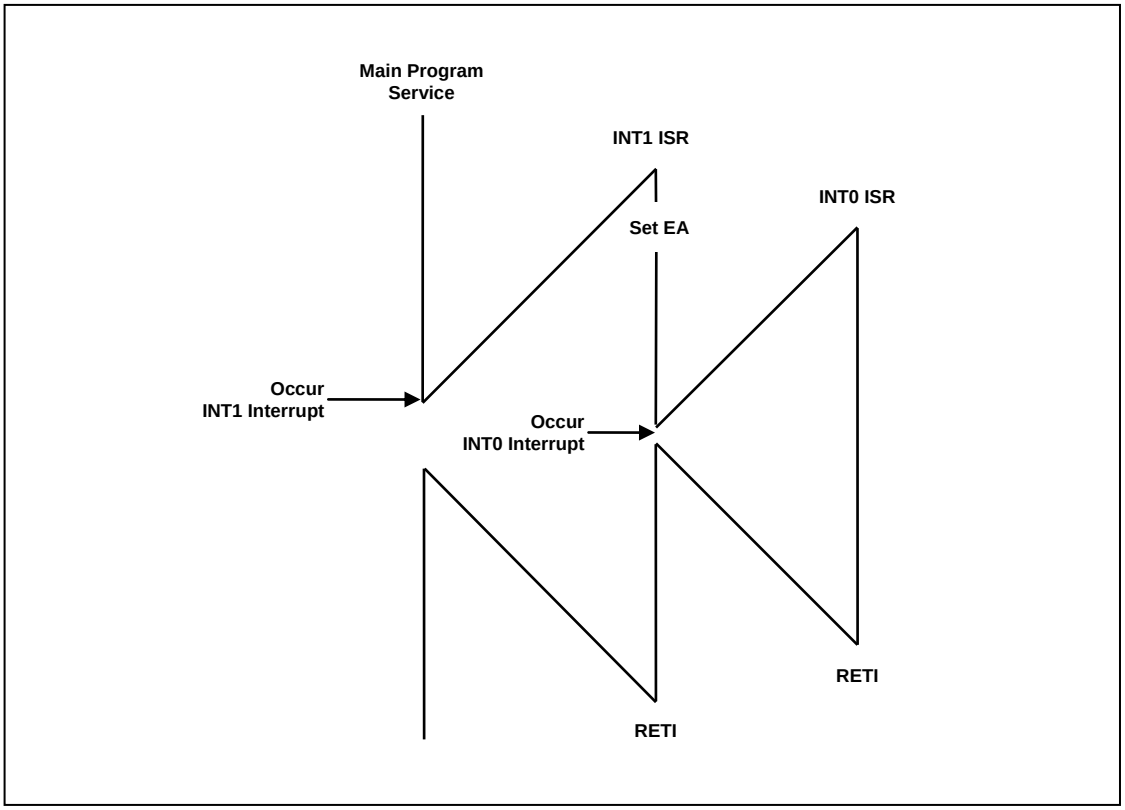


Figure 10.6 Effective Timing of Multi-Interrupt

Figure 10.6 shows an example of multi-interrupt processing. While INT1 is served, INT0 which has higher priority than INT1 is occurred. Then INT0 is served immediately and then the remain part of INT1 service routine is executed. If the priority level of INT0 is same or lower than INT1, INT0 will be served after the INT1 service has completed.

An interrupt service routine may be only interrupted by an interrupt of higher priority and, if two interrupts of different priority occur at the same time, the higher level interrupt will be served first. An interrupt cannot be interrupted by another interrupt of the same or a lower priority level. If two interrupts of the same priority level occur simultaneously, the service order for those interrupts is determined by the scan order.

10.8 Interrupt Enable Accept Timing

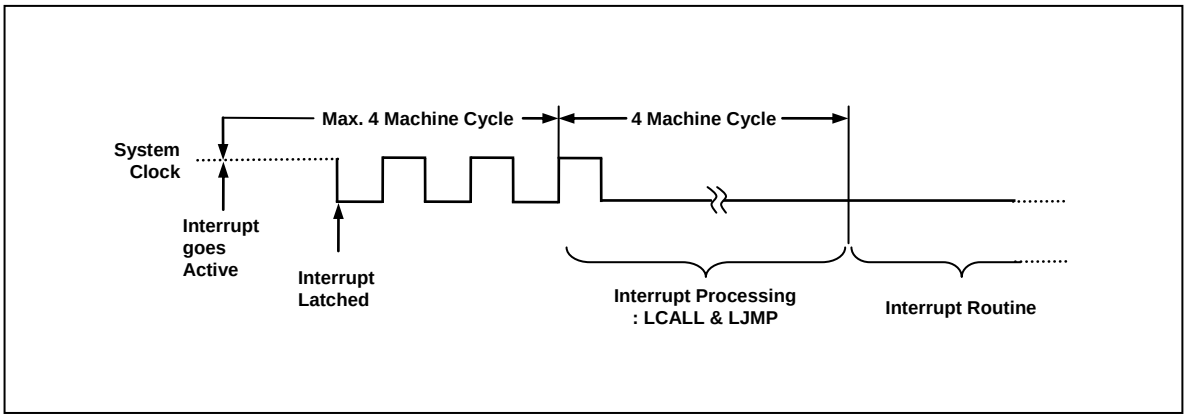


Figure 10.7 Interrupt Response Timing Diagram

10.9 Interrupt Service Routine Address

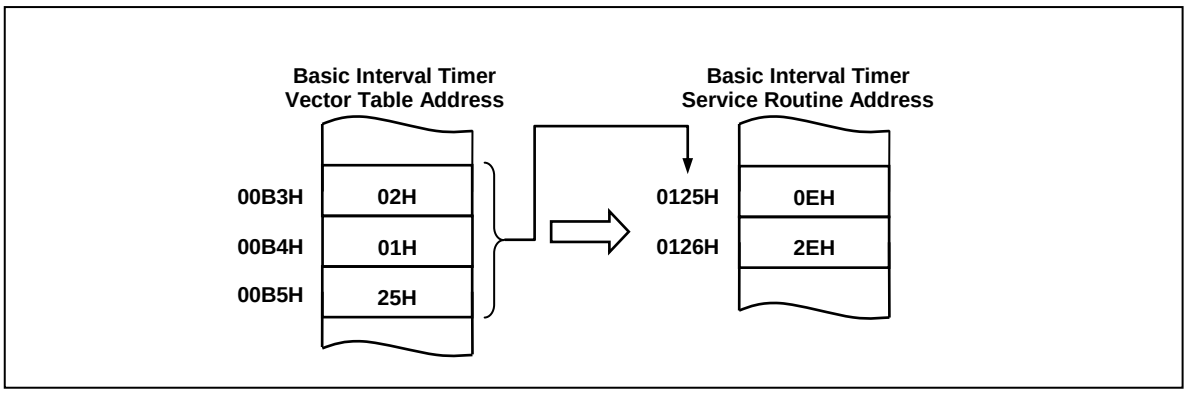
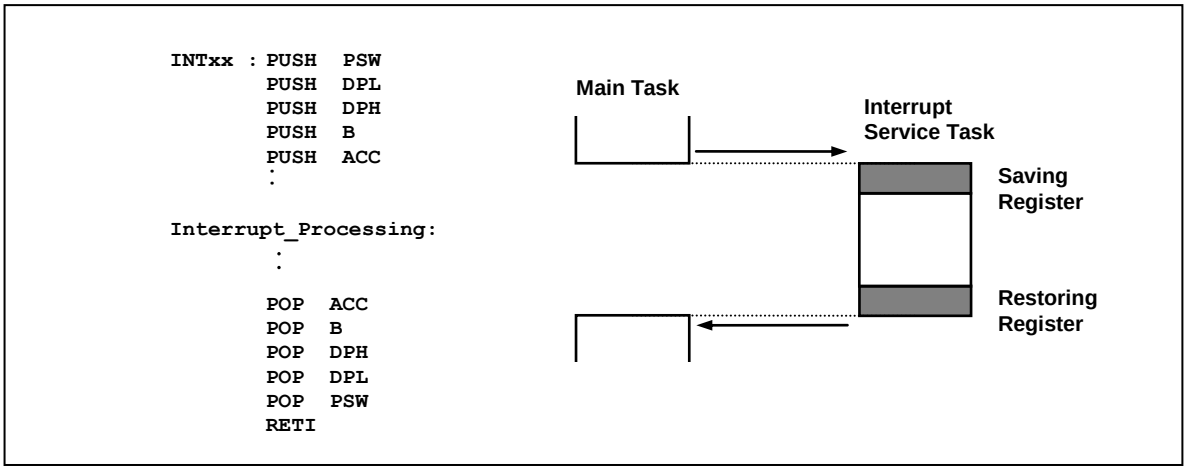


Figure 10.8 Correspondence between Vector Table Address and the Entry Address of ISR

10.10 Saving/Restore General-Purpose Registers



```
INTxx : PUSH   PSW
        PUSH   DPL
        PUSH   DPH
        PUSH   B
        PUSH   ACC
        :

Interrupt_Processing:
        :
        :

        POP   ACC
        POP   B
        POP   DPH
        POP   DPL
        POP   PSW
        RETI
```


Figure 10.9 Saving/Restore Process Diagram and Sample Source

10.11 Interrupt Timing

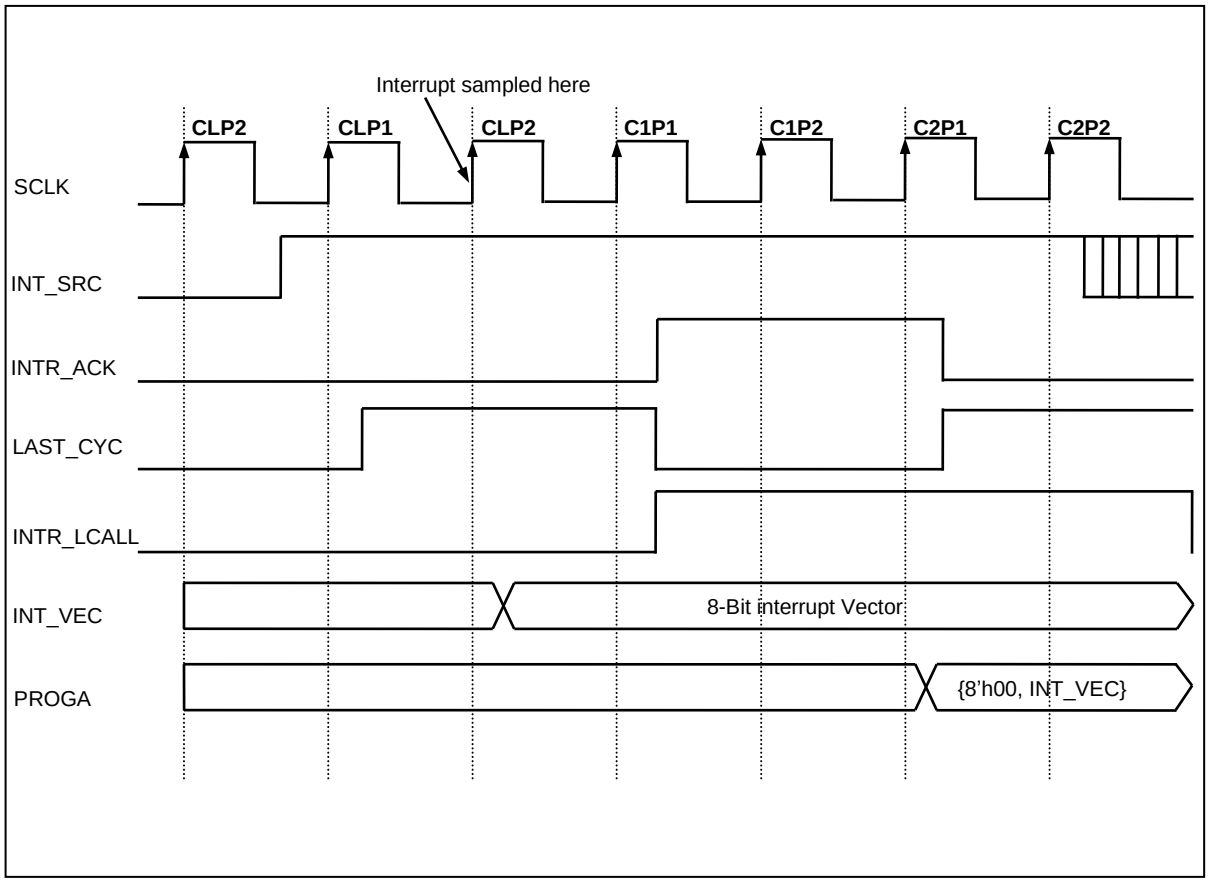


Figure 10.10 Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

Interrupt sources are sampled at the last cycle of a command. If an interrupt source is detected the lower 8-bit of interrupt vector (INT_VEC) is decided. M8051W core makes interrupt acknowledge at the first cycle of a command, and executes long call to jump to interrupt service routine.

NOTE) command cycle CLPx: L=Last cycle, 1=1st cycle or 1st phase, 2=2nd cycle or 2nd phase

10.12 Interrupt Register Overview

10.12.1 Interrupt Enable Register (IE, IE1, IE2, IE3)

Interrupt enable register consists of global interrupt control bit (EA) and peripheral interrupt control bits. Total 24 peripherals are able to control interrupt.

10.12.2 Interrupt Priority Register (IP0H/L, IP1H/L, IP2H/L, IP3H/L)

The 24 interrupts are divided into 6 groups which have each 4 interrupt sources. An individual interrupt can be assigned 4 levels interrupt priority using interrupt priority register. Level 3 is the highest priority, while level 0 is the lowest priority. After a reset IP0H/L, IP1H/L, IP2H/L, and IP3H/L, are cleared to '00H'. If interrupts have the same priority level, lower number interrupt is served first.

10.12.3 External Interrupt Flag Register (EIFLAG0, EIFLAG1, EIFLAG2)

The external interrupt flag 0 register (EIFLAG0), external interrupt flag 1 register (EIFLAG1) and external interrupt flag 2 register (EIFLAG2) are set to '1' when the external interrupt generating condition is satisfied. The flag is cleared when the interrupt service routine is executed. Alternatively, the flag can be cleared by writing '0' to it.

10.12.4 External Interrupt Polarity Register (EIPOL0H/L , EIPOL1H/L, EIPOL2H/L)

The external interrupt polarity 0 high/low register (EIPOL0H/L), external interrupt polarity 1 high/low register (EIPOL1H/L) and external interrupt polarity 2 high/low register (EIPOL2H/L) determines which type of rising/falling/both edge interrupt. Initially, default value is no interrupt at any edge.

10.12.5 Register Map

Table 10-2 Interrupt Register Map

Name	Address	Dir	Default	Description
IE	A8H	R/W	00H	Interrupt Enable Register
IE1	A9H	R/W	00H	Interrupt Enable Register 1
IE2	AAH	R/W	00H	Interrupt Enable Register 2
IE3	ABH	R/W	00H	Interrupt Enable Register 3
IP0L	92H	R/W	00H	Interrupt Priority 0 Low Register
IP0H	93H	R/W	00H	Interrupt Priority 0 High Register
IP1L	9AH	R/W	00H	Interrupt Priority 1 Low Register
IP1H	9BH	R/W	00H	Interrupt Priority 1 High Register
IP2L	9CH	R/W	00H	Interrupt Priority 2 Low Register
IP2H	9DH	R/W	00H	Interrupt Priority 2 High Register
IP3L	9EH	R/W	00H	Interrupt Priority 3 Low Register
IP3H	9FH	R/W	00H	Interrupt Priority 3 High Register
EIFLAG0	A3H	R/W	00H	External Interrupt Flag 0 Register
EIPOL0L	A6H	R/W	00H	External Interrupt Polarity 0 Low Register
EIPOL0H	A7H	R/W	00H	External Interrupt Polarity 0 High Register
EIFLAG1	A4H	R/W	00H	External Interrupt Flag 1 Register
EIPOL1L	ACH	R/W	00H	External Interrupt Polarity 1 Low Register
EIPOL1H	ADH	R/W	00H	External Interrupt Polarity 1 High Register
EIFLAG2	A5H	R/W	00H	External Interrupt Flag 2 Register
EIPOL2L	AEH	R/W	00H	External Interrupt Polarity 2 Low Register
EIPOL2H	AFH	R/W	00H	External Interrupt Polarity 2 High Register

10.13 Interrupt Register Description

The interrupt register is used for controlling interrupt functions. Also it has external interrupt control registers. The interrupt register consists of interrupt enable register (IE), interrupt enable register 1 (IE1), interrupt enable register 2 (IE2) and interrupt enable register 3 (IE3). For external interrupt, it consists of external interrupt flag 0 register (EIFLAG0), external interrupt polarity 0 high/low register (EIPOL0H/L), external interrupt flag 1 register (EIFLAG1), external interrupt polarity 1 high/low register (EIPOL1H/L), external interrupt flag 2 register (EIFLAG2) and external interrupt polarity 2 high/low register (EIPOL2H/L).

10.13.1 Register Description for Interrupt

IE (Interrupt Enable Register) : A8H

7	6	5	4	3	2	1	0
EA	–	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EA Enable or Disable All Interrupt bits
- 0 All Interrupt disable
- 1 All Interrupt enable

INT5E Enable or Disable External Interrupt 0 ~ 7 (EINT0 ~ EINT7)
- 0 Disable
- 1 Enable

INT4E Enable or Disable SPI2 Interrupt
- 0 Disable
- 1 Enable

INT3E Enable or Disable UART3 Rx Interrupt
- 0 Disable
- 1 Enable

INT2E Enable or Disable UART2 Rx Interrupt
- 0 Disable
- 1 Enable

INT1E Enable or Disable USI1 Rx Interrupt
- 0 Disable
- 1 Enable

INT0E Enable or Disable USI0 Rx Interrupt
- 0 Disable
- 1 Enable

IE1 (Interrupt Enable Register 1): A9H

7	6	5	4	3	2	1	0
–	–	INT11E	INT10E	INT9E	INT8E	INT7E	INT6E
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

INT11E Enable or Disable External interrupt 10 ~ 17 (EINT10 ~ EINT17)
- 0 Disable
- 1 Enable

INT10E Enable or Disable SPI3 interrupt
- 0 Disable
- 1 Enable

INT9E Enable or Disable UART3 Tx interrupt
- 0 Disable
- 1 Enable

INT8E Enable or Disable UART2 Tx interrupt
- 0 Disable
- 1 Enable

INT7E Enable or Disable USI1 Tx/I2C interrupt)
- 0 Disable
- 1 Enable

INT6E Enable or Disable USI0 Tx/I2C interrupt
- 0 Disable
- 1 Enable

IE2 (Interrupt Enable Register 2) : AAH

7	6	5	4	3	2	1	0
–	–	INT17E	INT16E	INT15E	INT14E	INT13E	INT12E
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT17E Enable or Disable Timer 0/1/2/3 match interrupt
- 0 Disable
- 1 Enable

INT16E Enable or Disable External interrupt 8 ~ A/18 ~ 19 (EINT8 ~ EINTA/EINT18 ~ EINT19)
- 0 Disable
- 1 Enable

INT15E Enable or Disable Timer 4 match interrupt
- 0 Disable
- 1 Enable

INT14E Enable or Disable Timer 5 match interrupt
- 0 Disable
- 1 Enable

INT13E Enable or Disable Timer 6/7/8/9 match interrupt
- 0 Disable
- 1 Enable

INT12E Enable or Disable UART4 Rx interrupt
- 0 Disable
- 1 Enable

IE3 (Interrupt Enable Register 3) : ABH

7	6	5	4	3	2	1	0
–	–	INT23E	INT22E	INT21E	INT20E	INT19E	INT18E
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

INT23E Enable or Disable Timer 0/1/2/3 overflow interrupt
- 0 Disable
- 1 Enable

INT22E Enable or Disable BIT Interrupt
- 0 Disable
- 1 Enable

INT21E Enable or Disable WDT Interrupt
- 0 Disable
- 1 Enable

INT20E Enable or Disable WT and SP overflow interrupt
- 0 Disable
- 1 Enable

INT19E Enable or Disable ADC Interrupt
- 0 Disable
- 1 Enable

INT18E Enable or Disable UART4 Tx interrupt
- 0 Disable
- 1 Enable

IP0L (Interrupt Priority 0 Low Register) : 92H

7	6	5	4	3	2	1	0
–	–	IP0L5	IP0L4	IP0L3	IP0L2	IP0L1	IP0L0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP0H (Interrupt Priority 0 High Register) : 93H

7	6	5	4	3	2	1	0
–	–	IP0H5	IP0H4	IP0H3	IP0H2	IP0H1	IP0H0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP0L[5:0], IP0H[5:0] Select IE Interrupt Priority

IP0Hx	IP0Lx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

IP1L (Interrupt Priority 1 Low Register) : 9AH

7	6	5	4	3	2	1	0
–	–	IP1L5	IP1L4	IP1L3	IP1L2	IP1L1	IP1L0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP1H (Interrupt Priority 1 High Register) : 9BH

7	6	5	4	3	2	1	0
–	–	IP1H5	IP1H4	IP1H3	IP1H2	IP1H1	IP1H0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP1L[5:0], IP1H[5:0] Select IE1 Interrupt Priority

IP1Hx	IP1Lx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

IP2L (Interrupt Priority 2 Low Register) : 9CH

7	6	5	4	3	2	1	0
–	–	IP2L5	IP2L4	IP2L3	IP2L2	IP2L1	IP2L0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP2H (Interrupt Priority 2 High Register) : 9DH

7	6	5	4	3	2	1	0
–	–	IP2H5	IP2H4	IP2H3	IP2H2	IP2H1	IP2H0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP2L[5:0], IP2H[5:0]] Select IE2 Interrupt Priority

IP2Hx	IP2Lx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

IP3L (Interrupt Priority 3 Low Register) : 9EH

7	6	5	4	3	2	1	0
–	–	IP3L5	IP3L4	IP3L3	IP3L2	IP3L1	IP3L0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP3H (Interrupt Priority 3 High Register) : 9FH

7	6	5	4	3	2	1	0
–	–	IP3H5	IP3H4	IP3H3	IP3H2	IP3H1	IP3H0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

IP3L[5:0], IP3H[5:0]] Select IE3 Interrupt Priority

IP3Hx	IP3Lx	Description
0	0	level 0 (lowest)
0	1	level 1
1	0	level 2
1	1	level 3 (highest)

EIFLAG0 (External Interrupt Flag 0 Register) : A3H

7	6	5	4	3	2	1	0
FLAG7	FLAG6	FLAG5	FLAG4	FLAG3	FLAG2	FLAG1	FLAG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EIFLAG0[6:5] When an External Interrupt 0 ~ 7 is occurred, the flag becomes '1'. The flag is cleared by writing '0' to the bit. Writing "1" has no effect. So, the flag should be cleared by software.

0 External Interrupt 0 ~ 7 not occurred

1 External Interrupt 0 ~ 7 occurred

EIPOL0H (External Interrupt Polarity 0 High Register): A7H

7	6	5	4	3	2	1	0
POL7		POL6		POL5		POL4	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL0H[7:0] External interrupt (EINT7, EINT6, EINT5, EINT4) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n =4, 5, 6 and 7

EIPOL0L (External Interrupt Polarity 0 Low Register): A6H

7	6	5	4	3	2	1	0
POL3		POL2		POL1		POL0	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL0L[7:0] External interrupt (EINT3, EINT2, EINT1, EINT0) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n =0, 1, 2 and 3

EIFLAG1 (External Interrupt Flag 1 Register) : A4H

7	6	5	4	3	2	1	0
FLAG17	FLAG16	FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EIFLAG1[7:0] When an External Interrupt 10 ~ 17 is occurred, the flag becomes '1'. The flag is cleared by writing '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 External Interrupt 10 ~ 17 not occurred

1 External Interrupt 10 ~ 17 occurred

EIPOL1H (External Interrupt Polarity 1 High Register): ADH

7	6	5	4	3	2	1	0
POL17		POL16		POL15		POL14	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL1H[7:0] External interrupt (EINT14,EINT15,EINT16,EINT17) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n =14, 15, 16 and 17

EIPOL1L (External Interrupt Polarity 1 Low Register): ACH

7	6	5	4	3	2	1	0
POL13		POL12		POL11		POL10	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL1L[7:0] External interrupt (EINT10,EINT11,EINT12,EINT13) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n =10, 11, 12 and 13

EIFLAG2(External Interrupt Flag 2 Register) : A5H

7	6	5	4	3	2	1	0
–	–	–	FLAG19	FLAG18	FLAGA	FLAG9	FLAG8
–	–	–	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

EIFLAG2[4:0] When an External Interrupt 8 ~ A/18 ~ 19 is occurred, the flag becomes '1'. The flag is cleared by writing '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 External Interrupt not occurred

1 External Interrupt occurred

EIPOL2H (External Interrupt Polarity 2 High Register): AFH

7	6	5	4	3	2	1	0
–	–	–	–	POL19		POL18	
–	–	–	–	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL2H[3:0] External interrupt (EINT18,EINT19) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n = 18 and 19

EIPOL2L (External Interrupt Polarity 2 Low Register): AEH

7	6	5	4	3	2	1	0
–	–	POLA		POL9		POL8	
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00H

EIPOL2L[5:0] External interrupt (EINT8,EINT9,EINTA) polarity selection

POLn[1:0]		Description
0	0	No interrupt at any edge
0	1	Interrupt on rising edge
1	0	Interrupt on falling edge
1	1	Interrupt on both of rising and falling edge

Where n = 8, 9 and A

11. Peripheral Hardware

11.1 Clock Generator

11.1.1 Overview

As shown in Figure 11.1, the clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and the peripheral hardware. It contains main/sub-frequency clock oscillator. The main/sub clock operation can be easily obtained by attaching a crystal between the XIN/SXIN and XOUT/SXOUT pin, respectively. The main/sub clock can be also obtained from the external oscillator. In this case, it is necessary to put the external clock signal into the XIN/SXIN pin and open the XOUT/SXOUT pin. The default system clock is 1MHz INT-RC Oscillator and the default division rate is sixteen. In order to stabilize system internally, it is used 1MHz INT-RC oscillator on POR.

- Calibrated Internal RC Oscillator (16 MHz)
  . INT-RC OSC/1 (16 MHz)
  . INT-RC OSC/2 (8 MHz)
  . INT-RC OSC/4 (4 MHz)
  . INT-RC OSC/8 (2 MHz)
  . INT-RC OSC/16 (1 MHz, Default system clock)
  . INT-RC OSC/32 (0.5 MHz)
- Main Crystal Oscillator (0.4~16 MHz)
- Sub Crystal Oscillator (32.768 kHz)
- Internal WDTRC Oscillator (5 kHz)

11.1.2 Block Diagram

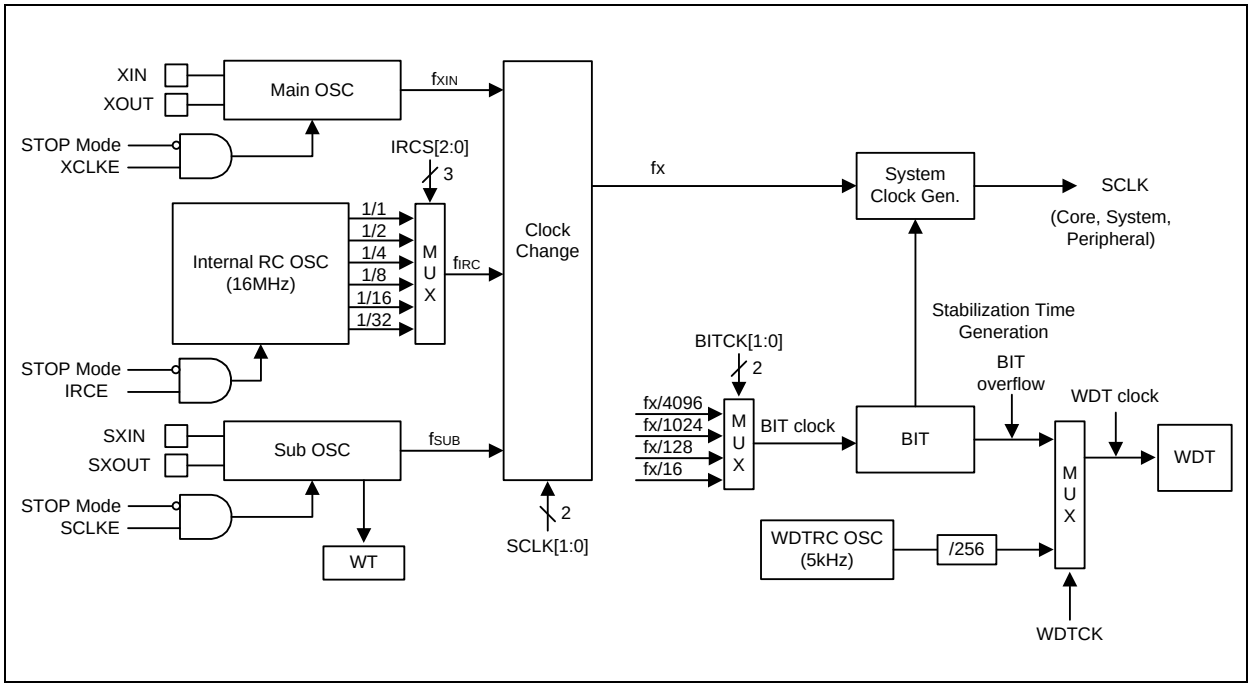


Figure 11.1 Clock Generator Block Diagram

11.1.3 Register Map

Table 11-1 Clock Generator Register Map

Name	Address	Dir	Default	Description
SCCR	8AH	R/W	00H	System and Clock Control Register
OSCCR	C8H	R/W	08H	Oscillator Control Register

11.1.4 Clock Generator Register Description

The clock generator register uses clock control for system operation. The clock generation consists of System and clock control register and oscillator control register.

11.1.5 Register Description for Clock Generator

SCCR (System and Clock Control Register) : 8AH

7	6	5	4	3	2	1	0
–	–	–	–	–	–	SCLK1	SCLK0
–	–	–	–	–	–	R/W	R/W

Initial value : 00H

SCLK [1:0] System Clock Selection Bit

SCLK1	SCLK0	Description
0	0	INT RC OSC (f_{IRC}) for system clock
0	1	External Main OSC (f_{XIN}) for system clock
1	0	External Sub OSC (f_{SUB}) for system clock
1	1	Not used

OSCCR (Oscillator Control Register) : C8H

7	6	5	4	3	2	1	0
–	–	IRCS2	IRCS1	IRCS0	IRCE	XCLKE	SCLKE
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 08H

IRCS[2:0] Internal RC Oscillator Post-divider Selection

IRCS2	IRCS1	IRCS0	Description
0	0	0	INT-RC/32 (0.5MHz)
0	0	1	INT-RC/16 (1MHz)
0	1	0	INT-RC/8 (2MHz)
0	1	1	INT-RC/4 (4MHz)
1	0	0	INT-RC/2 (8MHz)
1	0	1	INT-RC/1 (16MHz)
Other values			Not used

IRCE Control the Operation of the Internal RC Oscillator

0 Enable operation of INT-RC OSC

1 Disable operation of INT-RC OSC

XCLKE Control the Operation of the External Main Oscillator

0 Disable operation of X-TAL

1 Enable operation of X-TAL

SCLKE Control the Operation of the External Sub Oscillator

0 Disable operation of SX-TAL

1 Enable operation of SX-TAL

11.2 Basic Interval Timer

11.2.1 Overview

The MC97F2664 has one 8-bit basic interval timer that is free-run and can't stop. Block diagram is shown in Figure 11.2. In addition, the basic interval timer generates the time base for watchdog timer counting. It also provides a basic interval timer interrupt (BITIFR).

The MC97F2664 has these basic interval timer (BIT) features:

- During Power On, BIT gives a stable clock generation time
- On exiting Stop mode, BIT gives a stable clock generation time
- As timer function, timer interrupt occurrence

11.2.2 Block Diagram

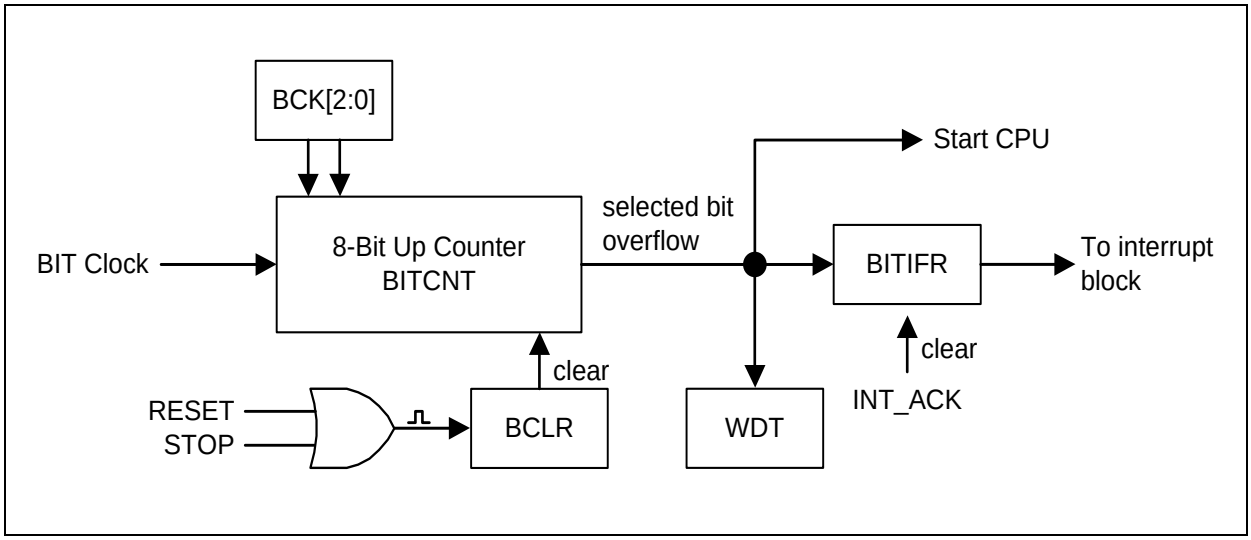


Figure 11.2 Basic Interval Timer Block Diagram

11.2.3 Register Map

Table 11-2 Basic Interval Timer Register Map

Name	Address	Dir	Default	Description
BITCNT	8CH	R	00H	Basic Interval Timer Counter Register
BITCR	8BH	R/W	01H	Basic Interval Timer Control Register

11.2.4 Basic Interval Timer Register Description

The basic interval timer register consists of basic interval timer counter register (BITCNT) and basic interval timer control register (BITCR). If BCLR bit is set to '1', BITCNT becomes '0' and then counts up. After 1 machine cycle, BCLR bit is cleared to '0' automatically.

11.2.5 Register Description for Basic Interval Timer

BITCNT (Basic Interval Timer Counter Register) : 8CH

7	6	5	4	3	2	1	0
BITCNT7	BITCNT6	BITCNT5	BITCNT4	BITCNT3	BITCNT2	BITCNT1	BITCNT0
R	R	R	R	R	R	R	R

Initial value : 00H

BITCNT[7:0] BIT Counter

BITCR (Basic Interval Timer Control Register) : 8BH

7	6	5	4	3	2	1	0
BITIFR	BITCK1	BITCK0	–	BCLR	BCK2	BCK1	BCK0
R/W	R/W	R/W	–	R/W	R/W	R/W	R/W

Initial value : 01H

BITIFR When BIT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.

0 BIT interrupt no generation

1 BIT interrupt generation

BITCK[1:0] Select BIT clock source

BITCK1	BITCK0	Description
0	0	fx/4096
0	1	fx/1024
1	0	fx/128
1	1	fx/16

BCLR If this bit is written to '1', BIT Counter is cleared to '0'

0 Free Running

1 Clear Counter

BCK[2:0] Select BIT overflow period

BCK2	BCK1	BCK0	Description
0	0	0	Bit 0 overflow (BIT Clock * 2)
0	0	1	Bit 1 overflow (BIT Clock * 4) (default)
0	1	0	Bit 2 overflow (BIT Clock * 8)
0	1	1	Bit 3 overflow (BIT Clock * 16)
1	0	0	Bit 4 overflow (BIT Clock * 32)
1	0	1	Bit 5 overflow (BIT Clock * 64)
1	1	0	Bit 6 overflow (BIT Clock * 128)
1	1	1	Bit 7 overflow (BIT Clock * 256)

11.3 Watch Dog Timer

11.3.1 Overview

The watchdog timer rapidly detects the CPU malfunction such as endless looping caused by noise or something like that, and resumes the CPU to the normal state. The watchdog timer signal for malfunction detection can be used as either a CPU reset or an interrupt request. When the watchdog timer is not being used for malfunction detection, it can be used as a timer to generate an interrupt at fixed intervals. It is possible to use free running 8-bit timer mode (WDTRSON='0') or watch dog timer mode (WDTRSON='1') as setting WDTCR[6] bit. If WDTCR[5] is written to '1', WDT counter value is cleared and counts up. After 1 machine cycle, this bit is cleared to '0' automatically. The watchdog timer consists of 8-bit binary counter and the watchdog timer data register. When the value of 8-bit binary counter is equal to the 8 bits of WDTCNT, the interrupt request flag is generated. This can be used as Watchdog timer interrupt or reset of CPU in accordance with the bit WDTRSON.

The input clock source of watch dog timer is the BIT overflow. The interval of watchdog timer interrupt is decided by BIT overflow period and WDTDR set value. The equation can be described as

WDT Interrupt Interval = (BIT Interrupt Interval) X (WDTDR Value+1)

11.3.2 WDT Interrupt Timing Waveform

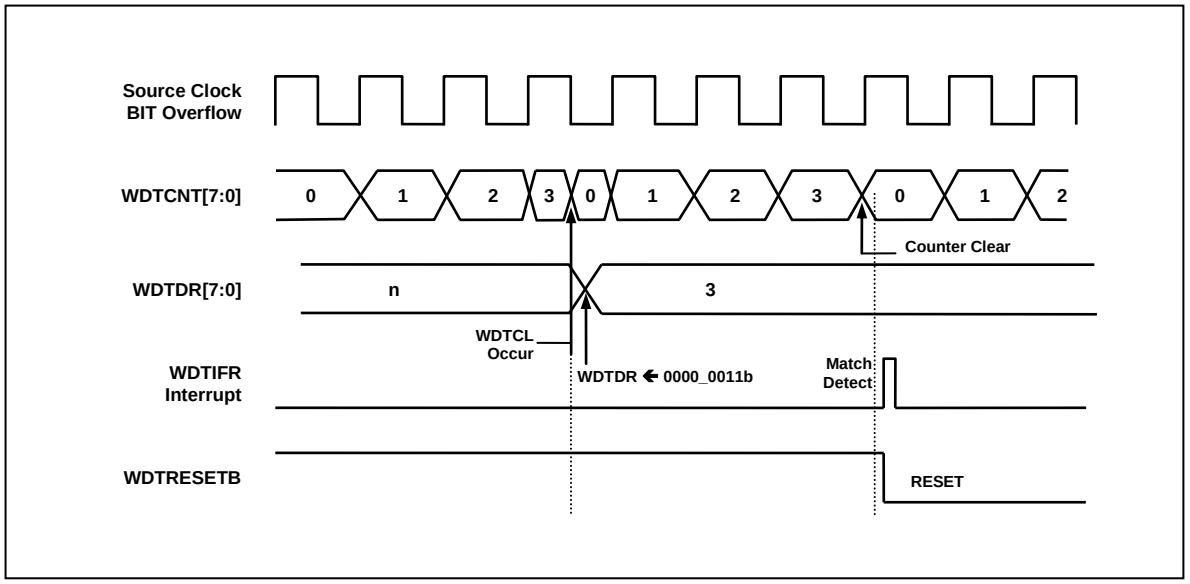


Figure 11.3 Watch Dog Timer Interrupt Timing Waveform

11.3.3 Block Diagram

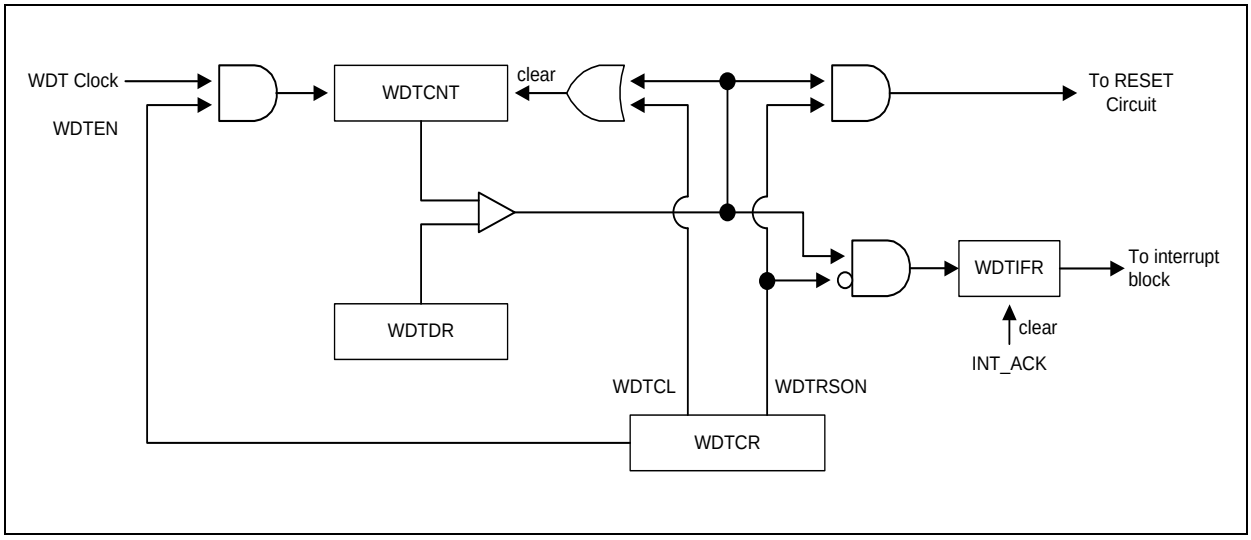


Figure 11.4 Watch Dog Timer Block Diagram

11.3.4 Register Map

Table 11-3 Watch Dog Timer Register Map

Name	Address	Dir	Default	Description
WDTCNT	8EH	R	00H	Watch Dog Timer Counter Register
WDTDR	8EH	W	FFH	Watch Dog Timer Data Register
WDTCR	8DH	R/W	00H	Watch Dog Timer Control Register

11.3.5 Watch Dog Timer Register Description

The watch dog timer register consists of watch dog timer counter register (WDTCNT), watch dog timer data register (WDTDR) and watch dog timer control register (WDTCR).

11.3.6 Register Description for Watch Dog Timer

WDTCNT (Watch Dog Timer Counter Register: Read Case) : 8EH

7	6	5	4	3	2	1	0
WDTCNT7	WDTCNT6	WDTCNT5	WDTCNT4	WDTCNT3	WDTCNT2	WDTCNT1	WDTCNT0
R	R	R	R	R	R	R	R

Initial value : 00H

WDTCNT[7:0] WDT Counter

WDTDR (Watch Dog Timer Data Register: Write Case) : 8EH

7	6	5	4	3	2	1	0
WDTDR7	WDTDR6	WDTDR5	WDTDR4	WDTDR3	WDTDR2	WDTDR1	WDTDR0
W	W	W	W	W	W	W	W

Initial value : FFH

WDTDR[7:0] Set a period
WDT Interrupt Interval=(BIT Interrupt Interval) x(WDTDR Value+1)
NOTE) Do not write “0” in the WDTDR register.

WDTCR (Watch Dog Timer Control Register) : 8DH

7	6	5	4	3	2	1	0
WDTEN	WDTRSON	WDTCL	–	–	–	WDTCK	WDTIFR
R/W	R/W	R/W	–	–	–	R/W	R/W

Initial value : 00H

WDTEN Control WDT Operation
0 Disable
1 Enable

WDTRSON Control WDT RESET Operation
0 Free Running 8-bit timer
1 Watch Dog Timer RESET ON

WDTCL Clear WDT Counter
0 Free Run
1 Clear WDT Counter (auto clear after 1 Cycle)

WDTCK Control WDT Clock Selection Bit
0 BIT overflow for WDT clock (WDTRC disable)
1 WDTRC for WDT clock (WDTRC enable)

WDTIFR When WDT Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal. Writing “1” has no effect.
0 WDT Interrupt no generation
1 WDT Interrupt generation

11.4 Watch Timer

11.4.1 Overview

The watch timer has the function for RTC (Real Time Clock) operation. It is generally used for RTC design. The internal structure of the watch timer consists of the clock source select circuit, timer counter circuit, output select circuit, and watch timer control register. To operate the watch timer, determine the input clock source, output interval, and set WTEN to '1' in watch timer control register (WTCR). It is able to execute simultaneously or individually. To stop or reset WT, clear the WTEN bit in WTCR register. Even if CPU is STOP mode, sub clock is able to be so alive that WT can continue the operation. The watch timer counter circuits may be composed of 21-bit counter which contains low 14-bit with binary counter and high 7-bit counter in order to raise resolution. In WTDR, it can control WT clear and set interval value at write time, and it can read 7-bit WT counter value at read time.

11.4.2 Block Diagram

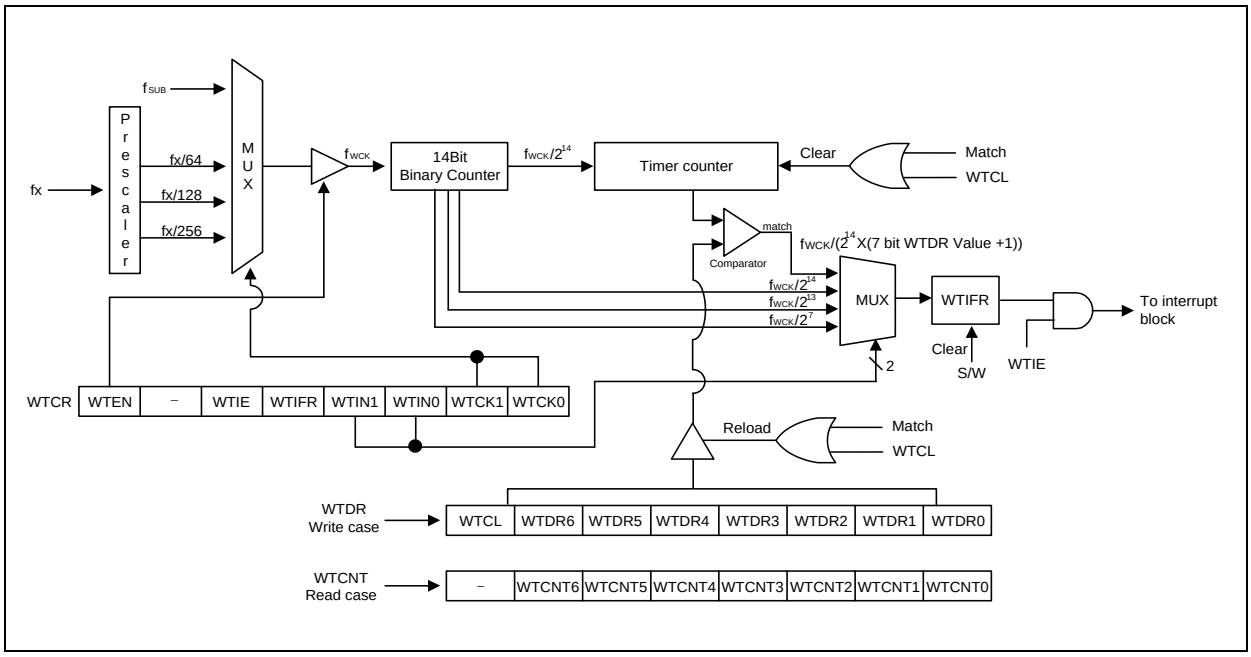


Figure 11.5 Watch Timer Block Diagram

11.4.3 Register Map

Table 11-4 Watch Timer Register Map

Name	Address	Dir	Default	Description
WTCNT	89H	R	00H	Watch Timer Counter Register
WTDR	89H	W	7FH	Watch Timer Data Register
WTCR	8FH	R/W	00H	Watch Timer Control Register

11.4.4 Watch Timer Register Description

The watch timer register consists of watch timer counter register (WTCNT), watch timer data register (WTDR), and watch timer control register (WTCR). As WTCR is 6-bit writable/readable register, WTCR can control the clock source (WTCK[1:0]), interrupt interval (WTIN[1:0]), and function enable/disable (WTEN). Also there is WT interrupt flag bit (WTIFR).

11.4.5 Register Description for Watch Timer

WTCNT (Watch Timer Counter Register: Read Case) : 89H

7	6	5	4	3	2	1	0
–	WTCNT6	WTCNT5	WTCNT4	WTCNT3	WTCNT2	WTCNT1	WTCNT0
–	R	R	R	R	R	R	R

Initial value : 00H

WTCNT[6:0] WT Counter

WTDR (Watch Timer Data Register: Write Case) : 89H

7	6	5	4	3	2	1	0
WTCL	WTDR6	WTDR5	WTDR4	WTDR3	WTDR2	WTDR1	WTDR0
R/W	W	W	W	W	W	W	W

Initial value : 7FH

WTCL Clear WT Counter

0 Free Run

1 Clear WT Counter (auto clear after 1 Cycle)

WTDR[6:0] Set WT period

WT Interrupt Interval=fwck/(2^14 x(7bit WTDR Value+1))

NOTE) Do not write “0” in the WTDR register.

WTCR (Watch Timer Control Register) : 8FH

7	6	5	4	3	2	1	0
WTEN	–	WTIE	WTIFR	WTIN1	WTIN0	WTCK1	WTCK0
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

WTEN Control Watch Timer

0 Disable

1 Enable

WTIE Enable or Disable Watch Timer Interrupt

0 Disable

1 Enable

WTIFR When WT Interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or automatically clear by software Writing "1" has no effect.

0 WT Interrupt no generation

1 WT Interrupt generation

WTIN[1:0] Determine interrupt interval

WTIN1	WTIN0	Description
0	0	f_{WCK}/2^7
0	1	f_{WCK}/2^13
1	0	f_{WCK}/2^14
1	1	f_{WCK}/(2^14 x (7bit WTDR Value+1))

WTCK[1:0] Determine Source Clock

WTCK1	WTCK0	Description
0	0	f_{SUB}
0	1	f_X/256
1	0	f_X/128
1	1	f_X/64

NOTE) f_X – System clock frequency (Where fx= 4.19MHz)

f_{SUB} – Sub clock oscillator frequency (32.768kHz)

f_{WCK} – Selected Watch timer clock

11.5 Timer 0/1/2/3

11.5.1 Overview

The 8-bit timer 0/1/2/3 consists of multiplexer, timer 0/1/2/3 counter register, timer 0/1/2/3 data register, timer 0/1/2/3 capture data register, and timer 0/1/2/3 control register (TnCNT, TnDR, TnCDR, TnCR).

It has three operating modes:

- 8-bit timer/counter mode
- 8-bit PWM output mode
- 8-bit capture mode

The timer/counter 0/1/2/3 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).

- TIMER 0/1/2/3 clock source: f_X/2, 4, 8, 32, 128, 512, 2048 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnCDR). In timer/counter mode, whenever counter value is equal to TnDR, TnO port toggles. Also the timer 0/1/2/3 outputs PWM waveform through PWMnO port in the PWM mode.

Table 11-5 Timer 0/1/2 Operating Modes

TnEN	TnMS[1:0]	TnCK[2:0]	Timer n
1	00	XXX	8 Bit Timer/Counter Mode
1	01	XXX	8 Bit PWM Mode
1	1X	XXX	8 Bit Capture Mode

11.5.2 8-Bit Timer/Counter Mode

The 8-bit timer/counter mode is selected by control register as shown in Figure 11.6.

The 8-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 0/1/2/3 can use the input clock with one of 2, 4, 8, 32, 128, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNT and TnDR is identical in timer 0/1/2/3, a match signal is generated and the interrupt of Timer n occurs. TnCNT value is automatically cleared by match signal. It can be also cleared by software (TnCC).

The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], EC0/EC1/EC2/EC3 port should be set to the input port by P54IO/P55IO/P56IO/P57IO bit.

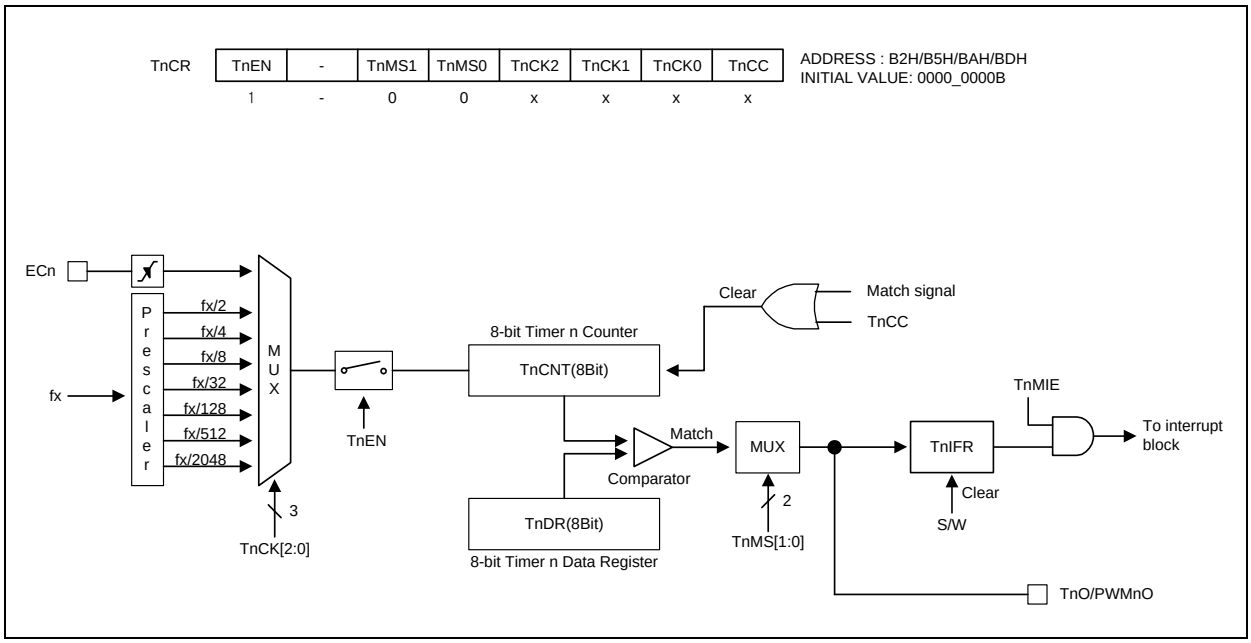


Figure 11.6 8-Bit Timer/Counter Mode for Timer 0/1/2/3 (Where n = 0, 1, 2, and 3)

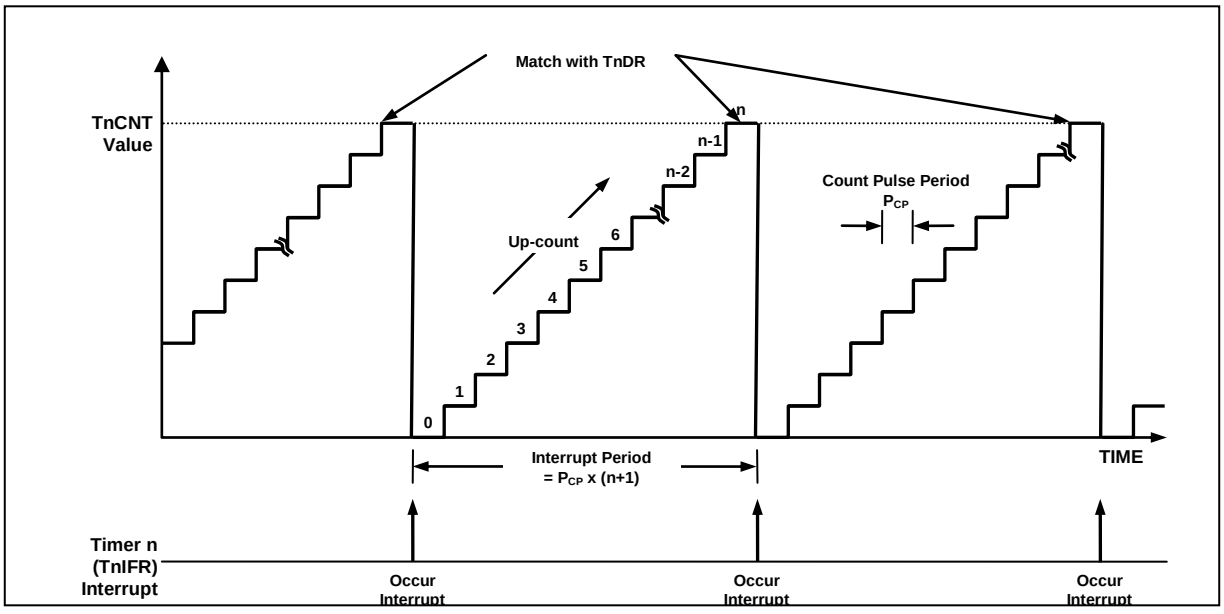


Figure 11.7 8-Bit Timer/Counter 0/1/2/3 Example (Where n = 0, 1, 2, and 3)

11.5.3 8-Bit PWM Mode

The timer 0/1/2/3 has a high speed PWM (Pulse Width Modulation) function. In PWM mode, TnO/PWMnO pin outputs up to 8-bit resolution PWM output. This pin should be configured as a PWM output by setting the T0O/PWM0O, T1O/PWM1O,T2O/PWM2O and T3O/PWM3O function by P5FSR0, P5FSR1, P5FSR2, and P5FSR3 bit. In the 8-bit timer/counter mode, a match signal is generated when the counter value is identical to the value of TnDR. When the value of TnCNT and TnDR is identical in timer n, a match signal is generated and the interrupt of timer 0/1/2/3 occurs. In PWM mode, the match signal does not clear the counter. Instead, it runs continuously, overflowing at “FFH”, and then continues incrementing from “00H”. The timer 0/1/2/3 overflow interrupt is generated whenever a counter overflow occurs. TnCNT value is cleared by software (TnCC) bit.

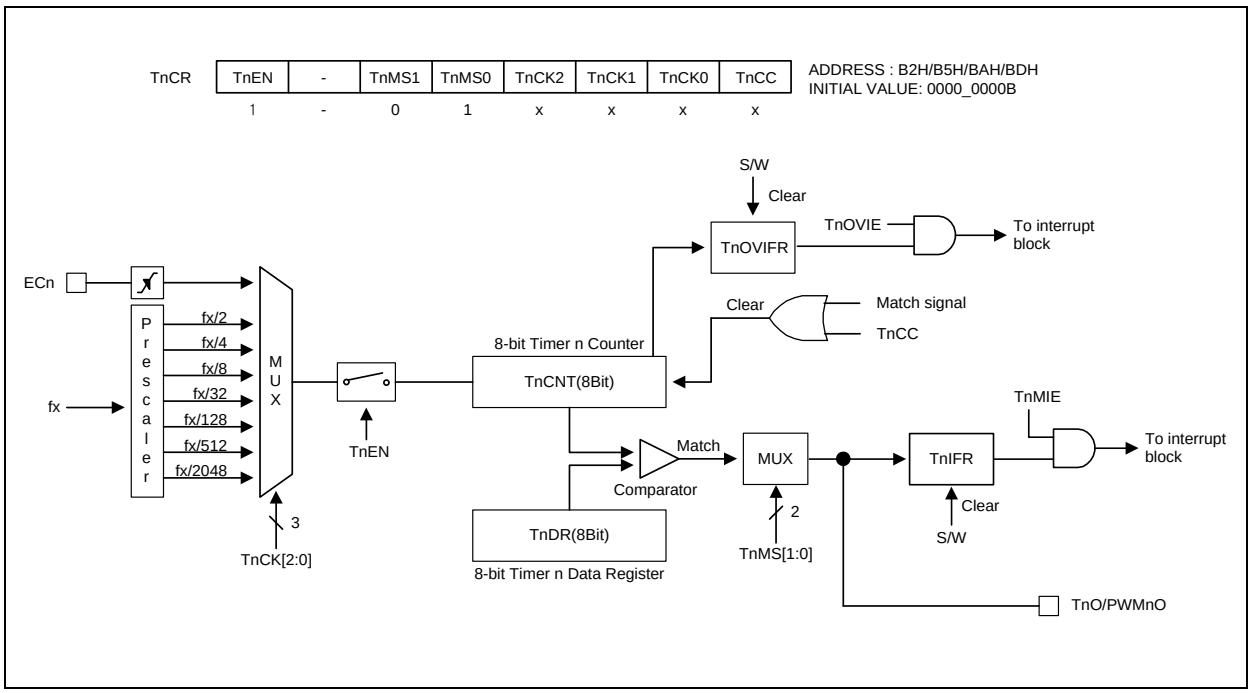


Figure 11.8 8-Bit PWM Mode for Timer 0/1/2/3 (Where n = 0, 1, 2, and 3)

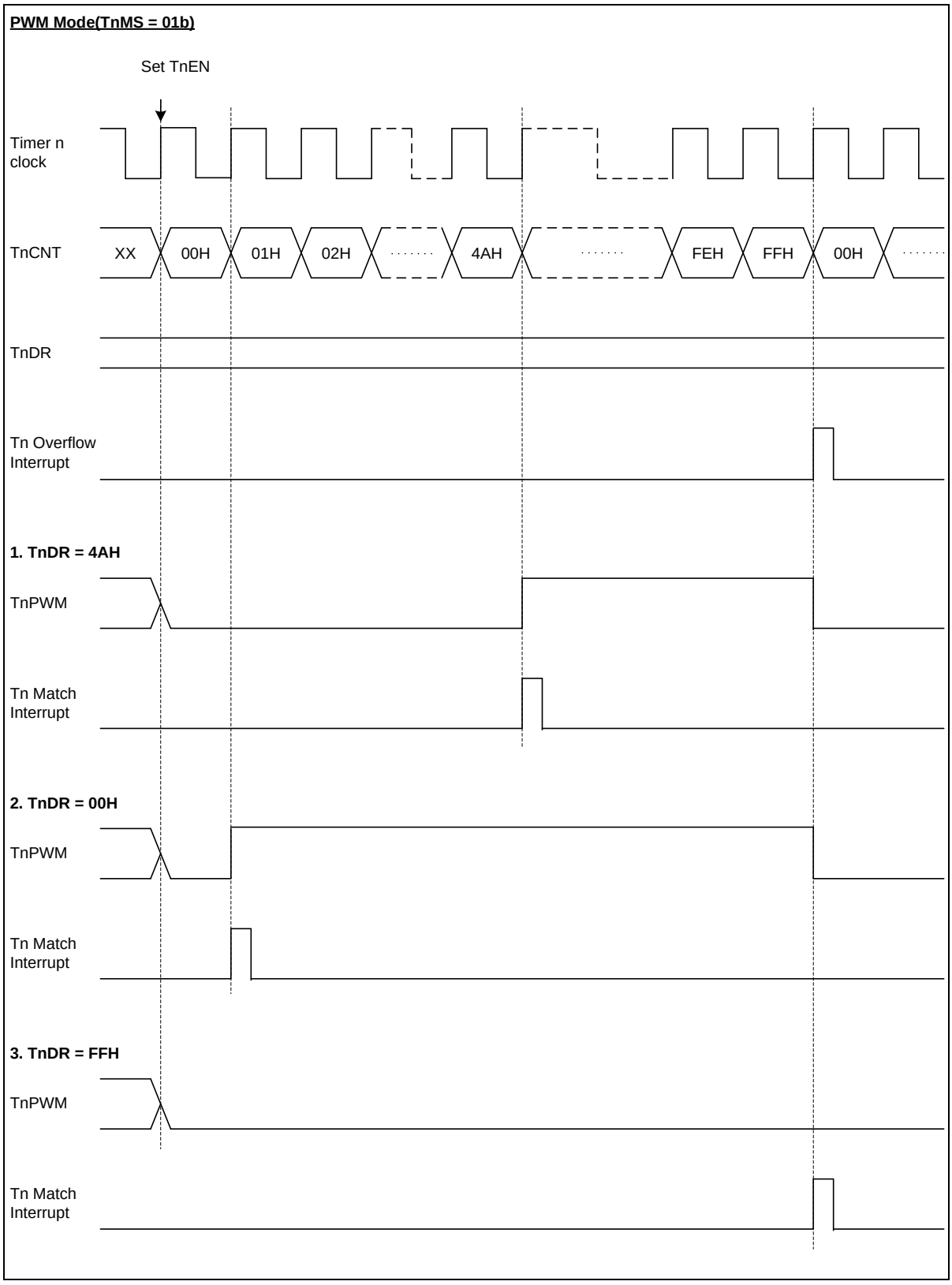


Figure 11.9 PWM Output Waveforms in PWM Mode for Timer 0/1/2 (Where n = 0, 1, 2, and 3)

11.5.4 8-Bit Capture Mode

The timer 0/1/2/3 capture mode is set by TnMS[1:0] as '1x'. The clock source can use the internal/external clock. Basically, it has the same function as the 8-bit timer/counter mode and the interrupt occurs when TnCNT is equal to TnDR. TnCNT value is automatically cleared by match signal and it can be also cleared by software (TnCC).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into TnCDR. In the timer n capture mode, timer n output (TnO) waveform is not available.

According to EIPOL1H/L registers setting, the external interrupt EINT1n function is chosen. Of course, the EINT1n pin must be set to an input port.

TnCDR and TnDR are in the same address. In the capture mode, reading operation reads TnCDR, not TnDR and writing operation will update TnDR.

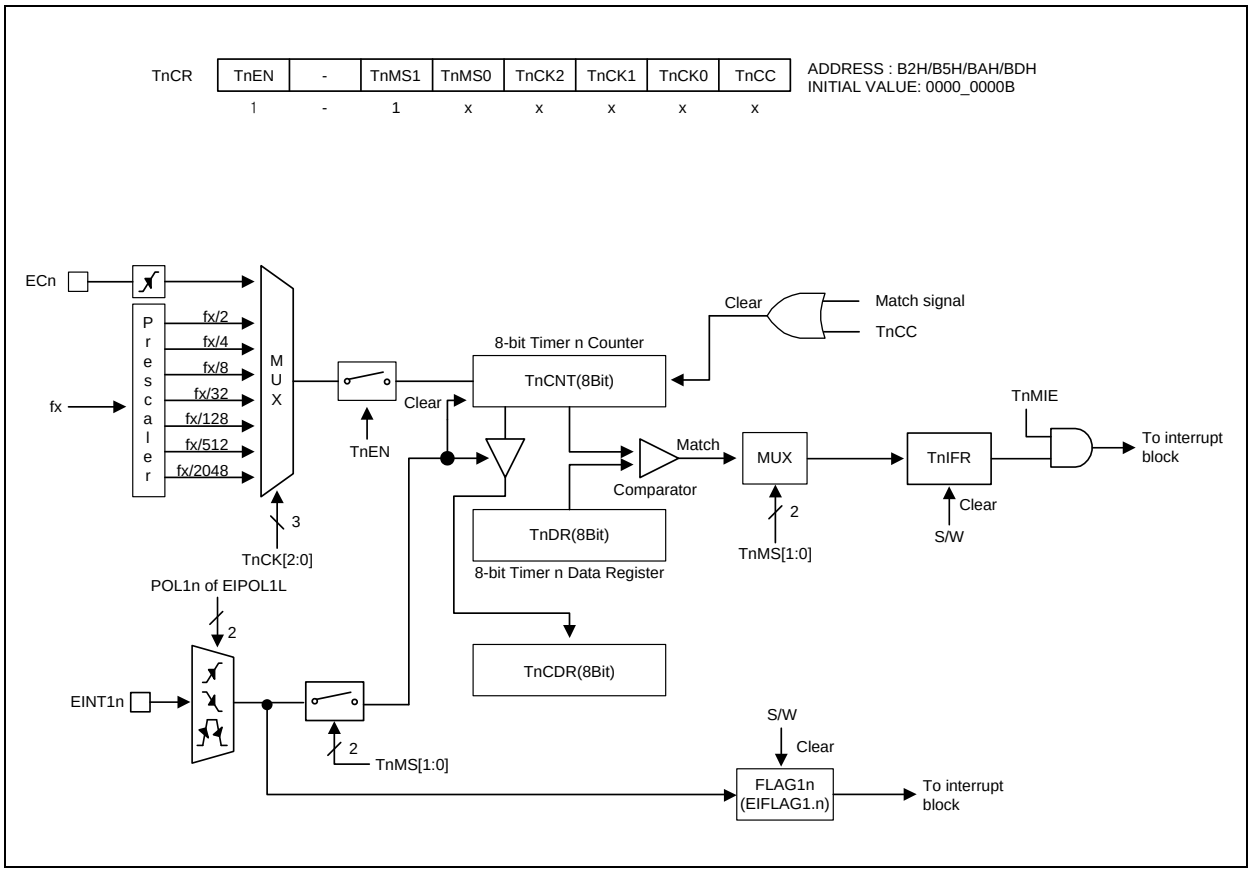


Figure 11.10 8-Bit Capture Mode for Timer 0/1/2/3 (Where n = 0, 1, 2, and 3)

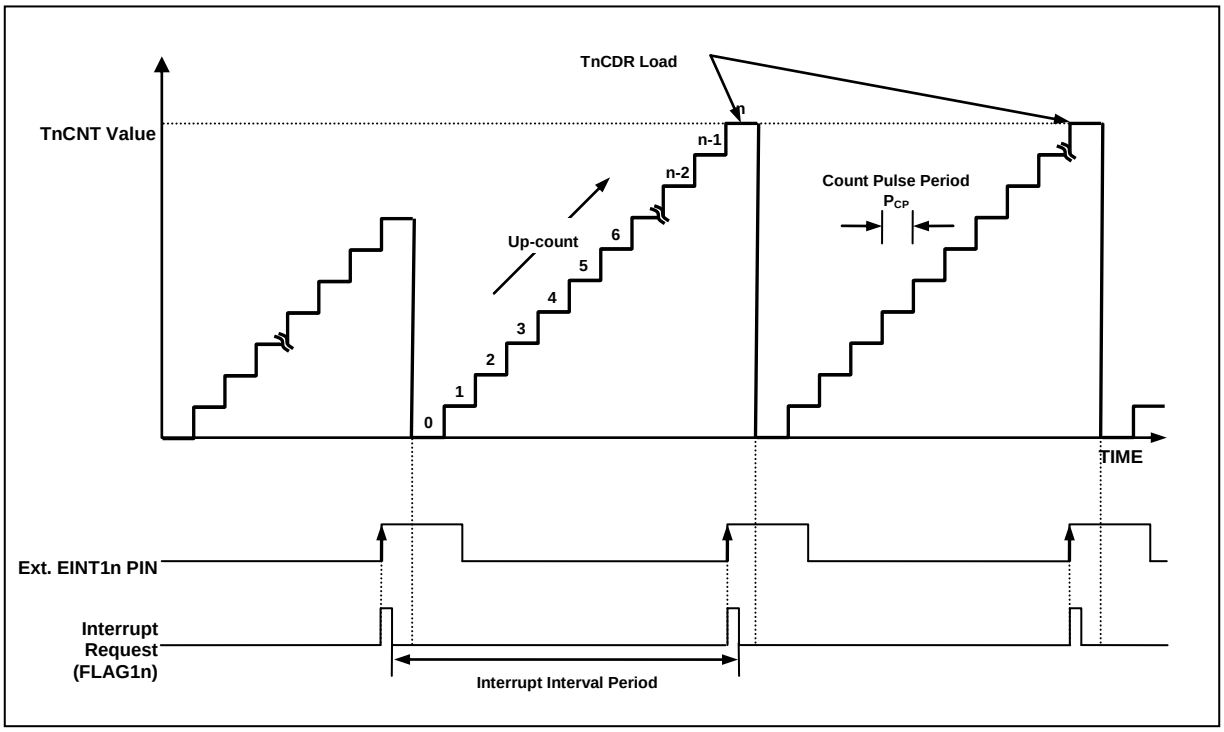


Figure 11.11 Input Capture Mode Operation for Timer 0/1/2/3 (Where n = 0, 1, 2, and 3)

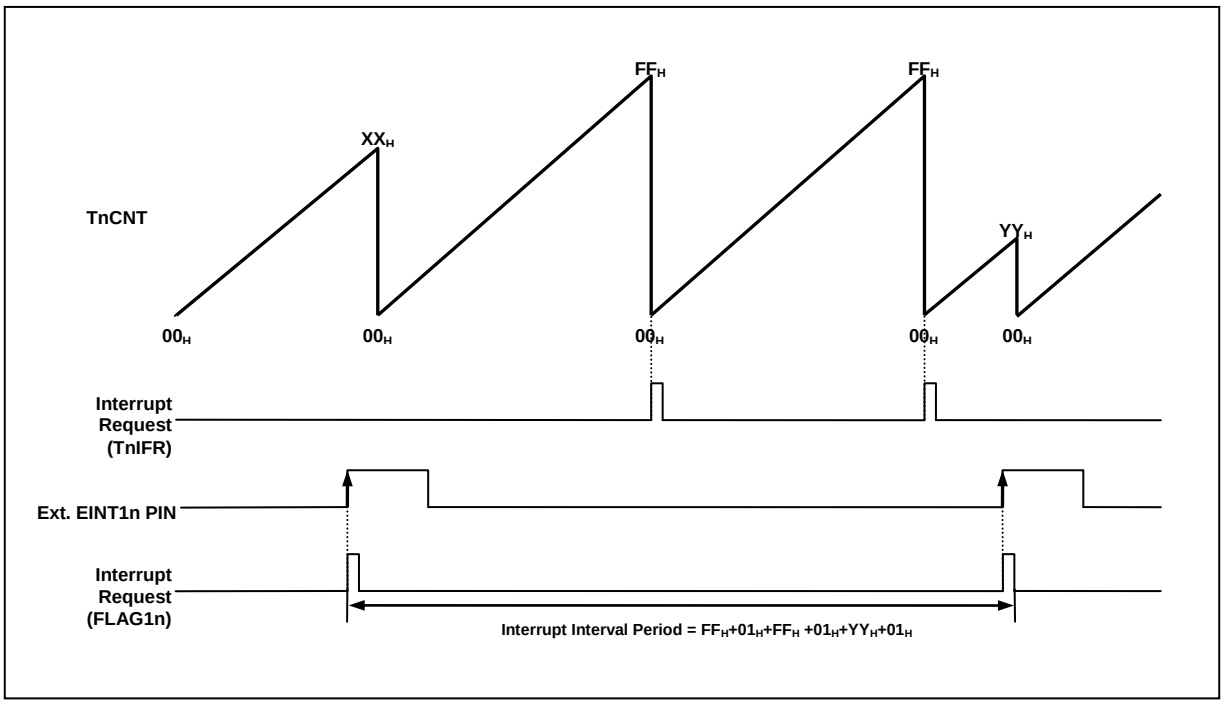


Figure 11.12 Express Timer Overflow in Capture Mode (Where n = 0, 1, 2, and 3)

11.5.5 Block Diagram

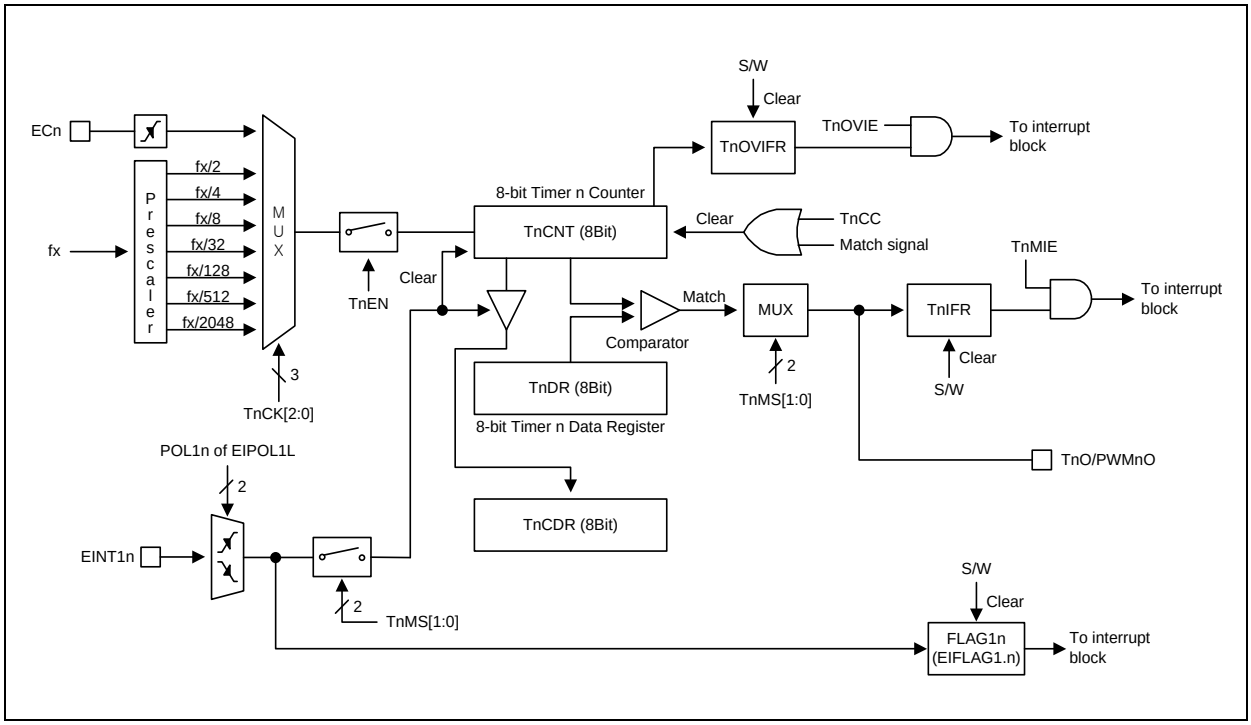


Figure 11.13 8-Bit Timer 0/1/2/3 Block Diagram (Where n = 0, 1, 2, and 3)

11.5.6 Register Map

Table 11-6 Timer 0/1/2/3 Register Map

Name	Address	Dir	Default	Description
TnCNT	B3H/B6H/BBH/BEH	R	00H	Timer n Counter Register
TnDR	B4H/B7H/BCH/BFH	R/W	FFH	Timer n Data Register
TnCDR	B4H/B7H/BCH/BFH	R	00H	Timer n Capture Data Register
TnCR	B2H/B5H/BAH/BDH	R/W	00H	Timer n Control Register
TINTCR	95H	R/W	00H	Timer Interrupt Control Register
TIFLAG0	96H	R/W	00H	Timer Interrupt Flag Register

11.5.6.1 Timer/Counter 0/1/2/3 Register Description

The timer/counter 0/1/2/3 register consists of timer 0/1/2/3 counter register (TnCNT), timer 0/1/2/3 data register (TnDR), timer 0/1/2/3 capture data register (TnCDR), timer 0/1/2/3 control register (TnCR), timer interrupt control register(TINTCR), and timer interrupt flag register(TIFLAG0).

11.5.6.2 Register Description for Timer/Counter 0/1/2/3

TnCNT (Timer n Counter Register) : B3H/B6H/BBH/BEH, n=0, 1, 2, and 3

7	6	5	4	3	2	1	0
TnCNT7	TnCNT6	TnCNT5	TnCNT4	TnCNT3	TnCNT2	TnCNT1	TnCNT0
R	R	R	R	R	R	R	R

Initial value : 00H

TnCNT[7:0] Tn Counter

TnDR (Timer n Data Register : Write only when it is capture mode) : B4H/B7H/BCH/BFH, n=0, 1, 2, and 3

7	6	5	4	3	2	1	0
TnDR7	TnDR6	TnDR5	TnDR4	TnDR3	TnDR2	TnDR1	TnDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnDR[7:0] Tn Data

TnCDR (Timer n Capture Data Register : Capture mode only) : B4H/B7H/BCH/BFH, n=0, 1, 2, and 3

7	6	5	4	3	2	1	0
TnCDR7	TnCDR6	TnCDR5	TnCDR4	TnCDR3	TnCDR2	TnCDR1	TnCDR0
R	R	R	R	R	R	R	R

Initial value : 00H

TnCDR[7:0] Tn Capture Data

TnCR (Timer n Control Register) : B2H/B5H/BAH/BDH, n=0, 1, 2, and 3

7	6	5	4	3	2	1	0
TnEN	–	TnMS1	TnMS0	TnCK2	TnCK1	TnCK0	TnCC
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

TnEN Control Timer n

0 Timer n disable

1 Timer n enable

TnMS[1:0] Control Timer n Operation Mode

TnMS1	TnMS0	Description
0	0	Timer/counter mode
0	1	PWM mode
1	x	Capture mode

TnCK[2:0] Select Timer n clock source. fx is a system clock frequency

TnCK2	TnCK1	TnCK0	Description
0	0	0	fx/2
0	0	1	fx/4
0	1	0	fx/8
0	1	1	fx/32
1	0	0	fx/128
1	0	1	fx/512
1	1	0	fx/2048
1	1	1	External Clock (ECn)

TnCC Clear timer n Counter

0 No effect

1 Clear the Timer n counter (When write, automatically cleared “0” after being cleared counter)

NOTES) 1. Match Interrupt is generated in Capture mode.

2. Refer to the timer interrupt flag register (TIFLAG0) for the T0/1/2/3 interrupt flags.

TINTCR(Timer Interrupt Control Register) : 95H

7	6	5	4	3	2	1	0
T3MIE	T2MIE	T1MIE	T0MIE	T3OVIE	T2OVIE	T1OVIE	T0OVIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T3MIE Enable or Disable Timer 3Match Interrupt
0 Disable
1 Enable

T2MIE Enable or Disable Timer 2 Match Interrupt
0 Disable
1 Enable

T1MIE Enable or Disable Timer 1 Match Interrupt
0 Disable
1 Enable

T0MIE Enable or Disable Timer 0 Match Interrupt
0 Disable
1 Enable

T3OVIE Enable or Disable Timer 3 Overflow Interrupt
0 Disable
1 Enable

T2OVIE Enable or Disable Timer 2 Overflow Interrupt
0 Disable
1 Enable

T1OVIE Enable or Disable Timer 1 Overflow Interrupt
0 Disable
1 Enable

T0OVIE Enable or Disable Timer 0 Overflow Interrupt
0 Disable
1 Enable

TIFLAG0(Timer Interrupt Flag Register) : 96H

7	6	5	4	3	2	1	0
T3OVIFR	T3IFR	T2OVIFR	T2IFR	T1OVIFR	T1IFR	T0OVIFR	T0IFR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

T3OVIFR When T3 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T3 overflow interrupt no generation

1 T3 overflow interrupt generation

T3IFR When T3 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T3 interrupt no generation

1 T3 interrupt generation

T2OVIFR When T2 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T2 verflow interrupt no generation

1 T2 verflow interrupt generation

T2IFR When T2 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T2 interrupt no generation

1 T3 interrupt generation

T1OVIFR When T1 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T1 overflow interrupt no generation

1 T1 overflow interrupt generation

T1IFR When T1 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T1 interrupt no generation

1 T1 interrupt generation

T0OVIFR When T0 overflow interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T0 overflow interrupt no generation

1 T0 overflow interrupt generation

T0IFR When T0 match interrupt occurs, this bit becomes '1'. The flag is cleared only by writing a '0' to the bit. So, the flag should be cleared by software. Writing "1" has no effect.

0 T0 interrupt no generation

1 T0 interrupt generation

11.6 Timer 4/5

11.6.1.1 Overview

The 16-bit timer 4/5 consists of multiplexer, timer 4/5 A data register high/low, timer 4/5 B data register high/low and timer 4/5 control register high/low (TnADRH, TnADRL, TnBDRH, TnBDRL, TnCRH, TnCRL).

It has four operating modes:

- 16-bit timer/counter mode
- 16-bit capture mode
- 16-bit PPG output mode (one-shot mode)
- 16-bit PPG output mode (repeat mode)

The timer/counter 4/5 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).

- TIMER n clock source: f_X/1, 2, 4, 8, 64, 512, 2048 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnBDRH/TnBDRL). Timer 4/5 outputs the comparison result between counter and data register through TnO port in timer/counter mode. Also Timer 4/5 outputs PWM wave form through PWMnO port in the PPG mode.

Table 11-7 Timer 4/5 Operating Modes

TnEN	TnMS[1:0]	TnCK[2:0]	Timer 4/5
1	00	XXX	16 Bit Timer/Counter Mode
1	01	XXX	16 Bit Capture Mode
1	10	XXX	16 Bit PPG Mode (one-shot mode)
1	11	XXX	16 Bit PPG Mode (repeat mode)

11.6.2 16-Bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.14.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 4/5 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNTH, TnCNTL and the value of TnADRH, TnADRL are identical in Timer 4/5 respectively, a match signal is generated and the interrupt of Timer 4/5 occurs. The TnCNTH, TnCNTL value is automatically cleared by match signal. It can be also cleared by software (TnCC).

The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], ECn port should be set to the input port by P72IO, P73IO bit.

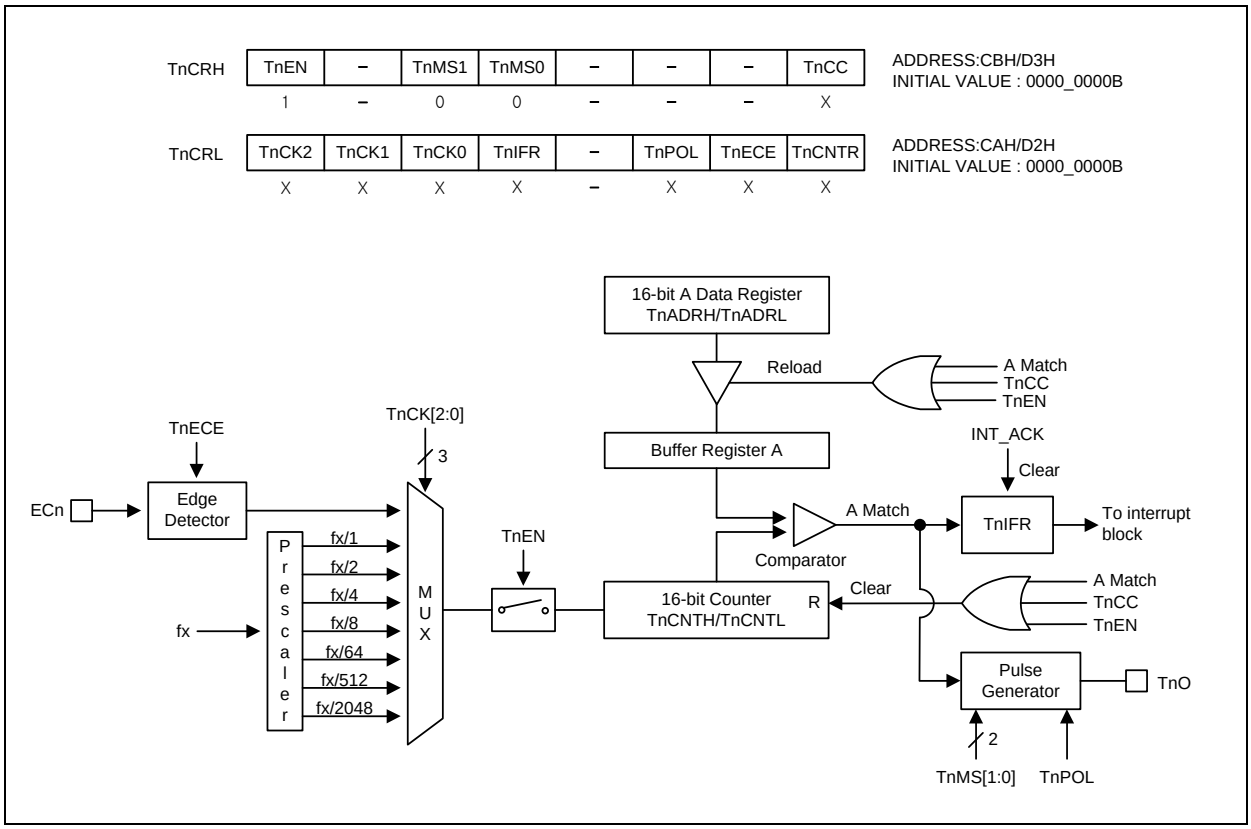


Figure 11.14 16-Bit Timer/Counter Mode for Timer 4/5 (where n= 4 and 5)

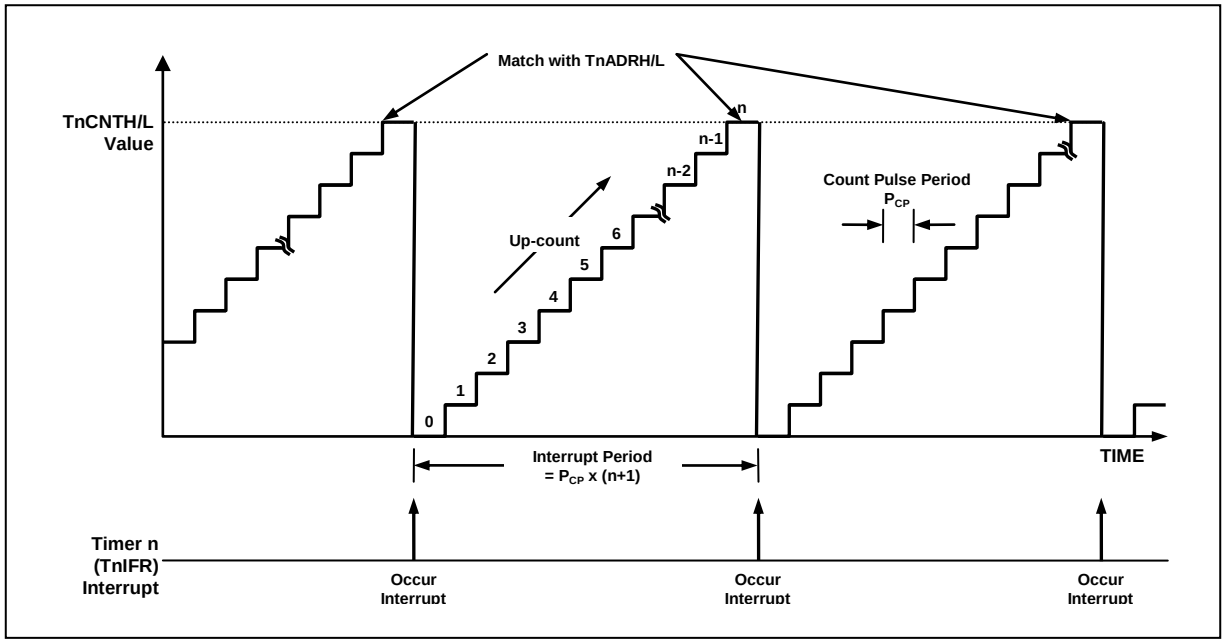


Figure 11.15 16-Bit Timer/Counter 4/5 Example (where n= 4 and 5)

11.6.3 16-Bit Capture Mode

The 16-bit timer 4/5 capture mode is set by TnMS[1:0] as '01'. The clock source can use the internal/external clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when TnCNTH/TnCNTL is equal to TnADRH/TnADRL. The TnCNTH, TnCNTL values are automatically cleared by match signal. It can be also cleared by software (TnCC).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into TnBDRH/TnBDRL.

According to EIPOL1H registers setting, the external interrupt EINT1n function is chosen. Of course, the EINT1n pin must be set as an input port.

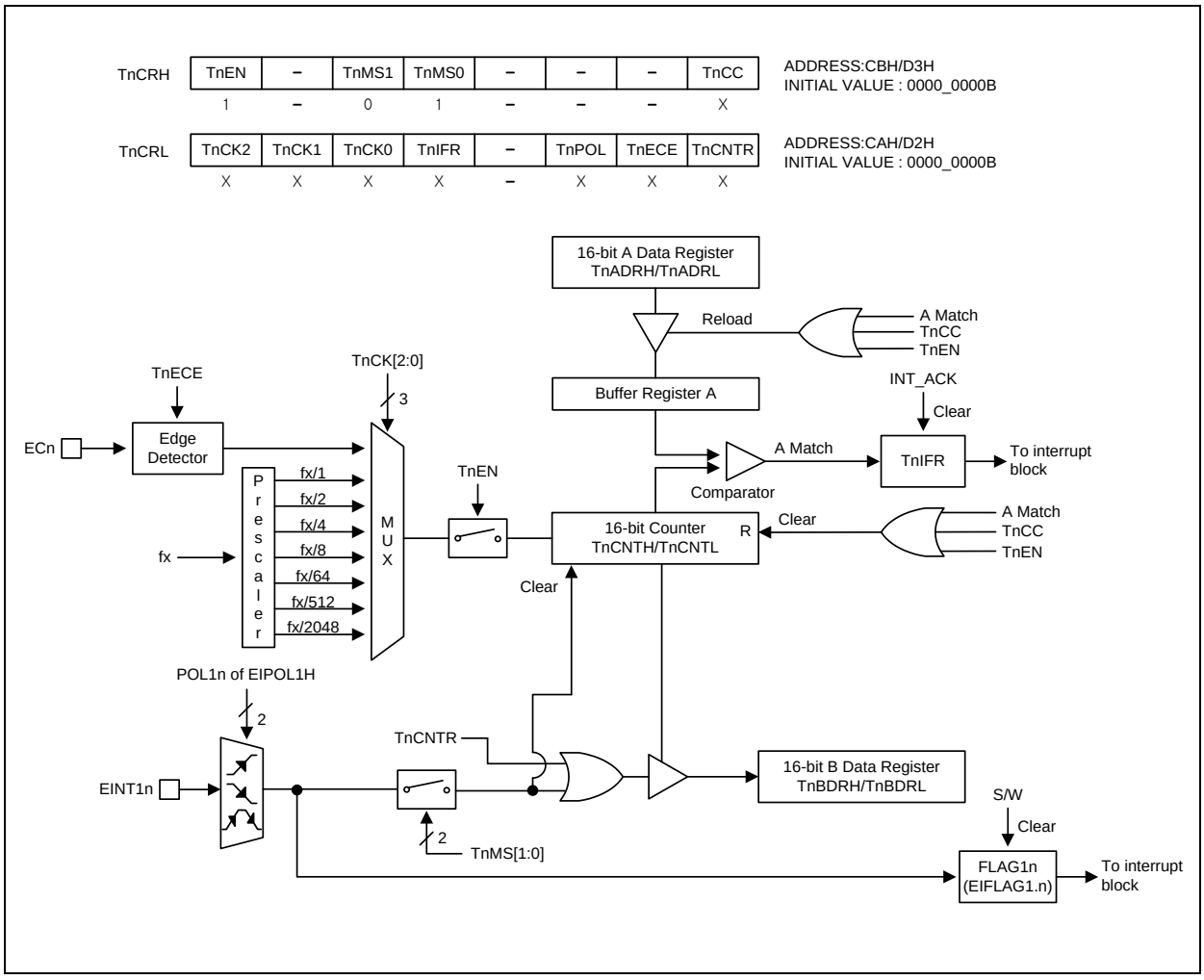


Figure 11.16 16-Bit Capture Mode for Timer 4/5 (where n= 4 and 5)

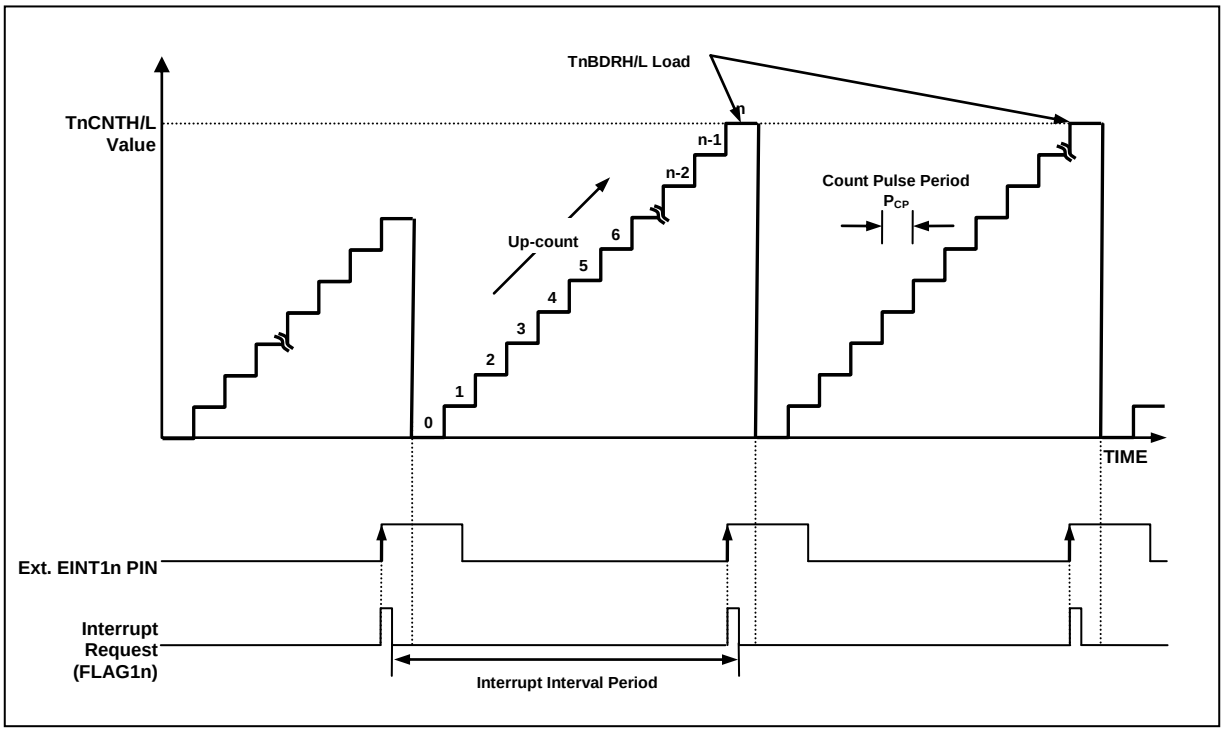


Figure 11.17 Input Capture Mode Operation for Timer 4/5 (where n= 4 and 5)

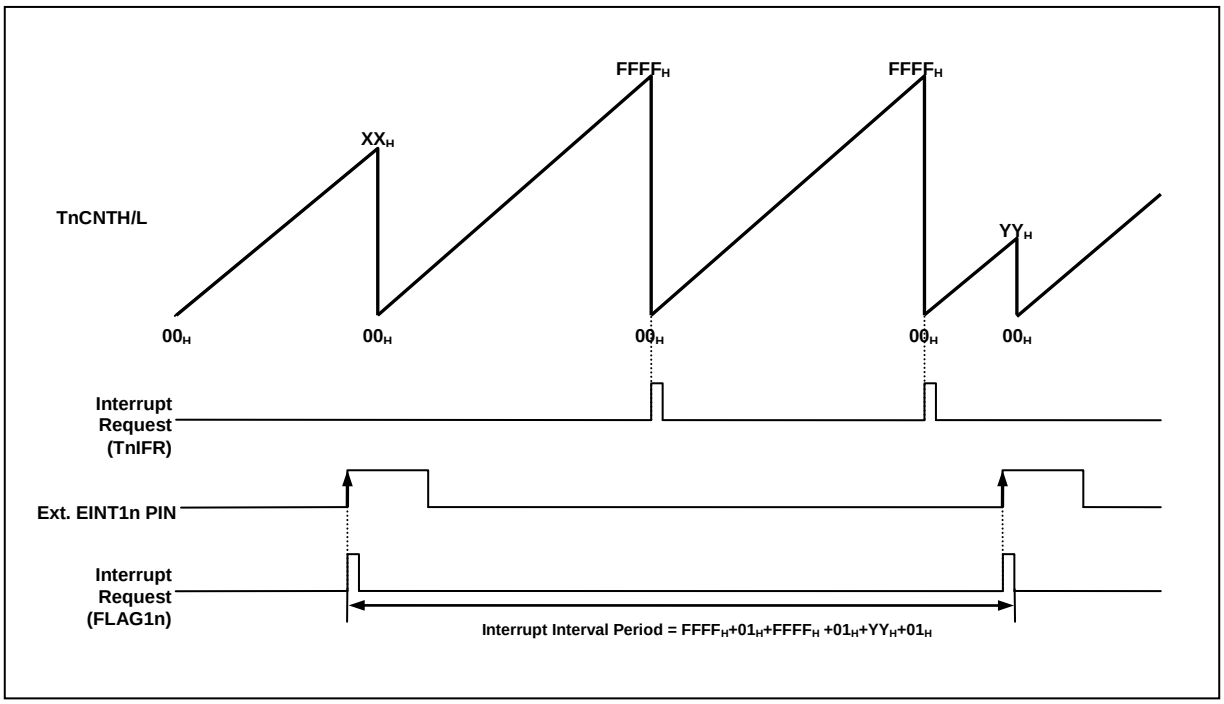
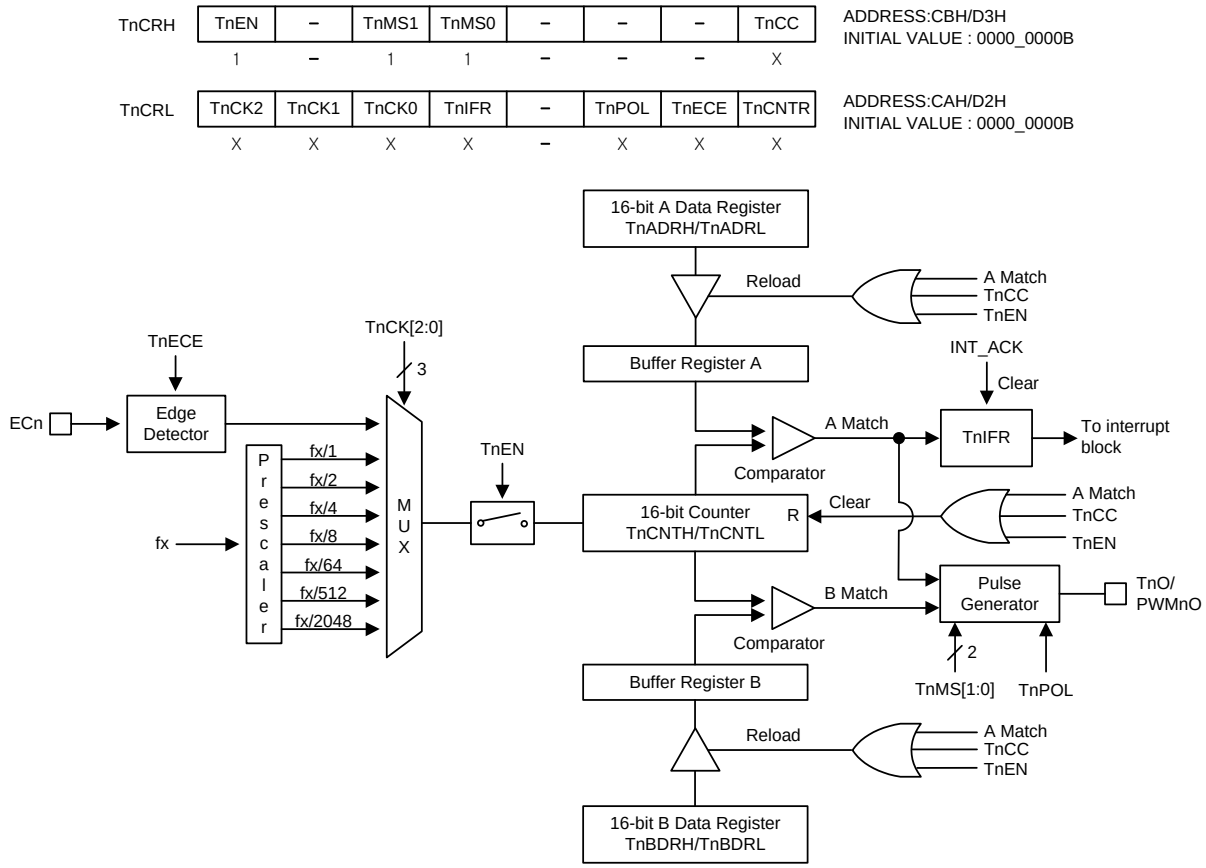


Figure 11.18 Express Timer Overflow in Capture Mode (where n= 4 and 5)

11.6.4 16-Bit PPG Mode

The timer 4/5 has a PPG (Programmable Pulse Generation) function. In PPG mode, TnO/PWMnO pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting P3FSR0,P3FSR1 to '1'. The period of the PWM output is determined by the TnADRH/TnADRL. And the duty of the PWM output is determined by the TnBDRH/TnBDRL.



NOTE) The TnEN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 11.19 16-Bit PPG Mode for Timer 4/5 (where n= 4 and 5)

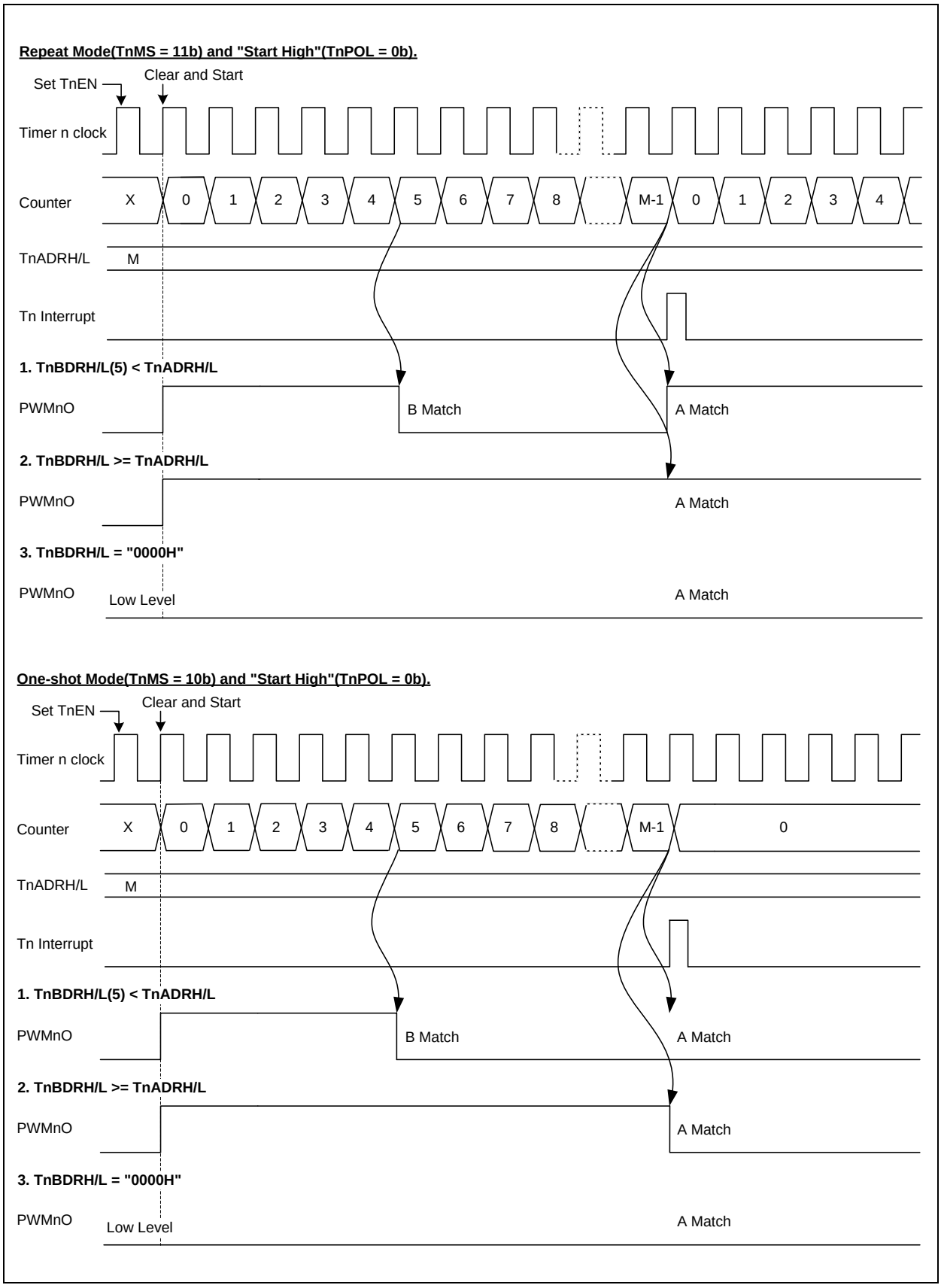


Figure 11.20 16-Bit PPG Mode Timming chart for Timer 4/5 (where n= 4 and 5)

11.6.5 Block Diagram

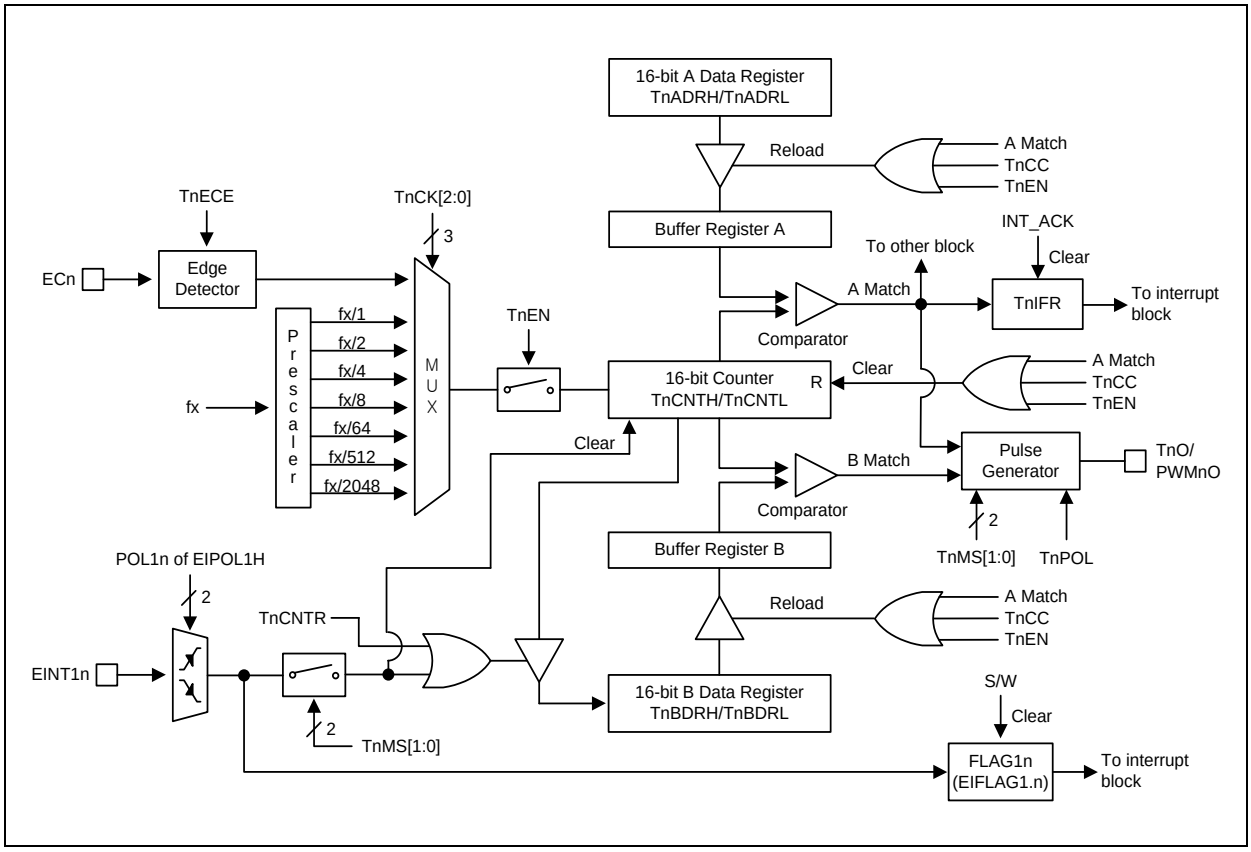


Figure 11.21 16-Bit Timer 4/5 Block Diagram (where n= 4 and 5)

11.6.6 Register Map

Table 11-8 Timer 2 Register Map

Name	Address	Dir	Default	Description
TnADRH	CDH/D5H	R/W	FFH	Timer n A Data High Register
TnADRL	CCH/D4H	R/W	FFH	Timer n A Data Low Register
TnBDRH	CFH/D7H	R/W	FFH	Timer n B Data High Register
TnBDRL	CEH/D6H	R/W	FFH	Timer n B Data Low Register
TnCRH	CBH/D3H	R/W	00H	Timer n Control High Register
TnCRL	CAH/D2H	R/W	00H	Timer n Control Low Register

11.6.6.1 Timer/Counter 4/5 Register Description

The timer/counter 4/5 register consists of timer 4/5 A data high register (TnADRH), timer 4/5 A data low register (TnADRL), timer 4/5 B data high register (TnBDRH), timer 4/5 B data low register (TnBDRL), timer 4/5 control high register (TnCRH) and timer 4/5 control low register (TnCRL).

11.6.6.2 Register Description for Timer/Counter 4/5

TnADRH (Timer n A data High Register) : CDH/D5H, n= 4 and 5

7	6	5	4	3	2	1	0
TnADRH7	TnADRH6	TnADRH5	TnADRH4	TnADRH3	TnADRH2	TnADRH1	TnADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnADRH[7:0] Tn A Data High Byte

TnADRL (Timer n A Data Low Register) : CCH/D4H, n= 4 and 5

7	6	5	4	3	2	1	0
TnADRL7	TnADRL6	TnADRL5	TnADRL4	TnADRL3	TnADRL2	TnADRL1	TnADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnADRL[7:0] Tn A Data Low Byte

NOTE) Do not write “0000H” in the TnADRH/TnADRL register when PPG mode

TnBDRH (Timer n B Data High Register) : CFH/D7H, n= 4 and 5

7	6	5	4	3	2	1	0
TnBDRH7	TnBDRH6	TnBDRH5	TnBDRH4	TnBDRH3	TnBDRH2	TnBDRH1	TnBDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnBDRH[7:0] Tn B Data High Byte

TnBDRL (Timer n B Data Low Register) : CEH/D6H, n= 4 and 5

7	6	5	4	3	2	1	0
TnBDRL7	TnBDRL6	TnBDRL5	TnBDRL4	TnBDRL3	TnBDRL2	TnBDRL1	TnBDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnBDRL[7:0] Tn B Data Low Byte

TnCRH (Timer n Control High Register) : CBH/D3H, n= 4 and 5

7	6	5	4	3	2	1	0
TnEN	–	TnMS1	TnMS0	–	–	–	TnCC
R/W	–	R/W	R/W	–	–	–	R/W

Initial value : 00H

TnEN Control Timer n

0 Timer n disable

1 Timer n enable (Counter clear and start)

TnMS[1:0] Control Timer n Operation Mode

TnMS1	TnMS0	Description
0	0	Timer/counter mode (TnO: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWMnO)
1	1	PPG repeat mode (PWMnO)

TnCC Clear Timer n Counter

0 No effect

1 Clear the Timer n counter (When write, automatically cleared “0” after being cleared counter)

TnCRL (Timer n Control Low Register) : CAH/D2H, n= 4 and 5

7	6	5	4	3	2	1	0
TnCK2	TnCK1	TnCK0	TnIFR	–	TnPOL	TnECE	TnCNTR
R/W	R/W	R/W	R/W	–	R/W	R/W	R/W

Initial value : 00H

TnCK[2:0] Select Timer n clock source. fx is main system clock frequency

TnCK2	TnCK1	TnCK0	Description
0	0	0	fx/2048
0	0	1	fx/512
0	1	0	fx/64
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	External clock (ECn)

TnIFR When Tn Interrupt occurs, this bit becomes ‘1’. For clearing bit, write ‘0’ to this bit or auto clear by INT_ACK signal. Writing “1” has no effect.

0 Tn Interrupt no generation

1 Tn Interrupt generation

TnPOL TnO/PWMnO Polarity Selection

0 Start High (TnO/PWMnO is low level at disable)

1 Start Low (TnO/PWMnO is high level at disable)

TnECE Timer n External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

TnCNTR Timer n Counter Read Control

0 No effect

1 Load the counter value to the B data register (When write, automatically cleared “0” after being loaded)

11.7 Timer 6/7/8/9

11.7.1.1 Overview

The 16-bit timer 6/7/8/9 consists of multiplexer, timer 6/7/8/9 A data register high/low, timer 6/7/8/9 B data register high/low and timer 6/7/8/9 control register high/low (TnADRH, TnADRL, TnBDRH, TnBDRL, TnCRH, TnCRL).

It has four operating modes:

- 16-bit timer/counter mode
- 16-bit capture mode
- 16-bit PPG output mode (one-shot mode)
- 16-bit PPG output mode (repeat mode)

The timer/counter 6/7/8/9 can be clocked by an internal or an external clock source (ECn). The clock source is selected by clock selection logic which is controlled by the clock selection bits (TnCK[2:0]).

- TIMER n clock source: f_X/1, 2, 4, 8, 64, 512, 2048 and ECn

In the capture mode, by EINT1n, the data is captured into input capture data register (TnBDRH/TnBDRL). Timer 6/7/8/9 outputs the comparison result between counter and data register through TnO port in timer/counter mode. Also Timer 6/7/8/9 outputs PWM wave form through PWMnO port in the PPG mode.

Table 11-9 Timer 6/7/8/9 Operating Modes

TnEN	TnMS[1:0]	TnCK[2:0]	Timer 6/7/8/9
1	00	XXX	16 Bit Timer/Counter Mode
1	01	XXX	16 Bit Capture Mode
1	10	XXX	16 Bit PPG Mode (one-shot mode)
1	11	XXX	16 Bit PPG Mode (repeat mode)

11.7.2 16-Bit Timer/Counter Mode

The 16-bit timer/counter mode is selected by control register as shown in Figure 11.22.

The 16-bit timer have counter and data register. The counter register is increased by internal or external clock input. Timer 6/7/8/9 can use the input clock with one of 1, 2, 4, 8, 64, 512 and 2048 prescaler division rates (TnCK[2:0]). When the value of TnCNTH, TnCNTL and the value of TnADRH, TnADRL are identical in Timer 6/7/8/9 respectively, a match signal is generated and the interrupt of Timer 6/7/8/9 occurs. The TnCNTH, TnCNTL value is automatically cleared by match signal. It can be also cleared by software (TnCC).

The external clock (ECn) counts up the timer at the rising edge. If the ECn is selected as a clock source by TnCK[2:0], ECn port should be set to the input port by P74IO, P35IO, P36IO, P37IO bit.

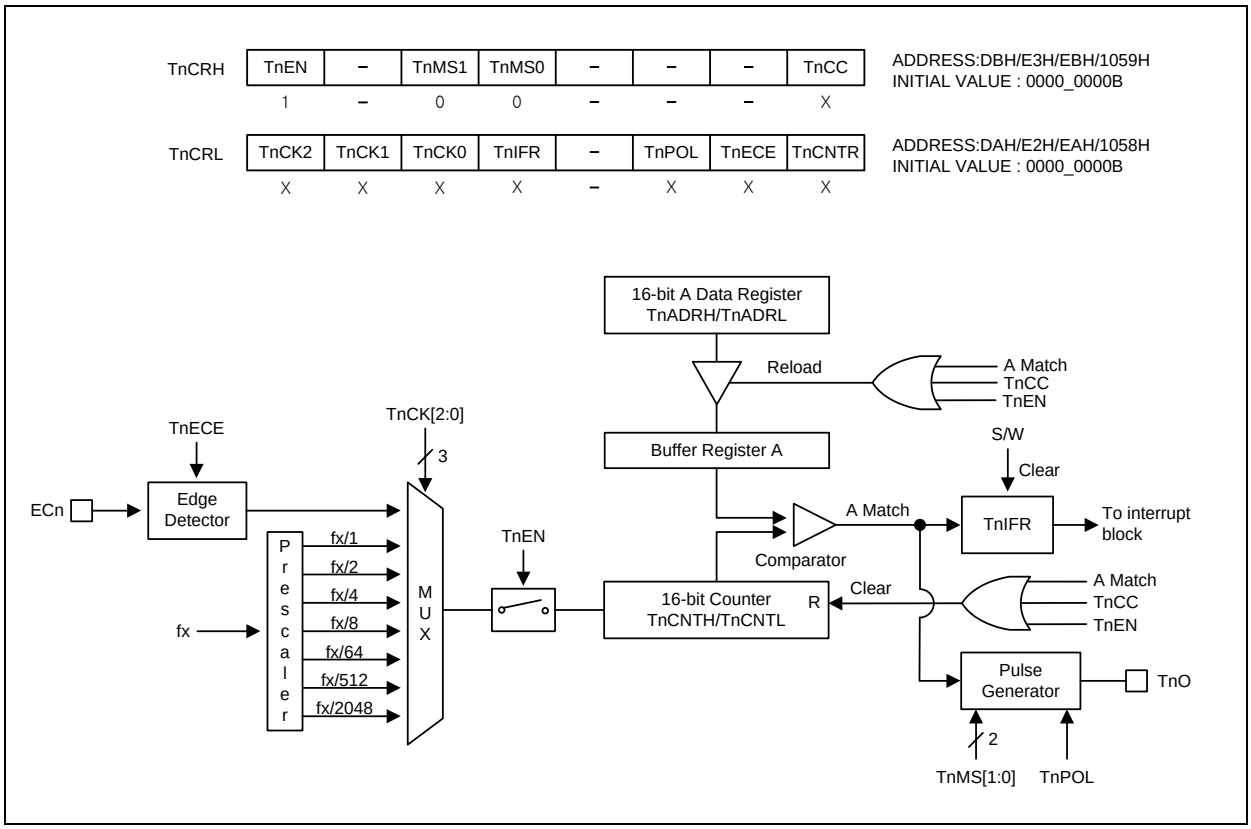


Figure 11.22 16-Bit Timer/Counter Mode for Timer 6/7/8/9 (where n= 6,7,8, and 9)

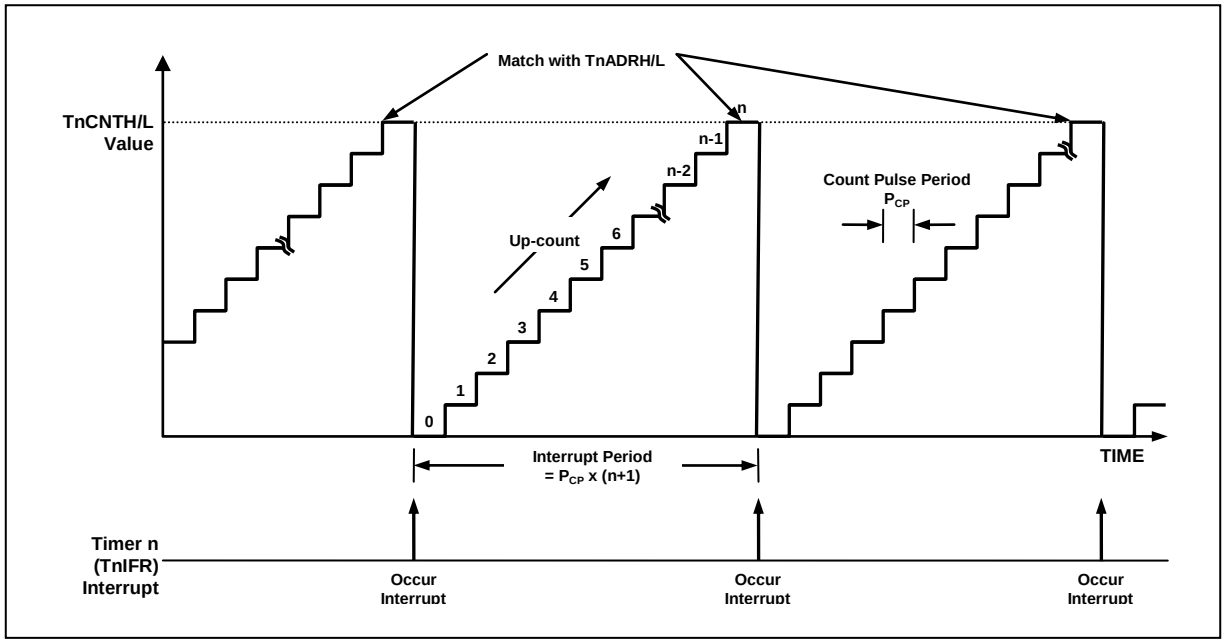


Figure 11.23 16-Bit Timer/Counter 6/7/8/9 Example (where n= 6,7,8,and 9)

11.7.3 16-Bit Capture Mode

The 16-bit timer 6/7/8/9 capture mode is set by TnMS[1:0] as '01'. The clock source can use the internal/external clock. Basically, it has the same function as the 16-bit timer/counter mode and the interrupt occurs when TnCNTH/TnCNTL is equal to TnADRH/TnADRL. The TnCNTH, TnCNTL values are automatically cleared by match signal. It can be also cleared by software (TnCC).

This timer interrupt in capture mode is very useful when the pulse width of captured signal is wider than the maximum period of timer.

The capture result is loaded into TnBDRH/TnBDRL.

According to EIPOL1H and EIPOL2H registers setting, the external interrupt EINT1n function is chosen. Of course, the EINT1n pin must be set as an input port.

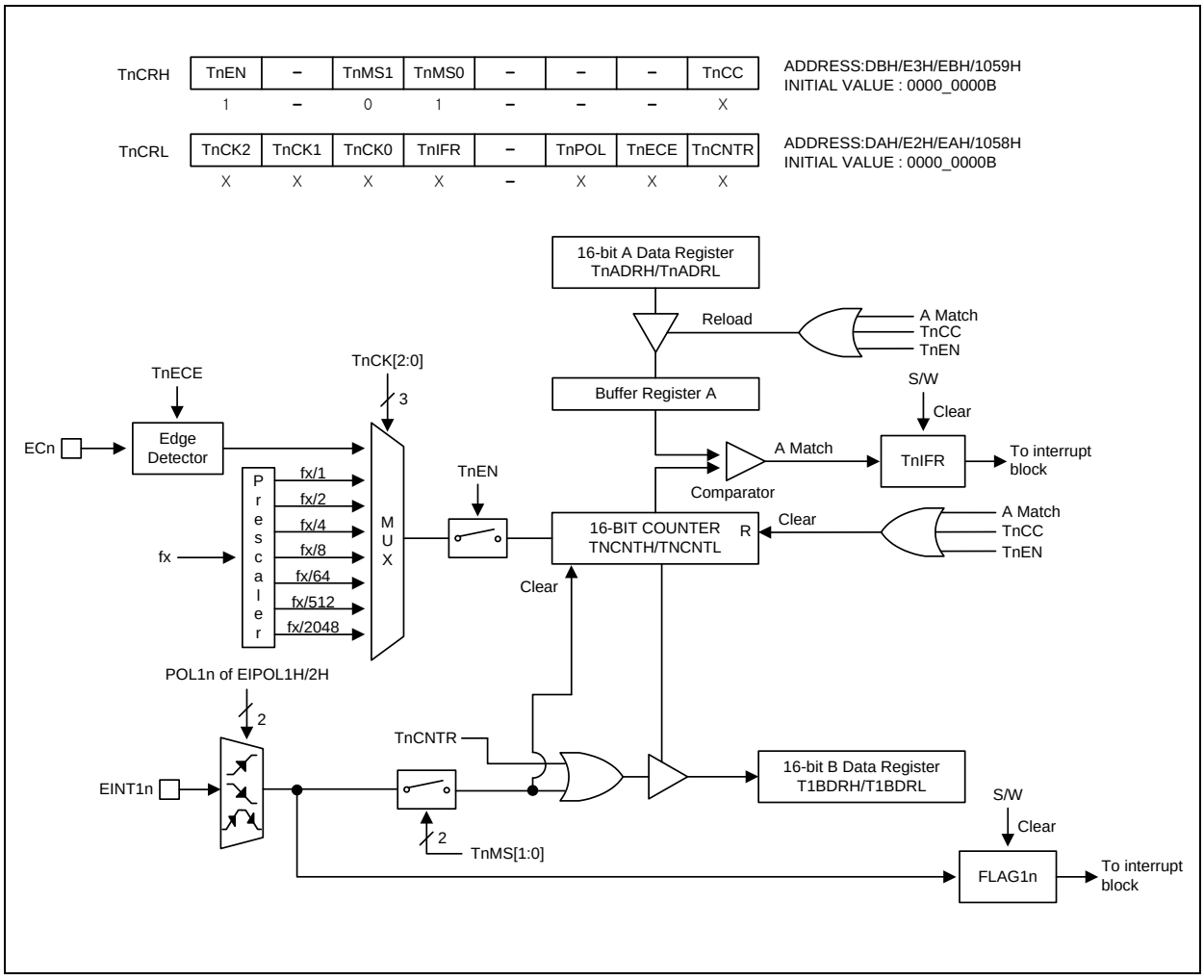


Figure 11.24 16-Bit Capture Mode for Timer 6/7/8/9 (where n= 6,7,8, and 9)

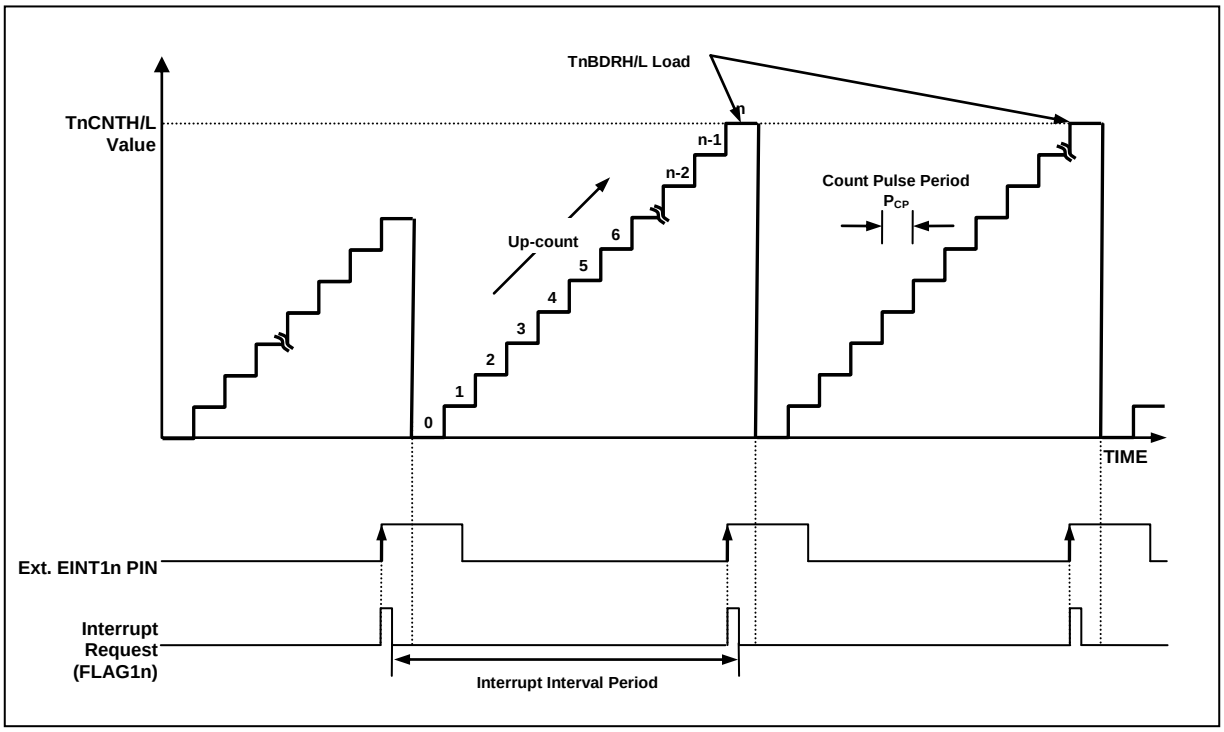


Figure 11.25 Input Capture Mode Operation for Timer 6/7/8/9 (where n= 6,7,8, and 9)

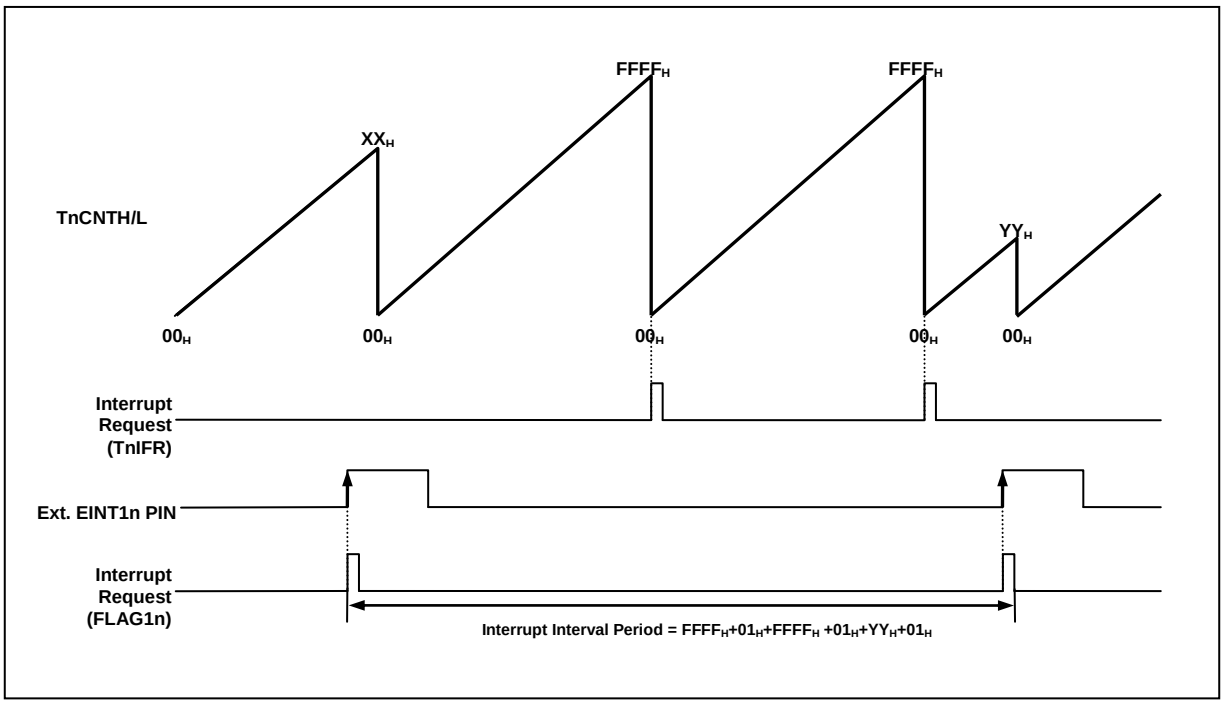
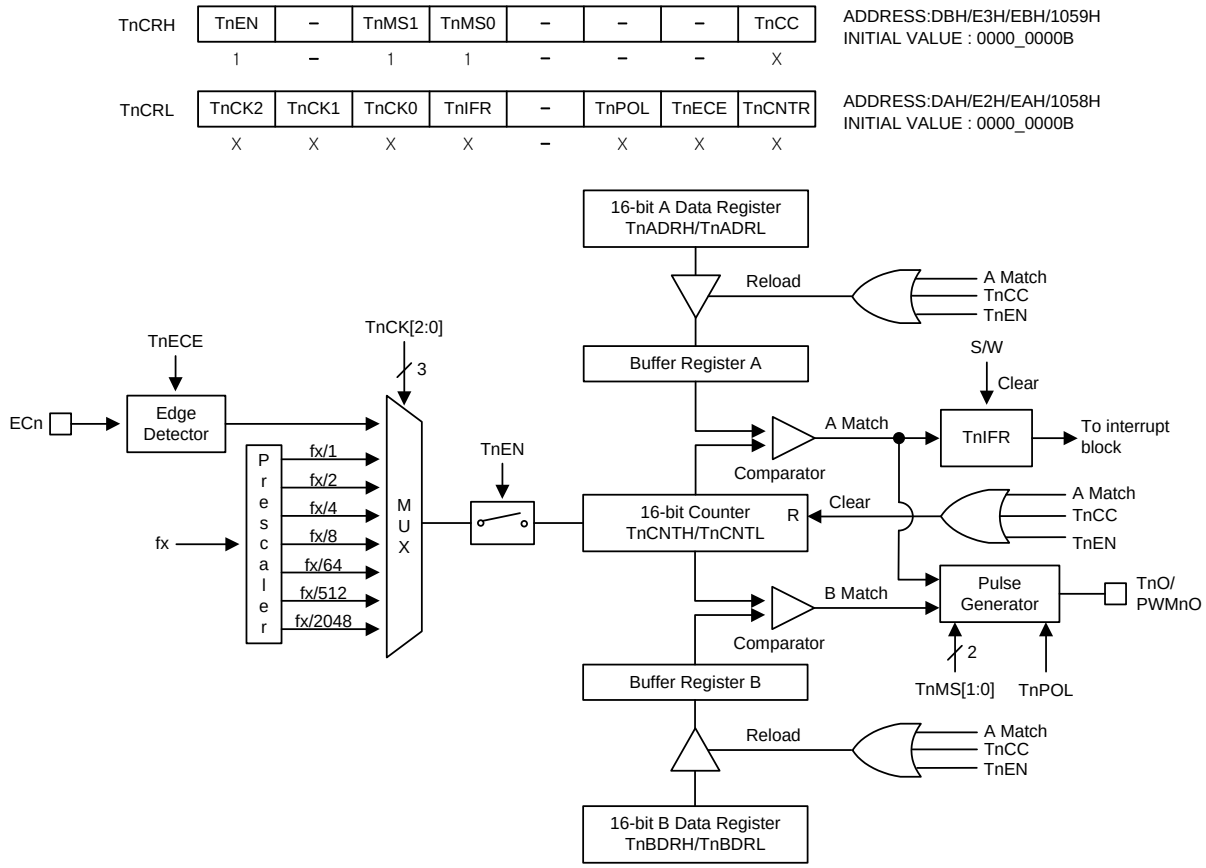


Figure 11.26 Express Timer Overflow in Capture Mode (where n= 6,7,8, and 9)

11.7.4 16-Bit PPG Mode

The timer 6/7/8/9 has a PPG (Programmable Pulse Generation) function. In PPG mode, TnO/PWMnO pin outputs up to 16-bit resolution PWM output. This pin should be configured as a PWM output by setting P3FSR2,P2FSR0,P2FSR1,P2FSR2 to '1'. The period of the PWM output is determined by the TnADRH/TnADRL. And the duty of the PWM output is determined by the TnBDRH/TnBDRL.



NOTE) The TnEN is automatically cleared to logic "0" after one pulse is generated at a PPG one-shot mode.

Figure 11.27 16-Bit PPG Mode for Timer 6/7/8/9 (where n= 6,7,8, and 9)

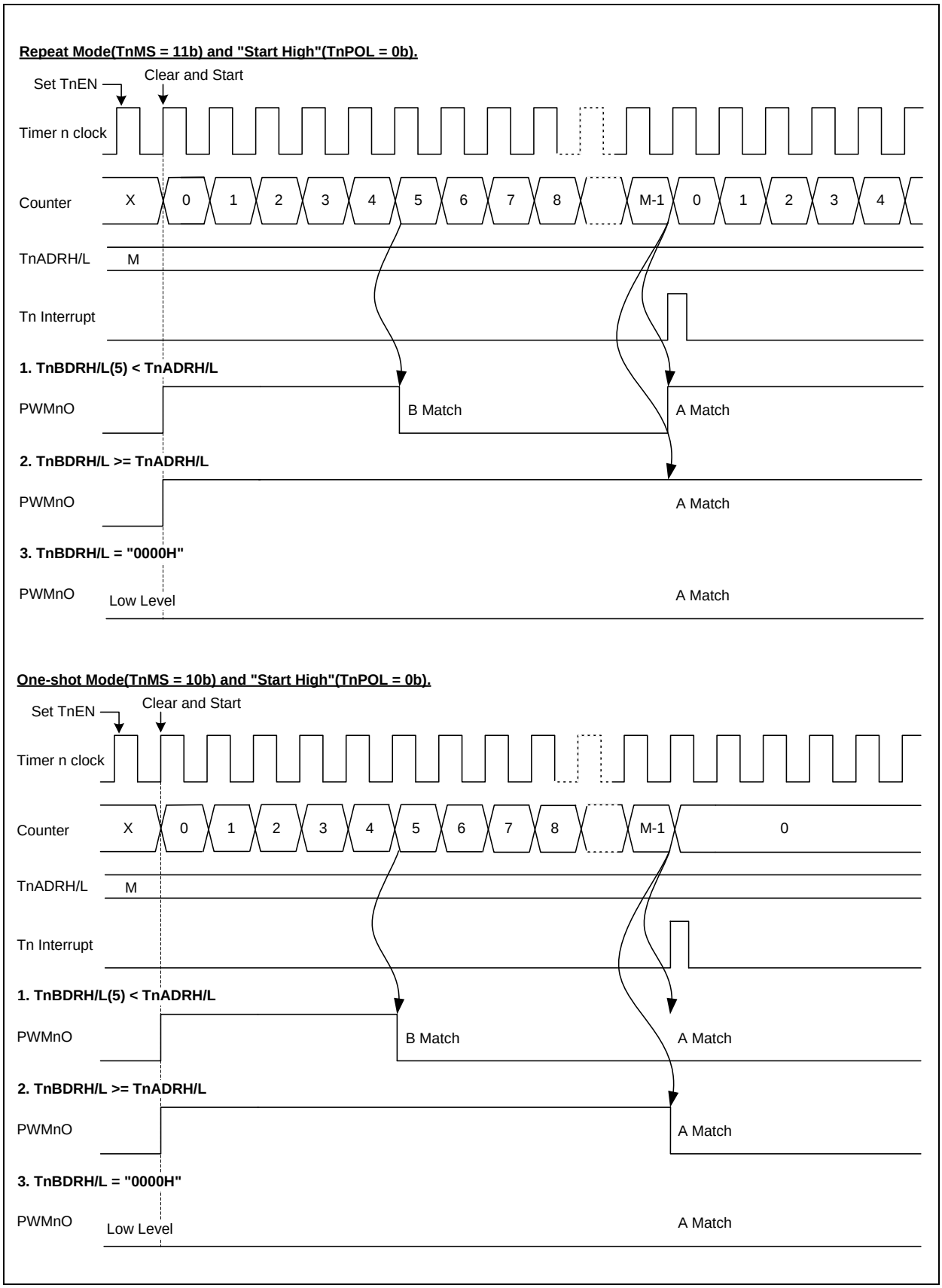


Figure 11.28 16-Bit PPG Mode Timming chart for Timer 6/7/8/9 (where n= 6,7,8, and 9)

11.7.5 Block Diagram

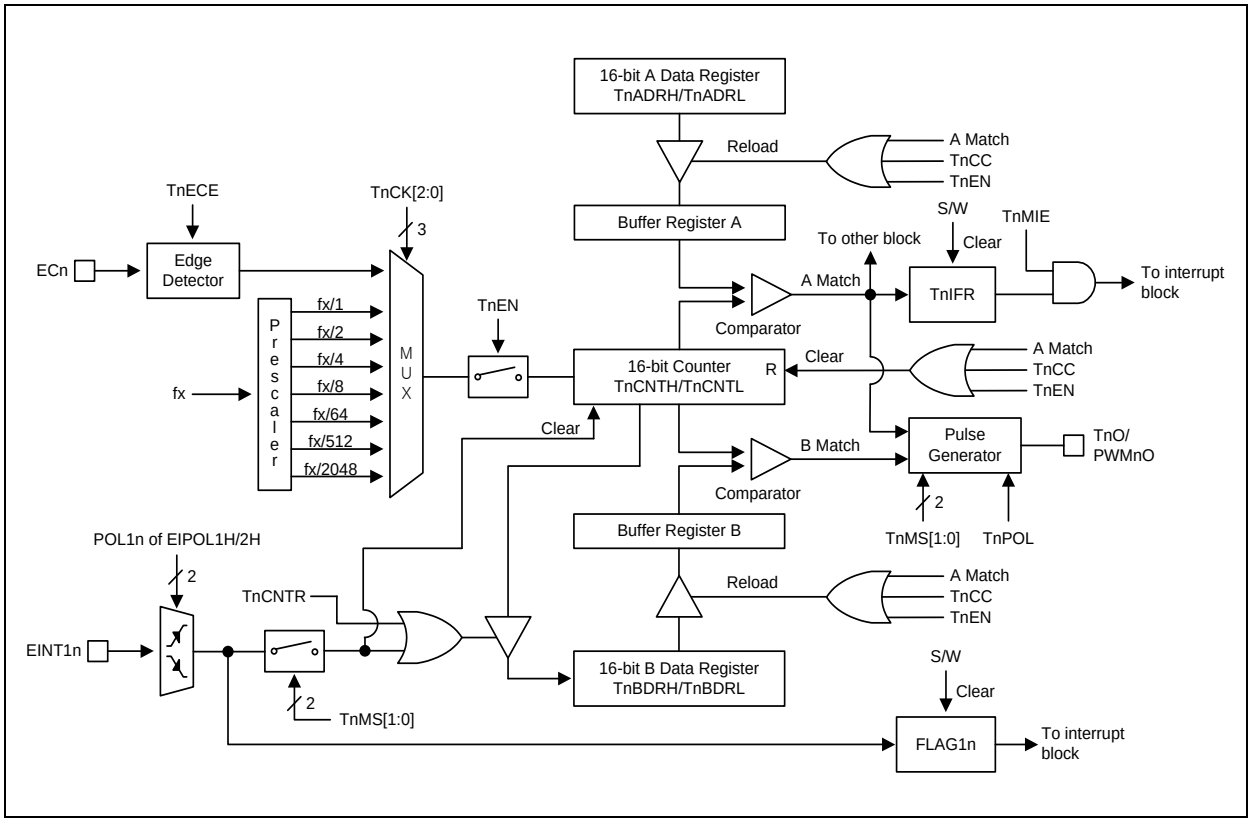


Figure 11.29 16-Bit Timer 6/7/8/9 Block Diagram (where n= 6,7,8, and 9)

11.7.6 Register Map

Table 11-10 Timer 2 Register Map

Name	Address	Dir	Default	Description
TnADRH	DDH/E5H/EDH/105BH	R/W	FFH	Timer n A Data High Register
TnADRL	DCH/E4H/ECH/105AH	R/W	FFH	Timer n A Data Low Register
TnBDRH	DFH/E7H/EFH/105DH	R/W	FFH	Timer n B Data High Register
TnBDRL	DEH/E6H/EEH/105CH	R/W	FFH	Timer n B Data Low Register
TnCRH	DBH/E3H/EBH/1059H	R/W	00H	Timer n Control High Register
TnCRL	DAH/E2H/EAH/1058H	R/W	00H	Timer n Control Low Register
TIFLAG1	97H	R/W	00H	Timer Interrupt Flag 1 Register

11.7.6.1 Timer/Counter 6/7/8/9 Register Description

The timer/counter 6/7/8/9 register consists of timer 6/7/8/9 A data high register (TnADRH), timer 6/7/8/9 A data low register (TnADRL), timer 6/7/8/9 B data high register (TnBDRH), timer 6/7/8/9 B data low register (TnBDRL), timer 6/7/8/9 control high register (TnCRH), timer 6/7/8/9 control low register (TnCRL), and timer interrupt flag register(TIFLAG1).

11.7.6.2 Register Description for Timer/Counter 6/7/8/9

TnADRH (Timer n A data High Register) : DDH/E5H/EDH/105BH (SFR/SFR/SFR/XSFR), n= 6, 7, 8, and 9

7	6	5	4	3	2	1	0
TnADRH7	TnADRH6	TnADRH5	TnADRH4	TnADRH3	TnADRH2	TnADRH1	TnADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnADRH[7:0] Tn A Data High Byte

TnADRL (Timer n A Data Low Register) : DCH/E4H/ECH/105AH (SFR/SFR/SFR/XSFR), n= 6, 7, 8, and 9

7	6	5	4	3	2	1	0
TnADRL7	TnADRL6	TnADRL5	TnADRL4	TnADRL3	TnADRL2	TnADRL1	TnADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnADRL[7:0] Tn A Data Low Byte

NOTE) Do not write “0000H” in the TnADRH/TnADRL register when PPG mode

TnBDRH (Timer n B Data High Register) : DFH/E7H/EFH/105DH (SFR/SFR/SFR/XSFR), n= 6, 7, 8, and 9

7	6	5	4	3	2	1	0
TnBDRH7	TnBDRH6	TnBDRH5	TnBDRH4	TnBDRH3	TnBDRH2	TnBDRH1	TnBDRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnBDRH[7:0] Tn B Data High Byte

TnBDRL (Timer n B Data Low Register) : **DEH/E6H/EEH/105CH (SFR/SFR/SFR/XSFR), n= 6, 7, 8, and 9**

7	6	5	4	3	2	1	0
TnBDRL7	TnBDRL6	TnBDRL5	TnBDRL4	TnBDRL3	TnBDRL2	TnBDRL1	TnBDRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

TnBDRL[7:0] Tn B Data Low Byte

TnCRH (Timer n Control High Register) : DBH/E3H/EBH/1059H (SFR/SFR/SFR/XSFR), n= 6, 7, 8, and 9

7	6	5	4	3	2	1	0
TnEN	TnMIE	TnMS1	TnMS0	–	–	–	TnCC
R/W	R/W	R/W	R/W	–	–	–	R/W

Initial value : 00H

TnEN Control Timer n

0 Timer n disable

1 Timer n enable (Counter clear and start)

TnMIE Enable or Disable Timer n Match Interrupt

0 Disable

1 Enable

TnMS[1:0] Control Timer n Operation Mode

TnMS1	TnMS0	Description
0	0	Timer/counter mode (TnO: toggle at A match)
0	1	Capture mode (The A match interrupt can occur)
1	0	PPG one-shot mode (PWMnO)
1	1	PPG repeat mode (PWMnO)

TnCC Clear Timer n Counter

0 No effect

1 Clear the Timer n counter (When write, automatically cleared “0” after being cleared counter)

Note) Refer to the timer interrupt flag 1 register (TIFLAG1) for the T6/T7/T8/T9 interrupt flags.

TnCRL (Timer n Control Low Register) : DAH/E2H/EAH/1058H (SFR/SFR/SFR/XSFR), n= 6, 7, 8, and 9

7	6	5	4	3	2	1	0
TnCK2	TnCK1	TnCK0	–	–	TnPOL	TnECE	TnCNTR
R/W	R/W	R/W	–	–	R/W	R/W	R/W

Initial value : 00H

TnCK[2:0] Select Timer n clock source. fx is main system clock frequency

TnCK2	TnCK1	TnCK0	Description
0	0	0	fx/2048
0	0	1	fx/512
0	1	0	fx/64
0	1	1	fx/8
1	0	0	fx/4
1	0	1	fx/2
1	1	0	fx/1
1	1	1	External clock (ECn)

TnPOL TnO/PWMnO Polarity Selection

0 Start High (TnO/PWMnO is low level at disable)

1 Start Low (TnO/PWMnO is high level at disable)

TnECE Timer n External Clock Edge Selection

0 External clock falling edge

1 External clock rising edge

TnCNTR Timer n Counter Read Control

0 No effect

1 Load the counter value to the B data register (When write, automatically cleared “0” after being loaded)

TIFLAG1(Timer Interrupt Flag 1 Register) : 97H

7	6	5	4	3	2	1	0
–	–	–	–	T9IFR	T8IFR	T7IFR	T6IFR
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

T9IFR When T9 interrupt occurs, this bit becomes ‘1’ The flag is cleared only by writing a ‘0’ to the bit. So, the flag should be cleared by software. Writing “1” has no effect.

0 T9 interrupt no generation

1 T9 interrupt generation

T8IFR When T8 interrupt occurs, this bit becomes ‘1’ The flag is cleared only by writing a ‘0’ to the bit. So, the flag should be cleared by software. Writing “1” has no effect.

0 T8 interrupt no generation

1 T8 interrupt generation

T7IFR When T7 interrupt occurs, this bit becomes ‘1’ The flag is cleared only by writing a ‘0’ to the bit. So, the flag should be cleared by software. Writing “1” has no effect.

0 T7 interrupt no generation

1 T7 interrupt generation

T6IFR When T6 interrupt occurs, this bit becomes ‘1’ The flag is cleared only by writing a ‘0’ to the bit. So, the flag should be cleared by software. Writing “1” has no effect.

0 T6 interrupt no generation

1 T6 interrupt generation

11.8 Buzzer Driver

11.8.1 Overview

The Buzzer consists of 6 bit counter, buzzer data register (BUZDR), and buzzer control register (BUZCR). The Square Wave (122Hz~62.5kHz @f_{BUZ} = 500kHz) is outputted through P33/BUZO pin. In buzzer data register (BUZDR),BUZDR[5:0] controls the buzzer frequency (look at the following expression) and BUZDIV[1:0] selects f_{BUZ} divided by DIV block . In buzzer control register (BUZCR), BUCK[2:0] selects source clock divided by prescaler.

$$f_{BUZO}(Hz) = \frac{\text{Oscillator Frequency}}{2 \times \text{BUZDIV} \times \text{Prescaler Ratio} \times (\text{BUZDR}+1)}$$

Table 11-11 Buzzer Frequency at f_{BUZ}= 500 kHz

BUZDR [5:0]	BUZDR [7:6]=00 fbuz/8	BUZDR [7:6]=01 fbuz/16	BUZDR [7:6]=10 fbuz/32	BUZDR [7:6]=11 fbuz/64
0	62.500	31.250	15.625	7.813
1	31.250	15.625	7.813	3.906
2	20.833	10.417	5.208	2.604
3	15.625	7.813	3.906	1.953
4	12.500	6.250	3.125	1.563
5	10.417	5.208	2.604	1.302
6	8.929	4.464	2.232	1.116
7	7.813	3.906	1.953	0.977
8	6.944	3.472	1.736	0.868
9	6.250	3.125	1.563	0.781
10	5.682	2.841	1.420	0.710
11	5.208	2.604	1.302	0.651
12	4.808	2.404	1.202	0.601
13	4.464	2.232	1.116	0.558
14	4.167	2.083	1.042	0.521
15	3.906	1.953	0.977	0.488
16	3.676	1.838	0.919	0.460
17	3.472	1.736	0.868	0.434
18	3.289	1.645	0.822	0.411
19	3.125	1.563	0.781	0.391
20	2.976	1.488	0.744	0.372
21	2.841	1.420	0.710	0.355
22	2.717	1.359	0.679	0.340
23	2.604	1.302	0.651	0.326
24	2.500	1.250	0.625	0.313
25	2.404	1.202	0.601	0.300
26	2.315	1.157	0.579	0.289
27	2.232	1.116	0.558	0.279
28	2.155	1.078	0.539	0.269
29	2.083	1.042	0.521	0.260
30	2.016	1.008	0.504	0.252
31	1.953	0.977	0.488	0.244

BUZDR [5:0]	BUZDR [7:6]=00 fbuz/8	BUZDR [7:6]=01 fbuz/16	BUZDR [7:6]=10 fbuz/32	BUZDR [7:6]=11 fbuz/64
32	1.894	0.947	0.473	0.237
33	1.838	0.919	0.460	0.230
34	1.786	0.893	0.446	0.223
35	1.736	0.868	0.434	0.217
36	1.689	0.845	0.422	0.211
37	1.645	0.822	0.411	0.206
38	1.603	0.801	0.401	0.200
39	1.563	0.781	0.391	0.195
40	1.524	0.762	0.381	0.191
41	1.488	0.744	0.372	0.186
42	1.453	0.727	0.363	0.182
43	1.420	0.710	0.355	0.178
44	1.389	0.694	0.347	0.174
45	1.359	0.679	0.340	0.170
46	1.330	0.665	0.332	0.166
47	1.302	0.651	0.326	0.163
48	1.276	0.638	0.319	0.159
49	1.250	0.625	0.313	0.156
50	1.225	0.613	0.306	0.153
51	1.202	0.601	0.300	0.150
52	1.179	0.590	0.295	0.147
53	1.157	0.579	0.289	0.145
54	1.136	0.568	0.284	0.142
55	1.116	0.558	0.279	0.140
56	1.096	0.548	0.274	0.137
57	1.078	0.539	0.269	0.135
58	1.059	0.530	0.265	0.132
59	1.042	0.521	0.260	0.130
60	1.025	0.512	0.256	0.128
61	1.008	0.504	0.252	0.126
62	0.992	0.496	0.248	0.124
63	0.977	0.488	0.244	0.122

11.8.2 Block Diagram

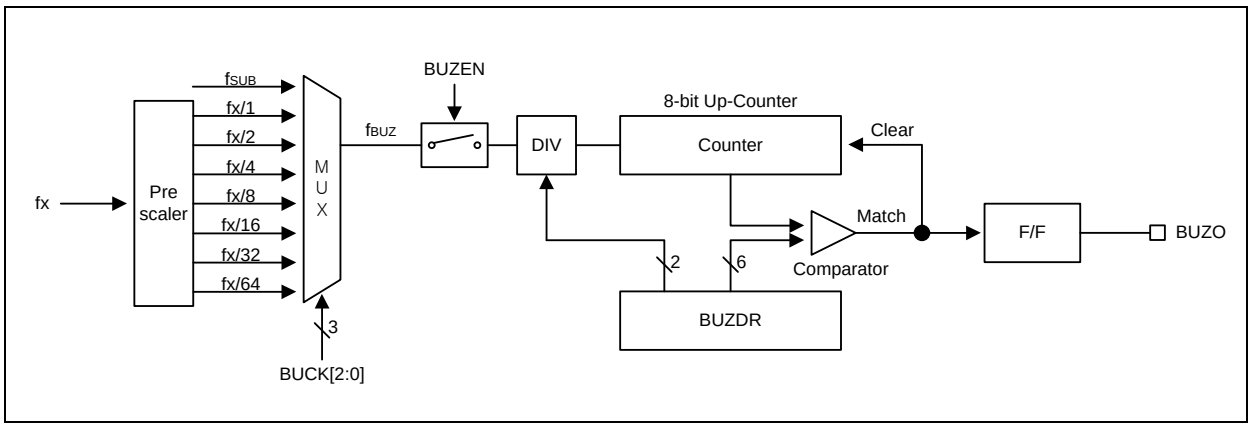


Figure 11.30 Buzzer Driver Block Diagram

11.8.3 Register Map

Table 11-12 Buzzer Driver Register Map

Name	Address	Dir	Default	Description
BUZDR	F3H	R/W	FFH	Buzzer Data Register
BUZCR	F2H	R/W	00H	Buzzer Control Register

11.8.4 Buzzer Driver Register Description

Buzzer driver consists of buzzer data register (BUZDR) and buzzer control register (BUZCR).

11.8.5 Register Description for Buzzer Driver

BUZDR (Buzzer Data Register) : F3H

7	6	5	4	3	2	1	0
BUZDIV1	BUZDIV0	BUZDR5	BUZDR4	BUZDR3	BUZDR2	BUZDR1	BUZDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

BUZDIV[1:0] Buzzer Clock Divider.

BUZDIV1	BUZDIV0	description
0	0	f_{BUZ}/8
0	1	f_{BUZ} /16
1	0	f_{BUZ} /32
1	1	f_{BUZ} /64

BUZDR[5:0] This bits control the Buzzer frequency
Its resolution is 00H ~ 3FH

BUZCR (Buzzer Control Register) : F2H

7	6	5	4	3	2	1	0
–	–	–	–	BUCK2	BUCK1	BUCK0	BUZEN
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

BUCK[2:0] Buzzer Driver Source Clock Selection

BUCK2	BUCK1	BUCK0	Description
0	0	0	fx/1
0	0	1	fx/2
0	1	0	fx/4
0	1	1	fx/8
1	0	0	fx/16
1	0	1	fx/32
1	1	0	fx/64
1	1	1	f_{SUB} (External Sub OSC)

BUZEN Buzzer Driver Operation Control

0	Buzzer Driver disable
1	Buzzer Driver enable

NOTE) fx: System clock oscillation frequency.

11.9 SPI 2/3

11.9.1 Overview

There is serial peripheral interface (SPI 2/3) one channel in MC97F2664. The SPI 2/3 allows synchronous serial data transfer between the external serial devices. It can do Full-duplex communication by 4-wire (MOSI2/3, MISO2/3, SCK2/3, SS2/3), support master/slave mode, can select serial clock (SCK2/3) polarity, phase and whether LSB first data transfer or MSB first data transfer.

11.9.2 Block Diagram

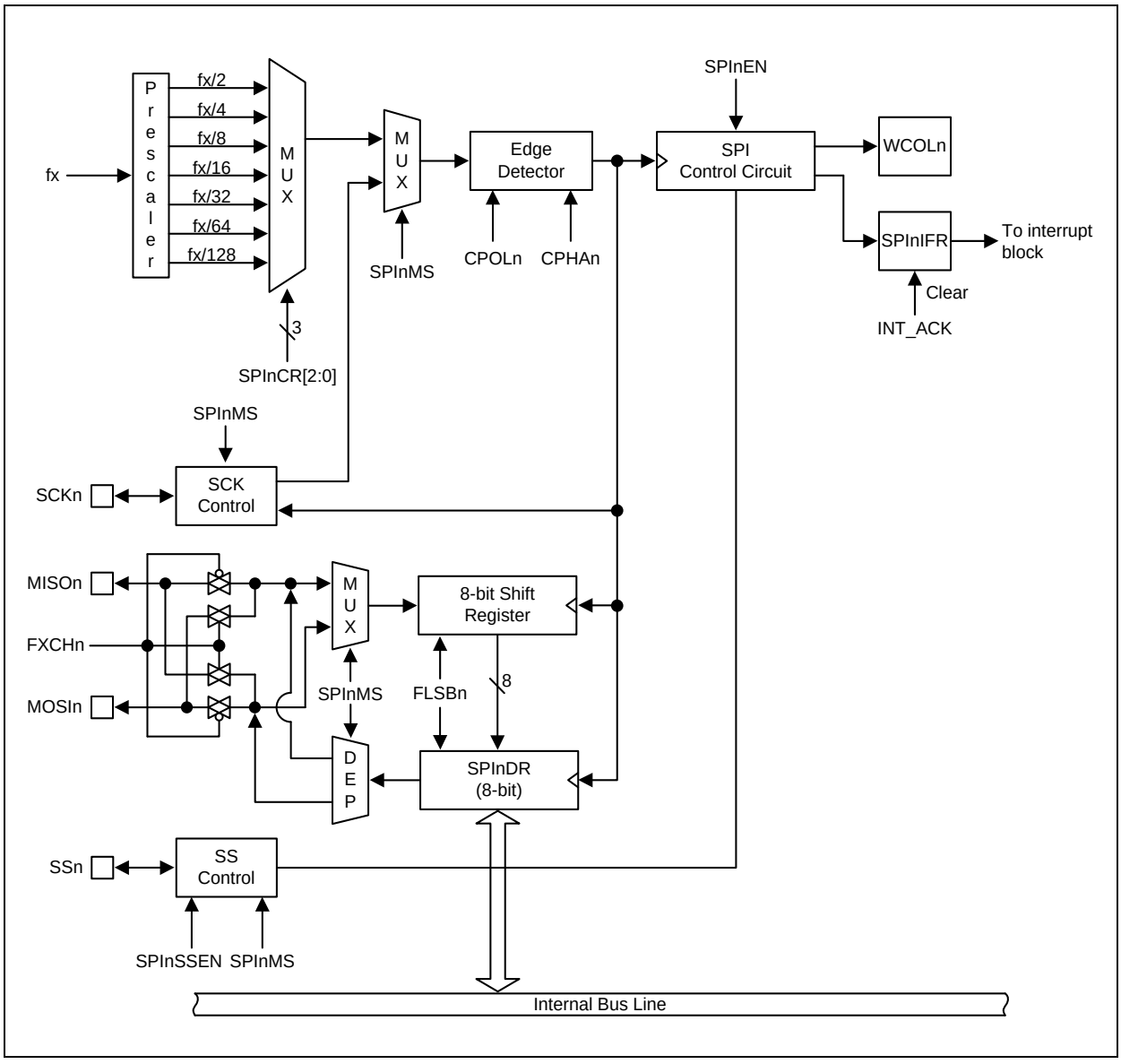


Figure 11.31 SPI 2/3 Block Diagram (where n = 2 and 3)

11.9.3 Data Transmit / Receive Operation

User can use SPI 2/3 for serial data communication by following step

1. Select SPI 2/3 operation mode(master/slave, polarity, phase) by control register SPInCR.
2. When the SPI 2/3 is configured as a Master, it selects a Slave by SS2/3 signal (active low).

   When the SPI 2/3 is configured as a Slave, it is selected by SS2/3 signal incoming from Master
3. When the user writes a byte to the data register SPInDR, SPI 2/3 will start an operation.
4. In this time, if the SPI 2/3 is configured as a Master, serial clock will come out of SCK2/3 pin. And Master shifts the eight bits into the Slave (transmit), Slave shifts the eight bits into the Master at the same time (receive). If the SPI 2/3 is configured as a Slave, serial clock will come into SCK2/3 pin. And Slave shifts the eight bits into the Master (transmit), Master shifts the eight bits into the Slave at the same time (receive).
5. When transmit/receive is done, SPInIFR bit will be set. If the SPI 2/3 interrupt is enabled, an interrupt is requested. And SPInIFR bit is cleared by hardware when executing the corresponding interrupt. If SPI 2/3 interrupt is disable, SPInIFR bit is cleared when user read the status register SPInSR, and then access (read/write) the data register SPInDR.

11.9.4 SS2/3 pin function

1. When the SPI 2/3 is configured as a Slave, the SS2/3 pin is always input. If LOW signal come into SS2/3 pin, the SPI 2/3 logic is active. And if ‘HIGH’ signal come into SS2/3 pin, the SPI 2/3 logic is stop. In this time, SPI 2/3 logic will be reset, and invalidated any received data.
2. When the SPI 2/3 is configured as a Master, the user can select the direction of the SS2/3 pin by port direction register (P37IO/P73IO). If the SS2/3 pin is configured as an output, user can use general P37IO/P73IO output mode. If the SS2/3 pin is configured as an input, ‘HIGH’ signal must come into SS2/3 pin to guarantee Master operation. If ‘LOW’ signal come into SS2/3 pin, the SPI 2/3 logic interprets this as another master selecting the SPI 2/3 as a slave and starting to send data to it. To avoid bus contention, MSB bit of SPICR will be cleared and the SPI 2/3 becomes a Slave and then, SPInIFR bit of SPInSR will be set, and if the SPI 2/3 interrupt is enabled, an interrupt is requested.

NOTES)

- When the SS2/3 pin is configured as an output at Master mode, SS2/3 pin’s output value is defined by user’s software (P37IO/P73IO). Before SPInCR setting, the direction of SS2/3 pin must be defined
- If you don’t need to use SS2/3 pin, clear the SPInSSEN bit of SPInSR. So, you can use disabled pin by P37IO/P73IO freely. In this case, SS2/3 signal is driven by ‘HIGH’ or ‘LOW’ internally. In other words, master is ‘HIGH’, salve is ‘LOW’
- When SS2/3 pin is configured as input, if ‘HIGH’ signal come into SS2/3 pin, SS_HIGH flag bit will be set. And you can clear it by writing ‘0’.

11.9.5 SPI 2/3 Timing Diagram

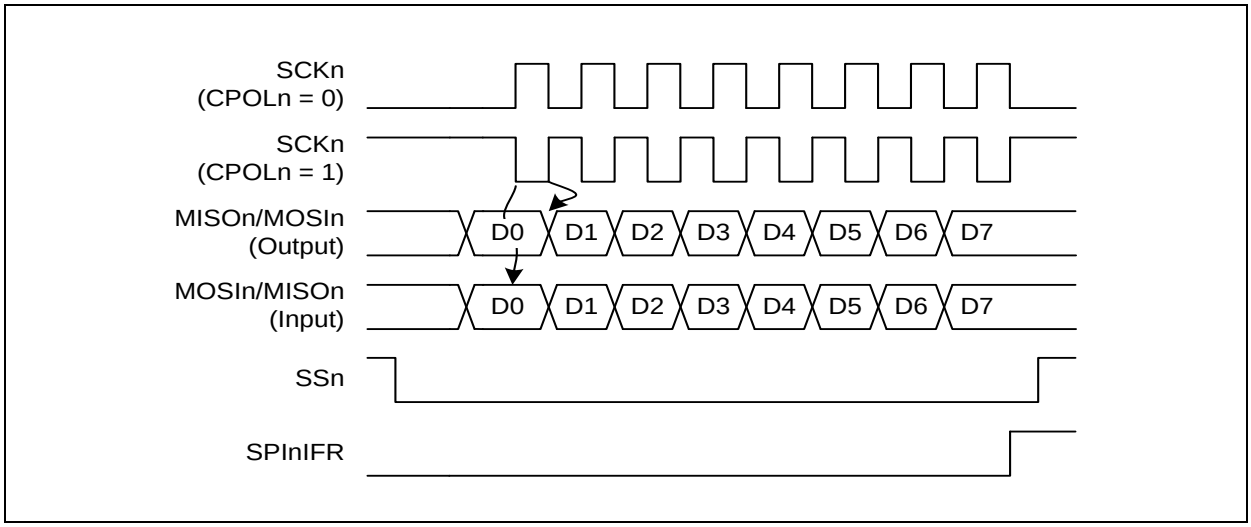


Figure 11.32 SPI 2/3 Transmit/Receive Timing Diagram at CPHA = 0 (Where n = 2 and 3)

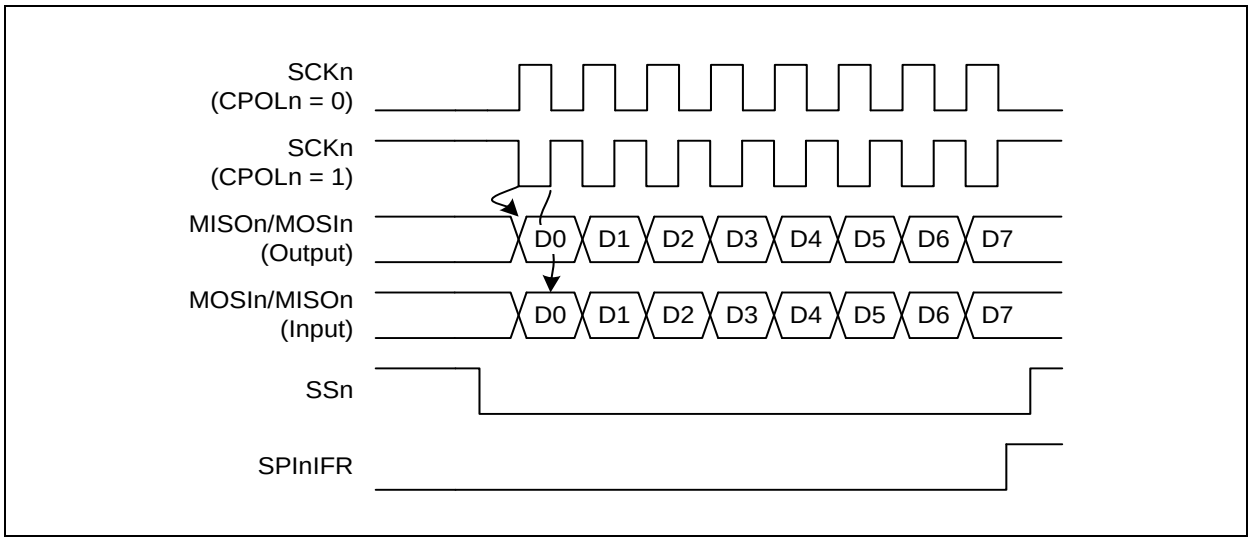


Figure 11.33 SPI 2/3 Transmit/Receive Timing Diagram at CPHA = 1 (Where n = 2 and 3)

11.9.6 Register Map

Table 11-13 SPI 2/3 Register Map

Name	Address	Dir	Default	Description
SPInSR	C4H/C7H	R/W	00H	SPI n Status Register
SPInDR	C3H/C6H	R/W	00H	SPI n Data Register
SPInCR	C2H/C5H	R/W	00H	SPI n Control Register

11.9.7 SPI 2/3 Register Description

The SPI 2/3 register consists of SPI 2/3 control register (SPInCR), SPI 2/3 status register (SPInSR) and SPI 2/3 data register (SPInDR)

11.9.8 Register Description for SPI 2/3

SPInDR (SPI 2/3 Data Register) : C3H/C6H, n= 2, 3

7	6	5	4	3	2	1	0
SPInDR7	SPInDR6	SPInDR5	SPInDR4	SPInDR3	SPInDR2	SPInDR1	SPInDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SPInDR [7:0] SPI 2/3 Data
When it is written a byte to this data register, the SPI 2/3 will start an operation.

SPInSR (SPI 2/3 Status Register) : C4H/C7H, n= 2, 3

7	6	5	4	3	2	1	0
SPIInFR	WCOLn	SS_HIGHn	–	FXCHn	SPInSSEN	–	–
R/W	R	R/W	–	R/W	R/W	–	–

Initial value : 00H

SPIIFRn When SPI 2/3 Interrupt occurs, this bit becomes '1'. If SPI 2/3 interrupt is enable, this bit is auto cleared by INT_ACK signal. And if SPI 2/3 Interrupt is disable, this bit is cleared when the status register SPnISR is read, and then access (read/write) the data register SPInDR Writing "1" has no effect.

0 SPI 2/3 Interrupt no generation

1 SPI 2/3 Interrupt generation

WCOLn This bit is set if any data are written to the data register SPInDR during transfer. This bit is cleared when the status register SPInSR is read, and then access (read/write) the data register SPInDR

0 No collision

1 Collision

SS_HIGHn When the SS2/3 pin is configured as input, if "HIGH" signal comes into the pin, this flag bit will be set.

0 Cleared when '0' is written

1 No effect when '1' is written

FXCHn SPI 2/3 port function exchange control bit.

0 No effect

1 Exchange MOSIn and MISOn function

SPInSSEN This bit controls the SS2/3 pin operation

0 Disable

1 Enable (The corresponding pin should be a normal input)

SPInCR (SPI 2/3 Control Register) : C2H/C5H, n= 2, 3

7	6	5	4	3	2	1	0
SPInEN	FLSBn	SPInMS	CPOLn	CPHAn	SPInDSCR	SPInSCR1	SPInSCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

SPInEN This bit controls the SPI 2/3 operation

0 Disable SPI 2/3 operation

1 Enable SPI 2/3 operation

FLSBn This bit selects the data transmission sequence

0 MSB first

1 LSB first

SPInMS This bit selects whether Master or Slave mode

0 Slave mode

1 Master mode

CPOLn This two bits control the serial clock (SCK2/3) mode.

CPHAn Clock polarity(CPOLn) bit determine SCK2/3's value at idle mode.

Clcok phase (CPHAn) bit determine if data are sampled on the leading or trailing edge of SCK2/3.

CPOLn	CPHAn	Leading edge	Trailing edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

SPInDSCR SPInSCR [1:0] These three bits select the SCK2/3 rate of the device configured as a master. When DSCR bit is written one, SCK2/3 will be doubled in master mode.

SPInDSCR	SPInSCR 1	SPInSCR 0	SCKn frequency
0	0	0	fx/4
0	0	1	fx/16
0	1	0	fx/64
0	1	1	fx/128
1	0	0	fx/2
1	0	1	fx/8
1	1	0	fx/32
1	1	1	fx/64

11.10 UART2/3/4

11.10.1 Overview

The universal asynchronous serial receiver and transmitter (UART2/3/4) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete

UART2/3/4 has baud rate generator, transmitter and receiver. The baud rate generator for asynchronous operation. The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows continuous transfer of data without any delay between frames. The receiver is the most complex part of the UART2/3/4 module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFOn (UARTnDR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

11.10.2 Block Diagram

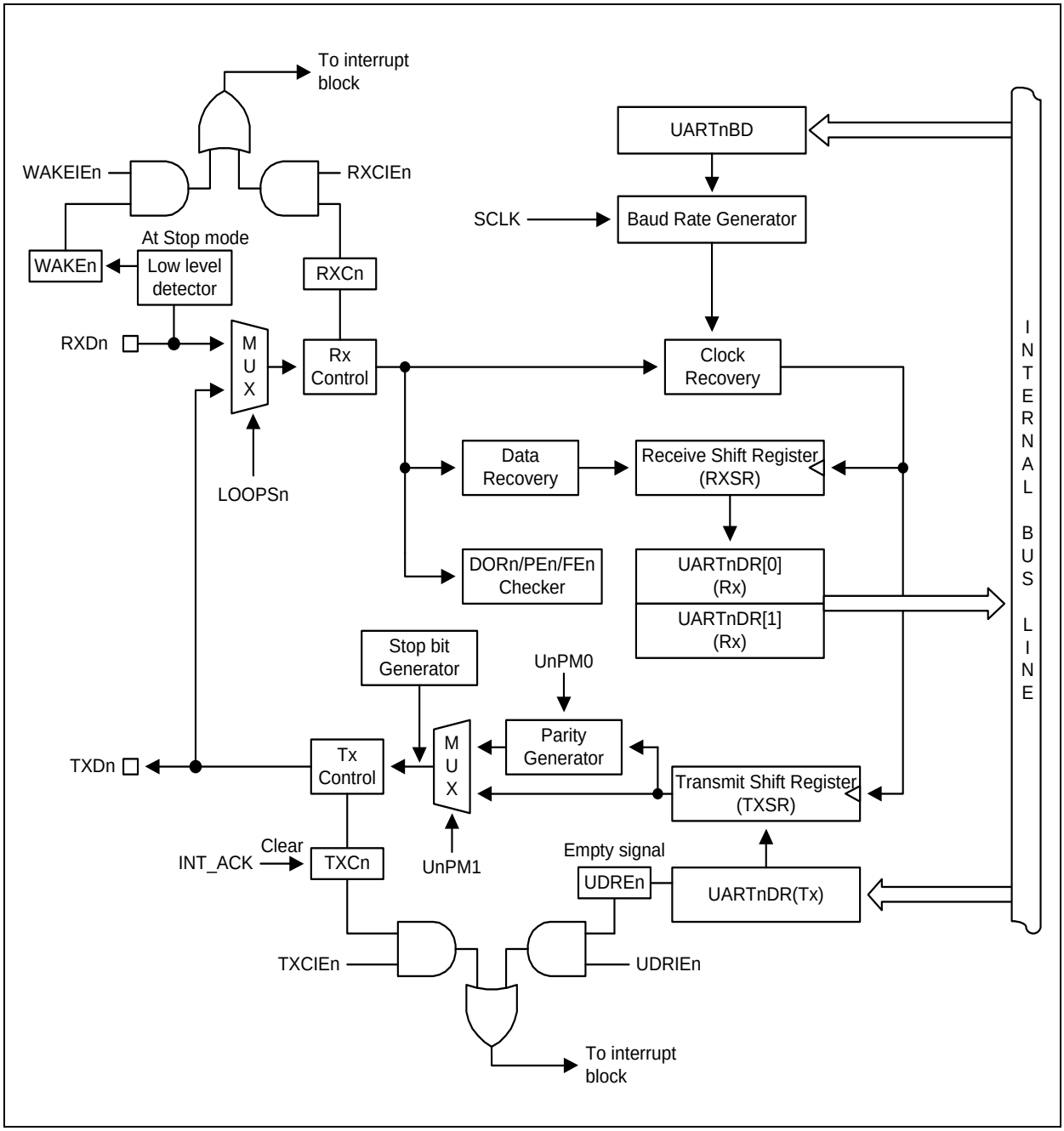


Figure 11.34 UART Block Diagram(where n = 2,3, and 4)

11.10.3 Clock Generation

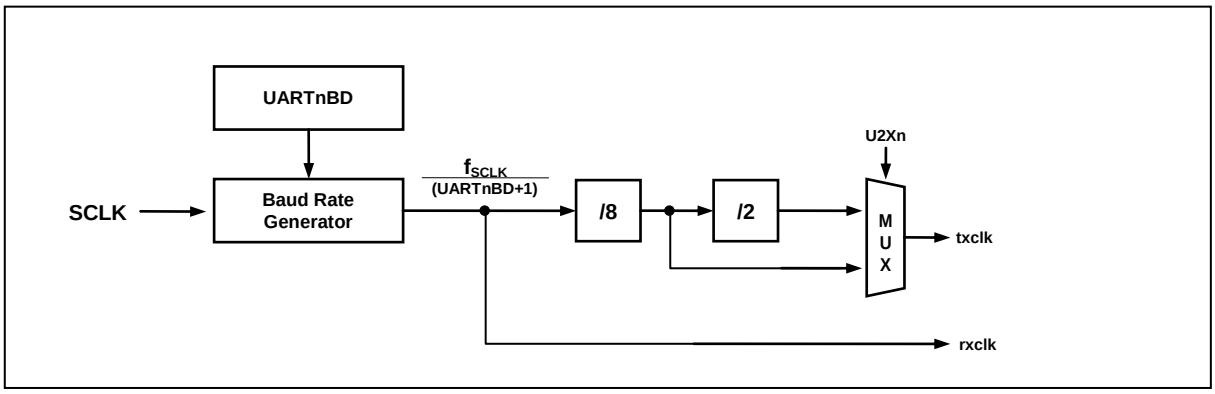


Figure 11.35 Clock Generation Block Diagram (where n = 2,3, and 4)

The clock generation logic generates the base clock for the transmitter and receiver.

Following table shows equations for calculating the baud rate (in bps).

Table 11-14 Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Normal Mode(U2Xn=0)	$\text{Baud Rate} = \frac{\text{fx}}{16(\text{UARTnBD} + 1)}$
Double Speed Mode(U2Xn=1)	$\text{Baud Rate} = \frac{\text{fx}}{8(\text{UARTnBD} + 1)}$

11.10.4 Data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection.

The UART2/3/4 supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The following figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

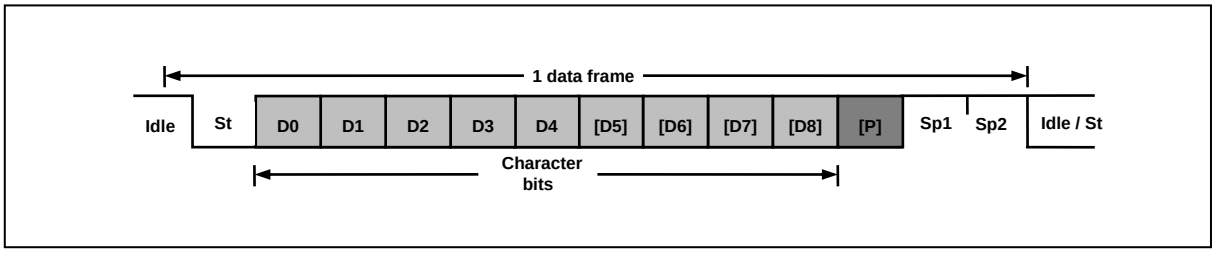


Figure 11.36 Frame Format

1 data frame consists of the following bits

- Idle No communication on communication line (TxD/RxD)
- St Start bit (Low)
- Dn Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

The frame format used by the UART2/3/4 is set by the UnSIZE[2:0], UnPM[1:0] and USBSn bits in UARTnCR1 and UARTnCR3 register. The Transmitter and Receiver use the same setting.

11.10.5 Parity bit

The parity bit is calculated by doing an exclusive-OR of all the data bits. If odd parity is used, the result of the exclusive-or is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

P_{even} : Parity bit using even parity

P_{odd} : Parity bit using odd parity

D_n : Data bit n of the character

11.10.6 UART2/3/4 Transmitter

The UART2/3/4 transmitter is enabled by setting the TXEn bit in UARTnCR2 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin of UART2/3/4 by the P4FSR3/ P4FSR4/P2FSR3. The baud-rate, operation mode and frame format must be setup once before doing any transmission.

11.10.6.1 Sending Tx data

A data transmission is initiated by loading the transmit buffer (UARTnDR register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used, the ninth bit must be written to the UnTX8 bit in UARTnCR3 register before it is loaded to the transmit buffer (UARTnDR register).

11.10.6.2 Transmitter flag and interrupt

The UART2/3/4 transmitter has 2 flags which indicate its state. One is UART2/3/4 data register empty flag (UDREn) and the other is transmit complete flag (TXCn). Both flags can be interrupt sources.

UDREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the data register empty interrupt enable (UDRIEn) bit in UARTnCR2 register is set and the global interrupt is enabled, UART2/3/4 data register empty interrupt is generated while UDREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXCn flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXCn bit in UARTnST register.

When the transmit complete interrupt enable (TXCIEn) bit in UARTnCR2 register is set and the global interrupt is enabled, UART2/3/4 transmit complete interrupt is generated while TXCn flag is set.

11.10.6.3 Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (UnPM[1]=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

11.10.6.4 Disabling Transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO).

11.10.7 UART Receiver

The UART2/3/4 receiver is enabled by setting the RXEn bit in the UARTnCR2 register. When the receiver is enabled, the RXDn pin should be set to the input port for the serial input pin of UART2/3/4 by P24IO/ P45IO/ P47IO bit. The baud-rate, mode of operation and frame format must be set before serial reception.

11.10.7.1 Receiving Rx data

The receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2^{nd} stop bit in the frame, the 2^{nd} stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the UARTnDR register.

If 9-bit characters are used (UnSIZE[2:0] = "111"), the ninth bit is stored in the RX8n bit position in the UARTnCR3 register. The 9^{th} bit must be read from the RX8n bit before reading the low 8 bits from the UARTnDR register. Likewise, the error flags FEn, DORn, PEn must be read before reading the data from UARTnDR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

11.10.7.2 Receiver Flag and Interrupt

The UART2/3/4 receiver has one flag that indicates the receiver state.

The receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the UARTnCR2 register is set and global interrupt is enabled, the UART2/3/4 receiver complete interrupt is generated while RXCn flag is set.

The UART2/3/4 receiver has three error flags which are frame error (FEn), data overrun (DORn) and parity error (PEn). These error flags can be read from the UARTnST register. As received data are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from UARTnDR register, read the UARTnST register first which contains error flags.

The frame error (FEn) flag indicates the state of the first stop bit. The FEn flag is ‘0’ when the stop bit was correctly detected as ‘1’, and the FE flag is ‘1’ when the stop bit was incorrect, i.e. detected as ‘0’. This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PEn) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (UnPM[1]=0), the PEn bit is always read ‘0’.

11.10.7.3 Parity Checker

If parity bit is enabled (UnPM[1]=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

11.10.7.4 Disabling Receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO).

11.10.7.5 Asynchronous Data Reception

To receive asynchronous data frame, the UART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin.

The data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of the baud-rate in normal mode and 8 times the baud-rate for double speed mode (U2Xn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the double speed mode.

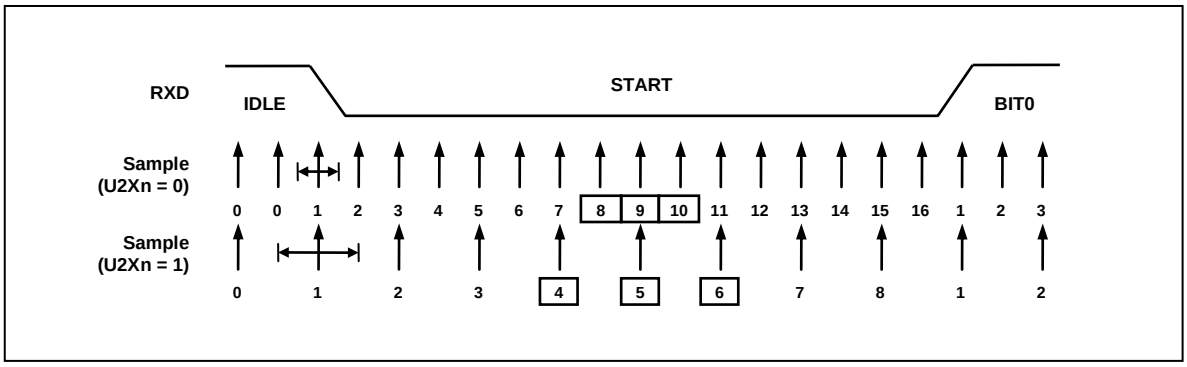


Figure 11.37 Start Bit Sampling (where n = 2,3, and 4)

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8,9, and 10 for normal mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode, and uses sample 8, 9, and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered to a logic ‘0’ and if more than 2 samples have high levels, the received bit is considered to a logic ‘1’. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

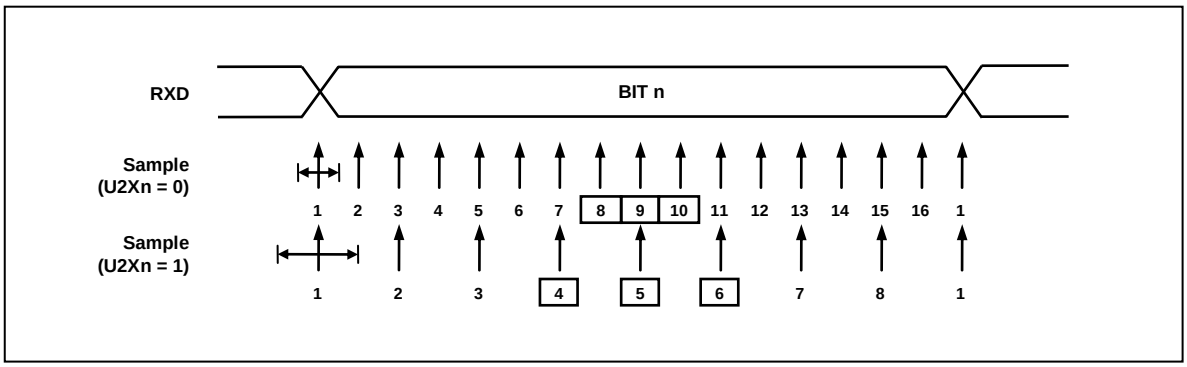


Figure 11.38 Sampling of Data and Parity Bit (where n = 2,3, and 4)

The process for detecting stop bit is like clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (FEn) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXD line to check a valid high to low transition is detected (start bit detection).

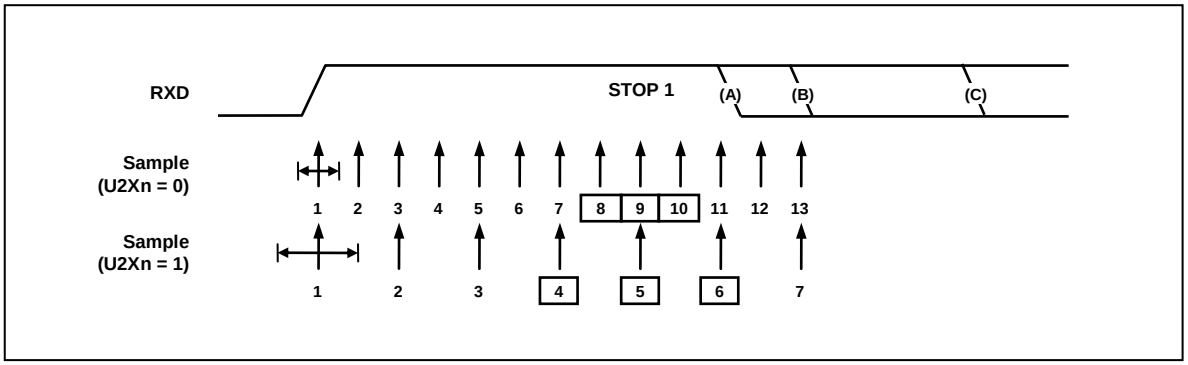


Figure 11.39 Stop Bit Sampling and Next Start Bit Sampling (where n = 2,3, and 4)

11.10.8 Register Map

Table 11-15 UART Register Map (where n = 2,3, and 4)

Name	Address	Dir	Default	Description
UARTnBD	103CH/1044H/104CH (XSFR)	R/W	FFH	UARTn Baud Rate Generation Register
UARTnDR	103DH/1045H/104DH (XSFR)	R/W	00H	UARTn Data Register
UARTnCR1	1038H/1040H/1048H (XSFR)	R/W	00H	UARTn Control Register 1
UARTnCR2	1039H/1041H/1049H (XSFR)	R/W	00H	UARTn Control Register 2
UARTnCR3	103AH/1042H/104AH (XSFR)	R/W	00H	UARTn Control Register 3
UARTnST	103BH/1043H/104BH (XSFR)	R/W	80H	UARTn Status Register

11.10.9 UART Register Description

UART2/3/4 module consists of UART2/3/4 baud rate generation register (UARTnBD), UART2/3/4 data register (UARTnDR), UART2/3/4 control register 1 (UARTnCR1), UART2/3/4 control register 2 (UARTnCR2) ,UART2/3/4 control register 3 (UARTnCR3), and UART2/3/4 status register (UARTnST).

11.10.10 Register Description for UART2/3/4

UARTnBD (UARTn Baud Rate Generation Register) : 103CH/1044H/104CH (XSFR), Where n = 2, 3, and 4

7	6	5	4	3	2	1	0
UARTnBD7	UARTnBD6	UARTnBD5	UARTnBD4	UARTnBD3	UARTnBD2	UARTnBD1	UARTnBD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

UARTnBD [7:0] The value in this register is used to generate internal baud rate. To prevent malfunction, do not write ‘0’.

UARTnDR (UARTn Data Register) : 103DH/1045H/104DH (XSFR), Where n = 2, 3, and 4

7	6	5	4	3	2	1	0
UARTnDR7	UARTnDR6	UARTnDR5	UARTnDR4	UARTnDR3	UARTnDR2	UARTnDR1	UARTnDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

UARTnDR [7:0] The UARTn Transmit Buffer and Receive Buffer share the same I/O address with this DATA register. The Transmit Data Buffer is the destination for data written to the UARTnDR register. Reading the UARTnDR register returns the contents of the Receive Buffer.

Write this register only when the UDREn flag is set.

UARTnCR1 (UARTn Control Register 1) : 1038H/1040H/1048H (XSFR), Where n = 2, 3, and 4

7	6	5	4	3	2	1	0
-	-	UnPM1	UnPM0	UnSIZE2	UnSIZE1	UnSIZE0	-
-	-	R/W	R/W	R/W	R/W	R/W	-

Initial value : 00H

UnPM[1:0] Selects Parity Generation and Check methods

UnPM1	UnPM0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

UnSIZE[2:0] Selects the Length of Data Bits in Frame

UnSIZE2	UnSIZE1	UnSIZE0	Data Length
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	1	1	9 bit
Other values			Reserved

UARTnCR2 (UARTn Control Register 2) : 1039H/1041H/1049H (XSFR), Where n = 2, 3, and 4

7	6	5	4	3	2	1	0
UDRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn	UARTnEN	U2Xn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

UDRIEn Interrupt enable bit for UARTn Data Register Empty

0 Interrupt from UDREn is inhibited (use polling)

1 When UDREn is set, request an interrupt

TXCIEn Interrupt enable bit for Transmit Complete

0 Interrupt from TXCn is inhibited (use polling)

1 When TXCn is set, request an interrupt

RXCIEn Interrupt enable bit for Receive Complete

0 Interrupt from RXCn is inhibited (use polling)

1 When RXCn is set, request an interrupt

WAKEIEn Interrupt enable bit for Wake in STOP mode. When device is in stop mode, if RXDn goes to LOW level an interrupt can be requested to wake-up system. At that time the UDRIEn bit and UARTnST register value should be set to ‘0b’ and “00H”, respectively.

0 Interrupt from Wake is inhibited

1 When WAKEn is set, request an interrupt

TXEn Enables the transmitter unit

0 Transmitter is disabled

1 Transmitter is enabled

RXEn Enables the receiver unit

0 Receiver is disabled

1 Receiver is enabled

UARTnEN Activate UARTn module by supplying clock. When one of TXEn and RXEn values is “1”, the UARTnEN bit always set to “1”.

0 UARTn is disabled (clock is halted)

1 UARTn is enabled

U2Xn This bit selects receiver sampling rate.

0 Normal operation

1 Double Speed operation

UARTnCR3 (UARTn Control Register 3) : 103AH/1042H/104AH (XSFR), Where n = 2, 3, and 4

7	6	5	4	3	2	1	0
-	LOOPSn	-	-	-	USBSn	UnTX8	UnRX8
-	R/W	-	-	-	R/W	R/W	R

Initial value : 00H

LOOPSn Controls the Loop Back Mode of UARTn, for test mode

0 Normal operation

1 Loop Back mode

USBSn Selects the length of stop bit.

0 1 Stop Bit

1 2 Stop Bit

UnTX8 The ninth bit of data frame in UARTn. Write this bit first before loading the UARTnDR register

0 MSB (9^{th} bit) to be transmitted is '0'

1 MSB (9^{th} bit) to be transmitted is '1'

UnRX8 The ninth bit of data frame in UARTn. Read this bit first before reading the receive buffer

0 MSB (9^{th} bit) received is '0'

1 MSB (9^{th} bit) received is '1'

UARTnST (UARTn Status Register) : 103BH/1043H/104BH (XSFR), Where n = 2, 3, and 4

7	6	5	4	3	2	1	0
UDREn	TXCn	RXCn	WAKEn	SOFTRSTn	DORn	FEn	PEn
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Initial value : 80H

UDREn The UDREn flag indicates if the transmit buffer (UARTnDR) is ready to receive new data. If UDREn is '1', the buffer is empty and ready to be written. This flag can generate a UDREn interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXCn This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXCn interrupt is executed. This flag can generate a TXCn interrupt.

0 Transmission is ongoing.

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely.

RXCn This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXCn flag can be used to generate a RXCn interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKEn This flag is set when the RXDn pin is detected low while the CPU is in stop mode. This flag can be used to generate a WAKEn interrupt. This bit should be cleared by program software.

0 No WAKEn interrupt is generated.

1 WAKEn interrupt is generated.

SOFTRSTn This is an internal reset and only has effect on UARTn. Writing '1' to this bit initializes the internal logic of UARTn and this bit is automatically cleared.

0 No operation

1 Reset UARTn

DORn This bit is set if a Data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data OverRun

1 Data OverRun detected

FEn This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read.

0 No Frame Error

1 Frame Error detected

PEn This bit is set if the next character in the receive buffer has a Parity Error to be received while Parity Checking is enabled. This bit is valid until the receive buffer is read.

0 No Parity Error

1 Parity Error detected

11.10.11 Baud Rate setting (example)

Table 11-16 Examples of UARTnBD Settings for Commonly Used Oscillator Frequencies

Baud Rate	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	UARTnBD	ERROR	UARTnBD	ERROR	UARTnBD	ERROR
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	-	-	1	-25.0%	1	8.5%
76.8k	-	-	1	0.0%	1	-18.6%
115.2k	-	-	-	-	-	-
230.4k	-	-	-	-	-	-

(continued)

Baud Rate	fx=3.6864MHz		fx=4.00MHz		fx=7.3728MHz	
	UARTnBD	ERROR	UARTnBD	ERROR	UARTnBD	ERROR
2400	95	0.0%	103	0.2%	191	0.0%
4800	47	0.0%	51	0.2%	95	0.0%
9600	23	0.0%	25	0.2%	47	0.0%
14.4k	15	0.0%	16	2.1%	31	0.0%
19.2k	11	0.0%	12	0.2%	23	0.0%
28.8k	7	0.0%	8	-3.5%	15	0.0%
38.4k	5	0.0%	6	-7.0%	11	0.0%
57.6k	3	0.0%	3	8.5%	7	0.0%
76.8k	2	0.0%	2	8.5%	5	0.0%
115.2k	1	0.0%	1	8.5%	3	0.0%
230.4k	-	-	-	-	1	0.0%
250k	-	-	-	-	1	-7.8%
0.5M	-	-	-	-	-	-

(continued)

Baud Rate	fx=8.00MHz		fx=11.0592MHz	
	UARTnBD	ERROR	UARTnBD	ERROR
2400	207	0.2%	-	-
4800	103	0.2%	143	0.0%
9600	51	0.2%	71	0.0%
14.4k	34	-0.8%	47	0.0%
19.2k	25	0.2%	35	0.0%
28.8k	16	2.1%	23	0.0%
38.4k	12	0.2%	17	0.0%
57.6k	8	-3.5%	11	0.0%
76.8k	6	-7.0%	8	0.0%
115.2k	3	8.5%	5	0.0%
230.4k	1	8.5%	2	0.0%
250k	1	0.0%	2	-7.8%
0.5M	-	-	-	-
1M	-	-	-	-

Where n = 2,3, and 4

11.11 USI0/1 (UART + SPI + I2C)

11.11.1 Overview

The USI0/1 consists of USI0/1 control register1/2/3/4, USI0/1 status register 1/2, USI0/1 baud-rate generation register, USI0/1 data register, USI0/1 SDA hold time register, USI0/1 SCL high period register, USI0/1 SCL low period register, and USI0/1 slave address register (USInCR1, USInCR2, USInCR3, USInCR4, USInST1, USInST2, USInBD, USInDR, USInSDHR, USInSCHR, USInSCLR, USInSAR).

The operation mode is selected by the operation mode of USI0/1 selection bits (USInMS[1:0]).

It has four operating modes:

- Asynchronous mode (UART)
- Synchronous mode
- SPI mode
- I2C mode

11.11.2 USI0/1 UART Mode

The universal synchronous and asynchronous serial receiver and transmitter (UART) is a highly flexible serial communication device. The main features are listed below.

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Baud Rate Generator
- Supports Serial Frames with 5,6,7,8, or 9 Data Bits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Double Speed Asynchronous communication mode

USI0/1 has three main parts of clock generator, Transmitter and receiver. The clock generation logic consists of synchronization logic for external clock input used by synchronous or SPI slave operation, and the baud rate generator for asynchronous or master (synchronous or SPI) operation.

The Transmitter consists of a single write buffer, a serial shift register, parity generator and control logic for handling different serial frame formats. The write buffer allows continuous transfer of data without any delay between frames. The receiver is the most complex part of the UART module due to its clock and data recovery units. The recovery unit is used for asynchronous data reception. In addition to the recovery unit, the receiver includes a parity checker, a shift register, a two-level receive FIFO (USInDR) and control logic. The receiver supports the same frame formats as the transmitter and can detect frame error, data overrun and parity errors.

11.11.3 USI0/1 UART Block Diagram

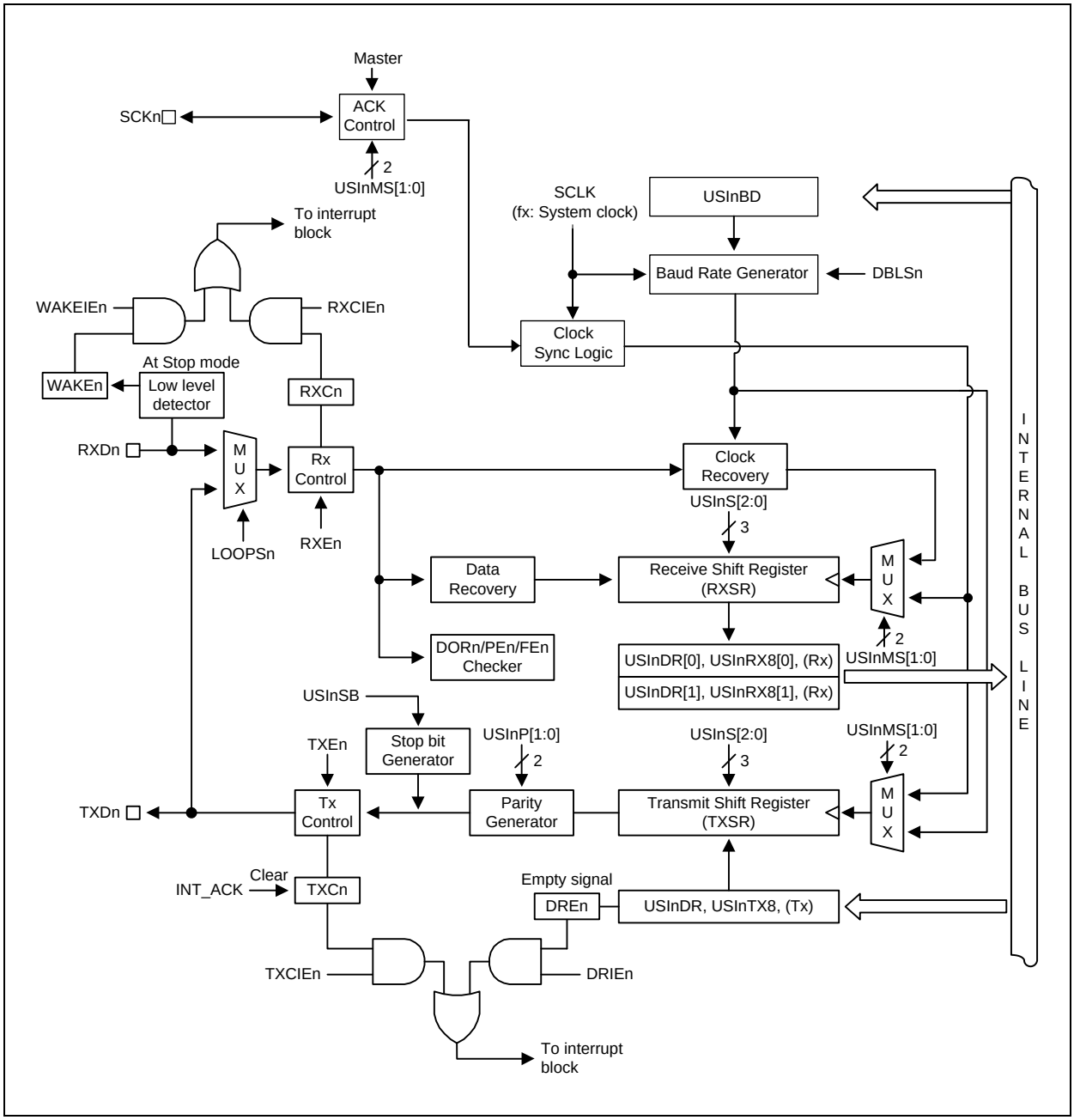


Figure 11.40 USI0/1 UART Block Diagram (Where n = 0 and 1)

11.11.4 USI0/1 Clock Generation

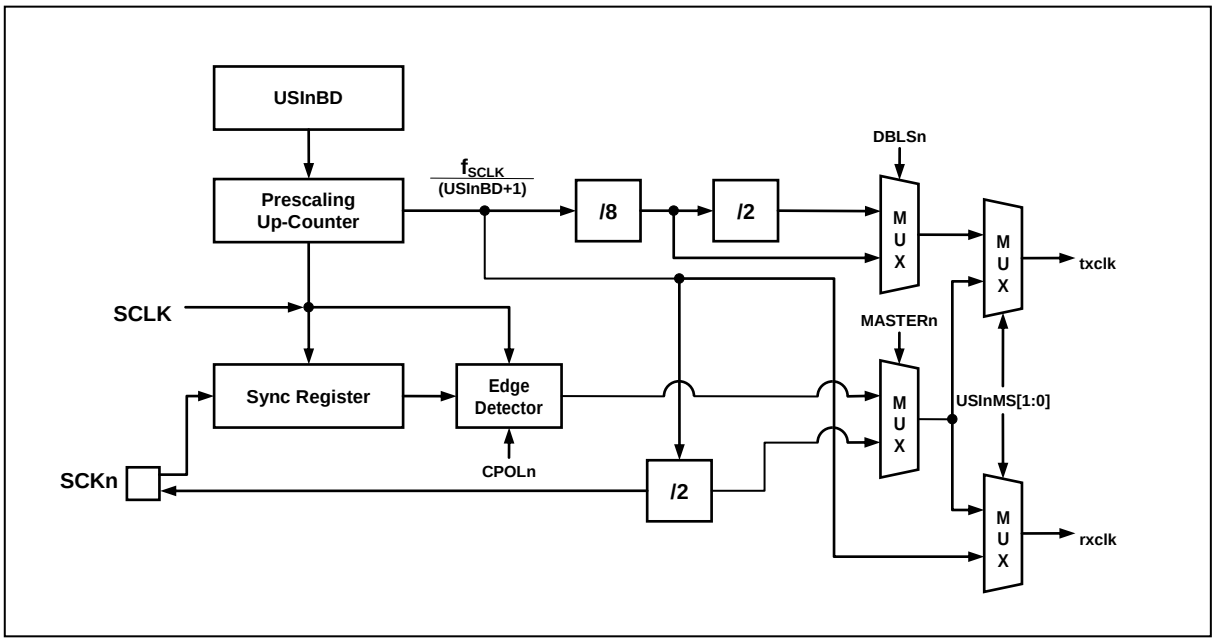


Figure 11.41 Clock Generation Block Diagram (USIn, where n = 0 and 1)

The clock generation logic generates the base clock for the transmitter and receiver. The USI0/1 supports four modes of clock operation and those are normal asynchronous, double speed asynchronous, master synchronous and slave synchronous mode. The clock generation scheme for master SPI and slave SPI mode is the same as master synchronous and slave synchronous operation mode. The USInMS[1:0] bits in USInCR1 register selects asynchronous or synchronous operation. Asynchronous double speed mode is controlled by the DBLSn bit in the USInCR2 register. The MASTERn bit in USInCR3 register controls whether the clock source is internal (master mode, output pin) or external (slave mode, input pin). The SCKn pin is active only when the USI0/1 operates in synchronous or SPI mode.

Following table shows the equations for calculating the baud rate (in bps).

Table 11-17 Equations for Calculating USI0/1 Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate
Asynchronous Normal Mode (DBLSn=0)	$\text{Baud Rate} = \frac{\text{fx}}{16(\text{USInBD}+1)}$
Asynchronous Double Speed Mode (DBLSn=1)	$\text{Baud Rate} = \frac{\text{fx}}{8(\text{USInBD}+1)}$
Synchronous or SPI Master Mode	$\text{Baud Rate} = \frac{\text{fx}}{2(\text{USInBD}+1)}$

11.11.5 USI0/1 External Clock (SCKn)

External clocking is used in the synchronous mode of operation.

External clock input from the SCKn pin is sampled by a synchronization logic to remove meta-stability. The output from the synchronization logic must be passed through an edge detector before it is used by the transmitter and receiver. This process introduces two CPU clock period delay. The maximum frequency of the external SCKn pin is limited up-to 1MHz.

11.11.6 USI0 Synchronous mode operation

When synchronous or SPI mode is used, the SCKn pin will be used as either clock input (slave) or clock output (master). Data sampling and transmitter is issued on the different edge of SCKn clock each other. For example, if data input on RXDn (MISOn in SPI mode) pin is sampled on the rising edge of SCKn clock, data output on TXDn (MOSIn in SPI mode) pin is altered on the falling edge.

The CPOLn bit in USInCR1 register selects which SCKn clock edge is used for data sampling and which is used for data change. As shown in the figure below, when CPOLn is zero, the data will be changed at rising SCKn edge and sampled at falling SCKn edge.

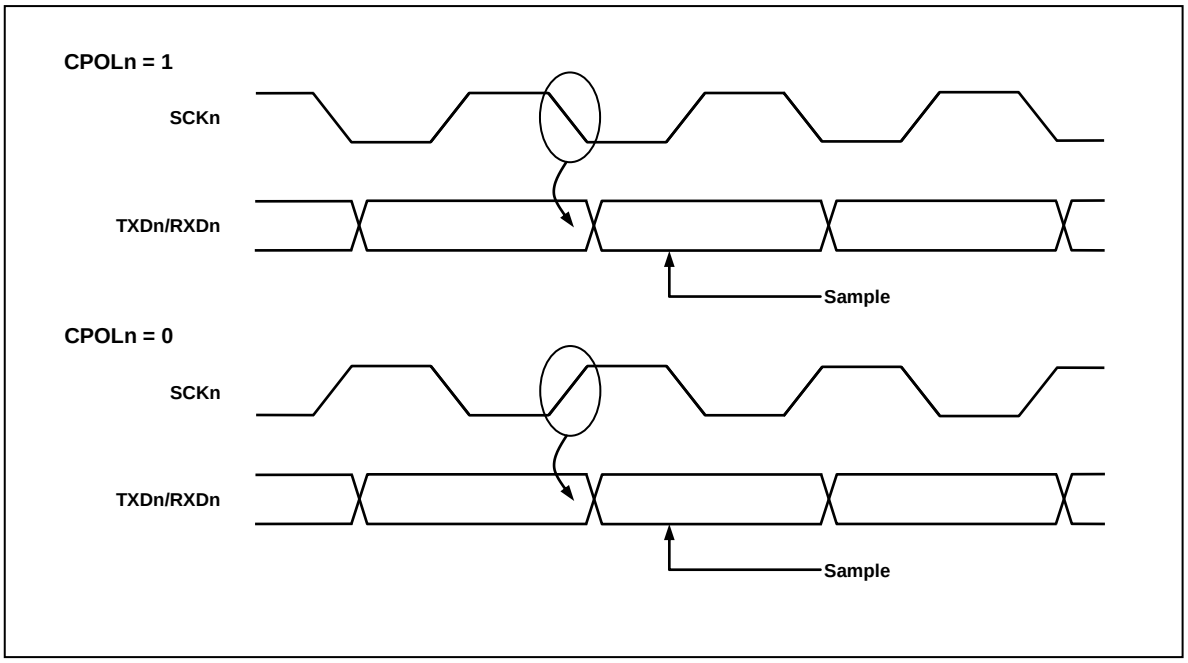


Figure 11.42 Synchronous Mode SCKn Timing (USIn , where n = 0 and 1)

11.11.7 USI0/1 UART Data format

A serial frame is defined to be one character of data bits with synchronization bits (start and stop bits), and optionally a parity bit for error detection.

The UART supports all 30 combinations of the following as valid frame formats.

- 1 start bit
- 5, 6, 7, 8 or 9 data bits
- no, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the start bit followed by the least significant data bit (LSB). Then the next data bits, up to nine, are succeeding, ending with the most significant bit (MSB). If parity function is enabled, the parity bit is inserted between the last data bit and the stop bit. A high-to-low transition on data pin is considered as start bit. When a complete frame is transmitted, it can be directly followed by a new frame, or the communication line can be set to an idle state. The idle means high state of data pin. The following figure shows the possible combinations of the frame formats. Bits inside brackets are optional.

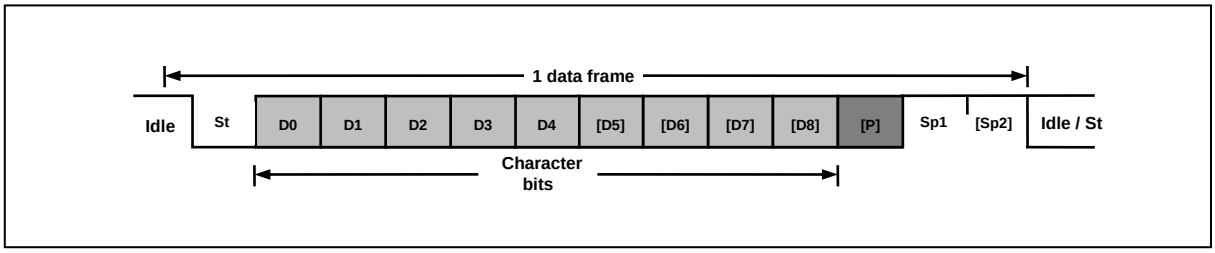


Figure 11.43 Frame Format (USI0/1)

1 data frame consists of the following bits

- Idle No communication on communication line (TXDn/RXDn)
- St Start bit (Low)
- Dn Data bits (0~8)
- Parity bit ---------- Even parity, Odd parity, No parity
- Stop bit(s) ---------- 1 bit or 2 bits

The frame format used by the UART is set by the USInS[2:0], USInPM[1:0] bits in USInCR1 register and USInSB bit in USInCR3 register. The Transmitter and Receiver use the same setting.

11.11.8 USI0/1 UART Parity bit

The parity bit is calculated by doing an exclusive-OR of all the data bits. If odd parity is used, the result of the exclusive-O is inverted. The parity bit is located between the MSB and first stop bit of a serial frame.

$P_{even} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 0$

$P_{odd} = D_{n-1} \wedge \ldots \wedge D_3 \wedge D_2 \wedge D_1 \wedge D_0 \wedge 1$

P_{even} : Parity bit using even parity

P_{odd} : Parity bit using odd parity

D_n : Data bit n of the character

11.11.9 USI0/1 UART Transmitter

The UART transmitter is enabled by setting the TXEn bit in USInCR2 register. When the Transmitter is enabled, the TXDn pin should be set to TXDn function for the serial output pin of UART by the P6FSR2/P4FSR0. The baud-rate, operation mode and frame format must be setup once before doing any transmission. In synchronous operation mode, the SCKn pin is used as transmission clock, so it should be selected to do SCKn function by P6FSR0/P4FSR2.

11.11.9.1 USI0/1 UART Sending Tx data

A data transmission is initiated by loading the transmit buffer (USInDR register I/O location) with the data to be transmitted. The data written in transmit buffer is moved to the shift register when the shift register is ready to send a new frame. The shift register is loaded with the new data if it is in idle state or immediately after the last stop bit of the previous frame is transmitted. When the shift register is loaded with new data, it will transfer one complete frame according to the settings of control registers. If the 9-bit characters are used in asynchronous or synchronous operation mode, the ninth bit must be written to the USInTX8 bit in USInCR3 register before it is loaded to the transmit buffer (USInDR register).

11.11.9.2 USI0/1 UART Transmitter flag and interrupt

The UART transmitter has 2 flags which indicate its state. One is UART data register empty flag (DREn) and the other is transmit complete flag (TXCn). Both flags can be interrupt sources.

DREn flag indicates whether the transmit buffer is ready to receive new data. This bit is set when the transmit buffer is empty and cleared when the transmit buffer contains data to be transmitted but has not yet been moved into the shift register. And also this flag can be cleared by writing ‘0’ to this bit position. Writing ‘1’ to this bit position is prevented.

When the data register empty interrupt enable (DRIEn) bit in USInCR2 register is set and the global interrupt is enabled, USInST1 status register empty interrupt is generated while DREn flag is set.

The transmit complete (TXCn) flag bit is set when the entire frame in the transmit shift register has been shifted out and there is no more data in the transmit buffer. The TXCn flag is automatically cleared when the transmit complete interrupt service routine is executed, or it can be cleared by writing ‘0’ to TXCn bit in USInST1 register.

When the transmit complete interrupt enable (TXCIEn) bit in USInCR2 register is set and the global interrupt is enabled, UART transmit complete interrupt is generated while TXCn flag is set.

11.11.9.3 USI0/1 UART Parity Generator

The parity generator calculates the parity bit for the serial frame data to be sent. When parity bit is enabled (USInPM1=1), the transmitter control logic inserts the parity bit between the MSB and the first stop bit of the frame to be sent.

11.11.9.4 USI0/1 UART Disabling Transmitter

Disabling the transmitter by clearing the TXEn bit will not become effective until ongoing transmission is completed. When the Transmitter is disabled, the TXDn pin can be used as a normal general purpose I/O (GPIO).

11.11.10 USI0/1 UART Receiver

The UART receiver is enabled by setting the RXEn bit in the USInCR2 register. When the receiver is enabled, the RXDn pin should be set to RXDn function for the serial input pin of UART by P6FSR1/P4FSR1. The baud-rate, mode of operation and frame format must be set before serial reception. In synchronous or SPI operation mode the SCKn pin is used as transfer clock, so it should be selected to do SCKn function by P6FSR0/P4FSR2. In SPI operation mode the SSn input pin in slave mode or can be configured as SSn output pin in master mode. This can be done by setting USInSSEN bit in USnCR3 register.

11.11.10.1 USI0/1 UART Receiving Rx data

When UART is in synchronous or asynchronous operation mode, the receiver starts data reception when it detects a valid start bit (LOW) on RXDn pin. Each bit after start bit is sampled at pre-defined baud-rate (asynchronous) or sampling edge of SCKn (synchronous), and shifted into the receive shift register until the first stop bit of a frame is received. Even if there's 2^{nd} stop bit in the frame, the 2^{nd} stop bit is ignored by the receiver. That is, receiving the first stop bit means that a complete serial frame is present in the receiver shift register and contents of the shift register are to be moved into the receive buffer. The receive buffer is read by reading the USInDR register.

If 9-bit characters are used (USInS[2:0] = "111"), the ninth bit is stored in the USInRX8 bit position in the USInCR3 register. The 9^{th} bit must be read from the USInRX8 bit before reading the low 8 bits from the USInDR register. Likewise, the error flags FEn, DORn, PEn must be read before reading the data from USInDR register. It's because the error flags are stored in the same FIFO position of the receive buffer.

11.11.10.2 USI0/1 UART Receiver Flag and Interrupt

The UART receiver has one flag that indicates the receiver state.

The receive complete (RXCn) flag indicates whether there are unread data in the receive buffer. This flag is set when there are unread data in the receive buffer and cleared when the receive buffer is empty. If the receiver is disabled (RXEn=0), the receiver buffer is flushed and the RXCn flag is cleared.

When the receive complete interrupt enable (RXCIEn) bit in the USInCR2 register is set and global interrupt is enabled, the UART receiver complete interrupt is generated while RXCn flag is set.

The UART receiver has three error flags which are frame error (FEn), data overrun (DORn) and parity error (PEn). These error flags can be read from the USInST1 register. As received data are stored in the 2-level receive buffer, these error flags are also stored in the same position of receive buffer. So, before reading received data from USInDR register, read the USInST1 register first which contains error flags.

The frame error (FEn) flag indicates the state of the first stop bit. The FEn flag is '0' when the stop bit was correctly detected as "1", and the FEn flag is "1" when the stop bit was incorrect, i.e. detected as "0". This flag can be used for detecting out-of-sync conditions between data frames.

The data overrun (DORn) flag indicates data loss due to a receive buffer full condition. DORn occurs when the receive buffer is full, and another new data is present in the receive shift register which are to be stored into the receive buffer. After the DORn flag is set, all the incoming data are lost. To prevent data loss or clear this flag, read the receive buffer.

The parity error (PEn) flag indicates that the frame in the receive buffer had a parity error when received. If parity check function is not enabled (USInPM1=0), the PEn bit is always read "0".

11.11.10.3 USI0/1 UART Parity Checker

If parity bit is enabled (USInPM1=1), the Parity Checker calculates the parity of the data bits in incoming frame and compares the result with the parity bit from the received serial frame.

11.11.10.4 USI0/1 UART Disabling Receiver

In contrast to transmitter, disabling the Receiver by clearing RXEn bit makes the Receiver inactive immediately. When the receiver is disabled, the receiver flushes the receive buffer, the remaining data in the buffer is all reset, and the RXDn pin can be used as a normal general purpose I/O (GPIO).

11.11.10.5 USI0/1 Asynchronous Data Reception

To receive asynchronous data frame, the UART includes a clock and data recovery unit. The clock recovery logic is used for synchronizing the internally generated baud-rate clock to the incoming asynchronous serial frame on the RXDn pin.

The data recovery logic samples and low pass filters the incoming bits, and this removes the noise of RXDn pin.

The next figure illustrates the sampling process of the start bit of an incoming frame. The sampling rate is 16 times of the baud-rate in normal mode and 8 times the baud-rate for double speed mode (DBLSn=1). The horizontal arrows show the synchronization variation due to the asynchronous sampling process. Note that larger time variation is shown when using the double speed mode.

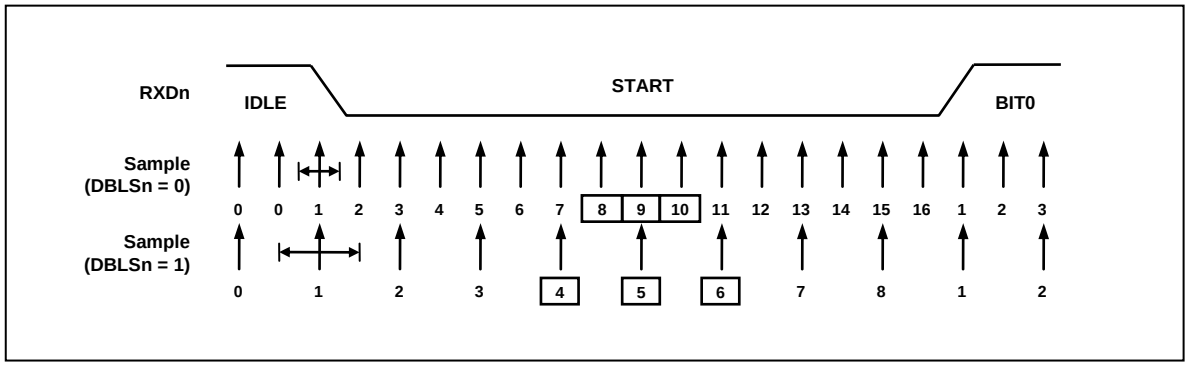


Figure 11.44 Asynchronous Start Bit Sampling (USIn, where n = 0 and 1)

When the receiver is enabled (RXEn=1), the clock recovery logic tries to find a high-to-low transition on the RXDn line, the start bit condition. After detecting high to low transition on RXDn line, the clock recovery logic uses samples 8, 9 and 10 for normal mode to decide if a valid start bit is received. If more than 2 samples have logical low level, it is considered that a valid start bit is detected and the internally generated clock is synchronized to the incoming data frame. And the data recovery can begin. The synchronization process is repeated for each start bit.

As described above, when the receiver clock is synchronized to the start bit, the data recovery can begin. Data recovery process is almost similar to the clock recovery process. The data recovery logic samples 16 times for each incoming bits for normal mode and 8 times for double speed mode, and uses sample 8, 9 and 10 to decide data value. If more than 2 samples have low levels, the received bit is considered to a logic ‘0’ and if more than 2 samples have high levels, the received bit is considered to a logic ‘1’. The data recovery process is then repeated until a complete frame is received including the first stop bit. The decided bit value is stored in the receive shift register in order. Note that the Receiver only uses the first stop bit of a frame. Internally, after receiving the first stop bit, the Receiver is in idle state and waiting to find start bit.

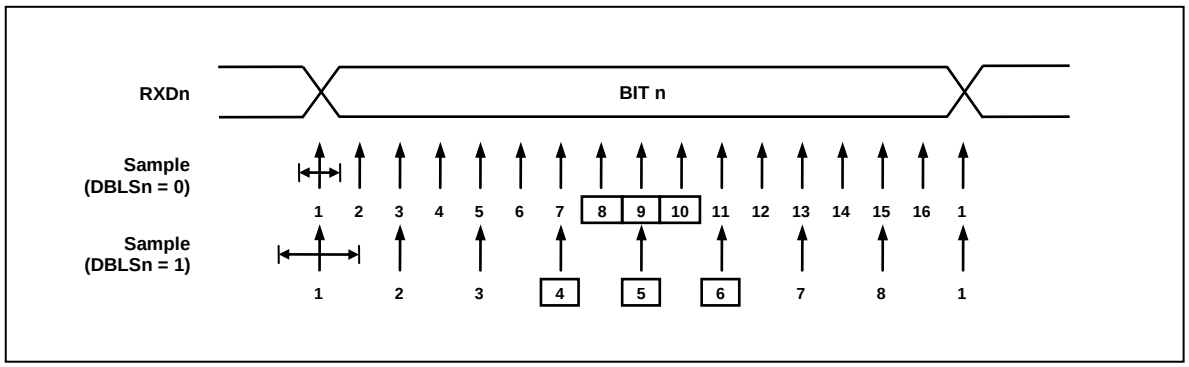


Figure 11.45 Asynchronous Sampling of Data and Parity Bit (USIn, where n = 0 and 1)

The process for detecting stop bit is like clock and data recovery process. That is, if 2 or more samples of 3 center values have high level, correct stop bit is detected, else a frame error (FEn) flag is set. After deciding whether the first stop bit is valid or not, the Receiver goes to idle state and monitors the RXDn line to check a valid high to low transition is detected (start bit detection).

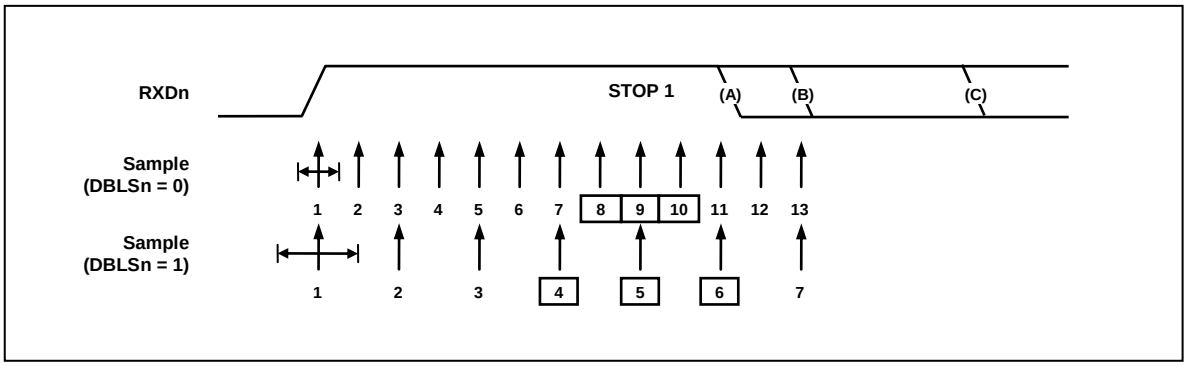


Figure 11.46 Stop Bit Sampling and Next Start Bit Sampling (USIn, where n = 0 and 1)

11.11.11 USI0/1 SPI Mode

The USI0/1 can be set to operate in industrial standard SPI compliant mode. The SPI mode has the following features.

- Full Duplex, Three-wire synchronous data transfer
- Mater and Slave Operation
- Supports all four SPI0 modes of operation (mode 0, 1, 2, and 3)
- Selectable LSB first or MSB first data transfer
- Double buffered transmit and receive
- Programmable transmit bit rate

When SPI mode is enabled (USInMS[1:0]=”11”), the slave select (SSn) pin becomes active LOW input in slave mode operation, or can be output in master mode operation if USInSSEN bit is set to ‘0’.

Note that during SPI mode of operation, the pin RXDn is renamed as MISOn and TXDn is renamed as MOSIn for compatibility to other SPI devices.

11.11.12 USI0/1 SPI Clock Formats and Timing

To accommodate a wide variety if synchronous serial peripherals from different manufacturers, the USI0/1 has a clock polarity bit (CPOLn) and a clock phase control bit (CPHAn) to select one of four clock formats for data transfers. CPOLn selectively insert an inverter in series with the clock. CPHAn chooses between two different clock phase relationships between the clock and data. Note that CPHAn and CPOLn bits in USInCR1 register have different meanings according to the USInMS[1:0] bits which decides the operating mode of USI0/1.

Table below shows four combinations of CPOLn and CPHAn for SPI mode 0, 1, 2, and 3.

Table 11-18 CPOLn Functionality (where n = 0 and 1)

SPI Mode	CPOLn	CPHAn	Leading Edge	Trailing Edge
0	0	0	Sample (Rising)	Setup (Falling)
1	0	1	Setup (Rising)	Sample (Falling)
2	1	0	Sample (Falling)	Setup (Rising)
3	1	1	Setup (Falling)	Sample (Rising)

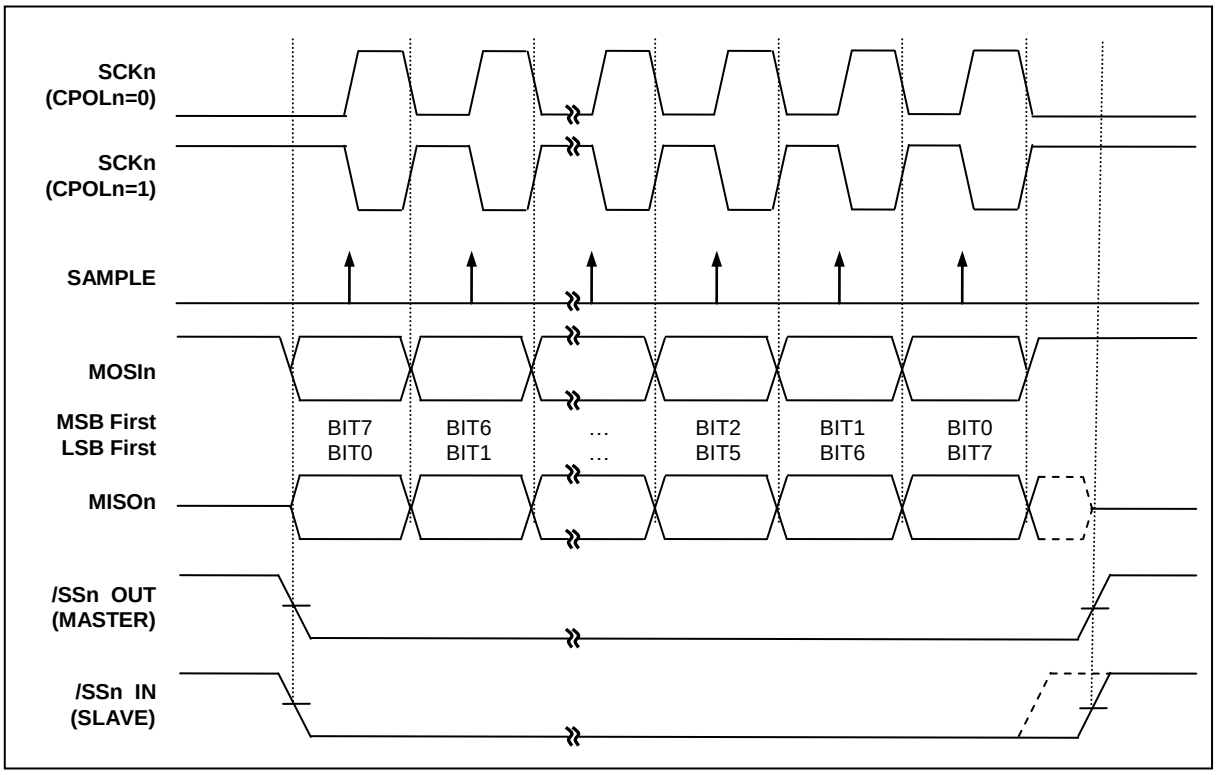


Figure 11.47 USI0/1 SPI Clock Formats when CPHAn=0 (where n = 0 and 1)

When CPHAn=0, the slave begins to drive its MISOn output with the first data bit value when SSn goes to active low. The first SCKn edge causes both the master and the slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the second SCKn edge, the USI0/1 shifts the second data bit value out to the MOSIn and MISOn outputs of the master and slave, respectively. Unlike the case of CPHAn=1, when CPHAn=0, the slave's SSn input must go to its inactive high level between transfers. This is because the slave can prepare the first data bit when it detects falling edge of SSn input.

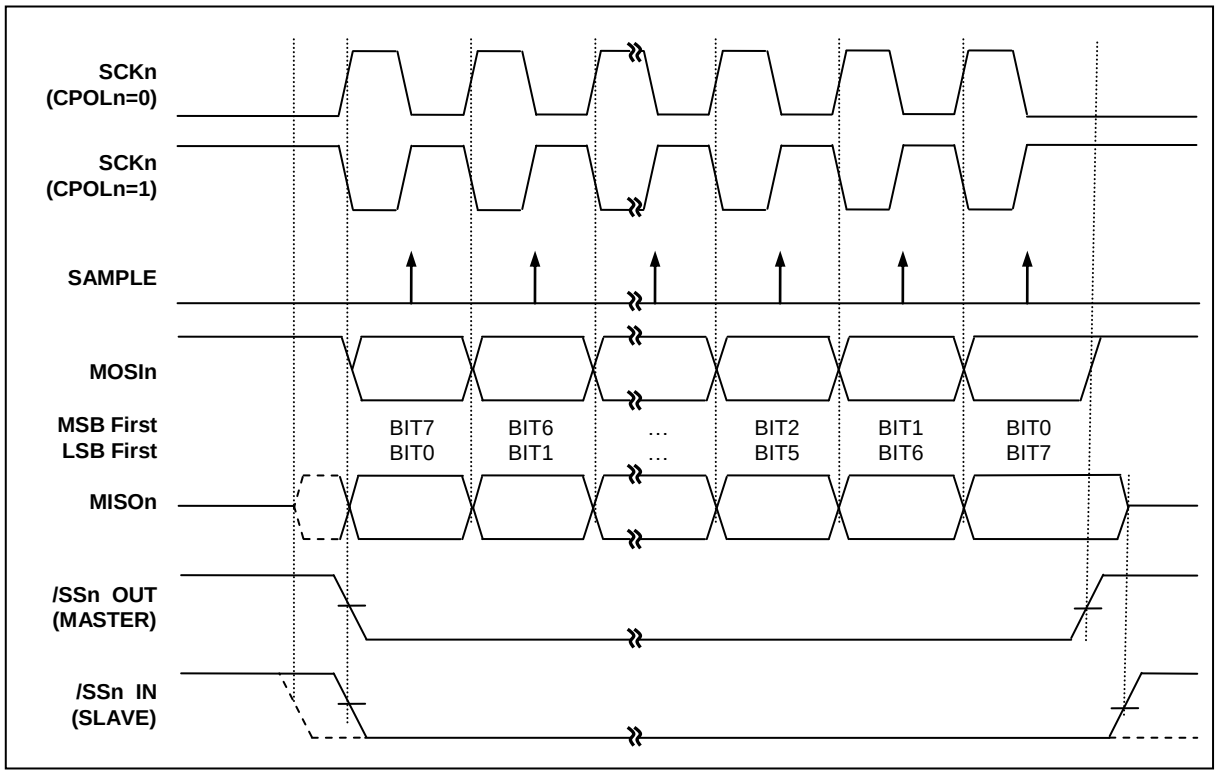


Figure 11.48 USI0/1 SPI Clock Formats when CPHAn=1 (where n = 0 and 1)

When CPHAn=1, the slave begins to drive its MISOn output when SSn goes active low, but the data is not defined until the first SCKn edge. The first SCKn edge shifts the first bit of data from the shifter onto the MOSIn output of the master and the MISOn output of the slave. The next SCKn edge causes both the master and slave to sample the data bit value on their MISOn and MOSIn inputs, respectively. At the third SCKn edge, the USI0/1 shifts the second data bit value out to the MOSIn and MISOn output of the master and slave respectively. When CPHAn=1, the slave's SSn input is not required to go to its inactive high level between transfers.

Because the SPI logic reuses the USI0/1 resources, SPI mode of operation is similar to that of synchronous or asynchronous operation. An SPI transfer is initiated by checking for the USI0/1 Data Register Empty flag (DREn=1) and then writing a byte of data to the USInDR Register. In master mode of operation, even if transmission is not enabled (TXEn=0), writing data to the USInDR register is necessary because the clock SCKn is generated from transmitter block.

11.11.13 USI0/1 SPI Block Diagram

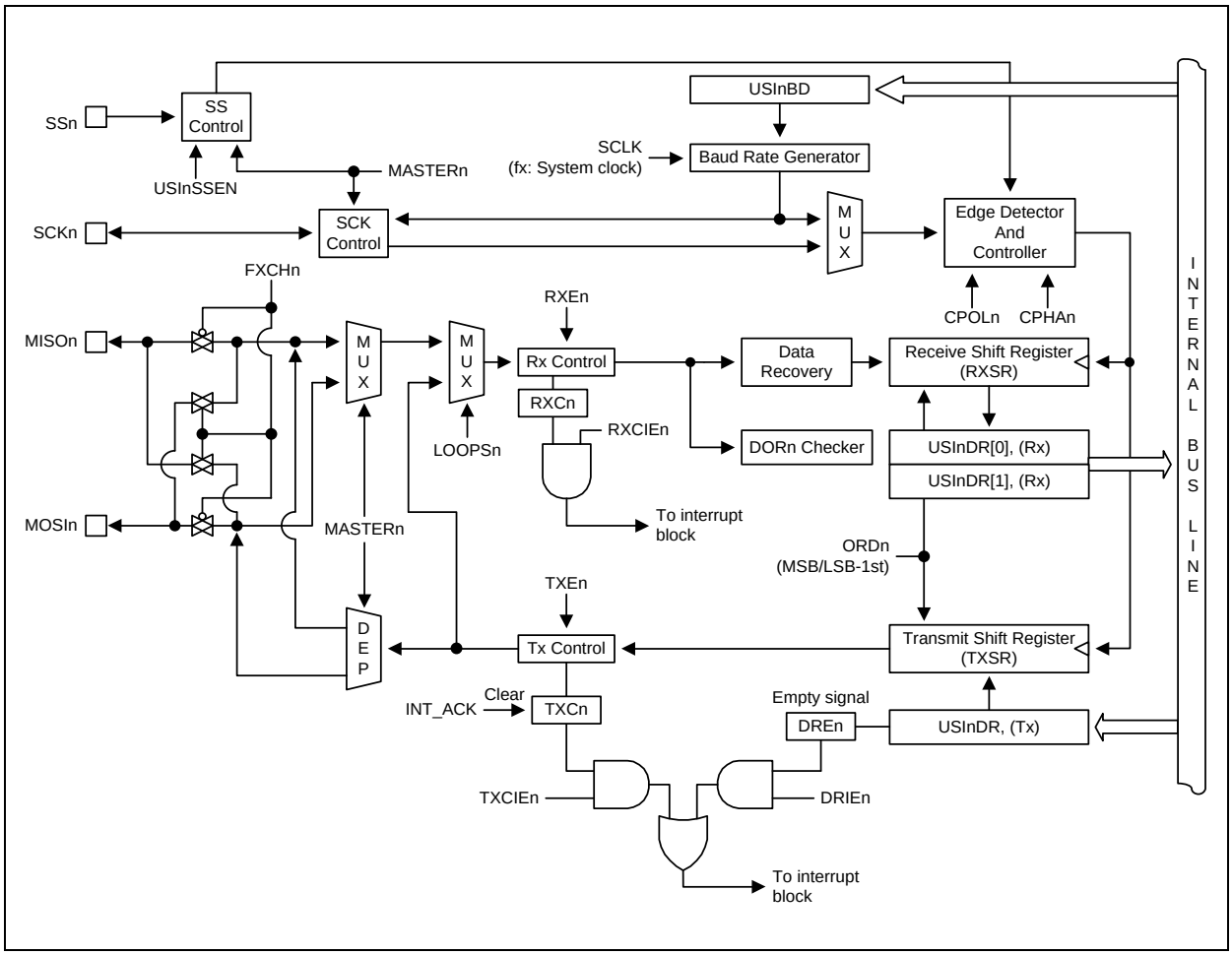


Figure 11.49 USI0/1 SPI Block Diagram (where n = 0 and 1)

11.11.14 USI0/1 I2C Mode

The USI0/1 can be set to operate in industrial standard serial communication protocols mode. The I2C mode uses 2 bus lines serial data line (SDAn) and serial clock line (SCLn) to exchange data. Because both SDAn and SCLn lines are open-drain output, each line needs pull-up resistor. The features are as shown below.

- Compatible with I2C bus standard
- Multi-master operation
- Up to 400kHz data transfer read speed
- 7 bit address
- Both master and slave operation
- Bus busy detection

11.11.15 USI0/1 I2C Bit Transfer

The data on the SDAn line must be stable during HIGH period of the clock, SCLn. The HIGH or LOW state of the data line can only change when the clock signal on the SCLn line is LOW. The exceptions are START(S), repeated START(Sr) and STOP(P) condition where data line changes when clock line is high.

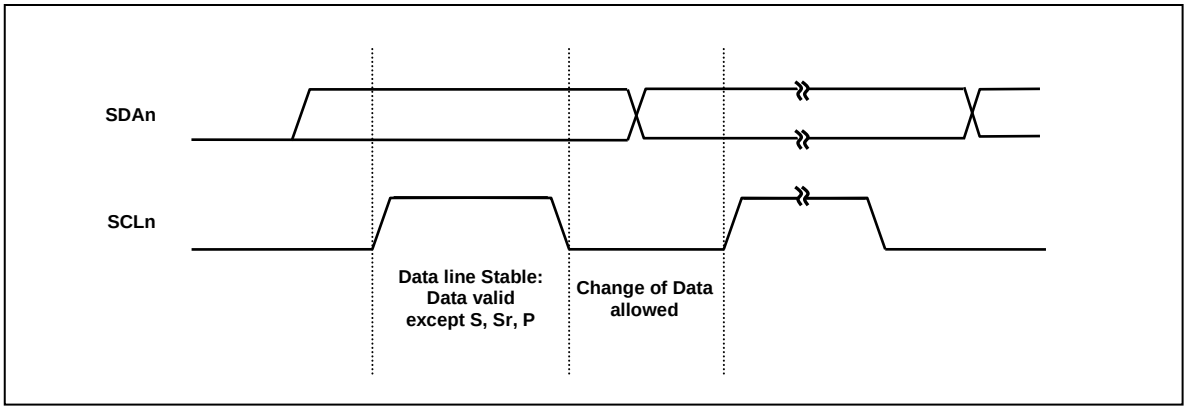


Figure 11.50 Bit Transfer on the I2C-Bus (USIn, where n = 0 and 1)

11.11.16 USI0/1 I2C Start / Repeated Start / Stop

One master can issue a START (S) condition to notice other devices connected to the SCLn, SDAn lines that it will use the bus. A STOP (P) condition is generated by the master to release the bus lines so that other devices can use it.

A high to low transition on the SDAn line while SCLn is high defines a START (S) condition.
A low to high transition on the SDAn line while SCLn is high defines a STOP (P) condition.

START and STOP conditions are always generated by the master. The bus is considered to be busy after START condition. The bus is considered to be free again after STOP condition, ie, the bus is busy between START and STOP condition. If a repeated START condition (Sr) is generated instead of STOP condition, the bus stays busy. So, the START and repeated START conditions are functionally identical.

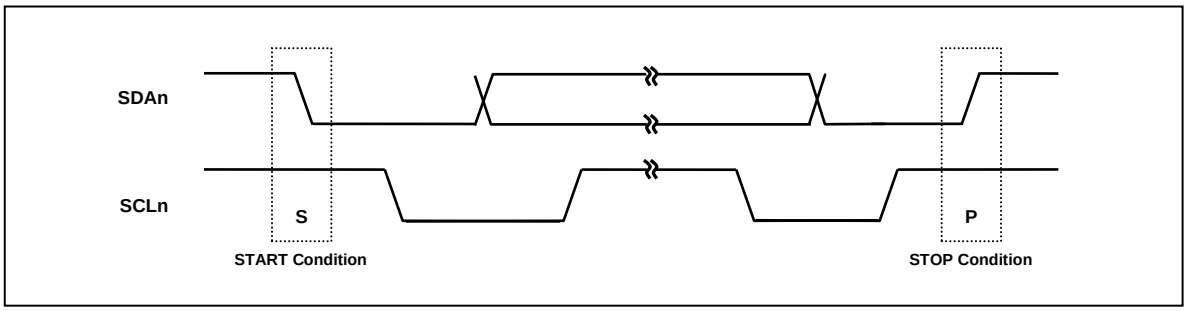


Figure 11.51 START and STOP Condition (USIn, where n = 0 and 1)

11.11.17 USI0/1 I2C Data Transfer

Every byte put on the SDAn line must be 8-bits long. The number of bytes that can be transmitted per transfer is unlimited. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first. If a slave can't receive or transmit another complete byte of data until it has performed some other function, it can hold the clock line SCLn LOW to force the master into a wait state. Data transfer then continues when the slave is ready for another byte of data and releases clock line SCLn.

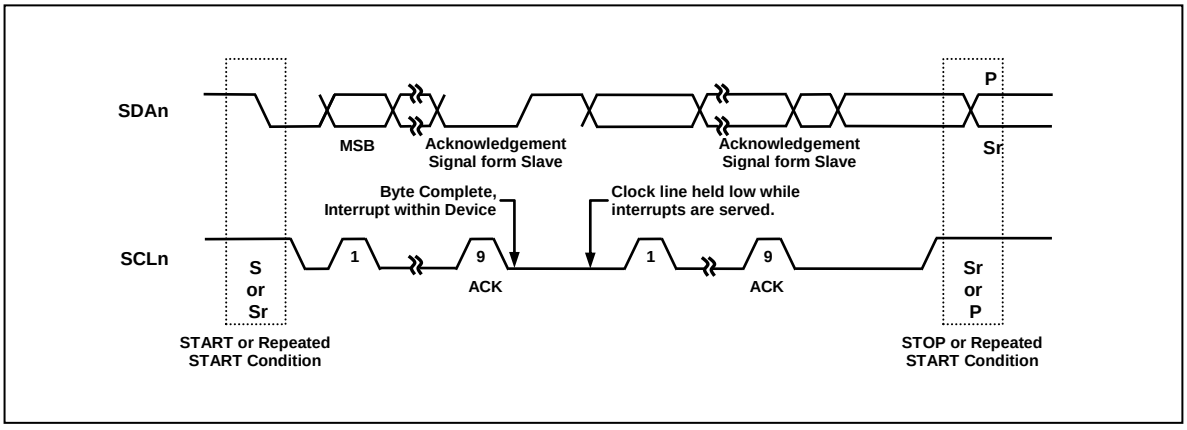


Figure 11.52 Data Transfer on the I2C-Bus (USIn, where n = 0 and 1)

11.11.18 USI0/1 I2C Acknowledge

The acknowledge related clock pulse is generated by the master. The transmitter releases the SDAn line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDAn line during the acknowledge clock pulse so that it remains stable LOW during the HIGH period of this clock pulse. When a slave is addressed by a master (Address Packet), and if it is unable to receive or transmit because it's performing some real time function, the data line must be left HIGH by the slave. And also, when a slave addressed by a master is unable to receive more data bits, the slave receiver must release the SDAn line (Data Packet). The master can then generate either a STOP condition to abort the transfer, or a repeated START condition to start a new transfer.

If a master receiver is involved in a transfer, it must signal the end of data to the slave transmitter by not generating an acknowledge on the last byte that was clocked out of the slave. The slave transmitter must release the data line to allow the master to generate a STOP or repeated START condition.

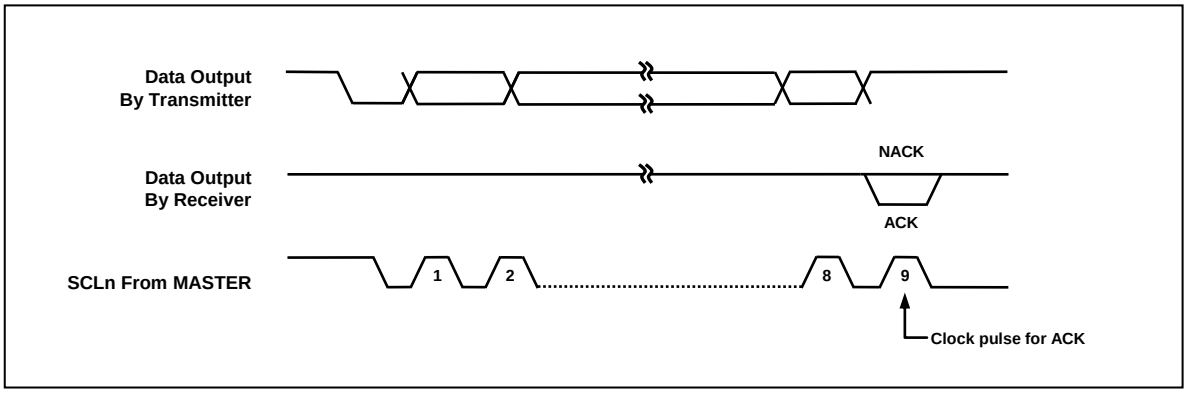


Figure 11.53 Acknowledge on the I2C-Bus (USIn, where n = 0 and 1)

11.11.19 USI0/1 I2C Synchronization / Arbitration

Clock synchronization is performed using the wired-AND connection of I2C interfaces to the SCLn line. This means that a HIGH to LOW transition on the SCLn line will cause the devices concerned to start counting off their LOW period and it will hold the SCLn line in that state until the clock HIGH state is reached. However the LOW to HIGH transition of this clock may not change the state of the SCLn line if another clock is still within its LOW period. In this way, a synchronized SCLn clock is generated with its LOW period determined by the device with the longest clock LOW period, and its HIGH period determined by the one with the shortest clock HIGH period.

A master may start a transfer only if the bus is free. Two or more masters may generate a START condition. Arbitration takes place on the SDAn line, while the SCLn line is at the HIGH level, in such a way that the master which transmits a HIGH level, while another master is transmitting a LOW level will switch off its DATA output state because the level on the bus doesn't correspond to its own level. Arbitration continues for many bits until a winning master gets the ownership of I2C bus. Its first stage is comparison of the address bits.

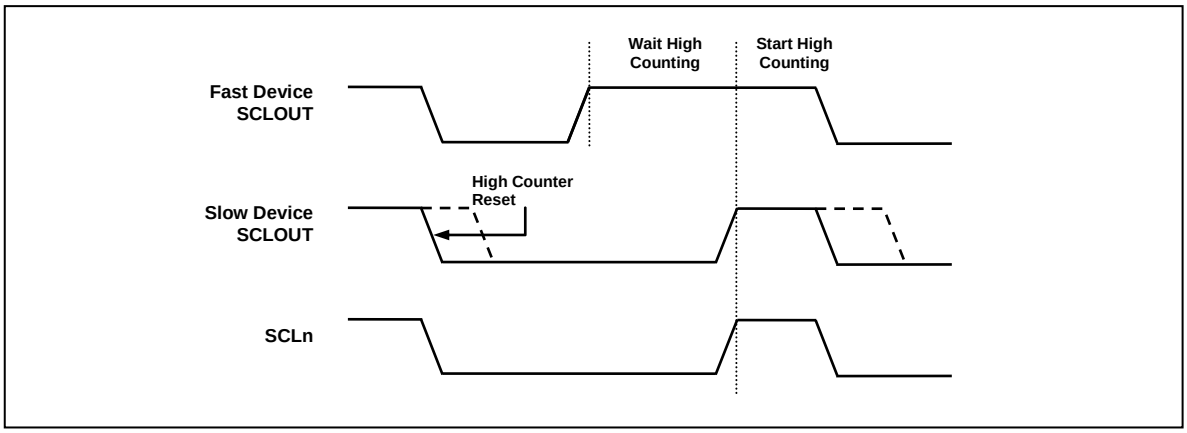


Figure 11.54 Clock Synchronization during Arbitration Procedure (USIn, where n = 0 and 1)

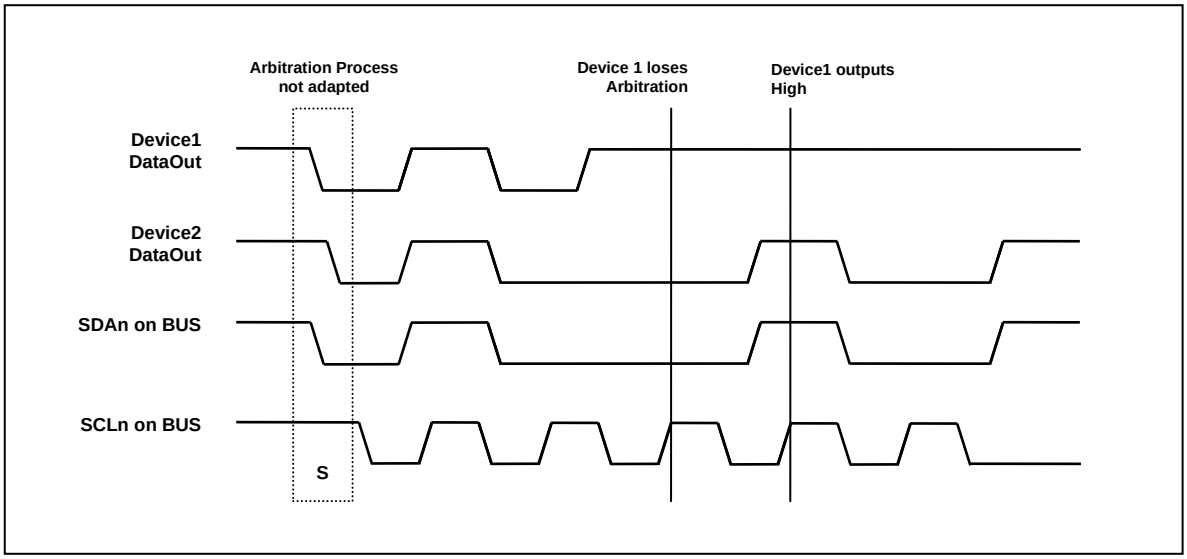


Figure 11.55 Arbitration Procedure of Two Masters (USIn, where n = 0 and 1)

11.11.20 USI0/1 I2C Operation

The I2C is byte-oriented and interrupt based. Interrupts are issued after all bus events except for a transmission of a START condition. Because the I2C is interrupt based, the application software is free to carry on other operations during a I2C byte transfer.

Note that when a I2C interrupt is generated, IICnIFR flag in USInCR4 register is set, it is cleared by writing an any value to USInST2. When I2C interrupt occurs, the SCLn line is hold LOW until writing any value to USInST2. When the IICnIFR flag is set, the USInST2 contains a value indicating the current state of the I2C bus. According to the value in USInST2, software can decide what to do next.

I2C can operate in 4 modes by configuring master/slave, transmitter/receiver. The operating mode is configured by a winning master. A more detailed explanation follows below.

11.11.20.1 USI0/1 I2C Master Transmitter

To operate I2C in master transmitter, follow the recommended steps below.

1. Enable I2C by setting USInMS[1:0] bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.
2. Load SLAn+W into the USInDR where SLAn is address of slave device and W is transfer direction from the viewpoint of the master. For master transmitter, W is '0'. Note that USInDR is used for both address and data.
3. Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.
4. Configure the USInSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of USInSCLR to the USInSDHR.
5. Set the STARTCn bit in USInCR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in USInDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALLn interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in USInST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOSTn bit in USInST2 is set, the ACKnEN bit in USInCR4 must be set and the received 7-bit address must equal to the USInSLA[6:0] bits in USInSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or STOP communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (Master) can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to USInDR.
   2) Master STOP data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLAn+R/W into the USInDR and set STARTCn bit in USInCR4.

   After doing one of the actions above, write any arbitrary to USInST2 to release SCLn line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in USInDR and if transfer direction bit is '1' go to master receiver section.
7. 1-Byte of data is being transmitted. During data transfer, bus arbitration continues.
8. This is ACK signal processing stage for data packet transmitted by master. I2C holds the SCLn LOW. When I2C loses bus mastership while transmitting data arbitrating other masters, the MLOSTn bit in USInST2 is set. If then, I2C waits in idle state. When the data in USInDR is transmitted completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases regardless of the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can receive more data from master. In this case, load data to transmit to USInDR.
   2) Master STOP data transfer even if it receives ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   3) Master transmits repeated START condition with not checking ACK signal. In this case, load SLAn+R/W into the USInDR and set the STARTCn bit in USInCR4.

   After doing one of the actions above, write any arbitrary to USInST2 to release SCLn line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in USInDR, and if transfer direction bit is '1' go to master receiver section.
9. This is the final step for master transmitter function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The next figure depicts above process for master transmitter operation of I2C.

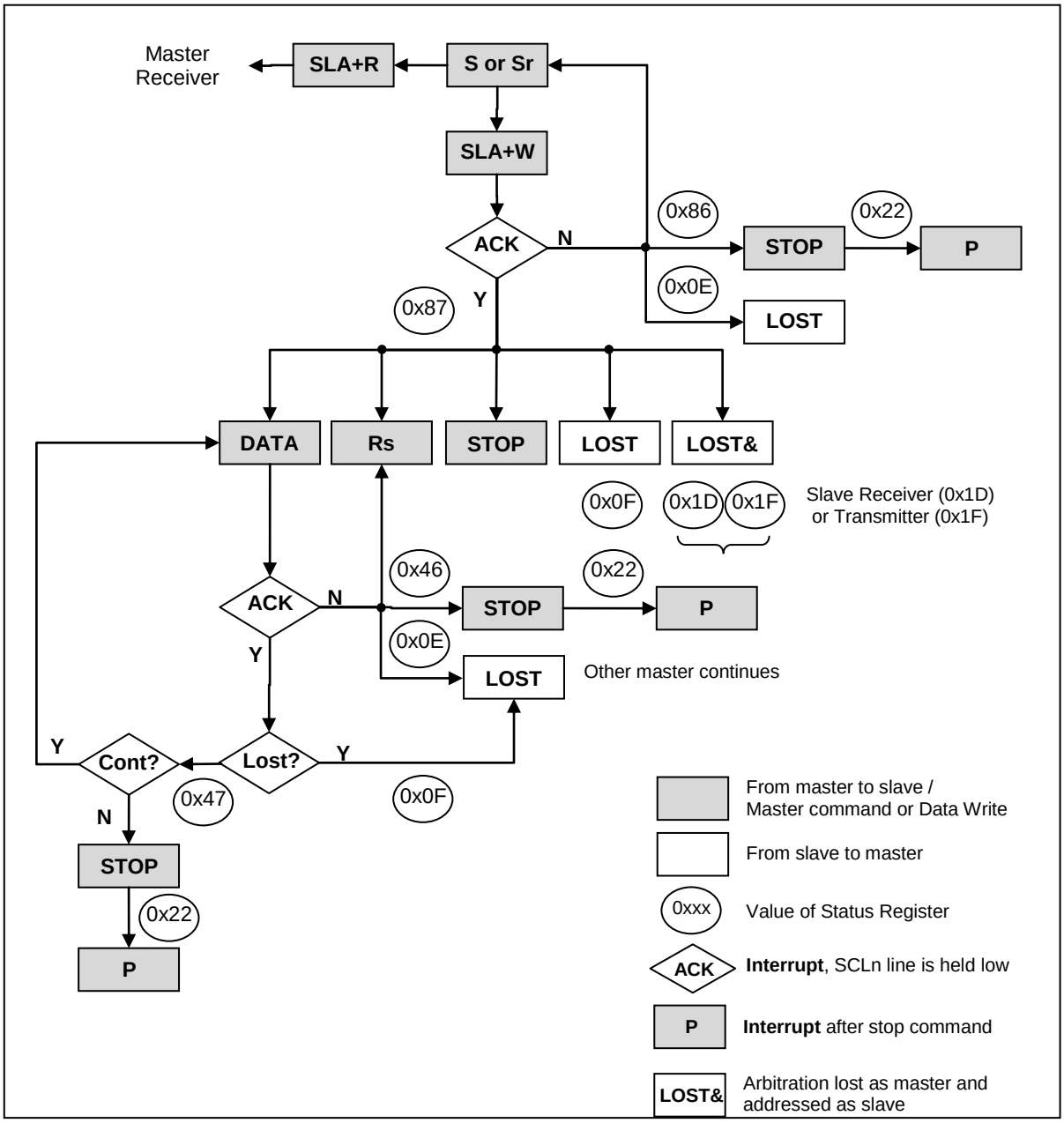


Figure 11.56 Formats and States in the Master Transmitter Mode (USIn, where n = 0 and 1)

11.11.20.2 USI0/1 I2C Master Receiver

To operate I2C in master receiver, follow the recommended steps below.

1. Enable I2C by setting USInMS[1:0] bits in USInCR1 and USInEN bit in USInCR2. This provides main clock to the peripheral.
2. Load SLAn+R into the USInDR where SLA is address of slave device and R is transfer direction from the viewpoint of the master. For master receiver, R is '1'. Note that USInDR is used for both address and data.
3. Configure baud rate by writing desired value to both USInSCLR and USInSCHR for the Low and High period of SCLn line.
4. Configure the USInSDHR to decide when SDAn changes value from falling edge of SCLn. If SDAn should change in the middle of SCLn LOW period, load half the value of USInSCLR to the USInSDHR.
5. Set the STARTCn bit in USInCR4. This transmits a START condition. And also configure how to handle interrupt and ACK signal. When the STARTCn bit is set, 8-bit data in USInDR is transmitted out according to the baud-rate.
6. This is ACK signal processing stage for address packet transmitted by master. When 7-bit address and 1-bit transfer direction is transmitted to target slave device, the master can know whether the slave acknowledged or not in the 9th high period of SCLn. If the master gains bus mastership, I2C generates GCALL interrupt regardless of the reception of ACK from the slave device. When I2C loses bus mastership during arbitration process, the MLOSTn bit in USInST2 is set, and I2C waits in idle state or can be operate as an addressed slave. To operate as a slave when the MLOSTn bit in USInST2 is set, the ACKnEN bit in USInCR4 must be set and the received 7-bit address must equal to the USInSLA[6:0] bits in USInSAR. In this case I2C operates as a slave transmitter or a slave receiver (go to appropriate section). In this stage, I2C holds the SCLn LOW. This is because to decide whether I2C continues serial transfer or STOP communication. The following steps continue assuming that I2C does not lose mastership during first data transfer.

   I2C (Master) can choose one of the following cases according to the reception of ACK signal from slave.

   1) Master receives ACK signal from slave, so continues data transfer because slave can prepare and transmit more data to master. Configure ACKnEN bit in USInCR4 to decide whether I2C ACKnowledges the next data to be received or not.
   2) Master STOP data transfer because it receives no ACK signal from slave. In this case, set the STOPCn bit in USInCR4.
   3) Master transmits repeated START condition due to no ACK signal from slave. In this case, load SLAn+R/W into the USInDR and set STARTCn bit in USInCR4.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1), move to step 7. In case of 2), move to step 9 to handle STOP interrupt. In case of 3), move to step 6 after transmitting the data in USInDR and if transfer direction bit is '0' go to master transmitter section.

7. 1-Byte of data is being received.
8. This is ACK signal processing stage for data packet transmitted by slave. I2C holds the SCLn LOW. When 1-Byte of data is received completely, I2C generates TENDn interrupt.

   I2C can choose one of the following cases according to the RXACKn flag in USInST2.

   1) Master continues receiving data from slave. To do this, set ACKnEN bit in USInCR4 to ACKnowledge the next data to be received.
   2) Master wants to terminate data transfer when it receives next data by not generating ACK signal. This can be done by clearing ACKnEN bit in USInCR4.
   3) Because no ACK signal is detected, master terminates data transfer. In this case, set the STOPCn bit in USInCR4.
   4) No ACK signal is detected, and master transmits repeated START condition. In this case, load SLAn+R/W into the USInDR and set the STARTCn bit in USInCR4.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1) and 2), move to step 7. In case of 3), move to step 9 to handle STOP interrupt. In case of 4), move to step 6 after transmitting the data in USInDR, and if transfer direction bit is '0' go to master transmitter section.

9. This is the final step for master receiver function of I2C, handling STOP interrupt. The STOP bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The processes described above for master receiver operation of I2C can be depicted as the following figure.

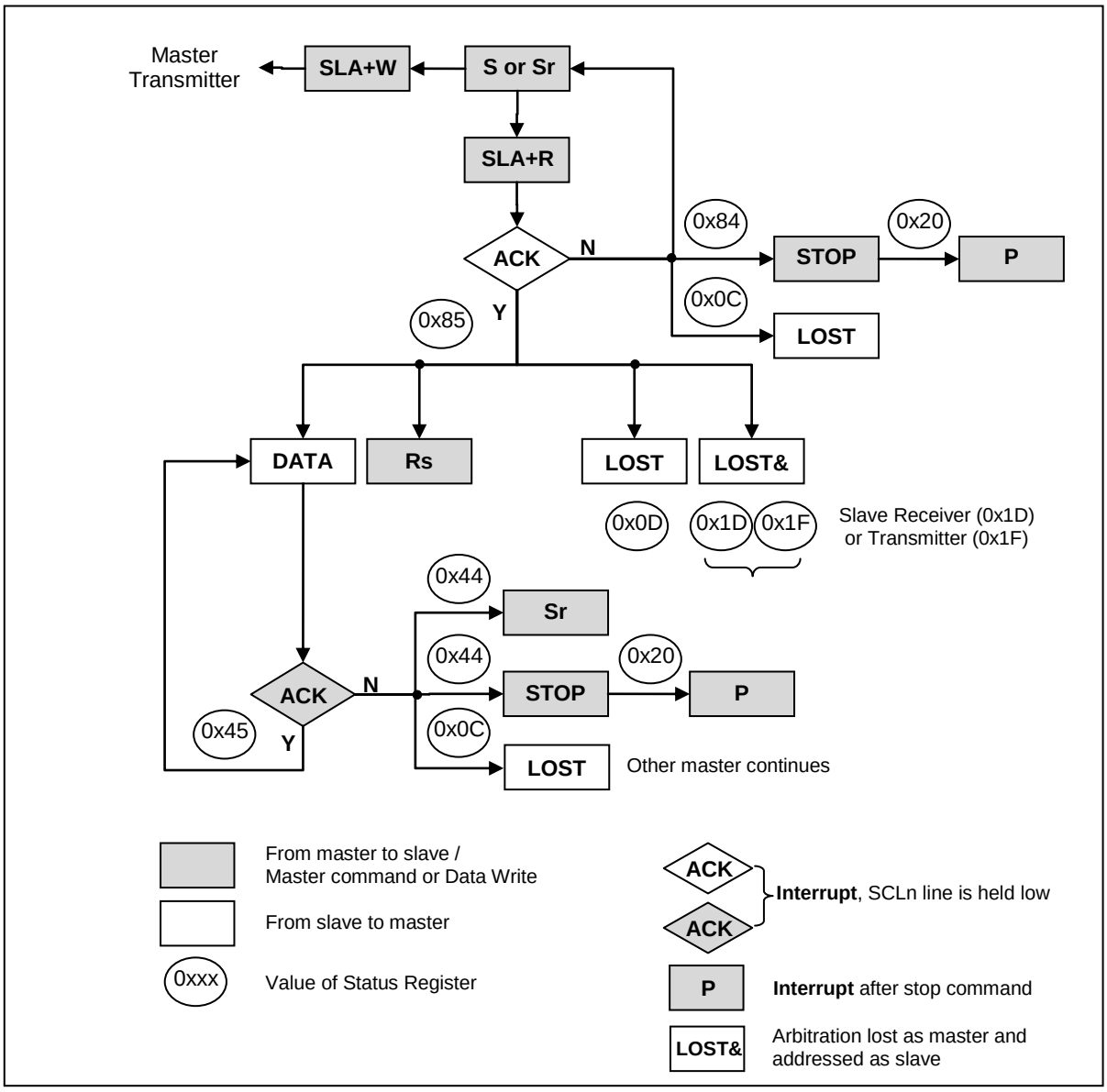


Figure 11.57 Formats and States in the Master Receiver Mode (USIn, where n = 0 and 1)

11.11.20.3 USI0/1 I2C Slave Transmitter

To operate I2C in slave transmitter, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into USInSDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USInSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USInMS[1:0] bits in USInCR1 , IICnIE bit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USInSLA[6:0] bits in USInSAR. If the GCALLn bit in USInSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to USInSLA[6:0] bits in USInSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to USInSLA[6:0] bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to USInSLA[6:0] bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs, load transmit data to USInDR and write arbitrary value to USInST2 to release SCLn line.
5. 1-Byte of data is being transmitted.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected and I2C waits STOP or repeated START condition.
   2) ACK signal from master is detected. Load data to transmit into USInDR.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave transmitter function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The next figure shows flow chart for handling slave transmitter function of I2C.

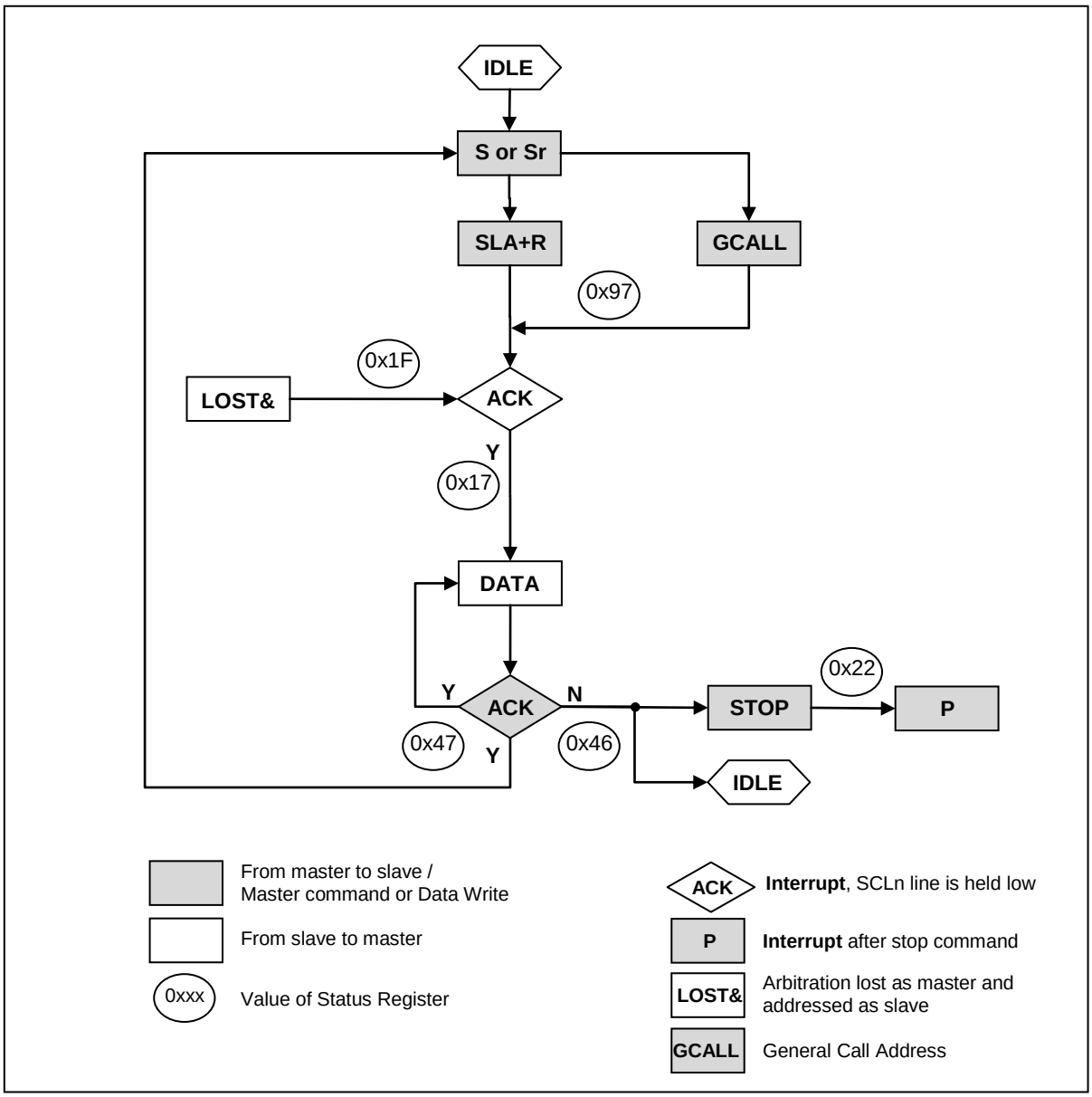


Figure 11.58 Formats and States in the Slave Transmitter Mode (USIn, where n = 0 and 1)

11.11.20.4 USI0/1 I2C Slave Receiver

To operate I2C in slave receiver, follow the recommended steps below.

1. If the main operating clock (SCLK) of the system is slower than that of SCLn, load value 0x00 into USInSDHR to make SDAn change within one system clock period from the falling edge of SCLn. Note that the hold time of SDAn is calculated by SDAH x period of SCLK where SDAH is multiple of number of SCLK coming from USInSDHR. When the hold time of SDAn is longer than the period of SCLK, I2C (slave) cannot transmit serial data properly.
2. Enable I2C by setting USInMS[1:0] bits in USInCR1, IICnIE bit in USInCR4 and USInEN bit in USInCR2. This provides main clock to the peripheral.
3. When a START condition is detected, I2C receives one byte of data and compares it with USInSLA[6:0] bits in USInSAR. If the GCALLn bit in USInSAR is enabled, I2C compares the received data with value 0x00, the general call address.
4. If the received address does not equal to SLAn bits in USInSAR, I2C enters idle state ie, waits for another START condition. Else if the address equals to SLAn bits and the ACKnEN bit is enabled, I2C generates SSELn interrupt and the SCLn line is held LOW. Note that even if the address equals to SLAn bits, when the ACKnEN bit is disabled, I2C enters idle state. When SSELn interrupt occurs and I2C is ready to receive data, write arbitrary value to USInST2 to release SCLn line.
5. 1-Byte of data is being received.
6. In this step, I2C generates TENDn interrupt and holds the SCLn line LOW regardless of the reception of ACK signal from master. Slave can select one of the following cases.

   1) No ACK signal is detected (ACKnEN=0) and I2C waits STOP or repeated START condition.
   2) ACK signal is detected (ACKnEN=1) and I2C can continue to receive data from master.

   After doing one of the actions above, write arbitrary value to USInST2 to release SCLn line. In case of 1) move to step 7 to terminate communication. In case of 2) move to step 5. In either case, a repeated START condition can be detected. For that case, move step 4.
7. This is the final step for slave receiver function of I2C, handling STOP interrupt. The STOPCn bit indicates that data transfer between master and slave is over. To clear USInST2, write any value to USInST2. After this, I2C enters idle state.

The process can be depicted as following figure when I2C operates in slave receiver mode.

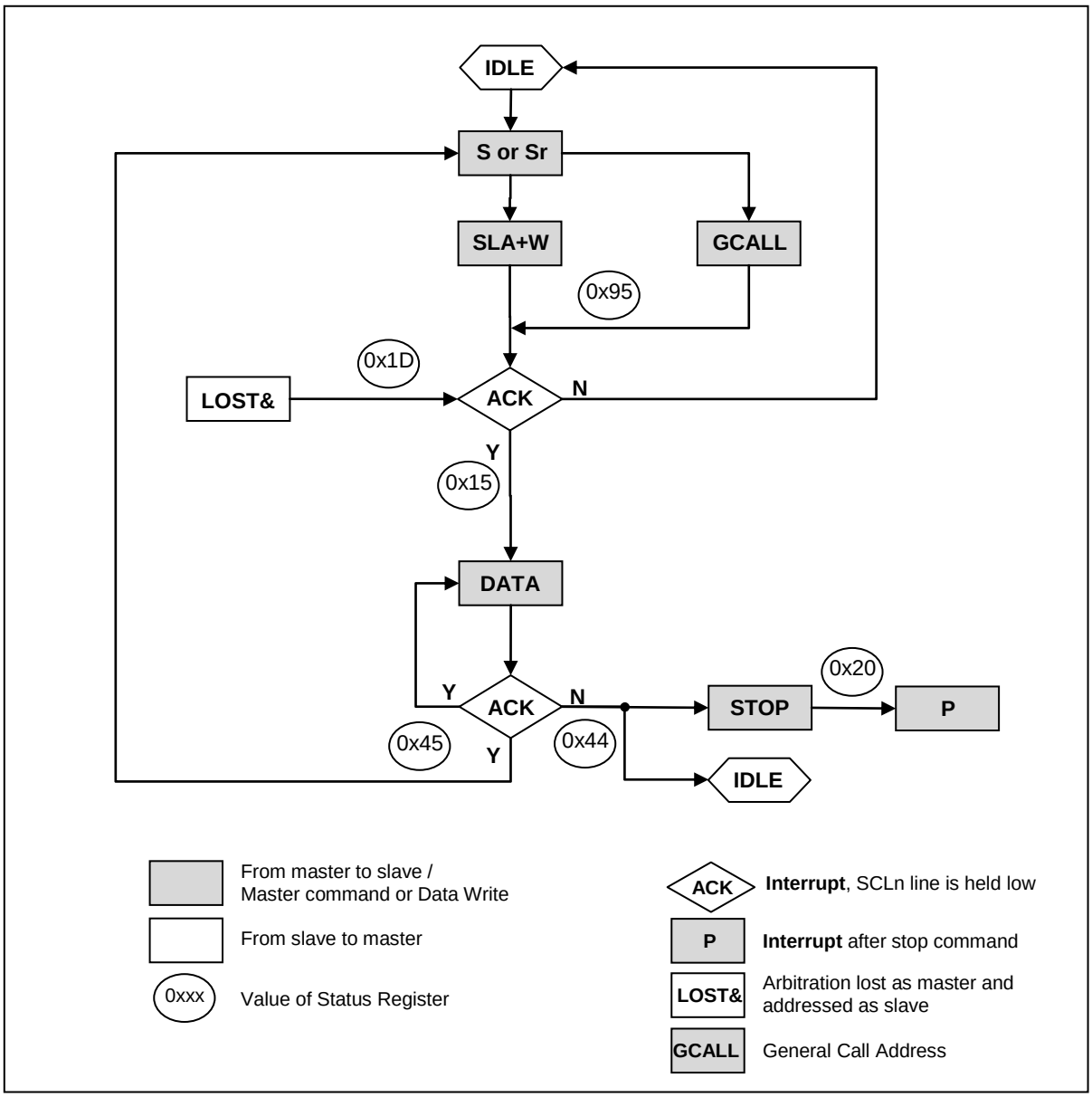
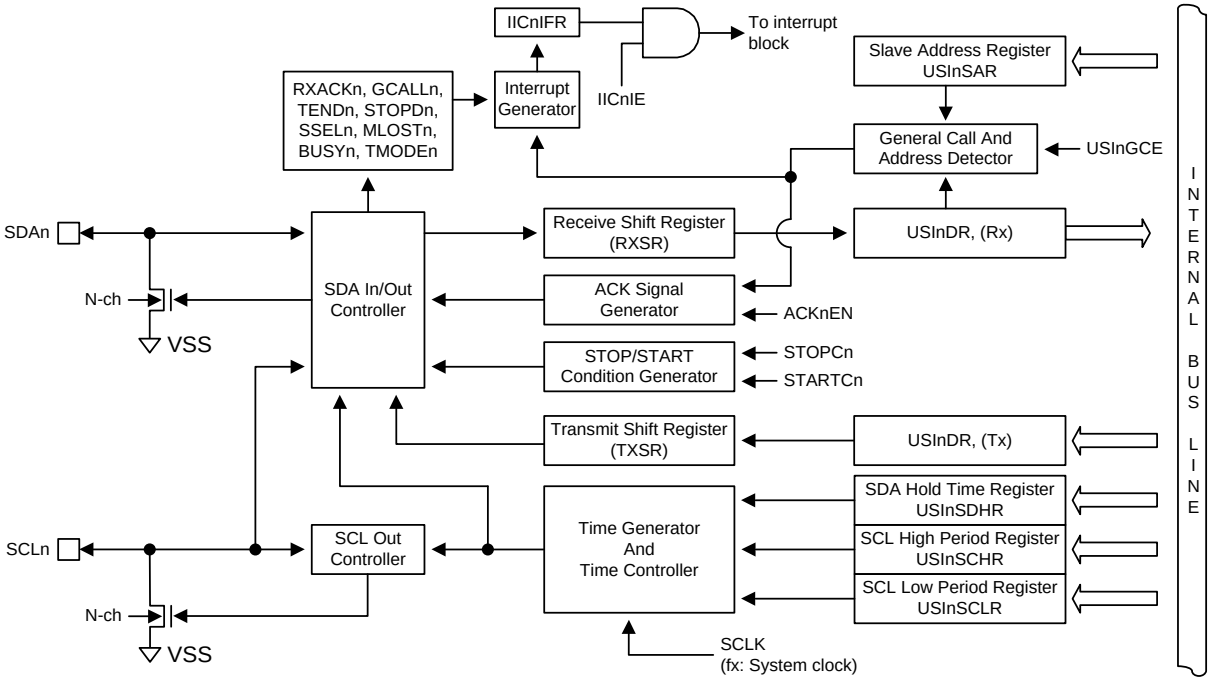


Figure 11.59 Formats and States in the Slave Receiver Mode (USIn, where n = 0 and 1)

11.11.21 USI0/1 I2C Block Diagram



NOTE) When the USI0/1 block is an I2C mode and the corresponding port is an sub-function for SCLn/SDAn pin, The SCLn/SDAn pins are automatically set to the N-channel open-drain outputs and the input latch is read in the case of reading the pins. The corresponding pull-up resistor is determined by the control register.

Figure 11.60 USI0/1 I2C Block Diagram (where n = 0 and 1)

11.11.22 Register Map

Table 11-19 USI0/1 Register Map (where n = 0 and 1)

Name	Address	Dir	Default	Description
USInBD	1022H/1032H (XSFR)	R/W	FFH	USIn Baud Rate Generation Register
USInDR	1024H/1034H (XSFR)	R/W	00H	USIn Data Register
USInSDHR	1023H/1033H (XSFR)	R/W	01H	USIn SDA Hold Time Register
USInSCHR	1026H/1036H (XSFR)	R/W	3FH	USIn SCL High Period Register
USInSCLR	1025H/1035H (XSFR)	R/W	3FH	USIn SCL Low Period Register
USInSAR	1027H/1037H (XSFR)	R/W	00H	USIn Slave Address Register
USInCR1	1018H/1028H (XSFR)	R/W	00H	USIn Control Register 1
USInCR2	1019H/1029H (XSFR)	R/W	00H	USIn Control Register 2
USInCR3	101AH/102AH (XSFR)	R/W	00H	USIn Control Register 3
USInCR4	101BH/102BH (XSFR)	R/W	00H	USIn Control Register 4
USInST1	1020H/1030H (XSFR)	R/W	80H	USIn Status Register 1
USInST2	1021H/1031H (XSFR)	R/W	00H	USIn Status Register 2

11.11.23 USI0/1 Register Description

USI0/1 module consists of USI0/1 baud rate generation register (USInBD), USI0/1 data register (USInDR), USI0/1 SDA hold time register (USInSDHR), USI0/1 SCL high period register (USInSCHR), USI0/1 SCL low period Register (USInSCLR), USI0/1 slave address register (USInSAR), USI0/1 control register 1/2/3/4 (USInCR1/2/3/4), USI0/1 status register 1/2 (USInST1/2).

11.11.24 Register Description for USI0/1

USInBD (USI0/1 Baud- Rate Generation Register: For UART and SPI mode) : 1022H/1032H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInBD7	USInBD6	USInBD5	USInBD4	USInBD3	USInBD2	USInBD1	USInBD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : FFH

USInBD[7:0] The value in this register is used to generate internal baud rate in asynchronous mode or to generate SCKn clock in SPI mode. To prevent malfunction, do not write ‘0’ in asynchronous mode and do not write ‘0’ or ‘1’ in SPI mode.

NOTE) In common with USInSAR register, USInBD register is used for slave address register when the USI0/1 I2C mode.

USInDR (USI0/1 Data Register: For UART, SPI, and I2C mode) : 1024H/1034H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInDR7	USInDR6	USInDR5	USInDR4	USInDR3	USInDR2	USInDR1	USInDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USInDR[7:0] The USIn transmit buffer and receive buffer share the same I/O address with this DATA register. The transmit data buffer is the destination for data written to the USInDR register. Reading the USInDR register returns the contents of the receive buffer.

Write to this register only when the DREn flag is set. In SPI master mode, the SCK clock is generated when data are written to this register.

USInSDHR (USI0/1 SDA Hold Time Register: For I2C mode) : 1023H/1033H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInSDHR7	USInSDHR6	USInSDHR5	USInSDHR4	USInSDHR3	USInSDHR2	USInSDHR1	USInSDHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 01H

USInSDHR[7:0] The register is used to control SDAn output timing from the falling edge of SCLn in I2C mode.

NOTE) that SDA is changed after t_{SCLK} X USInSDHR. In master mode, load half the value of USInSCLR to this register to make SDA change in the middle of SCL.

In slave mode, configure this register regarding the frequency of SCL from master.

The SDA is changed after t_{SCLK} X (USInSDHR+2) in master mode. So, to insure operation in slave mode, the value t_{SCLK} X (USInSDHR + 2) must be smaller than the period of SCL.

USInSCHR (USI0/1 SCL High Period Register: For I2C mode) : 1026H/1036H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInSCHR7	USInSCHR6	USInSCHR5	USInSCHR4	USInSCHR3	USInSCHR2	USInSCHR1	USInSCHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

USInSCHR[7:0] This register defines the high period of SCLn when it operates in I2C master mode.

The base clock is SCLK, the system clock, and the period is calculated by the formula: t_{SCLK} X (4 X USInSCHR + 2) where t_{SCLK} is the period of SCLK.

So, the operating frequency of I2C master mode is calculated by the following equation.

$$f_{I2C} = \frac{1}{t_{SCLK} \text{ X } (4 \text{ X } (\text{USInSCLR} + \text{USInSCHR}) + 4)}$$

USInSCLR (USI0/1 SCL Low Period Register: For I2C mode) : 1025H/1035H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInSCLR7	USInSCLR6	USInSCLR5	USInSCLR4	USInSCLR3	USInSCLR2	USInSCLR1	USInSCLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 3FH

USInSCLR[7:0] This register defines the high period of SCL when it operates in I2C master mode.

The base clock is SCLK, the system clock, and the period is calculated by the formula: t_{SCLK} X (4 X USInSCLR +2) where t_{SCLK} is the period of SCLK.

USInSAR (USI0/1 Slave Address Register: For I2C mode) : 1027H/1037H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInSLA6	USInSLA5	USInSLA4	USInSLA3	USInSLA2	USInSLA1	USInSLA0	USInGCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USInSLA[6:0] These bits configure the slave address of I2C when it operaties in I2C slave mode.

USInGCE This bit decides whether I2C allows general call address or not in I2C slave mode.

0 Ignore general call address

1 Allow general call address

USInCR1 (USI0/1 Control Register 1: For UART, SPI, and I2C mode) : 1018H/1028H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
USInMS1	USInMS0	USInPM1	USInPM0	USInS2	USInS1 ORDn	USInS0 CPHAn	CPOLn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

USInMS[1:0] Selects operation mode of USIn

USInMS1	USInMS0	Operation mode
0	0	Asynchronous Mode (UART)
0	1	Synchronous Mode (UART)
1	0	I2C mode
1	1	SPI mode

USInPM[1:0] Selects parity generation and check methods (only UART mode)

USInPM1	USInPM0	Parity
0	0	No Parity
0	1	Reserved
1	0	Even Parity
1	1	Odd Parity

USInS[2:0] When in asynchronous or synchronous mode of operation, selects the length of data bits in frame

USInS2	USInS1	USInS0	Data Length
0	0	0	5 bit
0	0	1	6 bit
0	1	0	7 bit
0	1	1	8 bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9 bit

ORDn This bit in the same bit position with USInS1. The MSB of the data byte is transmitted first when set to '1' and the LSB when set to '0' (onl SPI mode)

0	LSB-first
1	MSB-first

CPHAn This bit is in the same bit position with USInS0. This bit determines if data are sampled on the leading or trailing edge of SCK (only SPI mode).

CPOLn	CPHAn	Leading edge	Trailing edge
0	0	Sample (Rising)	Setup (Falling)
0	1	Setup (Rising)	Sample (Falling)
1	0	Sample (Falling)	Setup (Rising)
1	1	Setup (Falling)	Sample (Rising)

CPOLn This bit determines the clock polarity of ACK in synchronous or SPI mode.

0	TXD change@Rising Edge, RXD change@Falling Edge
1	TXD change@Falling Edge, RXD change@Rising Edge

USInCR2 (USI0/1 Control Register 2: For UART, SPI, and I2C mode) : 1019H/1029H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
DRIEn	TXCIEn	RXCIEn	WAKEIEn	TXEn	RXEn	USInEN	DBLSn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

DRIEn Interrupt enable bit for data register empty (only UART and SPI mode).

0 Interrupt from DREn is inhibited (use polling)

1 When DREn is set, request an interrupt

TXCIEn Interrupt enable bit for transmit complete (only UART and SPI mode).

0 Interrupt from TXCn is inhibited (use polling)

1 When TXCn is set, request an interrupt

RXCIEn Interrupt enable bit for receive complete (only UART and SPI mode).

0 Interrupt from RXCn is inhibited (use polling)

1 When RXCn is set, request an interrupt

WAKEIEn Interrupt enable bit for asynchronous wake in STOP mode. When device is in stop mode, if RXDn goes to low level an interrupt can be requested to wake-up system. (only UART mode). At that time the DRIEn bit and USInST1 register value should be set to '0b' and "00H", respectively.

0 Interrupt from Wake is inhibited

1 When WAKEn is set, request an interrupt

TXEn Enables the transmitter unit (only UART and SPI mode).

0 Transmitter is disabled

1 Transmitter is enabled

RXEn Enables the receiver unit (only UART and SPI mode).

0 Receiver is disabled

1 Receiver is enabled

USInEN Activate USIn function block by supplying.

0 USIn is disabled

1 USIn is enabled

DBLSn This bit selects receiver sampling rate (only UART).

0 Normal asynchronous operation

1 Double Speed asynchronous operation

USInCR3 (USI0/1 Control Register 3: For UART, SPI, and I2C mode) : 101AH/102AH (ESFR), n = 0, 1

7	6	5	4	3	2	1	0
MASTERn	LOOPSn	DISSCKn	USInSSEN	FXCHn	USInSB	USInTX8	USInRX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

MASTERn Selects master or slave in SPI and synchronous mode operation and controls the direction of SCKn pin

0 Slave mode operation (External clock for SCK).

1 Master mode operation(Internal clock for SCK).

LOOPSn Controls the loop back mode of USIn for test mode (only UART and SPI mode)

0 Normal operation

1 Loop Back mode

DISSCKn In synchronous mode of operation, selects the waveform of SCKn output

0 ACK is free-running while UART is enabled in synchronous master mode

1 ACK is active while any frame is on transferring

USInSSEN This bit controls the SSn pin operation (only SPI mode)

0 Disable

1 Enable (The SSn pin should be a normal input)

FXCHn SPI port function exchange control bit (only SPI mode)

0 No effect

1 Exchange MOSIn and MISOn function

USInSB Selects the length of stop bit in asynchronous or synchronous mode of operation.

0 1 Stop Bit

1 2 Stop Bit

USInTX8 The ninth bit of data frame in asynchronous or synchronous mode of operation. Write this bit first before loading the USInDR register

0 MSB (9^{th} bit) to be transmitted is ‘0’

1 MSB (9^{th} bit) to be transmitted is ‘1’

USInRX8 The ninth bit of data frame in asynchronous or synchronous mode of operation. Read this bit first before reading the receive buffer (only UART mode).

0 MSB (9^{th} bit) received is ‘0’

1 MSB (9^{th} bit) received is ‘1’

USInCR4 (USI0/1 Control Register 4: For I2C mode) : 101BH/102BH (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
IICnIFR	–	TXDLYENBn	IICnIE	ACKnEN	IMASTERn	STOPCn	STARTCn
R	–	R/W	R/W	R/W	R	R/W	R/W

Initial value : 00H

IICnIFR This is an interrupt flag bit for I2C mode. When an interrupt occurs, this bit becomes '1'. This bit is cleared when write any values in th USInST2. Writing "1" has no effect.

0 I2C interrupt no generation

1 I2C interrupt generation

TXDLYENBn USInSDHR register control bit

0 Enable USInSDHR register

1 Disable USInSDHR register

IICnIE Interrupt Enable bit for I2C mode

0 Interrupt from I2C is inhibited (use polling)

1 Enable interrupt for I2C

ACKnEN Controls ACK signal Generation at ninth SCL period.

0 No ACK signal is generated (SDA =1)

1 ACK signal is generated (SDA =0)

NOTES) ACK signal is output (SDA =0) for the following 3 cases.

1. When received address packet equals to USInSLA bits in USInSAR.

2. When received address packet equals to value 0x00 with GCALLn enabled.

3. When I2C operates as a receiver (master or slave)

IMASTERn Represent operating mode of I2C

0 I2C is in slave mode

1 I2C is in master mode

STOPCn When I2C is master, STOP condition generation

0 No effect

1 STOP condition is to be generated

STARTCn When I2C is master, START condition generation

0 No effect

1 START or repeated START condition is to be generated

USInST1 (USI0/1 Status Register 1: For UART and SPI mode) : 1020H/1030H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
DREn	TXCn	RXCn	WAKEn	USInRST	DORn	FEn	PEn
R/W	R/W	R	R/W	R/W	R	R/W	R/W

Initial value : 80H

DREn The DREn flag indicates if the transmit buffer (USInDR) is ready to receive new data. If DREn is '1', the buffer is empty and ready to be written. This flag can generate a DREn interrupt.

0 Transmit buffer is not empty.

1 Transmit buffer is empty.

TXCn This flag is set when the entire frame in the transmit shift register has been shifted out and there is no new data currently present in the transmit buffer. This flag is automatically cleared when the interrupt service routine of a TXCn interrupt is executed. This flag can generate a TXCn interrupt. This bit is automatically cleared.

0 Transmission is ongoing.

1 Transmit buffer is empty and the data in transmit shift register are shifted out completely.

RXCn This flag is set when there are unread data in the receive buffer and cleared when all the data in the receive buffer are read. The RXCn flag can be used to generate a RXCn interrupt.

0 There is no data unread in the receive buffer

1 There are more than 1 data in the receive buffer

WAKEn This flag is set when the RXDn pin is detected low while the CPU is in STOP mode. This flag can be used to generate a WAKEn interrupt. This bit is set only when in asynchronous mode of operation. This bit should be cleared by program software. (only UART mode)

0 No WAKE interrupt is generated.

1 WAKE interrupt is generated

USInRST This is an internal reset and only has effect on USIn. Writing '1' to this bit initializes the internal logic of USIn and this bit is automatically cleared to '0'.

0 No effect

1 Reset USIn

DORn This bit is set if a Data OverRun occurs. While this bit is set, the incoming data frame is ignored. This flag is valid until the receive buffer is read.

0 No Data OverRun

1 Data OverRun detected

FEn This bit is set if the first stop bit of next character in the receive buffer is detected as '0'. This bit is valid until the receive buffer is read. (only UART mode)

0 No Frame Error

1 Frame Error detected

PEn This bit is set if the next character in the receive buffer has a Parity Error to be received while Parity Checking is enabled. This bit is valid until the receive buffer is read. (only UART mode)

0 No Parity Error

1 Parity Error detected

USInST2 (USI0/1 Status Register 2: For I2C mode) : 1021H/1031H (XSFR), n = 0, 1

7	6	5	4	3	2	1	0
GCALLn	TENDn	STOPDn	SSELn	MLOSTn	BUSYn	TMODEn	RXACKn
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

GCALLn(NOTE) This bit has different meaning depending on whether I2C is master or slave. When I2C is a master, this bit represents whether it received AACK (address ACK) from slave.

0 No AACK is received (Master mode)

1 AACK is received (Master mode)

When I2C is a slave, this bit is used to indicated general call.

0 General call address is not detected (Slave mode)

1 General call address is detected (Slave mode)

TENDn(NOTE) This bit is set when 1-byte of data is transferred completely

0 1 byte of data is not completely transferred

1 1 byte of data is completely transferred

STOPDn(NOTE) This bit is set when a STOP condition is detected.

0 No STOP condition is detected

1 STOP condition is detected

SSELn(NOTE) This bit is set when I2C is addressed by other master.

0 I2C is not selected as a slave

1 I2C is addressed by other master and acts as a slave

MLOSTn(NOTE) This bit represents the result of bus arbitration in master mode.

0 I2C maintains bus mastership

1 I2C maintains bus mastership during arbitration process

BUSYn This bit reflects bus status.

0 I2C bus is idle, so a master can issue a START condition

1 I2C bus is busy

TMODEn This bit is used to indicate whether I2C is transmitter or receiver.

0 I2C is a receiver

1 I2C is a transmitter

RXACKn This bit shows the state of ACK signal

0 No ACK is received

1 ACK is received at ninth SCL period

NOTE) These bits can be source of interrupt.

When an I2C interrupt occurs except for STOP mode, the SCL line is hold LOW. To release SCL, write arbitrary value to USInST2. When USInST2 is written, the TENDn, STOPDn, SSELn, MLOSTn, and RXACKn bits are cleared.

11.12 Baud Rate setting (example)

Table 11-20 Examples of USI0BD and USI1BD Settings for Commonly Used Oscillator Frequencies

Baud Rate	fx=1.00MHz		fx=1.8432MHz		fx=2.00MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR
2400	25	0.2%	47	0.0%	51	0.2%
4800	12	0.2%	23	0.0%	25	0.2%
9600	6	-7.0%	11	0.0%	12	0.2%
14.4k	3	8.5%	7	0.0%	8	-3.5%
19.2k	2	8.5%	5	0.0%	6	-7.0%
28.8k	1	8.5%	3	0.0%	3	8.5%
38.4k	1	-18.6%	2	0.0%	2	8.5%
57.6k	-	-	1	-25.0%	1	8.5%
76.8k	-	-	1	0.0%	1	-18.6%
115.2k	-	-	-	-	-	-
230.4k	-	-	-	-	-	-

(continued)

Baud Rate	fx=3.6864MHz		fx=4.00MHz		fx=7.3728MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR
2400	95	0.0%	103	0.2%	191	0.0%
4800	47	0.0%	51	0.2%	95	0.0%
9600	23	0.0%	25	0.2%	47	0.0%
14.4k	15	0.0%	16	2.1%	31	0.0%
19.2k	11	0.0%	12	0.2%	23	0.0%
28.8k	7	0.0%	8	-3.5%	15	0.0%
38.4k	5	0.0%	6	-7.0%	11	0.0%
57.6k	3	0.0%	3	8.5%	7	0.0%
76.8k	2	0.0%	2	8.5%	5	0.0%
115.2k	1	0.0%	1	8.5%	3	0.0%
230.4k	-	-	-	-	1	0.0%
250k	-	-	-	-	1	-7.8%
0.5M	-	-	-	-	-	-

(continued)

Baud Rate	fx=8.00MHz		fx=11.0592MHz	
	USI0BD/USI1BD	ERROR	USI0BD/USI1BD	ERROR
2400	207	0.2%	-	-
4800	103	0.2%	143	0.0%
9600	51	0.2%	71	0.0%
14.4k	34	-0.8%	47	0.0%
19.2k	25	0.2%	35	0.0%
28.8k	16	2.1%	23	0.0%
38.4k	12	0.2%	17	0.0%
57.6k	8	-3.5%	11	0.0%
76.8k	6	-7.0%	8	0.0%
115.2k	3	8.5%	5	0.0%
230.4k	1	8.5%	2	0.0%
250k	1	0.0%	2	-7.8%
0.5M	-	-	-	-
1M	-	-	-	-

11.13 12-Bit A/D Converter

11.13.1 Overview

The analog-to-digital converter (A/D) allows conversion of an analog input signal to corresponding 12-bit digital value. The A/D module has fifteen analog inputs. The output of the multiplexer is the input into the converter which generates the result through successive approximation. The A/D module has four registers which are the A/D converter control high register (ADCCRH), A/D converter control low register (ADCCRL), A/D converter data high register (ADCDRH), and A/D converter data low register (ADCDRL). The channels to be converted are selected by setting ADSEL[3:0]. To execute A/D conversion, TRIG[1:0] bits should be set to 'xx'. The register ADCDRH and ADCDRL contains the results of the A/D conversion. When the conversion is completed, the result is loaded into the ADCDRH and ADCDRL, the A/D conversion status bit AFLAG is set to '1', and the A/D interrupt is set. During A/D conversion, AFLAG bit is read as '0'.

11.13.2 Conversion Timing

The A/D conversion process requires 4 steps (4 clock edges) to convert each bit and 10 clocks to set up A/D conversion. Therefore, total of 58 clocks are required to complete a 12-bit conversion: When fxx/8 is selected for conversion clock with a 12MHz fxx clock frequency, one clock cycle is 0.66 us. Each bit conversion requires 4 clocks, the conversion rate is calculated as follows:

4 clocks/bit × 12 bits + set-up time = 58 clocks,

58 clock × 0.66 us = 38.28 us at 1.5 MHz (12 MHz/8)

NOTE) The A/D converter needs at least 20 us for conversion time. So you must set the conversion time more than 20 us.

11.13.3 Block Diagram

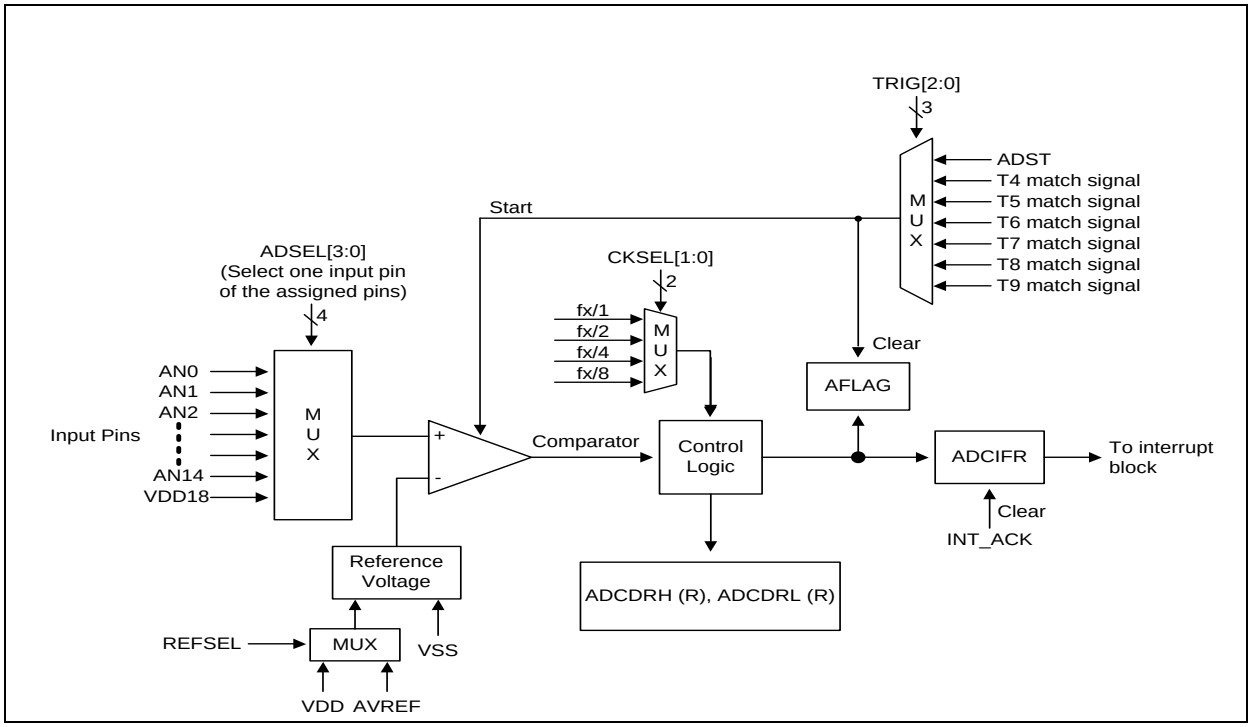


Figure 11.61 12-bit ADC Block Diagram

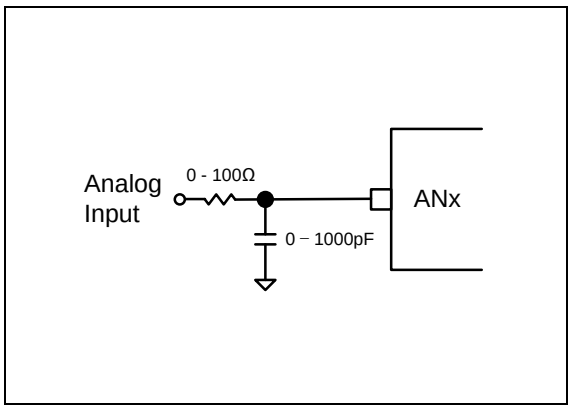


Figure 11.62 A/D Analog Input Pin with Capacitor

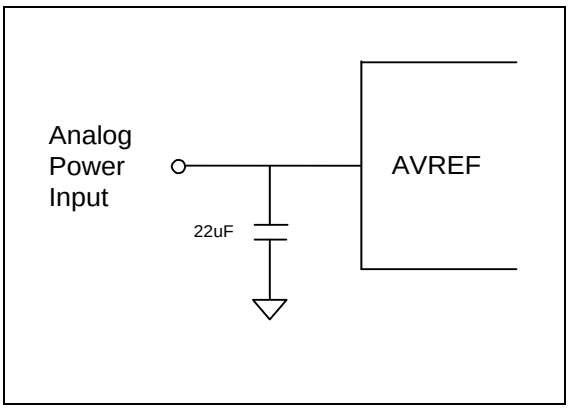


Figure 11.63 A/D Power (AVREF) Pin with Capacitor

11.13.4 ADC Operation

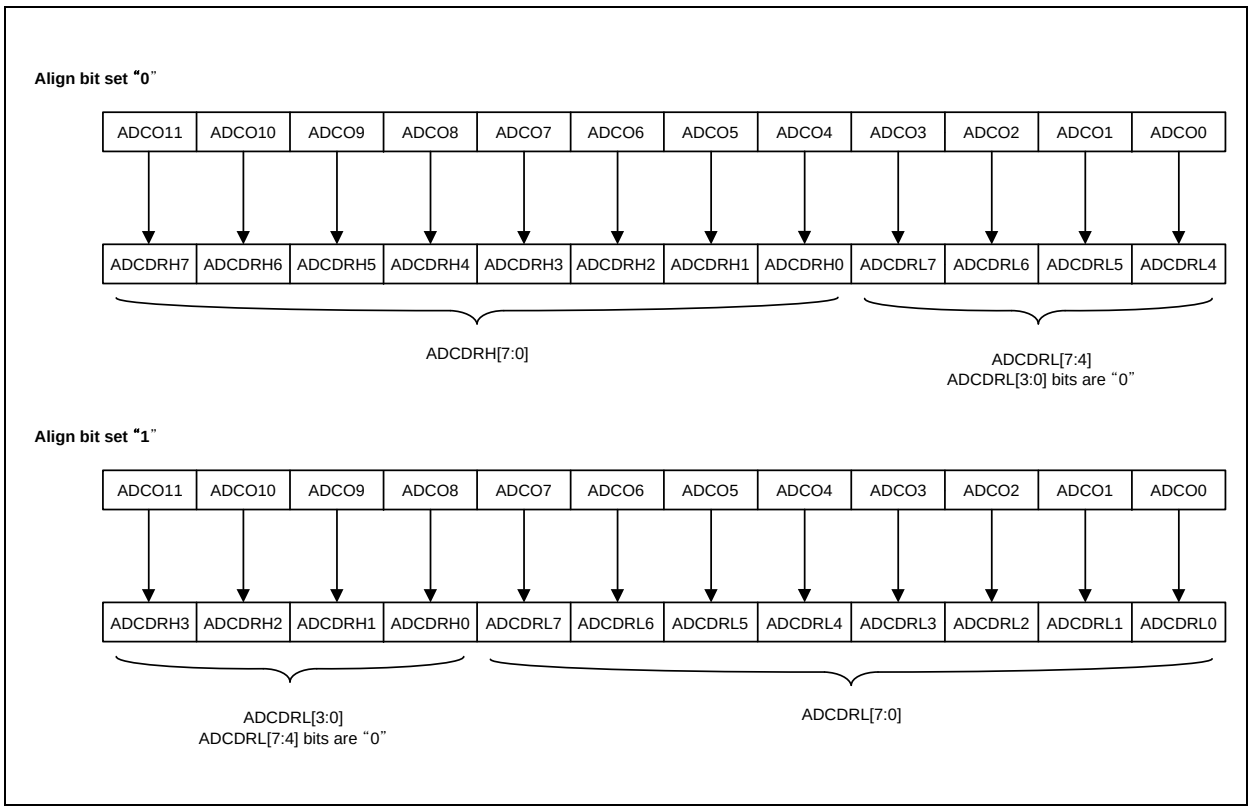


Figure 11.64 ADC Operation for Align Bit

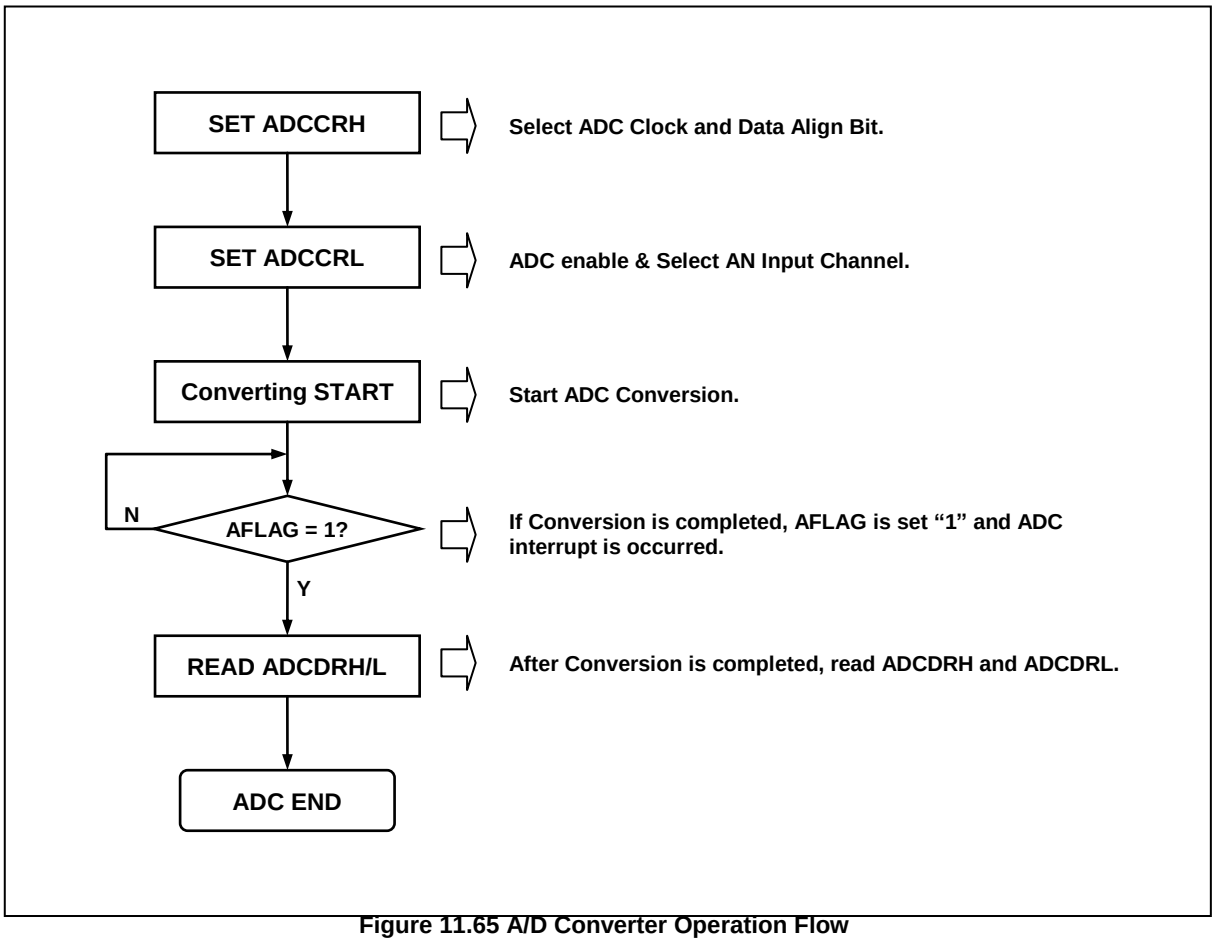


Figure 11.65 A/D Converter Operation Flow

11.13.5 Register Map

Table 11-21 ADC Register Map

Name	Address	Dir	Default	Description
ADCDRH	1053H (XSFR)	R	xxH	A/D Converter Data High Register
ADCDRL	1052H (XSFR)	R	xxH	A/D Converter Data Low Register
ADCCRH	1051H (XSFR)	R/W	00H	A/D Converter Control High Register
ADCCRL	1050H (XSFR)	R/W	00H	A/D Converter Control Low Register

11.13.6 ADC Register Description

The ADC register consists of A/D converter data high register (ADCDRH), A/D converter data low register (ADCDRL), A/D converter control high register (ADCCRH) and A/D converter control low register (ADCCRL).

11.13.7 Register Description for ADC

ADCDRH (A/D Converter Data High Register) : 1053H (XSFR)

7	6	5	4	3	2	1	0
ADDM11	ADDM10	ADDM9	ADDM8	ADDM7 ADDL11	ADDM6 ADDL10	ADDM5 ADDL9	ADDM4 ADDL8
R	R	R	R	R	R	R	R

Initial value : xxH

ADDM[11:4] MSB align, A/D Converter High Data (8-bit)

ADDL[11:8] LSB align, A/D Converter High Data (4-bit)

ADCDRL (A/D Converter Data Low Register) : 1052H (XSFR)

7	6	5	4	3	2	1	0
ADDM3 ADDL7	ADDM2 ADDL6	ADDM1 ADDL5	ADDM0 ADDL4	ADDL3	ADDL2	ADDL1	ADDL0
R	R	R	R	R-	R	R	R

Initial value : xxH

ADDM[3:0] MSB align, A/D Converter Low Data (4-bit)

ADDL[7:0] LSB align, A/D Converter Low Data (8-bit)

ADCCRH (A/D Converter High Register) : 1051H (XSFR)

7	6	5	4	3	2	1	0
ADCIFR	–	TRIG2	TRIG1	TRIG0	ALIGN	CKSEL1	CKSEL0
R/W	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

ADCIFR When ADC interrupt occurs, this bit becomes '1'. For clearing bit, write '0' to this bit or auto clear by INT_ACK signal. Writing "1" has no effect.

0 ADC Interrupt no generation

1 ADC Interrupt generation

TRIG[2:0] A/D Trigger Signal Selection(The ADC module is automatically disabled at stop mode)

TRIG2	TRIG1	TRIG0	Description
0	0	0	ADST
0	0	1	Timer 4 match signal
0	1	0	Timer 5 match signal
0	1	1	Timer 6 match signal
1	0	0	Timer 7 match signal
1	0	1	Timer 8 match signal
1	1	0	Timer 9 match signal
1	1	1	Not used

ALIGN A/D Converter data align selection.

0 MSB align (ADCDRH[7:0], ADCDRL[7:4])

1 LSB align (ADCRDH[3:0], ADCDRL[7:0])

CKSEL[1:0] A/D Converter Clock selection

CKSEL1	CKSEL0	Description
0	0	fx/1
0	1	fx/2
1	0	fx/4
1	1	fx/8

ADCCRL (A/D Converter Counter Low Register) : 1050H (XSFR)

7	6	5	4	3	2	1	0
STBY	ADST	REFSEL	AFLAG	ADSEL3	ADSEL2	ADSEL1	ADSEL0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Initial value : 00H

STBY Control Operation of A/D
(The ADC module is automatically disabled at stop mode)

0 ADC module disable

1 ADC module enable

ADST Control A/D Conversion start.

0 No effect

1 Trigger signal generation for conversion start

REFSEL A/D Converter Reference Selection

0 Internal Reference (VDD)

1 External Reference (AVREF)

AFLAG A/D Converter Operation State (This bit is cleared to '0' when the STBY bit is set to '0' or when the CPU is at STOP mode)

0 During A/D Conversion

1 A/D Conversion finished

ADSEL[3:0] A/D Converter input selection

ADSEL3	ADSEL2	ADSEL1	ADSEL0	Description
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13
1	1	1	0	AN14
1	1	1	1	VDD18

12. Power Down Operation

12.1 Overview

The MC97F2664 has two power-down modes to minimize the power consumption of the device. In power down mode, power consumption is reduced considerably. The device provides three kinds of power saving functions, Main-IDLE, Sub-IDLE and STOP mode. In three modes, program is stopped.

12.2 Peripheral Operation in IDLE/STOP Mode

Table 12-1 Peripheral Operation during Power Down Mode

Peripheral	IDLE Mode	STOP Mode
CPU	ALL CPU Operation are Disable	ALL CPU Operation are Disable
RAM	Retain	Retain
Basic Interval Timer	Operates Continuously	Stop
Watch Dog Timer	Operates Continuously	Stop (Can be operated with WDTRC OSC)
Watch Timer	Operates Continuously	Stop (Can be operated with sub clock)
Timer0~9	Operates Continuously	Halted (Only when the Event Counter Mode is Enabled, Timer operates Normally)
ADC	Operates Continuously	Stop
BUZ	Operates Continuously	Stop
SPI2/3	Operates Continuously	Only operate with external clock
UART2/3/4	Operates Continuously	Only operate with external clock
USI0/1	Operates Continuously	Only operate with external clock
Internal OSC (16MHz)	Oscillation	Stop when the system clock (fx) is f_{IRC}
WDTRC OSC (5kHz)	Can be operated with setting value	Can be operated with setting value
Main OSC (0.4~16MHz)	Oscillation	Stop when fx = f_{XIN}
Sub OSC (32.768kHz)	Oscillation	Stop when fx = f_{SUB}
I/O Port	Retain	Retain
Control Register	Retain	Retain
Address Data Bus	Retain	Retain
Release Method	By RESET, all Interrupts	By RESET, Timer Interrupt (EC0~ EC9), SPI (External clock), External Interrupt, UART by RX, WT (sub clock), WDT

12.3 IDLE Mode

The power control register is set to '01h' to enter the IDLE Mode. In this mode, the internal oscillation circuits remain active. Oscillation continues and peripherals are operated normally but CPU stops. It is released by reset or interrupt. To be released by interrupt, interrupt should be enabled before IDLE mode. If using reset, because the device becomes initialized state, the registers have reset value.

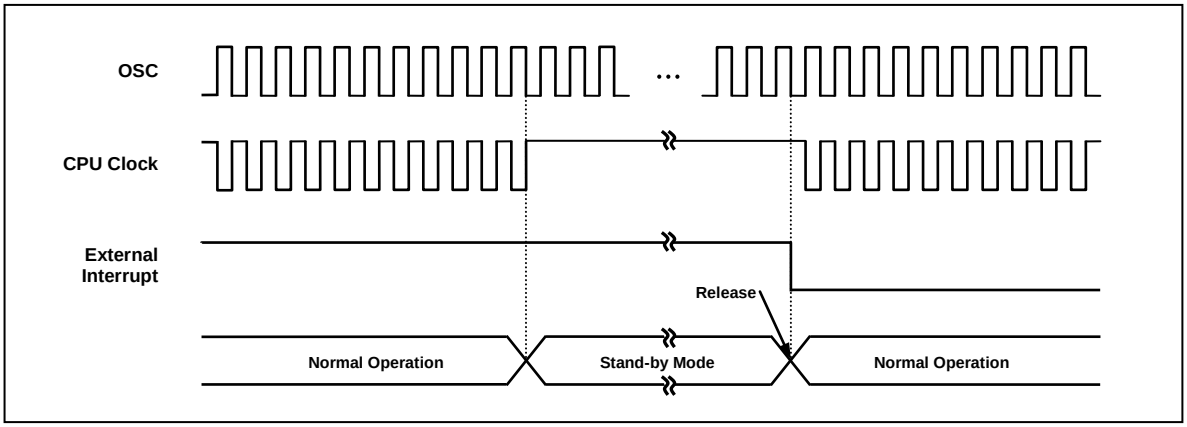


Figure 12.1 IDLE Mode Release Timing by External Interrupt

12.4 STOP Mode

The power control register is set to '03H' to enter the STOP Mode. In the stop mode, the selected oscillator, system clock and peripheral clock is stopped, but watch timer can be continued to operate with sub clock. With the clock frozen, all functions are stopped, but the on-chip RAM and control registers are held. For example, If the internal RC oscillator (f_{IRC}) is selected for the system clock and the sub clock (f_{SUB}) is oscillated, the internal RC oscillator stops oscillation and the sub clock is continuously oscillated in stop mode. At that time, the watch timer can be operated with the sub clock.

The source for exit from STOP mode is hardware reset and interrupts. The reset re-defines all the control registers.

When exit from STOP mode, enough oscillation stabilization time is required to normal operation. Figure 12.2 shows the timing diagram. When released from STOP mode, the Basic interval timer is activated on wake-up. Therefore, before STOP instruction, user must be set its relevant prescale divide ratio to have long enough time. This guarantees that oscillator has started and stabilized.

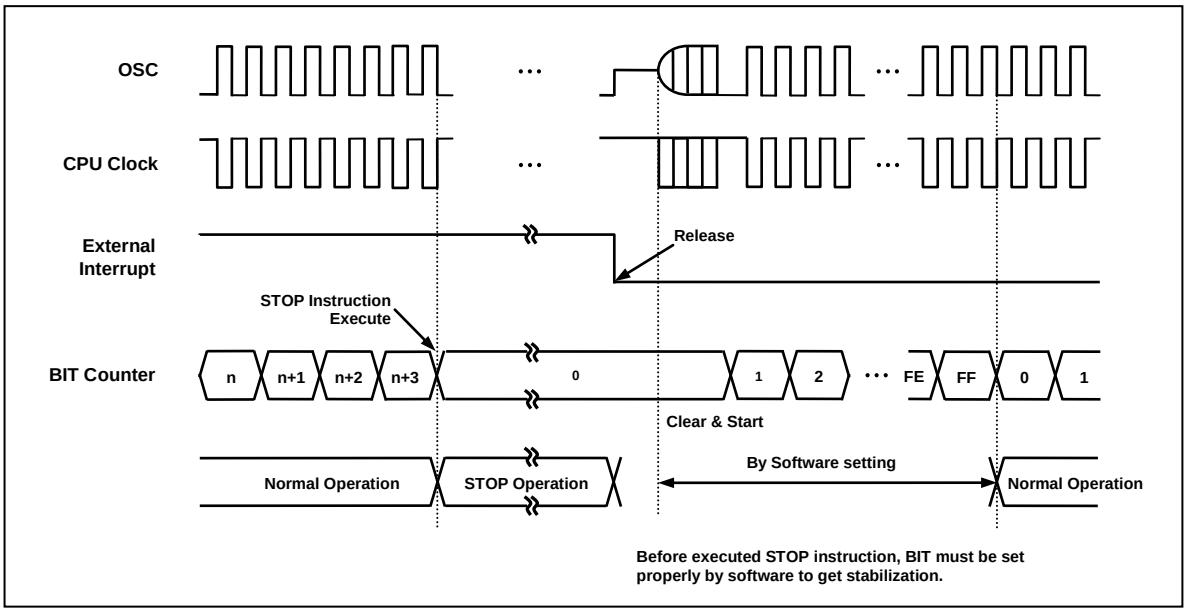


Figure 12.2 STOP Mode Release Timing by External Interrupt

12.5 Release Operation of STOP Mode

After STOP mode is released, the operation begins according to content of related interrupt register just before STOP mode start (Figure 12.3). If the global interrupt Enable Flag (IE.EA) is set to `1`, the STOP mode is released by the interrupt which each interrupt enable flag = `1` and the CPU jumps to the relevant interrupt service routine. Even if the IE.EA bit is cleared to '0', the STOP mode is released by the interrupt of which the interrupt enable flag is set to '1'.

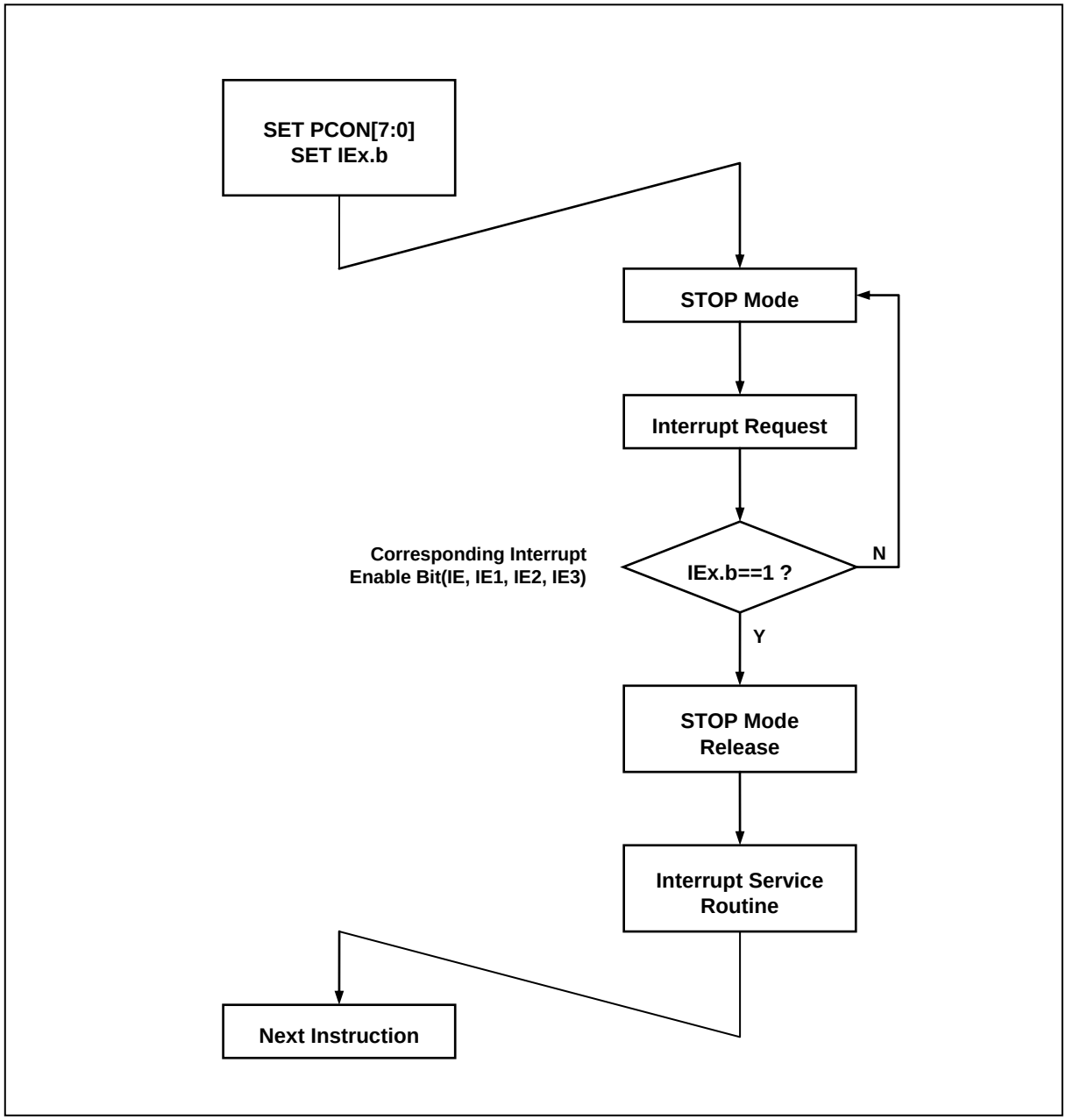


Figure 12.3 STOP Mode Release Flow

12.5.1 Register Map

Table 12-2 Power Down Operation Register Map

Name	Address	Dir	Default	Description
PCON	87H	R/W	00H	Power Control Register

12.5.2 Power Down Operation Register Description

The power down operation register consists of the power control register (PCON).

12.5.3 Register Description for Power Down Operation

PCON (Power Control Register) : 87H

7	6	5	4	3	2	1	0
PCON7	–	–	–	PCON3	PCON2	PCON1	PCON0
R/W	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

PCON[7:0]	Power Control	
	01H	IDLE mode enable
	03H	STOP mode enable
	Other Values	Normal operation

NOTES) 1. To enter IDLE mode, PCON must be set to ‘01H’.

2. To enter STOP mode, PCON must be set to ‘03H’.

3. The PCON register is automatically cleared by a release signal in STOP/IDLE mode.

4. Three or more NOP instructions must immediately follow STOP/IDLE mode like the below examples.

```
Ex1)  MOV   PCON, #01H   ; IDLE mode
      NOP
      NOP
      NOP
       •
       •
       •
```

```
Ex2)  MOV   PCON, #03H   ; STOP mode
      NOP
      NOP
      NOP
       •
       •
       •
```

13. RESET

13.1 Overview

The following is the hardware setting value.

Table 13-1 Reset State

On Chip Hardware	Initial Value
Program Counter (PC)	0000h
Accumulator	00h
Stack Pointer (SP)	07h
Peripheral Clock	On
Control Register	Refer to the Peripheral Registers

13.2 Reset Source

The MC97F2664 has five types of reset sources. The following is the reset sources.

- External RESETB
- Power ON RESET (POR)
- WDT Overflow Reset (In the case of WDTEN = `1`)
- Low Voltage Reset (In the case of LVREN = `0 `)
- OCD Reset

13.3 RESET Block Diagram

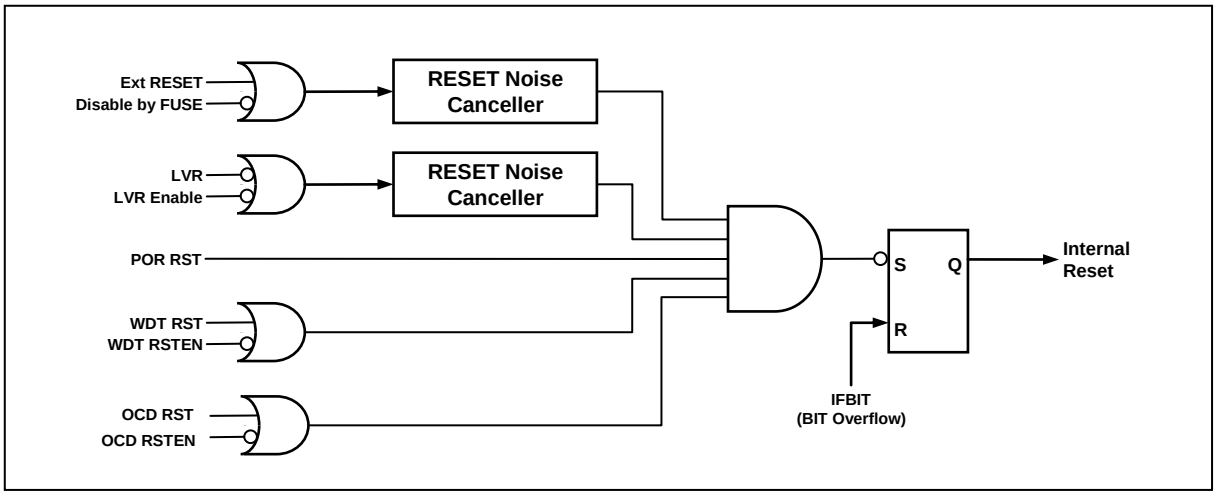


Figure 13.1 RESET Block Diagram

13.4 RESET Noise Canceller

The Figure 13.2 is the noise canceller diagram for noise cancellation of RESET. It has the noise cancellation value of about 2us (@V_{DD}=5V) to the low input of system reset.

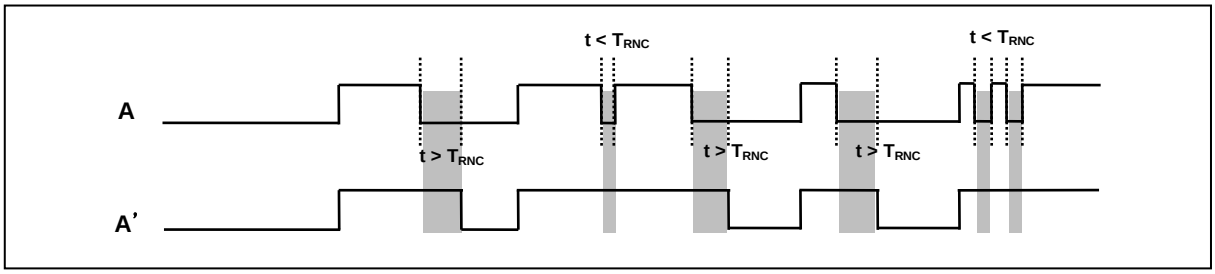


Figure 13.2 Reset noise canceller timer diagram

13.5 Power on RESET

When rising device power, the POR (Power On Reset) has a function to reset the device. If POR is used, it executes the device RESET function instead of the RESET IC or the RESET circuits.

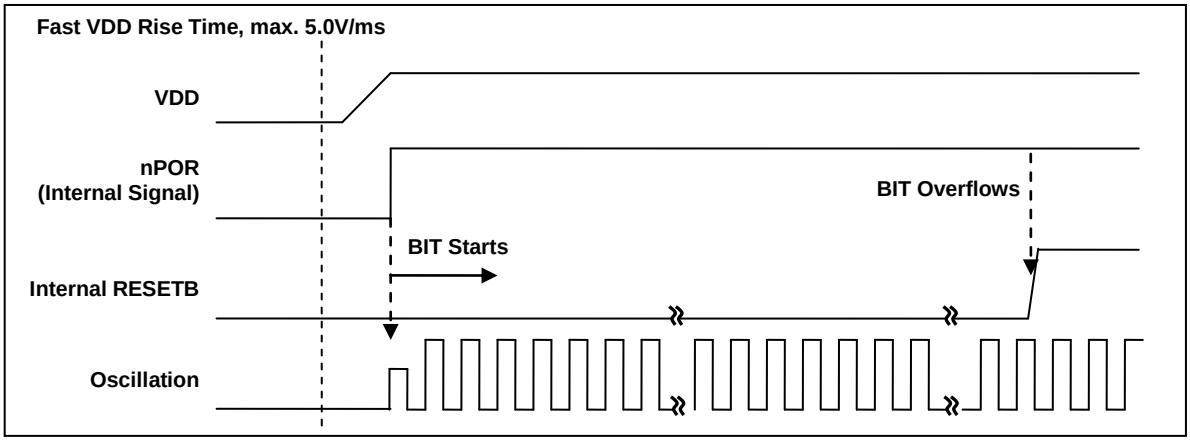


Figure 13.3 Fast VDD Rising Time

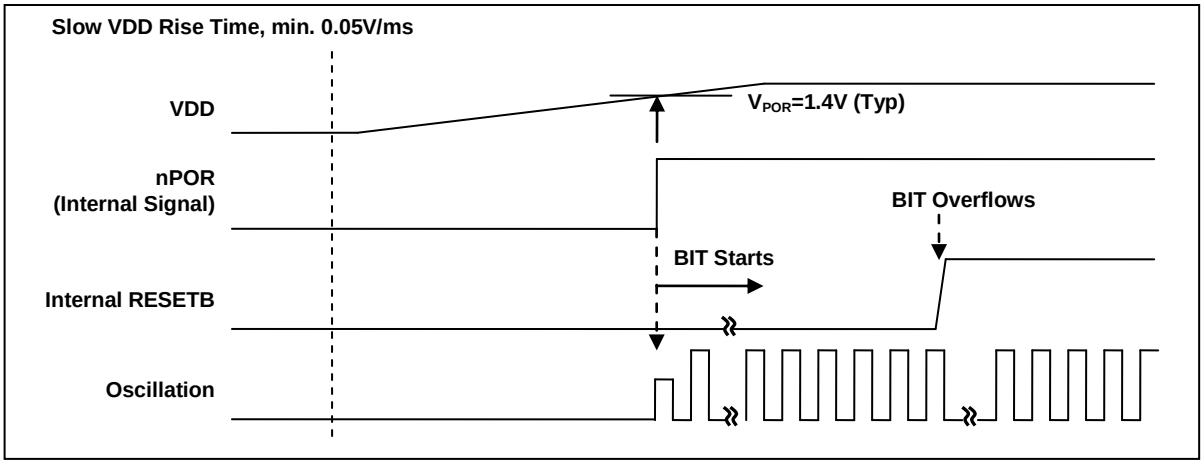


Figure 13.4 Internal RESET Release Timing On Power-Up

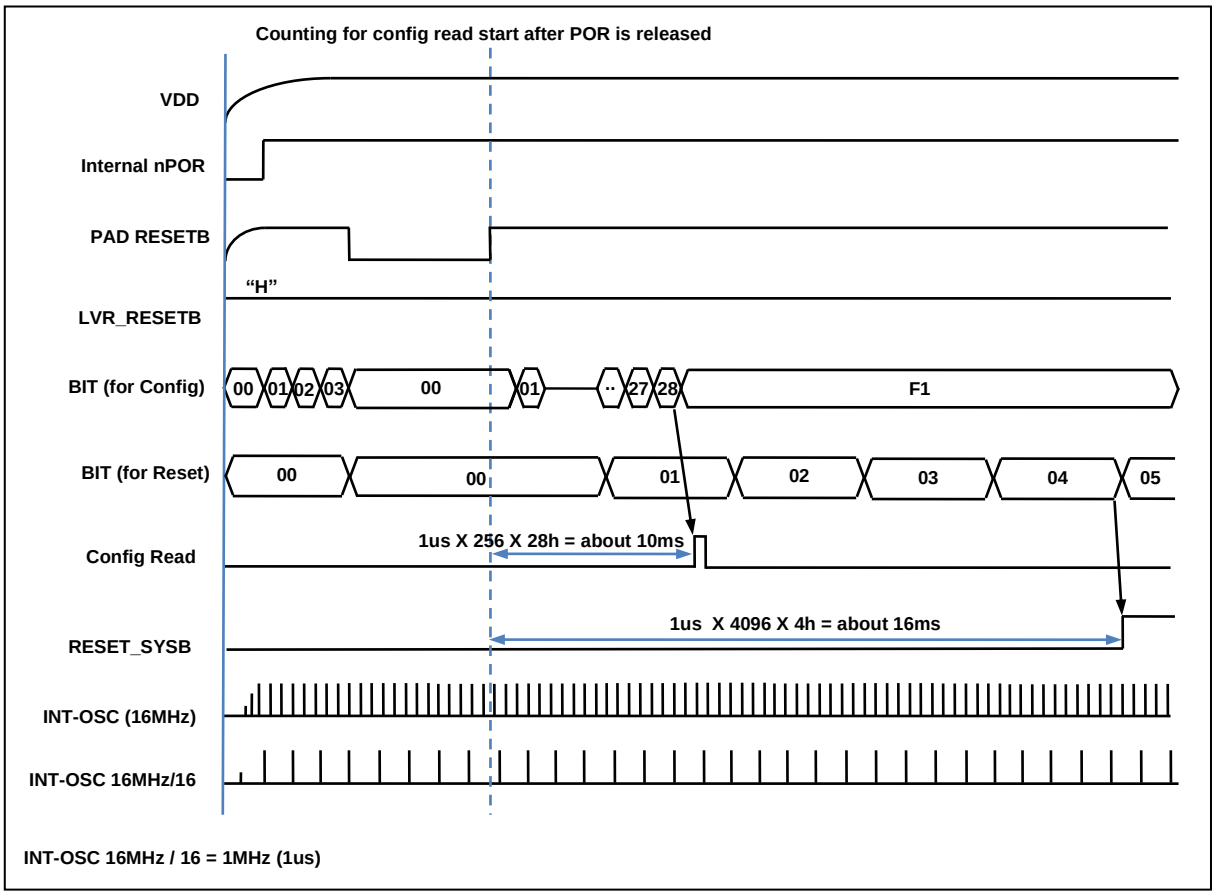


Figure 13.5 Configuration Timing when Power-on

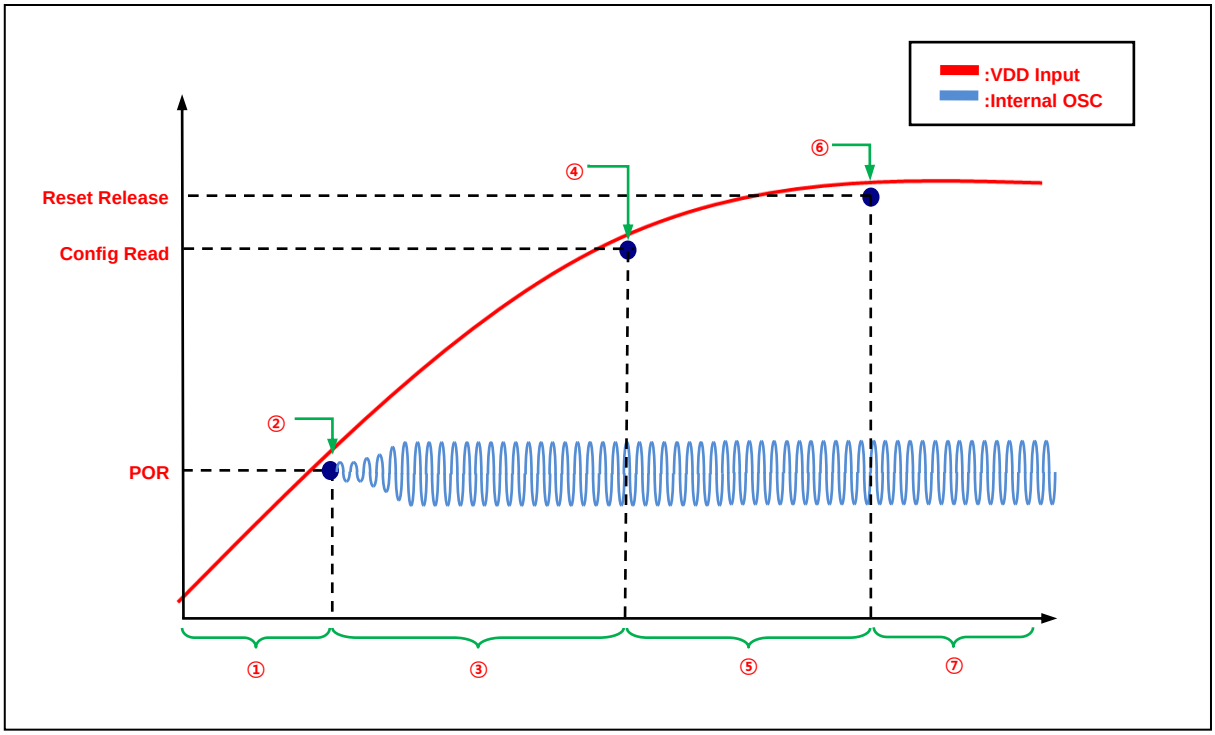


Figure 13.6 Boot Process WaveForm

Table 13-2 Boot Process Description

Process	Description	Remarks
①	-No Operation	
②	-1st POR level Detection	-about 1.4V
③	- (INT-OSC 16MHz/16)x256x28h Delay section (=10ms) -VDD input voltage must rise over than flash operating voltage for Config read	-Slew Rate >= 0.05V/ms
④	- Config read point	-about 1.5V ~ 1.6V -Config Value is determined by Writing Option
⑤	- Rising section to Reset Release Level	-16ms point after POR or Ext_reset release
⑥	- Reset Release section (BIT overflow) i) after16ms, after External Reset Release (External reset) ii) 16ms point after POR (POR only)	- BIT is used for Peripheral stability
⑦	-Normal operation	

13.6 External RESETB Input

The External RESETB is the input to a Schmitt trigger. If RESETB pin is held with low for at least 10us over within the operating voltage range and stable oscillation, it is applied and the internal state is initialized. After reset state becomes '1', it needs the stabilization time with 16ms and after the stable state, the internal RESET becomes '1'. The Reset process step needs 5 oscillator clocks. And the program execution starts at the vector address stored at address 0000H.

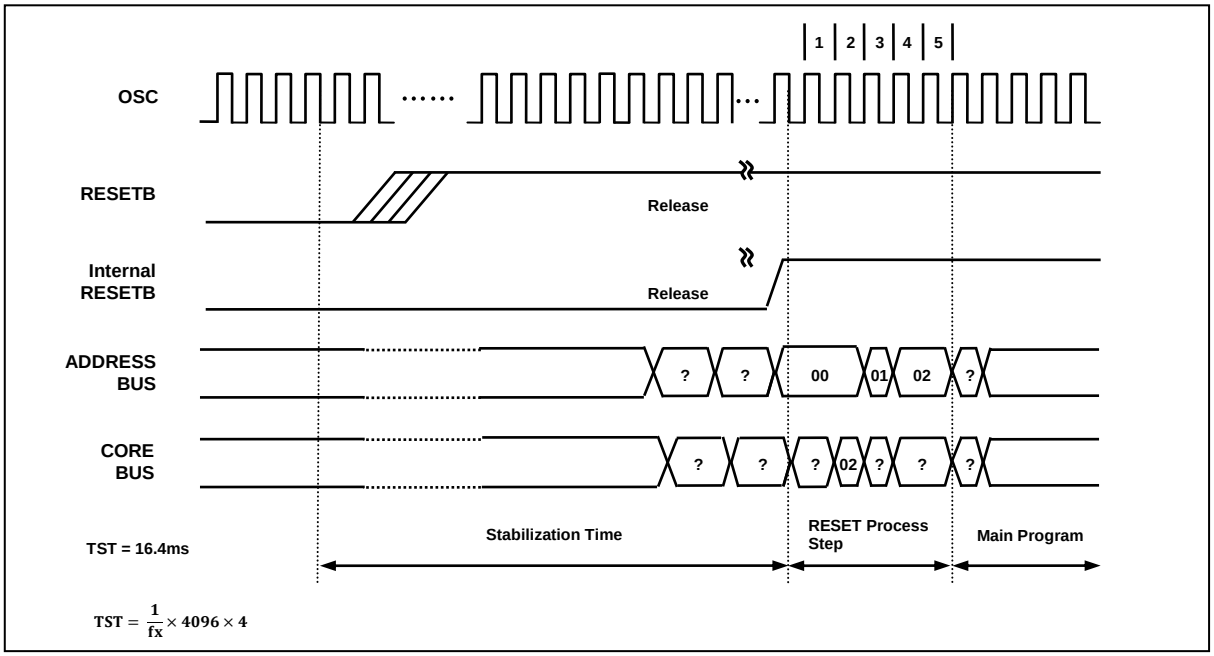


Figure 13.7 Timing Diagram after RESET

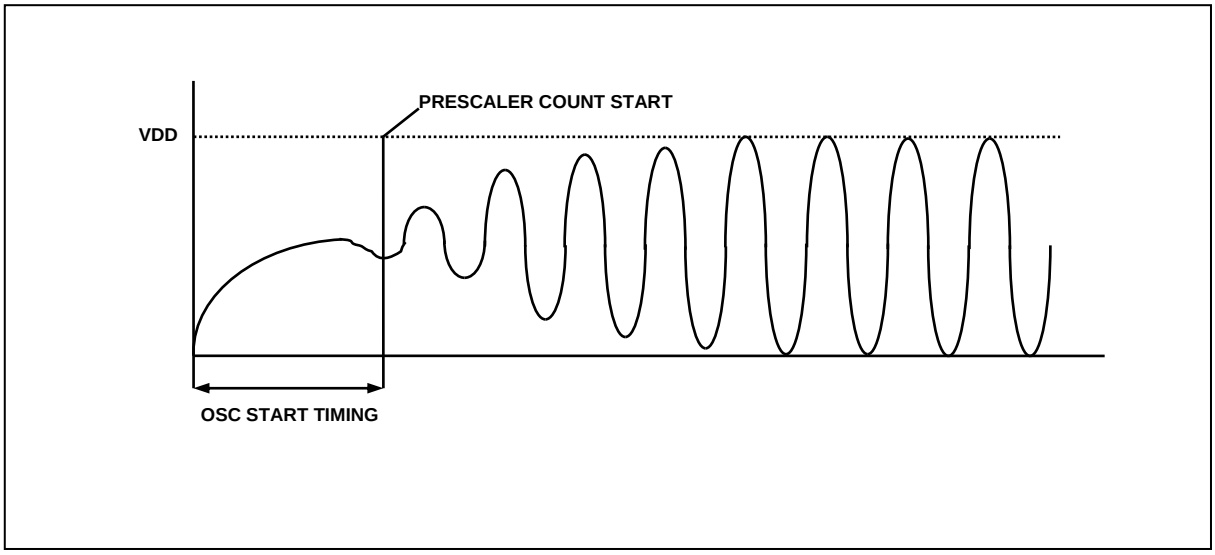


Figure 13.8 Oscillator generating waveform example

NOTE) As shown Figure 13.8, the stable generating time is not included in the start-up time.
The RESETB pin has a Pull-up register by H/W.

13.7 Brown Out Detector Processor

The MC97F2664 has an On-chip brown-out detection circuit (BOD) for monitoring the VDD level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by LVRVS[3:0] bit to be 1.60V/ 2.00V/ 2.10V/ 2.20V/2.32V/ 2.44V/ 2.59V/ 2.75V/ 2.93V/ 3.14V/ 3.38V/ 3.67V/ 4.00V/ 4.40V. In the STOP mode, this will contribute significantly to the total current consumption. So to minimize the current consumption, the LVREN bit is set to off by software.

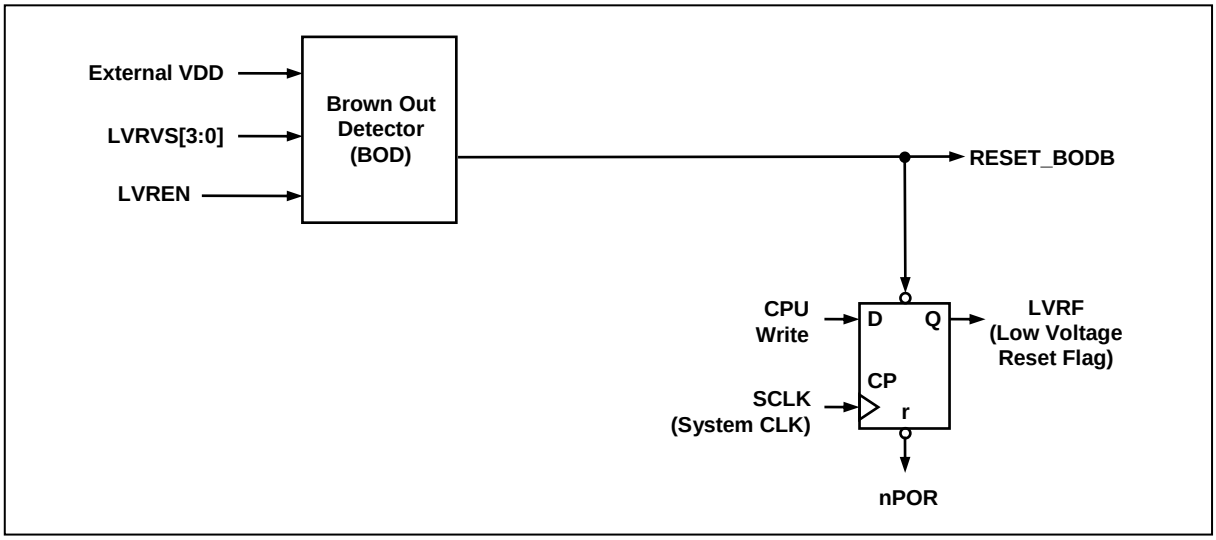


Figure 13.9 Block Diagram of BOD

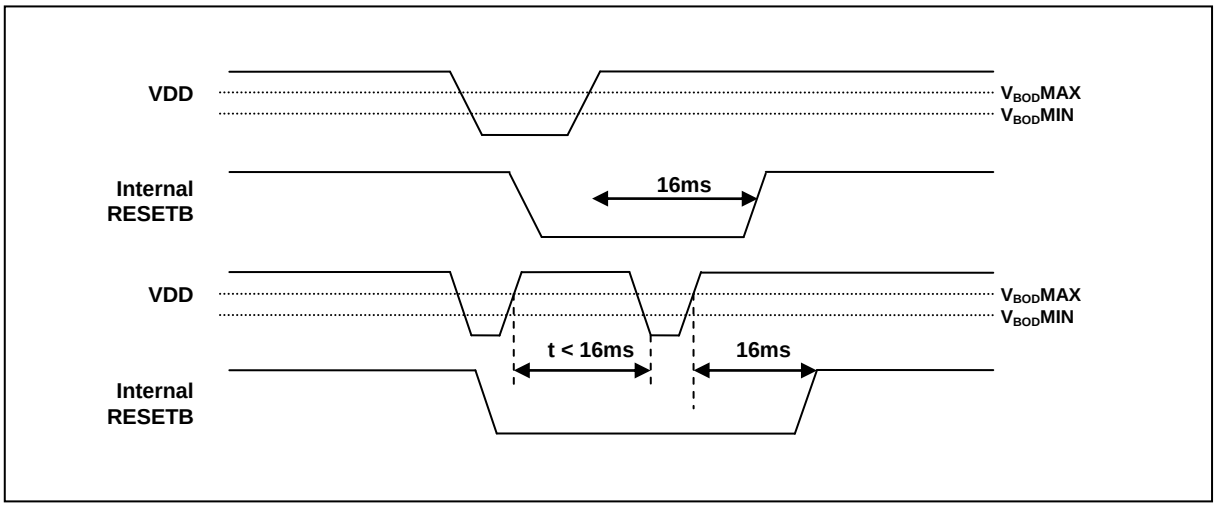


Figure 13.10 Internal Reset at the power fail situation

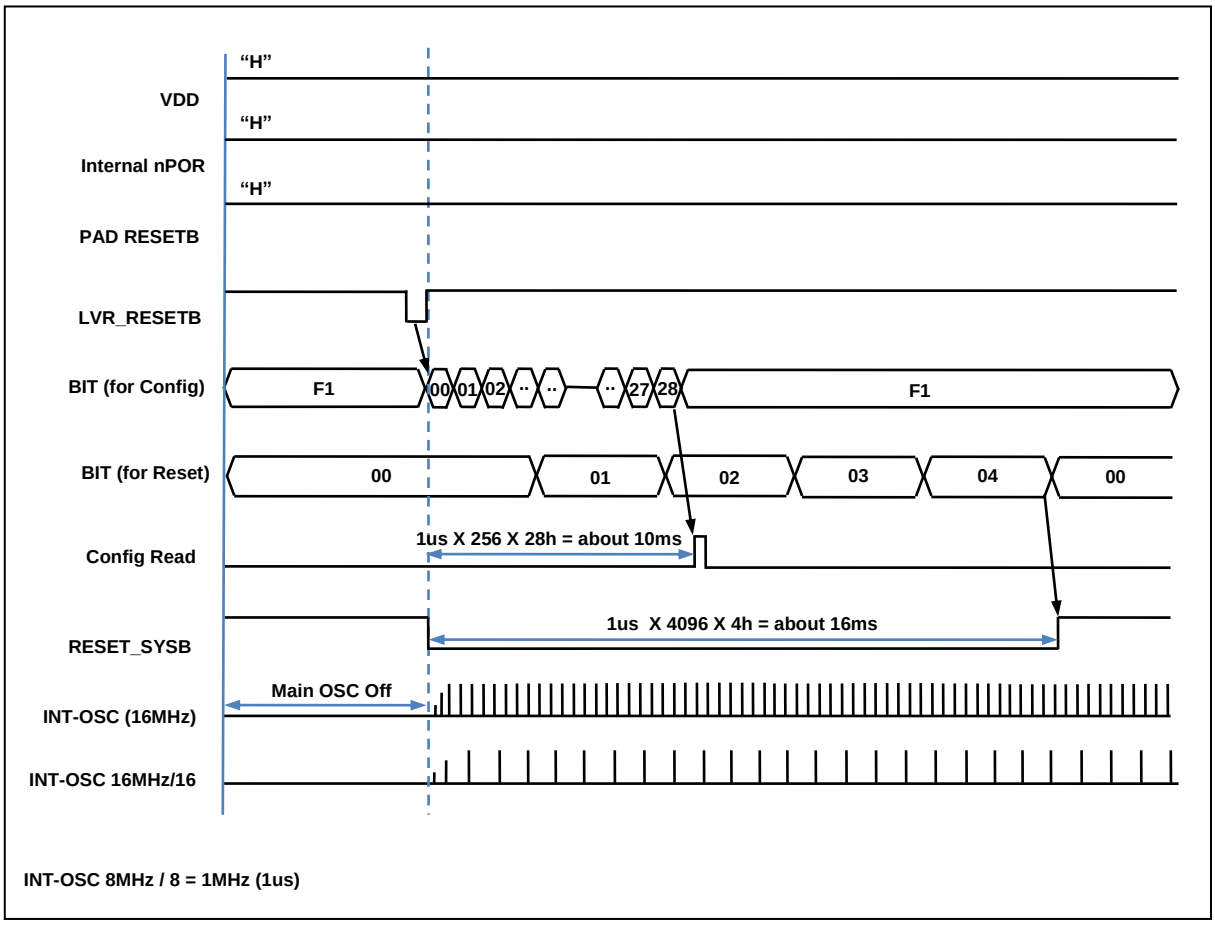


Figure 13.11 Configuration timing when BOD RESET

13.8 LVI Block Diagram

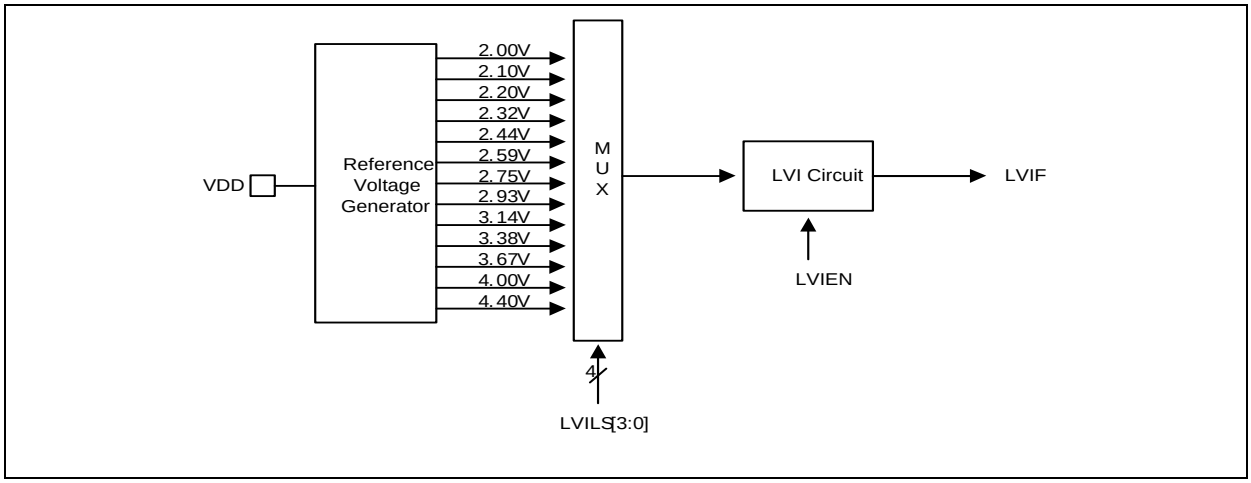


Figure 13.12 LVI Diagram

13.9 Register Map

Table 13-3 Reset Operation Register Map

Name	Address	Dir	Default	Description
RSTFR	E8H	R/W	80H	Reset Flag Register
LVRCR	D8H	R/W	00H	Low Voltage Reset Control Register
LVICR	86H	R/W	00H	Low Voltage Indicator Control Register

13.10 Reset Operation Register Description

The reset control register consists of the reset flag register (RSTFR), low voltage reset control register (LVRCR), and low voltage indicator control register (LVICR).

13.11 Register Description for Reset Operation

RSTFR (Reset Flag Register) : E8H

7	6	5	4	3	2	1	0
PORF	EXTRF	WDTRF	OCDRF	LVRF	–	–	–
R/W	R/W	R/W	R/W	R/W	–	–	–

Initial value : 80H

PORF Power-On Reset flag bit. The bit is reset by writing '0' to this bit.

0 No detection

1 Detection

EXTRF External Reset (RESETB) flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

WDTRF Watch Dog Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

OCDRF On-Chip Debug Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

LVRF Low Voltage Reset flag bit. The bit is reset by writing '0' to this bit or by Power-On Reset.

0 No detection

1 Detection

NOTES) 1. When the Power-On Reset occurs, the PORF bit is only set to "1", the other flag (WDTRF and OCDRF) bits are all cleared to "0".

2. When the Power-On Reset occurs, the EXTRF bit is unknown, At that time, the EXTRF bit can be set to "1" when External Reset (RESETB) occurs.

3. When the Power-On Reset occurs, the LVRF bit is unknown, At that time, the LVRF bit can be set to "1" when LVR Reset occurs.

4. When a reset except the POR occurs, the corresponding flag bit is only set to "1", the other flag bits are kept in the previous values.

LVRCR (Low Voltage Reset Control Register) : D8H

7	6	5	4	3	2	1	0
LVRST	–	–	LVRVS3	LVRVS2	LVRVS1	LVRVS0	LVREN
R/W	–	–	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

LVRST LVR Enable when Stop Release

0 Not effect at stop release

1 LVR enable at stop release

NOTES)

When this bit is '1', the LVREN bit is cleared to '0' by stop mode to release. (LVR enable)

When this bit is '0', the LVREN bit is not effect by stop mode to release.

LVRVS[3:0] LVR Voltage Select

LVRVS3	LVRVS2	LVRVS1	LVRVS0	Description
0	0	0	0	1.60V
0	0	0	1	2.00V
0	0	1	0	2.10V
0	0	1	1	2.20V
0	1	0	0	2.32V
0	1	0	1	2.44V
0	1	1	0	2.59V
0	1	1	1	2.75V
1	0	0	0	2.93V
1	0	0	1	3.14V
1	0	1	0	3.38V
1	0	1	1	3.67V
1	1	0	0	4.00V
1	1	0	1	4.40V
Other Values				Not available

LVREN LVR Operation

0 LVR Enable

1 LVR Disable

NOTES) 1. The LVRST, LVRVS[3:0] bits are cleared by a power-on reset but are retained by other reset signals.

2. The LVRVS[3:0] bits should be set to '0000b' while LVREN bit is "1".

LVICR (Low Voltage Indicator Control Register) : 86H

7	6	5	4	3	2	1	0
–	–	LVIF	LVIEN	LVILS3	LVILS2	LVILS1	LVILS0
–	–	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

LVIF Low Voltage Indicator Flag Bit

0 No detection

1 Detection

LVIEN LVI Enable/Disable

0 Disable

1 Enable

LVILS[3:0] LVI Level Select

LVILS3	LVILS2	LVILS1	LVILS0	Description
0	0	0	0	2.00V
0	0	0	1	2.10V
0	0	1	0	2.20V
0	0	1	1	2.32V
0	1	0	0	2.44V
0	1	0	1	2.59V
0	1	1	0	2.75V
0	1	1	1	2.93V
1	0	0	0	3.14V
1	0	0	1	3.38V
1	0	1	0	3.67V
1	0	1	1	4.00V
1	1	0	0	4.40V
Other Values				Not available

14. On-chip Debug System

14.1 Overview

14.1.1 Description

On-chip debug system (OCD2) of MC97F2664 can be used for programming the non-volatile memories and on-chip debugging. Detail descriptions for programming via the OCD2 interface can be found in the following chapter.

Figure 14.1 shows a block diagram of the OCD2 interface and the On-chip Debug system.

14.1.2 Feature

- Two-wire external interface: 1-wire serial clock input, 1-wire bi-directional serial data bus
- Debugger Access to:
  - All Internal Peripheral Units
  - Internal data RAM
  - Program Counter
  - Flash and Data EEPROM Memories
- Extensive On-chip Debug Support for Break Conditions, Including
  - Break Instruction
  - Single Step Break
  - Program Memory Break Points on Single Address
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the two-wire Interface
  - On-chip Debugging Supported by Dr.Choice®
- Operating frequency

  Supports the maximum frequency of the target MCU

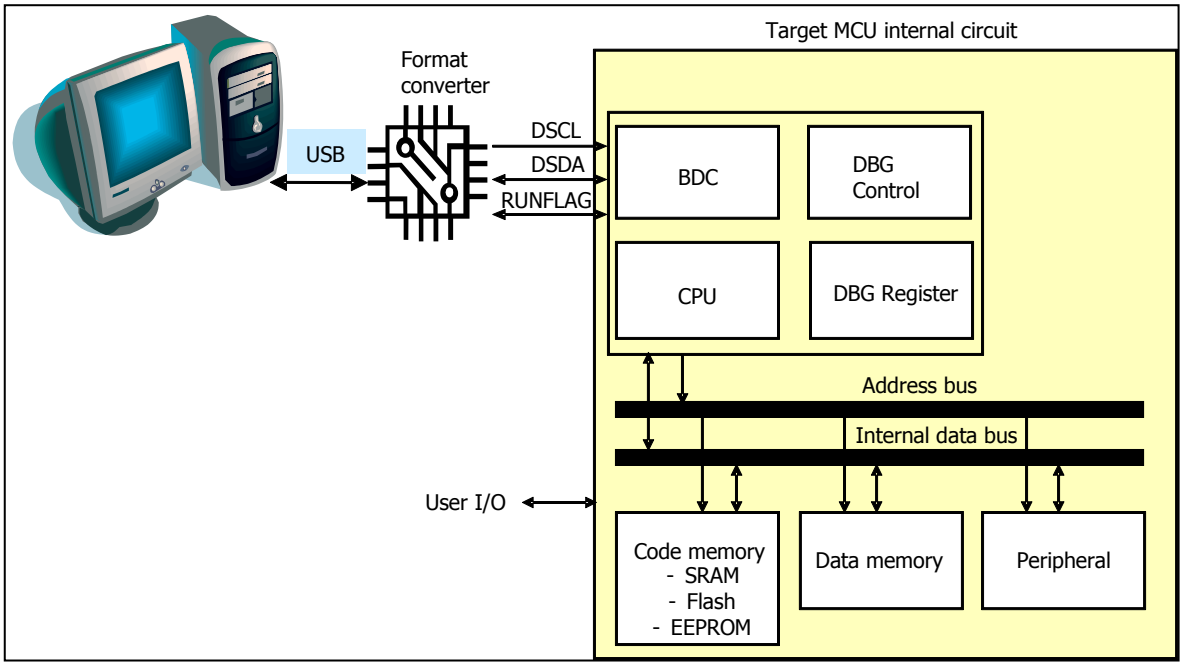


Figure 14.1 Block Diagram of On-Chip Debug System

14.2 Two-Pin External Interface

14.2.1 Basic Transmission Packet

- 10-bit packet transmission using two-pin interface.
- 1-packet consists of 8-bit data, 1-bit parity and 1-bit acknowledge.
- Parity is even of '1' for 8-bit data in transmitter.
- Receiver generates acknowledge bit as '0' when transmission for 8-bit data and its parity has no error.
- When transmitter has no acknowledge (Acknowledge bit is '1' at tenth clock), error process is executed in transmitter.
- When acknowledge error is generated, host PC makes stop condition and transmits command which has error again.
- Background debugger command is composed of a bundle of packet.
- Start condition and stop condition notify the start and the stop of background debugger command respectively.

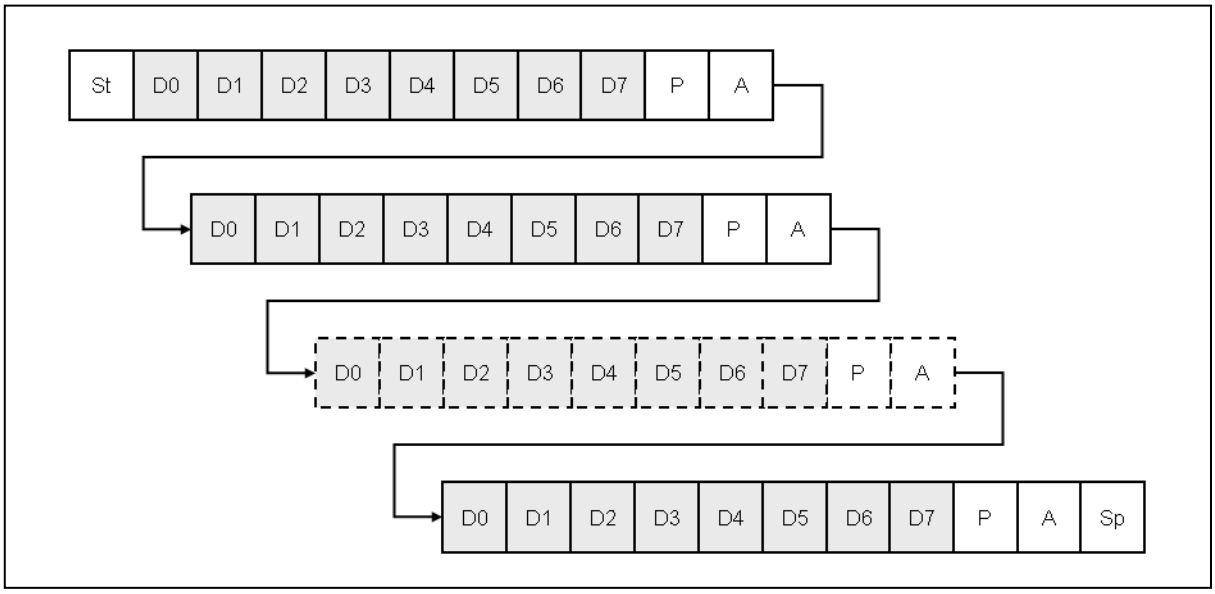


Figure 14.2 10-bit Transmission Packet

14.2.2 Packet Transmission Timing

14.2.2.1 Data Transfer

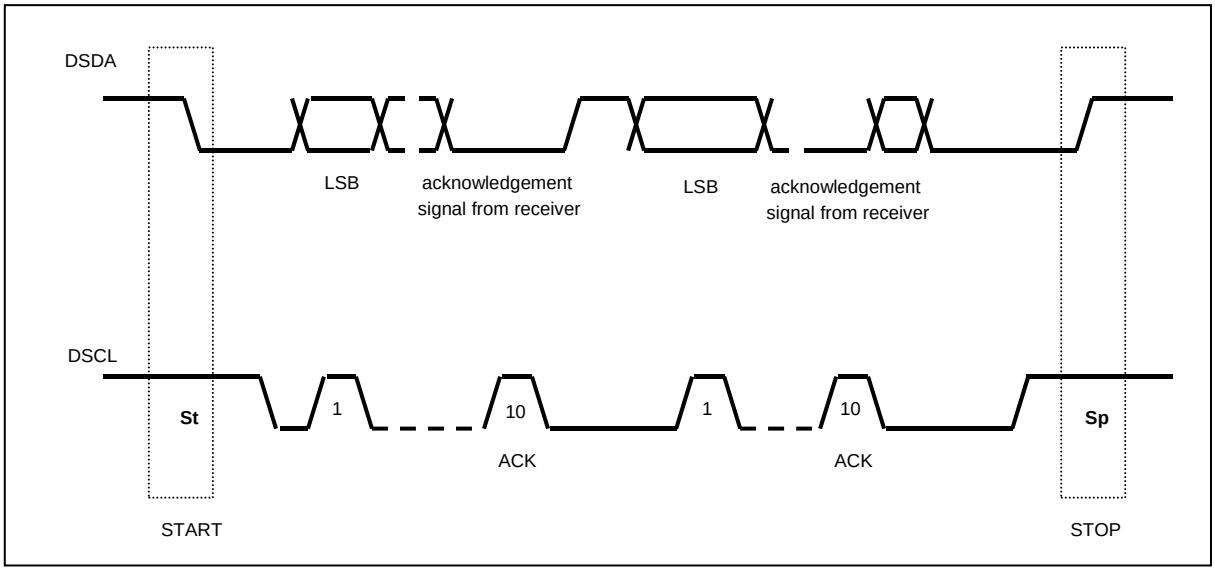


Figure 14.3 Data Transfer on the Twin Bus

14.2.2.2 Bit Transfer

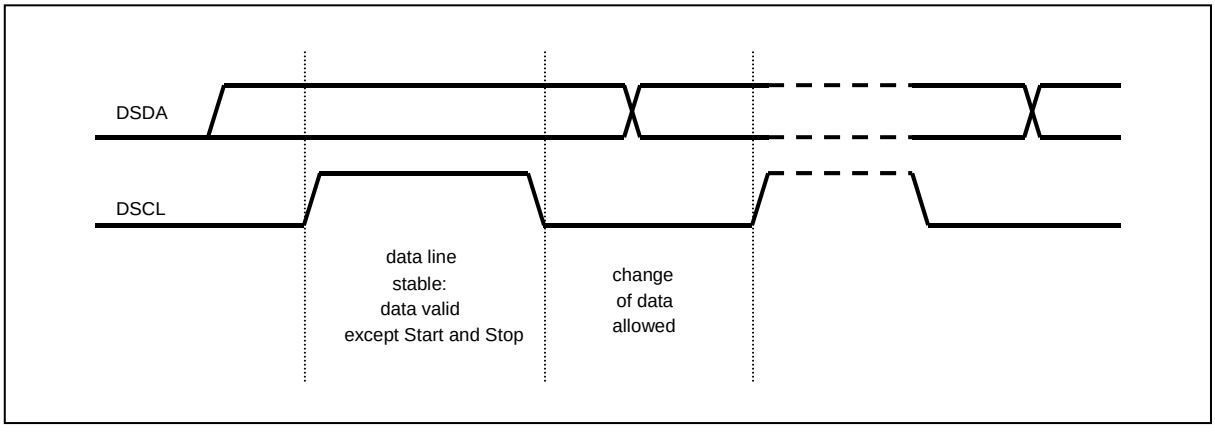


Figure 14.4 Bit Transfer on the Serial Bus

14.2.2.3 Start and Stop Condition

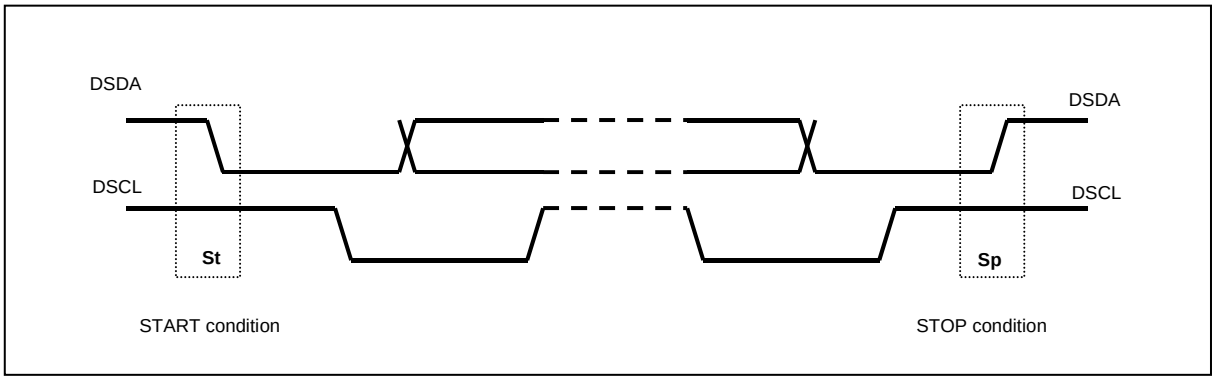


Figure 14.5 Start and Stop Condition

14.2.2.4 Acknowledge Bit

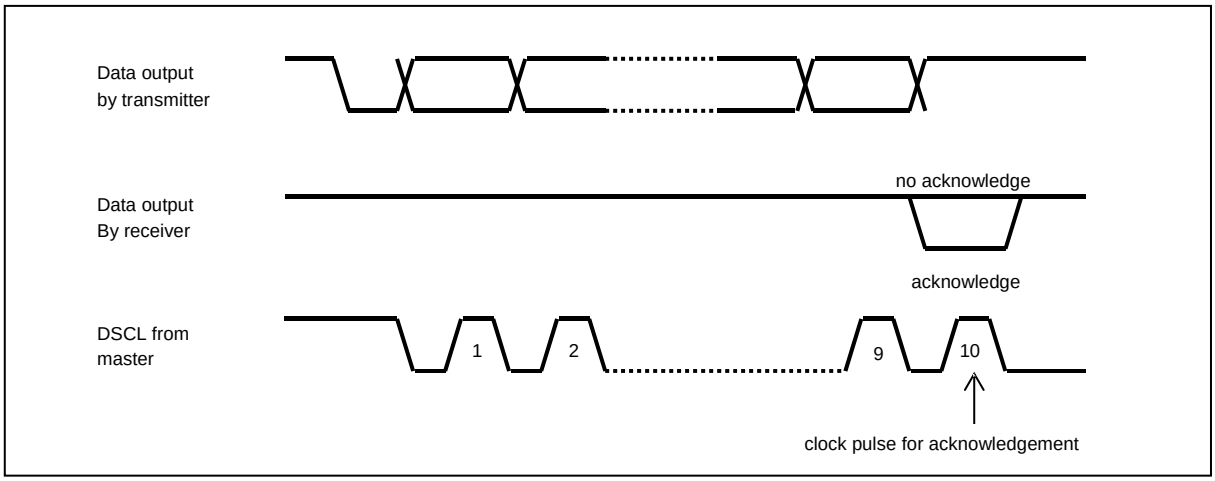


Figure 14.6 Acknowledge on the Serial Bus

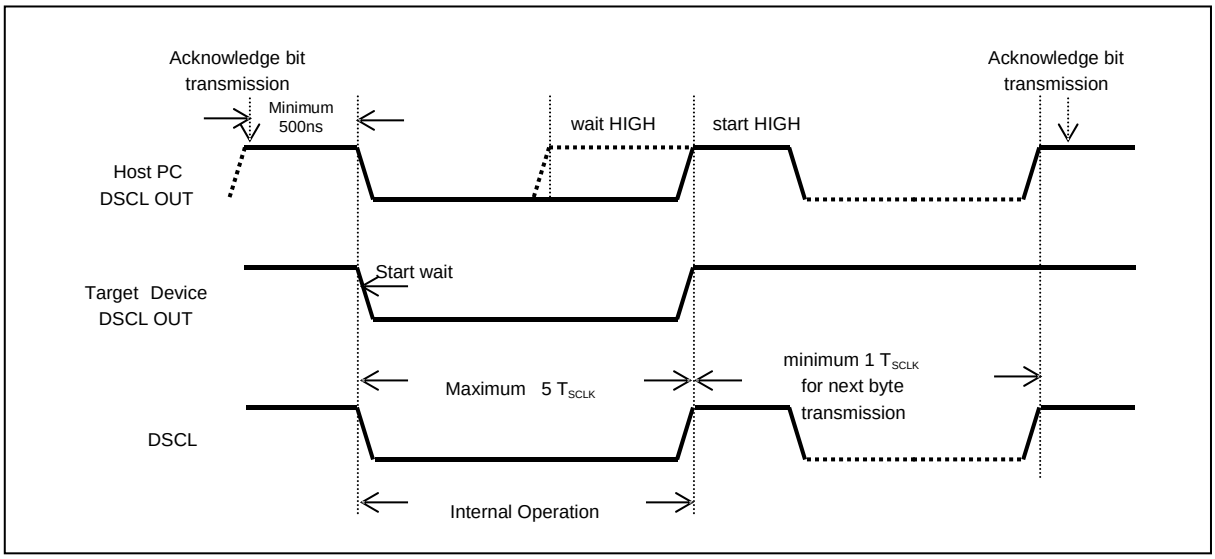


Figure 14.7 Clock Synchronization during Wait Procedure

14.2.3 Connection of Transmission

Two-pin interface connection uses open-drain (wire-AND bidirectional I/O).

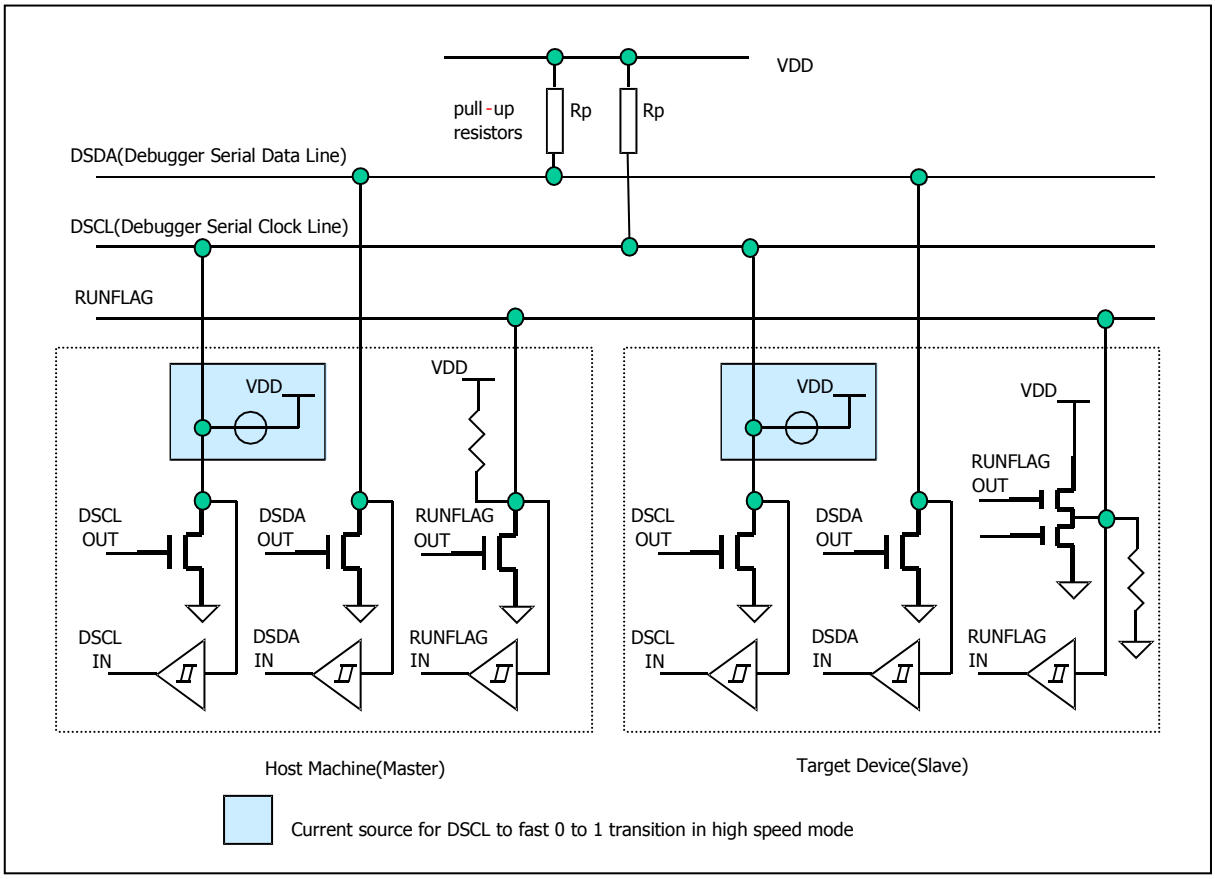


Figure 14.8 Connection of Transmission

15. Flash Memory

15.1 Overview

15.1.1 Description

MC97F2664 incorporates flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory can be read by 'MOVC' instruction and it can be programmed in OCD2, serial ISP mode or user program mode.

- Flash Size : 64k bytes
- Single power supply program and erase
- Command interface for fast program and erase operation
- Up to 10,000(sector 0~1019)/100,000(sector 1020~1023) program/erase at typical voltage and temperature for flash memory

15.1.2 Flash Program ROM Structure

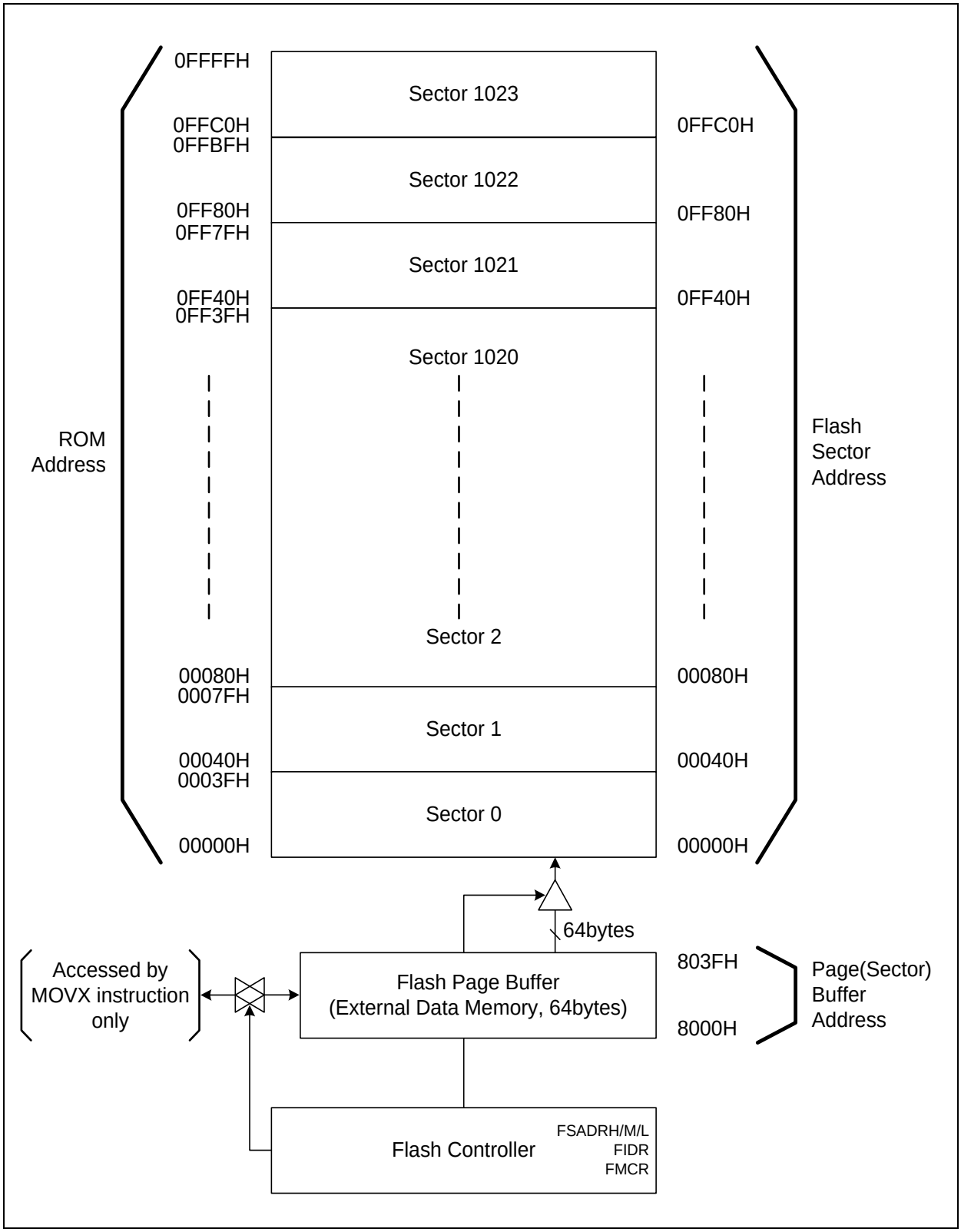


Figure 15.1 Flash Program ROM Structure

15.1.3 Register Map

Table 15-1 Flash Memory Register Map

Name	Address	Dir	Default	Description
FSADRH	FAH	R/W	00H	Flash Sector Address High Register
FSADRM	FBH	R/W	00H	Flash Sector Address Middle Register
FSADRL	FCH	R/W	00H	Flash Sector Address Low Register
FIDR	FDH	R/W	00H	Flash Identification Register
FMCR	FEH	R/W	00H	Flash Mode Control Register

15.1.4 Register Description for Flash Memory Control and Status

Flash control register consists of the flash sector address high register (FSADRH), flash sector address middle register (FSADRM), flash sector address low register (FSADRL), flash identification register (FIDR), and flash mode control register (FMCR). They are mapped to SFR area and can be accessed only in programming mode.

15.1.5 Register Description for Flash

FSADRH (Flash Sector Address High Register) : FAH

7	6	5	4	3	2	1	0
–	–	–	–	FSADRH3	FSADRH2	FSADRH1	FSADRH0
–	–	–	–	R/W	R/W	R/W	R/W

Initial value : 00H

FSADRH[3:0] Flash Sector Address High

FSADRM (Flash Sector Address Middle Register) : FBH

7	6	5	4	3	2	1	0
FSADRM7	FSADRM6	FSADRM5	FSADRM4	FSADRM3	FSADRM2	FSADRM1	FSADRM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSADRM[7:0] Flash Sector Address Middle

FSADRL (Flash Sector Address Low Register) : FCH

7	6	5	4	3	2	1	0
FSADRL7	FSADRL6	FSADRL5	FSADRL4	FSADRL3	FSADRL2	FSADRL1	FSADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FSADRL[7:0] Flash Sector Address Low

FIDR (Flash Identification Register) : FDH

7	6	5	4	3	2	1	0
FIDR7	FIDR6	FIDR5	FIDR4	FIDR3	FIDR2	FIDR1	FIDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value : 00H

FIDR[7:0] Flash Identification

Others No identification value

10100101 Identification value for a flash mode

(These bits are automatically cleared to logic ‘00H’ immediately after one time operation except “flash page buffer reset mode”)

FMCR (Flash Mode Control Register) : FEH

7	6	5	4	3	2	1	0
FMBUSY	–	–	–	–	FMCR2	FMCR1	FMCR0
R	–	–	–	–	R/W	R/W	R/W

Initial value : 00H

FMBUSY Flash Mode Busy Bit. This bit will be used for only debugger.

0 No effect when “1” is written

1 Busy

FMCR[2:0] Flash Mode Control Bits. During a flash mode operation, the CPU is hold and the global interrupt is on disable state regardless of the IE.7 (EA) bit.

FMCR2	FMCR1	FMCR0	Description
0	0	1	Select flash page buffer reset mode and start regardless of the FIDR value (Clear all 32bytes to ‘0’)
0	1	0	Select flash sector erase mode and start operation when the FIDR=”10100101b’
0	1	1	Select flash sector write mode and start operation when the FIDR=”10100101b’
1	0	0	Select flash sector Code Write Protection and start operation when the FIDR=”10100101b’

Others Values: No operation

(These bits are automatically cleared to logic ‘00H’ immediately after one time operation)

15.1.6 Serial In-System Program (ISP) Mode

Serial in-system program uses the interface of debugger which uses two wires. Refer to chapter 14 in details about debugger

15.1.7 Protection Area (User program mode)

MC97F2664 can program its own flash memory (protection area). The protection area can not be erased or programmed. The protection areas are available only when the PAEN bit is cleared to '0', that is, enable protection area at the configure option 2 if it is needed. If the protection area isn't enabled (PAEN ='1'), this area can be used as a normal program memory.

The size of protection area can be varied by setting of configure option 2.

Table 15-2 Protection Area size

Protection Area Size Select		Size of Protection Area	Address of Protection Area
PASS1	PASS0		
0	0	3.8k Bytes	0100H – 0FFFH
0	1	1.7k Bytes	0100H – 07FFH
1	0	768 Bytes	0100H – 03FFH
1	1	256 Bytes	0100H – 01FFH

NOTE) Refer to chapter 16 in configure option control.

15.1.8 Erase Mode

The sector erase program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write '0' to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Erase verify

Program Tip – sector erase

```
            MOV     FMCR,#0x01          ;page buffer clear
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     DPH,#0x80
            MOV     DPL,#0

Pgbuf_clr:  MOVX    @DPTR,A
            INC     DPTR
            DJNZ    R0, Pgbuf_clr       ;Write '0' to all page buffer

            MOV     FSADRH,#0x00
            MOV     FSADRM,#0x7F
            MOV     FSADRL,#0x40        ;Select sector 509
            MOV     FIDR,#0xA5          ;Identification value
            MOV     FMCR,#0x02          ;Start flash erase mode
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0                ;erase verify
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     R1,#0
            MOV     DPH,#0x7F
            MOV     DPL,#0x40

Erase_verify:
            MOVC    A,@A+DPTR
            SUBB    A,R1
            JNZ     Verify_error
            INC     DPTR
            DJNZ    R0, Erase_verify


Verify_error:
```

The Byte erase program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write '0' to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Erase verify

Program Tip – byte erase

```
MOV     FMCR,#0x01          ;page buffer clear
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#0
MOV     DPH,#0x80
MOV     DPL,#0
MOVX    @DPTR,A

MOV     DPH,#0x80
MOV     DPL,#0x05
MOVX    @DPTR,A             ;Write '0' to page buffer

MOV     FSADRH,#0x00
MOV     FSADRM,#0x7F
MOV     FSADRL,#0x40        ;Select sector 509
MOV     FIDR,#0xA5          ;Identification value
MOV     FMCR,#0x02          ;Start flash erase mode
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#0                ;erase verify
MOV     R1,#0
MOV     DPH,#0x7F
MOV     DPL,#0x40
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F40 = 0 ?
JNZ     Verify_error

MOV     A,#0
MOV     R1,#0
MOV     DPH,#0x7F
MOV     DPL,#0x45
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F45 = 0 ?
JNZ     Verify_error


Verify_error:
```

15.1.9 Write Mode

The sector Write program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write data to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Write verify

   NOTE) All data of the sector should be “00H” before writing data to a sector

Program Tip – sector write

```
            MOV     FMCR,#0x01          ;page buffer clear
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     DPH,#0x80
            MOV     DPL,#0

Pgbuf_WR:   MOVX    @DPTR,A
            INC     A
            INC     DPTR
            DJNZ    R0, Pgbuf_WR        ;Write data to all page buffer

            MOV     FSADRH,#0x00
            MOV     FSADRM,#0x7F
            MOV     FSADRL,#0x40        ;Select sector 509
            MOV     FIDR,#0xA5          ;Identification value
            MOV     FMCR,#0x03          ;Start flash write mode
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.
            NOP                         ;Dummy instruction, This instruction must be needed.

            MOV     A,#0                ;write verify
            MOV     R0,#64              ;Sector size is 64bytes
            MOV     R1,#0
            MOV     DPH,#0x7F
            MOV     DPL,#0x40

Write_verify:
            MOVC    A,@A+DPTR
            SUBB    A,R1
            JNZ     Verify_error
            INC     R1
            INC     DPTR
            DJNZ    R0, Write_verify


Verify_error:
```

The Byte Write program procedure in user program mode

1. Page buffer clear (FMCR=0x01)
2. Write data to page buffer
3. Set flash sector address register (FSADRH/FSADRM/FSADRL).
4. Set flash identification register (FIDR).
5. Set flash mode control register (FMCR).
6. Write verify

   NOTE) Data of the address should be “00H” before writing data to an address

Program Tip – byte write

```
MOV     FMCR,#0x01          ;page buffer clear
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#5
MOV     DPH,#0x80
MOV     DPL,#0
MOVX    @DPTR,A             ;Write data to page buffer

MOV     A,#6
MOV     DPH,#0x80
MOV     DPL,#0x05
MOVX    @DPTR,A             ;Write data to page buffer

MOV     FSADRH,#0x00
MOV     FSADRM,#0x7F
MOV     FSADRL,#0x40        ;Select sector 509
MOV     FIDR,#0xA5          ;Identification value
MOV     FMCR,#0x03          ;Start flash write mode
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.

MOV     A,#0                ;write verify
MOV     R1,#5
MOV     DPH,#0x7F
MOV     DPL,#0x40
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F40 = 5 ?
JNZ     Verify_error

MOV     A,#0
MOV     R1,#6
MOV     DPH,#0x7F
MOV     DPL,#0x45
MOVC    A,@A+DPTR
SUBB    A,R1                ;0x7F45 = 6 ?
JNZ     Verify_error

Verify_error:
```

15.1.10 Protection for Invalid Erase/Write

It should be taken care to the flash erase/write programming in code.
You must make preparations for invalid jump to the flash erase/write code by malfunction, noise, and power off.

1. User ID check routine for the flash erase/write code.

```
ErWt_rtn:
        ---
        MOV     FIDR,#10100101B     ;ID Code
        MOV     A,#ID_DATA_1        ;Ex) ID_DATA_1: 93H, ID_DATA_2: 85H, ID_DATA_3: 5AH
        CJNE    A,UserID1,No_WriteErase
        MOV     A,#ID_DATA_2
        CJNE    A,UserID2,No_WriteErase
        MOV     A,#ID_DATA_3
        CJNE    A,UserID3,No_WriteErase
        MOV     FMCR,#0x??          ;0x03 if write, 0x02 if erase
        ---
        ---
        RET

No_WriteErase:
        MOV     FIDR,#00H
        MOV     UserID1,#00H
        MOV     UserID2,#00H
        MOV     UserID3,#00H
        MOV     Flash_flag,#00H
        RET
```

If code is like the above lines, an invalid flash erase/write can be avoided.

2. It is important where the UserID1/2/3 is written. It will be remain the invalid flash erase/write problem if the UserID1/2/3 is written at the above line of the instruction “MOV FIDR,#10100101B”. So. It had better writing the UserID1/2/3 in another routine after return.

```
Decide_ErWt:
        ---
        MOV     Flash_flag1,#38H    ;Random value for example, in case of erase/write needs
        MOV     FSADRL,#20H         ;Here 20H is example,
        MOV     Flash_flag2,#75H
        RET
```

3. The flash sector address (FSADRH/FSADRM/FSADRL) should always keep the address of the flash which is used for data area. For example, The FSADRH/FSADRM is always 0x00/0x7f" if 0x7f00 to 0x7fff is used for data.

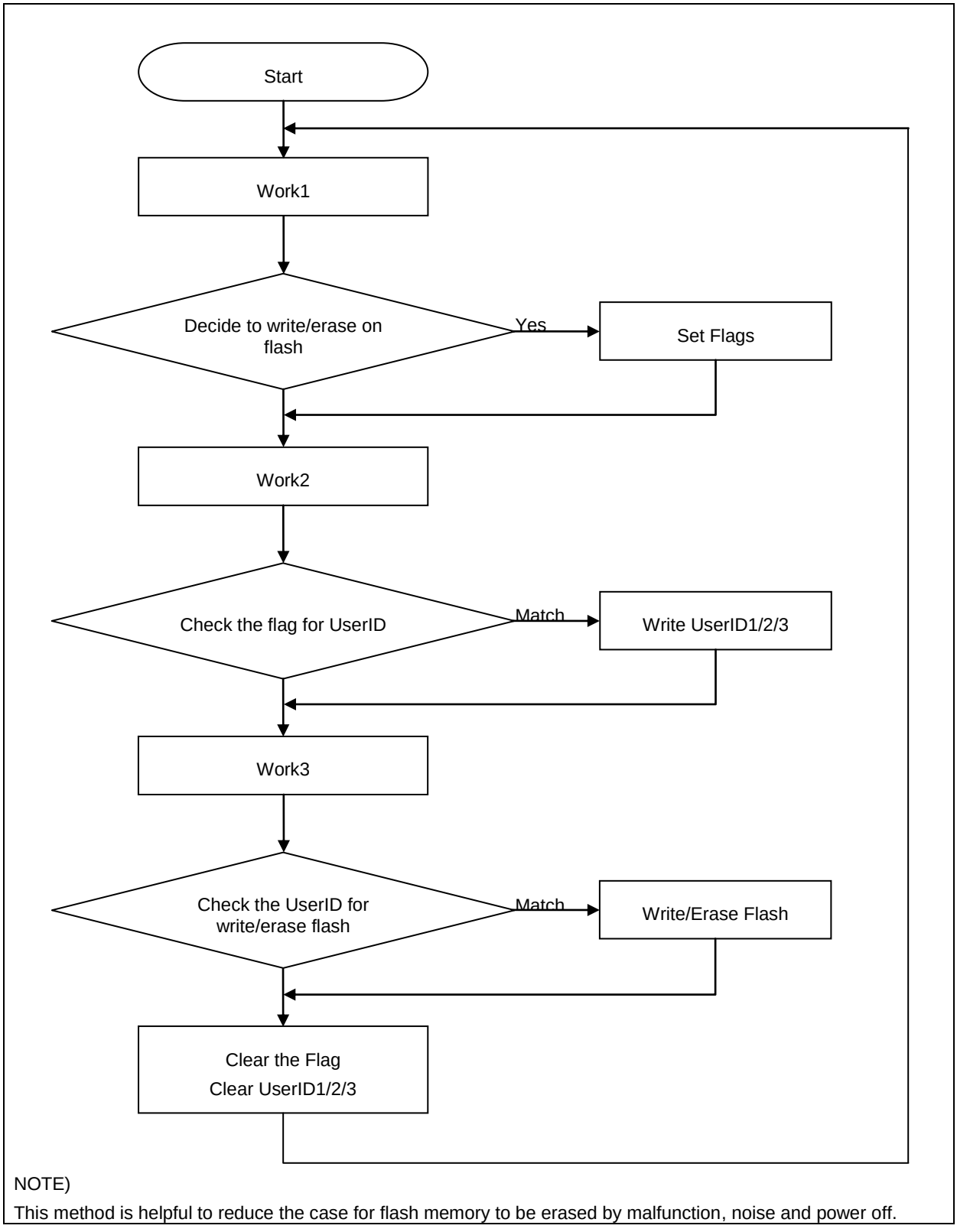
4. Overview of main

```
        ---
        CALL    Work1
        CALL    Decide_ErWt
        CALL    Work2
        CALL    ID_write
        CALL    Work3
        CALL    Flash_erase
        CALL    Flash_write
        ---
        ---
        ---

ID_wire:
        MOV     A,#38H
        CJNE    A,Flash_flag1,No_write_ID
        MOV     A,#75H
        CJNE    A,Flash_flag2,No_write_ID
        MOV     UserID1,#ID_DATA_1      ;Write Uiser ID1
        MOV     A,#38H
        CJNE    A,Flash_flag1,No_write_ID
        MOV     A,#75H
        CJNE    A,Flash_flag2,No_write_ID
        MOV     UserID2,#ID_DATA_2      ;Write Uiser ID2
        MOV     A,#38H
        CJNE    A,Flash_flag1,No_write_ID
        MOV     A,#75H
        CJNE    A,Flash_flag2,No_write_ID
        MOV     UserID3,#ID_DATA_3      ;Write Uiser ID3
        RET

No_write_ID:
        MOV     UserID1,#00H
        MOV     UserID2,#00H
        MOV     UserID3,#00H
        RET
```

15.1.10.1 Flow of Protection for Invalid Erase/Write



NOTE)

This method is helpful to reduce the case for flash memory to be erased by malfunction, noise and power off.

Figure 15.2 Flow of Protection for Invalid Erase/Write

15.1.11 Read Mode

The Reading program procedure in user program mode

1. Load receive data from flash memory on MOVC instruction by indirectly addressing mode.

Program Tip – reading

```
MOV     A,#0
MOV     DPH,#0x7F
MOV     DPL,#0x40           ;flash memory address

MOVC    A,@A+DPTR           ;read data from flash memory
```

15.1.12 Code Write Protection Mode

The Code Write Protection program procedure in user program mode

1. Set flash identification register (FIDR).
2. Set flash mode control register (FMCR).

Program Tip – Code Write Protection

```
MOV     FIDR,#0xA5          ;Identification value
MOV     FMCR,#0x04          ;Start flash Code Write Protection mode
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
NOP                         ;Dummy instruction, This instruction must be needed.
```

16. Configure Option

16.1 Configure Option Control

The data for configure option should be written in the configure option area (003EH – 003FH) by programmer (Writer tools).

CONFIGURE OPTION 1 : ROM Address 003FH

7	6	5	4	3	2	1	0
R_P	HL	VAPEN	–	–	–	–	RSTS

Initial value : 00H

R_P Code Read Protection
- 0 Disable
- 1 Enable

HL Code Write protection
- 0 Disable
- 1 Enable

VAPEN Vector Area (00H – FFH) Write Protection
- 0 Disable Protection (Erasable by instruction)
- 1 Enable Protection (Not erasable by instruction)

RSTS Select RESETB pin
- 0 Disable RESETB pin (P57/EC3)
- 1 Enable RESETB pin

CONFIGURE OPTION 2: ROM Address 003EH

7	6	5	4	3	2	1	0
–	–	–	–	–	PAEN	PASS1	PASS0

Initial value : 00H

PAEN Enable Specific Area Write Protection
- 0 Disable Protection (Erasable by instruction)
- 1 Enable Protection (Not erasable by instruction)

PASS [1:0] Select Specific Area for Write Protection

NOTE)

1. When PAEN = '1', it is applied.

PASS1	PASS0	Description
0	0	3.8k Bytes (Address 0100H – 0FFFH)
0	1	1.7k Bytes (Address 0100H – 07FFH)
1	0	768 Bytes (Address 0100H – 03FFH)
1	1	256 Bytes (Address 0100H – 01FFH)

17. APPENDIX

A. Instruction Table

Instructions are either 1, 2 or 3 bytes long as listed in the 'Bytes' column below.
Each instruction takes either 1, 2 or 4 machine cycles to execute as listed in the following table. 1 machine cycle comprises 2 system clock cycles.

ARITHMETIC				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ADD A,Rn	Add register to A	1	1	28-2F
ADD A,dir	Add direct byte to A	2	1	25
ADD A,@Ri	Add indirect memory to A	1	1	26-27
ADD A,#data	Add immediate to A	2	1	24
ADDC A,Rn	Add register to A with carry	1	1	38-3F
ADDC A,dir	Add direct byte to A with carry	2	1	35
ADDC A,@Ri	Add indirect memory to A with carry	1	1	36-37
ADDC A,#data	Add immediate to A with carry	2	1	34
SUBB A,Rn	Subtract register from A with borrow	1	1	98-9F
SUBB A,dir	Subtract direct byte from A with borrow	2	1	95
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	1	96-97
SUBB A,#data	Subtract immediate from A with borrow	2	1	94
INC A	Increment A	1	1	04
INC Rn	Increment register	1	1	08-0F
INC dir	Increment direct byte	2	1	05
INC @Ri	Increment indirect memory	1	1	06-07
DEC A	Decrement A	1	1	14
DEC Rn	Decrement register	1	1	18-1F
DEC dir	Decrement direct byte	2	1	15
DEC @Ri	Decrement indirect memory	1	1	16-17
INC DPTR	Increment data pointer	1	2	A3
MUL AB	Multiply A by B	1	4	A4
DIV AB	Divide A by B	1	4	84
DA A	Decimal Adjust A	1	1	D4

LOGICAL				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ANL A,Rn	AND register to A	1	1	58-5F
ANL A,dir	AND direct byte to A	2	1	55
ANL A,@Ri	AND indirect memory to A	1	1	56-57
ANL A,#data	AND immediate to A	2	1	54
ANL dir,A	AND A to direct byte	2	1	52
ANL dir,#data	AND immediate to direct byte	3	2	53
ORL A,Rn	OR register to A	1	1	48-4F
ORL A,dir	OR direct byte to A	2	1	45
ORL A,@Ri	OR indirect memory to A	1	1	46-47
ORL A,#data	OR immediate to A	2	1	44
ORL dir,A	OR A to direct byte	2	1	42
ORL dir,#data	OR immediate to direct byte	3	2	43
XRL A,Rn	Exclusive-OR register to A	1	1	68-6F
XRL A,dir	Exclusive-OR direct byte to A	2	1	65

XRL A, @Ri	Exclusive-OR indirect memory to A	1	1	66-67
XRL A,#data	Exclusive-OR immediate to A	2	1	64
XRL dir,A	Exclusive-OR A to direct byte	2	1	62
XRL dir,#data	Exclusive-OR immediate to direct byte	3	2	63
CLR A	Clear A	1	1	E4
CPL A	Complement A	1	1	F4
SWAP A	Swap Nibbles of A	1	1	C4
RL A	Rotate A left	1	1	23
RLC A	Rotate A left through carry	1	1	33
RR A	Rotate A right	1	1	03
RRC A	Rotate A right through carry	1	1	13

DATA TRANSFER				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOV A,Rn	Move register to A	1	1	E8-EF
MOV A,dir	Move direct byte to A	2	1	E5
MOV A,@Ri	Move indirect memory to A	1	1	E6-E7
MOV A,#data	Move immediate to A	2	1	74
MOV Rn,A	Move A to register	1	1	F8-FF
MOV Rn,dir	Move direct byte to register	2	2	A8-AF
MOV Rn,#data	Move immediate to register	2	1	78-7F
MOV dir,A	Move A to direct byte	2	1	F5
MOV dir,Rn	Move register to direct byte	2	2	88-8F
MOV dir,dir	Move direct byte to direct byte	3	2	85
MOV dir,@Ri	Move indirect memory to direct byte	2	2	86-87
MOV dir,#data	Move immediate to direct byte	3	2	75
MOV @Ri,A	Move A to indirect memory	1	1	F6-F7
MOV @Ri,dir	Move direct byte to indirect memory	2	2	A6-A7
MOV @Ri,#data	Move immediate to indirect memory	2	1	76-77
MOV DPTR,#data	Move immediate to data pointer	3	2	90
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	2	93
MOVC A,@A+PC	Move code byte relative PC to A	1	2	83
MOVX A,@Ri	Move external data(A8) to A	1	2	E2-E3
MOVX A,@DPTR	Move external data(A16) to A	1	2	E0
MOVX @Ri,A	Move A to external data(A8)	1	2	F2-F3
MOVX @DPTR,A	Move A to external data(A16)	1	2	F0
PUSH dir	Push direct byte onto stack	2	2	C0
POP dir	Pop direct byte from stack	2	2	D0
XCH A,Rn	Exchange A and register	1	1	C8-CF
XCH A,dir	Exchange A and direct byte	2	1	C5
XCH A,@Ri	Exchange A and indirect memory	1	1	C6-C7
XCHD A,@Ri	Exchange A and indirect memory nibble	1	1	D6-D7

BOOLEAN				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
CLR C	Clear carry	1	1	C3
CLR bit	Clear direct bit	2	1	C2
SETB C	Set carry	1	1	D3
SETB bit	Set direct bit	2	1	D2
CPL C	Complement carry	1	1	B3
CPL bit	Complement direct bit	2	1	B2

ANL C,bit	AND direct bit to carry	2	2	82
ANL C,/bit	AND direct bit inverse to carry	2	2	B0
ORL C,bit	OR direct bit to carry	2	2	72
ORL C,/bit	OR direct bit inverse to carry	2	2	A0
MOV C,bit	Move direct bit to carry	2	1	A2
MOV bit,C	Move carry to direct bit	2	2	92

BRANCHING				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
ACALL addr 11	Absolute jump to subroutine	2	2	11→F1
LCALL addr 16	Long jump to subroutine	3	2	12
RET	Return from subroutine	1	2	22
RETI	Return from interrupt	1	2	32
AJMP addr 11	Absolute jump unconditional	2	2	01→E1
LJMP addr 16	Long jump unconditional	3	2	02
SJMP rel	Short jump (relative address)	2	2	80
JC rel	Jump on carry = 1	2	2	40
JNC rel	Jump on carry = 0	2	2	50
JB bit,rel	Jump on direct bit = 1	3	2	20
JNB bit,rel	Jump on direct bit = 0	3	2	30
JBC bit,rel	Jump on direct bit = 1 and clear	3	2	10
JMP @A+DPTR	Jump indirect relative DPTR	1	2	73
JZ rel	Jump on accumulator = 0	2	2	60
JNZ rel	Jump on accumulator ≠ 0	2	2	70
CJNE A,dir,rel	Compare A,direct jne relative	3	2	B5
CJNE A,#d,rel	Compare A,immediate jne relative	3	2	B4
CJNE Rn,#d,rel	Compare register, immediate jne relative	3	2	B8-BF
CJNE @Ri,#d,rel	Compare indirect, immediate jne relative	3	2	B6-B7
DJNZ Rn,rel	Decrement register, jnz relative	2	2	D8-DF
DJNZ dir,rel	Decrement direct byte, jnz relative	3	2	D5

MISCELLANEOUS				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
NOP	No operation	1	1	00

ADDITIONAL INSTRUCTIONS (selected through EO[7:4])				
Mnemonic	**Description**	**Bytes**	**Cycles**	**Hex code**
MOVC @(DPTR++),A	M8051W/M8051EW-specific instruction supporting software download into program memory	1	2	A5
TRAP	Software break command	1	1	A5

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11→F1 (for example), are used for absolute jumps and calls, with the top 3 bits of the code being used to store the top three bits of the destination address.

The CJNE instructions use the abbreviation #d for immediate data; other instructions use #data.